



Documentation for Crucible 3.9

Contents

Getting Started	5
Supported platforms	5
Installing Crucible on Windows	8
Installing Crucible on Linux and Mac	11
Configuring JIRA Integration in the Setup Wizard	14
Starting to use Crucible	21
Installing and upgrading Git	29
Using Crucible	32
Using the Crucible screens	32
Browsing all reviews	34
Browsing source files	36
Browsing projects	38
Changing your User Profile	39
Using favourites	39
Using Wiki Markup in Crucible	41
Using RSS feeds in Crucible	46
Using keyboard shortcuts in Crucible	46
Crucible icons	47
The Crucible workflow	52
Defining your workflow	57
Roles and status classifications	58
Creating a review	59
Creating a review from JIRA	63
Creating a review from a URL	64
Creating a Snippet Review	64
Creating reviews from the command line	65
Adding content to the review	67
Iterative reviews	74
Creating patch files for pre-commit reviews	77
Choosing reviewers	82
Performing the review	84
Starting a review	85
Commenting on reviews	86
Sending a review's comments via email	89
Changeset discussions	89
Flagging defects	90
Viewing reports	91
Review Coverage report	93
Completing your review	97
Using the Review History Dialog	99
Summarising and closing the review	100
Managing your reviews	102
Using Review Reminders	102
Moving a review to another project	104
Using Progress Tracking	105
Using Time Tracking	107
Viewing people's statistics in Crucible	108
Viewing Project Statistics	112
Deleting a review	114
Searching Crucible	115
JIRA integration in Crucible	120
Creating JIRA issues from the review	124
Transitioning JIRA issues	125
Administering Crucible	126
Best practices for Crucible configuration	126
Crucible and FishEye	127

Administering projects	128
Creating a project	129
Storing all revisions under review	130
Enabling the moderator role	131
Setting default review objectives	131
Editing a project	132
Configuring repositories	133
Setting up a Git repository in Crucible	134
Setting up a Perforce repository in Crucible alone	136
Setting up a Subversion repository in Crucible alone	137
Enabling reviews from the server file system in Crucible	138
Setting up a repository via FishEye	139
Configuring commit hooks	139
Setting up users and security	144
Creating a user	144
Permissions	145
Creating a permission scheme	145
Associating a permission scheme with a project	148
Agile permissions schemes in Crucible	150
Migrating to an external database	151
Migrating to MySQL	152
Migrating to Oracle	156
Migrating to PostgreSQL	158
Migrating to SQL Server	161
Backing up and restoring Crucible data	164
Customising Crucible	171
Customising the welcome message	171
Customising email notifications	172
Freemarker Data Model for Email Templates	174
Customising the defect classifications	174
Configuring User Managed Mappings	177
Enabling Access Logging in Crucible	177
Linking Crucible to JIRA	179
User management limitations and recommendations	182
Supported fields for inline issue creation	184
Using Crucible gadgets	186
Linking to another application	188
Running Crucible as a Windows service	188
Managing add-ons	192
Crucible releases	193
End of Support Announcements for Crucible	199
End of Support Announcement for IBM ClearCase	202
Crucible upgrade guide	202
Upgrading from FishEye to Crucible	217
Crucible 3.9 release notes	217
Crucible 3.8 release notes	220
Crucible 3.7 release notes	221
Crucible 3.6 release notes	224
Crucible 3.5 release notes	227
Crucible 3.4 release notes	231
Crucible 3.3 release notes	235
Crucible 3.2 release notes	238
Crucible 3.1 release notes	241
Crucible 3.0 release notes	246
Security advisories	250
Crucible Security Advisory 2015-01-21	251
FishEye and Crucible Security Advisory 2014-05-21	252
FishEye and Crucible Security Advisory 2014-02-26	253
FishEye and Crucible Security Advisory 2013-07-16	254
FishEye and Crucible Security Advisory 2012-08-21	255
FishEye and Crucible Security Advisory 2012-05-17	256
FishEye and Crucible Security Advisory 2012-01-31	257

FishEye and Crucible Security Advisory 2011-11-22	259
FishEye and Crucible Security Advisory 2011-05-16	260
FishEye and Crucible Security Advisory 2011-01-12	262
Crucible Security Advisory 2010-06-16	265
Crucible Security Advisory 2010-05-04	267
Crucible FAQ	270
Crucible Resources	272
General FAQs	272
Can I deploy Crucible or FishEye as a WAR?	272
Does Crucible support SSL (HTTPS)?	273
How do I force reviews to include SVN property changes?	273
How to Automate Daily Crucible Backups	273
Licensing FAQ	273
What happens if I decide to stop using FishEye with Crucible	274
Do I need a FishEye licence to run Crucible?	275
Updating your Crucible license	275
Support Policies	276
Bug Fixing Policy	276
New Features Policy	277
Security Bugfix Policy	278
Troubleshooting	278
JIRA Integration Issues	278
Problems with very long comments and MySQL migration	279
Contributing to the Crucible documentation	279
Glossary	280
approve	281
authors in Crucible	281
code review	281
comment	281
creator	281
defect	281
moderator	281
participant	281
permission scheme	281
permissions in Crucible	282
projects in Crucible	283
review duration	283
reviewer	283
role	283
state	283
statement of objective	284
users in Crucible	284
Collecting analytics in Crucible	284

Getting Started

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

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

System requirements

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

Download and install Crucible

- [Windows](#)
- [Mac](#)
- [Linux](#)

Start using Crucible

For a short introduction see [Starting to use Crucible](#). You'll learn how to:

- [Add a repository](#)
- [Create a project](#)
- [Create and perform reviews](#)

Integrate Crucible with other Atlassian applications

As a first step, see [JIRA integration in Crucible](#).

Read more about using Crucible

To find out more about using Crucible with your team, see [Using Crucible](#).

To find out how to manage the Crucible server, see [Administering Crucible](#).

Supported platforms

This page lists the supported platforms for **Crucible 3.9.x** and its minor releases.

Key:  = Supported  = Deprecated  = Not Supported

Java		
Oracle JRE / JDK	 1.8  1.7	Crucible requires the Java Runtime (JDK or JRE), version as noted. Pre-release/Early access versions of Java are <i>not supported</i>. We highly recommend that you use the Oracle JVM (or OpenJDK for Linux only). Other Java implementations have not been tested. You can download an Oracle Java Runtime. For the OpenJDK, download and install

OpenJDK	 1.8 (Linux only)	instructions for Linux flavours are at http://openjdk.java.net/install/. Please note: - Once you have installed the Java, you need to set the JAVA_HOME environment variable. See Installing Crucible on Windows or Installing Crucible on Linux and Mac for details. - If you are using a 64-bit JVM, please ensure that you've set your max heap size (-Xmx) to a reasonable value, considering the RAM requirements of your system. - On OS X Java 7 performs significantly better than Java 6.  Support for Java 6 was removed in Crucible 3.9, as previously announced.
Operating Systems		
Microsoft Windows		Crucible is a pure Java application and should run on any platform provided the requirements for the JRE or JDK are satisfied. Although Crucible can be run in virtualised environments, Atlassian is not yet able to provide technical support for performance-related problems in a virtualised environment. If you do chose to run Crucible in a VM, please ensure that you choose a VM with good IO throughput.
Linux		
Apple Mac OS X		
Databases		
HSQLDB	 Bundled; for evaluation use only	The Crucible built-in database, running HSQLDB, is somewhat susceptible to data loss during system crashes. We recommend that you do not use HSQLDB for production systems. External databases are generally more resistant to data loss during a system crash and are more suited for production use. See the Crucible Database documentation for further details.
MySQL	 MySQL Enterprise Server 5.1+  MySQL Community Server 5.1+  MySQL 5.0  Maria, Percona	For MySQL:  For 5.1, versions earlier than 5.1.10 are not supported  For 5.6, versions earlier than 5.6.11 are not supported  For 5.7, versions earlier than 5.7.5 are not supported  Support for MySQL 5.0 was removed in Crucible 3.3. See End of Support

PostgreSQL	✓ 9.0, 9.1, 9.2, 9.3 ✓ 8.3, 8.4 ✗ 8.2	Announcements for Crucible. ✗ MariaDB and Percona variants of MySQL are not supported, and are known to cause issues when used with Crucible.
Oracle	✓ 12c ✓ 11g	✗ Support for PostgreSQL 8.2 was removed in Crucible 3.3. See End of Support Announcements for Crucible.
SQL Server	✓ 2012 ✓ 2008, 2008 R2 ✗ 2005	✗ Support for SQL Server 2005 was removed in Crucible 3.3. See End of Support Announcements for Crucible.
Web browsers		
Microsoft Internet Explorer	✓ 10.0, 11.0 ✗ 9.0	Minimum screen resolution of 1024x768. Recommended screen resolution of 1280x768 or above. ✗ Support for Internet Explorer 9 was removed in Crucible 3.9. See End of Support Announcements for Crucible.
Mozilla Firefox	✓ Latest stable version supported ✗ 3.6, 4.0	✗ Support for Firefox 3.6 and 4.0 was removed in Crucible 3.7.
Safari	✓ Latest stable version supported ✗ 4, 5	✗ Support for Safari 4 and 5 was removed in FishEye 3.7.
Chrome	✓ Latest stable version supported	
Version Control Systems		
Subversion (SVN)	Server: ✓ 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8 Client: ✓ SVNKit (bundled & the default) ✓ Native JavaHL 1.8 ✓ Native JavaHL 1.7 ✗ Native JavaHL 1.6	✗ Crucible 3.1, and later, do not support the native JavaHL 1.6 client. See Native support for SVN for discussion.
CVS (and CVSNT)	✓ All versions	
Perforce	✓ Client version 2007.3 or later ✓ Server version 2005.1 or later	The Server must support the ztag tagged protocol. ✗ Perforce Streams, introduced in 2011.1, is not currently supported. See FE-3886 - support for Streams in p4 OPEN

Git	✔ 2.5.0 ✔ 2.4.6 ✔ 2.2.1, 2.2.2 ✔ 2.1.4 ✔ 2.0.5 ✔ 1.9.5 ✔ 1.8.0.3, 1.8.1.5, 1.8.2.3, 1.8.3.4, 1.8.4.5, 1.8.5.6 ✔ 1.7.1.1, 1.7.2.5, 1.7.3.5, 1.7.4.5, 1.7.5.4, 1.7.6.6, 1.7.7.7, 1.7.8.6, 1.7.9.7, 1.7.10.5, 1.7.11.7, 1.7.12.4	These are the versions of Git that we currently test Crucible against. ✖ Git 1.8.4.3 is not supported. See STASH-4101 - Clone and fetch fail with "protocol error: impossibly long line" CLOSED [Security vulnerability CVE-2014-9390] affects multiple Git versions. Crucible itself is not affected, however you should update your <i>clients</i> to a patched maintenance version: v1.8.5.6, v1.9.5, v2.0.5, v2.1.4 and v2.2.1 or newer.
Mercurial	✔ 3.0.2, 3.1.2, 3.2.4, 3.4.2, 3.5.0 ✔ 2.0.2, 2.1.2, 2.2.3, 2.3.2, 2.4, 2.5.2, 2.6.3, 2.7.2, 2.8.2, 2.9.1 ✔ 1.5.1, 1.5.4, 1.6.4, 1.7.5, 1.8.4, 1.9.3	These are the versions of Mercurial that we currently test Crucible against. As of version 3.6.3, Crucible supports Mercurial 3. ✖ Mercurial 2.1 has a bug that makes it incompatible with Crucible. Please use Mercurial 2.1.1 or later. You should restart Crucible after upgrading Mercurial.
Atlassian applications		
Crowd	✔ Crowd 2.4.x+ ✔ Crowd client library: 2.4.1	From version 2.8.x, FishEye bundles the Crowd 2.4.1 client library, and supports the Crowd 2.4.x server, and later versions.
JIRA	✔ JIRA FishEye Plugin 6.1.0+	The JIRA FishEye Plugin is bundled with JIRA. If you are using earlier versions of JIRA you can upgrade the plugin in JIRA to get support for Crucible.

Deployment Notes for Source Code Repositories

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

WAR deployment

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

Single sign on with Atlassian Crowd

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

Installing Crucible on Windows

This page...

... describes how to perform a

Upgrading?

If you're upgrading your

Upgrading from FishEye?

If you already have FishEye

clean install of Crucible on Windows.

Crucible installation, read the [Crucible upgrade guide](#) first.

installed, you should read [Upgrading from FishEye to Crucible](#) instead.

1. Check supported platforms

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

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

2. Create a dedicated Crucible user (recommended)

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

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

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

3. Check your version of Java

In a command prompt, run this:

```
java -version
```

The version of Java should be **1.6.0 – 1.7.x** (or **1.7.x** for OpenJDK). 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 a supported version of Java, 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 – 1.7.x** (or **1.7.x** for OpenJDK).

4. Check that Windows can find Java

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

```
echo %JAVA_HOME%
```

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

▼ [If you don't see a path without spaces...](#)

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

For Windows 7:

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

5. Now it's time to get Crucible

Download the [Crucible installer](#) from the Atlassian download site.

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

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

6. Visit Crucible!

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

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

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

7. Connect to an external database (recommended)

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

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

8. Set up your mail server

Configure the Crucible email server so that users can get notifications from Crucible. See [Configuring SMTP](#).

9. Add users and repositories

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

Crucible will perform an initial index of your repositories, during which it accesses, indexes and 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.

10. Stop Crucible (optional)

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

```
bin\stop.bat
```

Installing Crucible on Linux and Mac

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

1. Check supported platforms

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

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

Related pages:

- [Installing Crucible on Windows](#)
- [Starting to use Crucible](#)
- [Supported platforms](#)

2. Create a dedicated Crucible user (recommended)

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

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

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

3. Check your version of Java

In a terminal, run this:

```
java -version
```

The version of Java should be **1.8.x** (or **1.7.x** for OpenJDK).

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

Download and install the [Oracle Java Platform JDK](#), or [OpenJDK](#).

Now try running 'java -version' again to check the installation. The version of Java should be **1.8.x** (or **1.7.x** for OpenJDK).

4. Check that the system can find Java

In a terminal, run this:

```
echo $JAVA_HOME
```

You should see a path something like:

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

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

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

5. Now it's time to get Crucible

1. [Download Crucible](#) from the Atlassian download site.
2. Please check your unzip program before extracting the downloaded zip file. Some archive-extract

programs cause errors when unzipping the Crucible zip file:

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

6. Tell Crucible where to store your data

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

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

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

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

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

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

7. Start Crucible!

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

```
bin/start.sh
```

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

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

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

8. Connect to an external database (recommended)

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

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

9. Set up your mail server

Configure the Crucible email server so that users can get notifications from Crucible. See [Configuring SMTP](#).

10. Add users and repositories

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

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

11. Stop Crucible (optional)

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

```
bin/stop.sh
```

Configuring JIRA Integration in the Setup Wizard

This page describes the 'Connect to JIRA' screen of the Crucible setup wizard.

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

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

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

Requirements: You need JIRA 4.3 or later.

On this page:

- [Connecting to JIRA in the Setup Wizard](#)
- [Troubleshooting](#)
- [Notes](#)

Related pages:

- [Linking Crucible to JIRA](#)
- [User management limitations and recommendations](#)
- [JIRA integration in Crucible](#)

Connecting to JIRA in the Setup Wizard

To configure JIRA integration while running the Crucible setup wizard:

1. Enter the following information on the '**Connect to JIRA**' step of the setup wizard:

- **JIRA Base URL**– The web address of your JIRA server. Examples:

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

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

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

Troubleshooting

- ✓ [Click to see troubleshooting information...](#)

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

Symptom	Cause	Solution
---------	-------	----------

The setup wizard displays one of the following error messages: - Failed to create application link from JIRA server at <URL> to this <application> server at <URL>. - Failed to create application link from this <application> server at <URL> to JIRA server at <URL>. - Failed to authenticate application link from JIRA server at <URL> to this <application> server at <URL>. - Failed to authenticate application link from <application> server at <URL> to this JIRA server at <URL>.	The setup wizard failed to complete registration of the peer-to-peer application link with JIRA. JIRA integration is only partially configured.	Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.
The setup wizard displays one of the following error messages: - Failed to register <application> configuration in JIRA for shared user management. Received invalid response from JIRA: <response> - Failed to register <application> configuration in JIRA for shared user management. Received: <response>	The setup wizard failed to complete registration of the client-server link with JIRA for user management. The peer-to-peer link was successfully created, but integration is only partially configured.	Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.
The setup wizard displays the following error message: Error setting Crowd authentication	The setup wizard successfully established the peer-to-peer link with JIRA, but could not persist the client-server link for user management in your config.xml file. This may be caused by a problem in your environment, such as a full disk.	Please investigate and fix the problem that prevented the application from saving the configuration file to disk. Then remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.
The setup wizard displays the following error message: Error reloading Crowd authentication	The setup wizard has completed the integration of your application with JIRA, but is unable to start synchronizing the JIRA users with your application.	Restart your application. You should then be able to continue with the setup wizard. If this solution does not work, please contact Atlassian Support.
The setup wizard displays the following error message: An error occurred: java.lang.IllegalStateException: Could not create the application in JIRA/Crowd (code: 500). Please refer to the logs for details.	The setup wizard has not completed the integration of your application with JIRA. The links are only partially configured. The problem occurred because there is already a user management configuration in JIRA for this <application> URL.	Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.

No users can log in after you have set up the application with JIRA integration.	Possible causes: • There are no users in the group that you specified on the 'Connect to JIRA' screen. • For FishEye: There are no groups specified in the 'groups to synchronize' section of your administration console. • For Stash: You may not have granted any JIRA groups or users permissions to log in to Stash.	Go to JIRA and add some usernames to the group. • For FishEye: Go to the FishEye administration screens and specify at least one group to synchronize. The default is 'jira-users'. • For Stash: Grant the Stash User permission to the relevant JIRA groups on the Stash Global permissions page. If this solution does not work, please contact Atlassian Support.
--	--	---

Solution 1: Removing a Partial Configuration – The Easiest Way

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

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

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

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

For example:

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

Or:

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

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

- Go to the following URL in your browser:

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

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

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

- The application links manifest will appear. Check the application ID in the `<id>` element.
- c. In JIRA, click '**Delete**' next to the application that you want to remove.
- 5. Go back to the setup wizard and try the 'Connect to JIRA' step again.

Solution 2: Removing a Partial Configuration – The Longer Way

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

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

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

For example:

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

Or:

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

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

- Go to the following URL in your browser:

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

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

For example:

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

- The application links manifest will appear. Check the application ID in the `<id>` element.
 - c. In JIRA, click **Delete** next to the application that you want to remove.
6. Add the application link in JIRA again, so that you now have a two-way trusted link between JIRA and your application:
 - a. Click **Add Application Link**. Step 1 of the link wizard will appear.
 - b. Enter the **server URL** of the application that you want to link to (the 'remote application').
 - c. Click **Next**.
 - d. Enter the following information:
 - **Create a link back to this server** – Check to add a two-way link between the two applications.
 - **Username and Password** – Enter the credentials for a username that has administrator access to the remote application.
Note: These credentials are only used to authenticate you to the remote application, so that Application Links can make the changes required for the new link. The credentials are not saved.
 - **Reciprocal Link URL** – The URL you give here will override the base URL specified in your remote application's administration console, for the purposes of the application links connection. Application Links will use this URL to access the remote application.
 - e. Click **Next**.
 - f. Enter the information required to configure authentication for your application link:
 - **The servers have the same set of users** – Check this box, because the users are the same in both applications.
 - **These servers fully trust each other** – Check this box, because you trust the code in both applications and are sure both applications will maintain the security of their private keys.
For more information about configuring authentication, see [Configuring Authentication for an Application Link](#).
 - g. Click **Create**.
 7. Configure a new connection for user management in JIRA:
 - a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:
 - In JIRA 4.3: Click **Other Applications** in the **Users, Groups & Roles** section of the JIRA administration screen.
 - In JIRA 4.4: Select **Administration** > **Users** > **JIRA User Server**.
 - b. **Add** an application.
 - c. Enter the **application name** and **password** that your application will use when accessing JIRA.

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


```
http://www.example.com:8080/jira/
http://jira.example.com
```
 - **Application name** and **Application password** – Enter the values that you defined for Confluence in the settings on JIRA.
 - e. Save the directory settings.
 - f. Define the **directory order** by clicking the blue up- and down-arrows next to each directory on the '**User Directories**' screen.
For details see [Connecting to Crowd or JIRA for User Management](#).
 - For FishEye/Crucible:
 - a. Click **Authentication** (under 'Security Settings').
 - b. Click **Setup JIRA/Crowd authentication**. Note, if LDAP authentication has already been set up, you will need to remove that before connecting to JIRA for user management.
 - c. Make the following settings:

Authenticate against	Select a JIRA instance
Application name and password	Enter the values that you defined for your application in the settings on JIRA.
JIRA URL	The web address of your JIRA server. Examples: <pre>http://www.example.com:8080/jira/ http://jira.example.com</pre>
Auto-add	Select Create a FishEye user on successful login so that your JIRA users will be automatically added as a FishEye user when they first log in.
Periodically synchronise users with JIRA	Select Yes to ensure that JIRA will synchronize all changes in the user information on a regular basis. Change the value for Synchronise Period if required.
When Synchronisation Happens	Select an option depending on whether you want to allow changes to user attributes from within FishEye.
Single Sign On	Select Disabled . SSO is not available when using JIRA for user management and if enabled will make the integration fail.

- d. Click **Next** and select at least one user group to be synchronised from JIRA. If necessary, you could create a new group in JIRA, such as 'fisheye-users', and select this group here.
 - e. Click **Save**.
 - For Stash:
 - a. Go to the Stash administration area.
 - b. Click **User Directories** in the left-hand panel.
 - c. **Add** a directory and select type **Atlassian JIRA**.
 - d. Enter the following information:
 - **Name** – Enter the name of your JIRA server.
 - **Server URL** – Enter web address of your JIRA server. Examples:

```
http://www.example.com:8080/jira/  
http://jira.example.com
```
 - **Application name** and **Application password** – Enter the values that you defined for Stash in the settings on JIRA.
 - e. Save the directory settings.
 - f. Define the directory order by clicking the blue up- and down-arrows next to each directory on the 'User Directories' screen.
- For details see [Connecting Stash to JIRA for user management](#).

Having trouble integrating your Atlassian products with Application Links?

We've developed a [guide to troubleshooting Application Links](#), to help you out. Take a look at it if you need help getting around any errors or roadblocks.

Notes

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

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

Starting to use Crucible

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

- [Add a repository](#)
- [Create a project](#)
- [Create and perform reviews](#)

For more information, see the [Crucible user's guide](#).

Assumptions

We're assuming that:

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

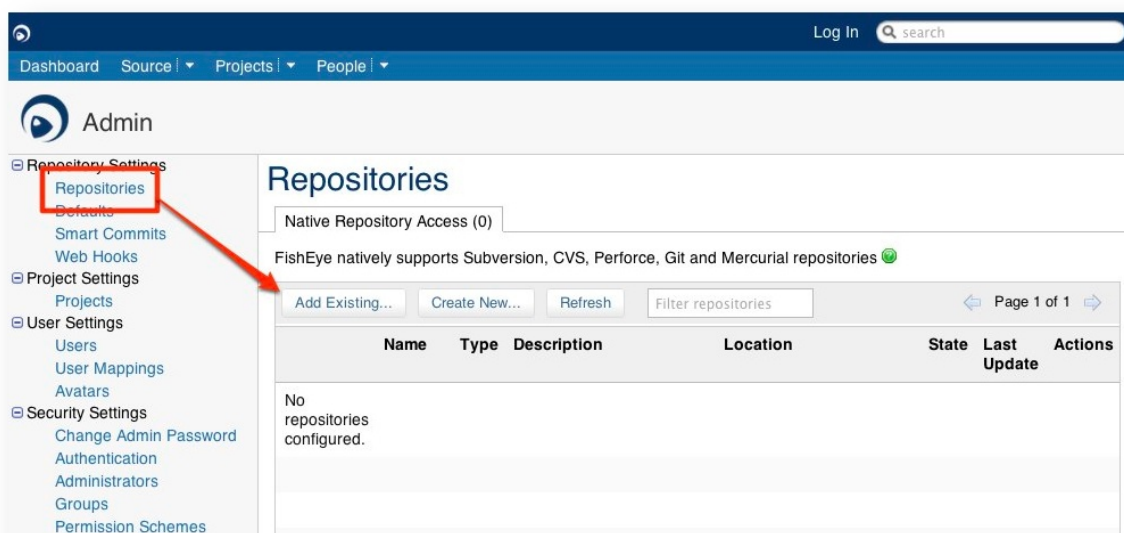
Add a repository

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

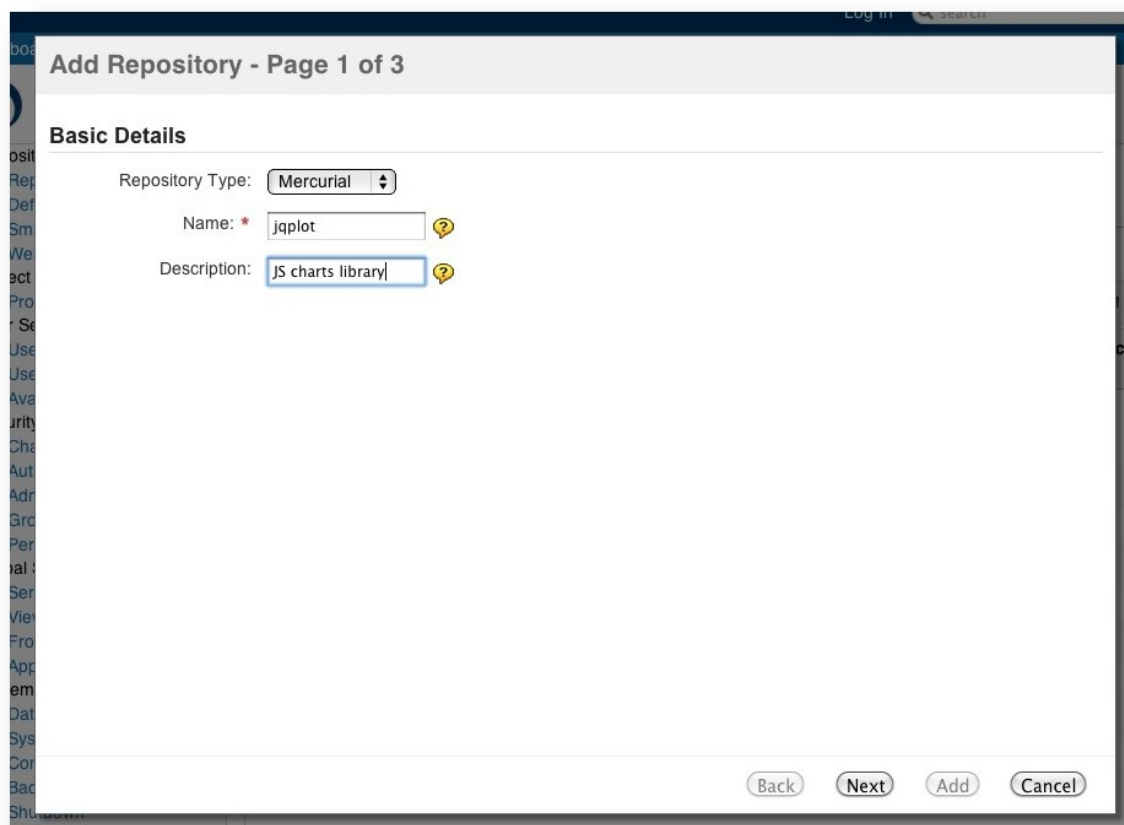
Go to the Admin area by clicking on the 'cog' at the top right and choosing **Administration**:



Click 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.

Add Repository - Page 2 of 3

Mercurial Connection Details

Repository Location: *

Mercurial Authentication

Authentication Style:

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

☐ Show advanced settings

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

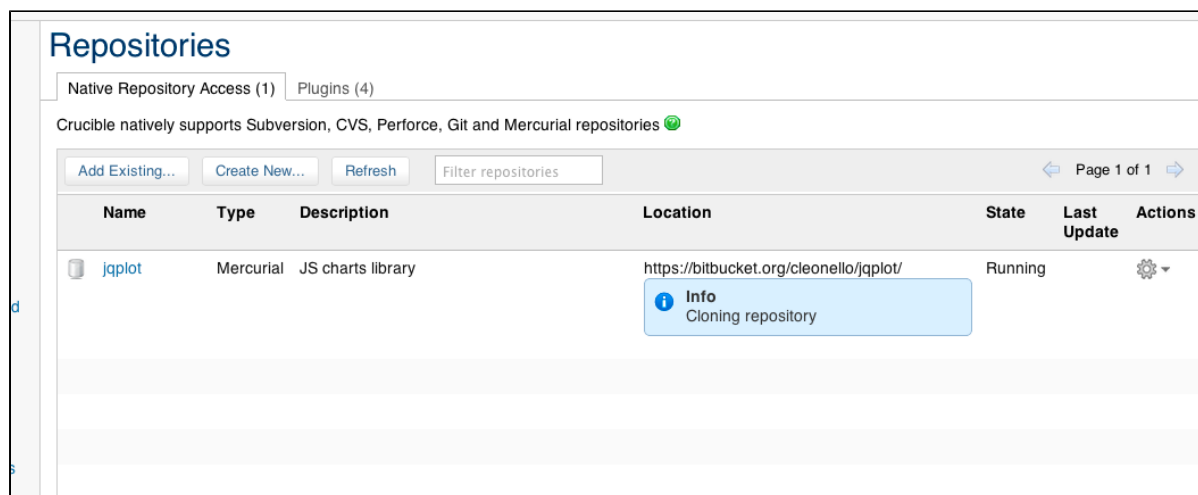
Add Repository - Page 3 of 3

Final Settings

Store Diff Info: ☐

Enable Repository After Adding: ☒

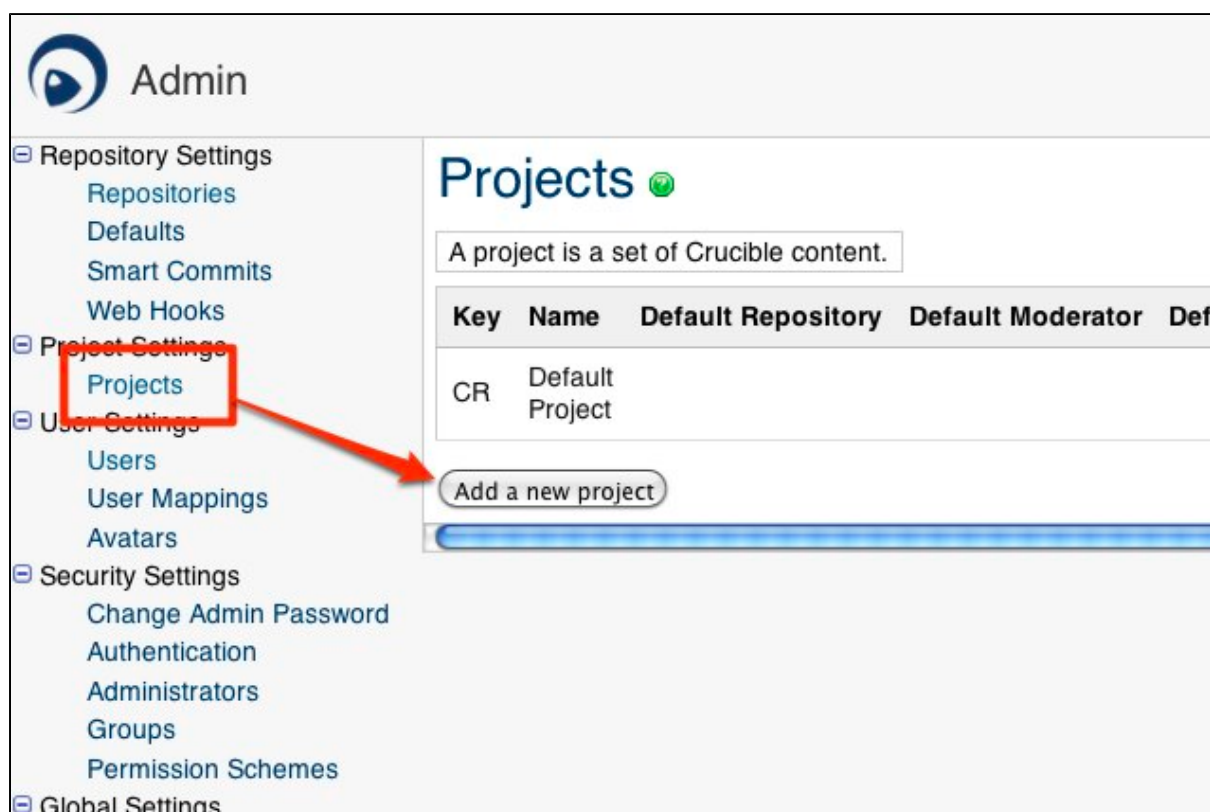
Your repository is now created in **Crucible** and the indexing should have started.



Create a project in Crucible

Crucible comes with a default project, with the key CR, but you will probably want to create your own projects to 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**.

Edit Project

Identification

Name

Key

Content

Default Repository

☒ Store the contents of files in reviews

Permissions Scheme 

Permission Scheme

Moderator 

☒ Enable the Moderator role for this project

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

Default Reviewers 

☐ By default, allow anyone to join reviews after creation

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

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

 Admin

- Repository Settings
 - Repositories
 - Defaults
 - Smart Commits
 - Web Hooks
- Project Settings
 - Projects**
- User Settings
 - Users
 - User Mappings
 - Avatars
- Security Settings
 - Change Admin Password
 - Authentication
 - Administrators
 - Groups
 - Permission Schemes
- Global Settings

Projects

A project is a set of Crucible content.

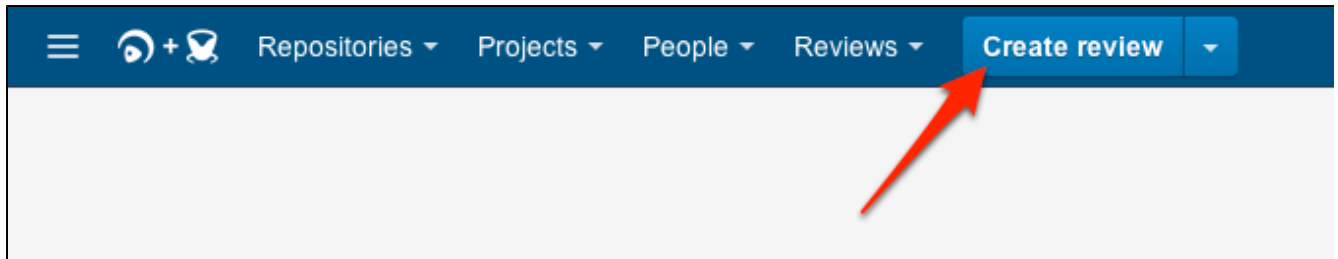
Key	Name	Default Repository	Default Moderator	Default Reviewer Users	Default Reviewer
CR	Default Project				
JQR	JQ Reviews	jqplot			

Create a review

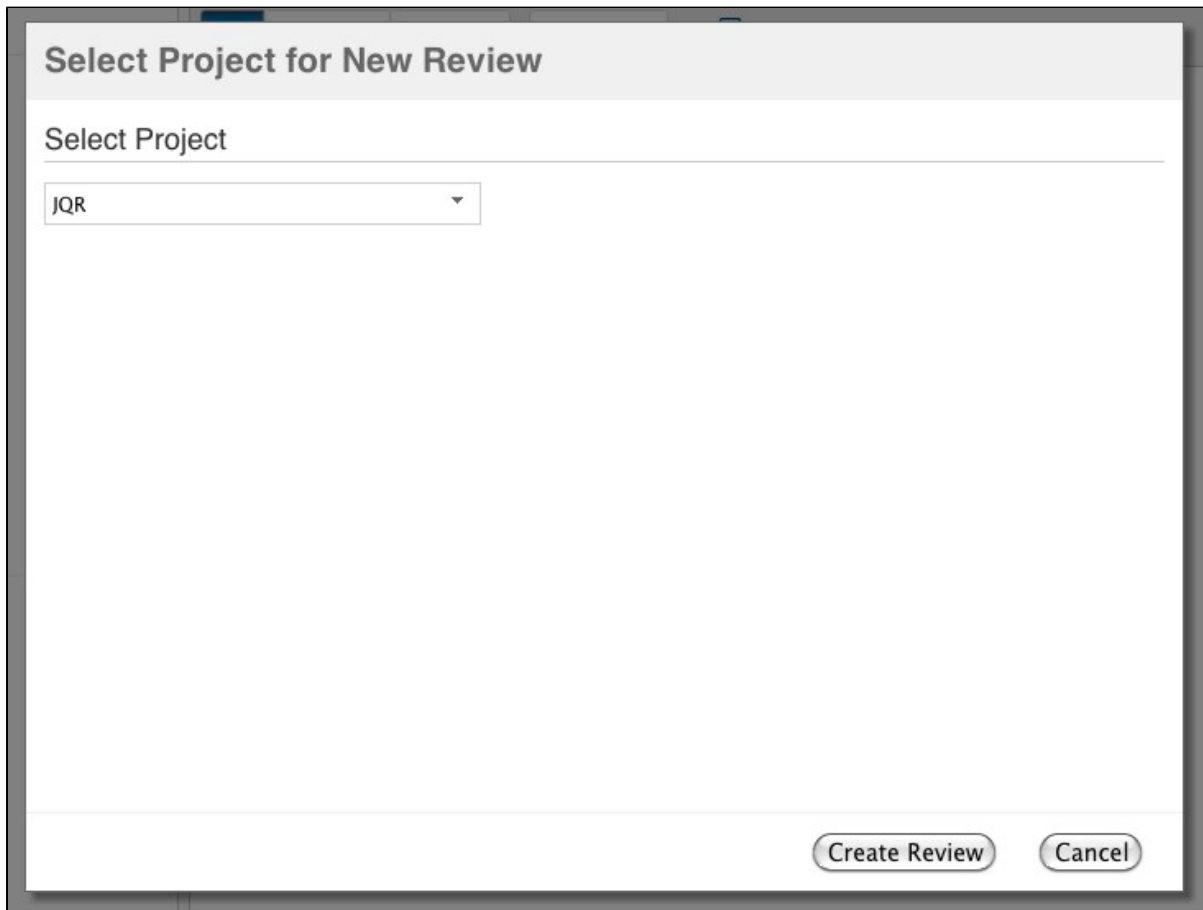
Now that you have your own project you can create reviews in it.

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 for the review.

Add Content to Review JQR-1

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

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](#)

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

Add Content to Review JQR-3

Repository: Author:
Branch: Tag:
Add to Review as: [Remove all revisions from review](#)

[Earlier Changesets](#) Go to changeset: [No More Changesets](#)

Changes in jqplot

Monday 13 Aug

☒ Chris Leonello <chris.leonello@gmail.com> 7:58 am
1118:0d4d1a4fe522 changed 1 file default [JQR-1](#) | [JQR-2](#)
Fix for horizontal waterfall plots.

Friday 10 Aug

☒ Chris Leonello <chris.leonello@gmail.com> 7:59 am
1117:658794324deb changed 1 file default [JQR-1](#) | [JQR-2](#)
Fixed example.

☐ 1116:6e69c9b6db6a changed 47 files default 7:51 am
Updated version and copyright.

☐ 1115:9e15fb8b84ed changed 1 file default 7:43 am
Fixed some lint.

☐ 1114:da13ae5eb3b2 changed 2 files default 4:29 am

[Add More Content](#) [Edit Details](#) [Abandon Review](#) [Start Review](#) [Done](#)

Once you have chosen your reviewers and updated the review information you can click on **Start Review** to begin the review process.

Edit Review Details JQR-3

Project: JQR

Title: Fix for horizontal waterfall plots.

Author: John Doe Moderator: John Doe

Reviewers: [Suggest reviewers...](#) ☐ Allow anyone to join

- Sean Carter
- Donald Glover

Objectives:

Due Date:

Linked Review: [Link](#)

Invited Reviewers: [Invite](#)

[Add Content](#) [Abandon Review](#) [Start Review](#) [Done](#)

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

JQR-3

Fix for horizontal waterfall plots.

Under Review for a few seconds

Author & Moderator: John Doe

Reviewers: Donald Glover, Sean Carter

Details

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

Files: 2

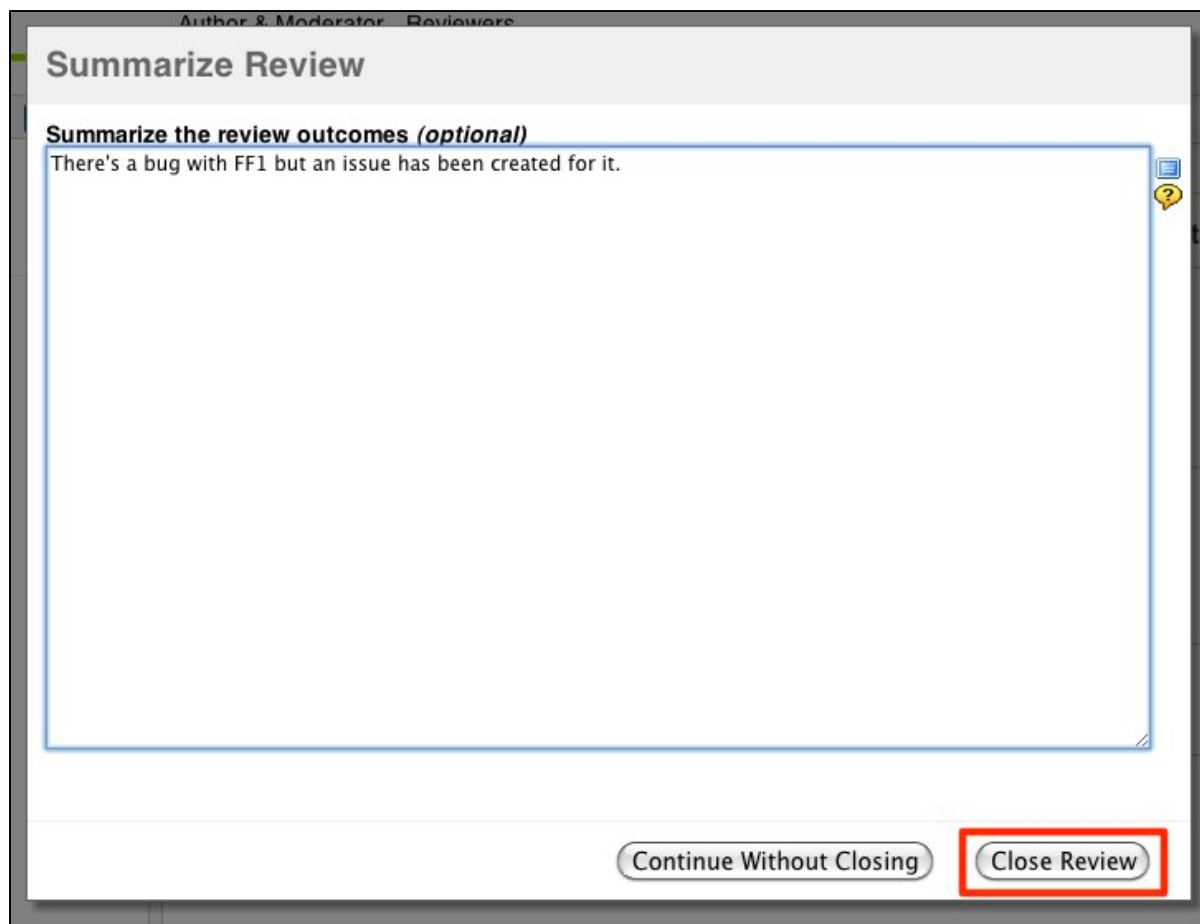
Objectives [Edit](#)

Please check if the CSS modifications are IE compliant

General Comments

There are no general comments on this review. [Add a general comment.](#)

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



Installing and upgrading Git

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

- [Check your version of Git](#)
- [Install or upgrade Git on Linux](#)
- [Install or upgrade Git on Mac OS X](#)
- [Install or upgrade Git on Windows](#)
- [Restart Crucible](#)

Check your version of Git

The versions of Git supported by Crucible are listed on [Supported platforms](#).

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

For example:

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

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

Install or upgrade Git on Linux

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

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

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

Now [check the Git version](#) – you should see the new version of Git.

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

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

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

Install or upgrade Git on Mac OS X

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

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



Now [check the Git version](#) – you should see the new version of Git.

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

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

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

Install or upgrade Git on Windows

Download the [Full installer for official Git for Windows](#). Installing Git for Windows (msysGit) also installs a

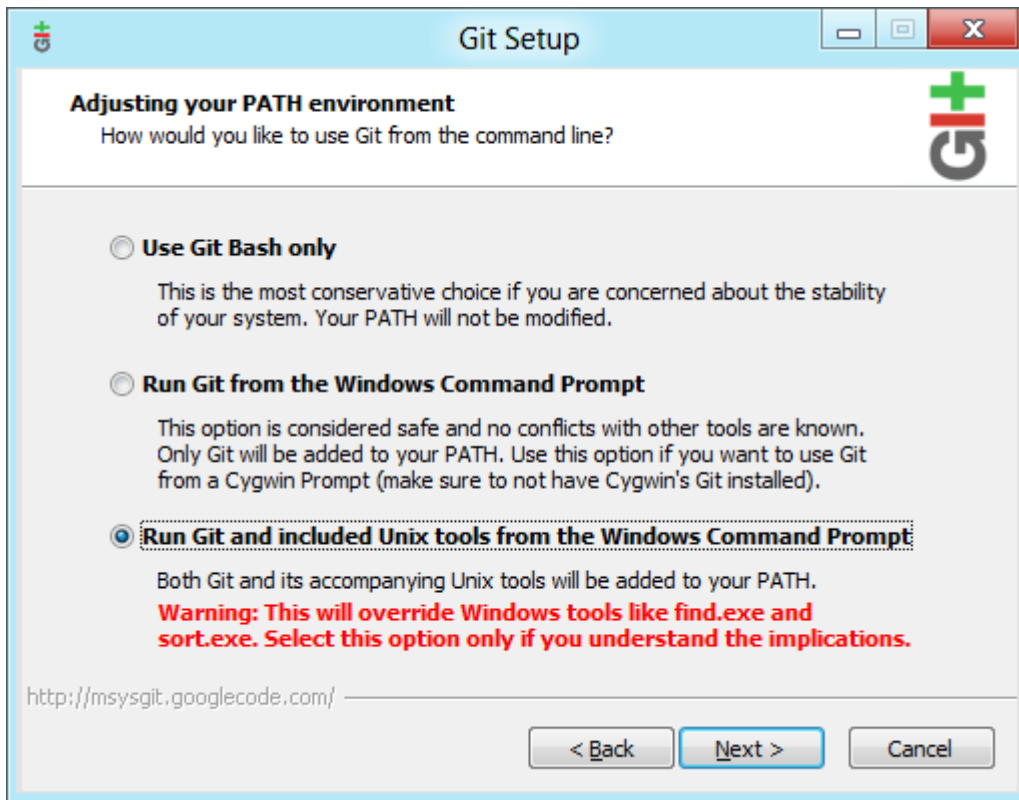
supported version of Perl.

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

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

Ensure that `git.exe` is available in the path:

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



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

Restart Crucible

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

On Windows:

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

```
bin\start.bat
```

On Linux and Mac OS X:

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

```
bin\start.sh
```

Using Crucible

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

This page provides an overview of how to use Crucible.

1. Point Crucible to your repositories

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

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

- [Stash](#)
- [Git](#)
- [Subversion](#)
- [Mercurial](#)
- [CVS](#)
- [Perforce](#)

2. Set up a Crucible project

A Crucible project allows you to

- define default [moderators](#), [authors](#) and [reviewers](#) for the reviews in that project.
- define which people are eligible to be [reviewers](#) for the reviews in that project.
- use [permission schemes](#) to restrict who can perform particular actions (e.g. 'Create Review') in that project.

A Crucible administrator can create new projects – see [Creating a project](#).

3. Review something!

When you create a review you'll want to:

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

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

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

See [Creating a review](#) for more information.

Using the Crucible screens

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

On this page:

- [Dashboard](#)
- [Header](#)

- [Recent activity](#)
- [Related pages](#)

Dashboard

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



in the [header](#).

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

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

Header



The table below explains the tabs in the Crucible header:

Tab	Function	Appears
Repositories	Displays contents of connected source repositories. The dropdown menu has links to recently visited repositories.	Only when FishEye is used with Crucible. Only when FishEye is used with Crucible for logged-in users.

Projects	Displays reviews and content from specific projects. The dropdown menu has links to recently visited projects.	All screens All screens for logged-in users.
People	Displays metrics on the users of the Crucible instance. The dropdown menu has links to recently visited user pages.	All screens. All screens for logged-in users.
Reviews	Allows you to search and report on reviews. The dropdown menu has links to recently visited reviews, as well as links to the Crucible Inbox and Outbox. Choose Reviews > Review dashboard to see the Review Dashboard that has more information about your reviews.	All screens All screens for logged-in users. .

Recent activity

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

Browsing commit activity

Commit activity includes files commits to repositories that you have selected as [favourites](#).

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

Browsing reviews activity

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

Related pages

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

Browsing all reviews

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

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

- [Your reviews](#)
- [Other reviews](#)
- [Custom filter](#)
- [Reports](#)
- [Related topics](#)

Your reviews

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

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

Inbox	
To Review	Reviews where you are a reviewer and haven't yet completed your review work.
Ready to Close	Reviews where you are a moderator and haven't yet summarised and closed the review.
In Draft	Reviews that you have created but have not yet been moved to the 'Approval' state or the 'Require Approval' state.
Require My Approval	Reviews where you are a moderator and need to approve the review.
Outbox	
Out for Review	Reviews that you are a participant of, that have review work that is yet to be completed by other reviewers.
Completed	Reviews that you are a participant of, and have been completed .
Archive	
Closed	Reviews that you are a participant of, that have been summarised and closed .
Abandoned	Reviews that you are a participant of, that have been abandoned. You may wish to delete these reviews.
My Open Snippets	All open snippets created by you.
My Snippets	All snippets created by you.

Screenshot: Browsing your reviews

Review Dashboard

My Reviews

Inbox (5)

To Review (1)

Ready to Close (0)

In Draft (4)

Require My Approval (0)

Outbox (0)

Out For Review (0)

Completed (0)

Archive (10)

Closed (5)

Abandoned (5)

My Snippets

My Open Snippets (6)

My Snippets (6)

Inbox

5 Reviews

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

Other reviews

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

All Open Reviews	Reviews that have not been summarised and closed yet.
All Closed Reviews	Reviews that have been summarised and closed .
All Reviews	All reviews, including open reviews, closed reviews and draft reviews.

All Open Snippets	All open snippets .
All Snippets	All snippets , i.e. open and closed snippets.

Screenshot: Browsing all open reviews

The screenshot shows the 'Review Dashboard' in Crucible. On the left, there's a sidebar with 'My Reviews' (Inbox, To Review, Ready to Close, In Draft, Require My Approval, Outbox, Out For Review, Completed, Archive, Closed, Abandoned) and 'My Snippets' (My Open Snippets, My Snippets). The main area is titled 'All Open Reviews' and displays a table of reviews. The table has columns: State, Review, Owner, Name, Files, Comments, Age, Due, and Reviewers. It shows 11 results, with the first few being 'Review' state and 'CR-FE-7365' through 'CR-CLOV-245'. The 'Name' column contains details about the review, such as 'FECRU-3474: Initialise repo markers if out of sync with states'.

Custom filter

You can filter reviews by author or by projects that you have selected as [favourites](#).

Reports

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

Related topics

[Viewing reports](#)

Browsing source files

When FishEye is installed with Crucible, the **Repositories** tab is available in the [header](#).

To browse source files:

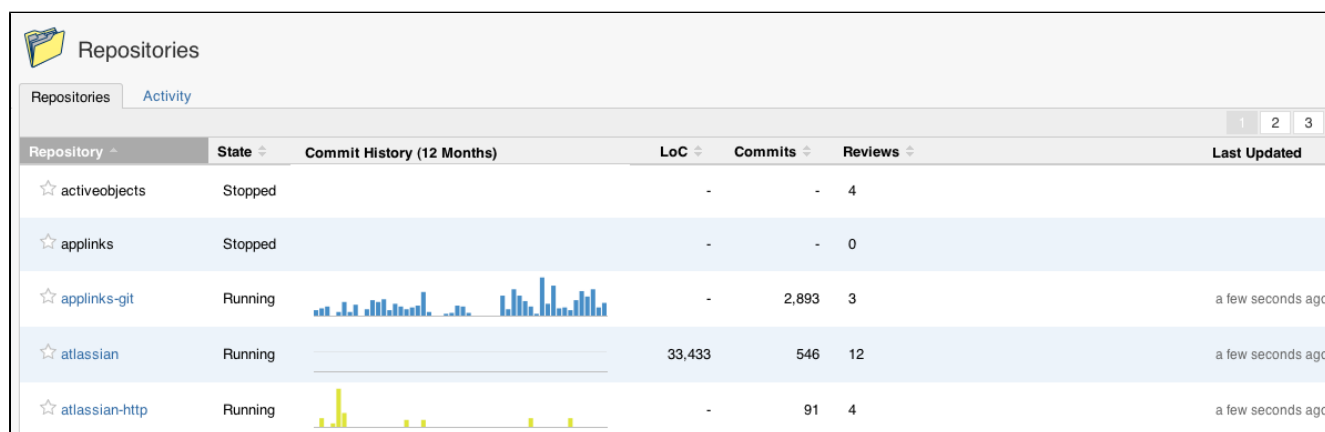
1. Choose **Repositories** > **All repositories** from the header. The 'Repositories' view will be displayed, showing summary information if you have multiple repositories set up. See the '[Viewing all repositories](#)' screenshot below.
2. Click the name for a repository to view its contents. See the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

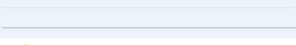
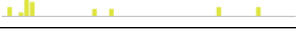
Tab	Description
-----	-------------

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

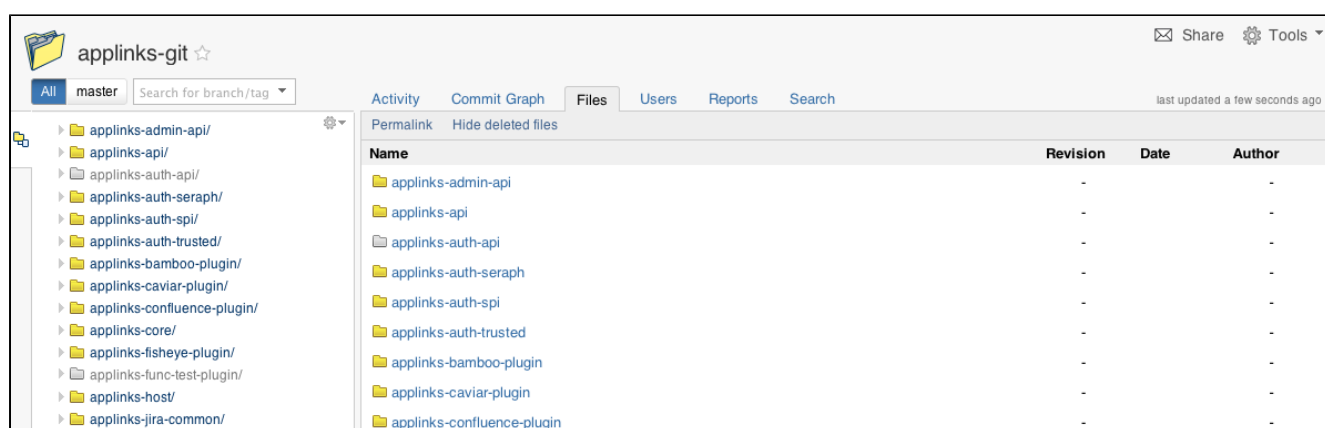
 To download files, click the **Source** tab for the desired file, then right-click **Raw**.

Screenshot: Viewing all repositories



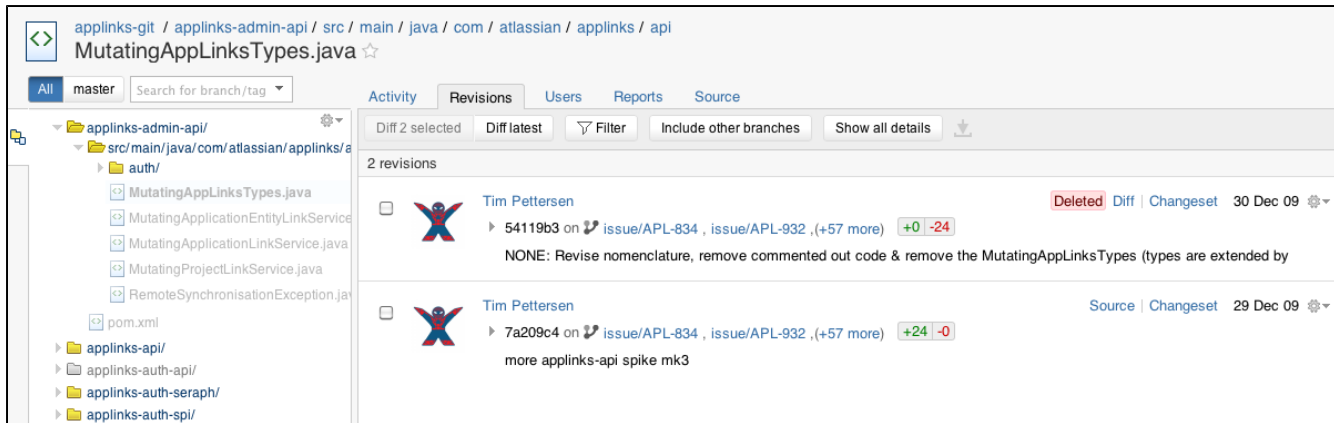
Repository	State	Commit History (12 Months)	LoC	Commits	Reviews	Last Updated
activeobjects	Stopped		-	-	4	
applinks	Stopped		-	-	0	
applinks-git	Running		-	2,893	3	a few seconds ago
atlassian	Running		33,433	546	12	a few seconds ago
atlassian-http	Running		-	91	4	a few seconds ago

Screenshot: Viewing a repository



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

Screenshot: Viewing a file



Browsing projects

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

The page for the project has the following sections:

Left navigation panel

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

Activity tab

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

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

Reviews tab

Lists all the reviews for the project.

 The Projects tab is only visible in Crucible. Read more about the [definition of a project](#).

Screenshot: The Crucible Project View

The screenshot shows the FishEye web interface. On the left, there's a sidebar with the FishEye logo, a list of repositories (CR-FE, FE-hg, FE, CGIT), a line history graph showing activity from 2004 to 2010, and statistics: Total Commits: 40,257, Last Commit: 01:04, Commits this week: 293, Total Lines of Code (LoC): 693,936. The main area is titled 'Activity' and 'Reviews'. It shows a list of reviews with details like author (Michael Heemskerk, Ping Lyons), status (closed, started review, started progress), and timestamps. A specific review by Michael Heemskerk is expanded, showing commit details and file changes.

Changing your User Profile

See [Changing your User Profile](#) in the FishEye documentation.

Using favourites

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

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

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

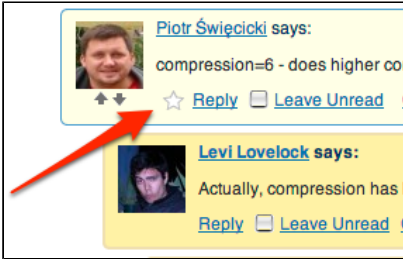
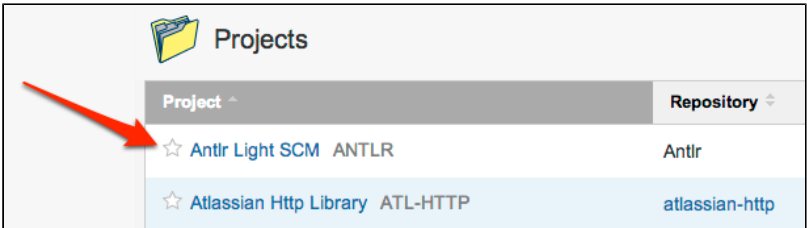
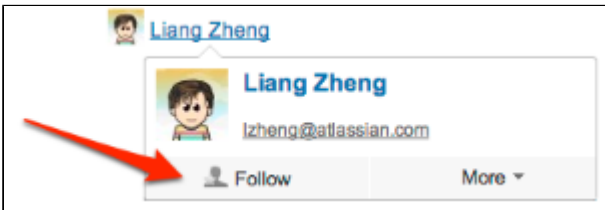
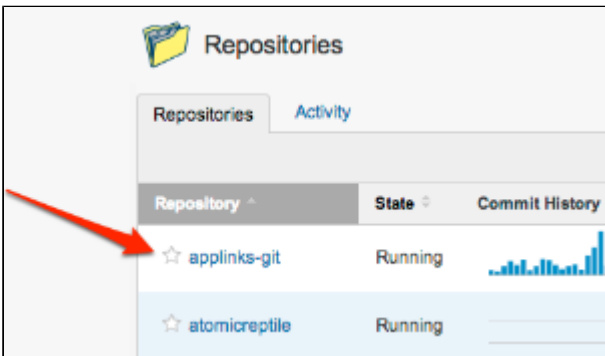
On this page:

- [Adding favourites](#)
- [Managing favourites](#)

Adding favourites

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

Item	Favourite it by...	Looks like this...
Review	Hover over the review name, for example in the 'Open Reviews' list of the Reviews Dashboard. In the popup, click the cog icon and choose Add to Favourites.	The screenshot shows a review titled 'CR-JST-15' with a cog icon. A dropdown menu is open, showing options: 'Upgrade task', 'run more L...', 'Add to Favourites' (highlighted with a red arrow), 'Download Review as Text', and 'View Review History'.

Review comment thread	Click the grey star in the first comment of the comment thread.	
Project	Click the grey star beside the project's name in the 'All Projects' list. See Browsing projects .	
Person	Hover over a person's username and click Follow .	
Repository	(Requires FishEye) In the 'Repositories' list, click the grey star beside the repository's name. See Browsing source files .	

Managing favourites

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

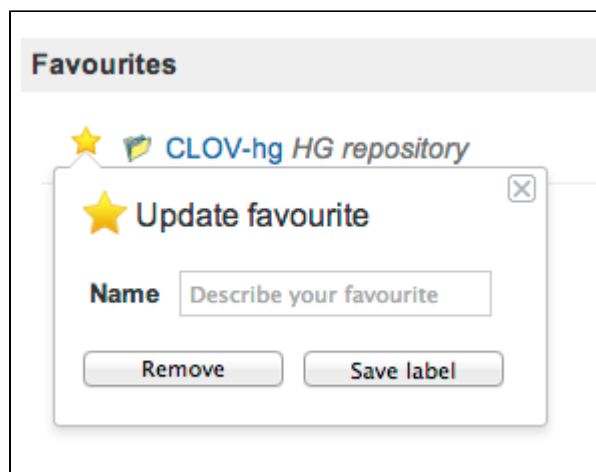
Profile settings

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

Favourites

- ★ 📁 CLOV-hg HG repository

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



Using Wiki Markup in Crucible

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

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

 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. <i>Normal heading</i>
h5.Small heading	Turns text into a heading at size 5. <i>Small heading</i>
h6.Smallest heading	Turns text into a heading at size 6. <i>Smallest heading</i>

Text Effects

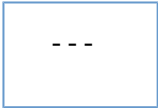
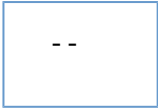
Text effects are used to change the formatting of words and sentences.

Notation	Description
bold	Makes text appear bold .
<i>_italic_</i>	Makes text appear in <i>italics</i> .
<u>+underline+</u>	Makes text appear <u>underlined</u> .
??citation??	Makes text appear in — <i>citation</i> form.
-striketrough-	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 - A numbered list - Second item - indented item 1 - indented item 2

Images

Images can be referenced from remote sources only.

Notation	Description
<code>!http://www.host.com/image.gif!</code>	The image will be displayed from the remote source.
<code>!http://www.host.com/image.gif align=right, vspace=4!</code>	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
<code> heading 1 heading 2 heading 3 col A1 col A2 col A3 col B1 col B2 col B3 </code>	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
<code>{noformat}</code>	Makes a preformatted block of text with no syntax highlighting. All the optional parameters of the <code>{noformat}</code> macro are valid for the <code>{panel}</code> macro as well. Example: This is a no-formatted piece of text, so <code>*no*</code> _formatting_ is done here.

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

Miscellaneous Markup Features

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

Notation	Description
<code>\X</code>	Escape special character X (i.e. <code>{}</code>)
<code>:)</code> , <code>:(</code>	Graphical emoticons (smileys): 😊, ☹️ .

Notation	:)	:(	:P	:D	;	(y)	(n)	(i)	(/)	(x)	(!)
Image	😊	☹️	😜	😄	😏	👍	👎	ℹ️	✅	❌	⚠️

Notation	(+)	(-)	(?)	(on)	(off)	(*)	(*r)	(*g)	(*b)	(*y)
Image										

Using RSS feeds in Crucible

Subscribing to an RSS feed

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

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

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

Using keyboard shortcuts in Crucible

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

General shortcuts

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

Custom navigation

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

Comment navigation

Key	Function
n	Go to next review comment
p	Go to previous review comment
shift + p	Go to first review comment
shift + n	Go to last review comment
l	Go to next thread (skips replies)
h	Go to previous thread (skips replies)
]	Go to next unread comment
[	Go to previous unread comment

r	Reply to a comment
m	Toggle comment read/unread status

File navigation

Key	Function
j	Go to the next element (file, comment, defect or diff hunk depending on your current context)
k	Go to the previous element (file, comment, defect or diff hunk depending on your current context)
shift + k	Go to first file
shift + j	Go to last file
u	Go to next unreviewed file
i	Go to previous unreviewed file
y	Set file reviewed and go to next unreviewed file
shift + y	Toggle file reviewed/unreviewed status
e	Expand current file
c	Collapse current file
shift + e	Expand all files
shift + c	Collapse all files

Crucible icons

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

Icon	Description
	View review-level comments

 A pixelated icon of a speech bubble with a cross inside, representing a comment.	Go to the previous comment
 A pixelated icon of a speech bubble with a cross inside, representing a comment.	Go to the next comment



[Add a comment](#)



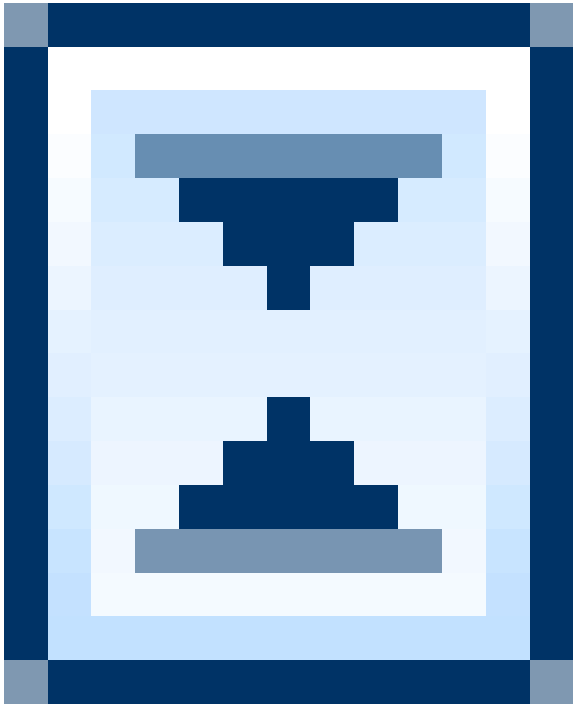
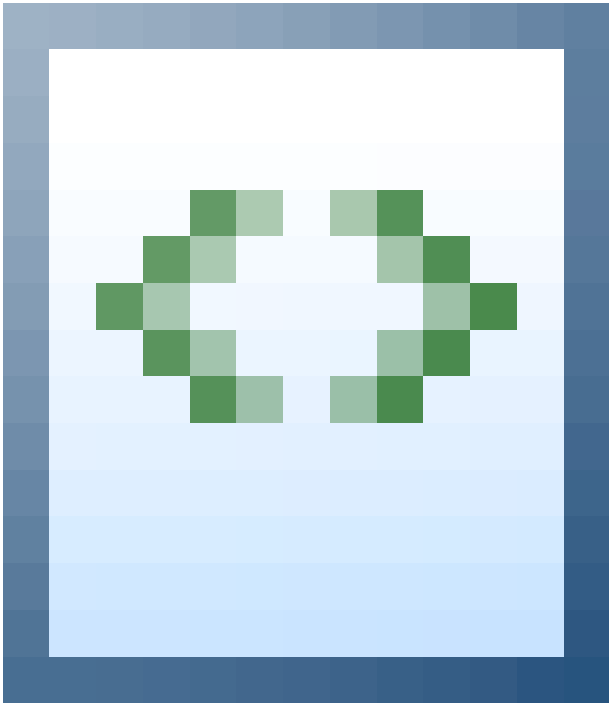
[Go to the previous file in this review](#)

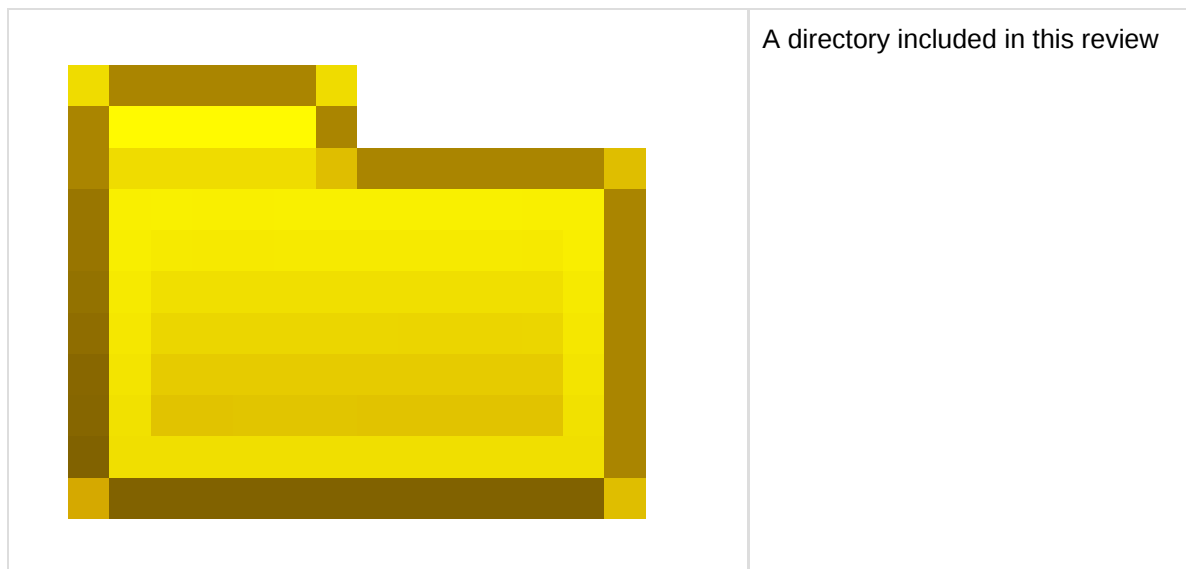
Go to the next file in this review



Expand all files



	Collapse all files
	A file included in this review



The Crucible workflow

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

On this page:

Crucible Workflow

Example workflow: Two participant code review

1. The author starts the review
2. The reviewer comments on the code
3. The author responds to the comments
4. The author closes the review

Roles

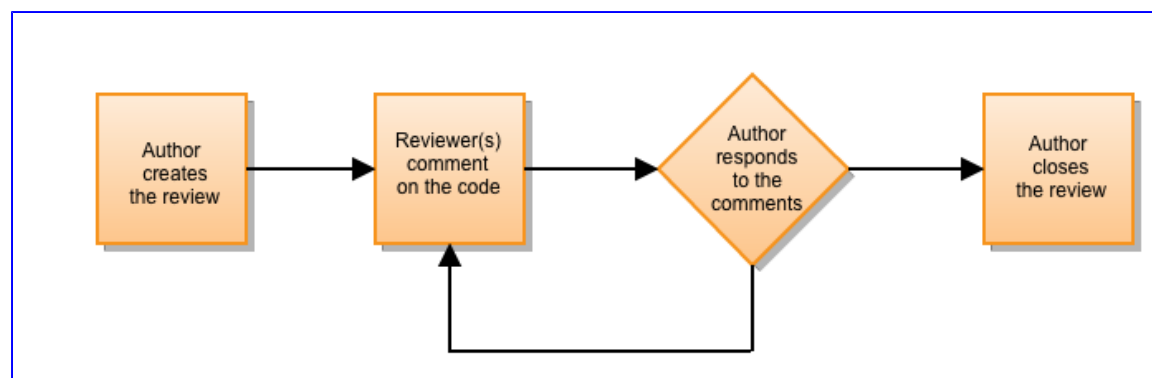
Crucible is a flexible application that caters for a wide range of team sizes and work styles. You will need to know about the basic roles used in Crucible:

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

Crucible Workflow

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

Diagram: Workflow for One-to-One Reviews



Need more information? Read more about the different forms of [workflow](#) in Crucible.

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

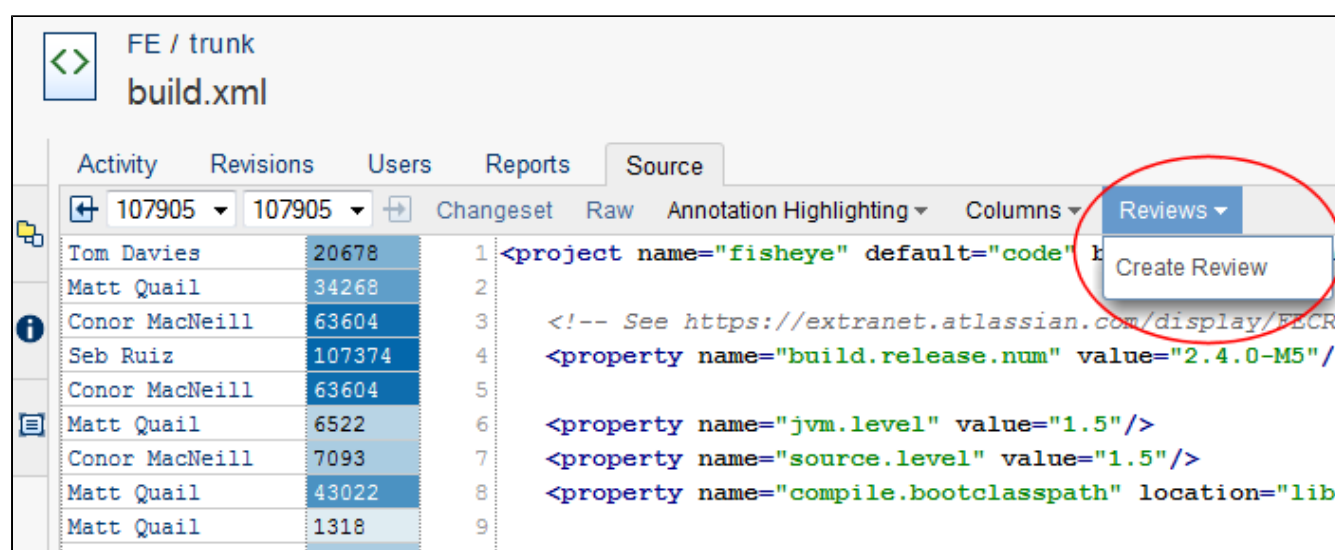
Example workflow: Two participant code review

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

1. The author starts the review

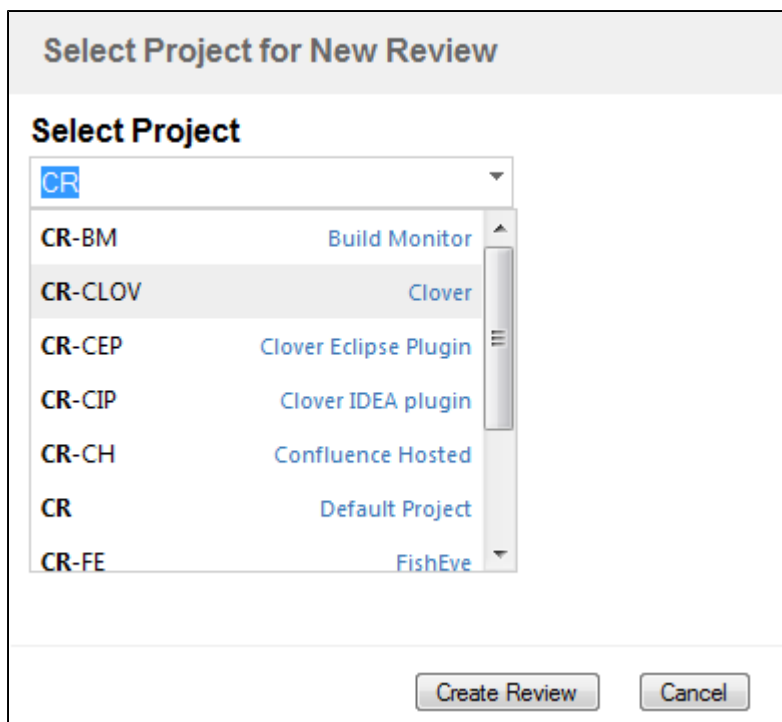
To begin, the code author sets up the review. There are a [number of ways to do this](#), but for this example, the author starts from the FishEye source view of the file he wants to review:

Screenshot: Opening a review from the FishEye Source view



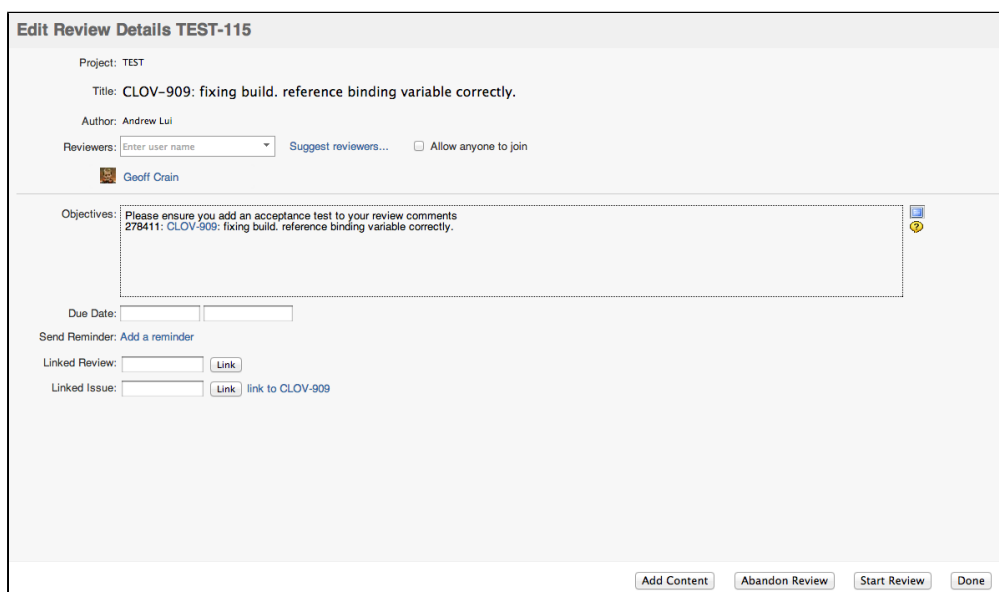
From the FishEye Source view, the author chooses **Reviews > Create Review**. If there are multiple projects, the Select Project dialogue opens.

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

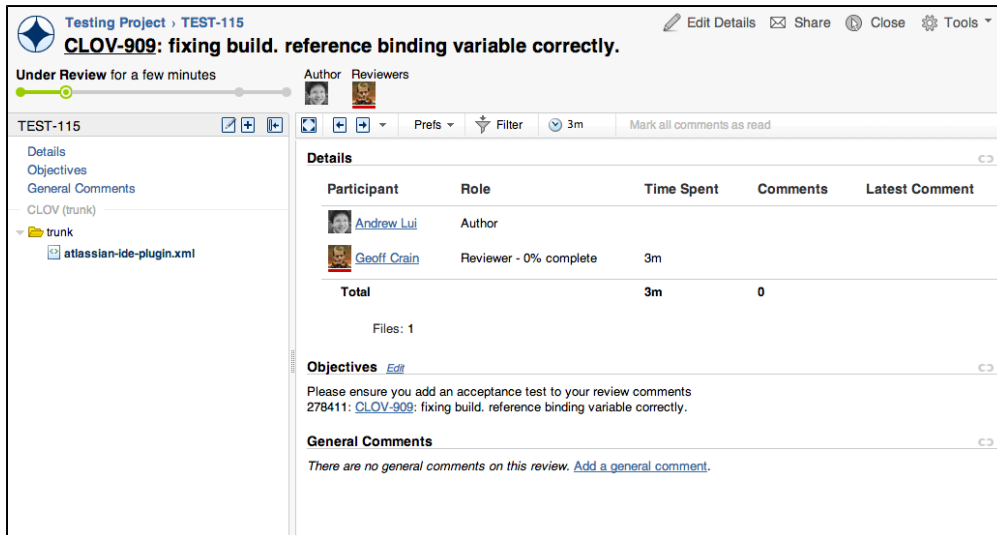


In the Edit Review dialogue, the author enters information needed for the review. This includes entering a title and description for the review, selecting reviewers, a due date and the key for a related JIRA issue (if any). The project and author are pre-selected.

The author can also add more content to the review, if they wish, by clicking **Add Content**. See [Adding content to the review](#).

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

Screenshot: A new Crucible review



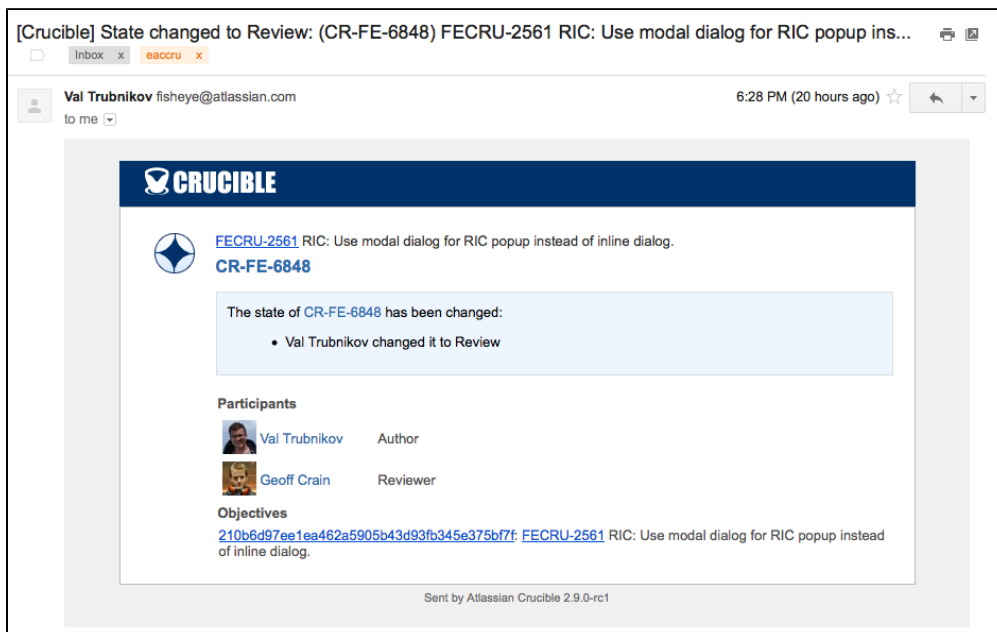
The draft review opens. In the draft stage, the author can check the contents of the review files to ensure they are correct and put in any notes for reviewers as comments. During the draft phase, no notification emails are sent out to reviewers. Once the author is finished with the draft phase, he clicks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

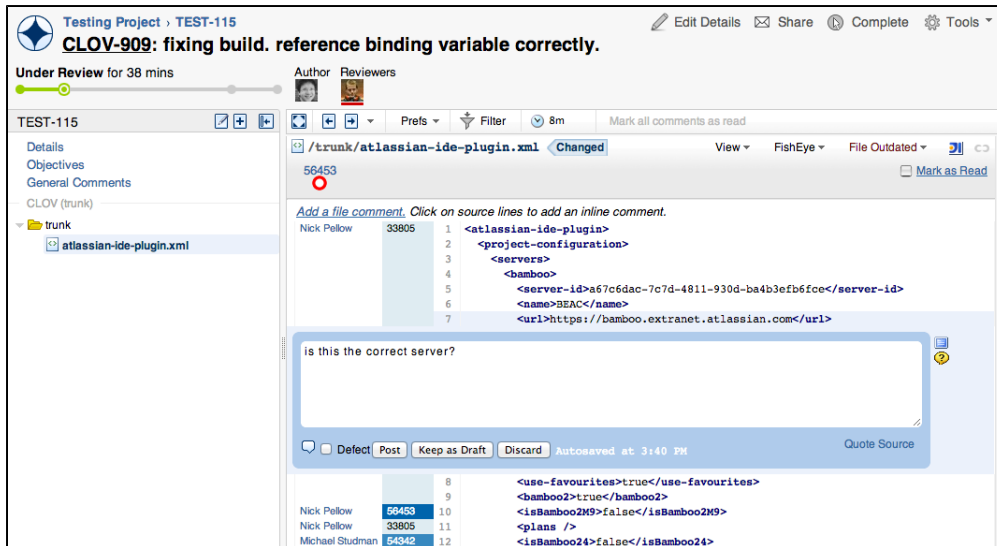
The reviewer will receive an email from Crucible with a link that they can follow to the review.

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



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

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

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

The reviewer repeats this process for all files in the review. Reviewers can leave the session and resume it later; their work is automatically saved.

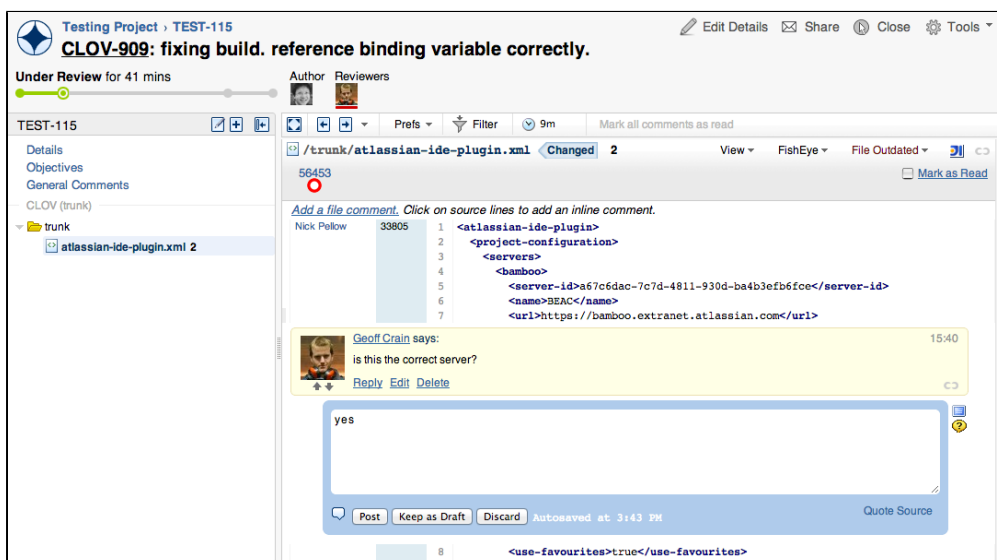
When the reviewer has finished their code review work, they click **Complete**.

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

3. The author responds to the comments

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

Screenshot: Comment threads in Crucible



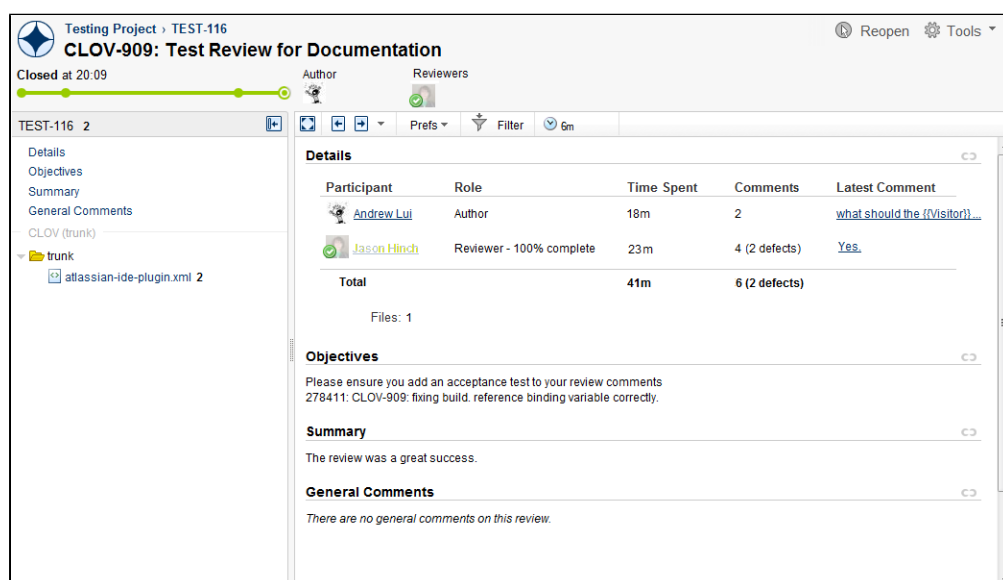
4. The author closes the review

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 can then add any final comments, and 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. This returns the status of the review to "Open", without changing the status of existing reviewers. Click **Edit Details** to add reviewers or to change other details of the review.

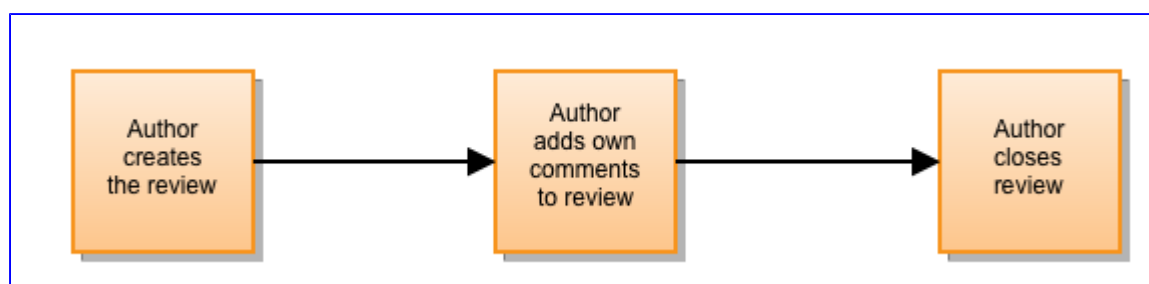
Defining your workflow

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-many reviews without a moderator \(Agile team\)](#)
- [Formal group reviews \(CMM team\)](#)

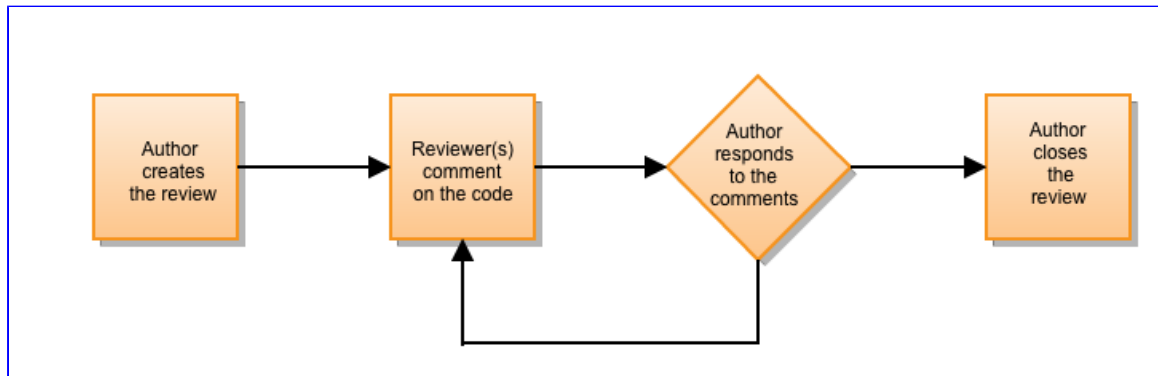
Lightweight code commenting with Crucible (individual)

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.

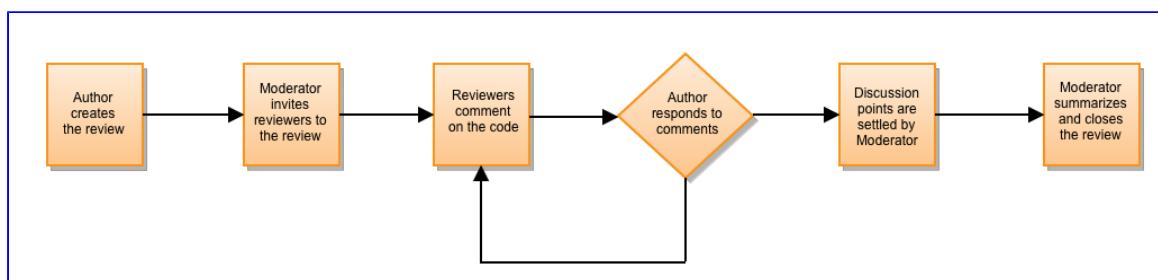


One-to-many reviews without a moderator (Agile team)

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.

**Formal group reviews (CMM team)**

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



To see a simple example of how to use Crucible with two people, see [The Crucible workflow](#).

Roles and status classifications

This page explains the roles and status classifications in Crucible.

- **Roles**
 - [Author](#)
 - [Creator/Moderator](#)
 - [Reviewer](#)
 - [User](#)
- **Status classifications**
 - [Draft](#)
 - [Under Review](#)
 - [Summarized](#)
 - [Closed](#)
 - [Abandoned](#)

Roles

Author

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

Creator/Moderator

The *creator* is the person who [creates the review](#). In most cases this person will also act as [moderator](#). The *moderator* is the person responsible for [creating](#) the review, [approving](#) the review, determining when reviewing is finished, [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

Draft

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

Under Review

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

Summarized

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

Closed

Closed reviews are complete.

Abandoned

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

See also the [Glossary of terms](#) used in Crucible.

Creating a review

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

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

See the [considerations](#) for when creating reviews at the end of this page.

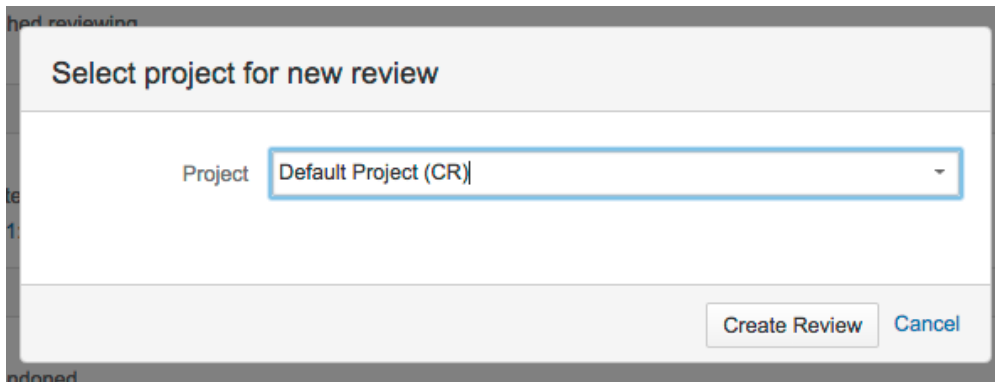
Note that only people with the 'Create' [permission](#) can create a review.

You can also create reviews by:

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

Create a review

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

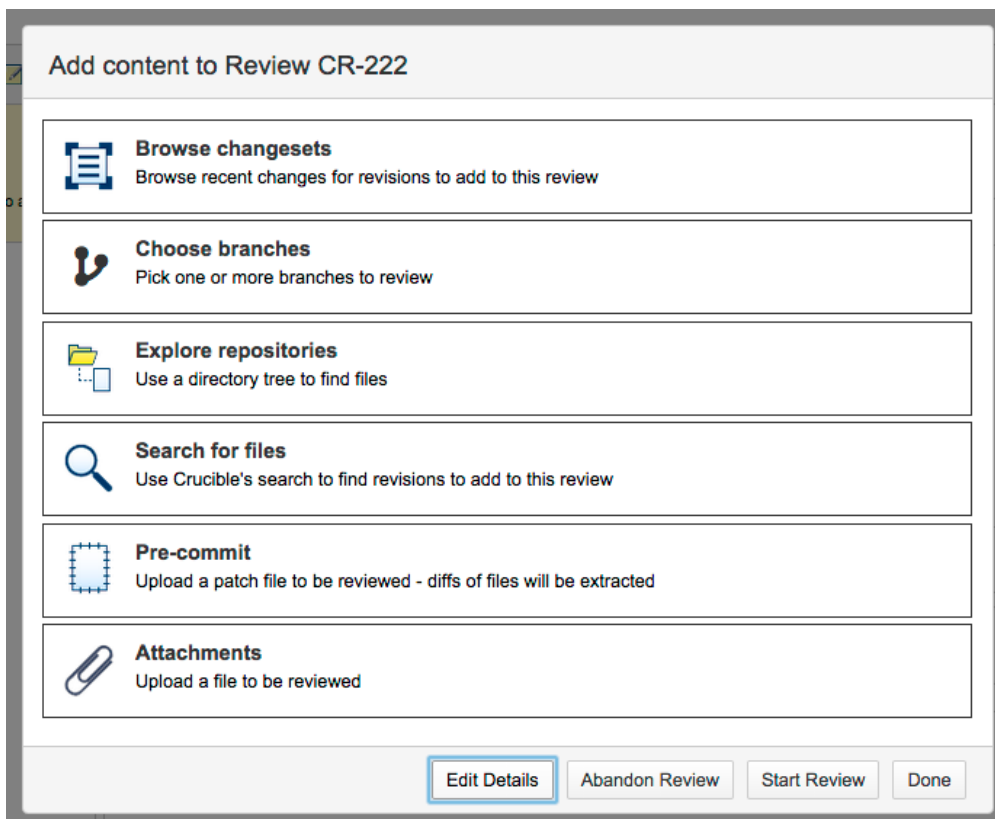


The screenshot shows a dialog box titled "Select project for new review". It contains a "Project" label and a dropdown menu currently showing "Default Project (CR)". At the bottom right, there are two buttons: "Create Review" and "Cancel".

Click **Create Review**.

Add content to the review

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



The screenshot shows a dialog box titled "Add content to Review CR-222". It contains six options, each with an icon and a description:

- Browse changesets**: Browse recent changes for revisions to add to this review (Icon: Document with lines)
- Choose branches**: Pick one or more branches to review (Icon: Branching diagram)
- Explore repositories**: Use a directory tree to find files (Icon: Folder)
- Search for files**: Use Crucible's search to find revisions to add to this review (Icon: Magnifying glass)
- Pre-commit**: Upload a patch file to be reviewed - diffs of files will be extracted (Icon: Patch file)
- Attachments**: Upload a file to be reviewed (Icon: Paperclip)

At the bottom, there are four buttons: "Edit Details" (highlighted with a blue border), "Abandon Review", "Start Review", and "Done".

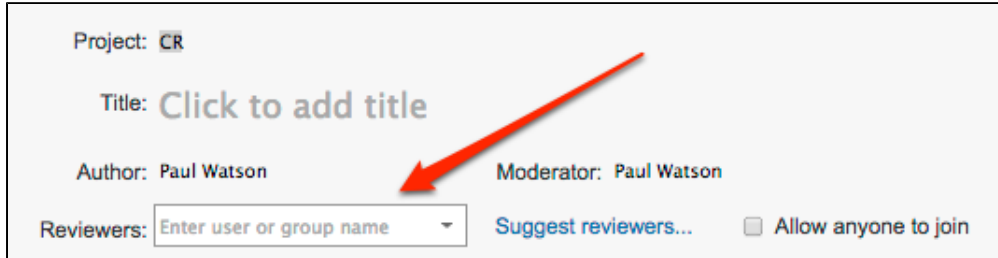
Reviews should be created with care to get the best value from them. We've found it's best not to overload the review – they should be focussed on just a few necessary files.

See [Adding content to the review](#) for more details.

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

Choose the reviewers

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



Project: CR

Title: Click to add title

Author: Paul Watson Moderator: Paul Watson

Reviewers: [Suggest reviewers...](#) ☐ Allow anyone to join

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

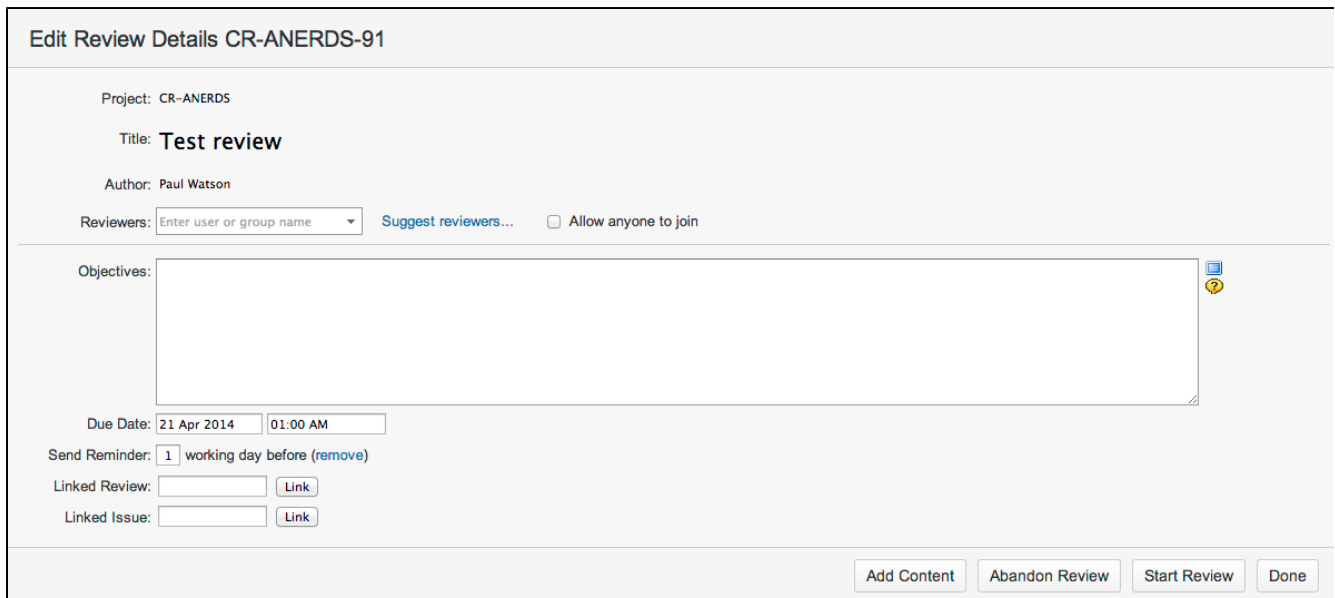
See [Choosing reviewers](#) and [Enabling the moderator role](#) for more details.

Complete other details for the review

You can set other details for the review, including:

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

The project, moderator and author are pre-selected.



Edit Review Details CR-ANERDS-91

Project: CR-ANERDS

Title: **Test review**

Author: Paul Watson

Reviewers: [Suggest reviewers...](#) ☐ Allow anyone to join

Objectives:

Due Date:

Send Reminder: working day before [\(remove\)](#)

Linked Review: [Link](#)

Linked Issue: [Link](#)

[Add Content](#) [Abandon Review](#) [Start Review](#) [Done](#)

Once you're finished, click **Done**.

Start the review

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

Click **Start Review** to make the review available. See [Performing the review](#).

When all the reviewers have performed their reviews, you can [summarise and close](#) the review.

The screenshot shows the 'Angry Nerds Reviews' interface for a review titled 'CR-ANERDS-91'. At the top, there are buttons for 'Edit Details', 'Start Review', 'Share', and 'Tools'. Below this, a status bar indicates the review is in 'DRAFT' mode, 'for a few seconds', and 'due in 6 days'. The author is listed as 'Author' with a profile picture. The review title is 'Test review'.

On the left sidebar, there is a section for 'CR-ANERDS-91' with a toolbar and a list of files. The first file is '.hgtags' with a red 'x' icon. Below the file list, there is a section for 'angrynerds (ANERDS-7-tagging)'.

The main content area is divided into several sections:

- Details:** A table showing the review details.

Participant	Role	Time Spent	Comments	Latest Comment
Paul Watson	Author	9m		
Total		9m	0	
- Objectives:** A section with a link to 'Edit' and a message: 'No objectives entered. Objectives let your reviewers know what the goals of the review are and guide their feedback. [Add some objectives](#).'
- Commits:** A section with a link to 'Edit' and a table showing commit details.

Commit	Author	Date	Comment
There are no general comments on this review. Add a general comment .			
- General Comments:** A section with a link to 'Edit' and a message: 'There are no general comments on this review. [Add a general comment](#)'.

Considerations when creating reviews

Review effectiveness

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

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

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

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

Crucible performance

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

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

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

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

Adding an entire directory's contents to a Crucible review

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

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

Creating a review from JIRA

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

? JIRA must be integrated with *both* FishEye and Crucible before you can do this.

See [Linking Crucible to JIRA](#) for information on how to set up an application link with JIRA.

See also [Creating JIRA issues from the review](#).

When using Crucible with JIRA 6.2.x and later

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

To create a review from a JIRA issue:

1. Go to the JIRA issue that relates to the work to be reviewed.
2. Click **Commits** in the Development panel.
3. Click the **FishEye / Crucible** tab (this exists if, for example, both FishEye/Crucible *and* Stash are linked to JIRA).
4. Start the process for creating a review for either a single commit, or for all the commits related to the JIRA issue.
5. In Crucible, the new review is in edit mode:
 - The content of the changeset becomes the content (i.e. files) to be reviewed.
 - The author of the commit becomes the [author](#) of the review, if Crucible is aware of this user. Otherwise the [creator](#) of the review becomes the author.
 - The [creator](#) of the review becomes the [moderator](#).
 - The commit log message is used as both the Title and [Statement of Objective](#).
 - Project selection popup is loaded by default with the last project used to create a review.
6. Choose **Tools > Start Review**, in Crucible, when you are ready.

When using Crucible with JIRA 6.1.x and earlier

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

▼ [Click here if you're using JIRA 6.1 or earlier...](#)

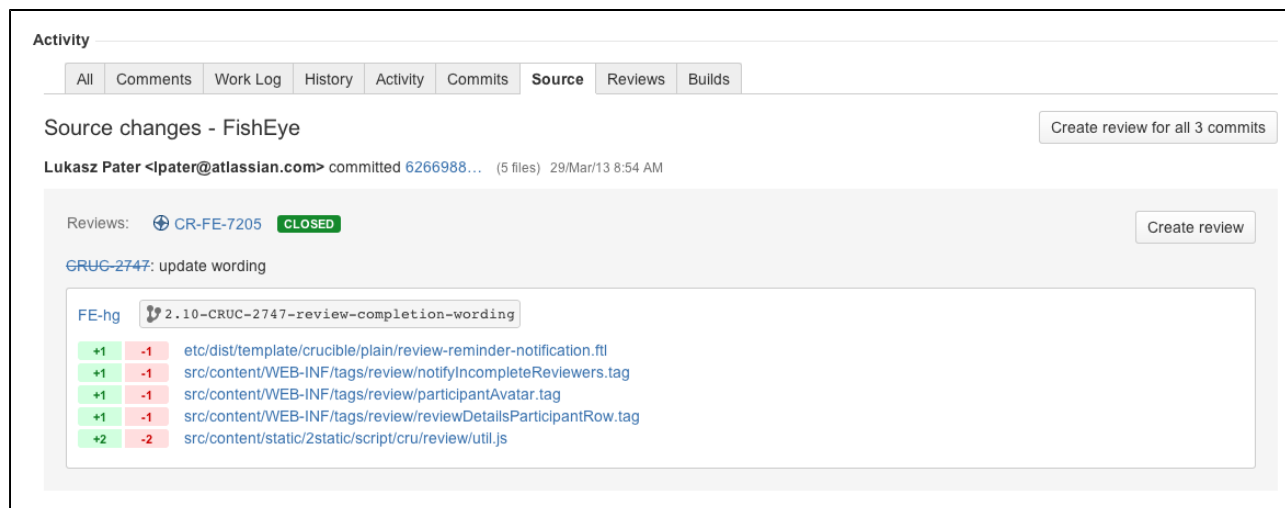
When Crucible is linked with JIRA 6.1.x, or earlier, Crucible content appears on the **Reviews** tab (and FishEye content appears on the **Source** tab) in JIRA.

To create a review from a JIRA issue:

1. Go to the issue in JIRA that relates to the work to be reviewed.
2. Under 'Activity', click the **Source** tab.
3. Either:
 - a. Click **Create review** to create a new review for a particular changeset.
 - b. Click **Create review for all commits** to include all changesets from the JIRA issue in the new review.
4. If a similar review already exists, you can add the changesets to that.
5. In Crucible, the new review is in edit mode:
 - The content of the changeset becomes the content (i.e. files) to be reviewed.
 - The author of the changeset becomes the [author](#) of the review, if Crucible is aware of this user. Otherwise the [creator](#) of the review becomes the author.
 - The [creator](#) of the review becomes the [moderator](#).
 - The commit log message is used as both the Title and [Statement of Objective](#).
6. Choose **Tools > Start Review**, in Crucible, when you are ready.

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

Screenshot: Adding a review from within JIRA



Creating a review from a URL

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

The format of your URL is as follows:

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

The parameters are as follows:

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

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

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

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

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

Creating a Snippet Review

This page explains how to create a simple code review using the Crucible Snippet Review feature. Snippet

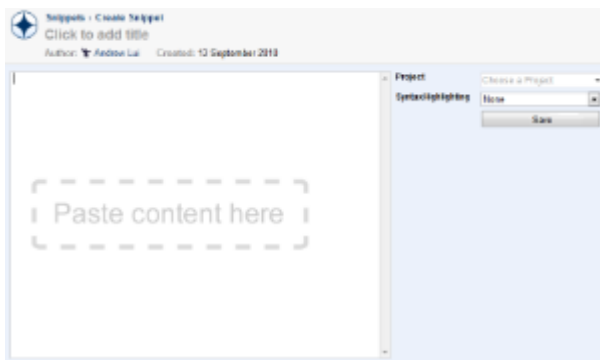
Reviews are designed to be lightweight ad-hoc code reviews.

To create a snippet review:

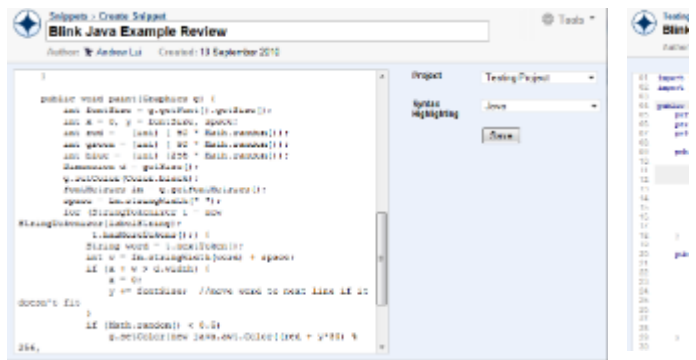
1. Copy the code to be reviewed from the source to your system clipboard.
2. Click **Create snippet** from the **Create review** menu in the Crucible toolbar.
3. Enter details for the snippet review:
 - Paste the code into the panel, where indicated.
 - Click on **Click to add title** near the top to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

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

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



Step 1



Step 2

Creating reviews from the command line

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

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

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

- Subversion
- Perforce
- Git
- Mercurial
- CVS

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

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

Known limitations

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

ission.

On this page:

- [Installing the Review CLI tool](#)
- [Python](#)
- [Using the Review CLI tool](#)

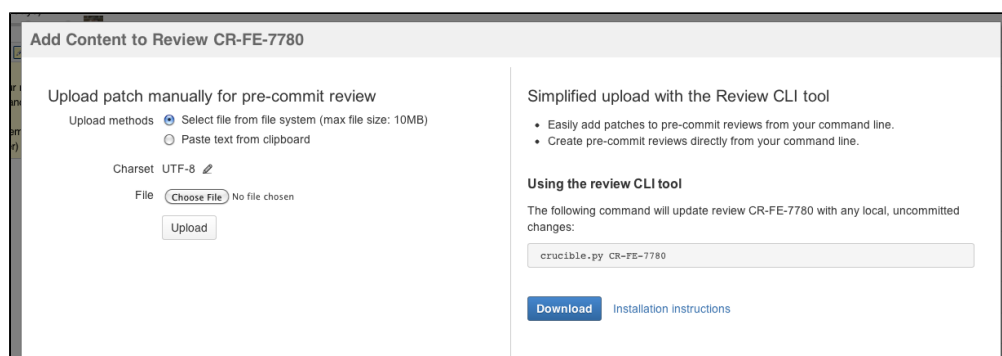
Related pages:

- [Creating a review](#)
- [Adding content to the review](#)

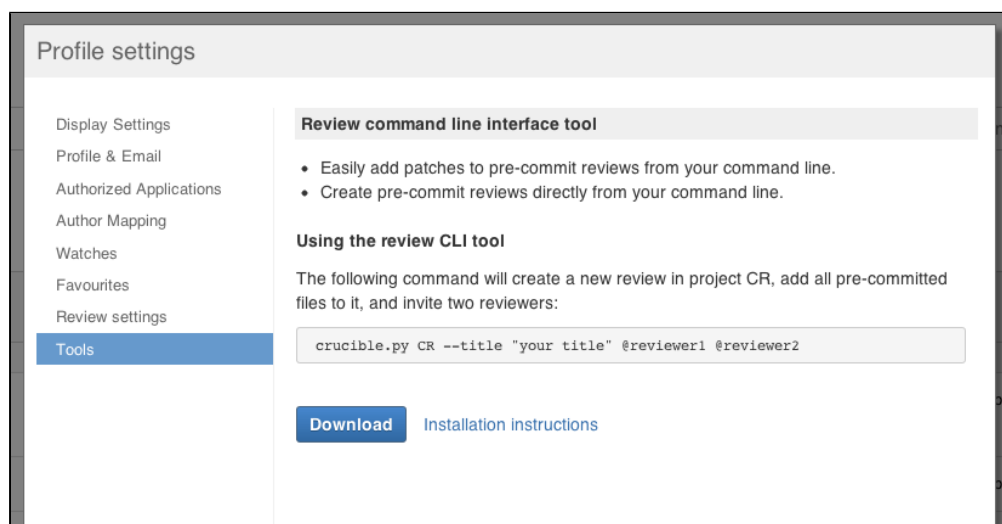
Installing the Review CLI tool

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

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



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



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

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

Python

Python 2.7 is required.

Linux

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

Windows

You can install Python by running the latest [Python 2.7 Installer](#) for Windows.

MacOS X

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

Using the Review CLI tool

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

Call the script with the following command:

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

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

Action	Syntax	Result
View usage help	<code>crucible.py --help</code>	Displays the help with descriptions of all the valid commandline arguments
Create a pre-commit review interactively	<code>crucible.py</code>	Gets the patch from the current SCM, prompts for the project and review title, and creates a draft review
Create a pre-commit review with a given project and title	<code>crucible.py -m "the review title" CR-FE</code>	Gets the patch from the current SCM, creates a draft review in the CR-FE project with the given title
Create a pre-commit review with moderator and reviewers	<code>crucible.py CR-FE @matt @joe --moderator ted</code>	Gets the patch from the current SCM, creates a review in the CR-FE project, adds matt and joe as the reviewers and ted as a moderator, starts the review
Create a pre-commit review anchored to a specific repository	<code>crucible.py -r repo1</code>	Creates a pre-commit review interactively, trying to anchor the patch to the given repository
Add a patch to a review	<code>crucible.py CR-FE-1204</code>	Gets the patch from the current SCM and adds it to the review CR-FE-1204
Create a review from diff output	<code>hg diff crucible.py</code>	Creates a pre-commit review interactively, taking the output of the first command as a patch
Create a review from a diff file	<code>crucible.py -f file.diff</code>	Creates a pre-commit review interactively, taking the output of the patch from the given file

Adding content to the review

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

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

Related pages:

- [Creating a review](#)
- [Iterative reviews](#)
- [Choosing reviewers](#)
- [Performing the review](#)

On this page:

- [Overview](#)
- [Changesets](#)
- [Branches](#)
- [Repository files](#)
- [Search for files](#)
- [Patch files for a pre-commit review](#)
- [Attachments](#)
- [Choose how reviewers see the content](#)

Overview

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

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

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

To add content to a review:

1. Log in to FishEye/Crucible and either:
 - Create a new review, as described on [Creating a review](#), or
 - Open an existing review, for which you are the [creator](#) or [moderator](#), and click the 'Add content' () button.
2. In the 'Add content to Review' dialog, click the option for the type of content you wish to add:

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

Add content to Review CR-222

- Browse changesets**
Browse recent changes for revisions to add to this review
- Choose branches**
Pick one or more branches to review
- Explore repositories**
Use a directory tree to find files
- Search for files**
Use Crucible's search to find revisions to add to this review
- Pre-commit**
Upload a patch file to be reviewed - diffs of files will be extracted
- Attachments**
Upload a file to be reviewed

[Edit Details](#) [Abandon Review](#) [Start Review](#) [Done](#)

Changesets

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

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

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

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

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

Add Content to Review TEST-323

Repository: atlassian Author: any
Branch: any Commit message: any Apply
Add to Review as: Diffs Remove all revisions from review

Scroll to changeset

Sunday 20 Mar 2011

Matthew Watson	40930 added 1 file trunk FOUR-24	9:22 pm
	Moved to atlassian account on bitbucket	
	40929 added 1 file trunk FOUR-24	9:21 pm
	Moved to atlassian account on bitbucket	
	40928 added 1 file trunk	9:18 pm
	Moved to atlassian account on bitbucket	
	40927 added 1 file trunk	9:18 pm
	Moved to atlassian account on bitbucket	
	40926 added 1 file trunk	7:56 pm
	Moved to atlassian account on bitbucket	
	40925 changed 1 file trunk	7:51 pm
	Moved to atlassian account on bitbucket	

Add More Content Edit Details Abandon Review Start Review Done

Branches

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

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

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

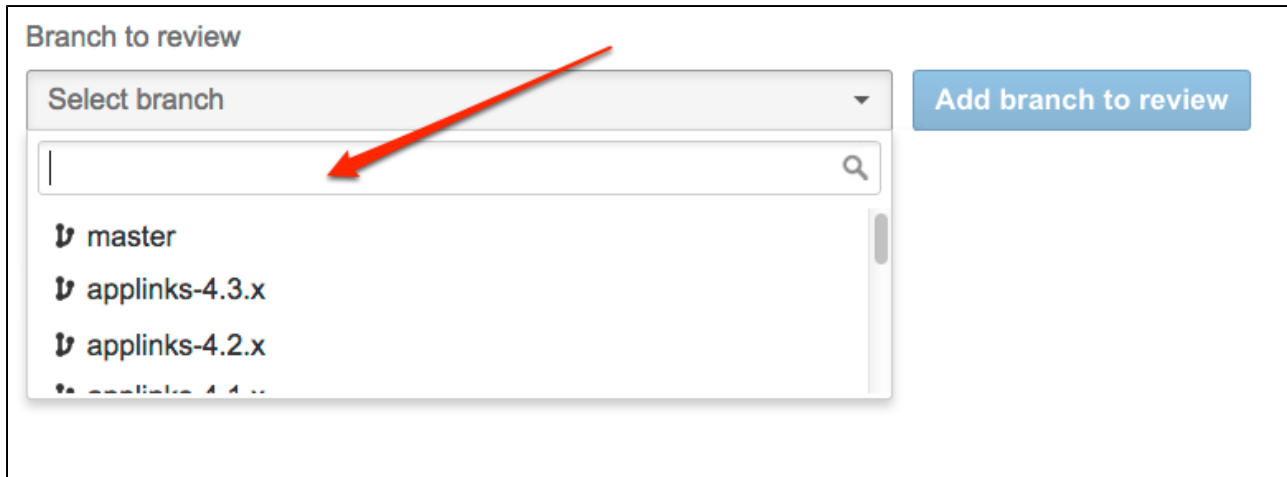
Add branches to the review

Repository Branch to review

Select repository Select branch

amkt
applinks-git
atomicreptile
boneco

Now choose the branch:



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

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

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

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

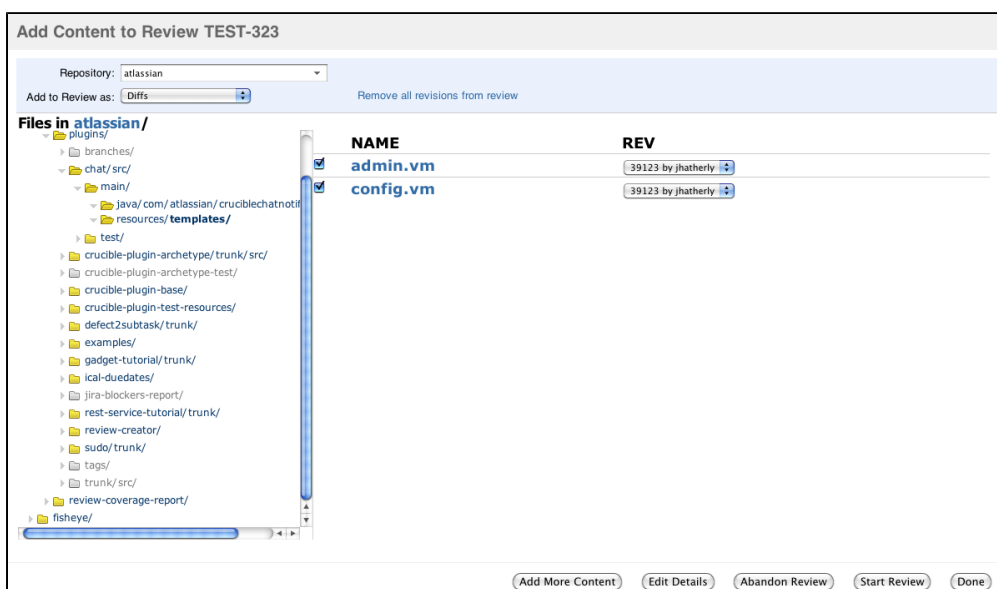
Repository files

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

- By default, the folders are sorted by path name but they can also be sorted by last-commit or first-commit.
- To select a particular revision of a file, select **Load full history...** from the revision number list. This will refresh the available options.
- You can [choose how reviewers will see the files you have added](#), as described below.

Note that:

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



Search for files

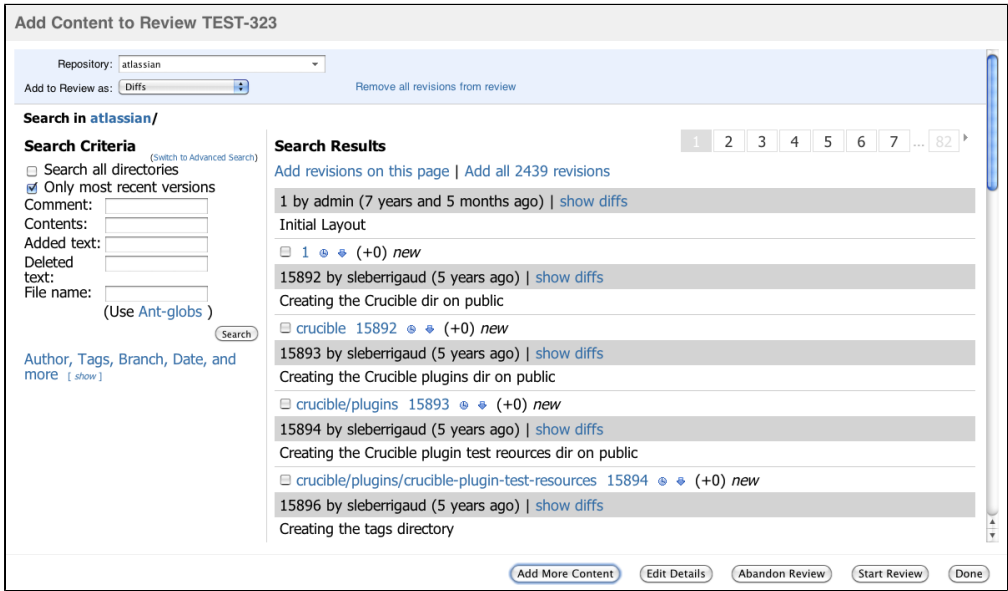
Click **Search for files** on the 'Add content to Review' dialog to search for files to add to your review.

 Search is only available when using **FishEye** with Crucible.

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

Read more about [searching your repositories](#) in the FishEye documentation.

You can [choose how reviewers will see the files you have added](#), as described below.



Add Content to Review TEST-323

Repository: atlassian

Add to Review as: **Diffs** Remove all revisions from review

Search in atlassian/

Search Criteria [\(Switch to Advanced Search\)](#)

☐ Search all directories

☒ Only most recent versions

Comment:

Contents:

Added text:

Deleted text:

File name:

(Use Ant-globs)

[Author, Tags, Branch, Date, and more](#) [\[show \]](#)

Search Results 1 2 3 4 5 6 7 ... 82

[Add revisions on this page](#) | [Add all 2439 revisions](#)

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

Initial Layout

☐ 1 (+0) new

15892 by sleberrigaud (5 years ago) | [show diffs](#)

Creating the Crucible dir on public

☐ crucible 15892 (+0) new

15893 by sleberrigaud (5 years ago) | [show diffs](#)

Creating the Crucible plugins dir on public

☐ crucible/plugins 15893 (+0) new

15894 by sleberrigaud (5 years ago) | [show diffs](#)

Creating the Crucible plugin test resources dir on public

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

15896 by sleberrigaud (5 years ago) | [show diffs](#)

Creating the tags directory

Patch files for a pre-commit review

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

Choose an upload method:

Select file from the file system	Charset – click the edit icon () to choose the character set being used. UTF-8 is the default.
Paste text from clipboard	Patch text — paste your copied text in this text area.

For more information see [Creating patch files for pre-commit reviews](#).

Add Content to Review TEST-400

Upload patch manually for pre-commit review

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

Charset UTF-8 

File Choose File No file chosen

Upload

Patch anchoring

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

See [Using Crucible patch anchoring](#) for more information.

Attachments

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

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

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

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

Choose how reviewers see the content

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

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

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

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

Iterative reviews

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

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

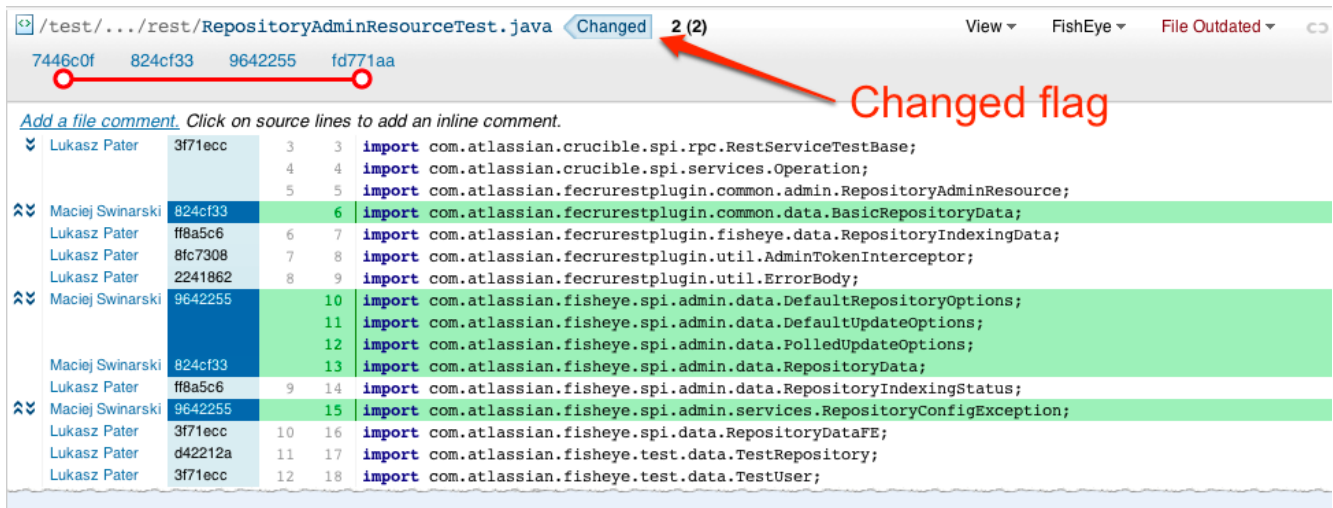
On this page:

- [Iterative post-commit reviews](#)

- Iterative pre-commit reviews
 - Initial patch upload
 - Iterative patch uploads
- Viewing diffs

Iterative post-commit reviews

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



See [Viewing diffs](#) below for information about the slider and diffs.

Iterative pre-commit reviews

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

See [Creating patch files for pre-commit reviews](#).

Initial patch upload

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

You upload the initial patch for a review in the usual way – see [Adding content to the review](#).

Iterative patch uploads

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

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

Add Content to Review CR-10

Upload patch manually for pre-commit review

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

Charset UTF-8 

File No file chosen

Existing patches in CR-10

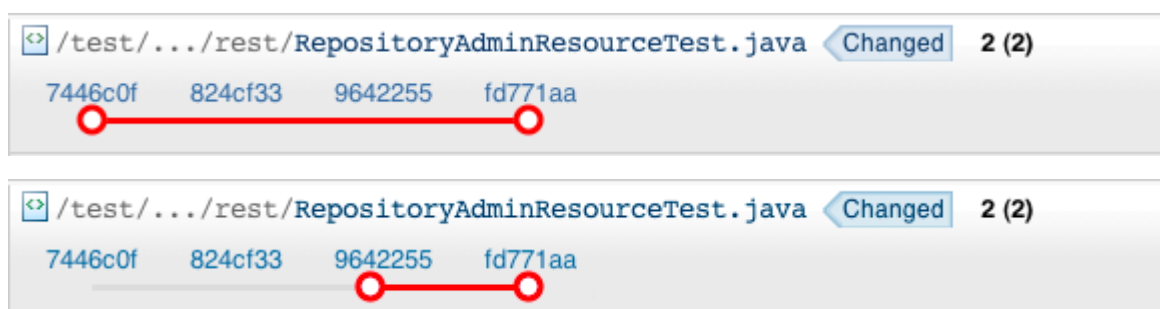
CR-10-patch-0.txt (8 minutes ago) - 2 iterations

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

Viewing diffs

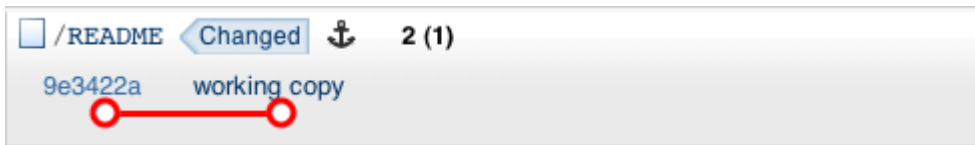
Crucible allows the reviewer to see the different revisions of a file within the same review. The 'slider' in the file view allows you to interactively select which revisions are compared in the displayed diff, and to see the full source content. Comments are connected to, and displayed with, a specific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

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



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

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



Creating patch files for pre-commit reviews

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

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

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

You can create the patch file from your local repository:

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

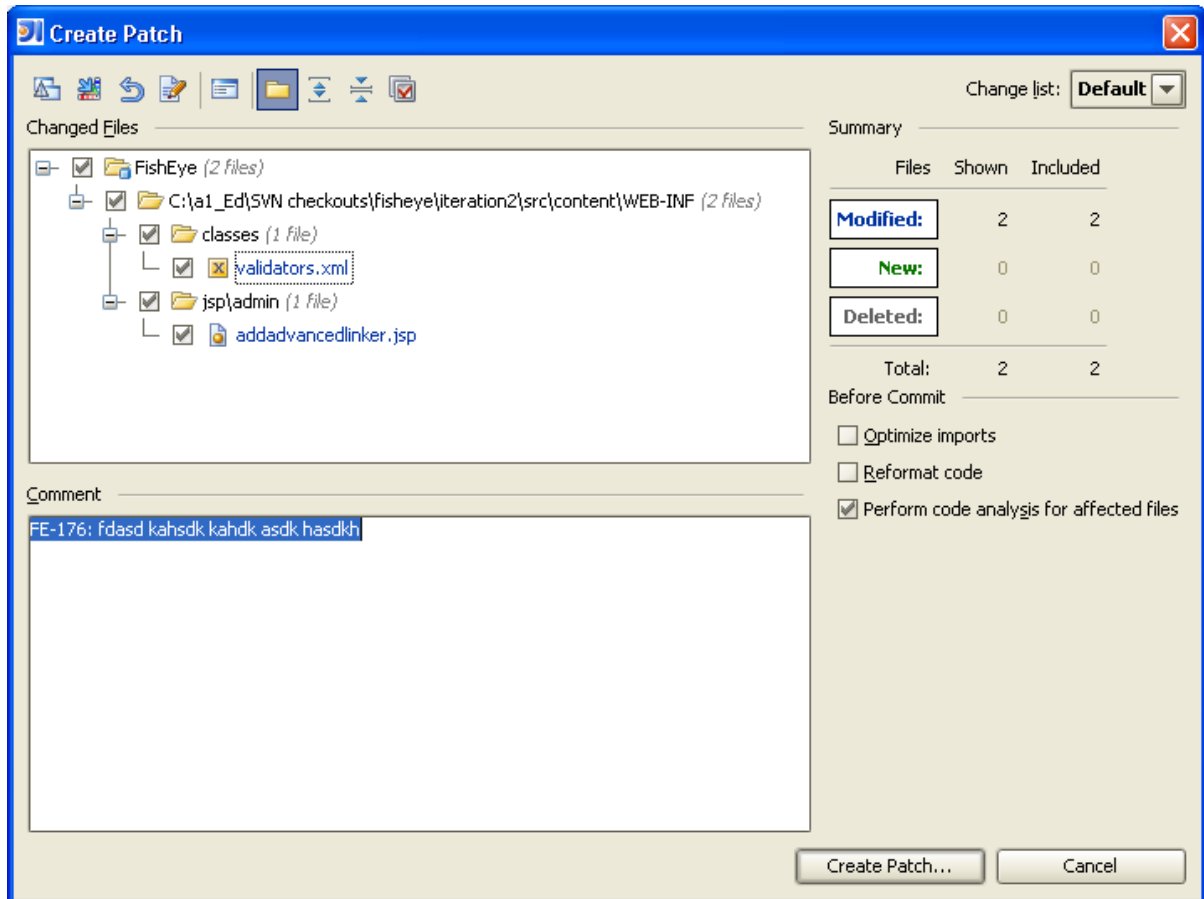
As an enhancement, Crucible's [patch anchoring](#) adds context:

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

On this page:

Creating a patch file from IntelliJ IDEA 7.0

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

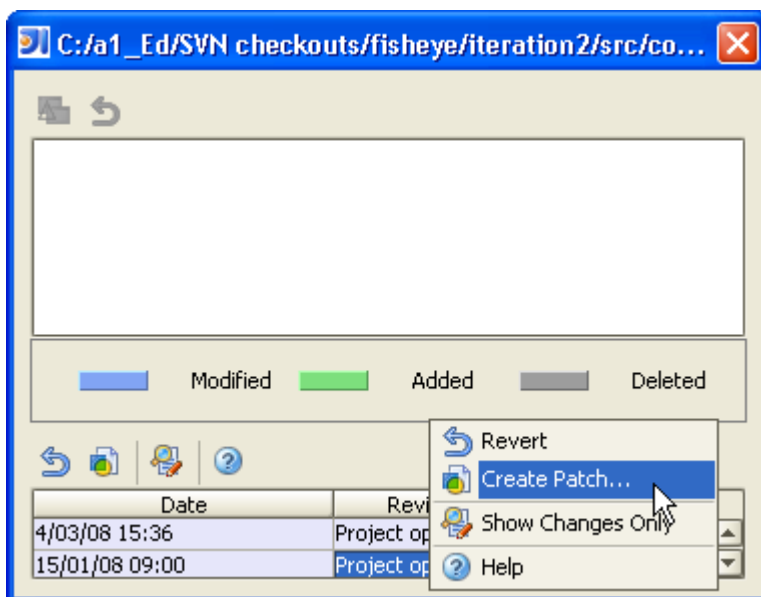


3. Click **Create Patch**, choose a location to save the patch file to, and click **OK**.

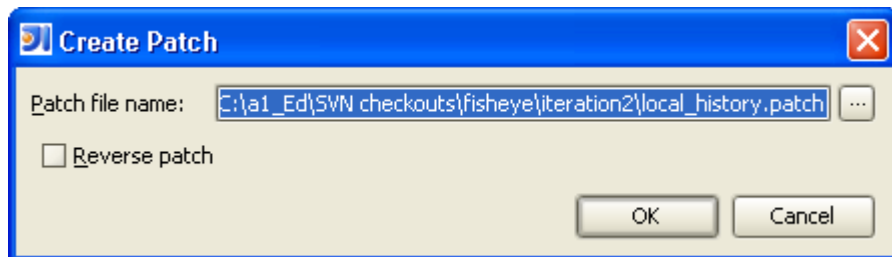
If you do not have the Create Patch command available in IDEA

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

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



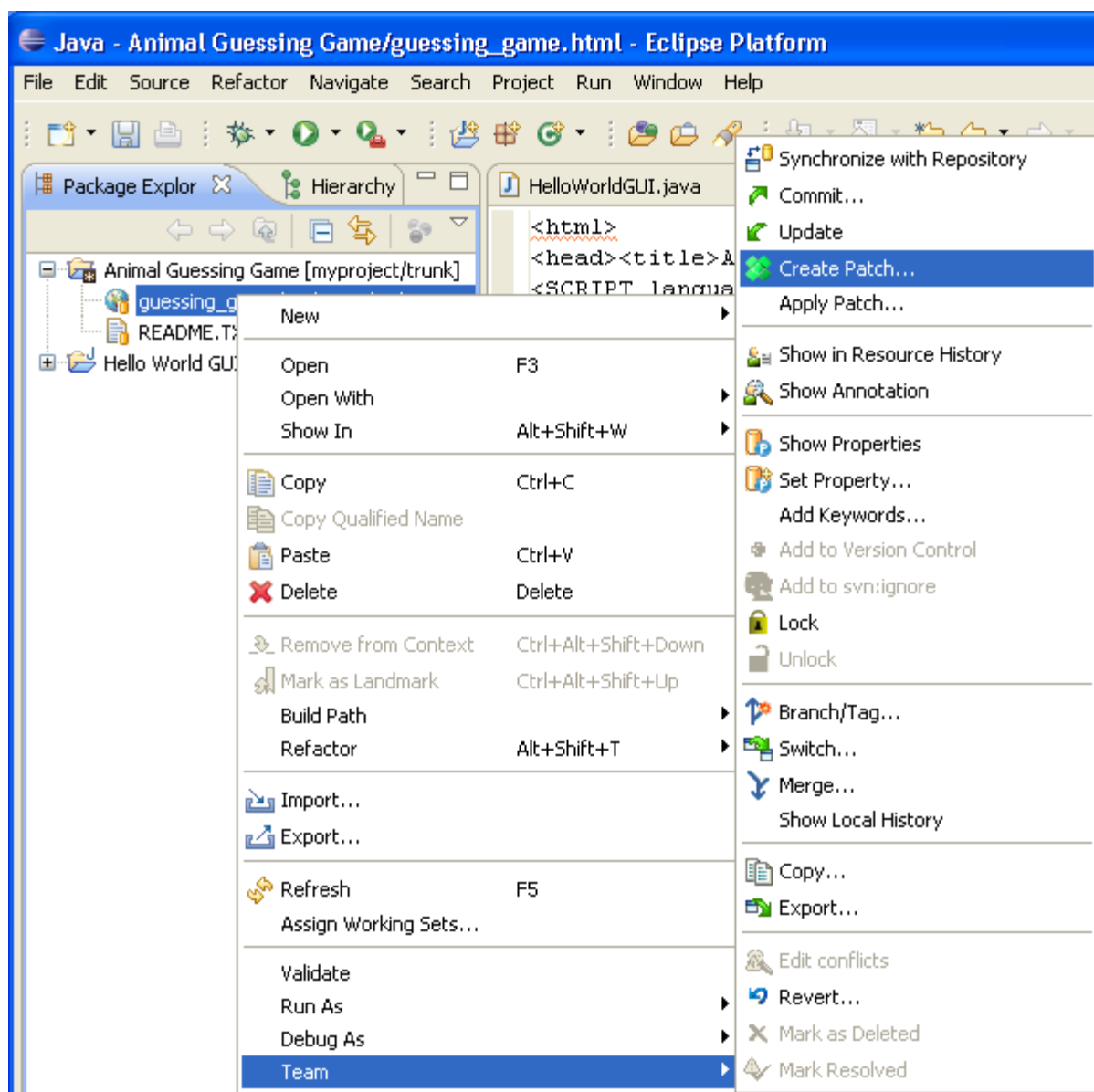
2. In the Local History view, right-click the revision number, and choose **Create Patch**.



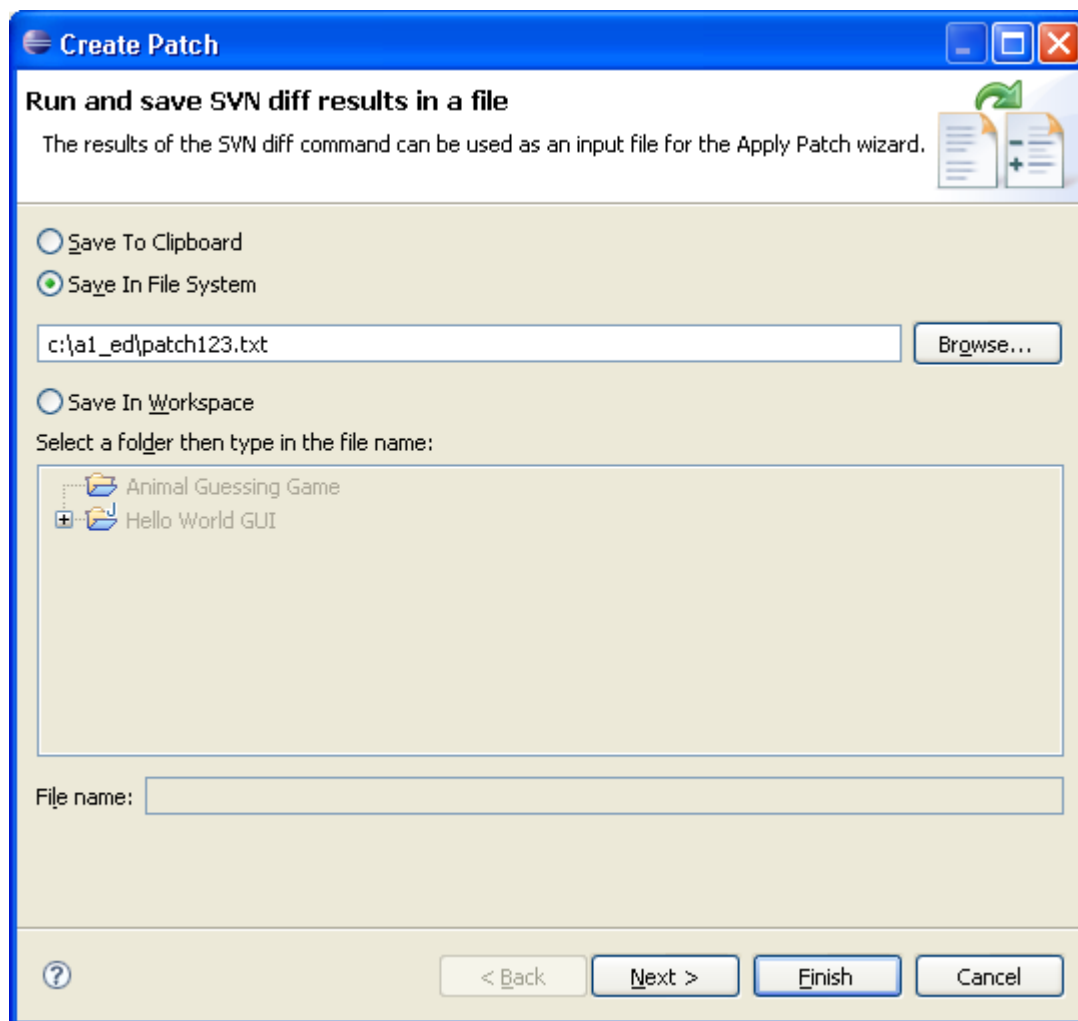
3. In the Create Patch dialog, choose a location for the patch file and a file name, then click **OK**.

Creating a patch file in Eclipse 3.3.1.1

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



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



Creating a patch file from the command line

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

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

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

	<pre>svn diff --diff-cmd diff -x "-U 10000" > patch.txt</pre>	Creates a patch file that shows all code in the file. - The built-in diff feature in <code>svn diff</code> does not support specifying lines of context, so you must tell Subversion to use an external diff command. - The second diff in the command needs to be the name of your external diff command. You might need to specify the full path to that command, such as <code>/usr/bin/diff</code>. - On the Windows platform, you may need a Unix-like emulator such as Cygwin, and install the optional diff command for that.
Perforce	Crucible 3.3.1 or earlier: <pre>p4 diff -dcu > patch.txt</pre> Crucible 3.3.2 or later: <pre>p4 -z tag diff -du > patch.txt</pre>	Creates a patch file with around three lines of code, before and after each change. Revision information is included by default, so if you have FishEye, you may be able to anchor the patch to get full context. The <code>-dcu</code> option provides a combination of "context format" and "unified format". It provides the diff in a standard unified diff format (which we need to parse the diff) as well as revision information (which we need to anchor to FishEye). Later versions of Perforce do not support <code>-dcu</code>, so <code>-z tag</code> should be used instead.
	<pre>diff -u /dev/null path_to_added_file >> patch.txt</pre>	Example of using GNU diff to append files individually to the patch in UNIX. Since Perforce diffs do not include added and deleted files, you should use <code>p4 opened</code> to find such files. Replace <code>path_to_added_file</code> with the actual path of your added file. You can follow a similar procedure with deleted files using <code>p4 print</code> to extract the previous version of the file.
	 Perforce does not directly support creating patches that include all lines of code. A workaround is to check out 'before' and 'after' versions of the file, and use GNU Diff to create a patch between the two files. That file could then be loaded into a Crucible review.	
Mercurial	<pre>hg diff > patch.txt</pre>	Creates a patch file with around three lines of code, before and after each change. Revision information is included by default, so if you have FishEye, you may be able to anchor the patch to get full context. If you use Git-style diffs (<code>--git</code>), the revision information will not be provided. This means that we cannot anchor the patch to a FishEye repository.
Git	<pre>git diff > patch.txt</pre>	Creates a patch file with around three lines of code, before and after each change. Revision information is included by default, so if you have FishEye, you may be able to anchor the patch to get full context.

Using Crucible patch anchoring to automatically add full context

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

To use patch anchoring:

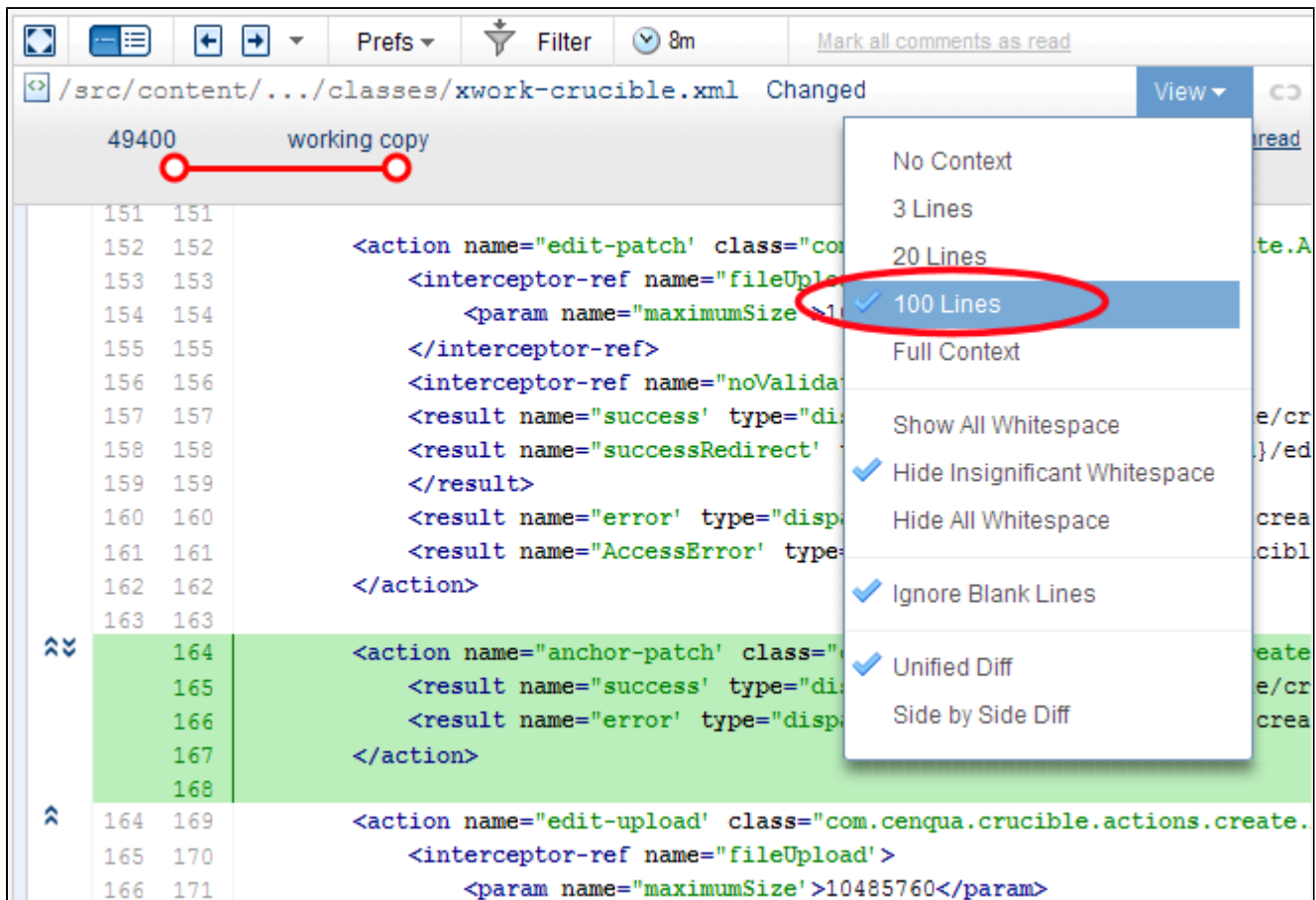
1. Click **Create review** in the Crucible header.

2. Click **Pre-commit**. In the dialog that appears, click **Choose File** to locate your file, then **Upload**. Crucible will now search for matches in the files in its database. Crucible will 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.

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

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

Screenshot: Viewing more than three lines with Patch Anchoring



Choosing reviewers

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

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

Adding users to a review

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

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

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

Edit Review Details CR-ANERDS-92

Project: CR-ANERDS

Title: [Click to add title](#)

Author: Paul Watson

Reviewers: [Suggest reviewers...](#) ☐ Allow anyone to join

Objectives:

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

Due Date:

Send Reminder: working day before (remove)

Linked Review: [Link](#)

Linked Issue: [Link](#)

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

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

Suggested reviewers

Crucible will automatically suggest reviewers, by 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.

Removing reviewers

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

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

Edit Review Details TEST-324

Project: TEST

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

Author: Paul Watson

Reviewers: [Suggest reviewers...](#) ☒ Allow anyone to join

Objectives:

Please ensure you add an acceptance test to your review comments

Checking the draft and starting the review

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

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

Next steps

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

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

Performing the review

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

On this page:

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

Deciding what needs to be reviewed

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

Browse your reviews on the Dashboard

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

Use the Dashboard to manage your reviews – see [Browsing all reviews](#).

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

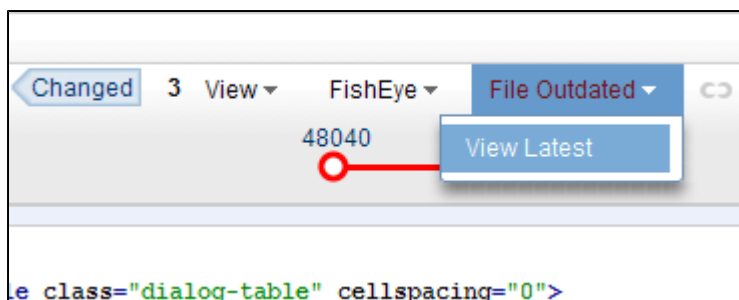
If email notifications are enabled (see [SMTP settings](#) in the FishEye documentation), reviewers will receive an email with information about the review. Click the link within the email to go directly to the review.

When files change during a review

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



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:



Next steps

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

Starting a review

On this page:

- Starting a review
- Editing a review once started

Starting a review

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

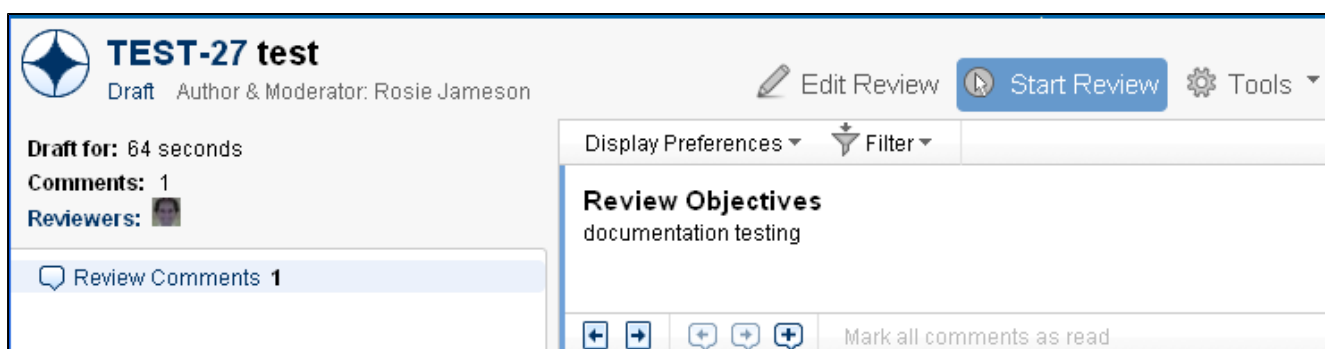
Once you have [selected the reviewers](#), the next stage is to notify the [reviewers](#) and the [author](#) (if different to the [moderator](#)) that they can start reviewing. The review has been in 'Draft' [state](#) until this point. Only the moderator has the permission to start a review.

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



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

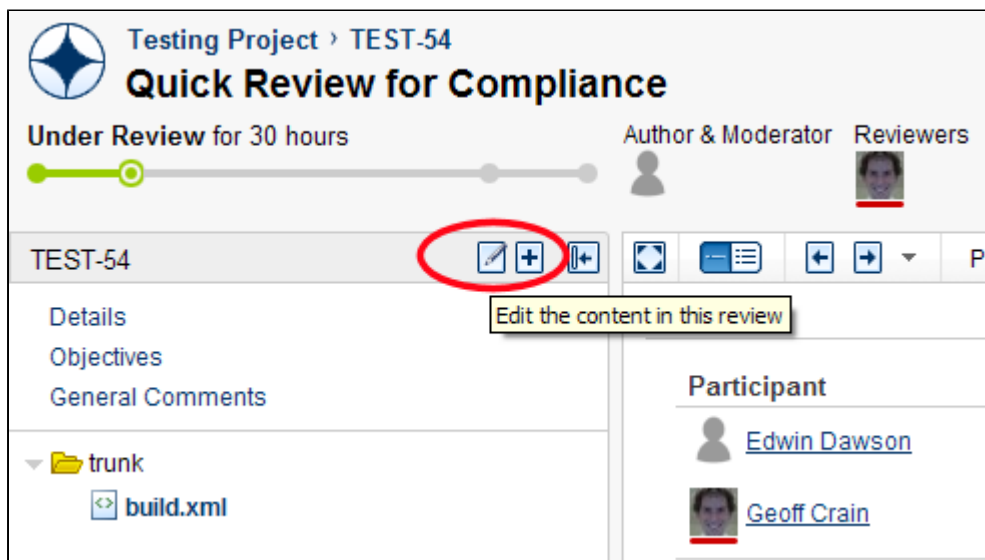
Editing a review once started

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

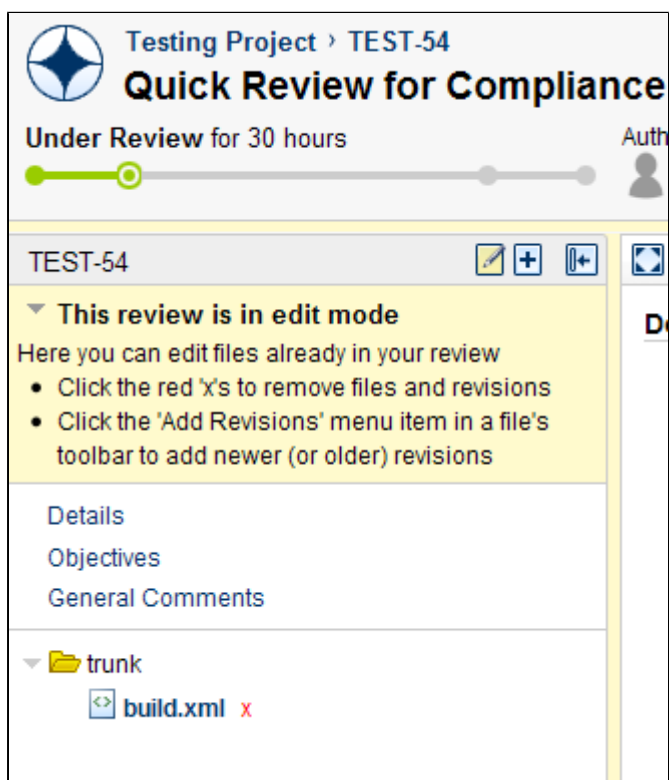
- Add files to a review by clicking the 'Add content' button.
- Remove files by clicking the 'Edit' button in the left navigation panel, then clicking the red cross icons beside files to be removed.

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



Commenting on reviews

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

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

On this page:

- [Locating existing comments](#)
 - [Use the Review Activity](#)
 - [Use the file tree](#)
- [Adding comments](#)
- [Draft comments](#)

Locating existing comments

Use the Review Activity

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

Activity 8

[Review Activity](#)
Click to see recent comments

[Mark as read](#)

Keyboard shortcuts

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

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

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

Scroll through the comments or use the keyboard shortcuts:

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

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

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

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

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

I see the SPECIFIC_DIFF is used only in:

- CreateReviewFromChangesetAjaxAction,
- ReviewCreationHelper

Mark as read · 01 Jul

Lukasz Pater commented on file [/src/java/.../util/ReviewCreationHelper.java](#)
Why drop back to booleans here?

Mark as read · 29 Jun

Marek Parfianowicz
Good point. I'll change method signature.

Mark as read · 01 Jul

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

Use the file tree

The number beside a filename, in the left-hand panel of a review, indicates the number of comments on that file. (The number of unread comments, if any, is shown in brackets.)

CRUC-7296: overflow-auto for code snippets

Activity 1

CR-FE-9472 8

Details
Objectives
General Comments
Number of files included: 3
FE-git
src/content/static/2static/style/cru
snippet.css 8

Attachments
exceed1.png
exceed2.png

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

40dda3c 108d008 710a3e3 b865b6d 82fa88f

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

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

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

[Add a file comment.](#) Click on source lines to add an inline comment.

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

Adding comments

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

To comment on...	Do this...	
The whole review	Click General Comments (in the left-hand panel), and simply type your comment (under 'General Comments').	
A source file under review	Click on the file in the left-hand panel, then click Add a file comment (just above the source code listing).	

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. A comment can only be deleted by the author of the comment.

 Read about [flagging defects](#) too.

Draft comments

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

Sending a review's comments via email

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

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

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

Changeset discussions

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

 You need to be logged in to create changeset comments.

Adding comments to changesets

To add a comment to a changeset:

1. Click on a changeset on the **Commits** tab for the repository. Display comments by clicking **Discuss** at the upper right corner, or the speech bubble icon  in the left margin.
2. Click **Add a comment** (under the repository details near the top left).
3. Type your comment. If required, you can tag your comment as being a defect note by clicking **Defect**.
4. Click **Post**.

Once submitted, others can respond to your comment by clicking **Reply**. Replies are threaded as separate comment discussions. You can right-click on the permalink icon



to copy a link to the comment. The comment author can edit or delete their own comments.

To hide the changeset comments, click the page icon



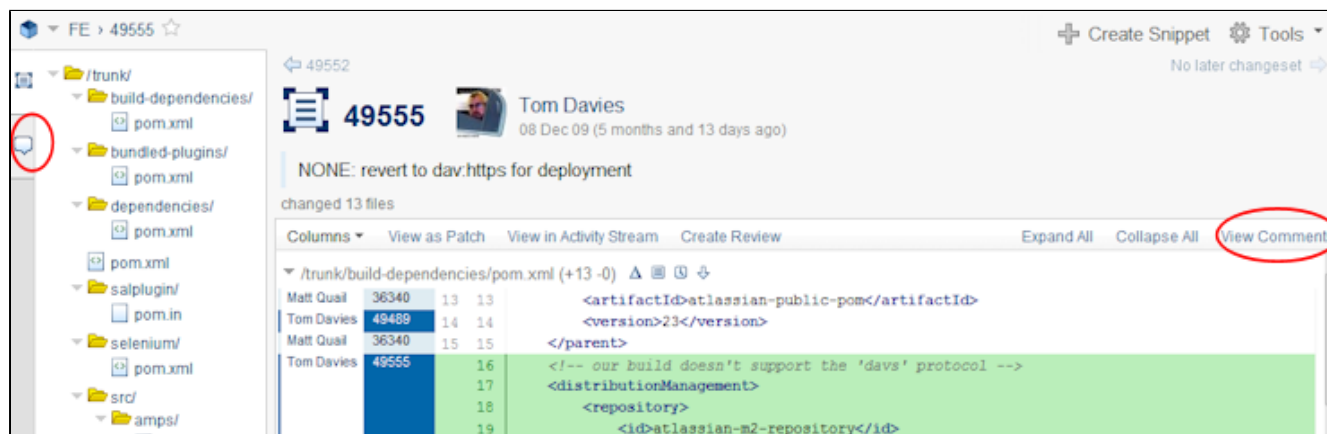
. You can display the comments panel by clicking the speech bubble icon



again.

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

Screenshot: Opening Changeset Discussions



Turning changeset discussions on and off

You can turn off changeset discussions in the Admin area:

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

By default, changeset discussions are on.

Notifications

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

Flagging defects

[Comments](#) in Crucible can be used to flag a defect in the code under review.

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

Screenshot: Defects

Is this variable name sensible?

Defect: Minor | Select Requires Re-Review | Select Classification | Post | Cancel

Geoff Crain says:
i think so - may not be complex enough, though
[Leave Unread](#)

13 May

```

7 + import java.io.Serializable;
12 8
13 - public ElementInfo{
14 -     BaseFileInfo containingFile, int re
15 -     SourceRegion region, int complexity,

```

Select Classification

- Missing
- Extra (superfluous)
- Ambiguous
- Inconsistent
- Improvement desirable
- Not conforming to standards
- Risk-prone
- Factually incorrect
- Not implementable
- Editorial

context,

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

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

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

Viewing reports

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

On this page:

- [Viewing the Review Blockers report](#)
- [Viewing the JIRA Blockers report](#)

See also:

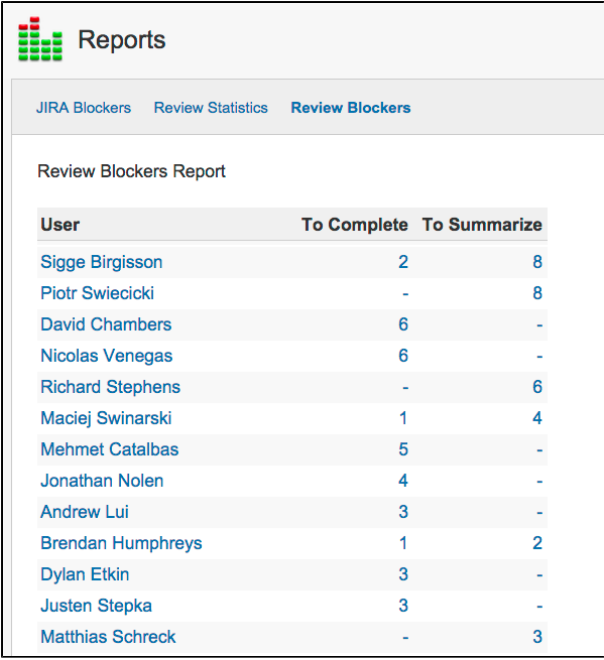
- [Viewing the Review Coverage report](#)

Viewing the Review Blockers report

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

1. Click the dropdown arrow next to the **Reviews** tab at the top of the page and select **Reports**.
2. Click **Review Blockers** (under the 'Reports' sub-tab).
 - Click a user's name to go to their '[Activity](#)' screen.
 - Click a number in the 'To Complete' or 'To Summarize' column to go to a list of reviews for that user.

Screenshot: 'Review Blockers' Report



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

Viewing the JIRA Blockers report

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

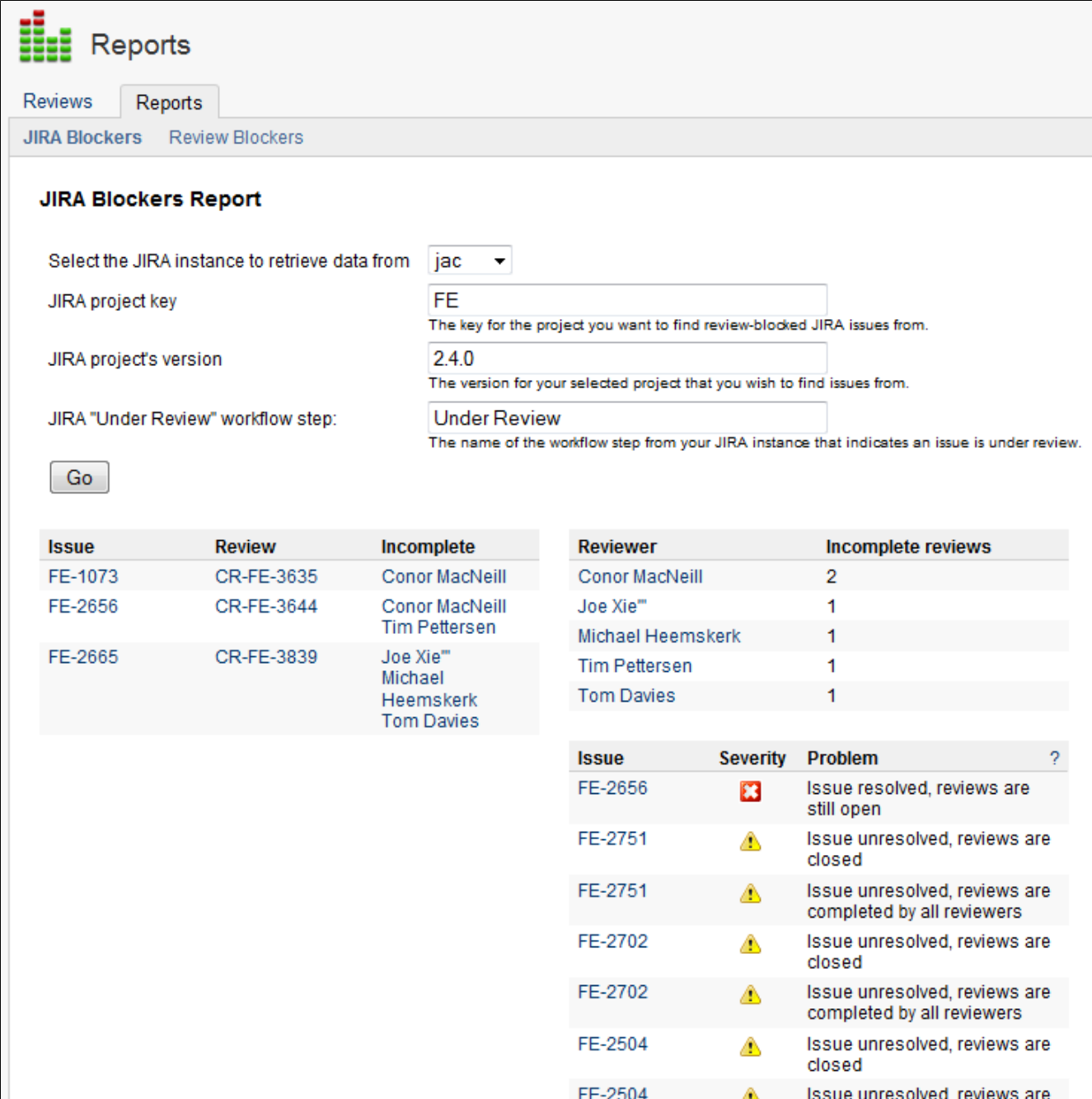
To view the 'JIRA Blockers' report:

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

The 'JIRA Blockers' report displays the following information:

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

Screenshot: 'JIRA Blockers' Report



Reports

Reviews | **Reports**

JIRA Blockers | Review Blockers

JIRA Blockers Report

Select the JIRA instance to retrieve data from:

JIRA project key:
The key for the project you want to find review-blocked JIRA issues from.

JIRA project's version:
The version for your selected project that you wish to find issues from.

JIRA "Under Review" workflow step:
The name of the workflow step from your JIRA instance that indicates an issue is under review.

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

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

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

Review Coverage report

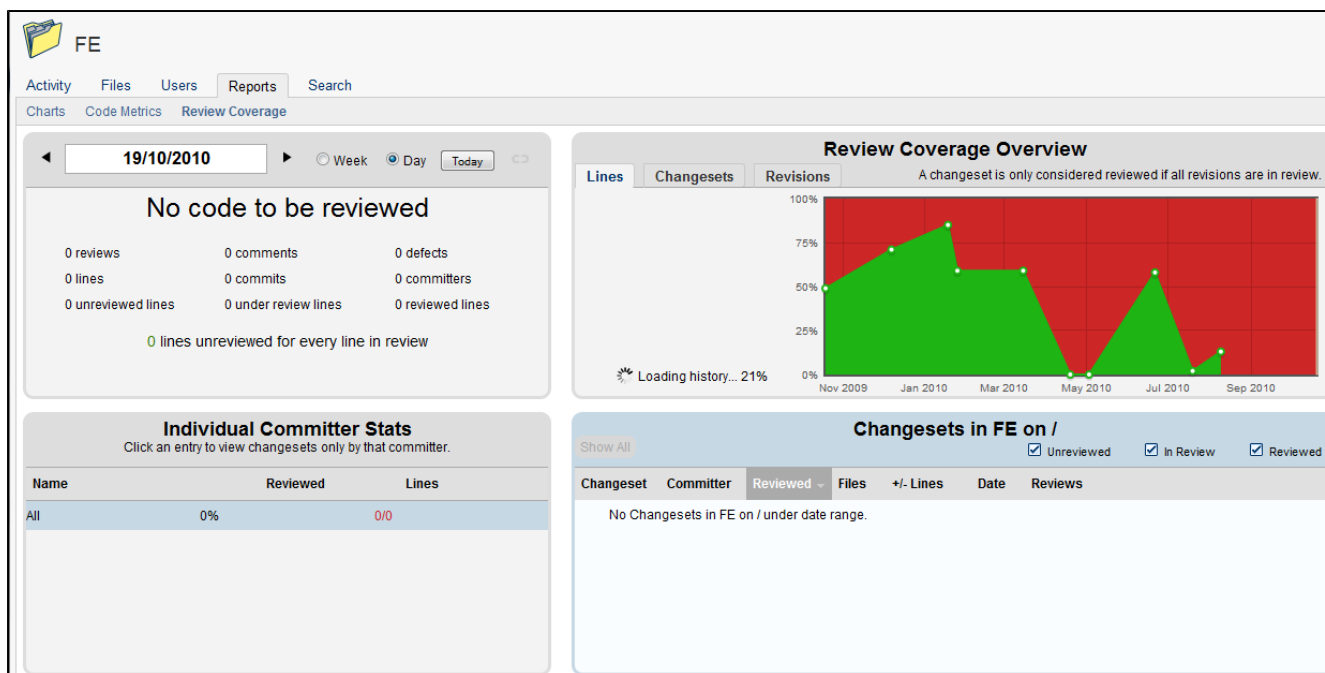
Crucible has useful reports that show you detailed statistics on review activity. The Review Coverage report allows you to see how much of the code, and which files, in your repository have been reviewed, and when. You can also access the reviews.

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

On this page:

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

Screenshot: *The Review Coverage report*



Opening the Review Coverage report

To open the Review Coverage report:

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

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

Using the Summary Panel

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

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

Screenshot: Summary Panel in the Review Coverage report



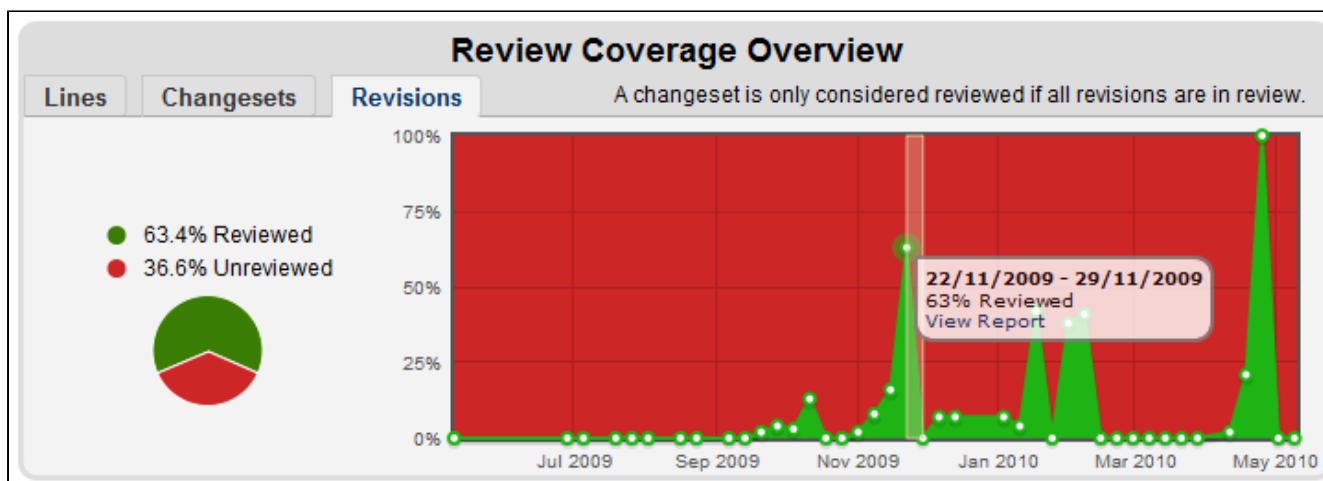
Note: Code coverage metrics only apply for post-commit reviews.

Using the Review Coverage Overview

The Review Coverage Overview shows a timeline of reviews, compared against their percentage of coverage. Hover your mouse cursor over the data points on the graph to see granular information and click through to a detailed weekly report.

You can click the tabs to view the coverage expressed as a percentage of lines of code, changesets or revisions.

Screenshot: Overview Panel in the Review Coverage report



Using the Individual Committer Stats panel

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

Screenshot: Individual Committer Statistics in the Review Coverage report

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

Using the Changesets panel

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

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

Completing your review

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

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



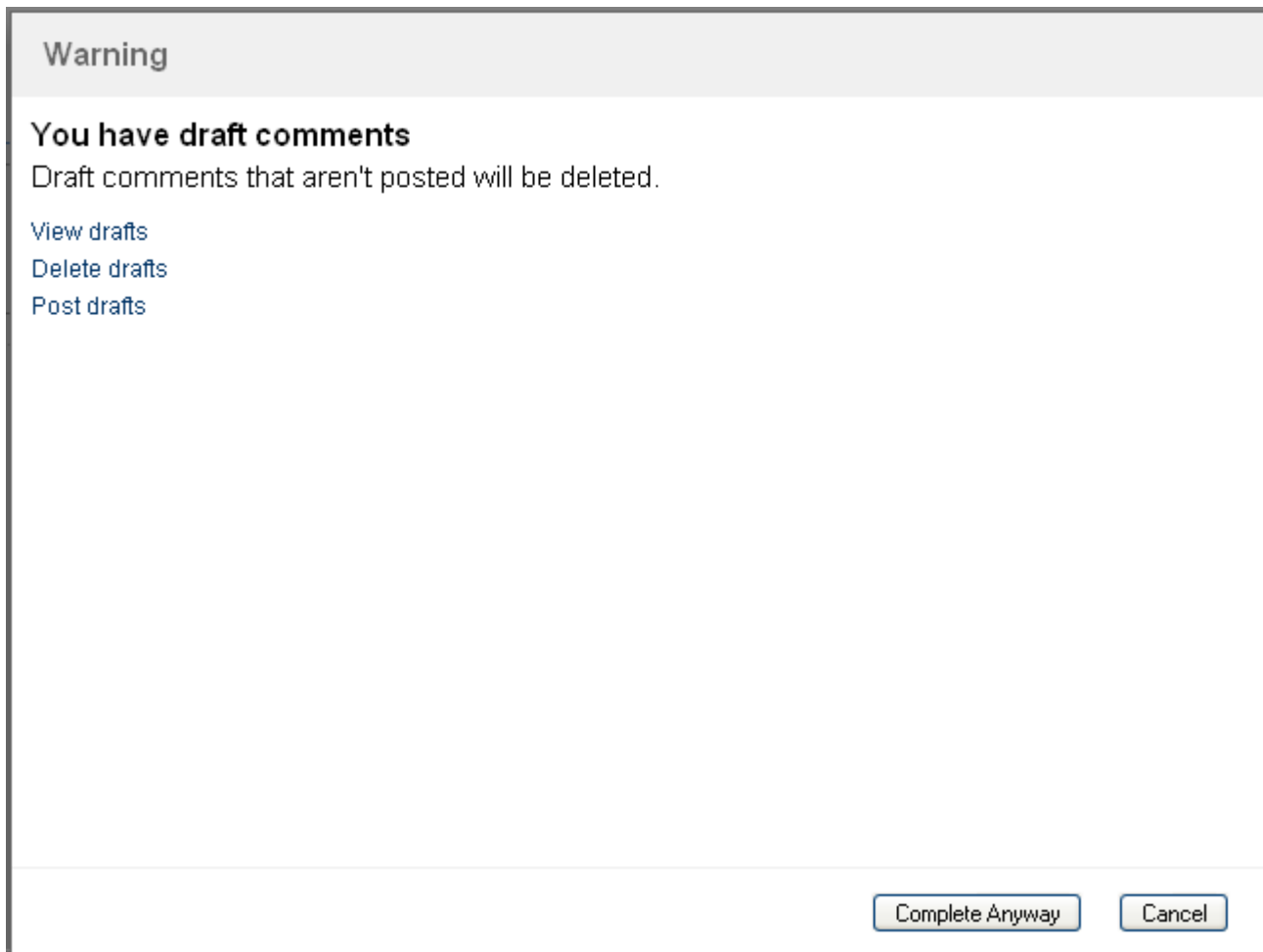
i Only people with the 'Complete' [permission](#) can complete a review.

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

Review Complete

You have marked TEST-27 as completed.
The following reviewer is not yet finished:

 [Geoff Crain](#)

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

Reviews requiring your attention:

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

[Dashboard](#) [Close](#)

Using the Review History Dialog

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

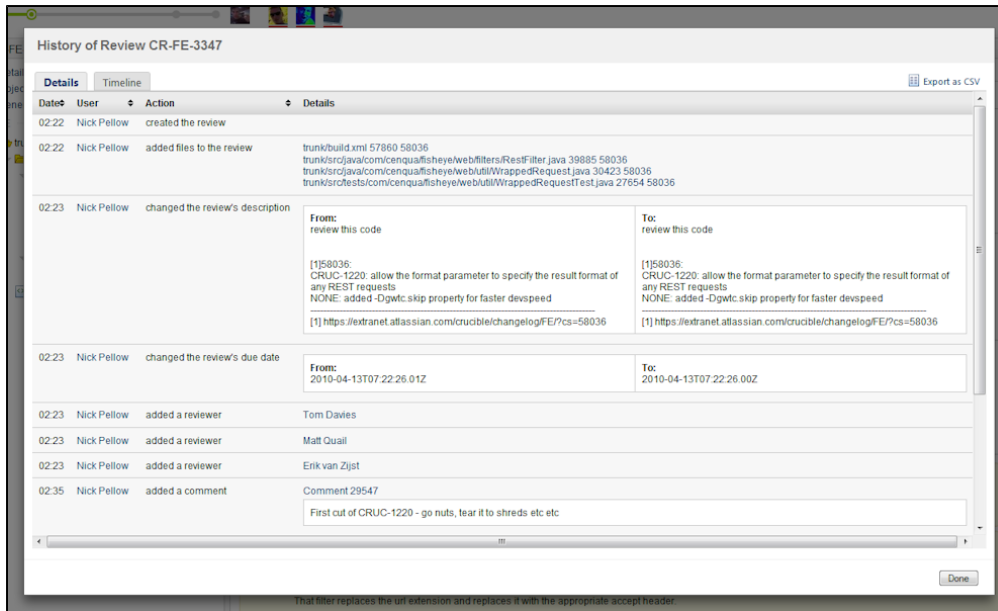
To open the Review History dialog:

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

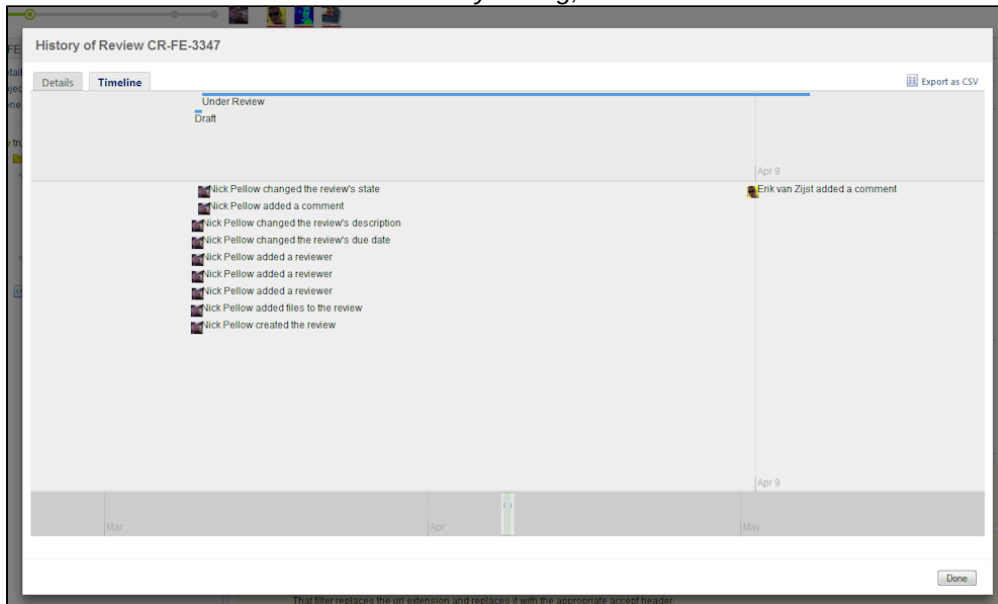
Click the **Timeline** tab at the top of the History dialog to see the review's events in a horizontal calendar. You can drag the calendar and the timeline at the bottom to scroll to other events.

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

Screenshot: The Crucible Review History dialog



Screenshot: The Crucible Review History Dialog, Timeline View



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 **Close**.

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

Normally, we recommend that you wait for all reviewers to [complete their reviews](#), before summarizing or closing the review.

The reviews that the reviewers have completed will be in your **Ready to Close** menu on the Dashboard.

To summarize a review,

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

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

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

TEST-27 test
Under Review Author & Moderator: Rosie Jameson

Under Review for: 102 minutes
Comments: 1

Reviewers:
Geoff Crain
Edwin Dawson

Review Objectives
documentation testing

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

Summarize Review

Summarize the review outcomes (optional)

Testing successful. Thanks for your time, everyone.

Continue Without Closing Close Review

Screenshot: Review Closed

Review Closed

Review Summary

Testing successful. Thanks for your time, everyone.

Reviews requiring your attention:

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

Dashboard

Close

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

- Reviews in the 'Review' or 'Summarize' state can be closed.
- Reviews in the 'Summarize' or 'Closed' state can be re-opened. Re-opening changes the review's state back to 'Under Review', allowing all participants to add comments.
- See [this page](#) for a list of the states that a review can be in.

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

Managing your reviews

See:

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

Using Review Reminders

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

Review authors and moderators can also do the following:

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

Reminders are only sent if Crucible's SMTP server is configured. Please see [Configuring the SMTP server](#).

Preset Reminders

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

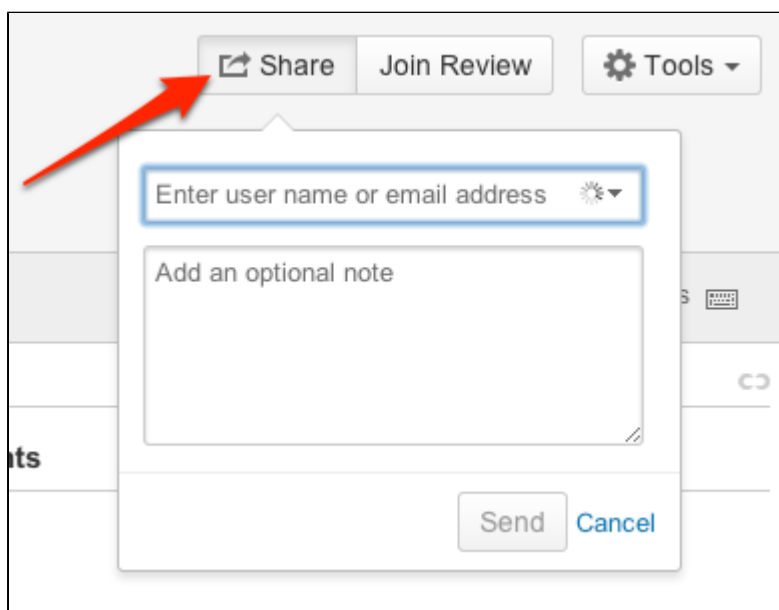
To add a reminder, firstly edit the review, then click **Add a reminder**. You can edit the reminder period.

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

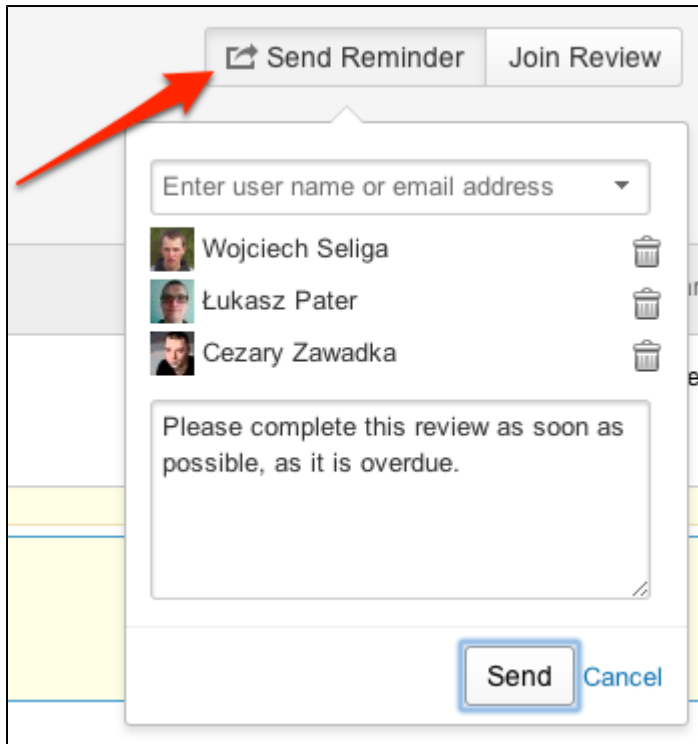
Manual Reminders

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

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



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



Moving a review to another project

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

To move a review between projects:

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

Screenshot: Changing a Review's Parent Project

TEST-29 A test review
Under Review Author & Moderator: Rosie Jameson

Under Review for: 0 seconds Display Preferences Filter

Edit Review

Review Details

Title:
A test review

Project: Testing Project

Moderator: Rosie Jameson

Author: Rosie Jameson

Select Reviewers:
Start typing the reviewer's name then press enter to select.
☒ Geoff Crain
☐ Allow anyone to join

Get Reviewer Suggestions:
Suggest Reviewers

Due Date:

Linked JIRA Issue:
Issue key: Link

Linked Review
Review Key: Link

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

Reset

Abandon Review Done

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.

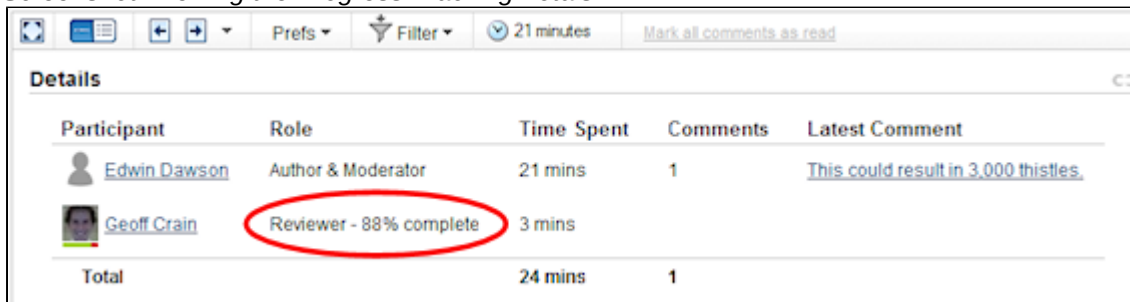
When participating in [iterative reviews](#), progress tracking also takes lines of code and revisions into account.

Viewing the progress tracking totals

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

i 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

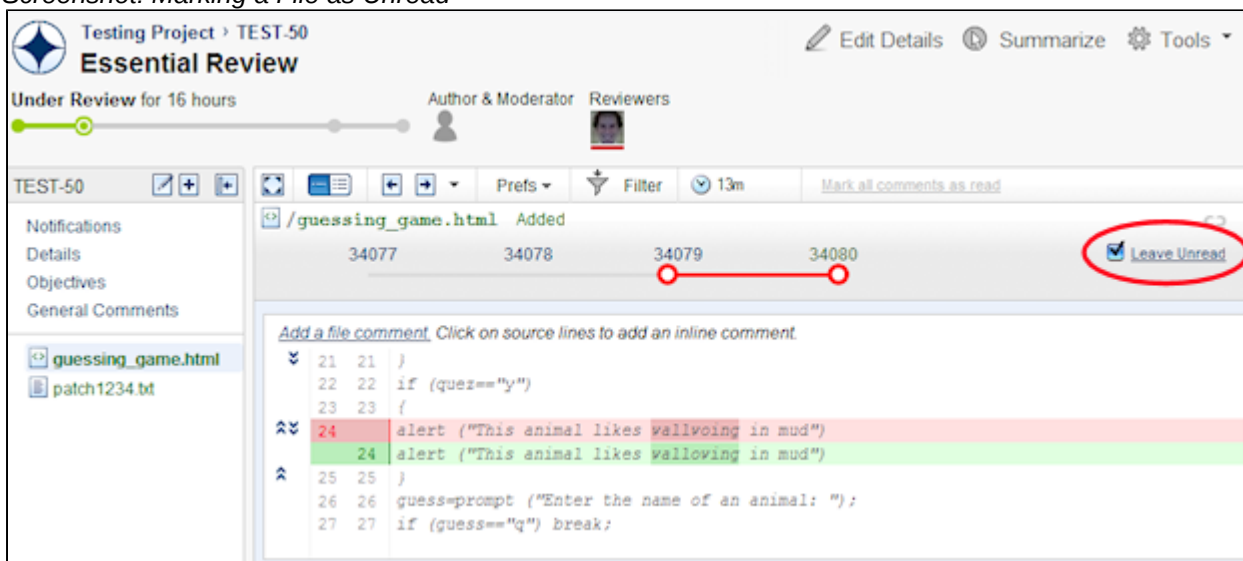


Participant	Role	Time Spent	Comments	Latest Comment
Edwin Dawson	Author & Moderator	21 mins	1	This could result in 3,000 thistles.
Geoff Crain	Reviewer - 88% complete	3 mins		
Total		24 mins	1	

How to adjust progress tracking on a review

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

Screenshot: Marking a File as Unread



Testing Project > TEST-50
Essential Review

Under Review for 16 hours

Author & Moderator: Reviewers:

TEST-50

Notifications
Details
Objectives
General Comments

guessing_game.html
 patch1234.txt

/guessing_game.html Added

34077 34078 34079 34080

☒ Leave Unread

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

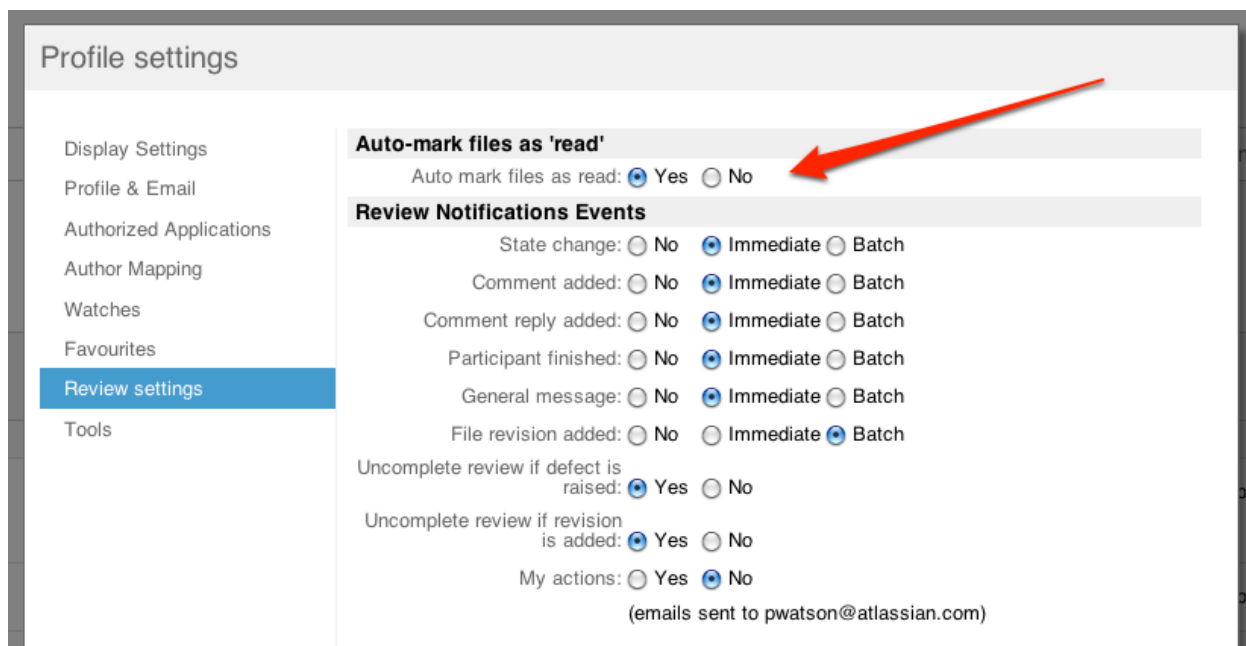
```

21 21  }
22 22  if (quiz=="y")
23 23  {
24 24  alert ("This animal likes wallvoing in mud")
24 24  alert ("This animal likes wallowing in mud")
25 25  }
26 26  guess=prompt ("Enter the name of an animal: ");
27 27  if (guess=="q") break;

```

Adjusting settings for progress tracking

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



Further reading

You may also want to learn about Crucible's [Time Tracking](#) feature.

Using Time Tracking

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

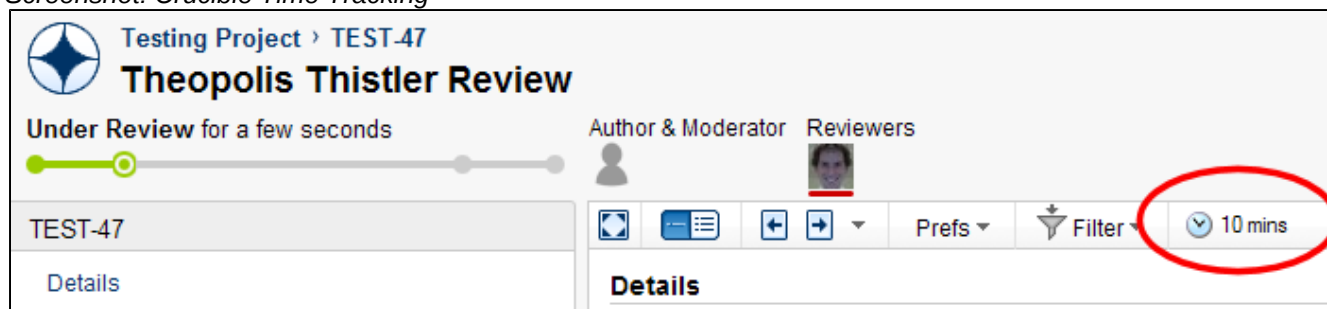
On this page:

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

How time tracking works in Crucible

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

Screenshot: *Crucible Time Tracking*



How to adjust the time tracked on a review

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

Viewing the time tracking totals

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

Screenshot: *Crucible Tracking Totals*

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

JIRA integration

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

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

Screenshot: JIRA Time Tracking Integration

Further reading

You may also want to learn about Crucible's [Progress Tracking](#) feature.

Viewing people's statistics in Crucible

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

On this page:

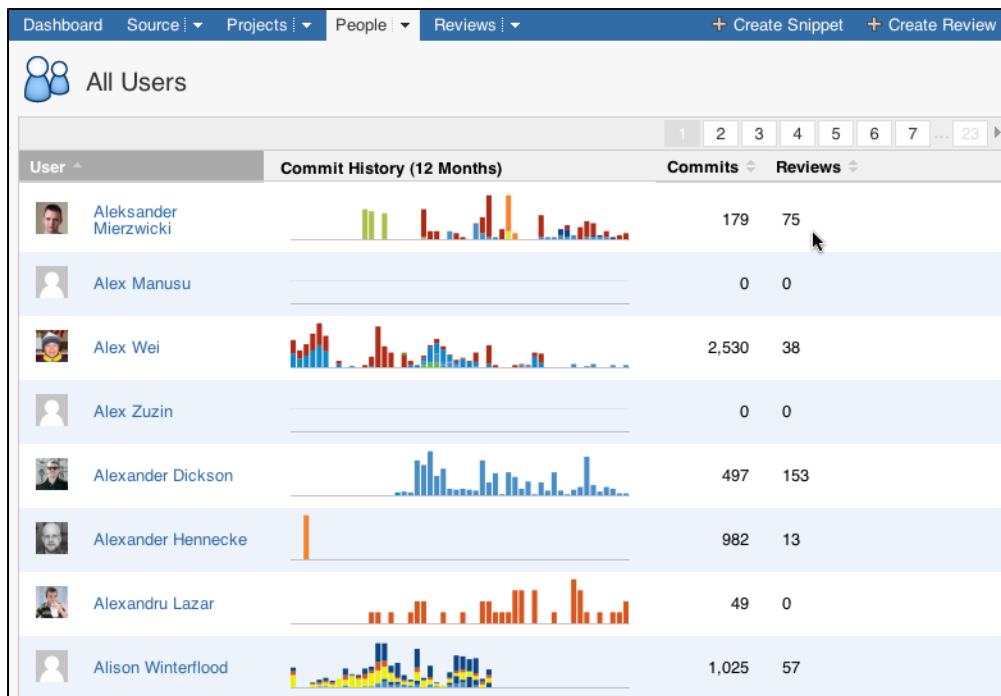
- [Opening the list of People](#)
- [Viewing a Person's Activity Screen](#)
- [Viewing charts of a person's activity](#)

Opening the list of People

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

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

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



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.

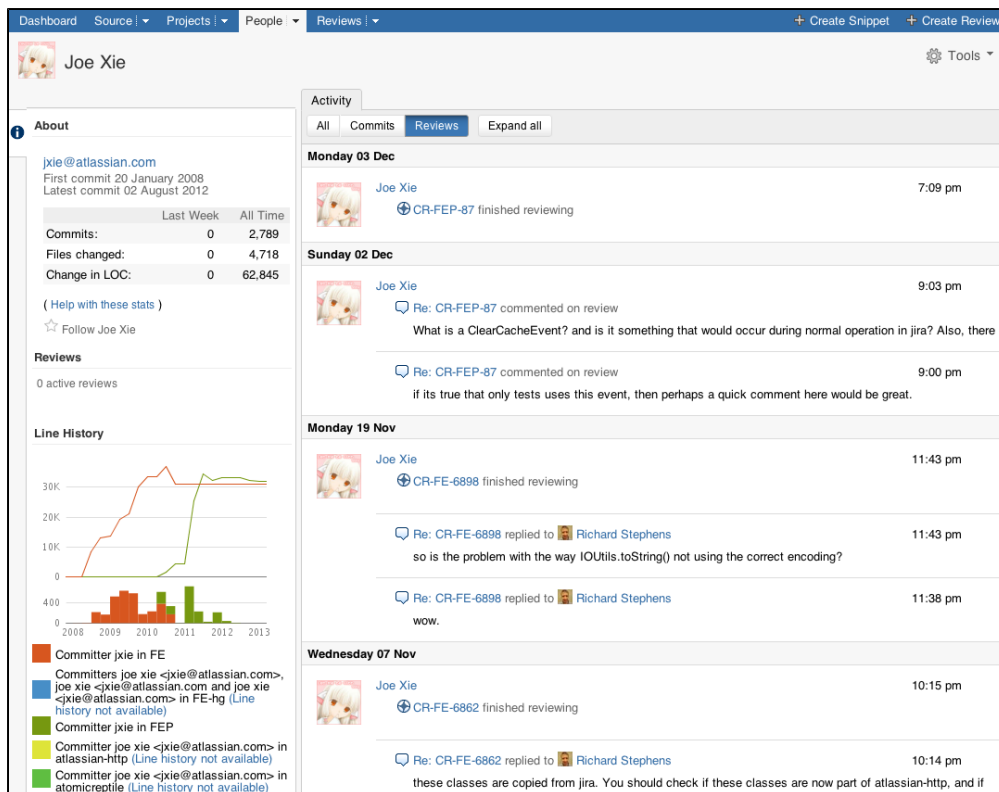
The right hand pane displays a list of all activity for this user. You can:

- 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



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

The screenshot shows the Crucible user profile for Joe Xie. The top navigation bar includes links for Dashboard, Source, Projects, People, and Reviews. The profile header shows the user's name, email (jxie@atlassian.com), and commit history (First commit: 20 January 2008, Latest commit: 02 August 2012). The 'About' section displays a table of activity metrics:

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

Below the table is a 'Line History' chart showing activity from 2008 to 2013. The chart has three data series: Committers jxie in FE (red line), Committers jxie in FE (blue line), and Committers jxie in FE (green line). The chart shows a significant increase in activity starting around 2010, peaking in 2011, and then declining. The 'Reviews' section shows a list of reviews, including CR-FEP-87 and CR-FEP-6898, with comments and replies. The 'Activity' section shows a list of reviews, including CR-FEP-87 and CR-FEP-6898, with comments and replies.

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

([Help with these stats](#))

☆ Follow Joe Xie

Reviews

To Review (0)

Ready to Close (0)

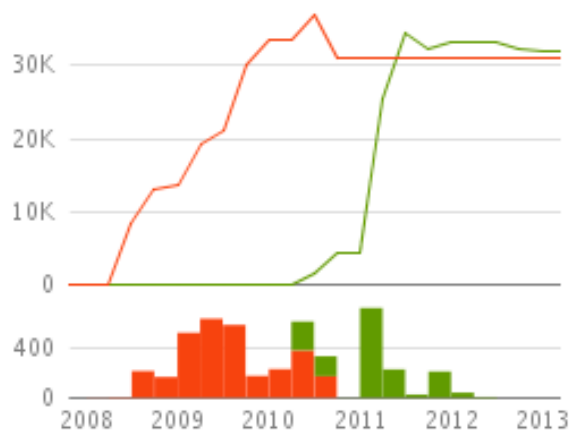
Out For Review (0)

Open (2)

Closed (1017)

⚠ Counts may include data that you cannot access.

Line History



- Committer jxie in FE
- Committers joe xie <jxie@atlassian.com>, joe xie <jxie@atlassian.com> and joe xie <jxie@atlassian.com> in FE-hg ([Line history not available](#))
- Committer jxie in FEP
- Committer joe xie <jxie@atlassian.com> in atlassian-http ([Line history not available](#))
- Committer joe xie <jxie@atlassian.com> in atomicreptile ([Line history not available](#))
- Committer joe xie <jxie@atlassian.com> in commithooks ([Line history not available](#))
- Committer joe xie <jxie@atlassian.com> in remote-issue-crea... ([Line history not available](#))

Commit Activity

52 week commits volume

• Reviews

The Reviews section shows several filters that you can click to constrain the review items shown in the right-hand pane. The options are **To Review**, **Ready to Close**, **Out For Review**, **Open** and **Closed**.

• Line History

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

• Commit Activity

The Commit Activity section shows four smaller charts; the first showing the volume of commits over a 52 week period; the second showing the relative number of commits on days of the week; the third showing the relative number of commits by the hour of the day when they were lodged; the last shows a commit calendar.

• Committer Mappings

The Committer Mappings section displays username mappings from various systems if they have several usernames in play.



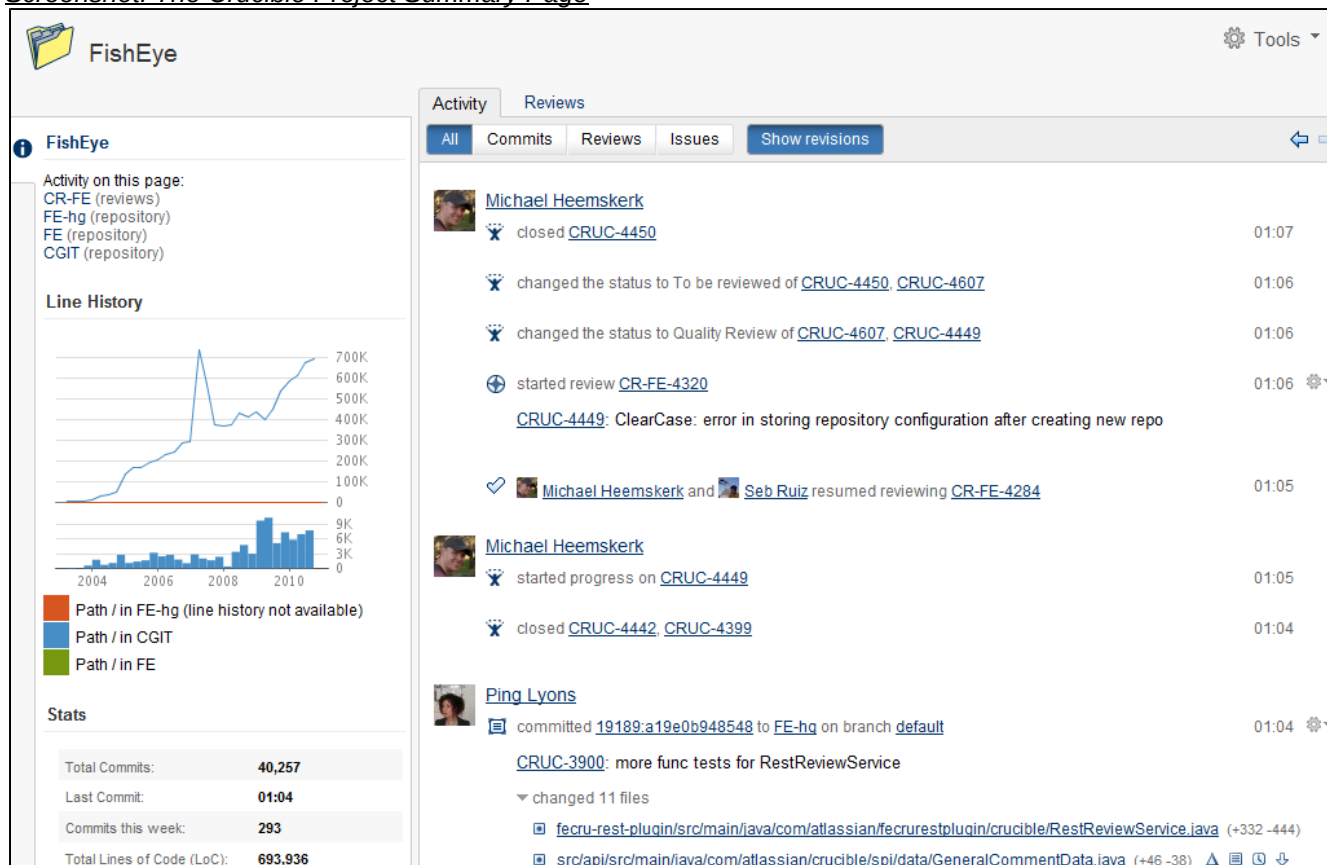
Viewing Project Statistics

This page explains the layout of the Project Summary page.

On this page:

- [Project Name Panel](#)
- [Project Line History Panel](#)
- [Project Stats Panel](#)
- [Project Commit Activity Chart](#)

When you click through to a Crucible Project from the [Projects Tab](#), the '**Project Summary**' screen opens.

Screenshot: The Crucible Project Summary Page

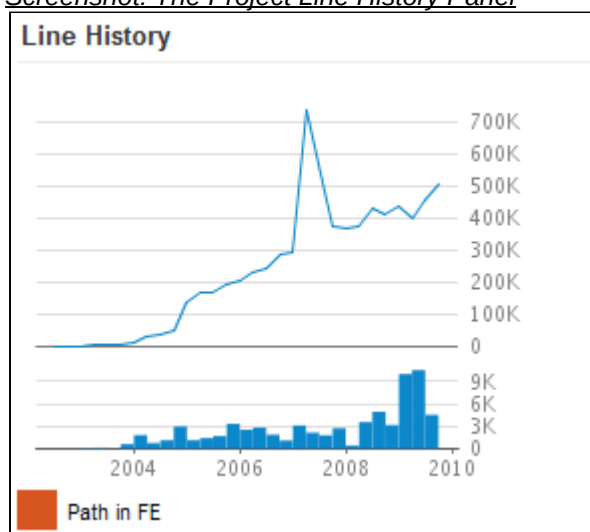
In the right hand pane, you can see an activity stream relating to this project. In the left hand pane, you can see various statistics charts relating to the project in context. These appear in a reduced size until you click them, when they will expand to show more information.

Project Name Panel

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

Project Line History Panel

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

Screenshot: The Project Line History Panel

Project Stats Panel

This panel contains a chart showing numerical data for commits, files changed and lines change, graphed over time.

Screenshot: The Project Stats Panel

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.

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.

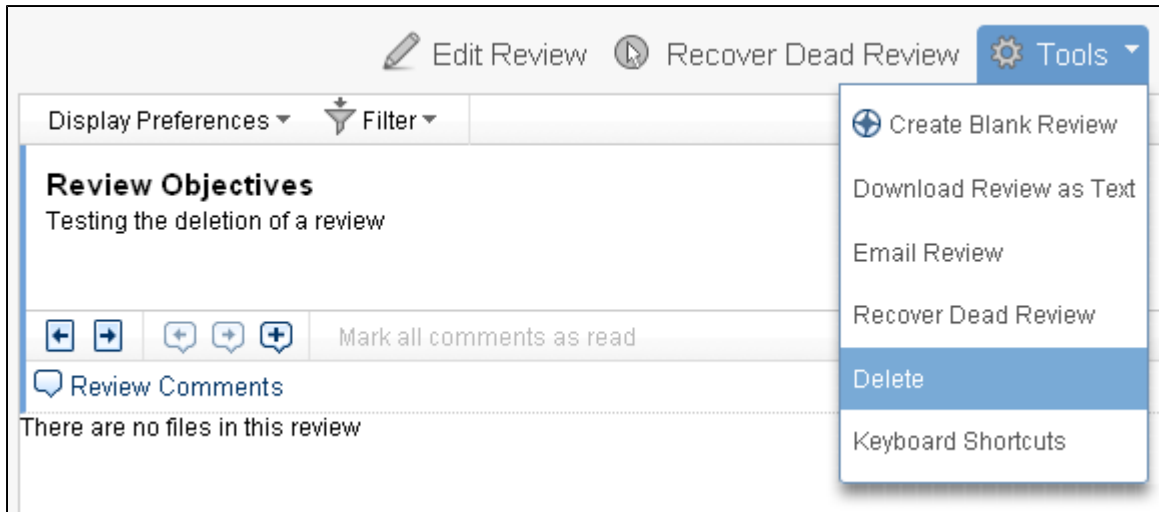
Related page:

- [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



Searching Crucible

Crucible has a powerful search engine that allows you to find reviews. There are two methods for searching in Crucible:

- **Quick Search** — The Quick Search allows you search all Crucible projects by entering a single search string. This search is the default search and will suggest "quick nav" results (header search box only). Results are weighted by most recent edit date.
- **Filtering Reviews** — An alternative method for searching for reviews is to display all reviews and apply a custom filter to the list. This is generally slower than searching, but allows you to specify filter criteria against a range of fields.
- **Comment Search** — If you want to find specific [review comments](#), Crucible provides a powerful comment search.

See also [Searching FishEye](#).

On this page:

- [Using the Quick Search](#)
- [Filtering reviews](#)
- [Searching for Review Comments](#)

Using the Quick Search

Before you begin:

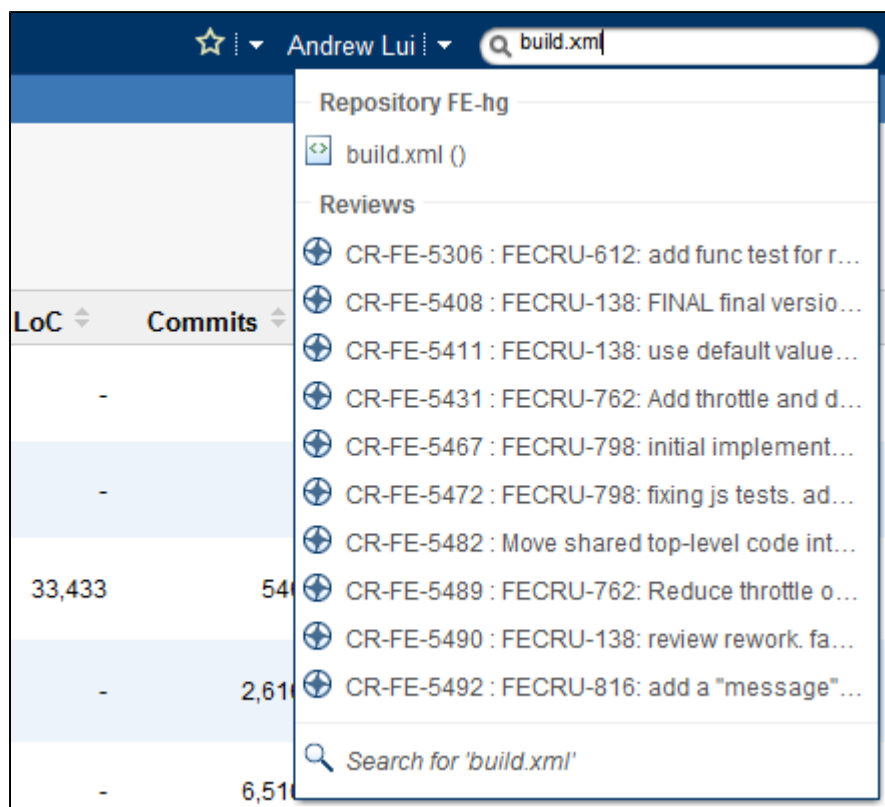
- The Quick Search will also return changesets and files, if you are using [FishEye](#) with Crucible. For information on searching FishEye, see [Searching FishEye](#) in the FishEye documentation.

To search Crucible using the Quick Search:

1. Enter your search terms in the search box in the Crucible header. Crucible offers a number of criteria that you can use to refine your results, see [Refining your Quick Search Criteria](#) below.
2. Results will appear in a dropdown, as you type. Results will attempt to be matched against the review name, project and user.
 - If you want to use a quick search result, use the up- and down-arrows on your keyboard and press enter or use your mouse to select the item.
 - If the quick search results don't have what you are looking for, press enter to run a search. Ensure that no items in the dropdown are selected when you press enter.
3. The Quick Search results page will be displayed. You can filter your results further, as described in [Filtering Quick Search Results](#) below. Results are sorted by relevance and boosted if they were edited recently. A maximum of 10 results are displayed per page.
 - If you have **integrated your Crucible instance with a JIRA instance**, you can display a summary of any JIRA issues referenced in your search results by hovering over the issue key. For more details, see [JIRA integration in Crucible](#).
4. If you want to run another search, enter your new criteria in the main search box or in the search box in the header.

Note, only the search box in the header provides "quick nav" results.

Screenshot: Quick Search displaying "quick nav" matches



Refining your Quick Search Criteria

You can refine your search criteria before executing the search:

Search Tool	Description	Example
Field Handles	Use a field handle in your criteria to restrict your search to a particular field. Note, you cannot have multiple field handles in a query. - detail — Search against title, objective, key, linked reviews and linked issues. - reviewcomment — Search against review comments.	
Searching for Discrete Strings	Enter a specific string within quotation marks and Crucible will match against the exact string. Note, this search is not case-sensitive.	Enter "CR-2818" and Crucible will only return results that match that exact string, i.e. it will not return a result with CR-FE-2818 or CR-28189.

Filtering Quick Search Results

You can filter Quick Search results using the controls in the left panel of the Quick Search page:

Filter	Description
All repositories	Type a repository name into the field, or click the down arrow to see a list of repositories.

Source type	Click Files and directories , Commit messages , Diffs , Content or Committers to restrict the search results to just that source type.
All projects	Select or enter the name of the project that you want to restrict your results to. For example, if you enter 'CR' then the search results page will refresh to display only reviews in the 'CR' project.  <i>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.</i>
Reviews	Click this link to restrict your results to reviews that have a title, objective, key, linked reviews or linked issues that match the search criteria.
Comments	Click this link to restrict your results to reviews that have comments that match the search criteria.
Last modified	Filter by the date of the last change.
By	Filter by author name.

Filtering reviews

Crucible allows you to view all the reviews/snippets that you are involved with, as well as everybody's reviews/snippets. You can filter these lists to find reviews.

To filter a list of reviews:

1. Click **Reviews** in the header.
2. Click the list of reviews that you want to start with, in the sidebar of the 'Review Dashboard', e.g. 'All Open Reviews'.
3. Click **Custom Filter** in the reviews sidebar.
4. Update the filters with your search criteria (see table below) and then click **Apply Filter**.

Filter	Description
Title	Find reviews by searching for words within the title.
Project	Find reviews under a particular project.
Author	Find reviews moderated by a particular authors .
Moderator	Find reviews moderated by a particular moderators .
Creator	Find reviews created by a particular creator .
Reviewer	Find reviews that are reviewed by a particular reviewer . This will default to the user logged in.
Reviewer Status	This is reliant on the above filter and is used to show reviews that have either been completed by the reviewer, not completed or all reviews.
Type	Choose either Review or Snippet .
Match Roles	To use all the above filters, choose all . To use any of the filters, choose any .
Review state checkboxes	Check any of the review state checkboxes (e.g. Draft , Pending Approval) to filter for reviews in those states.

Searching for Review Comments

To search for review comments:

1. Click **Reviews** in the header.
2. Enter your search string in the 'Comment Search' section at the bottom of the reviews sidebar.
3. Click **Search Comments**.
4. The 'Comment Search' page will display your results. You can refine your search using the search criteria on the page:

Project	Find comments on reviews under a particular project.
Comment content	Find comments that contain the specified text.
Review Permalid	Find comments made on the specified review.
After	Find comments made after a particular date.
Before	Find comments made before a particular date.
Comment Author	Find comments made by a particular user.
Search Type	Filter for comments marked as Defects . Check Comments to find comments that are not flagged as Defects .
Review State	Find comments on reviews that are in a particular state. See <i>Review State Filter</i> (above).
Ranking	Find defects have been given a particular ranking (e.g. 'Major', 'Minor').
Ranking	Find defects have been given a particular ranking (e.g. 'Major', 'Minor').

Screenshot: Search Comment Filter Options

Search Criteria

Search Criteria

Project:

Comment content:

Review Permalid:

After:

Before:

Comment Author

Search Type:

☐ Defects ☐ Comments

Review State:

☐ Draft ☐ Pending Approval ☐ Under Review
☐ Summarize ☐ Closed ☐ Abandoned
☐ Rejected ☐ Needs Fixing

Metrics:

Ranking:

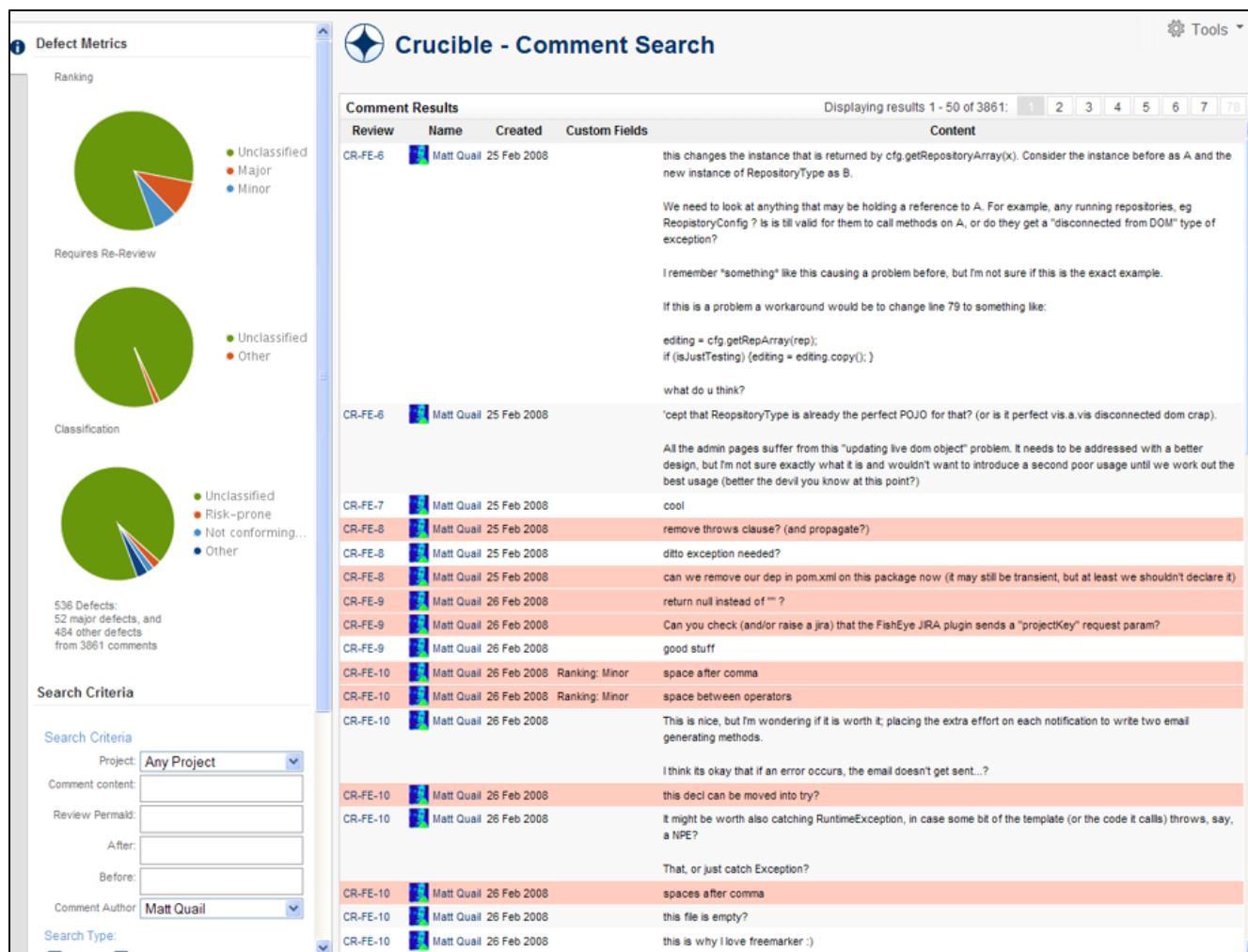
Requires Re-Review:

Classification:

Displaying Defect Metric Charts for Comment Search Results

Once you have retrieved results for a review comment search, you can click **Defect Metrics** in the left navigation pane to display defect classification charts.

Screenshot: Comment Search Results



JIRA integration in Crucible

JIRA is Atlassian's issue tracking and project management application.

When Crucible is integrated with JIRA, you and your team get all the benefits described on this page:

In Crucible, you can:

- See all the Crucible reviews related to a JIRA issue
- Create a Crucible review directly from an issue in JIRA
- Link your Crucible review to a JIRA issue
- Create a JIRA issue from a review comment
- Transition JIRA issues automatically
- Transition JIRA issues from within Crucible
- See issues from multiple instances of JIRA
- See open reviews or unreviewed commits for an entire version within JIRA

Related pages:

- [Creating a review from JIRA](#)
- [Creating JIRA issues from the review](#)
- [Transitioning JIRA issues](#)
- [Linking Crucible to JIRA](#)

Note that your Crucible and JIRA instances must be linked to make use of these JIRA integration features. See [Linking Crucible to JIRA](#).

The Crucible project requires a linked JIRA project before issues can be linked to reviews.

Check development progress of a version in JIRA

CRUCIBLE 3.3+

JIRA 6.4+

The Release Hub in JIRA shows the relevant issues and development information for a version – so you can determine which issues are likely to ship at a glance. With JIRA and Stash connected, the release page can warn you about open reviews or unreviewed commits that could cause problems for your release.

Version 2.0 UNRELEASED

Start: 03/Nov/14
Release: 15/Feb/15
[Release Notes](#)

48 Warnings

273 Issues in version

262 Issues done

7 Issues in progress

4 Issues to do

Warnings indicate when the status of a JIRA issue doesn't reflect related development activity. For example: an issue marked complete that has an open pull request should be marked as still being in progress.

Unreviewed Code
These issues have been marked complete but the commits are not part of a pull request or review.

1–4 of 4

P	T	Key	Summary	Assignee
		SSP-1663	UI is not loading in IE8	Andrew Swar
		SSP-1979	Incorrect permissions for new report	Andrew Swar
		SSP-1555	As a developer, I want to view the build status for an issue	Bruce Temple
		SSP-1660	Missing error message when Bamboo is unavailable	Eduardo Soai

1–4 of 4

From the Release Hub you can also:

- Release a version
- Mark a version as complete
- Move incomplete issues to other versions
- Trigger release builds (if JIRA is connected to Bamboo)
- Warnings that help you reconcile what is happening in development with JIRA data.

To view the Release Hub (with the project sidebar enabled), navigate to a project, click on **Releases**, then select a version listed. See [Checking the progress of a version](#) more detailed information about using the Release Hub in JIRA.

See all the Crucible reviews related to a JIRA issue

In a JIRA issue, the Development panel shows the number of reviews that are linked to the issue. Click the **reviews** link to see details of those reviews.

Link your Crucible review to a JIRA 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 specify a different issue key and click **Link** to save it.

Edit Review Details CR-FE-6139

Project: CR-FE

Title: CRUC-5879: Set default due due to duration rounded to nearest half hour instead of exact duration from

Author: Adam Ahmed

Reviewers: [Suggest reviewers...](#) ☒ Allow anyone to join

Seb Ruiz

Tim Pettersen

Objectives:

d3484be6a4122053e4e37457cb2b508566654fdb: CRUC-5879: Set default due due to duration rounded to nearest half hour instead of exact duration from now. This matches the UI options and avoids random due date changes when editing details and dumb-looking due dates showing up in the UI ('Due at 3:14:15 tomorrow' looks dumb)

Due Date:

Linked Review: [Link](#)

Linked Issue: **CRUC-5879** - Review due and reminder dates that change by < second are tracked

[Add Content](#) [Done](#)

Create a JIRA issue from a review comment

When viewing any review comment (general, file, inline), you can click **Create Issue** in the 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.

See [Creating JIRA issues from the review](#) for more details.

Create issue

JIRA Project*

JIRA Issue Type*

Summary*

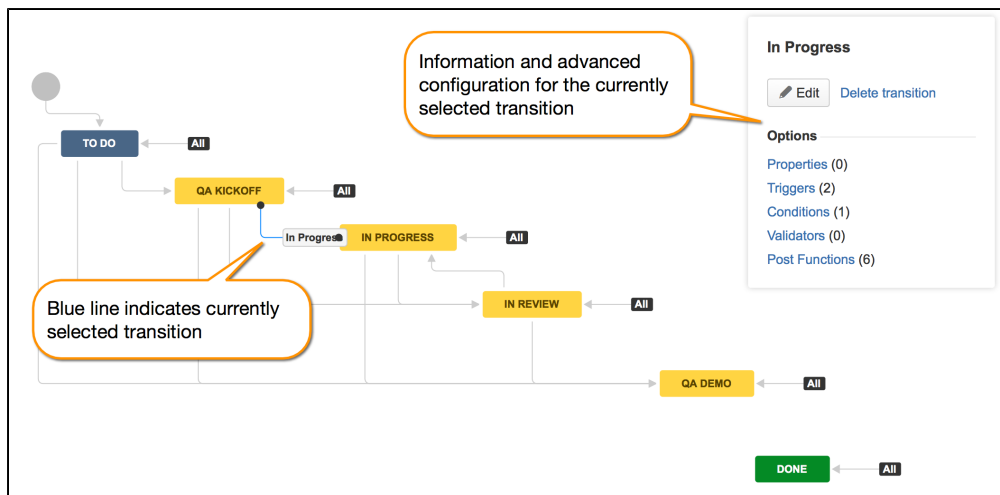
Description

The title of the target page for the help link should be "Setting the REST API token" This is it: https://confluence.atlassian.com/display/FISHEYE/Setting+the+REST+API+token

[Create](#) [Cancel](#)

Transition JIRA issues automatically

Your JIRA workflow can now respond to events in your linked development tools, for when a review is started, your JIRA workflow can be configured to automatically transition the related issue. Configure this from transitions within the JIRA workflow editor. (Available with JIRA 6.3.3 and later.)

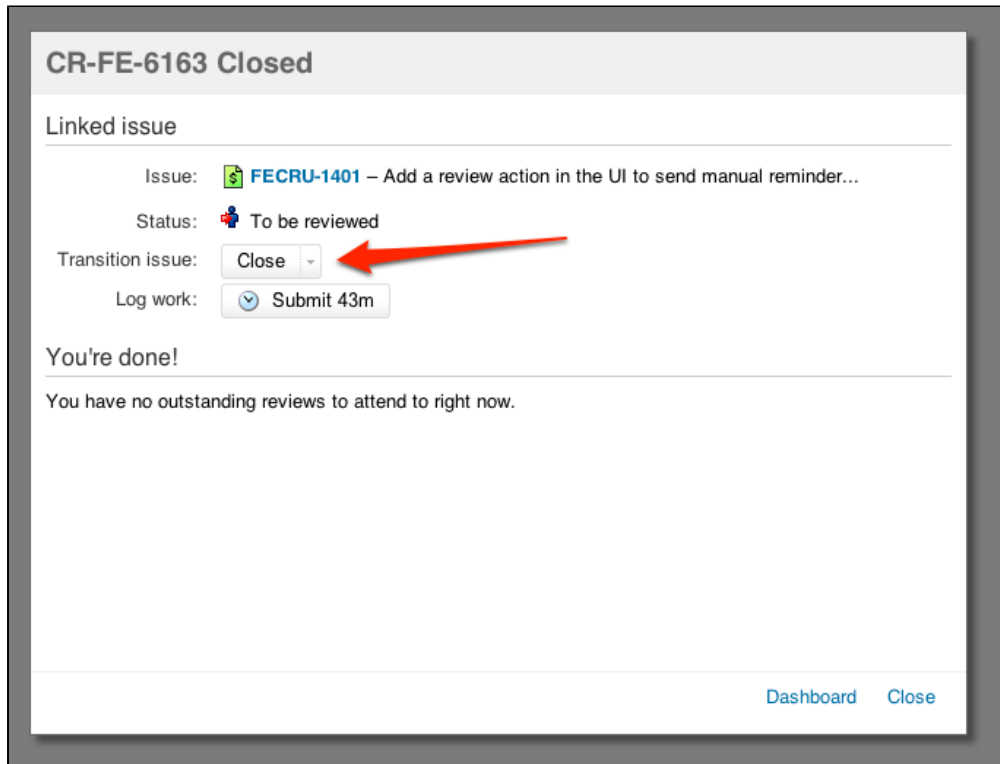


The events available in Crucible are:

- Review started
- Submitted for approval
- Review rejected
- Review abandoned
- Review closed
- Review summarized

Transition JIRA issues from within Crucible manually

For Crucible reviews that have [linked JIRA issues](#), you can advance the JIRA workflow for the issue from within Crucible. You can do this at any time by clicking the linked issue, or when you close the review:



See [Transitioning JIRA issues](#) for more details.

See issues from multiple instances of JIRA

Crucible can [link to more than one JIRA server](#) at a time, so different teams can work with their own projects in different JIRA instances, or a single team can link to issues across multiple JIRA servers.

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:

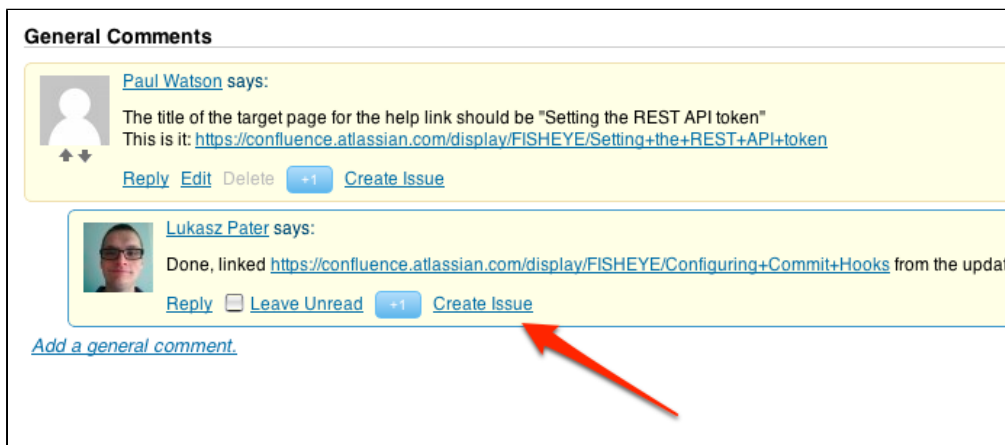
- Tracking of the status of the comment
- A faster way to pull out incidental suggestions raised in reviews as JIRA issues
- A quick link back to the comment from the JIRA issue, using Remote Issue Links.

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 [Creating a review from JIRA](#).

Transitioning JIRA issues

When Crucible is [linked to JIRA](#), you can advance the workflow for a JIRA issue directly from within Crucible.

You can transition a JIRA issue in two ways:

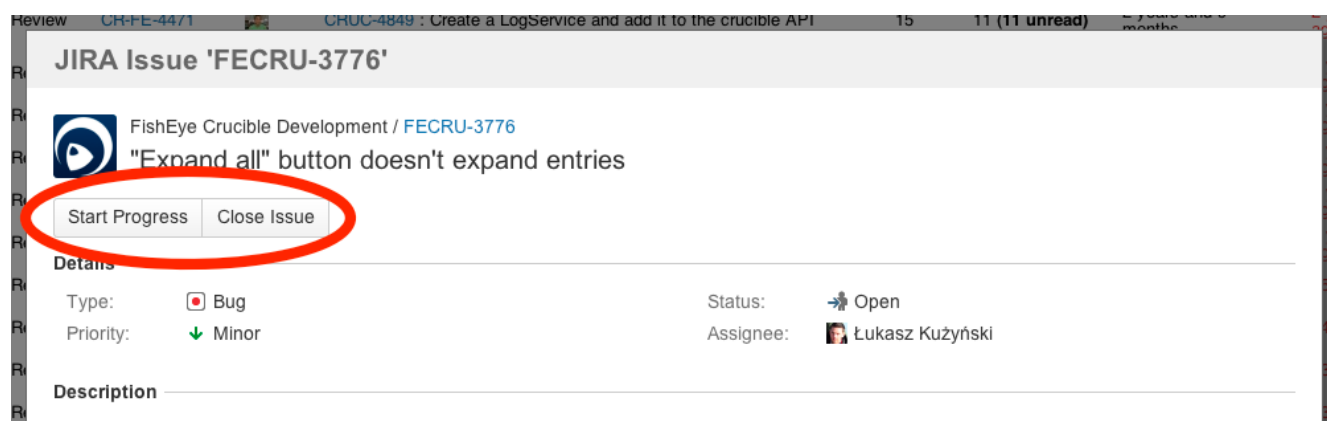
- [Transitioning any JIRA issue at any time](#)
- [Transitioning a linked JIRA issue when closing the review](#)

Related pages:

- [JIRA integration in Crucible](#)
- [Creating JIRA issues from the review](#)
- [Linking Crucible to JIRA](#)

Transitioning any JIRA issue at any time

You can easily [transition](#) a JIRA issue at any time from within Crucible. Click on a JIRA issue link anywhere in Crucible to see a dialog with the available workflow steps:



Click on a step in the dialog, and complete any displayed fields as required. If there are custom required fields that are unsupported by Crucible, just click **Edit this field in JIRA** to transition the issue directly in JIRA.

Transitioning a linked JIRA issue when closing the review

When closing a Crucible review you may also want to close a JIRA issue that is [linked to that review](#).

In the Review Summary screen, click **Close** near the top right. In the 'Closed' dialog, the available workflow transitions for the linked JIRA issue appear in the **Transition issue** dropdown:

CR-FE-6163 Closed

Linked issue

Issue: **FECRU-1401** – Add a review action in the UI to send manual reminder...

Status: To be reviewed

Transition issue: **Close**

Log work: Submit 43m

Choose a step from the dropdown, and click **Close**.

Notes

- Only the transitions accessible by the user are displayed.
- The list of available transitions only appears if the user has visibility to any available workflow transitions.
- Crucible administrators can turn off JIRA issue transitioning by disabling the Crucible Issue Transitioning Plugin. See [Managing add-ons](#).

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

Administering Crucible

The Admin area allows you to administer your Crucible instance and to manage your repositories, users and back-end settings.

Once Crucible is [installed](#) and running, you can log in to the Admin area by either:

- logging in with an administrator's account.
- clicking **Administration** at the foot of the page.
- navigating to `http://HOSTNAME:8060/admin/`, where HOSTNAME is the name of the server on which you installed Crucible.

Once logged in as an [administrator](#) you can get to the Admin area by clicking the 'cog' menu in the FishEye/Crucible header, and choosing **Administration**.

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

Topics

- [Best practices for Crucible configuration](#)
- [Crucible and FishEye](#)
- [Administering projects](#)
- [Configuring repositories](#)
- [Setting up users and security](#)
- [Permissions](#)
- [Migrating to an external database](#)
- [Backing up and restoring Crucible data](#)
- [Customising Crucible](#)
- [Linking Crucible to JIRA](#)
- [Linking to another application](#)
- [Running Crucible as a Windows service](#)
- [Managing add-ons](#)

Best practices for Crucible configuration

1. Set up a separate `FISHEYE_INST` folder location on the same system for Crucible's data.

This will allow for easy upgrades of the core program and neatly separated data backup.

2. Run Crucible on a dedicated machine, accessing its data on the local file system.

This is the best environment for swift Crucible performance. Avoid running Crucible in a virtual environment.

3. Do not give Crucible projects the same key as your JIRA projects.

When naming projects, take care to ensure that the key you assign to them is not the same as any of your JIRA projects. The reason for this is, if one of your Crucible projects has the same key as one of your projects in JIRA, then all links with that key will lead back to Crucible, rather than leading to JIRA, removing the ability to navigate between the two applications.

To avoid this, name your Crucible project keys differently. For example, you could place the following text at the beginning of each project key: CR- to distinguish it. So, for this case, if you have an existing JIRA key of 'RHUBARB', you would create a Crucible key called 'CR-RHUBARB' so that they do not conflict.

4. Do not use the built-in HSQLDB database for production use.

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.

To see a list of external databases that Crucible supports, see the [Supported platforms](#) page. For information on how to set up an external database, see the [Crucible Database documentation](#).

Crucible and FishEye

This page gives an overview of the joint installation of [Crucible](#) and [FishEye](#). Both Crucible and FishEye are [Atlassian](#) products.

- **FishEye** allows you to extract information from your source code repository and display it in sophisticated reports.
- **Crucible** allows you to request, perform and manage code reviews.
- Both of these products can run in isolation. If you are using Subversion, Git, Mercurial, CVS or Perforce you can significantly enhance your Crucible experience by also using FishEye.
 - See [What happens if I decide to stop using FishEye with Crucible?](#)

Your Crucible installation package includes the files required for FishEye

If you use FishEye and Crucible together, they run as one instance.

Purchasing and installing Crucible and FishEye

- Upgrading an existing Crucible installation to also use FishEye only requires a simple licence change in the admin area.
- Upgrading an existing FishEye installation to also use Crucible only requires a simple licence change in the admin area.
- See [Upgrading from FishEye to Crucible](#).

The <Crucible home directory> and FISHEYE_INST

Throughout the Crucible documentation, references are made to the <Crucible home directory>, which refers to the location of the Crucible application. Be aware that, when Crucible is run with FishEye, this location is equivalent to the location referred to by <FishEye home directory> in the FishEye documentation.

Crucible also makes use of this FishEye environment variable:

- FISHEYE_INST – the location of the FishEye data.

Refer to the FishEye documentation for more about the [environment variables](#).

Installing the Crucible binary files

See the following for Crucible install instructions:

- [Installing Crucible on Windows](#)
- [Installing Crucible on Linux and Mac](#)

Setting up a repository for use with stand-alone Crucible

For complete instructions, see [Configuring repositories](#).

Setting up a repository for use with FishEye and Crucible

If you intend to use Crucible and FishEye with:

- [Subversion](#), please read [Supported platforms](#), [Subversion client setup](#), and [granting permission to FishEye](#) to scan your repository.
- [Git](#), please read [Supported platforms](#) and [Git Client setup](#).
- [Perforce](#), please read [Supported platforms](#) and [Perforce Client setup](#).
- [CVS](#), please read [Supported platforms](#) and [CVS Client setup](#).
- [Mercurial](#), please read [Supported platforms](#) and [Mercurial Client setup](#).

Read more

You can find more information in:

- Crucible [Getting Started](#)
- FishEye [Getting started](#)

Administering projects

A Crucible *project* provides a way to group and manage related [reviews](#) – typically reviews that are all involved with the same software project. A Crucible project allows you to

- define default [moderators](#), [authors](#) and [reviewers](#) for the reviews in that project.
- define which people are eligible to be [reviewers](#) for the reviews in that project.
- use [permission schemes](#) to restrict who can perform particular actions (e.g. 'Create Review') in that project.

Every Crucible review belongs to a project. Each project has a *name* (e.g. ACME Development) and a *key* (e.g. ACME). The project key becomes the first part of that project's *review keys*, e.g. ACME-101, ACME-102, etc:

By default, Crucible contains one project. This default project has the key 'CR' and the name 'Default Project'.

When administering your Crucible projects from the projects listing in the [admin area](#), you can:

- Click **Add a new project** (at the bottom of the list) to [create a new project](#).
- Click **Edit** (in the 'Crucible Settings' column) for an existing project to [edit its settings](#).
- Click **Delete** for an existing project to delete it. See [Deleting a project](#) below.

Deleting a project

Before you begin:

- By default, Crucible contains one project; it has the key 'CR' and the name 'Default Project'. This project cannot be deleted.
- Deleted projects cannot be recovered.

To delete a Crucible project:

1. Go to the [admin area](#) and click **Projects** (under 'Project Settings').
2. Click **Delete** (in the 'Crucible Settings' column) for the project you wish to remove.
3. If empty, the project instantly disappears.

If the project contains reviews, you will be prompted to either delete all the reviews in the project, or move them into the default project.

Creating a project

A Crucible *project* provides a way to group and manage related [reviews](#) – typically reviews that are all involved with the same software project. A Crucible project allows you to

- define default [moderators](#), [authors](#) and [reviewers](#) for the reviews in that project.
- define which people are eligible to be [reviewers](#) for the reviews in that project.
- use [permission schemes](#) to restrict who can perform particular actions (e.g. 'Create Review') in that project.

Every Crucible review belongs to a project. Each project has a *name* (e.g. ACME Development) and a *key* (e.g. ACME). The project key becomes the first part of that project's *review keys*, e.g. ACME-101, ACME-102, etc:

By default, Crucible contains one project. This default project has the key 'CR' and the name 'Default Project'.

To create a new project:

1. Click the 'cog' menu in the Crucible header, and choose **Administration**. You need to be logged in as an [administrator](#) to see this.
2. Click **Projects** (under 'Project Settings').
3. Click **Add a new project** (at the bottom of the list).
4. Complete the fields in the 'Edit Project' page (described below).
5. Click **Save** to create your new project.

Identification	• Name – the plain language name as displayed in the Crucible interface. • Key – the project key used when giving reviews their unique code names. If you change the key for a project all open, and closed, reviews in the project are updated with the new key.
Content	• Default Repository – the repository that contains the source code for this project. This is the repository that will be searched by default when you add files to a review. • Store the contents of files in reviews – check to have the source files under review stored in the Crucible database, along with the comments and review data. This retains a copy of all the source files under review even if the repository is disconnected from Crucible. See Storing all revisions under review.
Permission Scheme	Permission Scheme – the permission scheme applied to this project. (A permission scheme controls who can perform particular actions, for example, create a review.)
Moderator	• Enable the Moderator role – clear to have reviews run by the author only. See Enabling the moderator role. • Default Moderator – the user who will be set as the moderator for all new reviews created in the project. Leave this field blank to force the review's creator to choose a moderator. See Creating a review.
Default Reviewers	• By default, allow anyone to join reviews after creation – sets the default state of the Allow anyone to join checkbox on the 'Choosing reviewers' screen. • Users – set the individual users to whom new reviews will be assigned by default. • Groups – set the groups to which new reviews will be assigned by default.
Allowed Review Participants	Use Users and Groups to restrict who can have a role (author/creator/moderator/reviewer) in this project's reviews. These users will be the only ones whose names appear when a review is assigned . Leave these fields blank to allow all users to have access.
Review Duration	Default duration – the default length of time (in week days) for reviews in this project. If you leave the field blank, then no time restriction is applied. Reviews that are overdue will show up in red on the reviewer's dashboards. Note that the review duration only affects the default due date that appears when creating a review . The review's creator or moderator can specify a different date if they wish.

Default
Review
Objectives

Default Objectives – specify some text that will appear by default in the **Review Objectives** field of each new review. This text can be edited, as with any text in the Review Objectives text box. See [Setting default review objectives](#).

Screenshot: The Edit Project Screen

Edit Project ⓘ

Identification

Name:

Key:

Default Review Content Repository

Default Repository:

☒ Store the contents of files in reviews

Project Permissions Scheme ⓘ

Permission Scheme:

Moderator ⓘ

☒ Enable the Moderator role for this project

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

Default Reviewers ⓘ

☐ By default, allow anyone to join reviews after creation

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

Groups: Start typing a group name then press enter to select.

Allowed Review Participants ⓘ (Leave blank to let all users access this project)

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

Groups: Start typing a group name then press enter to select.

Review Duration ⓘ

Default duration in week days:

Default Review Objectives ⓘ

Default Review Objectives:

Storing all revisions under review

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

i Note that:

- The storage of files must be set per-project. Also, the storage only applies to reviews created after Revision Storage is enabled. This means that for existing projects, pre-existing reviews will not have files stored unless you look at them again after Revision Storage is enabled.
- The underlying repository and file permissions are not preserved when files are stored with the review.

Enabling file storage

You can enable file revision storage on the 'Edit Project' page.

For an existing project:

1. Go to the [admin area](#) and click **Projects** (under 'Project Settings').
2. Click **Edit** (in the 'Crucible Settings' column) for an existing project.
3. Check **Store the contents of files in reviews** (under 'Content').
4. Click **Save**.

Enabling the moderator role

By default, Crucible projects do not have a moderator. This allows for a streamlined review handling process, where the review [author](#) is the sole person who starts and stops the review. However, you can enable the moderator role for Crucible projects, if required.

The moderator role can only be set by a Crucible administrator.

On this page:

- [Enabling the moderator role](#)
- [Removing the moderator role from an existing project](#)
- [Adding the moderator role to an existing project](#)

Enabling the moderator role

The [moderator](#) role is configurable in Crucible as a per-project setting. By default, all reviews have an [author](#) and a [moderator](#). However, the moderator role can be disabled.

To enable or disable the moderator role on a project:

1. Go to the [admin area](#) in Crucible and click **Projects** (under 'Project Settings').
2. Click **Edit** (in the 'Crucible Settings' column) for the project.
3. Check **Enable the Moderator role** (under 'Moderator').

Removing the moderator role from an existing project

When you remove the moderator role from an existing project, note that:

- Existing reviews, created before the change, will still retain the moderator they were assigned.
- Reviews created after the change will not have the moderator role.
- If the removal of the moderator conflicts with other Crucible project settings, a warning will be shown on the Projects page.

 If in doubt about the impact of removing the moderator role, you can create a new project, and set the moderator status while doing that.

Adding the moderator role to an existing project

If you add the moderator role to an existing project, note that:

- Existing reviews, created before the change, will still have no moderator.
- Reviews created after the change will have the moderator role added.
- If the addition of the moderator role conflicts with other Crucible project settings, a warning will be shown on the Project page.

 If in doubt about the impact of adding the moderator role, you can create a new project, and set the moderator status while doing that.

Setting default review objectives

To set default review objectives for all the reviews in a given project:

1. Go to the [admin area](#) and click **Projects** (under 'Project Settings').
2. Click **Edit** (in the 'Crucible Settings' column) for the project.
3. In the **Default Objectives** text box specify the text that will appear by default in the **Review Objectives** fi

eld of each new review. This text will be able to be edited, as with any text in the Review Objectives text box.

4. Click **Save**.

See [Editing a project](#) for more information about project defaults.

Screenshot: Default Review Objectives in Crucible

Editing a project

Once a project is created, an administrator can edit the default values for settings such as repository, moderator, allowed reviewers, allowed groups and permissions scheme. These are the settings that are applied to any new review created for the project.

To edit project settings:

1. In the admin area, click **Projects** (under 'Project Settings').
2. In the list of projects, click **Edit** in the ('Crucible' column) for the required project.
3. Modify the available settings, as required:

Identification	• Name – the plain language name as displayed in the Crucible interface. • Key – the project key used when giving reviews their unique code names. If you change the key for a project all open, and closed, reviews in the project are updated with the new key.
Content	• Default Repository – the repository that contains the source code for this project. This is the repository that will be searched by default when you add files to a review. • Store the contents of files in reviews – check to have the source files under review stored in the Crucible database, along with the comments and review data. This retains a copy of all the source files under review even if the repository is disconnected from Crucible. See Storing all revisions under review.
Permission Scheme	Permission Scheme – the permission scheme applied to this project. (A permission scheme controls who can perform particular actions, for example, create a review.)
Moderator	• Enable the Moderator role – clear to have reviews run by the author only. See Enabling the moderator role. • Default Moderator – the user who will be set as the moderator for all new reviews created in the project. Leave this field blank to force the review's creator to choose a moderator. See Creating a review.
Default Reviewers	• By default, allow anyone to join reviews after creation – sets the default state of the Allow anyone to join checkbox on the 'Choosing reviewers' screen. • Users – set the individual users to whom new reviews will be assigned by default. • Groups – set the groups to which new reviews will be assigned by default.
Allowed Review Participants	Use Users and Groups to restrict who can have a role (author/creator/moderator/reviewer) in this project's reviews. These users will be the only ones whose names appear when a review is assigned . Leave these fields blank to allow all users to have access.

Review Duration	Default duration – the default length of time (in week days) for reviews in this project. If you leave the field blank, then no time restriction is applied. Reviews that are overdue will show up in red on the reviewer's dashboards. Note that the review duration only affects the default due date that appears when creating a review . The review's creator or moderator can specify a different date if they wish.
Default Review Objectives	Default Objectives – specify some text that will appear by default in the Review Objectives field of each new review. This text can be edited, as with any text in the Review Objectives text box. See Setting default review objectives .

4. Add additional FishEye repositories to the project if required – click **Edit** in the 'FishEye content' column. See [How to add several repositories to one project in Crucible](#) for further information.

Screenshot: The Edit Project screen in Crucible

Admin

- Repository Settings
 - Repositories
 - Defaults
 - Smart Commits
 - Web Hooks
- Project Settings
 - Projects
- User Settings
 - Users
 - User Mappings
 - Avatars
- Security Settings
 - Change Admin Password
 - Authentication
 - Administrators
 - Groups
 - Permission Schemes
- Global Settings
 - Server
 - ViewCVS URL Mappings
 - Crucible
 - Front Page Customization
 - Application Links
- Systems Settings
 - Database
 - System Info
 - Content
 - Backup
 - Shutdown
 - Plugin Manager
 - Plugin Marketplace
 - Plugin Audit Log
 - FishEye Update Check
 - Atlassian Support Tools

Edit Project

Identification

Name

Key

Content

Default Repository

☒ Store the contents of files in reviews

Permissions Scheme

Permission Scheme

Moderator

☒ Enable the Moderator role for this project

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

Default Reviewers

☐ By default, allow anyone to join reviews after creation

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

Groups Start typing a group name then press enter to select.

Allowed Review Participants (Leave blank to let all users access this project)

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

Groups Start typing a group name then press enter to select.

Review Duration

Default duration

Default duration in week days

Default Review Objectives

Default Objectives

Configuring repositories

Crucible needs access to your source code before you can create any reviews. This page describes how you connect Crucible to a repository.

By the way, if you want to connect to a Git repository you'll need to have Git installed on the Crucible server. See [Installing and upgrading Git](#).

The instructions for configuring your repositories differ, *depending on your Crucible setup*:

If you are running Crucible with Atlassian's FishEye, you can manage your repositories using FishEye. See the [FishEye documentation](#) for more information.

If you are running Crucible alone, you can manage your repositories using native repository access in Crucible:

1. Go to the Crucible [admin area](#).
2. Click **Repositories** (under 'Repository Settings'), and then **Native Repository Access** (if necessary).
3. Click **Add Existing...**
4. Refer to the following pages in the [FishEye documentation](#) for details about setting up particular repositories:
 - [Stash](#)
 - [CVS](#)
 - [Git](#)
 - [Mercurial](#)
 - [Perforce](#)
 - [Subversion](#)

Please also see the [What happens if I decide to stop using FishEye with Crucible](#) page for important information about light FishEye.

Crucible SCM plugins superseded by Native Repository Access

Crucible now ships with native repository access, which allows you to connect to repositories without a working version of FishEye. Crucible SCM plugins will still work, but we recommend that you stop using them in favour of native repository access. See [What happens if I decide to stop using FishEye with Crucible](#) for instructions.

Setting up a Git repository in Crucible

This page describes how to use Crucible's native repository access to connect to a Git repository. The process for doing this depends on where the Git repository is hosted – connecting to a Git repo that is hosted in [Atlassian Stash](#) is much easier.

This native access uses a FishEye component (without requiring a FishEye license), which is why the description below refers to FishEye in places. See [What happens if I decide to stop using FishEye with Crucible](#) for more information.

Crucible interacts with Git repositories by executing the Git command in a separate process. Hence, the server running Crucible needs to have Git installed. Crucible indexes Git repositories by making a private, bare clone of your repository within Crucible's cache area. It uses this private clone for most Git operations. See [Git manifest](#) for more information about indexing Git repositories.

 See the [Supported platforms](#) page for the version of Git that is required by the server running Crucible.

Git repositories hosted in Stash

When [Atlassian Stash](#) is integrated with FishEye:

- You can easily add Git repositories to FishEye. The repository behaves just like a native repository in FishEye, so your team gets all the benefits of FishEye indexing, browsing and searching.
- The repository becomes available to Crucible (when integrated), so you can perform in-depth code reviews for changes in the repository.
- When you add a Stash repository to FishEye, a push to the repository will by default automatically trigger FishEye to run an incremental index. No further configuration is required – you don't have to configure polling for new commits, or set up dedicated FishEye web hooks in your Stash instance.

You'll need to have an account in the Stash instance, as well as permission to view the repository that you want to add.

1. Click the 'cog' menu in the FishEye header, and choose **Administration** (you'll need to be logged in as an [administrator](#) to see this link).
2. Click **Repositories** (under 'Repository Settings').
3. Click the **Stash repositories** tab, and authenticate with the Stash server if necessary.
4. If multiple instances of Stash are [connected](#) to FishEye, use the **Stash server** list to choose the instance of Stash that hosts the repository you wish to add.
5. Optionally, type a filter pattern to restrict the list of displayed repositories to those with a matching name, key or project.
6. Click **Add** for each repository that you wish to add to FishEye.

When you add a Stash repository:

- FishEye creates a read-only SSH key and adds that as an [access key](#) to the repository in Stash. If this operation fails, the key will be added as a user key to your profile in Stash. You can check if an access key was added by viewing the repository's settings in Stash. See [Using SSH keys to secure Git operations](#) in the Stash documentation for more information.

Git repositories hosted elsewhere

When adding or managing a Git repository, you can do the following:

- Define repository details, as described below.
- Set FishEye's [repository options](#).

To add an external Git repository to FishEye:

1. Click the 'cog' menu in the FishEye header, and choose **Administration** (you'll need to be logged in as an [administrator](#) to see this link).
2. Click **Repositories** (under 'Repository Settings').
3. Click **Native repository access**, and then **Add repository**.
4. Complete the wizard:

Step 1

Field	Description
Repository Type	Select Git .
Name	Enter a name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters. This will be repository name in FishEye.
Description	Enter a short description of this repository.

Step 2

Field	Description
Repository Location	Enter the URL describing the Git repository location. FishEye will clone this repository for indexing purposes. You can use any URL recognised by Git itself. FishEye recognizes the following URL's: git://server_name[:port]/path_to_repository http://server_name[:port]/path_to_repository https://server_name[:port]/path_to_repository ssh://server_name[:port]/path_to_repository file://[hostname]/path_to_repository Do not use spaces in your URL.
Path	(optional) Enter the path within the Git repository that you want FishEye to index. This lets you limit FishEye to indexing a subset of the complete Git repository.

Block Size	(<i>optional</i>) Enter how many commits you want FishEye to process in one batch. Larger values require more memory and increase the amount of work FishEye commits to the database in a single operation. The default is 400. The minimum being 1. This field only accepts positive whole numbers. Requires a repository restart.
Command Timeout	(<i>optional</i>) Enter the time that a single Git command is allowed to take to execute. Any command that exceeds this time is terminated and the operation will fail.
Rename Detection	(<i>optional</i>) Select which Git rename detection strategy FishEye will use to detect copy and move operations within the repository. Please refer to the Git documentation for more information .

Step 3

Field	Description
Store Diff Info	Tick this checkbox, if you want FishEye to cache information about file diffs in its database. This is required for some FishEye features. See Configuring Repository Details for more information on this setting.
Enable Repository After Adding	Tick this checkbox, to enable the repository after adding (i.e. when you click the 'Add' button).

Limitations

Indexing of a Git repository can be slow when new branches are pushed. Performance in such cases can be improved by setting a command line option for FishEye. See this [KB page](#) for more information.

Setting up a Perforce repository in Crucible alone

This page describes how to configure Crucible access to Perforce repositories.

Crucible SCM plugins superseded by Native Repository Access

Crucible now ships with native repository access, which allows you to connect to repositories without having FishEye integrated with your Crucible instance. Crucible SCM plugins will still work, but we recommend that you stop using them in favour of native repository access. See [What happens if I decide to stop using FishEye with Crucible](#) for instructions.

Setting up a Perforce repository using native access

To set up native access to a Perforce repository:

1. Go to the Crucible [admin area](#).
2. Click **Repositories** (under 'Repository Settings'), and then **Native Repository Access** (if necessary).
3. Click **Add Existing...**
4. Refer to [Perforce](#) in the FishEye documentation for detailed information about completing the wizard.

Setting up the Crucible Perforce SCM plugin

This section describes how to configure the [Crucible Perforce SCM plugin](#) to access Perforce repositories.

To set up Perforce in Crucible alone:

1. Ensure that the Perforce executable file is on the system path, in the Crucible server's [Environment Variables](#)
2. Start Crucible and go to the [admin area](#).
3. Click **Manage Add-ons** (under 'Systems Settings').
4. Find and click on the **Crucible Perforce SCM Plugin** (click **Show System Plugins**) and then click **Confi**

gure.

- Click **Add a repository** and complete the form:

Name	Choose a unique name for the repository.
Repository Server	Enter the base URL and port for the repository, for example: <code>example.com:666</code> .
Repository Path	Add the path to your Perforce repository. For example: <code>//depot/code/example/main</code> .
Perforce Username	Enter the username of the Perforce account that Crucible will use. (optional)  Note that this account should only have read-only access to the repository.
Perforce Password	Enter the password of the Perforce account that Crucible will use. (optional)

- Click **Save**.

Your Perforce repository is now set up for Crucible. You will be able to select changesets from it when creating reviews.

Notes

- There is no 'initial scanning' required in this process, as Crucible's access to Perforce (when running alone) is strictly on-demand. Data is not indexed, hence there is no scanning.
- Crucible executes the Perforce command-line tool to enable this functionality.

Setting up a Subversion repository in Crucible alone

This page describes how to configure Crucible access to Subversion repositories.

Setting up a Subversion repository using native access

To set up native access to a Subversion repository:

- Go to the Crucible [admin area](#).
- Click **Repositories** (under 'Repository Settings'), and then **Native Repository Access** (if necessary).
- Click **Add Existing...**
- Refer to [Subversion](#) in the FishEye documentation for detailed information about completing the wizard.

Setting up the Crucible Subversion SCM plugin

This section describes how to configure the Crucible Subversion SCM plugin to access Subversion repositories. The plugin is bundled with Crucible.

To set up Subversion in Crucible alone:

- Start Crucible and go to the [admin area](#).
- Click **Manage Add-ons** (under 'Systems Settings').
- Find and click on **Crucible Subversion SCM Plugin** (click **Show System Add-ons**) and then click **Configure**.
- Click **Add a repository** and complete the form:

Name	Choose a unique name for the repository.
Repository Root	Enter the repository root URL for the repository. See Finding your repository root below for instructions on how to find this.

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

5. Click **Save**.

Your Subversion repository is now set up for Crucible. You will be able to select changesets from it when creating reviews.

Note that there is no 'initial scanning' required in this process, as Crucible's access to Subversion (when running alone) is strictly on-demand. Data is not cached, hence scanning is not required.

Finding your repository root

Run the following command:

```
svn info SVN_URL
```

where SVN_URL is the complete URL of the repository you want to add.

You will get something like the following:

```
>svn info http://svn.example.com/svn5/
Path: svn5
URL: http://svn.example.com/svn5/
Repository Root: http://svn.example.com/
Repository UUID: ce062a09-193b-427a-a7b3-a85007076e5d
Revision: 83
Node Kind: directory
Last Changed Author: ryan
Last Changed Rev: 83
Last Changed Date: 2009-05-07 10:48:41 +1000 (Thu, 07 May 2009)
```

Next to "Repository Root" is the URL you should define as your repository root. The path will be whatever is remaining.

Enabling reviews from the server file system in Crucible

You can set up the server file system to be a code repository for Crucible. You will be able to browse files and directories on the hard drive and select files from it when creating reviews.

To set up the file system as a code repository in stand-alone Crucible:

1. Go to the [admin area](#) in Crucible.
2. Click **Manage Add-ons** (under 'Systems Settings').
3. Find and click on **Crucible FileSystem SCM Plugin** (click **Show System Plugins**, then click **Enable** if that is displayed).
4. Click **Configure** then **Add a repository**.
5. Complete the form:

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

6. Click **Save**.

Setting up a repository via FishEye

To use FishEye to access an external source control repository, such as Subversion or Git, for Crucible, see the FishEye documentation for how to [add a repository](#).

- [CVS](#)
- [Git](#)
- [Mercurial](#)
- [Perforce](#)
- [Subversion](#)

Building index and cache

FishEye needs to build an index and cache of the contents of your repository, so some information will not appear in FishEye until this is complete. This may take some time to complete, depending on the size of the repositories.

 We recommend you access the repository with a user that has only **read** access to the repository.

Configuring commit hooks

The [incremental indexing](#) process causes Crucible to poll all repositories at the specified interval to check for new commits, even though there might not be any new information to index. If you have a large number of repositories (> 100), this can lead to:

- A time lag between a commit being made and it appearing in Crucible.
- A high load on the Crucible server, and on the SCM.

Commit hooks allow you to set up your SCM so that indexing of a repository is triggered by a commit to the repository itself. This means that Crucible only runs the indexing process when necessary, and allows automatic [polling](#) to be disabled. Commits will appear sooner in Crucible, and the server load will be reduced.

To set up commit hooks you:

1. Set the REST API token in FishEye or Crucible. See the FishEye-Crucible [REST API](#) developer documentation.
2. Integrate the commit hook with your SCM.

On this page:

- [Triggering scanning remotely](#)
- [Integrating with your SCM](#)
 - [Bitbucket and GitHub](#)
 - [CVS](#)
 - [SVN](#)
 - [Perforce](#)
 - [Git](#)
 - [Mercurial](#)
- [Decreasing polling frequency](#)

Triggering scanning remotely

Once you've set your REST API token you can use it to trigger scanning when your repository is updated.

The basic way to do this is set up a shell script similar to:

```
echo Triggerring scan
/usr/bin/curl -X POST -H "X-Api-Key: <YOUR-API-KEY-HERE>" <URL>:<PORT></optional
CONTEXT>/rest-service-fecru/admin/repositories-v1/<REPOSITORY-NAME>/scan
```

e.g.

```
echo Triggerring scan
/usr/bin/curl -X POST -H "X-Api-Key: abcdefg123456"
http://atlas:8060/fecru/rest-service-fecru/admin/repositories-v1/widget/scan
```

Try running the script; if everything is fine, it will output "[]", and will trigger scanning in Crucible. If there are problems, curl will show an appropriate message.

If you're running on Windows, you'll need curl or a similar program. You can download the Windows version of curl [here](#). You'll need to save the script as a batch file (with the .bat extension).

Note: be sure to specify the full path to the curl binary on your system.

Integrating with your SCM

Bitbucket and GitHub

Both of these hosting services provide service hooks that can be used to trigger repository indexing in Crucible.

Bitbucket

In Bitbucket, go to the admin page for your repository, click **Hooks** and choose **FishEye**.

See the [Bitbucket documentation](#) for more information about setting up a Bitbucket service hook.

GitHub

In GitHub, go to the admin page for your repository, click **Service Hooks** and choose **FishEye** from the available hooks.

CVS

1. Checkout the CVSR00T module of your cvs repository:

```
cvs co CVSR00T
```

2. Edit the CVSR00T/loginfo file.
3. Add the following line to the file:

```
ALL /usr/bin/curl -X POST -H "X-Api-Key: <YOUR-API-KEY-HERE>" -m 20
<URL>:<PORT></optional
CONTEXT>/rest-service-fecru/admin/repositories-v1/<CVS-REPOSITORY-NAME>/scan >
/dev/null 2>&1 &
```

e.g.

```
ALL /usr/bin/curl -X POST -H "X-Api-Key: abcdefg123456"
http://atlas:8060/fecru/rest-service-fecru/admin/repositories-v1/cvs_widget/sc
an > /dev/null 2>&1 &
```

4. Commit your changes:

```
cvs commit CVSR00T/loginfo
```

SVN

1. Log into your svn server, go to the repository directory, find the hooks subdirectory there:

```
cd /var/www/svn/hooks
```

2. If it doesn't exist, create a new file called post-commit (or post-commit.bat on Windows), make sure it's executable by the user that the svn process runs as:

```
touch ./post-commit  
chmod 755 ./post-commit
```

3. Make sure the file starts with the following shebang line, pointing to your shell:

```
#!/bin/sh
```

4. Add the following to the post-commit file:

```
/usr/bin/curl -X POST -H "X-API-Key: <YOUR-API-KEY-HERE>" -m 20  
<URL>:<PORT></optional  
CONTEXT>/rest-service-fecru/admin/repositories-v1/<SVN-REPOSITORY-NAME>/scan >  
/dev/null 2>&1 &
```

e.g.

```
/usr/bin/curl -X POST -H "X-API-Key: abcdefg123456"  
http://atlas:8060/fecru/rest-service-fecru/admin/repositories-v1/svn_widget/sc  
an > /dev/null 2>&1 &
```

You can find more details about svn hooks [here](#).

Perforce

1. As a Perforce administrator execute the following command:

```
p4 triggers
```

2. The trigger table form will be presented.

3. Add a field value for the field 'Triggers', named trigger-X, where X is the next number available for the trigger:

```
trigger-04 change-commit //... "/usr/bin/curl -s -o /dev/null -X POST -H
X-API-Key:<YOUR-API-KEY-HERE> -m 20 <URL>:<PORT></optional
CONTEXT>/rest-service-fecru/admin/repositories-v1/<PERFORCE-REPOSITORY-NAME>/s
can
```

e.g.

```
trigger-04 change-commit //... "/usr/bin/curl -s -o /dev/null -X POST -H
"X-API-Key: abcdefg123456"
http://atlas:8060/fecru/rest-service-fecru/admin/repositories-v1/perforce_widg
et/scan
```

4. You can customize the trigger to run only for a specific depot or directory, by replacing the `//...` above (which causes the trigger to be executed for every file) by a standard Perforce file pattern syntax.

You can find more details about Perforce triggers in the [Perforce System Administrator's guide](#).

Git

1. Choose the repository you want to trigger the scans from. Usually this is the repository that all of your developers push to, and that you run CI from. Note that hooks are not propagated when cloning repositories.
2. Go to the hooks subdirectory of your repository:

```
cd /var/www/git/project/hooks
```

3. If it doesn't exist, create a new file called `post-receive`. Make sure it's executable by the Git server process.

```
touch ./post-receive
chmod 755 ./post-receive
```

4. Make sure the file starts with the following line, pointing to your shell:

```
#!/bin/sh
```

5. Add the following to the `post-receive` file:

```
/usr/bin/curl -X POST -H "X-API-Key: <YOUR-API-KEY-HERE>" -m 20
<URL>:<PORT></optional
CONTEXT>/rest-service-fecru/admin/repositories-v1/<GIT-REPOSITORY-NAME>/scan >
/dev/null 2>&1 &
```

e.g.

```
/usr/bin/curl -X POST -H "X-API-Key: abcdefg123456"
http://atlas:8060/fecru/rest-service-fecru/admin/repositories-v1/git_widget/sc
an > /dev/null 2>&1 &
```

NOTE: Not all methods of serving a Git repository support commit hooks - if serving over http, you need to use

smart-http (either using [git-httpd-backend](#) or a dedicated repository manager like gitolite). You can find more information about smart http [here](#). Serving the repository over ssh or git-daemon should allow you to run commit hooks as well.

Mercurial

1. Choose the repository you want to trigger the scans from. Usually this is the repository that all of your developers push to, and that you run CI from. Note that hooks are not propagated when cloning repositories.
2. Go to the .hg subdirectory of your repository:

```
cd /var/www/hg/project/.hg
```

3. If it doesn't exist create a file named hgrc :

```
touch ./hgrc
```

4. Add the following to the hgrc file:

```
[hooks]
changegroup = /usr/bin/curl -X POST -H "X-API-Key: <YOUR-API-KEY-HERE>" -m 20
<URL>:<PORT></optional
CONTEXT>/rest-service-fecru/admin/repositories-v1/<HG-REPOSITORY-NAME>/scan >
/dev/null 2>&1 &
```

e.g.

```
[hooks]
changegroup = /usr/bin/curl -X POST -H "X-API-Key: abcdefg123456"
http://atlas:8060/fecru/rest-service-fecru/admin/repositories-v1/hg_widget/sca
n > /dev/null 2>&1 &
```

NOTE: Not all methods of serving a Mercurial repository support commit hooks - if serving over http, you can't use static-http serving.

Decreasing polling frequency

Once your commit hook is set up and successfully notifying Crucible about new commits to your repository, you can decrease the polling frequency on your repository (for example to 1 or 2 hours, instead of the default 1 minute).

The screenshot shows the 'Repository "Ada-BB"' configuration page. On the left is a sidebar with links: Summary, Maintenance, SCM Details, Updates (selected), Linkers, Permissions, and Include/Exclude Paths. The main content area is titled 'Updates' and contains the text: 'Settings which affect how often FishEye and Crucible will check if there are new commits to the repository. need to define a custom updater. 🟢'. Below this, there is a checkbox 'Use the system default settings for updates' which is unchecked. Underneath, there is a 'Disable polling:' checkbox which is checked. At the bottom, the 'Polling Interval:' is set to '1 hour' in a text input field, with a help icon to its right.

With commit hooks configured, scheduled polling is only useful if the hook fails, for example because of connectivity issues to the server hosting Crucible. This will decrease the server load, but allow Crucible to still occasionally check for changes, and update the repository if needed. Note that after changing the polling

frequency, Crucible will need to be restarted.

```
trigger-04 change-commit //... "/usr/bin/curl -s -o /dev/null -X POST -H
X-Api-Key:YOUR-API-KEY-HERE -m 20
http://SERVER:PORT/CONTEXT/rest-service-fecru/admin/repositories-v1/PERFORCE-REPOSI
TORY-NAME/scan
```

Setting up users and security

User management and security settings are covered in the FishEye documentation. See:

- [Managing users and groups in FishEye](#)
- [Configuring FishEye security](#)

Other related security resources

- [Configuring User Managed Mappings](#)
- [Creating a permission scheme](#)
- [Creating a user](#)

Creating a user

Some user management functions are identical in both Crucible and FishEye. See more [user management documentation](#) in the [FishEye documentation](#).

There are two types of user accounts:

- 'Built-in' user accounts — these are stored in the local database for FishEye/Crucible.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if configured. See [Configuring external authentication sources](#).

Note regarding external directories:

- New users can only be added if they already exist in the external directory. Your external directory will not be modified.
- If you have enabled 'auto-add' for your [external directory](#), users who don't exist locally in FishEye/Crucible will be automatically added the first time they log in to Crucible.
- You can't use JIRA Cloud to manage your Crucible or FishEye users.

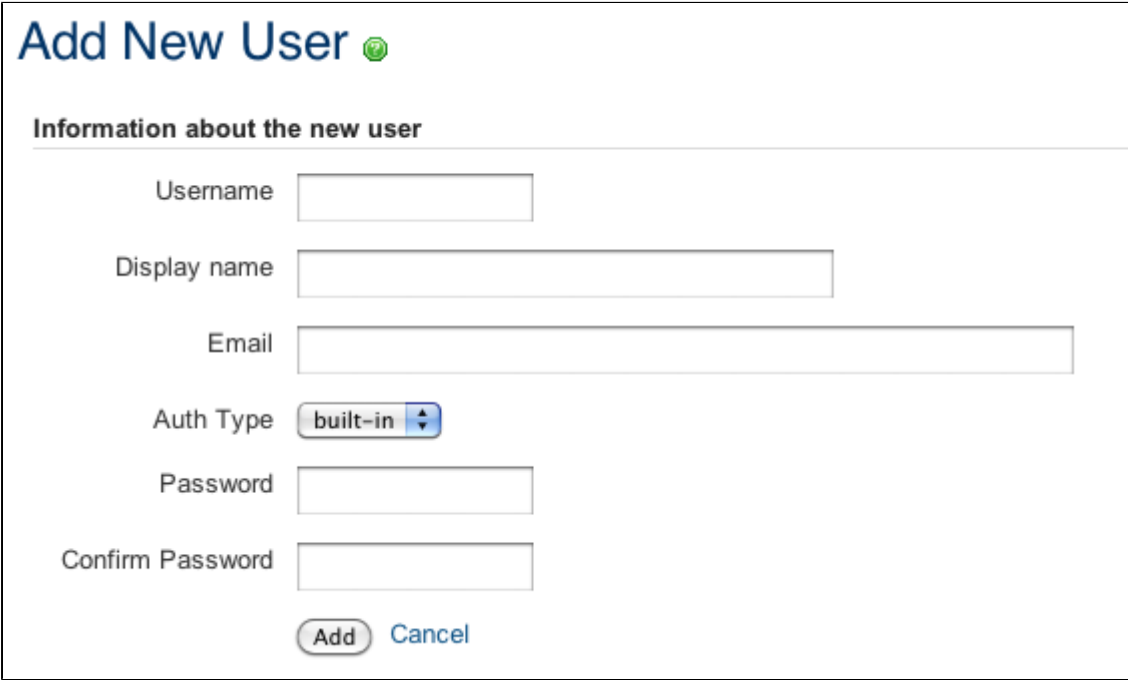
To add a new user:

1. In the [admin area](#), click **Users** under 'User Settings'.
2. Click **Add user** at the bottom of the screen.
3. Configure the following settings:

Username	Type the user's login name. You can use the following characters: • letters and numbers • hyphen ('-') • underscore ('_') • 'at' sign ('@')
Display name	The user's display name appears in the user interface.
Email	The email address is where the user will receive notifications .

Auth Type	Select either built-in or the appropriate external directory where the user identity will be stored.
Password	(For built-in users only)  The user can easily change their own password later.

4. Click **Add**.



Permissions

A *permission* is the ability to perform a particular action in Crucible, e.g. 'Create Review'.

Permissions in Crucible are managed through the use of permission schemes.

A *permission scheme* assigns particular [permissions](#) to any or all of the following:

- Particular [Users](#).
- Particular [Groups](#).
- All logged-in users.
- [Anonymous Users](#)
- People in particular [Review Roles](#), such as:
 - [Author](#)
 - [reviewer](#)
 - [creator](#)
 - [moderator](#)

The scheme's permissions will apply to all reviews belonging to the [project\(s\)](#) with which the scheme is associated.

You can create as many permission schemes as you wish. Each permission scheme can be associated with many projects or just one project, allowing you to tailor appropriate permissions for individual projects as required.

Further reading

- [Creating a permission scheme](#)
- [Associating a permission scheme with a project](#)
- [Agile permissions schemes in Crucible](#)

Creating a permission scheme

This page contains information on how to create a permission scheme in Crucible.

On this page:

- [Creating a permission scheme](#)
- [Editing a permission scheme](#)
- [List of Crucible permissions](#)

Creating a permission scheme

To create a permission scheme:

1. In the [admin area](#), click **Permission Schemes**, under 'Security Settings'.
2. Under 'Add a new permission scheme', enter a **Name** to uniquely identify your new scheme.
3. Click **Add**. Your new permission scheme will have the default assignees shown in the [permissions table](#) below.
4. If required, edit this permission scheme, as described below.

i Next step: see [Associating a permission scheme with a project](#).

Screenshot: Adding a Permission Scheme

Editing a permission scheme

To edit a permission scheme:

1. In the [admin area](#), click **Permission Schemes** under 'Security Settings'.
2. Click **edit** for the scheme you wish to change.
3. Click **edit** for the permission you wish to modify, and choose the appropriate assignee(s) for this permission:

Allow Anonymous users	Assign this permission to anonymous users
Allow All logged in users	Assign this permission to all logged-in users
Individuals	Enter a username to assign this permission to a particular user.
Groups	Enter a group name to assign this permission to a particular group of users.
Review Participants	Select check boxes to assign this permission to users who belong to any of the Reviewer / Creator / Author / Moderator participants .

Click **Save** when you are done.

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

4. Click **Save**.

Screenshot: Editing the 'Close' permission

Edit User Action

Close: *Ability to close a review once it has been summarized.*

☐ Allow All logged in users

Individuals

Start typing a user name then press enter to select.

☒ Seb Ruiz

Groups

Start typing a group name then press enter to select.

Review Participants ☒ Reviewer ☒ Creator ☒ Author ☒ Moderator

Save

Cancel

List of Crucible permissions

The following permissions are available:

Permission	Description	Default Assignees
Abandon	Ability to abandon (i.e. cancel) a review.	Creator Author Moderator
Approve	Ability to approve a review (i.e. issue it to the reviewers).	Creator Author Moderator
Close	Ability to close a review once it has been summarised.	Creator Author Reviewer Moderator
Comment	Ability to add or remove a comment to or from a review.	Creator Author Reviewer Moderator
Complete	Ability of a reviewer to change their individual review status to Complete.	Reviewer
Create	Ability to create a review.	All logged-in users
Delete	Ability to delete a review.	Creator Author Moderator
Edit Review Details	Ability to edit a review's details and change the set of revisions being reviewed.	Creator Author Reviewer Moderator

Re-Open	Ability to re-open a closed or abandoned review.	Creator Author Reviewer Moderator
Recover	Ability to resurrect an abandoned (i.e. cancelled) review.	Creator Author Reviewer Moderator
Reject	Ability to reject a review submitted for approval (i.e. prevent it from being issued to reviewers).	Creator Author Mode rator
Submit	Ability to submit a review for approval (i.e. request that the review be issued to the reviewers).	Creator Author Mode rator
Summarise	Ability to summarise a review. (Normally this would be done after all reviewers have completed their review.)	
Uncomplete	Ability of a reviewer to change their individual review status from Complete to Uncomplete.	Reviewer
View	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users Creator Author Reviewer Moderator

Associating a permission scheme with a project

This page explains how to associate a permission scheme with a Crucible project and show details of the default permission schemes included with Crucible.

On this page:

- [Associating a permission scheme with a Crucible project](#)
- [Overview of the permission schemes bundled with Crucible](#)
 - [Default permission scheme settings](#)
 - [Agile permission scheme settings](#)
- [Related links](#)

Associating a permission scheme with a Crucible project

To associate a permission scheme with a project:

1. In the [admin area](#), click **Projects** (under 'Project Settings').
2. Find the project you wish to associate with your permission scheme, and click **Edit** (in the 'Crucible Settings' column).
3. Under 'Permissions Scheme', choose a scheme from the **Permission Scheme** list.
 You will be shown a list of the schemes that have been created in Crucible. You can [create a new permission scheme](#) if necessary.
4. Click **Save**.

Overview of the permission schemes bundled with Crucible

Crucible comes with two permission schemes. 'Default' and 'Agile'. The following tables show the default settings

in detail; note that these can be easily edited by admin users to suit your needs.

Default permission scheme settings

This table shows the various permissions and which user groups have them by default.

Permission	Anonymous	All Logged In	Individuals	Groups	Review Roles
Abandon	false	false	None	None	Creator, Moderator
Approve	false	false	None	None	Moderator
Close	false	false	None	None	Moderator
Comment	false	false	None	None	Reviewer, Creator, Author, Moderator
Complete	false	false	None	None	Reviewer
Create	false	true	None	None	No roles. All logged-in users
Delete	false	false	None	None	Creator, Moderator
Edit Review Details	false	false	None	None	Creator, Moderator
Recover	false	false	None	None	Creator, Moderator
Reject	false	false	None	None	Moderator
Re-Open	false	false	None	None	Moderator
Submit	false	false	None	None	Creator
Summarize	false	false	None	None	Moderator
Uncomplete	false	false	None	None	Reviewer
View	false	true	None	None	Reviewer, Creator, Author, Moderator All logged-in users

 The default permission scheme has changed since Crucible 1.6.

Agile permission scheme settings

This table shows the various permissions and which user groups have them by default.

Permission	Anonymous	All Logged In	Individuals	Groups	Review Roles
Abandon	false	false	None	None	Creator, Author, Moderator
Approve	false	false	None	None	Creator, Author, Moderator
Close	false	false	None	None	Reviewer, Creator, Author, Moderator
Comment	false	false	None	None	Reviewer, Creator, Author, Moderator
Complete	false	false	None	None	Reviewer

Create	false	true	None	None	No roles. All logged-in users
Delete	false	false	None	None	Creator, Author, Moderator
Edit Review Details	false	false	None	None	Reviewer, Creator, Author, Moderator
Recover	false	false	None	None	Reviewer, Creator, Author, Moderator
Reject	false	false	None	None	Creator, Author, Moderator
Re-Open	false	false	None	None	Reviewer, Creator, Author, Moderator
Submit	false	false	None	None	Creator, Author, Moderator
Summarize	false	false	None	None	No roles.
Uncomplete	false	false	None	None	Reviewer
View	true	true	None	None	Reviewer, Creator, Author, Moderator Anonymous users All logged-in users

Related links

- [Creating a permission scheme](#)

Agile permissions schemes in Crucible

This page contains information about using and editing Agile permission schemes in Crucible.

Understanding the Agile Permissions Scheme

[Agile development](#) teams may not want to use the default Crucible permission schemes that require one person to approve or summarise reviews. Crucible ships with a pre-defined Agile permission scheme. By Agile, we mean permission schemes that have no [moderator](#) and very liberal permissions, suited to Agile or self-organising teams.

To use the Agile permissions scheme when creating a project, simply select **agile** from the **Permissions Scheme** list on the 'Edit Project' screen.

Considerations

 If you began your installation of Crucible with Crucible 2.1 or later, then this permission scheme will appear in the list of permission schemes in the administration menu.

 If you have upgraded from an earlier version of Crucible (pre Crucible 2.0), then the Agile permission may not appear by default. However, you can still create the equivalent by [disabling the moderator](#) when creating projects, allowing freer access to summarising, closing and generally tending to Crucible reviews.

 If you [disable the moderator role](#) on the Edit Project screen, then Crucible will check the current permission scheme. If the current permission scheme requires a moderator, a warning will be shown and you will be prompted to create a new permission scheme which will be called 'Agile' (or Agile-X if the name Agile already exists, where X is a number appended to the scheme name). The new permissions scheme will not require a moderator to carry out any actions.

Screenshot: Warnings on Permission Schemes that Require a Moderator

Project Permissions Scheme ⓘ

Permission Scheme: default ▼

The permission scheme "default" requires the Moderator role, which has been disabled.

All existing permission schemes require the Moderator role. Create an "unmoderated" permission scheme.

Moderator ⓘ

☐ Enable the Moderator role for this project

Migrating to an external database

This page contains information about migrating Crucible from its default embedded HSQL database to an external database. Advantages of using a database other than the embedded HSQL database include:

- **Improved Protection Against Data Loss:** The Crucible built-in database, running [HSQLDB](#) is somewhat susceptible to data loss during system crashes. External databases are generally more resistant to data loss during a system crash. HSQLDB is not supported in production environments and should only be used for evaluation purposes.
- **Performance & Scalability:** if you have many users on your Crucible instance, running the database on the same server as FishEye may slow it down. When using the embedded database, the database will always be hosted and run on the same server as Crucible.
- **Data Stored in the Crucible Database:** The Crucible database stores all information besides the cache for repository scans. This means all reviews, comments, review states, user data and user preferences information.

On this page:

- [Supported Databases](#)
- [Support for Other Databases](#)
- [Notes](#)

Supported Databases

You can use a number of alternatives to the built-in HSQLDB database for storing FishEye and Crucible's relational data. The supported alternative databases are listed on the [Supported platforms](#) page. Please note, that only the database versions listed on that page are supported.

The pages linked below outline the steps required to switch to an external database:

- [Migrating to MySQL](#)
- [Migrating to Oracle](#)
- [Migrating to PostgreSQL](#)
- [Migrating to SQL Server](#)

Support for Other Databases

If you are using another database product that you would like to see supported, please create a [JIRA issue](#) for it under the Crucible project.

Notes

Crucible uses Read Committed transaction isolation. There is no requirement to configure this explicitly when setting up an external database - Crucible will configure the transaction isolation when connecting to the

database.

Migrating to MySQL

This page describes how to use FishEye/Crucible with both MySQL Enterprise Server and MySQL Community Server. Note that when they are used together, FishEye and Crucible share the same external database.

To switch to a MySQL database, install MySQL and then follow the steps below. Please note that during the migration of database servers, the FishEye/Crucible instance will not be available to users or to external API clients.

 Note that for FishEye 2.9+, the JDBC driver for MySQL is *not bundled* with FishEye/Crucible (due to [licensing restrictions](#)).

MySQL 5.6.x and 5.7.x compatibility

For MySQL 5.6, FishEye/Crucible requires at least version 5.6.11.

For MySQL 5.7, FishEye/Crucible requires at least version 5.7.5.

Prerequisites

To start with:

1. Install a *supported version* of MySQL. Check the [Supported platforms](#) page for the exact versions that are supported. Note that MariaDB and Percona variants of MySQL are not supported, and are known to cause issues when used with FishEye.
2. Download and install the JDBC driver, if necessary. Note that for FishEye 2.9+, the JDBC driver for MySQL is *not bundled* with FishEye/Crucible (due to [licensing restrictions](#)).
 - a. Download the **MySQL Connector/J JDBC driver** from the [MySQL download website](#).
 - b. Expand the downloaded **zip/tar.gz file**.
 - c. Copy the `mysql-connector-java-x.y.zz-bin.jar` file to your `FISHEYE_INST/lib` directory. If the `lib/` directory doesn't already exist, you will need to create it.
 - d. Restart FishEye/Crucible.

On this page:

- [Prerequisites](#)
- [Step 1. Create a MySQL database](#)
- [Step 2. Configure FishEye/Crucible to use MySQL, and migrate data](#)

Related pages:

- [Migrating to PostgreSQL](#)
- [Migrating to Oracle](#)
- [Migrating to SQL Server](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Step 1. Create a MySQL database

Set up a MySQL database as follows:

- Configure the database to use the InnoDB storage engine
- Create a database on MySQL for FishEye/Crucible to use
- Create a FishEye user on the database
- Configure the database to use `utf8` character set encoding
- Configure the database to use `utf8_bin` collation (to ensure case sensitivity).

Here is an example of how to do that. When FishEye/Crucible and MySQL run on the same machine (accessible through `localhost`), issue the following commands (replacing `fisheyeuser` and `password` with you own

values):

```
mysql> SET GLOBAL storage_engine = 'InnoDB';
mysql> CREATE DATABASE fisheye CHARACTER SET utf8 COLLATE utf8_bin;
mysql> GRANT ALL PRIVILEGES ON fisheye.* TO 'fisheyeuser'@'localhost'
IDENTIFIED BY 'password';
mysql> FLUSH PRIVILEGES;
mysql> QUIT
```

For MySQL 5.6 and later, replace the first statement (SET GLOBAL storage_engine = 'InnoDB') with the following:

```
mysql> SET GLOBAL default_storage_engine = 'InnoDB';
```

This creates an empty MySQL database with the name `fisheye`, and a user that can log in from the host that FishEye is running on who has full access to the newly created database. In particular, the user should be allowed to create and drop tables, indexes and other constraints.

You will also need to set the Server Characterset to `utf8`. This can be done by adding the following in `my.ini` for Windows or `my.cnf` for other operating systems (create the file at `/etc/my.cnf` if it doesn't already exist). It has to be declared in the Server section, which is the section after `[mysqld]`:

```
[mysqld]
character-set-server=utf8
```

You'll need to restart MySQL for that change to take effect. Now use the `status` command to verify database character encoding information:

```
mysql> use fisheye;
mysql> status;
```

Screenshot: Using the MySQL status command

```
mysql> status
-----
/Applications/xampp/xamppfiles/bin/mysql Ver 14.12 Distrib 5.0.67, for apple-da
rwin8.11.1 (i686) using EditLine wrapper

Connection id:          5764
Current database:
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.0.67 Source distribution
Protocol version:       10
Connection:             Localhost via UNIX socket
Server character set:   utf8
Db character set:       utf8
Client character set:   latin1
Conn. character set:    latin1
UNIX socket:            /Applications/xampp/xamppfiles/var/mysql/mysql.sock
Uptime:                 29 days 1 hour 34 min 56 sec

Threads: 6  Questions: 49238134  Slow queries: 0  Opens: 460445  Flush tables: 1
Open tables: 4  Queries per second avg: 19.607
-----

mysql> 
```

Both the server and database character set must be UTF-8. You may not initially notice a problem because of the [character set overlap](#) but this is indeed a requirement. We've verified that when using a charset such as Swedish 7 bit as the server charset and UTF-8 as the database charset data corruption is easy to reproduce and obvious. This proves that the server character set is as important and that data corruption is possible when there is a charset mismatch. We absolutely cannot suggest a configuration where the server charset and database charset are anything other than UTF-8.

Step 2. Configure FishEye/Crucible to use MySQL, and migrate data

In order to migrate to a different database backend, you must create a backup of sql data, configure the database and finally import the data using a backup restoration process. This can be done from either the FishEye/Crucible administration console, which streamlines the process, or using the command line tool which FishEye/Crucible provides.

Option 1: Migrate using the UI (FishEye/Crucible Administration)

1. Navigate to the Database page in FishEye/Crucible's Administration console.

To log in to the Admin area, you can either:

- click **Administration** at the foot of the page.
- navigate to <http://HOSTNAME:8060/admin/>, where HOSTNAME is the name of the server on which you installed Fisheye.

Once logged in as an [administrator](#) you can also get to the Admin area by clicking the 'cog' menu in the FishEye/Crucible header, and choosing **Administration**.

2. Choose **Edit > Test Connection** to verify that FishEye/Crucible can log in to the database.
3. Select **MySQL** from the database type.
4. Fill in the appropriate fields, replacing the host, port, database name, username and password using the same connection details as used when creating the MySQL database in Step 1 above.
5. Click **Test Connection** to validate the values.

Screenshot: Testing the connection

Database Configuration

To switch to a different database, specify the database's configuration settings in the form below and use the Test Connection button to verify that the database can be used. We are currently connected to the Oracle database at jdbc:oracle:thin:@localhost:1521:xe.

Changes to connection pool settings will take effect after the application restarts before the new settings will take effect.

Type: 

Driver Location: 

URL:

User Name:

Password:

Minimum Pool Connections:

Maximum Pool Connections:

Parameters:

If this fails, verify that you have the MySQL JDBC driver `.jar` file in the classpath (see [Prerequisites](#) section above for instructions on how to install the driver). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

- Click **Save & Migrate** to start the migration process.

During the migration process (which will take several minutes, depending on the size of your database and network throughput), the product will be inaccessible to users and external API clients. Users will see a maintenance screen that informs them of the process. Should the migration fail for any reason, FishEye/Crucible will not switch to the new database, and will report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration.

Screenshot: Migrating the database

Migrating Database

 **Migrating database...**

From: jdbc:hsqldb:file:/Users/ervzijst/cenqua/demotion/var/data/crudb/crucible
To: jdbc:mysql://localhost/crucible

```

Started
Creating table definitions...
1 rows written, 0 tables completed.
9 rows written, 1 tables completed.
2328 rows written, 2 tables completed.
7327 rows written, 2 tables completed.
11143 rows written, 3 tables completed.
12676 rows written, 3 tables completed.
12740 rows written, 4 tables completed.
22739 rows written, 4 tables completed.
30492 rows written, 5 tables completed.
30493 rows written, 5 tables completed.
30497 rows written, 6 tables completed.
  
```

Option 2: Migrate using the command line

1. Create a backup of the sql data from the FishEye/Crucible instance. Read [Backing up and restoring FishEye data](#) and [Backing up and restoring Crucible data](#) for information on how to create a backup.
2. Run the following command from the <FishEye installation directory>/bin directory:

```
$ ./fisheyectl.sh restore --sql \  
    --file /path/to/backup.zip \  
    --dbtype mysql \  
    --jdbcurl jdbc:mysql://hostname/dbname \  
    --username crucible \  
    --password password
```

3. When the import is complete, FishEye/Crucible can be started and will use MySQL.

Migrating to Oracle

To switch to an Oracle database, install Oracle and follow the steps below. When they are used together, FishEye and Crucible share the same external database.

Please note that during the migration of database servers, the FishEye/Crucible instance will not be available to users or to external API clients.

Oracle support for FishEye/Crucible and Crucible was introduced in version 2.5.0. In order to migrate to Oracle, your instance must be *currently running* at least version 2.5. If you are running an older version, then you will be required to first upgrade FishEye/Crucible and then migrate.

On this page:

- [Step 1. Install and Create a Oracle Database](#)
- [Step 2. Configure FishEye/Crucible to use Oracle, and Migrate Data](#)

Related pages:

- [Migrating to MySQL](#)
- [Migrating to PostgreSQL](#)
- [Migrating to SQL Server](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Step 1. Install and Create a Oracle Database

1. The JDBC drivers for Oracle are bundled with FishEye/Crucible. Skip to step 2 if this meets your needs. If you want to install a specific, different version of the bundled JDBC driver, download the Oracle JDBC driver .jar file from the Oracle website (<http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>) and copy the .jar file to your FISHEYE_INST/lib directory (create the lib/ directory if it doesn't already exist). Move the existing JDBC .jar file to another location (and back it up). Restart FishEye/Crucible to have it pick up the new driver.
2. Because creating a database with Oracle is a complex process, we recommend speaking to your resident DBA for creation of a new database for usage with Crucible. We *highly* recommend installing Oracle with the AL32UTF8 encoding otherwise you may see encoding issues in the product.

Permissions

Ensure the database user has CREATE TABLE, CREATE SEQUENCE and CREATE TRIGGER permissions in addition to the read/write permissions to the database.

Step 2. Configure FishEye/Crucible to use Oracle, and Migrate Data

In order to migrate to a different database backend, you must create a backup of sql data, configure the database and finally import the data via a backup restoration process. This can be done from either the FishEye/Crucible administration console, which streamlines the process, or via the command line tool which FishEye/Crucible provides.

Option 1: Migrate using the UI (FishEye/Crucible Administration)

1. Navigate to the Database page in FishEye/Crucible's Administration console.

To log in to the Admin area, you can either:

- click **Administration** at the foot of the page.
- navigate to <http://HOSTNAME:8060/admin/>, where HOSTNAME is the name of the server on which you installed Fisheye.

Once logged in as an **administrator** you can also get to the Admin area by clicking the 'cog' menu in the FishEye/Crucible header, and choosing **Administration**.

2. Then choose **Edit > Test Connection** to verify that FishEye/Crucible can log in to the database.
3. Select **Oracle** from the database type
4. Fill in the appropriate fields, replacing the host, port, database name, username and password using the same connection details as used when creating the Oracle database in Step 1 above.
5. Click on **Test Connection** to validate the values

Screenshot: Testing the Connection

Database Configuration 🟢

To switch to a different database, specify the database's configuration settings in the form below and use the Test Connection button to verify that the database can be used. We are currently connected to the PostgreSQL database at jdbc:postgresql://localhost/crucible.

Changes to connection pool settings will not take effect until you click the Test Connection button.

Type: **Oracle** (Built-In PostgreSQL)

Driver Location: **MySQL**

URL: jdbc:oracle:thin:@localhost:1521:crucible

User Name: crucible

Password:

Minimum Pool Connections: 5

Maximum Pool Connections: 20

Parameters:

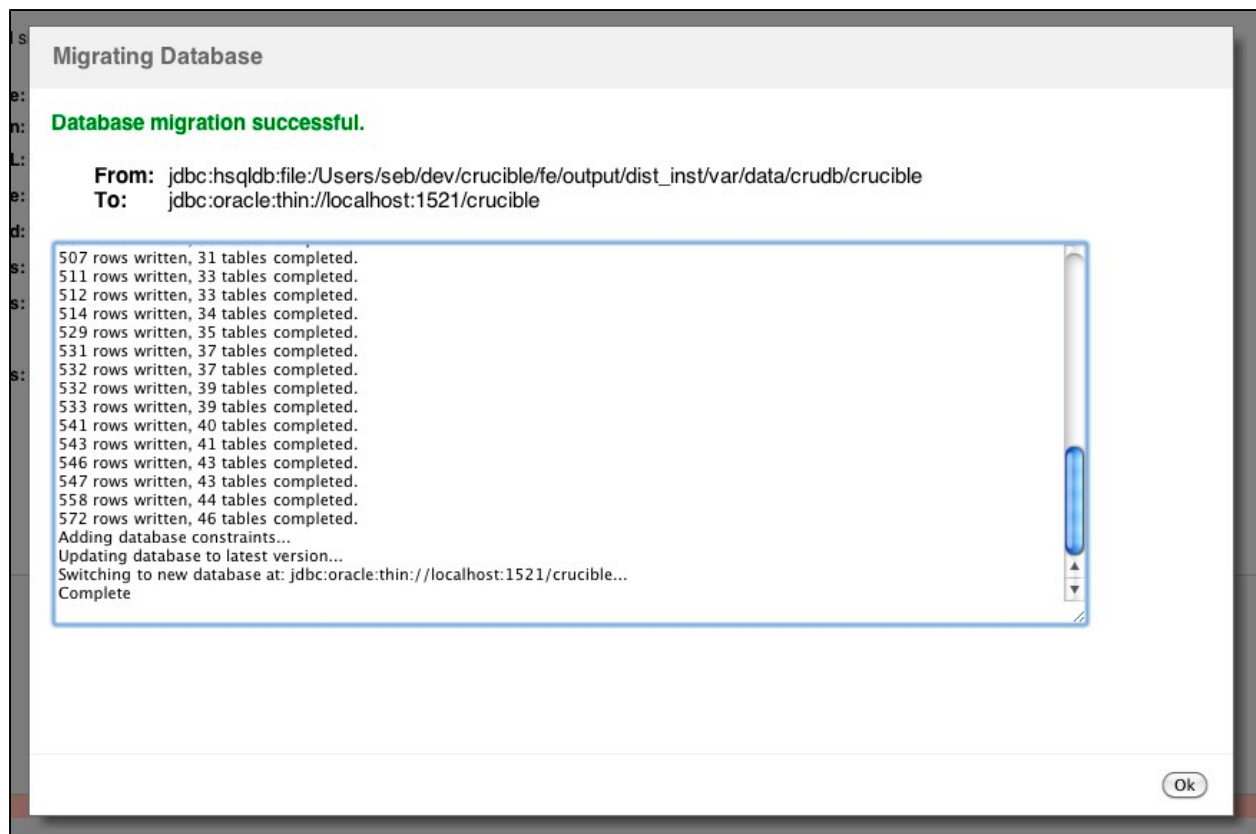
Buttons: Cancel, Save & Migrate, Test Connection, Save

If this fails, verify that you have the Oracle JDBC driver `.jar` file in the classpath (by placing the `.jar` file in `FISHEYE_INST/lib`). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

6. Click **Save & Migrate Data** to start the migration process.

During the migration process (which will take several minutes, depending on the size of your database and network throughput), the product will be inaccessible to users and external API clients. Users will see a maintenance screen that informs them of the process. Should the migration fail for any reason, FishEye/Crucible will not switch to the new database and report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration.

Screenshot: Migrating the Database



Option 2: Migrate using the command line

1. Create a backup of the **sql data** from the FishEye/Crucible instance. Information on how to create a backup can be found at [Backing up and restoring FishEye data \ Backing up and restoring Crucible data](#)
2. Run the following command from the <FishEye installation directory>/bin directory:

```
$ ./fisheyectl.sh restore --sql \
    --file /path/to/backup.zip \
    --dbtype oracle \
    --jdbcurl jdbc:oracle:thin:@hostname:port:dbname \
    --username crucible \
    --password password
```

3. When the import is complete, FishEye/Crucible can be started and will use Oracle.

Migrating to PostgreSQL

To switch to a PostgreSQL database, install PostgreSQL and follow the steps below. When they are used together, FishEye and Crucible share the same external database.

Please note that during the migration of database servers, the FishEye/Crucible instance will not be available to users or to external API clients.

On this page:

- [Step 1. Install and Create a PostgreSQL Database](#)
- [Step 2. Configure FishEye/Crucible to use PostgreSQL, and Migrate Data](#)

Related pages:

- [Migrating to MySQL](#)
- [Migrating to Oracle](#)
- [Migrating to SQL Server](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Step 1. Install and Create a PostgreSQL Database

1. The JDBC drivers for PostgreSQL are bundled with FishEye/Crucible. Skip to [Step 2](#) if this meets your needs. If you want to install a specific, different version of the bundled JDBC driver, download the PostgreSQL JDBC driver .jar file from the [PostgreSQL website](#) and copy the .jar file to your FISHEYE_INST/lib directory (create the lib/ directory if it doesn't already exist). Move the existing JDBC .jar file to another location (and back it up). Restart FishEye/Crucible to have it pick up the new driver.
2. Create a new database user (replacing 'username' and 'password' with the appropriate values):

```
$ psql
> create user username password 'password';
```

3. Create a UTF-8 database and make the newly created user the owner:

```
> create database crucible ENCODING 'UTF-8' OWNER username;
```

4. Make sure the user has full access to the database:

```
> grant all on database crucible to username;
```

Step 2. Configure FishEye/Crucible to use PostgreSQL, and Migrate Data

In order to migrate to a different database backend, you must create a backup of SQL data, configure the database and finally import the data via a backup restoration process. This can be done from either the FishEye/Crucible administration console, which streamlines the process, or via the command line tool which FishEye/Crucible provides.

Option 1: Migrate using the UI (FishEye/Crucible Administration)

1. Navigate to the Database page in FishEye/Crucible's Administration console.

To log in to the Admin area, you can either:

- click **Administration** at the foot of the page.
- navigate to <http://HOSTNAME:8060/admin/>, where HOSTNAME is the name of the server on which you installed Fisheye.

Once logged in as an [administrator](#) you can also get to the Admin area by clicking the 'cog' menu in the FishEye/Crucible header, and choosing **Administration**.

2. Choose **Edit > Test Connection** to verify that FishEye/Crucible can log in to the existing database.
3. Select PostgreSQL from the database **Type**.
4. Fill in the appropriate fields, using the same connection details as used when creating the PostgreSQL database in Step 1 above.
 - a. **Driver Location**: either your own PostgreSQL JDBC or the **Bundled** one that came with FishEye
 - b. **URL**: create this field by replacing the host, port, and database name with your own (i.e. jdbc:postgresql://localhost:5432/<dbname> e.g. jdbc:postgresql://localhost:5432/crucible)

- c. **Username:** your DB username
 - d. **Password:** your DB password
 - e. **Minimum Pool Connections:** 5 is the default
 - f. **Maximum Pool Connections:** 20 is the default
 - g. **Parameters:** (one per line)
 - i. useUnicode=true
 - ii. characterEncoding=UTF8
5. Click **Test Connection** to validate the values.

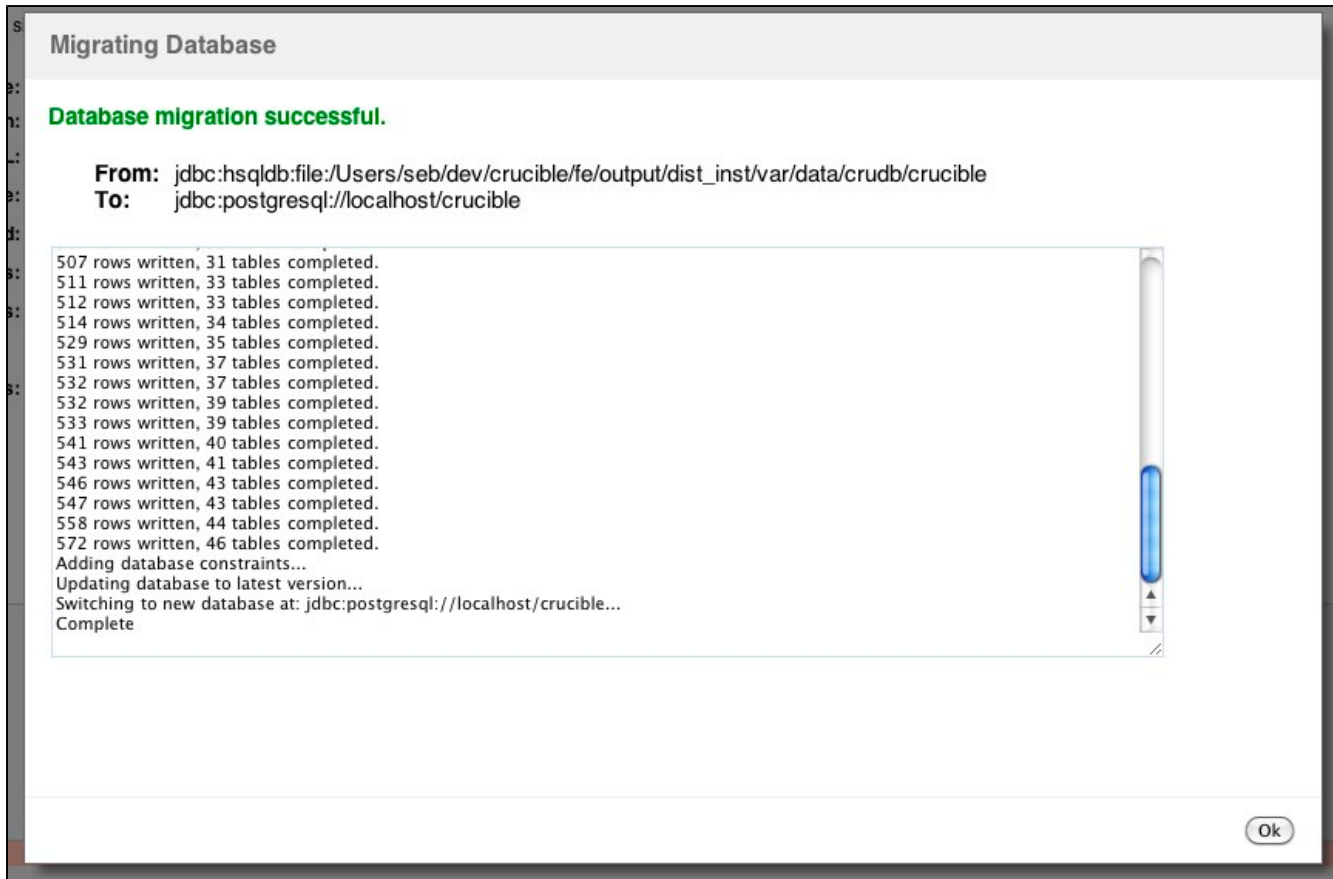
Screenshot: Testing the Connection

If this fails, verify that you have the PostgreSQL JDBC driver `.jar` file in the classpath (by placing the `.jar` file in `FISHEYE_INST/lib`). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

6. Click **Save & Migrate** to start the migration process.

During the migration process (which will take several minutes, depending on the size of your database and network throughput), the product will be inaccessible to users and external API clients. Users will see a maintenance screen that informs them of the process. Should the migration fail for any reason, FishEye/Crucible will not switch to the new database and report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration. If successful, the following message is displayed:

Screenshot: Migrating the Database



Option 2: Migrate using the command line

1. Create a backup of the sql data from the FishEye/Crucible instance. Information on how to create a backup can be found at [Backing up and restoring FishEye data / Backing up and restoring Crucible data](#).
2. Run the following command from the <FishEye installation directory>/bin directory:

```
$ ./fisheyectl.sh restore --sql \
    --file /path/to/backup.zip \
    --dbtype postgresql \
    --jdbcurl jdbc:postgresql://hostname/dbname \
    --username crucible \
    --password password
```

3. When the import is complete, FishEye/Crucible can be started and will use PostgreSQL.

Migrating to SQL Server

To migrate FishEye/Crucible to an SQL Server database, install SQL Server and follow the steps below. When they are used together, FishEye and Crucible share the same external database.

Before you begin

Check that you are using version of SQL Server that is supported for use with FishEye. See [Supported platforms](#).

An existing Java bug prevents connection with Java 1.6.0_29 and above (including Java 1.7.0). Read more about the issue and possible workarounds [here](#).

Step 1. Install and create an SQL Server database

See the [SQL Server Online resources \(MSDN\)](#) for instructions on how to install and create an SQL Server database.

Please note the following FishEye/Crucible-specific information when installing and creating an SQL Server database:

- The JDBC jtds drivers for SQLServer are bundled with FishEye/Crucible. We do not support using the Microsoft distributed jdbc driver.
- The FishEye database user must have permission to connect to the database and to create and populate tables.
- The database user should **not** be the database owner, but **should** be in the db_owner role. (See [SQL Server Startup Errors](#) for details.)
- Your database must be configured to use the Latin1_General_CS_AS collation set.
- Your database should be configured to use snapshot mode for the transaction isolation level. To enable snapshot mode, run:

```
ALTER DATABASE crucible
SET READ_COMMITTED_SNAPSHOT ON;
```

See [this](#) and [this](#) Microsoft MSDN articles for more information.

Note that it is preferable to run the above command after stopping FishEye/Crucible (and with no other applications connected to the SQL Server database), especially if you find that the alter statement does not complete quickly.

On this page:

- [Before you begin](#)
- [Step 1. Install and create an SQL Server database](#)
- [Step 2. Configure FishEye/Crucible to use SQL Server and migrate data](#)

Related pages:

- [Migrating to MySQL](#)
- [Migrating to PostgreSQL](#)
- [Migrating to Oracle](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Step 2. Configure FishEye/Crucible to use SQL Server and migrate data

In order to migrate to a different database backend, you must create a backup of your SQL data, configure the database and finally import the data via a backup restoration process. This can be done from either the Crucible [administration console](#), which streamlines the process, or via the [command line tool](#) which Crucible provides. These two methods are described below. The following resources may be of interest:

- [Backing up and restoring FishEye data](#)
- [Backing up and restoring Crucible data](#)
- [SQL Server Online resources \(MSDN\)](#)

Option 1: Migrate using the UI (FishEye/Crucible Administration)

Before you begin:

- Note, during the migration process (which will take several minutes, depending on the size of your database and network throughput), **the FishEye/Crucible instance will be inaccessible to users and external API clients**. Users will see a maintenance screen that informs them of the process.
- If you are attempting a migration after a previous migration has failed, you must drop all tables, indexes and constraints before attempting a new migration. This is because the destination database may contain data from the previous migration attempt.

- Verify that you have the jtds JDBC driver .jar file in the classpath (by placing the .jar file in FISHEYE_INST/lib).
- Ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

Please ensure that you use a **SQL Server** user account to log into your database, not a Windows user account.

To configure FishEye/Crucible to use SQL Server and migrate data using the administration console:

1. Navigate to the **Database** page (under 'System Settings') in FishEye/Crucible's Administration console.

To log in to the Admin area, you can either:

- click **Administration** at the foot of the page.
- navigate to <http://HOSTNAME:8060/admin/>, where HOSTNAME is the name of the server on which you installed FishEye.

Once logged in as an **administrator** you can also get to the Admin area by clicking the 'cog' menu in the FishEye/Crucible header, and choosing **Administration**.

2. Configure FishEye/Crucible to use SQL Server, as follows:

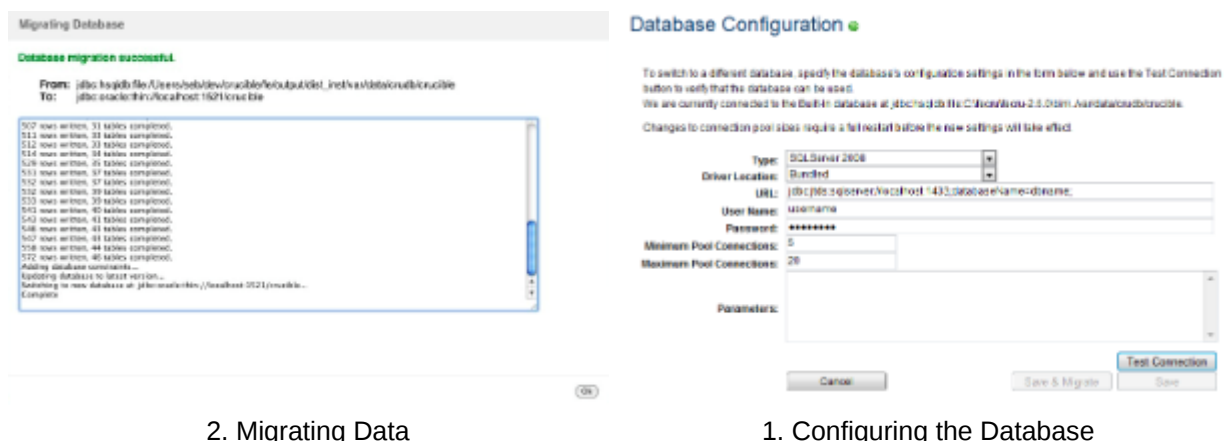
- Choose **Edit > Test Connection** to verify that FishEye/Crucible can log in to the database.
- Select **SQLServer** from the **Type** dropdown.
- Complete the appropriate fields, replacing the **URL** (host, port and database name), **User Name** and **Password** as required, using the same connection details as used when creating the SQL Server database in Step 1 above.

NOTE: The default SQL server instance listens on port 1433. If your instance is not the default, use the port number that is associated with your particular instance.

e.g. **URL**: jdbc:jtds:sqlserver://localhost:1433;databaseName=*your database name here*;

3. Click **Test Connection** to verify that FishEye/Crucible can log in to the database (see 'Testing the Connection' screenshot below).
4. Click **Save & Migrate Data** to start the migration process (see 'Migrating the Database' screenshot below). If the migration fails, FishEye/Crucible will not switch to the new database and will report the problems encountered.

Screenshots: Configuring FishEye/Crucible to use SQL Server and migrating data (click to view full-size images)



2. Migrating Data

1. Configuring the Database

Option 2: Migrate using the command line

To configure FishEye/Crucible to use SQL Server and migrate data using the command line:

1. Create a backup of the sql data from the FishEye/Crucible instance. Information on how to create a backup can be found at [Backing up and restoring FishEye data](#) \ [Backing up and restoring Crucible data](#)
2. Run the following command from the <FishEye installation directory>/bin directory:

```
$ ./fisheyectl.sh restore --sql \  
    --file /path/to/backup.zip \  
    --dbtype sqlserver2012 \  
    --jdbcurl  
"jdbc:jtds:sqlserver://hostname:port;databaseName=dbName;" \  
    --username crucible \  
    --password password
```

3. When the import is complete, FishEye/Crucible can be started and will use SQLServer.

Backing up and restoring Crucible data

Crucible data can be backed up from the admin interface or command line. This page contains the command syntax, options and the required procedure to backup and restore your Crucible instance.

Backing up Crucible data

The Crucible admin backup process

1. Navigate to the Crucible Admin area (click the **Administration** link in the footer of any Crucible page).
2. Click **Backup** (under 'System' heading in the left navigation bar).
3. The **File Path** field indicates where the backup file (in .zip format) will be stored. You can manually edit this path to change it. Under 'Include', a list of check boxes is shown, with the following items:

On this page:

- [Backing up Crucible data](#)
 - [The Crucible admin backup process](#)
 - [The Crucible command line backup process](#)
 - [Components of a Crucible backup](#)
 - [Backup command line options](#)
 - [Command line examples](#)
 - [Advanced backup command line settings](#)
 - [Known limitations](#)
- [Scheduling Crucible backups](#)
- [Restoring Crucible data](#)
 - [Restoring Crucible data from the command line](#)
 - [Command line restore options](#)
 - [Advanced command line restore settings](#)

- [Plugins and their configuration data](#)
- [SQL database](#)
- [Web templates](#)
- [Uploaded files and local copies of files under review.](#)
- [Repository and application caches.](#)
 - [Repository and application caches contain temporary data stored from repository scans and library caches that improve startup time. Both will be recreated automatically by re-scanning the source repositories, so the backup files can be reduced by a significant amount by excluding these \(if the cost of re-scanning is acceptable\).](#)

4. Once you have chosen your options, click **Create Backup Now**.

Screenshot: The Crucible Backup Screen

Backup

FishEye and Crucible will be unavailable until the backup completes

Depending on the number of repositories and reviews, the backup may take a long time to complete and may require large amounts of disk space (especially when you are including the cache). Please make sure you have enough disk space before proceeding.

Your managed repositories will not be backed up!

FishEye does not automatically backup content stored in your managed repositories. Please ensure that a backup process which is satisfactory to your requirements is in place. [Read more](#)

Create a new backup

Filename

Specify the filename on the server where FishEye and Crucible should export.
This file will be created in /opt/j2ee/domains/atlassian.com/eac-fisheye/data/backup

Include

- ☒ ActiveObjects
- ☒ repository and application caches
- ☒ plugins and their configuration data
- ☒ SQL database
- ☒ web templates
- ☒ uploaded files and local copies of files under review

Choose the components that FishEye and Crucible should include in the backup.

or [Manage Scheduled Backups](#)

The Crucible command line backup process

i Your Crucible instance must be running during the backup.

1. Open a command line interface on the Crucible server computer.
2. Navigate to the <Crucible home directory>/bin/ directory.
3. Run the backup command on the command line with the desired options.
4. The backup is created as a new Zip archive file and placed in the FISHEYE_INST/backup/ directory.

i Note that if your Crucible instance uses a custom FISHEYE_INST directory; make sure the environment variable is properly set when running the backup command.

Components of a Crucible backup

The Crucible backup is highly configurable and allows for many different configurations. This table shows the various components of the backup, what they are for and how they can be used.

Component	Purpose	Defaults
SQL Database	Refers to the SQL content database (used by both FishEye and Crucible and containing all user profile data, reviews and their comments).	Backed up by default.
Cache	The cache contains data that reflects the state of FishEye's repositories. Without it, FishEye must re-scan its repositories after a backup is restored. The cache also contains OSGI library data that increases startup time. These too can be excluded and will be generated automatically when the application is started.	The cache is not backed up by default as it tends to be large (running a risk of pushing the maximum file size for Java backups), whilst also representing replaceable data.
Plugins	Plugins are 3rd-party extensions that you may have installed, and configuration for all plugins (this includes configuration for Crucible's set of standard plugins).	Configuration data for all plugins are backed up by default, as well as all plugins installed in FISHEYE_INST/var/plugins/user.
Templates	In this context, these are custom freemarker templates that you or your users have created. They live in FISHEYE_INST/template.	Templates are backed up by default. You can choose to exclude them from the backup if your templates directory is covered by some other backup mechanism.

Uploads	In this context, uploads refers to files which are added to Crucible via the web interface (such as patch file reviews). It also includes each repository-backed file that went under review, when Crucible is configured to make a local copy of every reviewed file .	Uploads are backed up by default. You can choose not to back them up for example when the FISHEYE_INST/var/data/uploads directory is already covered by some other backup mechanism.
ActiveObjects	Configuration data stored by plugins	Backed up by default

Note that the backup will always include the configuration data (config.xml), your license file and the FishEye user data.

Backup command line options

These examples are for use in a Linux-like operating system. When using these commands on Windows, use the file name fisheyectl.bat and use the correct slashes. Run the command from the <Crucible home directory>/bin/ directory.

The basic syntax of the backup command is as follows:

```
$ ./fisheyectl.sh backup [OPTIONS]
```

To see inline help for all backup options, run the following command in the <Crucible home directory>/bin/ directory:

```
$ ./fisheyectl.sh backup --help
```

Option	Switch	Description	Default
Quiet mode	-q OR --quiet	Suppresses output	No
Output filename	-f OR --file	Specify a different path and filename to the FISHEYE_INST/backup/backup_YYYY-DD-MM_HHmm.zip file. When filename is omitted, the backup filename contains the date and time.	FISHEYE_INST/backup/backup_YYYY-DD-MM_HHmm.zip
Compression level	--compression OR -c	Sets the Zip compression level, from 1-9. Runs at level 6 if no argument is passed.	Yes (6)
Anonymise	-a OR --anonymise	Anonymises the SQL database by replacing all text with 'x'. This is only useful when sending a backup to Atlassian as part of a support case. Please do not anonymise data unless the Support Engineer handling your support case has specifically requested the data anonymised (as often anonymised data will not help reproduce the issue).	No
Cache Backup	--cache	Include the repository caching files in the backup. These hold information gained from scanning the repositories and can be quite large (many gigabytes). However, it can shorten the time needed to re-scan the repositories after data is restored.	No. By default, repository caching files are not backed up.

Command line examples

 These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename fisheyectl.bat and use the correct slashes. Run the command from the <Crucible home directory>/bin/ directory.

Backing up with compression of 9, quiet mode and setting an output location

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

Backup including cache data (also includes all default components)

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

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

```
$ ./fisheyectl.sh restore --cache
```

Advanced backup command line settings

In some cases it might be preferable to only backup a limited set of items. This could be useful when your instance uses an external database such as MySQL or PostgreSQL and your DBA has already configured automatic backups in the database. The commands below allow this.

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

Known limitations

Please note that the below limitations are common for any Java based backup tool.

Archives Containing Over 65535 Files

Versions of Java earlier than v1.6 (b25) are incapable of handling zip files that contain more than 65,535 files. The solution for this problem is to either upgrade to a version of Java later than v1.6 (b25), or ensure that the archive does not exceed the threshold (contains less than 65,535 files). The FishEye cache (not included in backups by default) can be a contributor of many small files. Hence, exclude the cache from backups if this is likely to be a concern.

Archives Larger Than 4GB

Java has trouble reading and writing zip files that are larger than 4GB. As of release 1.5 Java appears capable of reliably creating archives that are over 4GB, but remains unable to extract them. For details see [Sun's bug report](#). Also be aware of the fact that some file systems (including [FAT32](#)) have trouble with files larger than 4GB.

As a workaround, make sure you do not create archives that are larger than 4GB. The FishEye cache (not

included in backups by default) can be a contributor of a lot of small files (although these tend to compress very well). If you still want to archive everything and end up with an archive that is too large, consider creating separate backups for the FishEye cache and uploaded files respectively.

Scheduling Crucible backups

To set a schedule for automatic backups, open the administration screen and click '**Backup**' under '**System**' on the left navigation bar. The 'Backup' page opens. Now, click the link '**Manage Scheduled Backups**' at the bottom of the page. The 'Scheduled Backups' page opens.

On the 'Scheduled Backups' page, click '**Edit**' to adjust the backup schedule. Set the desired options and click '**Save**'.

The options for scheduled backups are detailed in the table below.

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

Screenshot: Scheduling Backups in FishEye and Crucible

Scheduled Backups

 Please choose the schedule for your backups carefully. FishEye and Crucible will be unavailable for the duration of the backup process.

 Your managed repositories will not be backed up!
FishEye does not automatically backup content stored in your managed repositories. Please ensure that a backup process which is satisfactory to your requirements is in place. [Read more](#)

Schedule backups

Disable scheduled backups ☐ Switch scheduled backups on and off.

Backup directory The location to store the backups. Edit config.xml to change this setting.

Backup file prefix The prefix for the generated backup file.

Backup file date pattern The date/time format you specify will be added to the file prefix to differentiate between backup files.

Backup frequency How often do you want FishEye and Crucible to run a backup.

Backup Time (HH:mm) What time would you like FishEye and Crucible to run the backup, e.g. 17:30.
Note that this is 24 hour notation in local server time. Current local server time is: 02:04.

Include ☐ ActiveObjects
☒ repository and application caches
☒ plugins and their configuration data
☒ SQL database
☒ web templates
☒ uploaded files and local copies of files under review
Choose the components that FishEye and Crucible should include in the backups.

 Be aware that scheduled backups can fill up disks unless you regularly move or delete old archives.

Restoring Crucible data

Restoring Crucible data from the command line

 There is currently no way to restore a backup from the web interface because Crucible must be shut down during a data restore.

Restoring a backup will irreversibly overwrite the data of your installation with the data from the backup archive.

If you made a backup from production which connected to an external database, and restore this backup to a test server without specifying another database to restore too, you will drop and restore to your production database. Thus when restoring to a test server, always ensure you specify the correct database to restore to (or restore to an in-built database).

1. [Install Crucible](#) into a new, empty directory (this must be the same version that the backup was created from, or later).
 Note that you cannot restore data into versions of Crucible which are older than the version that created the backup.
2. Make sure the Crucible instance is not running.
3. Open a command line interface on the Crucible server computer.
4. Run the restore command on the command line with the desired options.
5. The specified elements will be restored.
6. Start the Crucible instance.
7. When using FishEye integrated with Crucible, you will need to re-index your repositories after restoring data, unless the backup archive was created with the `--cache` option.

Command line restore options

 These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyect1.bat` and use the correct slashes. Run the command from the `<Crucible> home`

directory>/bin/ directory.

The basic syntax of the restore command is as follows:

```
$ ./fisheyectl.sh restore -f /path/to/backup_2009-10-02_1138.zip [OPTIONS]
```

To see inline help for all backup options, run the following command in the <Crucible home directory>/bin/ directory:

```
$ ./fisheyectl.sh restore --help
```

Restores a FishEye/Crucible backup instance.

If you are using an external database (as opposed to the default built-in database), make sure the JDBC driver file is present in the FISHEYE_INST/lib directory when running restore.

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

Advanced command line restore settings

By default, the restore program will restore all items found in the backup archive (so if you included the caches using the --cache option, these will automatically be restored). However, it is possible to only restore a subset of items from the backup, by explicitly specifying the item names on the command line and only those will be restored.

Option	Switch	Description
Restore FishEye cache	--cache	Restore the repository cache backup.
Restore plugins	--plugins	Restore 3rd-party plugins and their configuration data.
Restore templates	--templates	Restore freemarker templates from the backup (the restored instance will use the built-in templates).
Restore uploads	--uploads	Restore uploads (e.g. patch files uploaded into Crucible and contents of files under review).
Restore Crucible reviews	--sql	Restore the SQL database containing user profiles, reviews and review comments.
Set database type	--dbtype OR -t	SQL database type (mysql, postgresql, sqlserver2008, sqlserver2012 or hsql). Only required when restoring to a database location different to that used at backup time.
Set JDBC URL	--jdbcurl OR -j	JDBC URL of the SQL database. Only required when restoring to a database location different to that used at backup time (not applicable for hsql).

Set JDBC username	--username OR -u	JDBC username of the SQL database. Only required when restoring to a database location different to that used at backup time (not applicable for hsql).
JDBC password	--password OR -p	JDBC password of the SQL database. Only required when restoring to a database location different to that used at backup time (not applicable for hsql).
JDBC class	--driver OR -d	Specifies the JDBC driver class name needed to access the SQL database. Only required when restoring to a database location different to that used at backup time and when using a different JDBC driver than the standard driver associated with the database specified through --dbtype. (Not applicable for 'built-in'.)

Notes on migrating backup data

When the process restores a SQL database, it looks at the configuration data (`config.xml`) included in the backup archive to learn which database product was used and how to connect to it. When Crucible uses the built-in HSQLDB database (which is the default), the restored instance will also use that.

However, when the restored instance will use a different database than the backed up instance (for instance, HSQLDB was used at the time the backup was created, but it needs to be restored on MySQL), use the command line options to point the process to the new database.

Command line example: migrating backup data to MySQL

i These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fishyeectl.bat` and use the correct slashes. Run the command from the `<Crucible home directory>/bin/` directory.

Restoring to a Crucible instance that uses a different database (ensure the mysql driver jar file is present in the `FISHEYE_INST/lib` directory)

```
$ ./fishyeectl.sh restore \
--username john \
--password smith \
--jdbcurl jdbc:mysql://localhost:3306/crucible \
--dbtype mysql \
--file /path/to/backup_2009-10-02_1138.zip
```

Customising Crucible

This section describes the ways in which you can configure Crucible.

- [Customising the welcome message](#)
- [Customising email notifications](#)
- [Customising the defect classifications](#)
- [Configuring User Managed Mappings](#)
- [Enabling Access Logging in Crucible](#)

Customising the welcome message

Crucible administrators can customise the welcome message that is displayed when Crucible starts by clicking **Front Page Customization** (under 'Global Settings') in the admin area.

You can provide either or both of:

- a custom welcome message that is displayed to users when they first log in.
- a custom support message that also appears on the opening page, and which gives contact details for your own support organisation.

The changes you make are applied immediately, without needing to restart Crucible.

The screenshot shows a web interface titled "Front Page Customization". It has two tabs: "Welcome Message" (active) and "Support Message". Below the tabs, a text description states: "The welcome message is displayed to users when they first log in." There are two radio button options: "Show the default welcome message" (unselected) and "Show a custom welcome message:" (selected). Below the selected option is a large text input area with the placeholder text "Enter your welcome message here...". A small note below the input area says "If blank, the default welcome message will be used." At the bottom, there are two buttons: "Save changes" (in blue) and "revert" (in grey).

Using HTML

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

```
<a href="http://www.atlassian.com">Link to Atlassian Home Page</a>
```

Editing the config.xml file

You can also directly edit the XML file that contains the welcome and support messages. This file is called `config.xml`, and is located in the Crucible installation folder.

To do this, simply add the following XML tags to `config.xml`:

```
<content>
  <front-page-message>Example welcome message here</front-page-message>
  <support-message>Example support message here</support-message>
</content>
```

Customising email notifications

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

Editing Crucible email templates

Template files for Crucible are stored in the `<Crucible home directory>/template/crucible/` folder and the `<Crucible home directory>/template/shared` folder.

There are sets of templates for both HTML and plain-text emails, as listed in the table below. Note that these templates do not support embedding full diffs into notifications. They are only for changing the appearance and order of certain content inside the messages.

You can edit templates in any text editor:

- Either: Stop Crucible before editing templates to avoid disrupting notifications that may otherwise be sent.
- Or: Avoid editing a live template file, as Crucible may try to use it while you are editing. This could have unpredictable results. Instead, back up the template file (it's wise to keep original versions of all these files), edit a copy you have made, then overwrite the 'live' template once you have finished.
- After an edit, the change to the email template will take place immediately. No restart is required.

Advanced editing of Crucible email templates

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

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

See [Freemarker Data Model for Email Templates](#) in the FishEye documentation for details of the data model used by FishEye/Crucible.

Crucible email template files

The following template files for Crucible notification are stored in the <Crucible home directory>/template/crucible/ folder, or its subfolders and <Crucible home directory>/template/shared/ or its subfolders

Template filename	HTML	Plain-text
common-mention-macros.ftl	Both	
notification-subject.ftl	Shared	
changeset-header.ftl	Shared	
util.ftl	Shared	
all-completed-notification.ftl	✓	✓
all-uncompleted-notification.ftl	✓	✓
batch-comment-note.ftl	✓	✓
batch-reviewer-note.ftl	✓	✓
batch-revision-added-note.ftl	✓	✓
batch-state-note.ftl	✓	✓
comment-render-macro.ftl	✓	✗
crucible-layout.ftl	✓	✓
cscomment-notification.ftl	✓	✓
general-notification.ftl	✓	✓
invite.ftl	✓	✗
mention-macros.ftl	✓	✓

mention-user-in-changeset-comment-notification.ftl	✓	✓
mention-user-in-review-comment-notification.ftl	✓	✓
mention-user-in-review-description-notification.ftl	✓	✓
review-allcomments.ftl	✗	✓
review-header.ftl	✓	✗
review-participant-row.ftl	✓	✗
review-reminder-notification.ftl	✓	✓
review-summary.ftl	✓	✓
share-footer.ftl	✓	✗
share-header.ftl	✓	✗
share-notification.ftl	✓	✓

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

Freemarker Data Model for Email Templates

Customising Crucible email templates with Freemarker

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

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

Example

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

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

Customising the defect classifications

This page explains how to customise defects and their classifications in Crucible.

On this page:

- [Defects in Crucible comments](#)
- [Changing classification settings](#)
- [Default Crucible classifications](#)
 - [Ranking](#)
 - [Classification](#)

Defects in Crucible comments

Defects are comments made by [reviewers](#) that indicate a problem in a review. Defects can be classified by rank and type. Custom classifications can also be defined. The default classifications are shown in the screenshot below.

Seb Ruiz 5a4bdde 60 60 #snippet-root-pane .snippet-content {

Liang Zheng 4ea4bbe 61 61 background-color: #ebf2f9;

Seb Ruiz fb25cfb 62 62 border-top: 1px solid #CCC;

Does this fit with the design guidelines?

Quote source ?

☒ Defect Major

☒ Select Classification

Wrong

Can be improved

Risk-prone

Security hole

Over engineered

Code duplication

Comment Keep as draft Discard

Autosaved at 5:17 PM

Seb Ruiz 5a4bdde 63 63 border-bottom:

64 64 padding: 10px;

65 65 }

Changing classification settings

i Only Crucible Admin users can edit defect classifications.

To change the default classifications:

1. In the Admin area, click **Crucible** under 'Global Settings'.
2. Click **Edit Defect Classifications**.
3. You can add and remove classifications, and add fields to, or remove fields from, classifications.

Any changes made to defect classifications will only apply to reviews created after the change is saved.

Screenshot: Editing defect classifications in Crucible

Defect Classifications

Note: Changes will only apply to new reviews
Current defect classification version is 9.

Ranking

Name

Ranking

Values

Major

remove

Minor

remove

Add field

Remove this Classification

Classification

Name

Classification

Values

Wrong

remove

Can be improved

remove

Risk-prone

remove

Security hole

remove

Over engineered

remove

Code duplication

remove

Add field

Remove this Classification

Add a new Classification

Save changes

Cancel

Default Crucible classifications

There are two default defect classifications that are preset in Crucible: Ranking and Classification. These settings (and their sub-categories) can be edited or removed; other custom classifications can be added.

Ranking

Crucible users can rank a defect as **Major** or **Minor**, indicating the importance of the defect.

Classification

This setting helps to provide more detail about the defect. This classification can be set to one of the options described in the following table:

Value	Description
Missing	The defect applies to code or information that is missing (absent).
Extra (superfluous)	The defect applies to code or information that should be removed.
Ambiguous	The defect applies to code or information that is not clear or easy to understand.
Inconsistent	The defect applies to code or information that is applied in several different ways.
Improvement desirable	The defect applies to code or information that needs to be revised.
Not conforming to standards	The defect applies to code or information that breaks established conventions.

Risk-prone	The defect applies to code or information that takes unacceptable risks.
Factually incorrect	The defect applies to code or information that is wrong.
Not implementable	The defect applies to code or information that may be impossible to create.
Editorial	The defect applies to code or information where the classification as a defect may be subject to personal opinion.

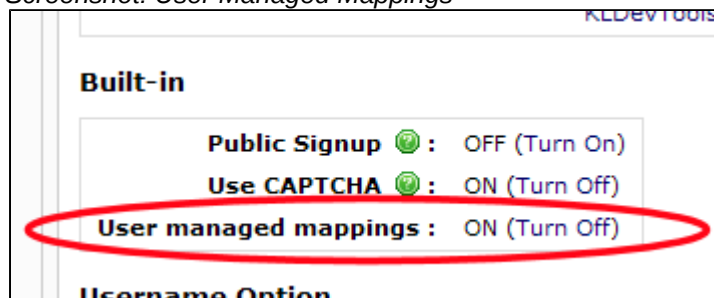
Configuring User Managed Mappings

In Crucible, Administrators can control whether users can use the [Author Mapping](#) setting to map their own Crucible usernames to repository committer accounts or not. By default, the setting allows users to set their own mappings.

If you wish to lock down the mappings for security or audit reasons, this setting lets you restrict all management of mappings to Crucible administrators only.

To do this, click **Administration** in the footer of the Crucible interface and then **Authentication** (under 'Security Settings') in the left navigation bar. You can set User Managed Mappings either **On** or **Off**. The setting is applied immediately.

Screenshot: User Managed Mappings



Enabling Access Logging in Crucible

To enable access logging in FishEye 3.0 and later:

1. Stop Fisheye/Crucible,
2. Create the file <FishEye install directory>/content/WEB-INF/jetty-web.xml with the following content:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD
Configure//EN" "http://www.eclipse.org/jetty/configure.dtd" >
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="handler">
    <New
class="com.cenqua.fisheye.web.jetty.FishEyeRequestLogHandler">
      <Set name="requestLog">
        <New id="RequestLogImpl"
class="org.eclipse.jetty.server.NCSARequestLog">
          <Arg><SystemProperty name="jetty.logs"
default="./var/log/" />fisheye-access-yyyy_mm_dd.log</Arg>
          <Set name="retainDays">90</Set>
          <Set name="append">true</Set>
          <Set name="extended">false</Set>
          <Set name="LogTimeZone">GMT</Set>
        </New>
      </Set>
    </New>
  </Set>
</Configure>
```

3. Restart Fisheye/Crucible.

This will create an access log in `<FishEye install directory>/var/log/fisheye-access-yyyy_mm_dd.log` format (e.g. `fisheye-access-2010_03_17.log`). If you want to change the path to your `FISHEYE_INST` directory, change the `default="./var/log/"` to the path to the log folder in `FISHEYE_INST`.

The log directory must exist

If the path to the log directory given by the `default` attribute of the `SystemProperty` tag (defined in the line 10 in the `jetty-web.xml` above) does not exist, then FishEye will fail to start and will not log any error message.

The path given in the example below is correct when `FISHEYE_INST` and the `<FishEye install directory>` are the same directory, otherwise please use the absolute path of your `FISHEYE_INST/var/log` directory.

Log format

The logs are written in [NCSA format](#):

```

172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=TestRepo&ou
tputtype=image HTTP/1.1" 200 256
172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=npanday&out
puttype=image HTTP/1.1" 200 177
172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=jutils&outp
uttype=image HTTP/1.1" 200 775
172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=P4&outputty
pe=image HTTP/1.1" 200 177
172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=Rails&outpu
ttype=image HTTP/1.1" 200 1311
172.20.5.186 - - [17/Mar/2010:22:50:22 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=FE-2363&out
puttype=image HTTP/1.1" 200 128

```

Please refer to the [Jetty documentation](#) for more configuration options.

Compatibility

If you are using an earlier version of FishEye than FishEye 2.7.8, replace `com.cenqua.fisheye.web.jetty.FishEyeRequestLogHandler` by `org.mortbay.jetty.handler.RequestLogHandler`.

FishEyeRequestLogHandler was added in 2.7.8 to fix an issue where the user credentials would not be added to the NCSA log: [FE-3040](#).

Linking Crucible to JIRA

[JIRA](#) is Atlassian's issue tracking and project management application.

See [JIRA integration in Crucible](#) for a description of all the integrations you get when Crucible is linked with JIRA.

This page describes how to integrate JIRA with Crucible. If you're linking Crucible to an Atlassian Cloud JIRA instance, please see [Linking to server applications from your cloud instance](#).

When Crucible was first installed, JIRA integration may have been configured using the setup wizard, which configures the JIRA connection automatically for user management. See [Configuring JIRA integration in the Setup Wizard](#).

However, JIRA integration with Crucible can be configured at any time after installation, as described on this page.

Initial configuration in JIRA

Configure the following setting in each instance of JIRA that you wish to link to Crucible:

- [Allow remote API access](#)

Consider also the following, to make full use of the integration between Crucible and JIRA:

- [Enable subtasks](#)
- [Allow unassigned issues](#)

On this page:

- [Initial configuration in JIRA](#)
- [Linking Crucible with JIRA instances](#)
- [Configuring an application link with JIRA](#)
- [Connecting to JIRA for user management](#)
- [Inline issue creation](#)

Related pages:

- [JIRA integration in Crucible](#)
- [User management limitations and recommendations](#)

Linking Crucible with JIRA instances

You can integrate Crucible with one or more instances of JIRA from the Crucible administration area. There are two parts to integrating with a JIRA instance:

- Setting up an [application link](#) between JIRA and Crucible, described below, for sharing information and facilitating integration features.
- Linking Crucible with JIRA for [delegating user and group management](#) to your JIRA server. You would only do this with one instance of JIRA.

Configuring an application link with JIRA

This section describes how to create a two-way applications link between Crucible and JIRA.

To create a new application link between Crucible and JIRA:

1. Go to your Crucible administration screen and click **Application Links** (under 'Global Settings').
2. Enter the URL for the JIRA instance you want to link to and click **Create new link**.
3. Complete the application link wizard to connect Crucible to your JIRA server. You *must* make use of the automatic link-back from JIRA to Crucible to get full integration (you'll need the JIRA system administrator global permission for that).

For FishEye 3.2 and later, creating a new application link now uses OAuth by default and enables both 3-legged OAuth (3LO) and 2-legged OAuth (2LO).

To update an existing application link with JIRA:

When you update an older application link to use OAuth, 3-legged authentication is applied by default. However, you'll need to explicitly enable 2-legged authentication in order to see Crucible information in the JIRA Development panel (when integrated with JIRA 6.2 or later). Enable 2LO for the application link as follows:

1. Go to the Crucible admin area and click **Add-ons > Application Links**.
2. Click **Edit** for the app link with JIRA.
3. For both **Outgoing Authentication** and **Incoming Authentication**:
 - a. Click **OAuth**
 - b. Check **Allow 2-legged OAuth**.
 - c. Click **Update**.

The application link update process will involve you logging into JIRA for a short time to configure the JIRA end of the link, before returning you to the Crucible.

Connecting to JIRA for user management

This option is available only with **JIRA 4.3 and later**.

To manage your Crucible users using JIRA, configure a connection in JIRA and then set up the user directory in Crucible. Both steps are described below.

To configure a new connection for user management in JIRA:

1. Log in to JIRA as a user with the 'JIRA System Administrators' global permission.
2. Click **Administration** on the JIRA top navigation bar.
 - 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.

To set up the JIRA user directory in Crucible:

Go to your Crucible administration screen.

1. Click **Authentication** (under 'Security Settings').
2. 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.
3. 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: <pre>http://www.example.com:8080/jira/ http://jira.example.com</pre>
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.

4. 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.
5. Click **Save**.

Inline issue creation

Inline issue creation allows a user to create a JIRA issue from a review comment. The user must have the 'Comment' [permission](#) in Crucible to see the **Create Issue** link in the comment. See [Creating JIRA issues from](#)

[the review](#).

This requires that Crucible is integrated with JIRA 5.0, or later, and is disabled if Crucible is integrated with an earlier version of JIRA.

When creating the issue, the Create Issue dialog only displays required fields for the selected issue type.

- A JIRA administrator can configure other fields to be required (and so displayed) from within JIRA. See [Specifying Field Behaviour](#).
- Not all field types are supported. In particular, unbundled custom field types are not supported. See [Supported fields for inline issue creation](#).
- If any required field for the selected issue type is not supported, Crucible displays an error, with a link to create the issue directly in JIRA. This link will pre-populate the project, issue type, summary and description fields, but will not create a link from the comment to the issue.

In the new issue in JIRA, you see a link back to the comment in Crucible. However:

- Issue links may be disabled in JIRA.
- The JIRA login that Crucible uses needs permission to create links.
- The FishEye plugin (available from the [Atlassian Marketplace](#)) must be installed in JIRA for permission checking on the display of issue links in JIRA.

You can disable the Inline Issue Creation plugin in Crucible to restore the earlier behaviour.

Having trouble integrating your Atlassian products with Application Links?

We've developed a [guide to troubleshooting Application Links](#), to help you out. Take a look at it if you need help getting around any errors or roadblocks.

User management limitations and recommendations

This page describes the optimal configurations and limitations that apply when you are connecting Crucible to JIRA for user management.

On this page:

- [Recommendations for connecting to JIRA for user management](#)
 - [Single Sign-On Across Multiple Applications is Not Supported](#)
 - [Custom Application Connectors are Not Supported](#)
 - [Custom Directories are Not Supported](#)
 - [Load on JIRA instance](#)
 - [JIRA OnDemand not supported](#)
 - [Recommendations](#)
- [Related pages](#)

Recommendations for connecting to JIRA for user management

Please consider the following limitations and recommendations when connecting to a JIRA server for user management.

Single Sign-On Across Multiple Applications is Not Supported

When you connect to JIRA for user management, you will not have single sign-on across the applications connected in this way. JIRA, when acting as a directory manager, does not support SSO.

Custom Application Connectors are Not Supported

JIRA, Confluence, FishEye, Crucible and Bamboo can connect to a JIRA server for user management. Custom application connectors will need to use the new REST API.

Custom Directories are Not Supported

Earlier versions of JIRA supported OSUser Providers. It was therefore possible write a special provider to obtain

user information from any external user directory. This is no longer the case.

Load on JIRA instance

If your JIRA instance is already under high load, then using it as a User Server will increase that load.

JIRA OnDemand not supported

You cannot use JIRA OnDemand to manage standalone users. OnDemand users and users within your self-hosted Atlassian applications need to be managed separately.

Recommendations

Your environment	Recommendation
If all the following are true: • If your JIRA instance is not under high load. • You want to share user and group management across just a few applications, such as one JIRA server and one Confluence server, or two JIRA servers. • You do not need single sign-on (SSO) between JIRA and Confluence, or between two JIRA servers. • You do not have custom application connectors. Or, if you do have them, you are happy to convert them to use the new REST API. • You are happy to shut down all your servers when you need to upgrade JIRA.	Your environment meets the optimal requirements for using JIRA for user management.
If one or more of the following are true: • If your JIRA instance is already under high load. • You want to share user and group management across more than 5 applications. • You need single sign-on (SSO) across multiple applications. • You have custom applications integrated via the Crowd SOAP API, and you cannot convert them to use the new REST API. • You are not happy to shut down all your servers when you need to upgrade JIRA.	We recommend that you install Atlassian Crowd for user management and SSO.

If you are considering creating a custom directory connector to define your own storage for users and groups...

Please see if one of the following solutions will work for you:

- If you have written a custom provider to support a specific LDAP schema, please check the supported LDAP schemas to see if you can use one of them instead.
- If you have written a custom provider to support nested groups, please consider enabling nested groups in the supported directory connectors instead.
- If you have written a custom provider to connect to your own database, please consider loading the data into the application's database instead.
- If you need to keep the custom directory connection, please consider whether [Atlassian Crowd](#) meets your requirements. See the documentation on [Creating a Custom Directory Connector](#).

Related pages

[Linking Crucible to JIRA](#)

Supported fields for inline issue creation

Inline issue creation allows a user to create a JIRA issue from a review comment. See [Creating JIRA issues from the review](#).

This requires JIRA 5.0, or later, and is disabled if Crucible is integrated with an earlier version of JIRA. See [Linking Crucible to JIRA](#).

When creating the issue, the Create Issue dialog only displays required fields for the selected issue type. However, a JIRA administrator can configure other fields to be required (and so displayed) from within JIRA. See [Specifying Field Behavior](#).

Note that:

- Unbundled custom field types in JIRA are not supported for inline issue creation.
- If any required field for the selected issue type is not supported, Crucible displays an error, with a link to create the issue directly in JIRA. This link will pre-populate the project, issue type, summary and description fields, but will not create a link from the comment to the issue.

The tables below list the supported JIRA field types.

Built-in fields

Field Name	Supported
Affects Version/s	YES
Assignee	N/A
Attachment	N/A
Comments	N/A
Component/s	YES
Description	YES
Due Date	NO
Environment	YES

Fix Version/s	YES
Issue Type	N/A
Labels	YES
Linked Issues	NO
Log Work	NO
Priority	YES
Reporter	N/A
Resolution	N/A
Security Level	YES
Summary	YES
Time Tracking	YES

Bundled custom field types

Field Name	Supported
Bug Import Id	N/A
Date Picker	NO
Free Text Field	YES
Hidden Job Switch	NO
Job Checkbox	NO
Multi Checkboxes	NO
Multi Select	NO
Number Field	YES
Radio Buttons	NO
Select List	NO
Text Field	YES
User Picker	NO
Cascading Select	NO
Date Time	NO
Group Picker	NO
Import Id	N/A

Labels	YES
Multi Group Picker	NO
Multi User Picker	NO
Project Picker	YES
Read-only Text Field	N/A
Single Version Picker	YES
URL Field	YES
Version Picker	YES

Using Crucible gadgets

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](#)

1. Add an application link from FishEye to JIRA
2. Add the Crucible gadget to JIRA's Gadget Directory
3. Add the Crucible gadget to a JIRA dashboard

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
Hassle Review Blockers	This gadget shows you who you are still waiting on; in other words, which reviewers haven't completed reviews. The URL for this gadget is: <pre>http://HOSTNAME:8060/rest/gadgets/1.0/g/com.atlassian.fecru.fecru-gadgets-plu</pre> 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 The URL for this gadget is: <pre>http://HOSTNAME:8060/rest/gadgets/1.0/g/com.atlassian.fecru.fecru-gadgets-plu</pre> 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 The URL for this gadget is: <pre>http://HOSTNAME:8060/rest/gadgets/1.0/g/com.atlassian.fecru.fecru-gadgets-plu</pre> Where HOSTNAME : 8060 is the hostname of your Crucible instance.
Review Coverage	This gadget shows content from the innovative Review Coverage report, letting you investigate how The URL for this gadget is as follows: <pre>http://HOSTNAME:PORT/CONTEXT/rest/gadgets/1.0/g/com.atlassian.crucible.plugins</pre> Where HOSTNAME : 8060 is the hostname of your Crucible instance. A known bug may be preventing this gadget to be used. Please track CRUC-6186 - 'Resource

1. Add an application link from FishEye to JIRA

Go to the [admin area](#) in FishEye and set up an application link from FishEye to JIRA.

When completing the wizard, leave the **Also create a link** checkbox selected, and 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.

Linking to another application

Application Links (sometimes called "AppLinks") is a bundled plugin that allows you to link Atlassian applications to each other. Linking two applications allows you to share information and access one application's functions and resources from within the other.

1. Click the 'cog' menu in the Crucible header, and choose **Administration**. You need to be logged in as an [administrator](#) to see this.
2. Choose **Application Links** in the left-hand panel. The Application Links configuration page displays links that have already been set up.
3. In the **Application URL** box, supply the URL of the application you want to link to and then select **Create new link**.
4. Use the wizard to finish configuring the link. If the application you are linking to does not have the Application Links plugin, you must supply additional information to set up a link with OAuth authentication.

When you complete the wizard, the Application Links plugin will create the link between your applications using the most secure authentication method that is supported between the two application types. After the link has been set up, it will appear on the "Configure Application Links" page. You can use this page to change the configuration of application links to make them more secure or to change the link settings:

- To edit the settings of the application link (for example, to [change the authentication type](#) of the link), select **Edit**.
- If you've set up multiple links to the same type of application (for example, multiple JIRA servers), you can use the **Make Primary** link to specify which application is the default instance. See [Making a Primary Link for Links to the Same Application Type](#) for more information.
- After you've linked applications, you also connect the areas of those applications that contain information relating to your project or team (for example, you can connect a project's Confluence space with a JIRA project). These types of links are called [project links](#).

Impersonating and Non-Impersonating Authentication Types

Application links allow you to configure 'impersonating' and 'non-impersonating' authentication types:

- **Impersonating authentication types** make requests on behalf of the user who is currently logged in. People will see only the information that they have permission to see. This includes OAuth and Trusted Applications authentication, and should only be used when two servers share the same user base.
- **Non-impersonating authentication types** always use a pre-configured user, and not the logged-in user, when making a request. The server handling the request determines the level of access to use, and this is used for requests from all users. This includes basic HTTP authentication.

Having trouble integrating your Atlassian products with Application Links?

We've developed a [guide to troubleshooting Application Links](#), to help you out. Take a look at it if you need help getting around any errors or roadblocks.

Running Crucible as a Windows service

For Crucible 3.4 and later, 32-bit and 64-bit Windows installers are available. Each installer sets up the service wrapper, adds Crucible as a Windows service, and starts the service, automatically.

The installer is the recommended way to install Crucible as a service on Windows. See [Installing Crucible on Windows](#).

If you use the FishEye installer for Windows, you can edit JVM settings using the tool included with the installer. See [JVM system properties](#) for more information.

The content on this page describes the old, deprecated method of configuring Crucible as a Windows service, and is only retained here for informational purposes.

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](#)

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 <Crucible home directory> (that is, the directory into which Crucible was originally installed). Note, the resulting folder structure should be similar to <Crucible home directory>\wrapper or <Crucible home directory>\wrapper\bin, etc and NOT <Crucible home directory>\wrapper\wrapper or <Crucible 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 <Crucible 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 <Crucible home directory>\var\log\wrapper.log.

4. Set the FISHEYE_INST environment variable (and other Crucible-specific environment variables) in the <Crucible 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 <Crucible home directory>\wrapper\bin and run Fisheye-Install-NTService.bat. If you run into any problems starting the wrapper, you'll find its logs in <Crucible 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 Crucible 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 <Crucible home directory>\wrapper\conf\wrapper.conf file.

If you run into any problems starting the wrapper, you'll find its logs in <Crucible 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=false
wrapper.java.additional.7=-Dcom.sun.management.jmxremote.ssl=false
wrapper.java.additional.8=-Dcom.sun.management.jmxremote.authenticate=false
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.

How to change JVM settings when using FishEye installer

If you are using the new Crucible installer for Windows, **up to Crucible 3.4.3** you can edit JVM options using this tool:

<http://www.apache.org/dist/commons/daemon/binaries/windows/commons-daemon-1.0.15-bin-windows.zip>

To use it, you need to rename "prunmgr.exe" from the zip file to "**Atlassian FishEye.exe**" or "**Atlassian Crucible.exe**" depending on which installer you used. Then JVM options such as Xmx and XX:MaxPermSize can be edited on the **Java** tab.

For Crucible 3.4.4 onwards the tool is included in the installer. You can use it by going to Windows Start Menu > All Programs > Crucible > Configure Crucible.

Ensure to restart Crucible service after changing the JVM parameters.

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:\install\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.

Managing add-ons

An add-on is an installable component that supplements or enhances the functionality of Crucible in some way. For example, the [Automatic Review Creator](#) automatically creates a Crucible review each time a commit is made to a FishEye repository. Other add-ons are available for reviewing changes in Git or Subversion source code repositories, and for using Crucible features directly from the IntelliJ IDEA interface.

Crucible comes with many pre-installed add-ons (called system add-ons). You can install more add-ons, either by acquiring the add-on from the [Atlassian Marketplace](#) or by uploading it from your file system. This means that you can install add-ons that you have developed yourself. For information about developing your own add-ons for Crucible, see the [Crucible Developer documentation](#).

On this page:

- [About the Universal Plugin Manager \(UPM\)](#)
- [Administering Add-ons in Crucible](#)

You may notice that the terms 'add-on' and 'plugin' both appear in the Atlassian documentation and tools. While the terms are often used interchangeably, there is a difference. A plugin is a type of add-on that can be installed into an Atlassian host application. Plugins are what developers create with the Atlassian SDK. But there are other types of add-ons as well. For example, the JIRA client is an add-on that runs as a separate program rather than as a plugin to JIRA. This documentation uses the term 'add-on' most often.

About the Universal Plugin Manager (UPM)

You administer add-ons for Crucible using the Universal Plugin Manager (UPM). The UPM is itself an add-on that exposes add-on administration pages in the Crucible Administration Console. UPM works across Atlassian applications, providing a consistent interface for administering add-ons in Crucible, Confluence, Fisheye, JIRA, Stash and Bamboo.

UPM comes pre-installed in recent versions of all Atlassian applications, so you do not normally need to install it yourself. However, like other add-ons, the UPM software is subject to regular software updates. Before administering add-ons in Crucible, therefore, you should [verify your version](#) of the UPM and update it if needed.

Administering Add-ons in Crucible

You can update the UPM, or any add-on, from the UPM's own add-on administration pages. Additionally, you can perform these tasks from the UPM administration pages:

- Install or remove add-ons
- Configure add-on settings
- Discover and install new add-ons from the [Atlassian Marketplace](#)
- Enable or disable add-ons and their component modules

To manage add-ons, go to the Crucible admin area, and click **Manage Add-ons**, under 'System Settings'.

For information on performing these add-on administration tasks, see the [Universal Plugin Manager documentation](#).

Crucible releases

The following pages can be found in the [latest documentation for Crucible](#):

- the [Crucible upgrade guide](#)
- the [Crucible security advisories](#)
- the [End of support announcements for Crucible](#)
- the full release notes for every Crucible release.



You can get automated notifications about Crucible releases by subscribing to the [Atlassian dev tools blog](#).

The change logs for the releases (linked below) have details of the related bug-fix releases.

Crucible 3.9

28 July 2015

- New Review Activity gives easy access to review comments
- Several performance improvements
- Oracle 12c is now supported
- Git 2.5.0 and Mercurial 3.5 are now supported
- Java 7 support is removed
- Internet Explorer 9 support is removed

Read more in the [Crucible 3.9 release notes and changelog](#).

See also the [Crucible upgrade guide](#).

Crucible 3.8

28 April 2015

- Support for more SCM diff formats by the patch parsers
- New database index for the Review Dashboard and Review Search
- Improved Git indexing time for new branches

Read more in the [Crucible 3.8 release notes and changelog](#).

See also the [Crucible upgrade guide](#).

Crucible 3.7

27 January 2015

- Automatic updates for branch reviews
- Create branch reviews from the activity stream
- Git 2.x support
- Mercurial 3.x support since Crucible 3.6.3
- Internet Explorer 9 support is [deprecated](#)

Read more in the [Crucible 3.7 release notes and changelog](#).

See also the [Crucible upgrade guide](#).

Crucible 3.6

28 October 2014

- Branch reviews
- Actions menu
- Crucible analytics
- Performance improvements for diff calculations and the user listing page
- SSL cipher suite configuration change

Read more in the [Crucible 3.6 release notes and changelog](#).

See also the [Crucible upgrade guide](#).

Crucible 3.5

22 July 2014

- ADG improvements to review comments
- Administration REST APIs – /projects and /permission-schemes/
- No config needed for Stash repository update events
- New passivation mechanism to optimize Java VM heap usage
- Java 8 is now supported
- JIRA versions earlier than 5.0 are no longer supported

Read more in the [Crucible 3.5 release notes and changelog](#).

See the [Crucible upgrade guide](#).

Crucible 3.4

15 April 2014

- Importing Git repositories from Atlassian Stash
- Git indexing performance gains
- Review actions dialog
- Administration REST APIs – /repositories/
- Automatic fullscreen mode for reviews
- Header stalking behaviour
- Installers for 64-bit and 32-bit Windows

Read more in the [Crucible 3.4 release notes and changelog](#).

See the [Crucible upgrade guide](#).

Crucible 3.3

11 February 2014

- Improved review subheader
- Administration REST APIs
- Subversion 1.8 is now supported
- Microsoft Internet Explorer 11 is now supported

Read more in the [Crucible 3.3 release notes and changelog](#).

See the [Crucible upgrade guide](#).

Crucible 3.2

27 November 2013

- Quick Search filters
- User data moved to the SQL database
- Improved protection against XSRF
- Internally managed Git repositories no longer supported by FishEye 3.2
- Microsoft SQL Server 2012 is now supported ([support for SQL Server 2005 is deprecated](#))
- Microsoft Internet Explorer 10 is now supported ([support for IE 8 is deprecated](#))
- MySQL 5.0 [is deprecated](#)
- PostgreSQL 8.2 [is deprecated](#)
- FishEye communication with JIRA versions older than 5.0 is [no longer supported](#)
- More in the [release notes and changelog](#)

See the [Crucible upgrade guide](#).

Crucible 3.1

27 August 2013

- Dashboard improvements
- QuickNav and QuickSearch improvements
- New JIRA issue dialog
- Transition JIRA issues from within Crucible
- Small improvements: native SVN 1.7, OpenJDK
- More in the [release notes and changelog](#)

See the [Crucible upgrade guide](#).

Crucible 3.0

30 May 2013

- Redesigned user experience
- Review CLI tool
- Iterative pre-commit reviews
- Platform upgrades: Jetty 8, Infinity 3 DB
- Optimised indexing for new SVN repositories
- Small improvements
- More in the [release notes and changelog](#)

See the [Crucible upgrade guide](#).

Crucible 2.10

15 January 2013

- Inline issue creation
- Repository management REST API
- Repository review creation limit
- More in the [release notes](#)
- [Crucible 2.10 upgrade guide](#)

▼ [Crucible older release notes \(click to expand\)](#)

Crucible 2.9

14 November 2012

- Simpler JIRA integration

- Better performance of the Reviews tab in JIRA
- Faster review creation for large teams
- More in the [release notes](#)
- [Crucible 2.9 upgrade guide](#)

Crucible 2.8

15 August 2012

- Mentions
- Shares
- Improved performance for the projects listing
- Support for Subversion 1.7
- End of life announcements
- More in the [release notes](#)
- [Crucible 2.8 Upgrade Guide](#)

Crucible 2.7

7 September 2011

- JIRA Transitions in Crucible
- Review Reminders
- Small Improvements
- More in the [release notes](#)
- [Crucible 2.7 Upgrade Guide](#)

Crucible 2.6

6 June 2011

- New Quick Search
- HTML Emails for Reviews
- Dashboard and Navigation Improvements
- SQL Server Support
- Oracle Support
- Review Creation without Metadata Changes
- Improved Patch Anchoring
- More in the [release notes](#)
- [Crucible 2.6 Upgrade Guide](#)

Crucible 2.5

8 February 2011

- Oracle (Beta)
- Redesigned Activity Stream
- Improved Header
- Comment Notification Batching
- More in the [release notes](#)
- [Crucible 2.5 Upgrade Guide](#)

Crucible 2.4

20 October 2010

- Easier Application Linking
- Native Repository Access

- Starter Licenses
- Adding Changesets to Reviews Simplified
- User Interface Improvements
- Snippets Tweaks
- More in the [release notes](#)
- [Crucible 2.4 Upgrade Guide](#)

Crucible 2.3

26 May 2010

- Snippet Reviews
- Changeset Discussions
- Mercurial SCM Alpha
- Review Coverage report
- Revamped Installation Process
- Gadgets
- More in the [release notes](#)
- [Crucible 2.3 Upgrade Guide](#)

Crucible 2.2

18 February 2010

- Smart Pre-Commit (Patch) Support
- 'No Moderator' Reviews
- Wizard-Like Review Creation
- Integrated Timetracking Between Crucible and JIRA
- Edit Mode for Reviews
- More in the [release notes](#)
- [Crucible 2.2 Upgrade Guide](#)

Crucible 2.1

12 November 2009

- Wiki Markup Rendering
- Progress Tracking
- Usability and Productivity Updates
- Streamlined JIRA Integration
- Review Time Tracking
- Review History Dialog
- "Blockers" Reports
- Threaded Comments
- Plugin Developer Tools
- More in the [release notes](#)
- [Crucible 2.1 Upgrade Guide](#)

Crucible 2.0

30 June 2009

- Support for iterative reviews
- New User Interface
- Indicators for read/unread comments
- Enhanced JIRA integration
- More in the [release notes](#)
- [Crucible 2.0 Upgrade Guide](#)

Crucible 1.6

23 September 2008

- Support for non-FishEye repositories
- Confluence page reviews
- Shared file system repositories
- Enhanced pre-commit reviews & image support
- Multiple admin users
- Expanded API
- More in the [release notes](#)
- [Crucible 1.6 Upgrade Guide](#)

Crucible 1.5

14 April 2008

- Project Dashboard
- Filtered comments & defects search, with statistical summary
- Customisable email templates
- Improvements to Crucible Plugin API beta
- More in the [release notes](#)
- [Crucible 1.5 Upgrade Guide](#)

Crucible 1.2

5 Dec 2007

- Reviews grouped into projects
- Customisable permission schemes
- Plugin API
- Enhancements to user management
- JIRA integration
- Crucible 1.2 includes FishEye 1.4
- More in the [release notes](#)
- [Crucible 1.2 Upgrade Guide](#)

Crucible 1.1

18 September 2007

- Pre-commit review (patch review)
- Review participants can keep track of their progress through a review by marking each file as "done"
- Side-by-side diff mode within the Review display
- Syntax highlighting when displaying a diff
- More in the [release notes](#)
- [Crucible 1.1 Upgrade Guide](#)

[Crucible 1.0.x Changelog](#)

▼ [Security advisories \(click to expand\)](#)

Security advisories

- [Crucible Security Advisory 2015-01-21](#)
- [FishEye and Crucible Security Advisory 2014-05-21](#)
- [FishEye and Crucible Security Advisory 2014-02-26](#)
- [FishEye and Crucible Security Advisory 2013-07-16](#)
- [FishEye and Crucible Security Advisory 2012-08-21](#)

- [FishEye and Crucible Security Advisory 2012-05-17](#)
- [FishEye and Crucible Security Advisory 2012-01-31](#)
- [FishEye and Crucible Security Advisory 2011-11-22](#)
- [FishEye and Crucible Security Advisory 2011-05-16](#)
- [FishEye and Crucible Security Advisory 2011-01-12](#)
- [Crucible Security Advisory 2010-06-16](#)
- [Crucible Security Advisory 2010-05-04](#)

End of Support Announcements for Crucible

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

End of support matrix for upcoming versions of Crucible

Platform	Announcement date	Crucible end of support
Internet Explorer 9	27 January 2015	As of April 2015
Java 7	28 October 2014	As of April 2015

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 new versions of databases, web browsers etc. are released, the cost of supporting multiple platforms grows exponentially, making it harder to provide the level of support our customers have come to expect from us. Therefore, we no longer support platform versions marked as end-of-life by the vendor, or very old versions that are no longer widely used.

End of support announcements

- [Deprecated Crucible support for Internet Explorer 9 \(announced 27 January 2015\)](#)
- [Deprecated Crucible support for Java 7 \(announced 28 October 2014\)](#)
- [Deprecated Crucible support for Java 6 \(announced 22 July 2014\)](#)
- [Deprecated Crucible support for MySQL 5.0 \(announced 27 November 2013\)](#)
- [Deprecated Crucible support for PostgreSQL 8.2 \(announced 27 November 2013\)](#)
- [Deprecated Crucible support for SQL Server 2005 \(announced 27 November 2013\)](#)
- [Deprecated Crucible support for Internet Explorer 8 \(announced 27 November 2013\)](#)
- [Deprecated Crucible 3.2 support for older versions of JIRA \(announced 27 August 2013\)](#)
- [Deprecated database support for Crucible \(announced 4 October 2011\)](#)
- [Deprecated web browsers for Crucible \(announced 21 March 2011\)](#)
- [Deprecated Java platforms for Crucible \(announced 21 March 2011\)](#)
- [Deprecated SCM repository support for Crucible \(announced 4 April 2011\)](#)

Deprecated Crucible support for Internet Explorer 9 (announced 27 January 2015)

Atlassian announces the deprecation of Crucible support for Microsoft Internet Explorer 9. Crucible will no longer support IE 9 after April 2015.

If you have questions or concerns regarding this announcement, please email eo1-announcement@atlassian.com.

Deprecated Crucible support for Java 7 (announced 28 October 2014)

Atlassian announces the deprecation of Crucible support for Java 7. Crucible will no longer support Java 7 after

April 2015.

If you have questions or concerns regarding this announcement, please email `eo1-announcement` at `atlassian dot com`.

Deprecated Crucible support for Java 6 (announced 22 July 2014)

Atlassian announces the deprecation of Crucible support for Java 6. We will no longer support Java 6 in Crucible 3.6. Crucible 3.6 is expected to be released later in 2014.

If you have questions or concerns regarding this announcement, please email `eo1-announcement` at `atlassian dot com`.

Deprecated Crucible support for MySQL 5.0 (announced 27 November 2013)

Atlassian announces the deprecation of Crucible support for MySQL 5.0. We will no longer support MySQL 5.0 in Crucible 3.3. Crucible 3.3 is expected to be released in the first half of 2014.

If you have questions or concerns regarding this announcement, please email `eo1-announcement` at `atlassian dot com`.

Deprecated Crucible support for PostgreSQL 8.2 (announced 27 November 2013)

Atlassian announces the deprecation of Crucible support for PostgreSQL 8.2. We will no longer support PostgreSQL 8.2 in Crucible 3.3. Crucible 3.3 is expected to be released in the first half of 2014.

If you have questions or concerns regarding this announcement, please email `eo1-announcement` at `atlassian dot com`.

Deprecated Crucible support for SQL Server 2005 (announced 27 November 2013)

Atlassian announces the deprecation of Crucible support for SQL Server 2005. We will no longer support SQL Server 2005 in Crucible 3.3. Crucible 3.3 is expected to be released in the first half of 2014.

If you have questions or concerns regarding this announcement, please email `eo1-announcement` at `atlassian dot com`.

Deprecated Crucible support for Internet Explorer 8 (announced 27 November 2013)

Atlassian announces the deprecation of Crucible support for IE8. We will no longer support IE8 in Crucible 3.3. Crucible 3.3 is expected to be released in the first half of 2014.

If you have questions or concerns regarding this announcement, please email `eo1-announcement` at `atlassian dot com`.

Deprecated Crucible 3.2 support for older versions of JIRA (announced 27 August 2013)

Atlassian announces the deprecation of support for Crucible communication with older versions of Atlassian JIRA. We will stop supporting older versions of JIRA as follows:

- From Crucible 3.2, support for Crucible to JIRA communication for versions of JIRA earlier than 5.0, will end. Please note that communication from JIRA to Crucible will continue to work as it currently does. Crucible 3.2 is expected to be released late in 2013.

If you have questions or concerns regarding this announcement, please email `eo1-announcement` at `atlassian dot com`.

Deprecated database support for Crucible (announced 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 `eo1-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 (announced 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 `eo1-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 (announced 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 eo1-announcement@atlassian.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 Crucible for bugs affecting Java Platform 5 (JDK/JRE 1.5).

Deprecated SCM repository support for Crucible (announced 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 eo1-announcement@atlassian.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 after 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 features.
- 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 **has 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](#).

Crucible upgrade guide

This page describes how to upgrade to a new version of Crucible. We *strongly recommend* that you update Crucible by performing the steps below.

Note that:

- This update process does *not* perform an in-place upgrade, but installs the new version of Crucible into a fresh installation directory. The new Crucible uses your existing Crucible home directory and external database.
- For production environments we recommend that you test the Crucible update on a QA server before deploying to production.

If you want to migrate Crucible, as well as upgrade, see [Migrating and Upgrading FishEye/Crucible](#).

1. Review the upgrade notes

There are [specific upgrade notes](#) further down this page for each version of Crucible.

You should read the relevant sections for each version between your current version of Crucible and the version you are upgrading to.

2. Backup your Crucible data

Back up your **entire** Crucible instance (see [Backing up and restoring Crucible data](#)):

- If you are backing up your Crucible instance via the Admin interface, tick all of the 'Include' checkboxes (e.g. plugins, templates, uploads, SQL database, etc).
- If you are backing up your Crucible instance using the command-line interface, do not use any [exclusion options](#).

3. Stop Crucible

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

```
bin/stop.sh
```

4. Install Crucible

This update process does *not* perform an in-place upgrade, but installs the new version of Crucible into a fresh installation directory. The new Crucible uses your existing Crucible home directory and external database.

Download the Crucible [zip file](#). On Windows, download the [installer](#). See [Installing Crucible on Linux and Mac](#) or [Installing Crucible on Windows](#) for detailed installation instructions.

Your upgrade procedure depends on whether you are using a [FISHEYE_INST](#) directory (i.e. "FishEye instance" directory).

- The [FISHEYE_INST](#) directory is the FishEye data directory (*not* the installation directory) and has a location defined by the [FISHEYE_INST](#) environment variable. It is used to keep the FishEye data completely separate from the FishEye/Crucible application files. We recommend that you configure FishEye/Crucible to use a [FISHEYE_INST](#) directory for production instances. Read more about [FISHEYE_INST](#) in [Installing FishEye on Windows](#) or [Installing FishEye on Linux and Mac](#).
- The <FishEye home directory> is the location of the FishEye/Crucible application files.

Method 1: Using a [FISHEYE_INST](#) directory

▼ [Click here to expand...](#)

If you have FishEye/Crucible configured to use a [FISHEYE_INST](#) directory, then follow the instructions below. This is the recommended scenario for production installations.

1. Shut down your existing FishEye/Crucible server, using `bin\stop.bat` or `bin\stop.sh` from

- the <FishEye home directory>.
2. Make a backup of your FISHEYE_INST directory.
 3. Download [FishEye](#) or [Crucible](#).
 4. Extract the new FishEye/Crucible version to a new directory.
 5. Leave your `FISHEYE_INST` environment variable set to its existing location. Both FishEye and Crucible use this variable.
 - Please be aware that jar files in the `FISHEYE_INST/lib` directory may conflict with those required for FishEye's normal operation. Jar files in this directory should be limited to those which provide functionality not provided by FishEye (e.g. database drivers).
 6. Start FishEye/Crucible from the new installation directory by running `bin/run.sh`. (Use `run.bat` on Windows.)
 7. Follow any version-specific instructions found in the [FishEye upgrade guide](#) or [Crucible upgrade guide](#).

Method 2: Without a FISHEYE_INST directory

▼ [Click here to expand...](#)

If you do not have FishEye/Crucible configured to use a `FISHEYE_INST` directory and do not want to set one up, then follow the instructions below. The <FishEye home directory> is the location of the existing FishEye/Crucible installation. Note that this is the typical scenario for evaluation installations, and is not recommended for production installations.

You will need to copy some files from your old FishEye/Crucible installation to your new one.

1. Download [FishEye](#) or [Crucible](#).
2. Extract the new FishEye/Crucible archive into a directory such as <New FishEye home directory>.
3. Shut down the old FishEye/Crucible server, using `bin\stop.bat` or `bin\stop.sh` from the <FishEye home directory>.
4. Copy <FishEye home directory>/config.xml to <New FishEye home directory>.
5. Delete the following directories from the <New FishEye home directory>/var directory:
 - <New FishEye home directory>/var/cache
 - <New FishEye home directory>/var/data
 - <New FishEye home directory>/var/log
6. Copy (or move) the following directories from <FishEye home directory>/var to <New FishEye home directory>/var:
 - <FishEye home directory>/var/cache
 - <FishEye home directory>/var/data
 - <FishEye home directory>/var/log

DO NOT include the following directories when you do that:

 - <FishEye home directory>/var/osgi-cache
 - <FishEye home directory>/var/plugins
 - <FishEye home directory>/var/tmp
7. Delete the <New FishEye home directory>/cache directory.
8. Copy (or move) the <FishEye home directory>/cache directory to <New FishEye home directory>/cache.
9. Start FishEye/Crucible from the new installation by running <New FishEye home directory>/bin/run.sh. (Use `run.bat` on Windows.)
10. Follow any version-specific instructions found in the [FishEye upgrade guide](#) or [Crucible upgrade guide](#).

Method 3: Without a FISHEYE_INST directory, but would like to set one up

▼ [Click here to expand...](#)

If you do not have FishEye/Crucible configured to use a `FISHEYE_INST` directory but would like to set one up, then follow the instructions below. You may wish to do this when reconfiguring an existing installation for a production environment.

The `FISHEYE_INST` directory is the FishEye data directory, which has a location defined by the `FISHEYE_INST` environment variable, and which should be completely separate from the <FishEye home directory>. The <FishEye home directory> is the location of the existing FishEye/Crucible installation.

1. Download [FishEye](#) or [Crucible](#).
2. Shut down the existing FishEye/Crucible server, using `bin\stop.bat` or `bin\stop.sh` from the `<FishEye home directory>`.
3. Set up the `FISHEYE_INST` [environment variable](#), then create the `FISHEYE_INST` directory on your file system.
4. Copy `<FishEye home directory>/config.xml` to the `FISHEYE_INST` directory.
5. Copy the `<FishEye home directory>/var` directory to the `FISHEYE_INST` directory.
6. Copy the `<FishEye home directory>/cache` directory to the `FISHEYE_INST` directory.
7. If it exists, copy the `<FishEye home directory>/data` directory to the `FISHEYE_INST` directory.
8. Extract the new FishEye/Crucible archive into a directory such as `<New FishEye home directory>`.
9. Start FishEye/Crucible from the new installation by running `<New FishEye home directory>/bin/run.sh`. (Use `run.bat` on Windows.)
 - If your configuration is not automatically picked up and you cannot see your existing repositories, check your **Administration > Sys-Info** page, where you will see information about the `<FishEye home directory>` and `FISHEYE_INST`. Check that your `FISHEYE_INST` is pointing to the right directory.
10. Follow any version-specific instructions found in the [FishEye upgrade guide](#) or [Crucible upgrade guide](#).

Method 4: Using the FishEye Installer for Windows

▼ [Click here to expand...](#)

Using a `FISHEYE_INST` directory (including when previously running as a Windows Service using the wrapper)

▼ [Click here to expand...](#)

If you have FishEye/Crucible configured to use a `FISHEYE_INST` directory, then follow the instructions below. This is the recommended scenario for production installations. This also applies when previously running FishEye as a Windows Service and the `FISHEYE_INST` location is defined within the wrapper.

1. Shut down your existing FishEye/Crucible server, using `bin\stop.bat` or `bin\stop.sh` from the `<FishEye home directory>`.
If you currently run FishEye or Crucible as a Windows service, see [Upgrading FishEye on Windows](#) for instructions on uninstalling the service.
2. Make a backup of your `FISHEYE_INST` directory.
3. Download [FishEye](#) or [Crucible](#).
4. Run the installer.
5. Select the option to use the **Custom Install**. Proceed with the install.
6. Select the folder where FishEye will be installed. In this step you can leave the default value selected. Proceed with the install.
7. Select the folder where your data will be stored, which in this case is the folder pointed to by your `FISHEYE_INST`. Ensure to select the correct folder in order to have all your data being read by FishEye. For example, if your `FISHEYE_INST` is currently set to `C:\Atlassian\fish_inst`, this is the folder you have to select. Proceed with the install until the end.
 - Please be aware that jar files in the `FISHEYE_INST/lib` directory may conflict with those required for FishEye's normal operation. Jar files in this directory should be limited to those which provide functionality not provided by FishEye (e.g. database drivers).
8. FishEye will be installed and running as a service by the end of the install process.
9. Follow any version-specific instructions found in the [FishEye upgrade guide](#) or [Crucible upgrade guide](#).

Without a `FISHEYE_INST` directory (including when previously running as a Windows Service using the wrapper)

▼ [Click here to expand...](#)

If you do not have FishEye/Crucible configured to use a `FISHEYE_INST` directory, the installer will create a directory for you and will create a `FISHEYE_INST` environment variable pointing to it. In the following instructions, `<FishEye home directory>` refers to the location of the previous FishEye/Crucible installation. This also applies when previously running FishEye as a Windows Service and there is no `FISHEYE_INST` defined.

1. Shut down your existing FishEye/Crucible server, using `bin\stop.bat` or `bin\stop.sh` from the <FishEye home directory>.
If you currently run FishEye or Crucible as a Windows service, see [Upgrading FishEye on Windows](#) for instructions on uninstalling the service.
2. Download [FishEye](#) or [Crucible](#).
3. Run the installer.
4. Select the option to use the **Default Install**. You may also use **Custom Install** if you want to select the directories where FishEye will be installed and where your data will be stored.
Proceed with the install.
5. FishEye will be installed and running as a service by the end of the install process.
6. Stop FishEye service (names as Atlassian FishEye).
7. Copy <FishEye home directory>/config.xml to <FISHEYE_INST>.
8. Delete the following directories from the <FISHEYE_INST>/var directory:
 - <FISHEYE_INST>/var/cache
 - <FISHEYE_INST>/var/data
 - <FISHEYE_INST>/var/log
9. Copy (or move) the following directories from <FishEye home directory>/var to <FISHEYE_INST>/var:
 - <FishEye home directory>/var/cache
 - <FishEye home directory>/var/data
 - <FishEye home directory>/var/log
 DO NOT include the following directories when you do that:
 - <FishEye home directory>/var/osgi-cache
 - <FishEye home directory>/var/plugins
 - <FishEye home directory>/var/tmp
10. Delete the <FISHEYE_INST>/cache directory.
11. Copy (or move) the <FishEye home directory>/cache directory to <FISHEYE_INST>/cache.
12. Start the Atlassian FishEye service.
13. Follow any version-specific instructions found in the [FishEye upgrade guide](#) or [Crucible upgrade guide](#).

5. Update any custom configurations

Once the new version of Crucible is installed, remember to update any custom configurations in the new version of Crucible, for example your SQL driver and your wrapper.config file.

If you are using MySQL, read about the [JDBC driver](#).

If you currently run Crucible as a Windows service and are installing the new version of Crucible in a new location, you need to uninstall and then reinstall Crucible as a Windows service. Please see [Upgrading FishEye on Windows](#) for instructions.

For Crucible 3.4 and later, on Windows, you can edit the Java VM properties using the tool included in the download. See [JVM system properties](#).

6. Start Crucible

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

```
bin/start.sh
```

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

Version-specific update notes

This section provides specific update notes for each version of Crucible. These notes supplement the primary update guide above.

You should read the relevant sections for each version between your current version of Crucible and the version you are upgrading to.

- [Crucible 3.9 upgrade notes](#)
- [Crucible 3.8 upgrade notes](#)
- [Crucible 3.7 upgrade notes](#)
- [Crucible 3.6 upgrade notes](#)
- [Crucible 3.5 upgrade notes](#)
- [Crucible 3.4 upgrade notes](#)
- [Crucible 3.3 upgrade notes](#)
- [Crucible 3.2 upgrade notes](#)
- [Crucible 3.1 upgrade notes](#)
- [Crucible 3.0 upgrade notes](#)

Crucible 3.9 upgrade notes

Please also see:

- the general [Upgrade steps](#) section above.
- the [Crucible 3.9 release notes](#).
- the Crucible [Supported platforms](#) page.

Lucene index upgrade task

A reindex of the Lucene index runs on startup after upgrading to Crucible 3.9. This is required because new fields have been added to the Lucene index to enable sorting of reviews by the Review column (



~~CRUC-6560~~ - Review dashboard can't be sorted by Review key **CLOSED**).

Reindexing takes about 10 minutes for 15000 reviews. During reindex, Crucible is usable but some reviews may not be visible.

Smart Commits now use the JIRA REST API

As of version 3.9.0, Crucible uses the [JIRA REST API](#) instead of the JIRA FishEye Plugin for JIRA Smart Commits. This change ensures that you will be able to use Smart Commits with future versions of JIRA. Note that using Smart Commits with previous versions of Crucible (earlier than 3.9.0) and JIRA 7 will cause an error notification via email.

Git clone changes

As of version 3.9.0, Crucible turns Git garbage collection off when cloning a repository (by adding the `gc . pruneExpire=never` option) to prevent unreferenced objects being removed from local clones. Also, when cloning a repository, `git config` is run on each repository during instance startup. This change bumps the Git cache version (`CACHE_VERSION`) to 22.

Supported platform upgrades

- Oracle 12c is now supported.
- Git 2.4.6 and Mercurial 3.4.2 are now supported.
- Support for Java 7 has been removed from Crucible 3.9, [as previously announced](#).
- Support for Internet Explorer 9 has been removed from Crucible 3.9, [as previously announced](#).

Known issues for Crucible 3.9

T	Key	Summary	Status
	CRUC-7321	Reviews in 'Waiting for Approval' State Cannot be Transitioned if Changed to a Permission Scheme without a Moderator	OPEN
	CRUC-7318	Dialogs content is not scrollable when CRAS is opened	OPEN
	CRUC-7316	long comment may result with StackOverflowError	OPEN
	CRUC-7315	Collapsed comment body is not properly aligned	OPEN
	CRUC-7302	Deleted binary file on review rendered as text	OPEN
	CRUC-7296	Review Snippet exceeds area for very long words	OPEN
	CRUC-7270	review summary html encoded too many times	OPEN
	CRUC-6487	ConstraintViolationException: could not insert: [com.cenqua.crucible.model.CommentReadStatus]	OPEN
	CRUC-7325	Edit Review Details Dialog Should Prevent Reload When Changes Have Been Made	OPEN

Authenticate to retrieve your issues

9 issues

Crucible 3.8 upgrade notes

Please also see:

- the general [Upgrade steps](#) section above.
- the [Crucible 3.8 release notes](#).
- the Crucible [Supported platforms](#) page.

New database index to improve Review Dashboard and Review Search pages load time

A database upgrade task is run on startup that adds a database index. This task should take no more than a few minutes, and should complete within seconds on most instances.

Improved Git indexing time for newly created branches

In order to allow faster Git indexing, a new field was added to the internal repository caches. The cache for each Git repository is automatically upgraded when the repository is started for the first time after upgrading. It is expected to take under a minute per repository, but may take slightly longer if repositories have thousands of branches.

Supported platform upgrades

- Java 7 is deprecated, [as previously announced](#).

Known issues for Crucible 3.8

T	Key	Summary	Status
	CRUC-5570	Crucible comment and general search potentially excludes valid results	OPEN
	CRUC-7315	Collapsed comment body is not properly aligned	OPEN
	CRUC-7307	"Create Review for All Commits" in JIRA Marks Added File as Modified in Review	OPEN
	CRUC-7287	Multiple Spaces after Project Key in Smart Commit causes reviewers not to be assigned	OPEN

	CRUC-7258	log4j cannot find log appenders for amazonaws	OPEN
	CRUC-7245	No reviews displayed when browsing Review Dashboard as not logged in user although counters suggest otherwise	OPEN
	CRUC-7239	Crucible Comment Search is Case Sensitive	OPEN
	CRUC-7237	Every user admin action leaves old url	OPEN
	CRUC-7229	Esc key cancels "Edit Details" when "Wiki Markup Tips" is open	OPEN
	CRUC-6866	Stats "Last Commit" field shows incorrect date (FishEye + Crucible)	OPEN
	CRUC-7313	Navigating to comments doesn't scroll the page to the comment when displayed "above"	OPEN

Authenticate to retrieve your issues

11 issues

Crucible 3.7 upgrade notes

Please also see:

- the general [Upgrade steps](#) section above.
- the [Crucible 3.7 release notes](#).
- the [Crucible Supported platforms](#) page.

Supported platform upgrades

- Java 7 support is deprecated – please see the [End of Support Announcements for Crucible](#).

Known issues for Crucible 3.7

T	Key	Summary	Status
	CRUC-7292	Adding changesets from branch is broken when specific privileges apply to repos	OPEN
	CRUC-7186	Error when trying to add the latest revision of a file to a Review if comment is present on previous revision	OPEN
	CRUC-7226	new comment shows previous comment preview	OPEN
	CRUC-7225	Browsers have problem processing big revisions	OPEN
	CRUC-7220	Revisions from closed heads are considered ancestors for files	OPEN
	CRUC-7212	Days to complete metrics doesn't handle reopened and in draft state reviews	OPEN
	CRUC-7209	Truncated error/warning message when updating a review	OPEN
	CRUC-7201	Inconsistent diff when using branch review during initial indexing	OPEN
	CRUC-7197	JIRA Blockers Report detect Abandoned Review as Open	OPEN
	CRUC-7196	crucible drops the rename in patch files	OPEN
	CRUC-7192	Reviews created for the first commit in a branch do not show diff information if created during indexing	OPEN
	CRUC-7183	Unable to Publish Draft Comments if they Contain a Draft Reply Using 'Post Drafts' Link	OPEN
	CRUC-7160	branch review auto update issue on review reopening	OPEN
	CRUC-7154	Horizontal scroll bar doesn't show up on code side of dual panel for Confluence	OPEN
	CRUC-7148	crucible.py not finding matching FishEye repository for Subversion	OPEN
	CRUC-7108	adding reviewers group conflicts with allowed groups	OPEN

	CRUC-7075	Crucible does not render Assignee field properly.	OPEN
	CRUC-6847	Remind REST API Issue	OPEN
	CRUC-2351	"Remove All Revisions from Review" link formatting is broken	OPEN
	CRUC-7112	Trying to create a branch review for a changeset that has had a branch removed recently might pick the removed branch and show a Internal Server Error	OPEN

Authenticate to retrieve your issues

20 issues

Crucible 3.6 upgrade notes

Please also see:

- the general [Upgrade steps](#) section above.
- the [Crucible 3.6 release notes](#).
- the Crucible [Supported platforms](#) page.

Supported platform upgrades

- Java 6 is not supported by Crucible 3.6, as previously [announced](#).
- Java 7 support is deprecated – please see the [End of Support Announcements for Crucible](#) .

SSLv3 support disabled by default

As of Crucible 3.6, SSLv3 support is disabled by default. This shouldn't affect normal operations in supported browsers. If you need to re-enable SSLv3 support, please consult [Configuring SSL cipher suites for Jetty](#).

Known issues for Crucible 3.6

T	Key	Summary	Status
	CRUC-7265	Better showing renames/moves in branch reviews	OPEN
	CRUC-7264	Files are not marked as added in CVS	OPEN
	CRUC-7071	Adding content via Branch Review doesn't set the reviews default repository	OPEN
	CRUC-6987	Using the API, you can transition a review to a state that's not normally in the workflow	OPEN
	CRUC-6980	Safari - Selecting a code on source page scroll the whole page	OPEN
	CRUC-6952	Review revision pre-load policy (loadFrX)	OPEN
	CRUC-6937	Migration to MySQL 5.7.4 failure	OPEN

Authenticate to retrieve your issues

7 issues

Crucible 3.5 upgrade notes

An upgrade from an earlier version of FishEye/Crucible to 3.5.0 may cause problems if you have

upgraded the Universal Plugin Manager Plugin to a newer version than is shipped with FishEye/Crucible 3.5.0.

The workaround for this is to remove the custom installed version of the Universal Plugin Manager Plugin.

After upgrading from 3.4.5 to 3.5.0, this error is printed in the web browser when you try to access some pages:

```
javax.servlet.jsp.JspException: javax.el.ELException:
java.lang.NullPointerException: couldn't locate WebResourceIntegration
service
```

Workaround:

- Stop the new FishEye instance;
- Remove your newer version of the Universal Plugin Manager Plugin at \$FISHEYE_INST/var/plugins/user/plugin.xxxxxx.atlassian-universal-plugin-manager-plugin*.jar;
- Start the new FishEye instance again.

Please also see:

- the general [Upgrade steps](#) section above.
- the [Crucible 3.5 release notes](#).
- the Crucible [Supported platforms](#) page.
- the [End of Support Announcements for Crucible](#).

Known issues for Crucible 3.5

T	Key	Summary	Status
	CRUC-7255	Commit messages with an issue key preceded by a dash are not visible in JIRA development panel, despite having perform job set	OPEN
	CRUC-7247	Commit messages with an issue key preceded by a dash character are not visible in the JIRA development status panel.	OPEN
	CRUC-6911	Suggested reviewers dialog gets confused for binary files	OPEN
	CRUC-6895	Layout becomes broken after comment is added	OPEN
	CRUC-6892	Strange behaviour when clicking objectives on the sidebar on Review page	OPEN
	CRUC-6876	Rest API doesn't show e-mail on users	OPEN
	CRUC-6875	Comment preview not cleared when adding new comments.	OPEN
	CRUC-6845	Searching in Firefox change focus	OPEN

Authenticate to retrieve your issues

8 issues

Crucible 3.4 upgrade notes

Please also see:

- the general [Upgrade steps](#) section above.
- the [Crucible 3.4 release notes](#).
- the Crucible [Supported platforms](#) page.
- the [End of Support Announcements for Crucible](#).

Windows installer

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

See [Installing Crucible on Windows](#) for detailed installation instructions.

Download the Crucible installer [here](#).

Crucible may now bind to a different IP address on Windows

Prior to Crucible 3.4, a bug in Crucible ([JIRA](#) [FE-4909](#) **CLOSED**) meant that Crucible may not have correctly bound to the IP address you configured.

This may have happened if you configured Crucible to bind to a single IP address on a network interface that has several IP addresses; Crucible may in fact have bound to a different IP address. For example, if you have an interface with the IP addresses 1.2.3.4 and 1.2.3.5, and you configured Crucible to use 1.2.3.5, it may have incorrectly bound to 1.2.3.4.

Now that the bug is fixed, Crucible 3.4, and later versions, will now correctly bind to the configured IP address, although this may now be different from the previously bound address.

v1 REST API resources deprecated

Note that the 'v1' REST API resources are deprecated and will be removed in a future release. See the [FishEye Crucible REST API](#).

Known issues for Crucible 3.4

T	Key	Summary	Status
	CRUC-6930	Multiline comment with Cyrillic symbols at the end of the line is displayed as single line comment	OPEN
	CRUC-6892	Strange behaviour when clicking objectives on the sidebar on Review page	OPEN
	CRUC-6841	Korean Characters On Review Comments Displayed As Question Mark When External Database (mySQL) Is Used	OPEN
	CRUC-6817	Abandoned reviews are included when applying a workflow transition for unreviewed changesets	OPEN
	CRUC-6811	Reloading webpage drops decoration	OPEN
	CRUC-6790	Make default metrics handling more robust	OPEN
	CRUC-6776	links to JIRA issues and Crucible reviews rendered incorrectly	OPEN
	CRUC-6773	Problem with finding text in firefox in FishEye and Crucible	OPEN

Authenticate to retrieve your issues

8 issues

Crucible 3.3 upgrade notes

Please also see the [Upgrade steps](#) section above.

As previously announced, the following platforms are no longer supported by Crucible 3.3:

- Internet Explorer 8

- MySQL 5.0
- PostgreSQL 8.2
- SQL Server 2005

Please read the [End of Support Announcements for Crucible](#).

Supported platform upgrades

- SVN 1.8 is supported by Crucible 3.3.
- Microsoft Internet Explorer 11 is supported by Crucible 3.3.

See the Crucible [Supported platforms](#).

Known issues for Crucible 3.3

T	Key	Summary	Status
	CRUC-7307	"Create Review for All Commits" in JIRA Marks Added File as Modified in Review	OPEN
	CRUC-6975	"User" in "Add Content to Review" Dialog Does Not Include Authors	OPEN
	CRUC-6968	Review keys in review title get wrongly shown on JIRA issues matching a substring of the key	OPEN
	CRUC-6790	Make default metrics handling more robust	OPEN
	CRUC-6745	Can't configure Mail Server with gmail account configured for 2-step authentication	OPEN
	CRUC-6741	Smart commit is not parsed if there is "#" before that.	OPEN
	CRUC-6730	REST API doesn't allow to create a review if the patch can't be anchored	OPEN
	CRUC-6720	Creating a review of a binary revision before SVN processing is complete renders binary file as text	OPEN
	CRUC-6703	Author field does nothing when trying to filter a SCMSource repo	OPEN
	CRUC-6702	Impossible to selecting a repo with a name greater than 80 chars when adding to review	OPEN

Authenticate to retrieve your issues

10 issues

Crucible 3.2 upgrade notes

Please also see the [Upgrade steps](#) section above, and read the [End of Support Announcements for Crucible](#) page.

Please note the following changes in Crucible 3.2:

REST endpoint change

For Crucible 3.2, the `RestReviewService.remindIncompleteReviewers()` (`/rest-service/reviews-v1/{reviewId}/remind`) end point was changed from accepting 'application/x-www-form-urlencoded' content type with 'message' and 'recipient' params to 'application/json' content type with 'message' and 'recipients' JSON fields. See <https://docs.atlassian.com/fisheye-crucible/latest/wadl/crucible.html#d2e1881>.

User data is moved from data0.bin to the SQL database

An upgrade task is run on startup that moves user data to the SQL database. We are doing this to mitigate the risk of data corruption or loss.

XSRF protection

FishEye/Crucible 3.2.0 includes protection against XSRF attacks.

If you have implemented a LightSCM plugin, and have used the SimpleConfigurationServlet base class provided in the scmutils library, you will need to modify your administration page so that it performs 'delete' operations using an HTTP POST, not a GET.

You can have FishEye convert your anchor tags to form POSTs at runtime by giving them the class "anchor-post". For example, the anchor:

```
<a class="anchor-post" href="#"
data-href="/fsscm?name=TEST&delete=true">Delete</a>
```

will be converted into a form which POSTs to /fsscm with form parameters name=TEST and delete=true.

Internally managed Git repositories no longer supported

As [previously announced](#), internally managed Git repositories are no longer supported by FishEye 3.2.

Please read the [migration guide](#) for information about options and procedures for migrating your internally managed Git repositories – note that we recommend that you upgrade to FishEye 3.2 before migrating any internally managed repositories.

Supported platform upgrades

- Communication with JIRA versions older than 5.0 is [no longer supported](#).
- Microsoft SQL Server 2012 is now supported ([support for SQL Server 2005 is deprecated](#)).
- Microsoft Internet Explorer 10 is now supported ([support for IE 8 is deprecated](#)).
- MySQL 5.0 [is deprecated](#).
- PostgreSQL 8.2 [is deprecated](#).
- The Atlassian AUI plugin has been upgraded to AUI 5.2.
- jQuery has been upgraded to 1.8.3.
- backbonejs has been upgraded to 1.0.

See the Crucible [Supported platforms](#).

Known issues for Crucible 3.2

T	Key	Summary	Status
	CRUC-6158	Review performance in IE 8&9 is pathetic	OPEN
	CRUC-6790	Make default metrics handling more robust	OPEN
	CRUC-6485	Indicate that the listing of reviewers is truncated in the Edit Details view	OPEN

Authenticate to retrieve your issues

3 issues

Crucible 3.1 upgrade notes

Please also see the [Upgrade steps](#) section above, and read the [End of Support Announcements for Crucible](#) page.

Please note the following changes in Crucible 3.1:

Crucible 3.1 Merge some per-repository Lucene indices into a global cross-repository Lucene index

Crucible 3.1 has greatly improved performance and scalability for QuickSearch and QuickNav. To achieve this, the per-repository 'METADATA' Lucene indices will be moved into a single global cross-repository Lucene index. This means Crucible is able to search across more repositories in less time because now only a single search index needs to be queried instead of the previous N. Merging these indices into the single cross-repository index can be refreshed in two ways:

1. **Recommended:** As an upgrade task that is run automatically when Crucible 3.1 is run for the first time.
2. As an offline process on a separate staging server.

During the automatic upgrade task, Crucible is fully usable and functional, although search results for files, commits and committers may be incomplete.

In our testing we have found that the automatic upgrade task took 4 hours to complete on a Crucible instance with 144 repositories of different kinds, with 58 GB of data in the FISHEYE_INST folder (excluding logs). We are confident that the automatic upgrade task is suitable for the majority of production Crucible installations. *It is worth repeating that the instance was fully functional (reviews, JIRA Integration, Activity Streams, Charts etc) apart from Quick Nav and Quick Search during this time.*

Nevertheless, where required, we provide [instructions for performing the reindex as an offline process](#) on a separate staging server.

Plugin Settings will be moved from the config.xml to the SQL database

As of Crucible 3.1.0, plugin settings which were previously stored in the <properties> element inside config.xml will be stored in the SQL database. This includes settings for any bundled plugins such as ApplicationLinks, the UniversalPluginManager etc, and any 3rd party plugins.

An upgrade task is run on startup which will first insert all the properties found in config.xml into a new table in the SQL database. Once successful, the properties are removed from config.xml.

As part of this change, the RepositoryOptions.setProperties (Map<String, String>properties) and RepositoryOptions.getProperties() methods have been removed from our API. If you are using a plugin which uses either of these methods, you will need to update the plugin to a version which uses the Spring component PluginSettingsFactory. Plugins can use this to access the migrated global and per-repository properties that were previously available via the old RepositoryOptions API.

Known issues for Crucible 3.1

T	Key	Summary	Status
	CRUC-6790	Make default metrics handling more robust	OPEN
	CRUC-6596	Resolving long content hashes from short content hashes in git patches goes over the whole contentHashDB for each hash in patch	OPEN

[Authenticate to retrieve your issues](#)

2 issues

Crucible 3.0 upgrade notes

Please also see the [Upgrade steps](#) section above.

Please note the following changes in Crucible 3.0:

Jetty 8

Crucible 3.0 now uses Jetty 8 as its web server and Java servlet container. This change should be completely transparent when you upgrade to Crucible 3.0. However, if you have customised either your `jetty-web.xml` file, or the `maxFormContentSize` [system property](#), you will need to update those in the new

version. See [Enabling Access Logging in FishEye](#) and this [FishEye KB page](#) for more information.

Infinity DB

Crucible 3.0 now uses the InfinityDB 3.0 database internally to provide improved performance for concurrent access to Crucible. This change is transparent to users in all respects.

Pipelined indexing

Crucible 3.0 introduces a new indexing approach that splits the repository indexing process into separate tasks that can be performed in a phased and concurrent way. Users will benefit from the way in which Crucible functionality, such as review creation, now becomes available as indexing progresses. This change is transparent to users in all other respects. See [Pipelined indexing](#).

Improved handling of user preferences with session cookies

Upgrading may result in some users losing their preferences.

SQL Server transaction isolation configuration

We recommend a configuration change for SQL Server to use snapshot mode for the transaction isolation level – see [Migrating to SQL Server](#). This change avoids occasional database deadlocks, and prevents performance warning messages in the logs and admin screens.

Known issues for Crucible 3.0

T	Key	Summary	Status
	CRUC-6521	File creation is not displayed as change	OPEN
	CRUC-7209	Truncated error/warning message when updating a review	OPEN
	CRUC-7091	Changing a project key doesn't update the review index	OPEN
	CRUC-6987	Using the API, you can transition a review to a state that's not normally in the workflow	OPEN
	CRUC-6790	Make default metrics handling more robust	OPEN
	CRUC-6596	Resolving long content hashes from short content hashes in git patches goes over the whole contentHashDB for each hash in patch	OPEN
	CRUC-6550	Code Review UI real estate misallocation	OPEN
	CRUC-6528	User still suggested for review after disabling for him Crucible	OPEN
	CRUC-6520	If the P4DIFF environment variable is set, the crucible.py review tool might not be able to get the patch properly	OPEN
	CRUC-6487	ConstraintViolationException: could not insert: [com.cenqua.crucible.model.CommentReadStatus]	OPEN
	CRUC-6470	Inline issue creation: when not logged in to Crucible the review comment icon is not fetched	OPEN
	CRUC-6447	Providing a FishEye licence does not allow repositories to be started until FishEye itself has been stopped and started	OPEN
	CRUC-5833	Problem creating FRXDO from frx: Index: 395	OPEN

Authenticate to retrieve your issues

13 issues

Checking for known issues and troubleshooting the Crucible upgrade

If something is not working correctly after you have completed the steps above to upgrade your Crucible installation, please check for known Crucible issues and try troubleshooting your upgrade as described below:

- **Check for known issues.** Sometimes we find out about a problem with the latest version of Crucible after we have released the software. In such cases we publish information about the known issues in the Crucible Knowledge Base. Please check the [Fisheye and Crucible Known Issues](#) in the Crucible Knowledge Base and follow the instructions to apply any necessary patches if necessary.
- **Did you encounter a problem during the Crucible upgrade?** Please refer to the guide to [troubleshooting upgrades](#) in the Crucible Knowledge Base.
- If you encounter a problem during the upgrade and cannot solve it, please create a [support ticket](#) and one of our support engineers will help you.

Upgrading from FishEye to Crucible

If you have been using FishEye and now want to move to Crucible, you can do this without losing your FishEye repositories.

You simply need to add a Crucible license to your existing FishEye installation, and then follow the initial configuration steps below.

You do not need to download Crucible from Atlassian.

To add your Crucible license key, go to the admin area and click **System Info** (under 'System Settings'). Click **Edit License** (under 'License') to enter your Crucible license key. You can view your license key [here](#). The Crucible functionality will be instantly unlocked.

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

On this page:

- [Initial Crucible configuration](#)

Initial Crucible configuration

1. You can access FishEye/Crucible immediately by going to <http://HOSTNAME:8060/> in a browser.
2. If you do not already have user accounts configured, you will need to do this via the Administration screens or by configuring FishEye/Crucible to use external authentication.
To add users:
 - Open the [FishEye Administration screens](#) at <http://HOSTNAME:8060/admin/>.
 - Click **Users** under 'User Settings' in the admin area.
Read more details about the different ways of [creating users](#).
3. Crucible can email each review participant on a range of changes. Each user can then set up their own preferences. This is described in the [User Profile guide](#). First, you must [set up the SMTP Server](#).

Crucible 3.9 release notes

4 August 2015

Try it for FREE ➞

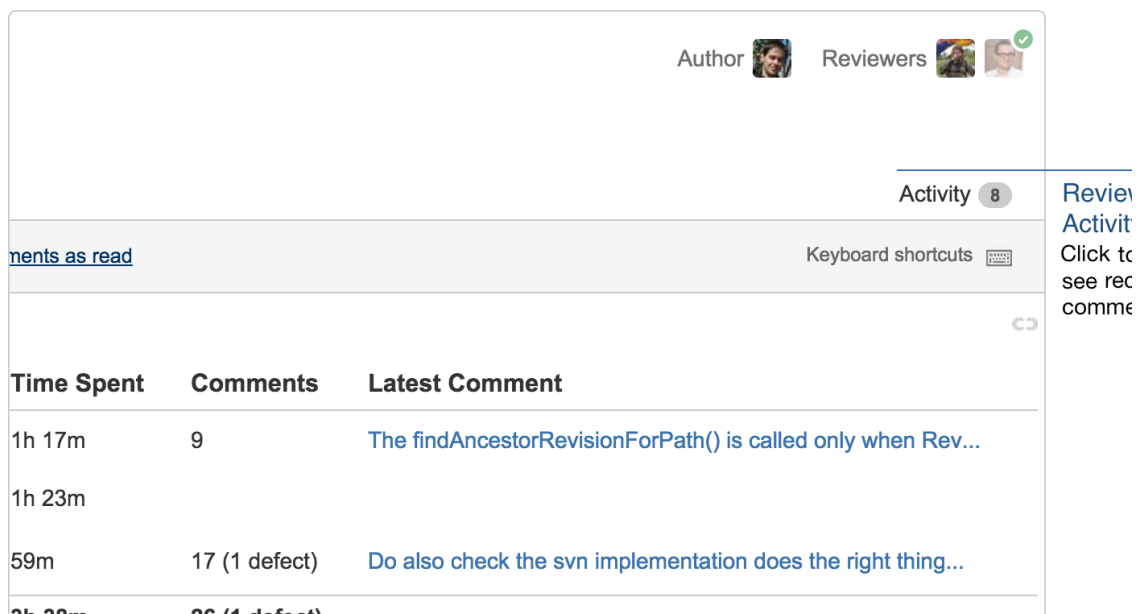
Today's release of [Crucible 3.9](#) brings a new way to see comment activity on reviews, and a raft of performance improvements.

If you are updating an earlier version of Crucible, please read the [Crucible upgrade guide](#) and the [End of Support Announcements for Crucible](#). The Crucible 3.9 changelog is [below](#).

Review activity

When you go to a Crucible review, you want to quickly see the recent comments from your team. That's just what the Review Activity shows you, so now it's easy to catch up on a review, or to find outdated or hidden comments.

Click **Activity** at the top right of the review (or use the keyboard shortcut 'shift' + 'J'):



Author  Reviewers  

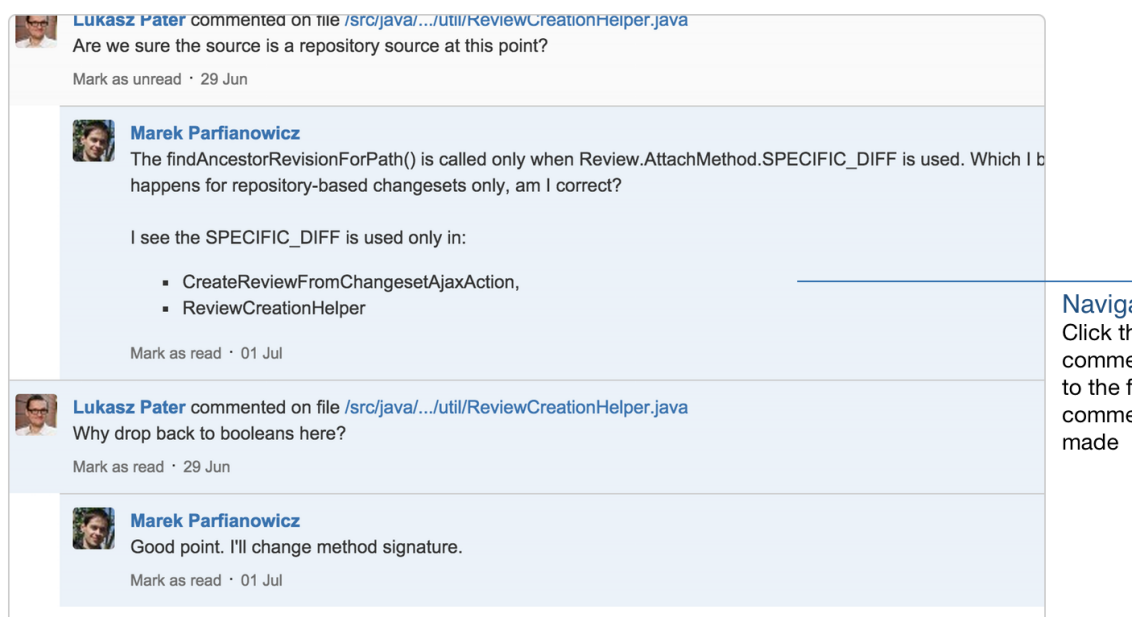
Activity **8**

[Mark all as read](#) Keyboard shortcuts 

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

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

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



 **Lukasz Pater** commented on file `/src/java/.../util/ReviewCreationHelper.java`
Are we sure the source is a repository source at this point?
Mark as unread · 29 Jun

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

I see the `SPECIFIC_DIFF` is used only in:

- `CreateReviewFromChangesetAjaxAction`,
- `ReviewCreationHelper`

Mark as read · 01 Jul

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

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

[Read more about commenting on reviews...](#)

Performance improvements

We think you'll notice a much faster experience with Crucible!

Review page load time

We've introduced permissions caching to speed up the loading of a review. The cache lifespan is request only.

In testing, load time for a review with about 500 comments on a single FRX diminished from 9s to a little over a second.

Project Review Dashboard load time

The **Reviews** tab for a project now loads faster than ever!

We've optimized the calculation of the average number of files per review in the 'Statistics' section, and now only display the unread comments count for the reviews that you are participating in. Further, the maximum open reviews label has been removed.

Faster searching by review Permalid

We've tweaked searching for review comments by the review Permalid to take fuller advantage of database indexes, resulting in a very useful performance gain.

Small improvements

Oracle 12c is supported

The bundled Oracle JDBC driver is now version 12cR1.

DVCS support

Git 2.5.0 and Mercurial 3.5 are now supported.

Java 7 support is removed

As [previously announced](#), Crucible 3.9 does not support Java 7.

Internet Explorer 9 support is removed

As [previously announced](#), Crucible 3.9 does not support Microsoft Internet Explorer 9.

Change log

This section will contain information about the Crucible 3.9 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#) and the [End of Support Announcements for Crucible](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.9.x releases.

4 August 2015 - Crucible 3.9.0

T	Key	Summary
	CRUC-6346	Create Report Based on Comments
	CRUC-6002	Crucible enhancements to comments functionality
	CRUC-2839	Review Summary Reports
	CRUC-7297	Upgrade bundled Oracle JDBC driver to 12cR1
	CRUC-7273	Defect Classification that are not specified should not appear in a review comment
	CRUC-7262	truncate git message in review title
	CRUC-7037	Oracle 12c Support
	CRUC-6898	New comments are harder to see in Crucible 3.5.x due to colours change
	CRUC-6082	Hard to reach hidden comments
	CRUC-6076	Filter Review History

CRUC-7312	scrollbar is not synchronized after you remove big comment
CRUC-7300	Same comment rendered twice on review page
CRUC-7290	Crucible tries to login to Perforce repo at shutdown
CRUC-7289	Project Review Dashboard is slow
CRUC-7279	Summarized Reviews not Shown on Review Dashboard for Participants
CRUC-7263	Disable hash link in blame column when FishEye is not licensed
CRUC-7259	NullPointerException When Creating a Review Where Latest Changeset Has No Commit Message
CRUC-7246	Advanced Comment Search problems with Review Permalid
CRUC-7241	Large reviews load page slow due to hibernate session dirty checking
CRUC-7133	Bug on Add Revisions in Crucible

Authenticate to retrieve your issues

Showing 20 out of 37 issues

Crucible 3.8 release notes

28 April 2015

Try it for FREE ➡

Today's release of [Crucible 3.8](#) includes a number of performance improvements, and a wider variety of diff formats are now accepted for pre-commit reviews.

If you are updating an earlier version of Crucible, please read the [Crucible upgrade guide](#) and the [End of Support Announcements for Crucible](#). The Crucible 3.8 changelog is [below](#).

Performance improvements

We've continued to make performance improvements in Crucible, so your development team can deliver software even faster. For Crucible 3.8, we've made the following improvements:

- A faster activity stream and quicker changeset log rendering through better memory utilization.
- A faster review dashboard and review search pages, in some cases resulting in a greater than 200 times faster page load time.
- Improved Git indexing time for newly created branches. We observed branch indexing time reduced more than 3 times.

Support for more patch formats

Pre-commit reviews are commonly used by many development teams. We have reviewed the diff formats generated by supported SCMs and improved Crucible's patch parsers to handle a wider variety of them.

Small improvements

Java 7 is deprecated

Crucible 3.8 is the last version to support Java 7, [as previously announced](#).

Change log

This section will contain information about the Crucible 3.8 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#) and the [End of Support Announcements for Crucible](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.8.x releases.

19 June 2015 - Crucible 3.8.1

T	Key	Summary
	CRUC-7133	Bug on Add Revisions in Crucible
	CRUC-6245	Disabling "Auto mark files as read" is not working

Authenticate to retrieve your issues

2 issues

28 April 2015 - Crucible 3.8.0

T	Key	Summary
	CRUC-7205	Slow review listing on review dashboard
	CRUC-7195	Misspelling in anchor message
	CRUC-7188	Typo in Crucible user profile settings
	CRUC-7159	Author mapping is failing with NPE
	CRUC-7139	Branch review help links to the page that does not exist
	CRUC-7138	Hidden comments counter is not visible
	CRUC-7083	Tracking DB statements in BoneCP - fixing DB statements leak
	CRUC-7050	Slow loadFrxFxajax caused by blame (getFileHistory)
	CRUC-7040	Irrelevant changes are not hidden in Branch reviews for SVN
	CRUC-7035	Creating review from Commit Graph results in Changeset not found
	CRUC-6448	Pre-commit Review Incorrectly Identifies Files in Patch When Patch Contains "----" (3 Dashes)
	CRUC-6114	Pre-commit review error: SVN patch containing Property changes cannot be uploaded.
	CRUC-6091	Crucible doesn't understand git diffs
	CRUC-6062	Crucible standalone shows "View in Graph" option, results in 404
	CRUC-5882	Adding then deleting a comment results in an empty notification email
	CRUC-5794	Improve syncing of deactivated users in Crucible

Authenticate to retrieve your issues

16 issues

Crucible 3.7 release notes

27 January 2015

Try it for FREE ➔

Today's release of [Crucible 3.7](#) includes some great improvements for branch reviews, as well as better support for Git and Mercurial.

If you are updating an earlier version of Crucible, please read the [Crucible upgrade guide](#) and the [End of Support Announcements for Crucible](#). The Crucible 3.7 changelog is [at the bottom](#) of this page.

Branch reviews are now automatically updated

Since Crucible 3.6 you've been able to easily create a review for the particular repository branch you've been working on.

Now, with Crucible 3.7, your branch review is quickly and automatically updated whenever new commits are made to the repository branch. All review participants get notified and are able to resume their reviews immediately. Just remember to create a review for your branch first!

Details

Participant	Role	Time Spent	Comments	Latest Comment
System Administrator	Author & Moderator	1m		
Total		1m	0	

Objectives [Edit](#)

- FECRU-5411 initial version of analytics

Branches in review

Repository: [git-test](#) Branches: [3.7-FECRU-5411-UI-analytics](#) [master](#)

General Comments

What do you want to say?

You review is automatically updated

Review updated: 1 file has been up

Read more about [Branch reviews](#).

Create a branch review from the activity stream

You can now create a branch review directly from the activity stream.

Once your changeset is indexed and visible in the activity stream, just click on the cog and choose **Create review for branch**. You'll see the option to add a branch to the review.

Activity stream showing commits and a context menu for creating a branch review.

Commits:

- 9d1b329 changed 1 file [master](#), 3.7 and 2 more [CR-FE-9145](#) 9:15 pm
- d6eb842 changed 1 file [master](#), 3.7 and 2 more [CR-FE-9145](#) 8:47 pm
- 541a850 changed 2 files [3.5-CRC-7434-file-camelcase-search-system-pr](#) 8:47 pm
- 4922dd0 changed 3 files [master](#), 3.7 and 5 more [CR-FE-9139](#) [CR-FE-9145](#) 4:45 pm
- c4ef1e5 changed 0 files [master](#), 3.7 and 4 more 4:15 pm

Context menu options:

- Create review
- Create review for branch
- View in graph
- Add to favourites

Other branch review improvements

We think you'll like these other improvements for branch reviews too. You can remove files from branch reviews and they won't be added again on next update, unless there were further changes to those files. And performance has been improved too – adding content to a review is now up to 10x faster. Read more about [Branch reviews](#).

Better support for Git and Mercurial

We've improved Crucible's support for the two most popular distributed version control systems on the market, Git and Mercurial.

Not only have we added support for Git 2.0, 2.1, 2.2 and Mercurial 3.x., but our intensive testing has revealed that some Git repositories are being indexed up to 8% faster, compared with Git 1.9.1. See [Supported platforms](#).

The latest versions of Git and Mercurial have fixes for the recently discovered CVE-2014-9390 vulnerability. Crucible is *not* directly affected by this vulnerability, but we recommend upgrading all your Git and Mercurial clients to the latest release. See [Atlassian update for Git and Mercurial vulnerability](#).

Small improvements

Deprecated support for Internet Explorer 9

Support for IE 9 is deprecated, and will be removed in April 2015. See [End of Support Announcements for Crucible](#).

Removed support for Firefox 3.6 and 4.0

Support for Firefox 3.6 and 4.0 has been removed in Crucible 3.7. See [Supported platforms](#).

Removed support for Safari 4 and 5

Support for Safari 4 and 5 has been removed in Crucible 3.7. See [Supported platforms](#).

Change log

This section will contain information about the Crucible 3.7 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#) and the [End of Support Announcements for Crucible](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.7.x releases.

7 April 2015 - Crucible 3.7.1

T	Key	Summary
	CRUC-7139	Branch review help links to the page that does not exist
	CRUC-7138	Hidden comments counter is not visible
	CRUC-7083	Tracking DB statements in BoneCP - fixing DB statements leak
	CRUC-7035	Creating review from Commit Graph results in Changeset not found

[Authenticate](#) to retrieve your issues

4 issues

27 January 2015 - Crucible 3.7.0

T	Key	Summary
	CRUC-7104	Branch review auto update
	CRUC-7101	create branch review from activity stream cog
	CRUC-6994	Slow commit list for branch reviews
	CRUC-7190	Full file visible in branch review instead of diff
	CRUC-7187	SCM setup screen contains a broken link
	CRUC-7103	User links broken on review page "more reviewers" popup
	CRUC-7102	Files added in a branch should be shown as added in the branch review
	CRUC-7092	Branch Review shouldn't re-add removed files or file revisions
	CRUC-7079	Anonymous user accessing a review with branch review will have "Rolled back transaction" warning
	CRUC-7066	broken REST documentation links
	CRUC-7065	/dev-status/1.0/details/repositories (used for commit details in JIRA) executes a project permission query for each changeset returned
	CRUC-7064	Queries with IN clauses not cached properly
	CRUC-7038	ORA-01000 - maximum open cursors exceeded
	CRUC-6993	Branch reviews doesn't handle excluded commits - NullPointerException
	CRUC-6991	{{ Monospaced }} comment blocks are rendered in large font
	CRUC-6985	Repository and branch selector dropdown does not show any remote request error
	CRUC-6965	Query to count number of reviews per repository is expensive
	CRUC-6962	Provide commits to JIRA (and other fusion clients) even when no FishEye license is installed
	CRUC-4180	Email Subject template does not allow for modification

Authenticate to retrieve your issues

19 issues

Crucible 3.6 release notes

28 October 2014

Try it for FREE ➞

Today we're proud to release **Crucible 3.6**. With a new way to add just the code that has changed to a review, and a new quick-nav Actions menu, Crucible helps your team perform better code reviews, faster!

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#) and the [End of Support Announcements for Crucible](#). The Crucible 3.6 changelog is [at the bottom](#) of this page.

Branch reviews

Branch reviews in Crucible provide the team with a quick and easy way to review changes made on a branch, discuss those changes, and make further modifications before the branch is merged to master or

your main development branch.

When you're adding content to a review, simply click **Choose branches**. From here, select the repository, the branch with the changes and the base branch (usually the branch you want to merge into once the review is complete). Crucible shows you a preview of the commits that will be added to your new, effective, review:

Add branches to the review

Repository

Branch to review


applinks-git


applinks-4.3.x

Add branch to review

Branched from


master

change

Recent commits

Author	Commit	Message	Commit date
 Automated Release Scrip...	163c036	APLDEV-197 added mvnvm.properties to manage maven version	5 days ago

Read more about [branch reviews](#)...

Quicker navigation

Now you can perform many review operations (such as Join, Close, and Complete), and navigate to administration tasks, without leaving the keyboard.

Simply type '.' (dot) to see a list common actions. Continue typing to quickly filter the actions available in your current context.



Small improvements

Performance improvements

The diff calculation times for the review and changeset pages have been significantly reduced, and the user listing page has been tuned, so it now loads faster!

SSL cipher suite configuration

Included and excluded cipher suites and protocols are now defined in the `config.xml` file, not the `jetty-web.xml` file as previously. See [Configuring SSL cipher suites for Jetty](#) for details.

Java 7 deprecation

We [announce](#) that Crucible will stop supporting Java 7 after April 2015.

Product analytics disclosure

In the spirit of openness, we disclose that, as of version 3.6, Crucible will collect user event data unless disabled by an administrator, as per Atlassian's [Privacy Policy](#). Analytics help us to determine the frequency of feature usage within the product, all towards improving Crucible.

See [Collecting analytics in Crucible](#).

Change log

This section will contain information about the Crucible 3.6 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.6.x releases.

23 January 2015 - Crucible 3.6.4

This release contains FishEye fixes only - see [FishEye 3.6 release notes#23January2015-FishEye3.6.4](#)

29 December 2014 - Crucible 3.6.3

T	Key	Summary
	CRUC-7065	<code>/dev-status/1.0/details/repositories</code> (used for commit details in JIRA) executes a project permission query for each changeset returned

Authenticate to retrieve your issues

1 issue

4 December 2014 - Crucible 3.6.2

T	Key	Summary
	CRUC-6994	Slow commit list for branch reviews
	CRUC-7055	OGNL Double Evaluation Vulnerability
	CRUC-7038	ORA-01000 - maximum open cursors exceeded
	CRUC-6993	Branch reviews doesn't handle excluded commits - NullPointerException
	CRUC-6991	<code>{{ Monospaced }}</code> comment blocks are rendered in large font
	CRUC-6985	Repository and branch selector dropdown does not show any remote request error
	CRUC-6965	Query to count number of reviews per repository is expensive
	CRUC-6962	Provide commits to JIRA (and other fusion clients) even when no FishEye license is installed

Authenticate to retrieve your issues

8 issues

30 October 2014 - Crucible 3.6.1

T	Key	Summary
	CRUC-6984	Branch Review doesn't work with Crucible only license

Authenticate to retrieve your issues

1 issue

28 October 2014 - Crucible 3.6.0

T	Key	Summary
	CRUC-6885	Ability to close Reviews from anywhere on the page
	CRUC-6594	Allow configuring SSL cipher suites and protocols in jetty ssl connector
	CRUC-6929	Show name, instead of, avatar in the Crucible Review listing
	CRUC-6746	Migration/restore fails while using MySQL > 5.5
	CRUC-4971	Allow to add the entire branch to be reviewed
	CRUC-6958	Edited draft comments are logged with old text in review history
	CRUC-6940	Poor performance on review creation with large changeset
	CRUC-6926	The action dialog doesn't reload the actions when the review author reload the review
	CRUC-6903	Possible NPE in DCM.getContents if DB data mangled
	CRUC-6819	Unable to migrate review to different project due to Project Allowed Reviewer Group
	CRUC-6508	Never wiki-render Display Names in Crucible comments
	CRUC-6464	NPE thrown when trying to change project key
	CRUC-6186	'Resource not found' Error in Review Coverage Gadget

Authenticate to retrieve your issues

13 issues

Crucible 3.5 release notes

22 July 2014

Try it for FREE ➞

Today we're proud to release [Crucible 3.5](#). With great improvements in the user interface and new API calls for managing instances, we think this release will delight both users and administrators.

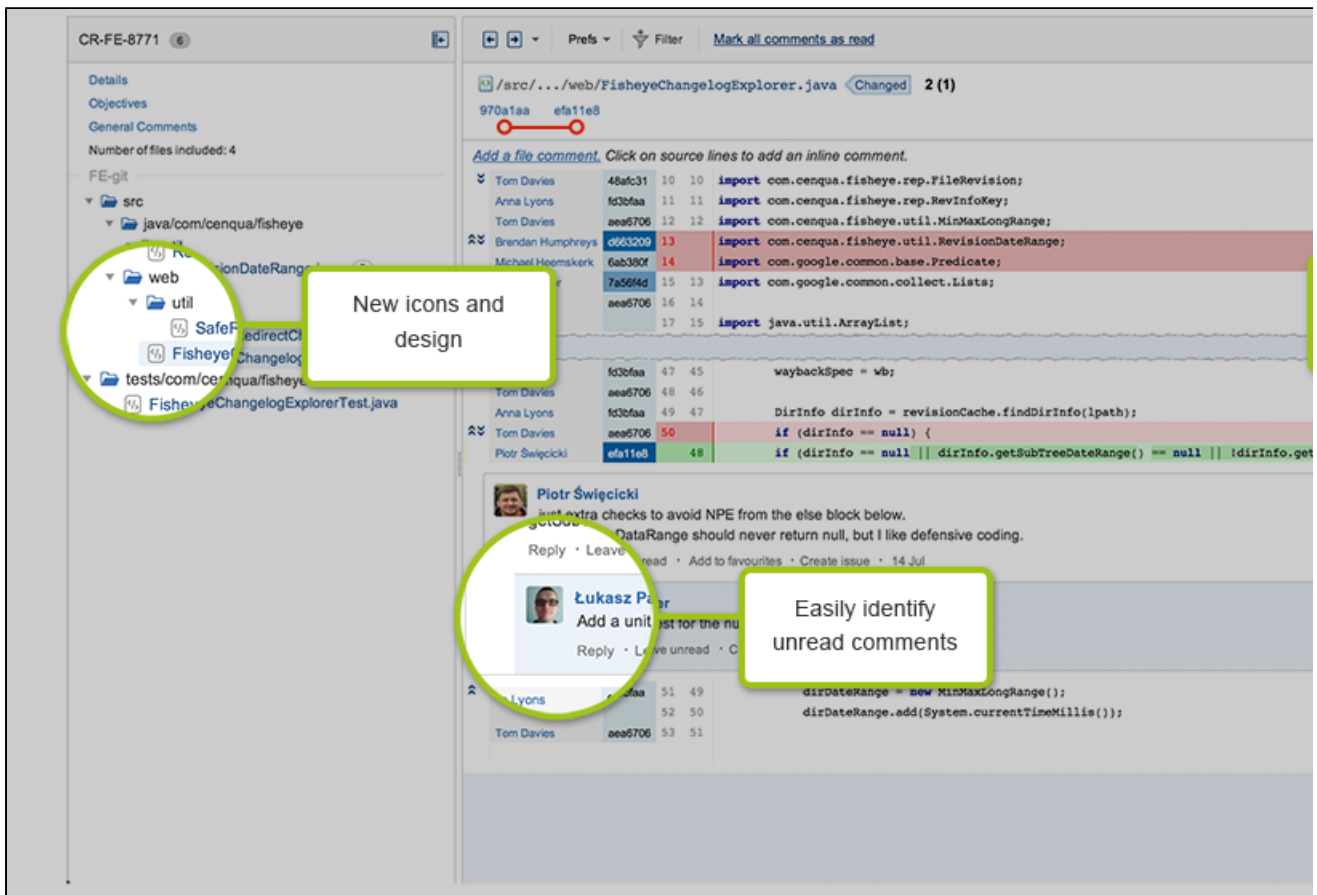
If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#). Please also

read the [End of Support Announcements for Crucible](#). The Crucible 3.5 changelog is [at the bottom](#) of this page.

More ADG goodness...

Reviewing code is the core activity when you're using Crucible, so it's important for that experience to be as good as possible. For Crucible 3.5 we've applied the new Atlassian Design Guidelines to review comments:

- The comments design has been aligned with other Atlassian products, such as Confluence and Stash.
- A blue bar on the right helps you to track the focused comment when navigating with the 'j' and 'k' shortcuts.
- Unread comments are easily identified by their blue background.
- You can post comments from the keyboard – just press Ctrl+Enter (or Cmd+Enter).



Better integration with Atlassian Stash

In [Crucible 3.4](#), we introduced a new integration with Stash that lets you index your Stash Git repositories in Crucible with just a single click.

Now, with Crucible 3.5, we're pushing the integration a step further by using Stash events to trigger incremental indexing of your Git repositories in Crucible. When you add a Stash repository to Crucible, the configuration is optimized so that Crucible only requests new commits once Stash notifies that a push has occurred. You don't have to configure polling anymore, or set up Crucible web hooks in your Stash instance. You'll also get the best network performance straight off.

See [Integrating FishEye with Stash](#).

New REST APIs for automating administration

For Crucible 3.5, we've extended the REST APIs further with `/projects` and `/permission-schemes/` endpoints.

Along with the `/users/`, `/groups/` and `/repositories/` endpoints that were made available in the Crucible 3.3 and 3.4 releases, you can now manage the configuration of your Crucible instance automatically.

Note that the 'v1' REST API resources are deprecated and will be removed in a future release.

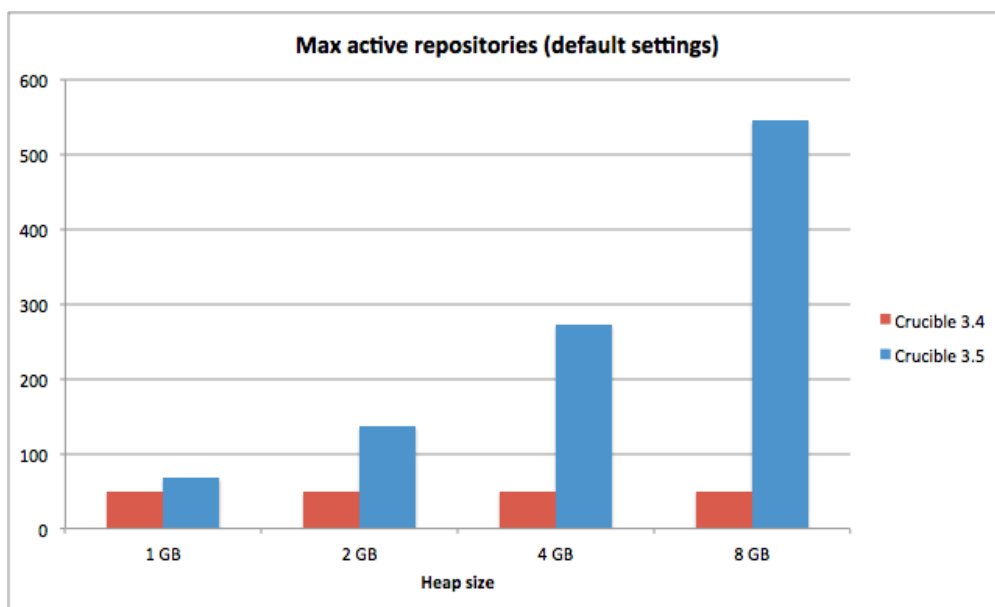
We've also published new [Python examples on Bitbucket](#) to help you build your own interfaces.

See the [FishEye Crucible REST API](#).



Optimized memory usage = Many more active repositories

We've added a new passivation mechanism that allows FishEye to take advantage of larger Java VM heaps. For large instances this provides a huge increase in the number of repositories that can be actively requesting data:



See [How FishEye uses memory](#) for details.

Small improvements

Platform upgrades

- Crucible now supports Java 8. See [Supported platforms](#).
- Crucible no longer supports versions of JIRA earlier than JIRA 5.0.

Change log

This section will contain information about the Crucible 3.5 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.5.x releases.

21 Janury 2014 - Crucible 3.5.5

T	Key	Summary
	CRUC-7055	OGNL Double Evaluation Vulnerability

Authenticate to retrieve your issues

1 issue

17 September 2014 - Crucible 3.5.4

T	Key	Summary
	CRUC-7049	Time submitted in minutes from Crucible being logged as hours in JIRA issues
	CRUC-6940	Poor performance on review creation with large changeset
	CRUC-6890	Minutes entered as hours into Jira when closing review

Authenticate to retrieve your issues

3 issues

27 August 2014 - Crucible 3.5.3

T	Key	Summary
	CRUC-6924	"Create Link" to JIRA issue fails if commit message contains key in lower case
	CRUC-6746	Migration/restore fails while using MySQL > 5.5
	CRUC-6903	Possible NPE in DCM.getContents if DB data mangled
	CRUC-4834	Control port connection refused problems when using the command line tools

Authenticate to retrieve your issues

4 issues

06 August 2014 - Crucible 3.5.2

T	Key	Summary
	CRUC-6907	When adding groups, add more than just the 9 first users
	CRUC-6880	REST Auth Service 500 error on Crucible 3.5

Authenticate to retrieve your issues

2 issues

23 July 2014 - Crucible 3.5.1

T	Key	Summary

Authenticate to retrieve your issues

No issues found

22 July 2014 - Crucible 3.5.0

T	Key	Summary
	CRUC-5901	Allow create and delete project via REST API
	CRUC-6871	NullPointerException in reviews on deleted binary files, if there was property only change before deletion
	CRUC-6864	Creating a review comment on a line of code makes the source jump around
	CRUC-6849	Crucible truncates review code
	CRUC-6833	Losing page scroll height upon switching diff mode in review preferences
	CRUC-6820	Reviews in approval state render as Draft in the review header
	CRUC-6767	ReviewService methods using RepositorySources may fail if there's no Disposer defined
	CRUC-6708	Triggering certain API calls from within event handlers might cause a db deadlock
	CRUC-6592	As a mail client user, I want Fecru emails from one review to stack
	CRUC-6568	Duplicated files on review - race condition

Authenticate to retrieve your issues

10 issues

Crucible 3.4 release notes

15 April 2014

Today we're proud to release [Crucible 3.4](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#). Please also read the [End of Support Announcements for Crucible](#).

The Crucible 3.4 changelog is [at the bottom](#) of this page.

Importing Git repositories from Atlassian Stash

When FishEye is integrated with Atlassian [Stash](#) a FishEye administrator can now easily add Stash repositories to FishEye – with just a single click. Once added, the repository behaves just like a native repository in FishEye, so your team gets all the benefits of FishEye indexing, browsing and searching. Furthermore, the repository becomes available to Crucible (when integrated), so you can perform in-depth code reviews for changes made to the Stash repository. [Read more...](#)

Repositories

Native repository access (119) Stash repositories Plugins (4)

Add Repositories hosted in Atlassian Stash to FishEye. ⓘ

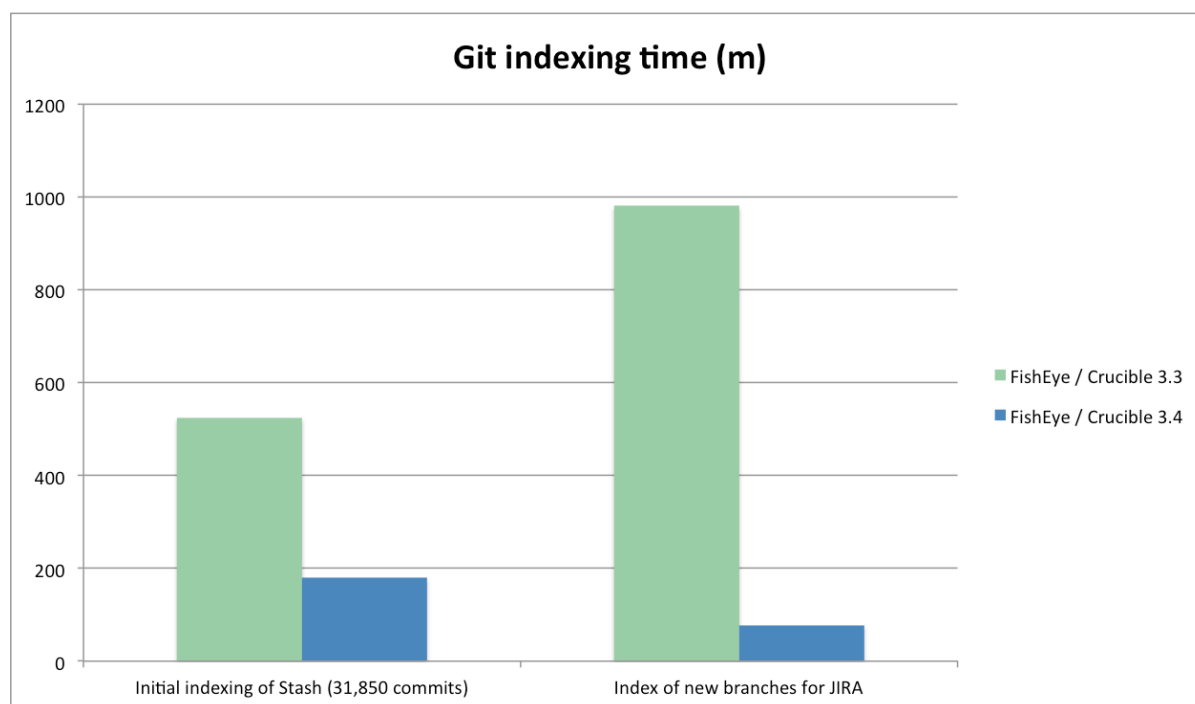
Stash server Stash Dev Q Type to filter repositories

Repository	FishEye repository	Actions
aaaaTest / Test Project & other things		<button>Add</button>
access-log-parser / Stash	STASH-access-log-parser	ADDED
access-log-parser / Jens Schumacher		<button>Add</button>
adam2 / Blitz	BLTZ-adam2	ADDED
AMKT / Test Project & other things	TEST-amkt	ADDED

Git indexing performance gains

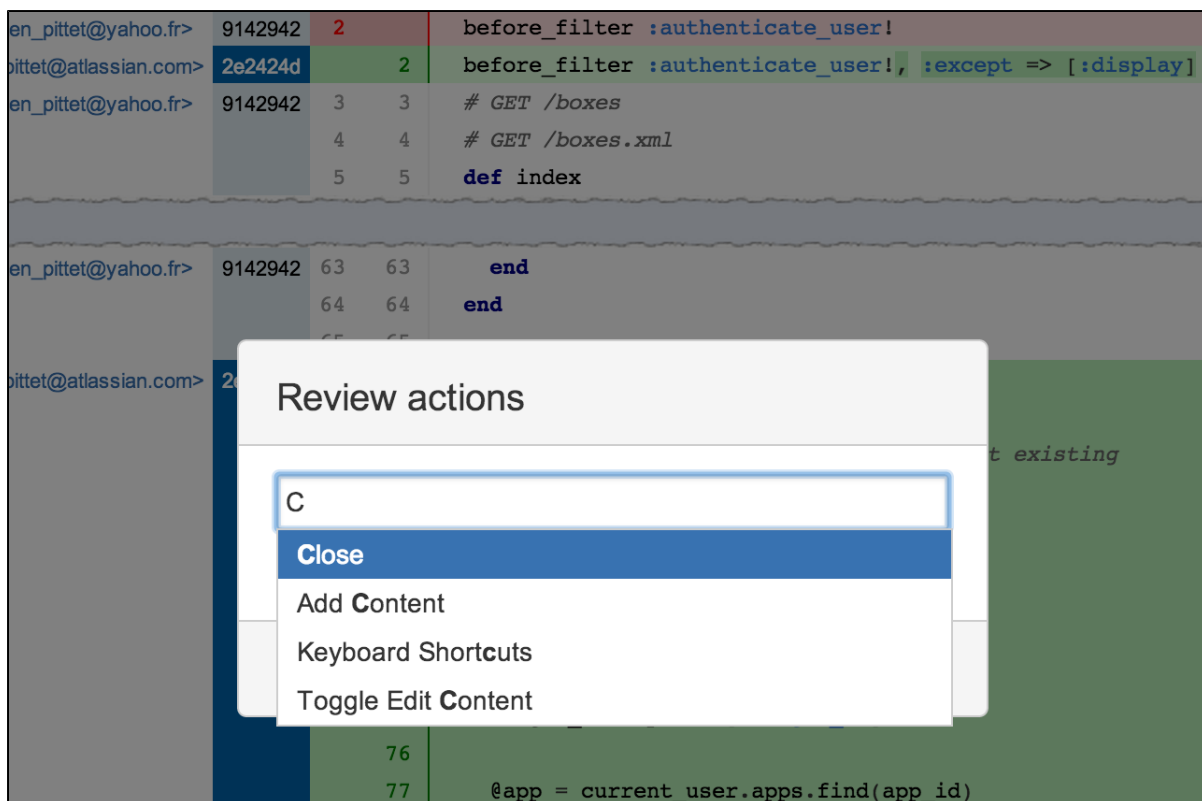
So that your experience with Git repositories is as good as possible, we've significantly improved the performance of the Git indexing operations in FishEye. You can [read more](#) about how this has been achieved.

The FishEye [upgrade guide](#) describes your options for how you run the upgrade task that is required.



Review actions dialog - just type dot!

We've added a new 'Review actions' dialog to speed up working with reviews. Simply type '.' (that is, dot, full stop, period, decimal point), then start typing the name of an action, to see a list of available commands. You'll be able to navigate through your review actions to edit or close a review, without having to reach for your mouse!



Administration REST APIs – part 2

In Crucible 3.3, we added `/users/` and `/groups/` resources so you could programmatically maintain users and groups, and manage group memberships.

Now, for Crucible 3.4, we've extended the Crucible REST APIs even further with the `/repositories/` resource to allow you to automate a range of administration tasks.

Note that the 'v1' REST API resources are deprecated and will be removed in a future release.

See the [FishEye Crucible REST API](#).



Windows installer for Crucible

We've produced 32-bit and 64-bit installers for Crucible on Windows. Each installer adds Crucible as a Windows service, and starts the service, automatically. The express install creates, by default, a Data directory and a separate install directory in `C:\Atlassian`. The custom install mode allows you to choose different locations for the install and Data directories. Download the Crucible installer [here](#).

Small improvements

Automatic fullscreen mode for reviews

Now you automatically get fullscreen mode as you scroll down to review changes made in the code. The file browser and the revision selector stay in place to make it easier for you to navigate through the diffs and comments (see 'Header behaviour' below).

Header behavior

You'll notice that the headers for changeset, source code and review listings now stalk – they stay visible when you scroll down. Just a small improvement to make life easier.

Change log

This section will contain information about the Crucible 3.4 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.4.x releases.

16 September 2014 - Crucible 3.4.7

This release contains FishEye fixes only

27 August 2014 - Crucible 3.4.6

T	Key	Summary
	CRUC-6746	Migration/restore fails while using MySQL > 5.5
	CRUC-6867	Header is wrapping for very narrow pages

Authenticate to retrieve your issues

2 issues

1 July 2014 - Crucible 3.4.5

T	Key	Summary
	CRUC-6849	Crucible truncates review code
	CRUC-6833	Losing page scroll height upon switching diff mode in review preferences
	CRUC-6822	IE9 - Review "browse changeset" dialog does not appear
	CRUC-6818	UTF-8 patches are not correctly displayed

Authenticate to retrieve your issues

4 issues

21 May 2014 - Crucible 3.4.4

T	Key	Summary
	CRUC-6810	Administrator password reset

Authenticate to retrieve your issues

1 issue

8 May 2014 - Crucible 3.4.3

This release contains FishEye fixes only

7 May 2014 - Crucible 3.4.2

This release contains FishEye fixes only

6 May 2014 - Crucible 3.4.1

T	Key	Summary
	CRUC-6808	The reviewers are not refreshed, when adding new content to a review
	CRUC-6600	Whitespace at the end of a pasted patch are stripped off
	CRUC-6316	Uploading patch through REST API fail to anchor to a revision due to CRLF ending

Authenticate to retrieve your issues

3 issues

15 April 2014 - Crucible 3.4.0

T	Key	Summary
	CRUC-474	Allow adding a group as moderators/reviewers of reviews
	CRUC-6719	Allow bulk activation of Crucible users
	CRUC-6744	Add option to license user for Crucible access upon user creation
	CRUC-6743	Project page gets date of most recent changes very inefficiently
	CRUC-6736	Review Attachments do not have their encoding correctly determined
	CRUC-6728	P4 diff on recent versions of Perforce does not produce the unified diff header we expect, even with the correct flags
	CRUC-6592	As a mail client user, I want Fecru emails from one review to stack
	CRUC-6566	Reviews are not being sorted based on "Name" column
	CRUC-6469	CommentDeletedEvent is not getting triggered.

Authenticate to retrieve your issues

9 issues

Crucible 3.3 release notes

11 February 2014

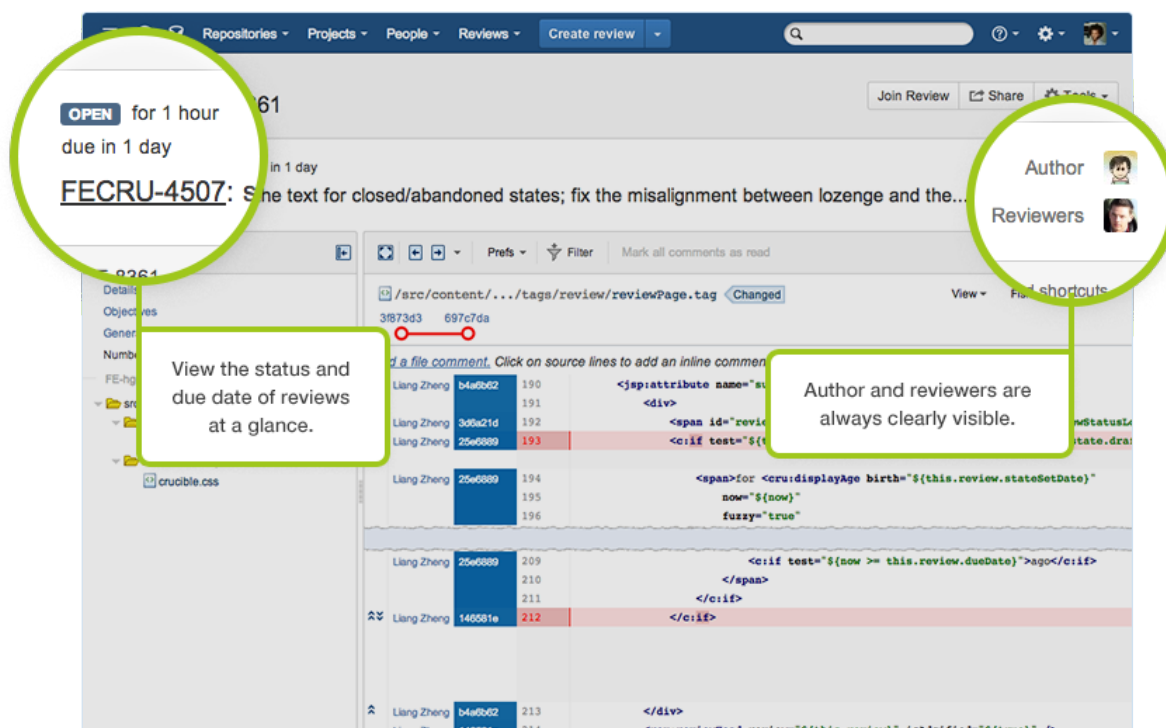
Today we're proud to release [Crucible 3.3](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#). Please also read the [End of Support Announcements for Crucible](#).

The Crucible 3.3 changelog is [at the bottom](#) of this page.

Improved review header

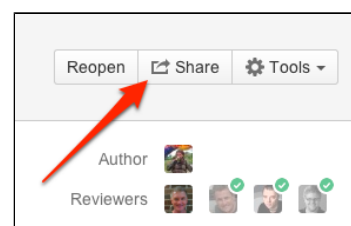
Continuing the work to enhance your experience of using Crucible, we've improved the review header. The header now has a cleaner layout, and we think you'll find it easier to see key information such as the review status, due date and who the author and reviewers are:



We'd like to call out the keyboard shortcut to toggle fullscreen mode (Shift-F), which hides the header and frees up that precious screen real estate. By the way, Shift-T toggles the left-hand file tree panel.

Sharing replaces inviting a guest reviewer

Crucible allows you to share a review with anyone, which is quicker and more versatile than inviting a guest reviewer, so we've removed that ability from Crucible 3.3.



Administration REST APIs

We're extending the Crucible REST APIs to allow you to automate a range of administration tasks. For Crucible 3.3, we've added `/users/` and `/groups/` resources – these allow you to programmatically retrieve, update and delete both users and groups, and to add users to, or remove them from, groups.

See [FishEye Crucible REST API](#).

Support for Subversion 1.8

Crucible now supports Subversion 1.8, so you can use Crucible to review your Subversion 1.8 repositories. See [Native support for SVN 1.7](#).



Support for Internet Explorer 11

Crucible now supports Microsoft Internet Explorer 11.

See [Supported platforms](#).



Small improvements

Platform changes

- Support for IE 8 has now ended with the release of Crucible 3.3.
- Support for MySQL 5.0 has now ended with the release of Crucible 3.3.
- Support for PostgreSQL 8.2 has now ended with the release of Crucible 3.3.
- Support for SQL Server 2005 has now ended with the release of Crucible 3.3.

See [Supported platforms](#).

Change log

This section will contain information about the Crucible 3.3 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.3.x releases.

21 May 2014 - Crucible 3.3.4

T	Key	Summary
	CRUC-6810	Administrator password reset
	CRUC-6744	Add option to license user for Crucible access upon user creation

Authenticate to retrieve your issues

2 issues

Crucible 3.3.4 is a security maintenance release. See [FishEye and Crucible Security Advisory 2014-05-21](#).

4 April 2014 - Crucible 3.3.3

This release contains FishEye fixes only

19 March 2014 - Crucible 3.3.2

T	Key	Summary
	CRUC-6743	Project page gets date of most recent changes very inefficiently
	CRUC-6736	Review Attachments do not have their encoding correctly determined
	CRUC-6728	P4 diff on recent versions of Perforce does not produce the unified diff header we expect, even with the correct flags
	CRUC-6469	CommentDeletedEvent is not getting triggered.

Authenticate to retrieve your issues

4 issues

21 February 2014 - Crucible 3.3.1

This release contains FishEye fixes only

11 February 2014 - Crucible 3.3.0

T	Key	Summary
	CRUC-6685	Display project name in Create Review project dropdown
	CRUC-6710	Branch and User are not longer filtered on "add changset to review" dialog
	CRUC-6687	Filter changesets by user rather than by guessing user's default committer
	CRUC-6682	Add Content to Review: repo names are truncated, leading to inability to distinguish repos with similar names.
	CRUC-6315	Error on writing config.xml causes file corruption
	CRUC-5798	Linking review throws JavaScript errors
	CRUC-5347	Author does not have visibility of a peer review when it has moved to the Moderator's To Summarize Inbox

Authenticate to retrieve your issues

7 issues

Crucible 3.2 release notes

27 November 2013

Today we're excited to release [Crucible 3.2](#), which brings a range of usability and platform improvements.

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#). Please also read the [End of Support Announcements for Crucible](#).

The Crucible 3.2 changelog is [at the bottom](#) of this page.

High design values with the ADG

We've been applying the [Atlassian Design Guidelines](#) to make using Crucible a better experience – the layout and behaviour have been improved, and if you use other Atlassian products then Crucible will feel very familiar from the moment you log in. In particular, for Crucible 3.2, the header area for a review now has a much cleaner layout:

CR-FE-8192

Details

Participant	Role	Time Spent	Comments
Conor MacNeill	Author	3m	
Lukasz Pater	Reviewer - 0% reviewed		
Thomas Davies	Reviewer - 0% reviewed		
Total		3m	0

Linked Issue: [FECRU-4055](#) – Handling of deleted files in Git manifest tracking

Quick Search filters

We've added filters to QuickSearch, to make finding the content you want even quicker!

Now, in Crucible 3.2, you can also filter by the last modified date and by author.

Of course, for a while now, you've been able to filter search results by reviews and review comments, and restrict the search scope for reviews to a particular project. And when used with [Atlassian FishEye](#), you can filter for different source types such as files and directories, commit messages, diffs, file content, and committers.

QuickSearch filters

Source: [test](#) Search 78,283 results for **test** (1.662s)

Restrict to one repo

Restrict to a project

Restrict by date

Restrict by author

Read more about [searching in Crucible...](#)

Internally managed Git repositories are no longer supported

As [previously announced](#), internally managed Git repositories are no longer supported by FishEye 3.2.

Please read the [migration guide](#) for information about options and procedures for migrating your internally managed Git repositories.

Crucible communication to JIRA versions older than 5.0 is no longer supported

As [previously announced](#), communication from Crucible to JIRA versions older than 5.0 is no longer supported by Crucible 3.2. Please note that communication from JIRA to Crucible will continue to work as it currently does.

Small improvements

Platform upgrades

- Microsoft SQL Server 2012 is now supported ([support for SQL Server 2005 is deprecated](#)).
- Microsoft Internet Explorer 10 is now supported ([support for IE 8 is deprecated](#)).
- MySQL 5.0 [is deprecated](#).
- PostgreSQL 8.2 [is deprecated](#).
- The Atlassian AUI plugin has been upgraded to AUI 5.2.
- jQuery has been upgraded to 1.8.3.
- backbonejs has been upgraded to 1.0.

See [Supported platforms](#).

Improved protection against XSRF attacks

We've added a security token to protect Crucible from XSRF attacks and to allow plugin developers to add protection to their plugins. See the [Crucible upgrade guide](#) for details of changes required by LightSCM plugins.

User data moved to the SQL database

To prevent any chance of loss, the user data store has relocated. See the [Crucible Upgrade guide](#) for details.

Change log

This section will contain information about the Crucible 3.2 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.2.x releases.

21 May 2014 - Crucible 3.2.5

T	Key	Summary
	CRUC-6810	Administrator password reset

Authenticate to retrieve your issues

1 issue

Crucible 3.2.5 is a security maintenance release. See [FishEye and Crucible Security Advisory 2014-05-21](#).

15 January 2014 - Crucible 3.2.4*This release contains FishEye fixes only***16 December 2013 - Crucible 3.2.3***This release contains FishEye fixes only***11 December 2013 - Crucible 3.2.2***This release contains FishEye fixes only***6 December 2013 - Crucible 3.2.1***This release contains FishEye fixes only***27 November 2013 - Crucible 3.2.0**

T	Key	Summary
	CRUC-6609	JIRA issue transition after closing review looks bad
	CRUC-6662	Privilege escalation
	CRUC-6624	Document changes to the sharing features of Crucible
	CRUC-6620	Broken tooltip in quicknav for reviews
	CRUC-6608	Review indexing interrupted by killing application does not continue after restart
	CRUC-6595	Some Ajax calls never return, eventually timeout with a 502 proxy error
	CRUC-6588	Make it possible to query only selected users via REST
	CRUC-6587	REST /users-v1/username should return a limited data set for other users
	CRUC-6586	Should return avatar from external server if configured
	CRUC-4697	Empty strings accepted in proxy-host / proxy-scheme
	CRUC-3008	REST should return a 401 when using invalid basic http credentials instead of falling back to anonymous

Authenticate to retrieve your issues

11 issues

Crucible 3.1 release notes

27 August 2013

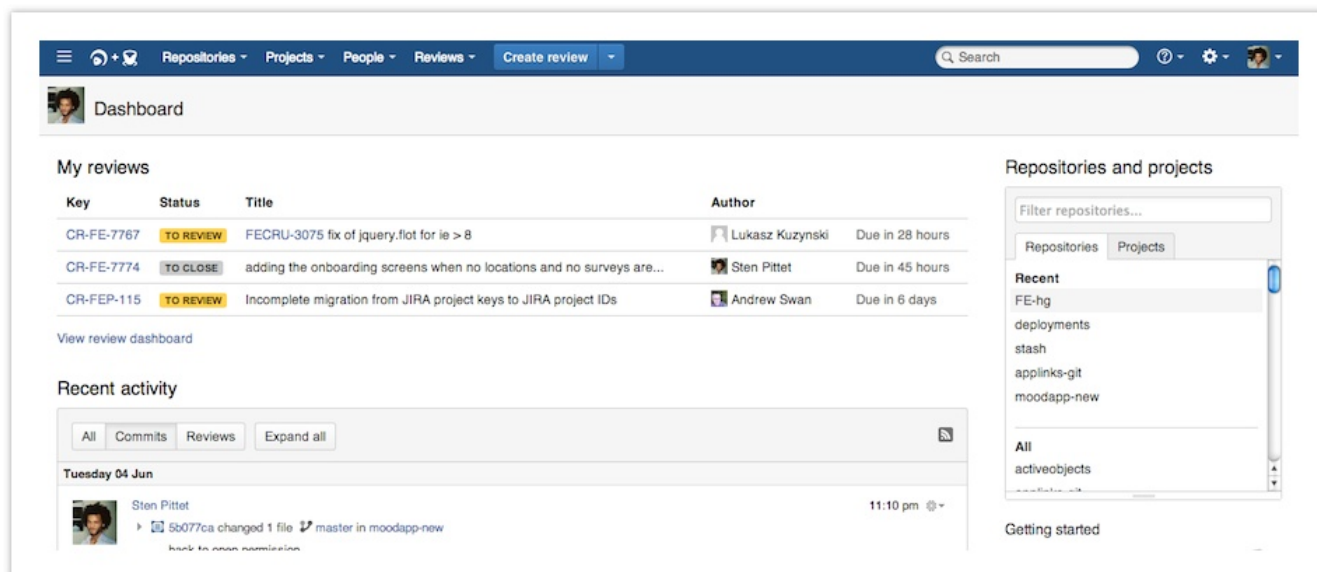
Today we're excited to release [Crucible 3.1](#), which brings you a cleaner and faster interface. We've also added some further JIRA integration points.

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#). Please also read the [End of Support Announcements for Crucible](#).

The Crucible 3.1 changelog is [at the bottom](#) of this page.

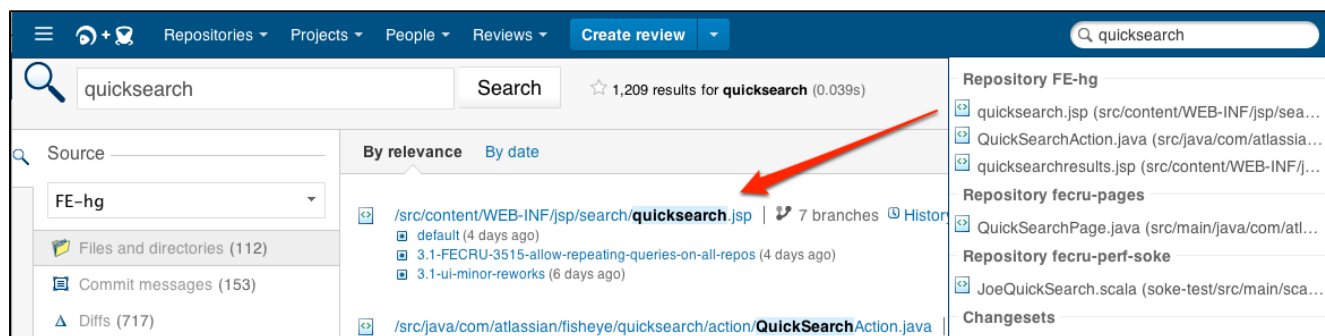
The new Dashboard

Crucible's new dashboard makes your pending reviews the highest priority, putting them right at the top and clearly indicating their urgency. The repository and project navigator lets you jump to your recent content (try pressing Enter as soon as you launch the dashboard), and includes filters and keyboard shortcuts that facilitate navigation, especially when your organisation has hundreds of repositories.



QuickNav and QuickSearch improvements

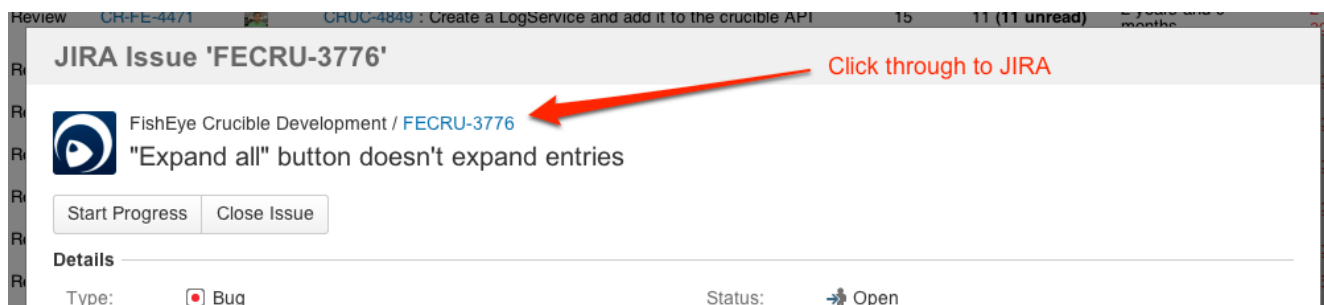
Previously in Crucible, QuickSearch was limited to searching 100 repositories concurrently. Now, in Crucible 3.1, you can search across files and directories, commit messages, and committers, for every one of your repositories. And it's 40% faster. And you can order search results by date. Furthermore, QuickNav now uses the same indexes as QuickSearch, so it's faster too, and it produces the same results as QuickSearch:



Note that upgrading to Crucible 3.1 requires an upgrade to the cross-repository index. See the [Crucible upgrade guide](#) for details.

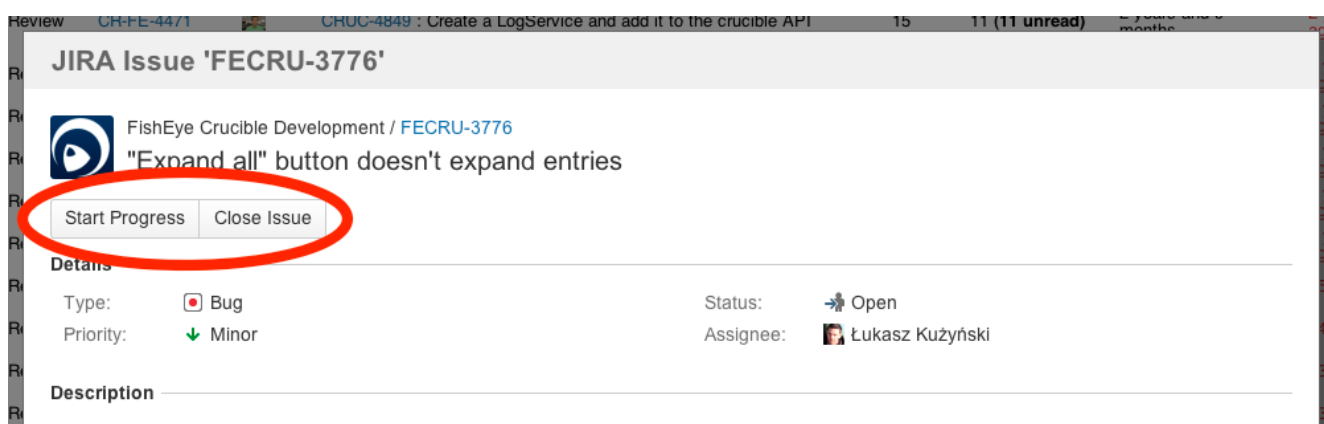
New JIRA issue dialog

Now, when you click on any linked JIRA issue key in Crucible, you'll see a dialog that displays a summary of the issue details. From that dialog, you can [transition the workflow](#) for the issue, and easily click through to edit the issue in JIRA:



Transition JIRA issues from within Crucible

Previously in Crucible, you could transition a JIRA issue linked to a review at the time that you closed the review. Now you can advance the workflow for any JIRA issue mentioned in Crucible at any time, right from within Crucible – just click the issue link to see the transitions available to you:

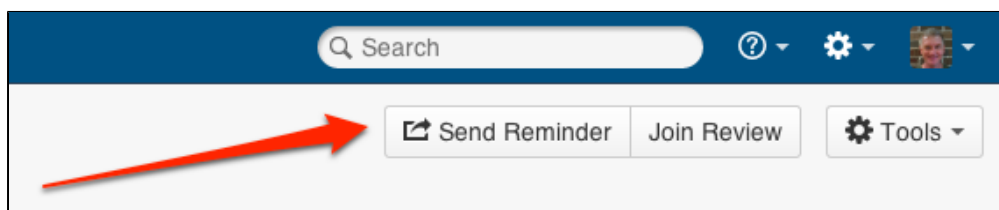


[Read more about JIRA issue transitions in Crucible ...](#)

Small improvements

Simplified reminder for overdue reviewers

Alerting reviewers about an overdue review is now a simple 1-click action:



Improved browsing performance

The display of the file source page is now 60% faster in Crucible 3.1.

Native SVN 1.7 support

Crucible 3.1 now supports the native JavaHL 1.7 SVN client. It does not support the native JavaHL 1.6 SVN client. See [Native support for SVN](#) for more details.

OpenJDK support

Crucible 3.1 now supports OpenJDK 1.7. See [Supported platforms](#) for details.

Change log

This section will contain information about the Crucible 3.1 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.1.x releases.

21 May 2014 - Crucible 3.1.7

T	Key	Summary
	CRUC-6810	Administrator password reset

Authenticate to retrieve your issues

1 issue

Crucible 3.1.7 is a security maintenance release. See [FishEye](#) and [Crucible Security Advisory 2014-05-21](#).

2 December 2013 - Crucible 3.1.6

There are no Crucible-specific fixes in this release. See the [FishEye 3.1 release notes](#) for some general issues that have been fixed.

T	Key	Summary
	CRUC-6662	Privilege escalation

Authenticate to retrieve your issues

1 issue

30 October 2013 - Crucible 3.1.5

T	Key	Summary
	CRUC-6644	Error removing files from pre-commit patch

Authenticate to retrieve your issues

1 issue

8 October 2013 - Crucible 3.1.4

This release also includes some minor security fixes.

T	Key	Summary
	CRUC-6620	Broken tooltip in quicknav for reviews

Authenticate to retrieve your issues

1 issue

17 September 2013 - Crucible 3.1.3

Fixes a bug which can cause review indexing to fail when upgrading from versions prior to 3.0.

T	Key	Summary
	CRUC-6595	Some Ajax calls never return, eventually timeout with a 502 proxy error

Authenticate to retrieve your issues

1 issue

11 September 2013 - Crucible 3.1.2

- a few XSS security issues addressed

29 August 2013 - Crucible 3.1.1

T	Key	Summary
	CRUC-6584	Unable to create reviews after upgrade to 3.1

Authenticate to retrieve your issues

1 issue

27 August 2013 - Crucible 3.1.0

T	Key	Summary
	CRUC-6384	Allow issue transitions for issues raised in Inline Issue Creation
	CRUC-6530	Remove Crucible Transition error as a warning
	CRUC-6949	"Comparison method violates its general contract!" Error is thrown when attempting to load a review
	CRUC-6534	Type when starting FishEye/Crucible
	CRUC-6525	upgrade to 3.x release occasionally fails when deduplicating paths in db
	CRUC-6417	Better handling of Authentication Errors when manually linking a Review to a JIRA Issue
	CRUC-6416	Optimize number of database queries for Crucible instances with Many Projects and Permission Schemes

CRUC-6339 Send notifications for replies to comments even if the user is not in the review

CRUC-5503 Favouriting isn't possible on cross-repository searches

Authenticate to retrieve your issues

9 issues

Crucible 3.0 release notes

30 May 2013

Today we're excited to release [Crucible 3.0](#), which introduces huge indexing performance gains – especially for Subversion repositories – and significant upgrades to how you perform pre-commit reviews. We think development teams will be able to work faster, every day.

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#).

The Crucible 3.0 changelog is [at the bottom](#) of this page.

Redesigned user experience

We've made a few key design changes to help you work faster with your reviews:

New global header

The new header across the top of the page makes it quicker to access recent repositories, projects, people and reviews, and has the large **Create review** button at top center, right where you can find it.

Application navigator

The new application navigator, at the left of the header, connects you directly to your other applications, such as JIRA and Bamboo. Now you can switch between Crucible and JIRA – or any other Atlassian application – all from the Crucible header. No more bookmarks in your browser; we do the job for you. Admins can easily configure which apps appear in the navigator – just click **Application navigator** in the admin area.

JIRA Reviews tab redesign

We wanted to simplify the experience but still give you the important information you need to make decisions around your issues. We think you'll like it:

Activity

All

Comments

Work Log

History

Activity

Commits

Source

Reviews

Builds

CR-FE-7319 **CLOSED**

[FEGRU-3310](#) document @experimental REST calls added for barracuda performance testing
{cs:id=42f8aedebeef471f2adb5c6dac08d03635b11b17b|rep=FE-hg}
: [FEGRU-3310](#) added javadocs, test

CR-FE-7341 **DRAFT**

Crucible code reviews made easy
Inline comments and threaded discussions.

Review CLI tool

Crucible now ships with a command line tool that allows you to quickly create reviews in Crucible directly from your terminal. The tool takes the uncommitted changes in your workspace and creates a review for them in Crucible. You can use the tool to create new reviews, and to update existing pre-commit reviews with new patches. [Read more..](#)

Iterative pre-commit reviews

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

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

Create reviews sooner for new SVN repositories

We've introduced a new approach that splits the indexing process into separate tasks that can be performed in a phased and concurrent way. This allows you to start using core functionality in Crucible, such as creating reviews, up to 15X sooner than in Crucible 2.10. You can get on with your work, while Crucible quietly completes the fine details of indexing in the background. [Read more...](#)

Small improvements

Faster browsing for large teams

Crucible now uses the InfinityDB 3.0 internal database to provide faster concurrent access. Your team gets a better browsing experience!

Quicker review creation

We've optimised some SQL statements in Crucible. You spend less time creating, or modifying, reviews.

Better review creation from JIRA when Stash is integrated

We've improved the integration between JIRA and Crucible. Now you can create a review for all the commits related to a JIRA issue, even when Atlassian Stash is integrated with JIRA.

Activity

All Comments Work Log History Activity Commits **Source** Reviews Builds

Source changes - FishEye

Maciej Swinarski <mswinarski@atlassian.com> committed [7a373ea...](#) (2 files) 28/May/13 2:42 AM

Reviews: none Create review

merge with 3.0-[FECRU-3310](#)

FE-hg 3.0

-	-	build.xml
-	-	src/java/com/atlassian/fecru/gwt/admin/client/ui/repositories/displays/vieworedit/SummaryTab.ui.xml

Diff hunk shortcuts

We've changed the 'j' and 'k' keyboard shortcuts to now work with diff hunks when looking at a review, so as to be consistent with their new behaviour in FishEye.

Improved handling of user preferences with session cookies

We've made session cookies more robust, but this may result in your existing preferences being lost at upgrade time.

REST API change

The 'groups-v1' REST endpoint now requires the user to be an administrator – it will fail authentication and return a 403 otherwise. [Read more...](#)

Change log

This section will contain information about the Crucible 3.0 minor releases as they become available. These releases will be free to all customers with [active Crucible software maintenance](#).

If you are upgrading from an earlier version of Crucible, please read the [Crucible upgrade guide](#).

The issues listed below are just the highlights of all those that have been resolved for the Crucible 3.0.x releases.

21 May 2014 - Crucible 3.0.4

T	Key	Summary
	CRUC-6810	Administrator password reset

Authenticate to retrieve your issues

1 issue

Crucible 3.0.4 is a security maintenance release. See [FishEye](#) and [Crucible Security Advisory 2014-05-21](#).

23 July 2013 - Crucible 3.0.3

T	Key	Summary
	CRUC-6556	Error editing permission scheme action containing deleted user
	CRUC-6552	Searching over REST for a JIRA Project key doesnt look at Review Linked Issues
	CRUC-6537	Deleting and recreating projects with keys which differ only in case can cause NonUniqueResultException on Viewing Review

Authenticate to retrieve your issues

3 issues

2 July 2013 - Crucible 3.0.2

T	Key	Summary
	CRUC-6503	Crucible upgrade may fail in the middle if DB socket timeout too short
	CRUC-6555	Can't use distinct in queries on tables with CLOB types in Oracle
	CRUC-6504	Review pre-commit tool can fail to include all the lines for an added file with Perforce
	CRUC-6416	Optimize number of database queries for Crucible instances with Many Projects and Permission Schemes

Authenticate to retrieve your issues

4 issues

6 June 2013 - Crucible 3.0.1

T	Key	Summary
	CRUC-6491	duplicated null cru_stored_path rows leads to 3.0 upgrade problems

Authenticate to retrieve your issues

1 issue

28 May 2013 - Crucible 3.0.0

T	Key	Summary
	CRUC-6434	Add disclaimer that "Create crucible review" is only available with FishEye integration
	CRUC-6414	Update on 'Editing a Project' documentation
	CRUC-6271	Update on Using Gadgets in Crucible documentation
	CRUC-6410	Allow /filter/details to include a date range filter
	CRUC-6361	Change semantics of j/k shortcuts in Crucible, be consistent with the new diff navigation in Fisheye
	CRUC-6394	Crucible Project Repository Link is Broken if Leading Slash Is Omitted
	CRUC-6391	Documentation Update for JIRA Integration

	CRUC-6349	Crucible and FishEye Page Contains Outdated Information
	CRUC-4719	Improve Project Creation/Editing with large user base
	CRUC-4518	Crucible regularly hangs and is very slow
	CRUC-2998	Iterative pre-commit reviews
	CRUC-1452	REST API should notify client if session token is invalid
	CRUC-228	Helper tools needed to smooth the process for creating patch reviews
	CRUC-6466	Dismiss barracuda's feature discovery message when it is closed.
	CRUC-6463	Multiple open transactions in idle instance
	CRUC-6454	Scheduled user synchronization doesn't close DB sessions
	CRUC-6412	Stream has already been closed on getting Notifications
	CRUC-6408	URLs in comments with # followed by ! and other characters are not correctly generated.
	CRUC-6398	Unable to link to JIRA issues containing a project key longer than 10 characters, containing underscores _
	CRUC-6395	Inconsistent Repository Links on Project Page

Authenticate to retrieve your issues

Showing 20 out of 31 issues

Security advisories

Finding and reporting a security vulnerability

The channel for reporting a security issue is detailed in [How to Report a Security Issue](#).

Publication of security advisories

Atlassian's approach to publishing security advisories is detailed in [Security Advisory Publishing Policy](#).

Severity levels

Atlassian's scale for measuring security issues is detailed in [Severity Levels for Security Issues](#).

Our patch policy

Atlassian's approach to releasing patches is detailed in our [Security Patch Policy](#).

Security advisories

- [Crucible Security Advisory 2015-01-21](#)
- [FishEye and Crucible Security Advisory 2014-05-21](#)
- [FishEye and Crucible Security Advisory 2014-02-26](#)
- [FishEye and Crucible Security Advisory 2013-07-16](#)
- [FishEye and Crucible Security Advisory 2012-08-21](#)
- [FishEye and Crucible Security Advisory 2012-05-17](#)
- [FishEye and Crucible Security Advisory 2012-01-31](#)
- [FishEye and Crucible Security Advisory 2011-11-22](#)
- [FishEye and Crucible Security Advisory 2011-05-16](#)

- [FishEye and Crucible Security Advisory 2011-01-12](#)
- [Crucible Security Advisory 2010-06-16](#)
- [Crucible Security Advisory 2010-05-04](#)

Crucible Security Advisory 2015-01-21

Note: As of September 2014, we no longer issue binary bug patches. Instead we create new maintenance releases for the major versions we backport to. Please see our [Security Bug fix Policy](#) for more details.

Date of Advisory: 21st January 2015

Product: Atlassian Crucible

Summary of Vulnerability

This advisory discloses a **critical severity** security vulnerability that exists in all versions of Crucible up to and including 3.6.1.

- **Customers who have downloaded and installed Crucible** should upgrade their existing **Crucible** installations to fix this vulnerability.

Atlassian is committed to improving product security. The vulnerability listed in this advisory has been discovered internally by Atlassian.

OGNL Double Evaluation Vulnerability

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in our [Atlassian severity levels](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have discovered and fixed a vulnerability in our fork of WebWork. Attackers can use this vulnerability to execute Java code of their choice on systems that use this framework. The attacker needs to be able to access the Crucible web interface.

All versions of Crucible up to and including 3.6.1 are affected by this vulnerability. This issue can be tracked here:  **CRUC-7055** - OGNL Double Evaluation Vulnerability CLOSED

Risk Mitigation

If you are unable to upgrade your Crucible server you can do the following as a **temporary workaround**:

- Block access to your Crucible server web interface from untrusted networks, such as the Internet.
- Block at a reverse proxy or a firewall all requests matching the following regular expression pattern in URI parameters.

```
.*(?:%25|\\$|%24)(?:[{(|)%7B|%28}).*(?:[(#)|%28|%23]).*(?:[}]|%7D|%29).*
```

Fix

Releases 3.5.5, 3.6.2 (and any subsequent newer releases) are available to fix the vulnerability for versions 3.5 and 3.6 respectively. You can download these releases from:

- Crucible - <https://www.atlassian.com/software/crucible/download>

Upgrade (recommended)

The vulnerabilities and fix versions are described in the sections above.

Atlassian recommend that you upgrade to the latest version. For a full description of the latest version of Crucible, see its [release notes](#).

It is advised that you upgrade to the latest version of Crucible, as there are no longer binary patches made available.

Support

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

References

Security Bug fix Policy	As per our new policy, critical security bug fixes will be back ported to major software versions for up to 12 months for FishEye and Crucible. We will release new maintenance releases for the versions covered by the new policy instead of binary patches. Binary patches will no longer be released.
Severity Levels for security issues	Atlassian security advisories include a severity level. This severity level is based on our self-calculated CVSS score for each specific vulnerability. CVSS is an industry standard vulnerability metric. You can also learn more about CVSS at FIRST.org .

FishEye and Crucible Security Advisory 2014-05-21

This advisory discloses a critical security vulnerability that we have found in Crucible and fixed in a recent version of Crucible.

- **Customers who have downloaded and installed Crucible** should upgrade their existing Crucible installations.

The vulnerability affects Crucible version 3.x.

Atlassian is committed to improving product security. We [fully support the reporting of vulnerabilities](#) and we appreciate it when people work with us to identify and solve the problem.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

Administrator password reset

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in [Severity Levels of Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

An unauthenticated user is able to set the admin password of Crucible to any value, gaining admin access to the Crucible instance as a result.

The vulnerability affects Crucible version 3.x. Versions earlier than 3.0 are not vulnerable. The vulnerability has been fixed in recent releases 3.0.4, 3.1.7, 3.2.5, 3.3.4, 3.4.4.

The issue is tracked in [CRUC-6810](#) - Administrator password reset **CLOSED**.

Risk Mitigation

If you are unable to upgrade your Crucible server you can do the following as a **temporary workaround**:

- Disallow external HTTP(S) access to your Crucible server.

Fix

This vulnerability can also be fixed by upgrading Crucible to one of the following versions: 3.0.4, 3.1.7, 3.2.5, 3.3.4, 3.4.4. You can download these versions of Crucible from the [download center](#). There are no patches available.

The [Security Patch Policy](#) describes when and how we release security patches and security upgrades for our products.

FishEye and Crucible Security Advisory 2014-02-26

This advisory details a critical security vulnerability that we have found in Crucible and fixed in recent versions of Crucible.

- **Customers who have downloaded and installed Crucible** should upgrade their existing Crucible installations to fix this vulnerability.
- **Atlassian OnDemand customers** are not affected because OnDemand does not include Crucible.

The vulnerability affects all versions of Crucible up to and including 3.1.5 and 2.10.7.

Atlassian is committed to improving product security. We [fully support the reporting of vulnerabilities](#) and we appreciate it when people work with us to identify and solve the problem.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com>.

User privilege escalation

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in [Severity Levels of Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have identified and fixed a vulnerability in Crucible which allowed unauthenticated users to commit actions on behalf of any other authorised user. In order to exploit this vulnerability, an attacker requires access to your Crucible web interface.

A Crucible server is only vulnerable if it has been configured to be a part of an Application link with [Trusted Applications authentication](#). Unless you chose OAuth authentication when configuring previously created links, Trusted Applications was used by default.

The vulnerability affects all supported versions of Crucible up to and including 3.1.5 and 2.10.7. It has been fixed in 3.2.0, 3.1.6 and 2.10.8. The issue is tracked in [CRUC-6662 - Privilege escalation](#) **CLOSED**.

Risk Mitigation

If you are unable to upgrade your Crucible server you can do the following as a **temporary workaround**:

- Block access to your Crucible server web interface from untrusted networks, such as the Internet.
- Remove any Application links that use Trusted Applications authentication and re-create them using OAuth.

Fix

This vulnerability can be fixed by upgrading Crucible. If you have any questions, please raise a support request at <http://support.atlassian.com>. We recommend upgrading.

The [Security Patch Policy](#) describes when and how we release security patches and security upgrades for our products. Only security upgrades are released for Crucible. Patches are not released for Crucible.

Upgrading Crucible

Upgrade to Crucible 3.2.0, 3.1.6, 2.10.8 or a later version, which fixes this vulnerability. For a full description of these releases, see the [Crucible Release Notes](#).

Patches

Binary patches are not available for this advisory. You need to either install one of the patch releases or apply recommended temporary workarounds.

FishEye and Crucible Security Advisory 2013-07-16

This advisory discloses security vulnerabilities that we have found in FishEye/Crucible and fixed in a recent version of FishEye/Crucible.

- **Customers who have downloaded and installed FishEye/Crucible** should upgrade their existing FishEye/Crucible installations to fix this vulnerability.
- **Atlassian OnDemand customers** are not affected by any of the issues described in this advisory.

Atlassian is committed to improving product security.

The vulnerability listed in this advisory is a **vulnerability in a third-party framework - Struts 2 / WebWork 2** that is used by FishEye/Crucible. The vulnerability has been independently discovered by Atlassian and reported to the Struts maintainers.

More details about the underlying Struts vulnerability CVE-2013-2251 are available at the [CVE database](#) and in the [Struts advisory](#).

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

OGNL injection in WebWork 2

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have fixed a vulnerability in WebWork 2, which is a part of the Struts web framework. In specific circumstances, attackers can use this vulnerability to execute Java code of their choice on systems that use these frameworks. In the case of FishEye/Crucible, the attacker needs to be able to access the FishEye/Crucible web interface. A valid user account is not required to exploit this vulnerability.

Customers should be advised that this affects all versions of FishEye/Crucible, except FishEye/Crucible OnDemand, FishEye/Crucible 2.10.6 and FishEye/Crucible 3.0.2 or later. The issue can be tracked here:  [FE-4729](#) - Webwork 2 code injection vulnerability 

Risk Mitigation

If you are unable to upgrade or patch your FishEye or Crucible server: as a **temporary workaround**, you can do the following:

- Block access to all URLs on a Web Application Firewall or a reverse proxy that contain any of the following strings: "redirect:", "action:" or "redirect-action:" strings. A partial example for an nginx server is below. Note that the example only covers the "redirect:" prefix and does not account for any URL encoding that may be present.

```
location ~* ^/<path to your FishEye>/ {
    if ($args ~* "redirect:") {
        return 403;
    }
    proxy_pass http://$host.internal$request_uri;
}
```

or

- Block access to your FishEye or Crucible server from untrusted networks, such as the Internet.

Fix

This vulnerability can be fixed by upgrading FishEye/Crucible. There are no patches available for this vulnerability — for any questions, please raise a support request at <http://support.atlassian.com/>.

The [Security Patch Policy](#) describes when and how we release security patches and security upgrades for our products.

Upgrading FishEye/Crucible

The fix versions for this vulnerability are described in the 'Description' section above.

We recommend that you upgrade to the latest version of FishEye/Crucible, if possible. For a full description of the latest version of FishEye/Crucible, see the release notes: [FishEye releases](#) / [Crucible release notes](#). You can download the latest version of FishEye/Crucible from the download centre: [FishEye only](#) / [FishEye and Crucible](#). Additionally, if you have large repositories you may want to take a look at the our guide to [Avoiding Long Reindex Times When Upgrading](#).

FishEye and Crucible Security Advisory 2012-08-21

This advisory discloses security vulnerabilities that we have found in FishEye and/or Crucible and fixed in a recent version of FishEye and/or Crucible.

- **Customers who have downloaded and installed FishEye and/or Crucible** should upgrade their existing FishEye and/or Crucible installations to fix this vulnerability.
- **Atlassian OnDemand and JIRA Studio customers** are not affected by any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerability listed in this advisory has been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [Elevation of privileges vulnerability](#)

Elevation of privileges vulnerability

Severity

Atlassian rates the severity level of this vulnerability as **Medium**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have identified and fixed a vulnerability in FishEye and Crucible that results from behaviour of certain

third-party frameworks used in FishEye and Crucible. This vulnerability allows any attacker to:

- Set the FishEye and Crucible instance to allow anonymous access
- Set the FishEye and Crucible instance to allow anonymous signup

All versions of FishEye and Crucible up to and including 2.7.14 are affected by this vulnerability. The vulnerability is fixed in FishEye and/or Crucible 2.8.0 and later. This issue can be tracked at

• [FE-4222](#) - FishEye privilege escalation vulnerability **CLOSED** and

• [CRUC-6188](#) - Crucible privilege escalation vulnerability **CLOSED**.

The table below describes the FishEye and/or Crucible versions and the specific functionality affected by the vulnerabilities.

FishEye and/or Crucible Vulnerability	Affected versions	Fixed Version	Issue Tracking
Elevation of privileges	2.5.x or earlier	2.5.9	FE-4222 - FishEye privilege escalation vulnerability CLOSED CRUC-6188 - Crucible privilege escalation vulnerability CLOSED
	2.6.x	2.6.9	
	2.7.x	2.7.15, 2.8.0	

Note: The email we sent out wrongly states that fixed versions are 2.5.8 and 2.6.7. FishEye and Crucible development team apologise for the mistake.

Risk Mitigation

If you cannot upgrade immediately, you can disable all access from the public Internet to your FishEye and/or Crucible instance to prevent external attacks.

Fix

Upgrade

The vulnerabilities and fix versions are described in the 'Vulnerability' section above.

We recommend that you upgrade to the latest version of FishEye and/or Crucible, if possible. For a full description of the latest version of FishEye and Crucible, see the [FishEye release notes](#) and [Crucible release notes](#). You can download the latest version of FishEye and Crucible from the [FishEye download centre](#) and [Crucible download centre](#).

There are no patches available.

FishEye and Crucible Security Advisory 2012-05-17

This advisory discloses a **critical** security vulnerability that exists in all versions of FishEye and Crucible up to and including 2.7.11.

- **Customers who have downloaded and installed FishEye or Crucible** should upgrade their existing FishEye and Crucible installations to fix this vulnerability.
- **Enterprise Hosted customers** need to request an upgrade by raising a support request at <http://support.atlassian.com> in the "Enterprise Hosting Support" project.
- **JIRA Studio and Atlassian OnDemand customers** are not affected by any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerability listed in this advisory has been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [Critical XML Parsing Vulnerability](#)
 - [Severity](#)
 - [Description](#)
 - [Risk Mitigation](#)
 - [Fix](#)

Critical XML Parsing Vulnerability

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, medium or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have identified and fixed a vulnerability in FishEye and Crucible that results from the way third-party XML parsers are used in FishEye and Crucible.

This vulnerability allows an attacker to:

- execute denial of service attacks against the FishEye and Crucible server, and
- read all local files readable to the system user under which FishEye and Crucible runs.

An attacker does not need to have an account with the affected FishEye or Crucible server to exploit this vulnerability.

All versions of FishEye and Crucible **up to and including 2.7.11** are affected by this vulnerability. This issue can be tracked here:  [FE-4016 - FishEye XML Vulnerability](#) CLOSED

Risk Mitigation

We recommend that you upgrade your FishEye and Crucible installation to fix this vulnerability.

Alternatively, if you are not in a position to upgrade immediately, you should do **all** of the following until you can upgrade. Please note, these measures will only limit the impact of the vulnerability, they will not mitigate it completely.

- Disable access to the Remote, SOAP and XML-RPC APIs, if these remote APIs are not required. Note that remote API access is disabled by default. See [enabling plugins](#) for instructions.
- Disable public access (such as [anonymous access](#) and [public signup](#)) to your FishEye or Crucible instance until you have applied the necessary upgrade.
- Ensure that your FishEye/Crucible system user is restricted as described in [best practices for configuring FishEye security](#).

Fix

Upgrade

Upgrade to FishEye and Crucible 2.7.12 or later which fixes this vulnerability. For a full description of these releases, see the [FishEye](#) and [Crucible](#) release notes. The following releases have also been made available to fix these issues in older FishEye and Crucible versions. You can download these versions from the [FishEye](#) and [Crucible](#) download centres.

- FishEye and Crucible 2.6.8 for FishEye and Crucible 2.6
- FishEye and Crucible 2.5.8 for FishEye and Crucible 2.5

Patches

There are no patches available for this vulnerability.

FishEye and Crucible Security Advisory 2012-01-31

This advisory discloses a **CRITICAL** security vulnerability that we have found in versions of FishEye and Crucible from 2.0 up to and including 2.7.8. You need to upgrade your existing FishEye and Crucible installations to fix these vulnerabilities. Enterprise Hosted customers should request an upgrade by raising a support request at <http://support.atlassian.com> in the "Enterprise Hosting Support" project. Neither FishEye nor Crucible in Studio and Atlassian OnDemand are vulnerable to any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerabilities listed in this advisory have been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [Code Injection Vulnerability](#)
 - [Severity](#)
 - [Description](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)

Code Injection Vulnerability

Severity

Atlassian rates the severity level of this vulnerability as **CRITICAL**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, medium or low.

Description

We have identified and fixed a code injection vulnerability in FishEye and Crucible caused by an underlying vulnerability in the **third-party** Webwork 2 framework. This vulnerability allows an attacker to run arbitrary Java code on a FishEye/Crucible server with user privileges of the FishEye/Crucible process. This vulnerability is a variant of a recently disclosed [Struts2 vulnerability](#). The vulnerability exists in FishEye and Crucible pages accessible only by users with administrative privileges. It can be exploited with use of social engineering, e.g. by having the administrator click on a specially crafted link.

The maintainer of the original library can be contacted at <http://struts.apache.org/>

Vulnerability

The table below describes the FishEye and Crucible versions and the specific functionality affected by the command injection vulnerability.

FishEye and Crucible Component	Affected FishEye and Crucible Versions	Fixed Version	Issue Tracking
WebWork 2	2.0 - 2.7.8	2.6.7 2.7.9	FE-3891

Risk Mitigation

We recommend that you upgrade your FishEye and Crucible installations to fix this vulnerability.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can restrict access to your instance of FishEye/Crucible by using a firewall.

Fix

FishEye and Crucible 2.6.7, 2.7.9 and later versions fix this issue. View the issue linked above for information

on fix versions. For a full description of the latest version of FishEye and Crucible, see the [FishEye](#) and [Crucible](#) release notes. You can download the latest versions from the [FishEye](#) and [Crucible](#) download centres. There are no patches available for these issues.

FishEye and Crucible Security Advisory 2011-11-22

This advisory discloses a number of security vulnerabilities that we have found and fixed in versions of FishEye/Crucible earlier than 2.5.7. You need to upgrade your existing FishEye and Crucible installations to fix these vulnerabilities. Enterprise Hosted customers should request an upgrade by raising a support request at <http://support.atlassian.com> in the "Enterprise Hosting Support" project. Neither JIRA Studio nor Atlassian OnDemand are vulnerable to any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerabilities listed in this advisory have been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [XSS Vulnerabilities](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerabilities](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Permission Verification Vulnerabilities](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)

XSS Vulnerabilities

Severity

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, medium or low. These vulnerabilities are **not** critical.

Risk Assessment

We have identified and fixed a number of stored cross-site scripting (XSS) vulnerabilities which affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). XSS vulnerabilities allow an attacker to embed their own JavaScript into a FishEye/Crucible page. You can read more about XSS attacks on [Wikipedia](#), at [The Web Application Security Consortium](#) and other places on the web.

Vulnerabilities

The table below describes the FishEye/Crucible versions and the specific functionality affected by the XSS vulnerabilities.

FishEye/Crucible Feature	Affected FishEye/Crucible Versions	Fixed Version	Issue Tracking
FishEye user profile - display name	2.5.4 and earlier	2.5.5	FE-3797
FishEye user profile - snippets in a user's comment	2.5.4 and earlier	2.5.5	FE-3798

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable [public signup](#) to your instance as a temporary mitigation until you have applied the upgrade. For tighter access control, you can restrict access to trusted [groups](#).

Fix

FishEye and Crucible 2.5.5 and later versions fix two these issues. View the issues linked above for information on fix versions. You can download the latest versions of FishEye and Crucible from the download centres ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

Permission Verification Vulnerabilities

Severity

Atlassian rates the severity level of these vulnerabilities as **medium**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, medium or low. These vulnerabilities are **not** critical.

Risk Assessment

We have identified and fixed two permission verification bugs which affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). These vulnerabilities allow users to view metadata for changesets and reviews, from repositories/projects that they do not have access to, via tooltips.

Vulnerability

The table below describes the FishEye/Crucible versions and the specific functionality affected by the vulnerabilities.

FishEye/Crucible Feature	Affected FishEye/Crucible Versions	Fixed Version	Issue Tracking
FishEye changeset tooltips	FishEye 2.4.6 to 2.5.6	FishEye 2.5.7	FE-3811
Crucible review tooltips	Crucible 2.4.6 to 2.5.6	Crucible 2.5.7	CRUC-5811

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can [disable anonymous access](#) to your instance. Logged-in users will still be able to view metadata that they do not have permission to view, but anonymous users will be prevented from accessing this information.

Fix

FishEye and Crucible 2.5.7 and later versions fix these two issues. View the issues linked above for information on earlier fix versions for each issue. For a full description of this release, see the [FishEye 2.5 changelog](#) and [Crucible 2.5 Changelog](#). You can download the latest version of FishEye and Crucible from the download centres ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

FishEye and Crucible Security Advisory 2011-05-16

This advisory announces a number of security vulnerabilities that we have found and fixed in recent versions of

FishEye/Crucible. You need to upgrade your existing FishEye/Crucible installations to fix these vulnerabilities. Enterprise Hosted customers should request an upgrade by filing a ticket at <http://support.atlassian.com>. JIRA Studio is not vulnerable to any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerabilities listed in this advisory have been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [XSS Vulnerabilities](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)

XSS Vulnerabilities

Severity

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a number of cross-site scripting (XSS) vulnerabilities which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). XSS vulnerabilities potentially allow an attacker to embed their own JavaScript into a FishEye/Crucible page. You can read more about XSS attacks at cgisecurity.com, [The Web Application Security Consortium](#) and other places on the web.

Vulnerability

The table below describes the FishEye/Crucible versions and the specific functionality affected by the XSS vulnerabilities.

FishEye/Crucible Feature	Affected FishEye/Crucible Versions	Issue Tracking
Crucible snippets	Crucible 2.4.5 to 2.5.0	CRUC-5734
Crucible author mapping	Crucible 2.4.5 to 2.5.0	CRUC-5735
Crucible changeset comments in search results	Crucible 2.3.0 to 2.5.0	CRUC-5736
Crucible comments search	Crucible 2.2.6 to 2.5.0	CRUC-5737
FishEye/Crucible dashboard - review activity	FishEye/Crucible 2.2.8 to 2.5.2	FE-3031
FishEye/Crucible reviews list	FishEye/Crucible 2.2.8 to 2.5.2	FE-3032

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable [public signup](#) to your instance until you have applied the upgrade. For even tighter control, you could restrict access to trusted [groups](#).

Fix

FishEye/Crucible 2.5.4 fixes all of these issues. View the issues linked above for information on earlier fix versions for each issue. For a full description of this release, see the [FishEye 2.5 changelog](#) and [Crucible 2.5 Changelog](#). You can download the latest version of FishEye/Crucible from the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

Our thanks to **Marian Ventuneac** of <http://www.ventuneac.net>, who reported [FE-3031](#) and [FE-3032](#). We fully support the [reporting of vulnerabilities](#) and we appreciate it when people work with us to identify and solve the problem.

FishEye and Crucible Security Advisory 2011-01-12

This advisory announces a number of security vulnerabilities that we have found and fixed in recent versions of FishEye/Crucible. You need to upgrade your existing FishEye/Crucible installations to fix these vulnerabilities. Enterprise Hosted customers should request an upgrade by filing a ticket at <http://support.atlassian.com>. JIRA Studio is not vulnerable to any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerabilities listed in this advisory have been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [XSS Vulnerabilities](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Administration password logged in debug log](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Review comment search returns comments that a user has no permission to view](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Anonymous global access exposes entire user list](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)

XSS Vulnerabilities

Severity

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a number of cross-site scripting (XSS) vulnerabilities which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). XSS vulnerabilities potentially allow an attacker to embed their own JavaScript into a FishEye/Crucible page. You can read more about XSS attacks at cgisecurity.com, The Web Application Security Consortium and other places on the web.

Vulnerability

The table below describes the FishEye/Crucible versions and the specific functionality affected by the XSS vulnerabilities.

FishEye/Crucible Feature	Affected FishEye/Crucible Versions	Issue Tracking
Crucible review – linked JIRA issue	Crucible 2.0.1 – 2.3.7	CRUC-5307
Crucible email reviews	Crucible 2.2.0 – 2.4.0	CRUC-5308
Crucible review reload	Crucible 2.2.0 – 2.4.2	CRUC-5309
Crucible edit review details screen	Crucible 2.2.0 – 2.3.7	CRUC-5345
FishEye repository configuration	FishEye 2.4.0	CRUC-5310
FishEye charts	FishEye 2.2.0 - 2.4.0	CRUC-5311
FishEye/Crucible code macro	FishEye/Crucible 2.2.0 – 2.4.0	CRUC-5312
FishEye/Crucible changeset page heading	FishEye/Crucible 2.3.2 – 2.4.0	CRUC-5313

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable [public signup](#) to your instance until you have applied the upgrade. For even tighter control, you could restrict access to trusted [groups](#).

Fix

FishEye/Crucible 2.4.4 fixes all of these issues. View the issues linked above for information on earlier fix versions for each issue. For a full description of this release, see the [FishEye 2.4 changelog](#) and [Crucible 2.4 Changelog](#). You can download the latest version of FishEye/Crucible from the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

Administration password logged in debug log

Severity

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a vulnerability in the FishEye/Crucible logging which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). This vulnerability allows administrator passwords to be logged in clear text when debug logging is enabled.

Vulnerability

This vulnerability affects FishEye and Crucible 2.2.0 to 2.4.0.

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable access logging. See [Enabling Access Logging in FishEye](#) and [Enabling Access Logging in Crucible](#). You can also apply file restrictions to your log files. Note, this issue only occurs when DEBUG logging is turned on (off by default) when an administrator logs in.

Fix

FishEye/Crucible 2.4.2 and later fix this issue. For a full description of this release, see the [FishEye 2.4 changelog](#) and [Crucible 2.4 Changelog](#). You can download the latest version of FishEye/Crucible from the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix this vulnerability. You must upgrade your FishEye/Crucible installation.

Review comment search returns comments that a user has no permission to view

Severity

Atlassian rates the severity level of these vulnerabilities as **medium**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a vulnerability in the Crucible review comment search which may affect Crucible instances, including publicly available instances (that is, internet-facing servers). This vulnerability allows review comments to be displayed for projects that are not publicly viewable.

Vulnerability

This vulnerability affects Crucible 2.2.0 to 2.4.3.

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Fix

FishEye/Crucible 2.2.5, 2.3.8 and 2.4.4 fix this issue. For a full description of this release, see the [FishEye 2.4 changelog](#) and [Crucible 2.4 Changelog](#). You can download these versions of FishEye/Crucible via the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix this vulnerability. You must upgrade your FishEye/Crucible installation.

Anonymous global access exposes entire user list

Severity

Atlassian rates the severity level of these vulnerabilities as **medium**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a vulnerability in the FishEye/Crucible anonymous global access which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). This vulnerability exposes the user list (usernames and emails) of a FishEye/Crucible instance for access when anonymous global access is enabled.

Vulnerability

This vulnerability affects FishEye and Crucible 2.2.0 to 2.4.3.

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable global anonymous access. See [Configuring Anonymous Access](#).

Fix

FishEye/Crucible 2.2.6, 2.3.8 and 2.4.4 fix this issue. For a full description of this release, see the [FishEye 2.4 changelog](#) and [Crucible 2.4 Changelog](#). You can download these versions of FishEye/Crucible via the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix this vulnerability. You must upgrade your FishEye/Crucible installation.

Crucible Security Advisory 2010-06-16

The 2.3.3 release of Crucible contains some security related fixes, which are part of the shared FishEye architecture. The following information for FishEye applies equally to Crucible.

The [Crucible Download Centre](#) has the updates for Crucible.

In this advisory:

- [Remote Code Exploit Vulnerability](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Download Patches for Earlier FishEye / Crucible Versions](#)
 - [Patch for FishEye / Crucible 2.3.2](#)
 - [Patch for FishEye / Crucible 2.2.3](#)

Remote Code Exploit Vulnerability

Severity

Atlassian rates this vulnerability as **critical**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank a vulnerability as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a remote code exploit vulnerability which affects FishEye and Crucible instances.

Vulnerability

This vulnerability allows a motivated attacker to call remote code on the host server.

All versions of FishEye/Crucible up to version 2.3.2 are affected by this vulnerability.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.3.2.	2.3.3 update, also available as patches for 2.3.2 and 2.2.3.	This vulnerability allows a motivated attacker to call remote code on the host server.	Critical

This vulnerability has been discovered in XWork by OpenSymphony, a command pattern framework which is used by FishEye and Crucible.

About the XWork Framework:

- See the [OpenSymphony XWork page](#) for more information about XWork.

Risk Mitigation

We strongly recommend either upgrading or patching your FishEye/Crucible installation to fix this vulnerability. Please see the 'Fix' section below.

Fix

These issues have been fixed in FishEye 2.3.3 (see the [changelog](#)), which you can download from the [download centre](#).

It has also been fixed in Crucible 2.3.3 (see the [changelog](#)), which you can download from the [download centre](#).

Later versions will include protection from this vulnerability.

This fix is also provided as a patch for FishEye/Crucible 2.3.2 and 2.2.3, which you can download from [links on this page](#). Customers on earlier point versions of FishEye/Crucible will have to upgrade to version 2.3.2 or 2.2.3 before applying the patch. Atlassian recommends you upgrade to FishEye/Crucible 2.3.3.

Download Patches for Earlier FishEye / Crucible Versions

These patch releases contain security fixes, which apply to the shared FishEye architecture that is the basis of both FishEye and Crucible.

Please note that these patches are for specific point versions of FishEye (2.3.2 and 2.2.3). If you are running an earlier version than these, you will need to upgrade to a version specifically addressed by one of these patches. Atlassian strongly recommends that you upgrade to [FishEye 2.3.3 / Crucible 2.3.3](#) or later.

[MD5 checksums](#) are provided to allow verification of the downloaded files.

Patch for FishEye / Crucible 2.3.2

File	FishEye / Crucible Version	Release Date	MD5 Checksum
fisheye-crucible-2.3.2-patch1.zip	2.3.2	16th June, 2010	6fe98db821a6d26f26907688af2ccd84

Patch for FishEye / Crucible 2.2.3

File	FishEye / Crucible Version	Release Date	MD5 Checksum
------	----------------------------	--------------	--------------

fisheye-crucible-2.2.3-patch1.zip	2.2.3	16th June, 2010	6fe98db821a6d26f26907688af2ccd84
---	-------	-----------------	----------------------------------

Our thanks to **Meder Kydyraliev** of the **Google Security Team** who discovered this vulnerability. Atlassian [fully supports the reporting of vulnerabilities](#) and appreciates it when people work with Atlassian to identify and solve the problem.

Crucible Security Advisory 2010-05-04

The 2.2.3 release of Crucible contains some security related fixes, which are part of the shared FishEye architecture. The following information for FishEye applies equally to Crucible.

The [Crucible Download Centre](#) has the updates for Crucible.

In this advisory:

- [Admin Escalation Vulnerability](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [XSS Vulnerabilities in FishEye](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Prevention of Brute Force Attacks](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Changed Behaviour in FishEye](#)
- [Download Patches for Earlier FishEye / Crucible Versions](#)
 - [Patch for FishEye / Crucible 2.1.4](#)
 - [Patch for FishEye / Crucible 2.0.6](#)
 - [Patch for FishEye 1.6.6](#)
 - [Patch for Crucible 1.6.6](#)

Admin Escalation Vulnerability

Severity

Atlassian rates this vulnerability as **critical**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank a vulnerability as critical, high, moderate or low.

Risk Assessment

We have identified and fixed an admin escalation vulnerability, which affects FishEye instances. This vulnerability has security implications and is especially important for anyone running publicly accessible instances of FishEye.

Vulnerability

This vulnerability allows a motivated attacker to perform admin actions.

All versions of FishEye from version 1.6.0-beta2 (including 1.6.0) through to 2.2.1 are affected by these admin escalation vulnerabilities.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.2.1	2.2.3 update, also available as patches for certain versions, listed on this page .	This vulnerability allows a motivated attacker to perform admin actions.	Critical

Risk Mitigation

We strongly recommend either upgrading or patching your FishEye installation to fix this vulnerability. Please see the 'Fix' section below.

Note: If you are an Atlassian JIRA Studio customer, we have assessed that your system is secure and implemented additional protections for it.

Fix

These issues have been fixed in FishEye 2.2.3 (see the [changelog](#)), which you can download from the [download centre](#). Later versions will include protection from this vulnerability.

This fix is also provided as a patch for FishEye 2.1.4, 2.0.6 and 1.6.6, which you can download from [this page](#). Customers on earlier point versions of FishEye will have to upgrade to version 2.1.4, 2.0.6 or 1.6.6 before applying the patch. We recommend you upgrade to FishEye 2.2.3.

XSS Vulnerabilities in FishEye

Severity

Atlassian rates these vulnerabilities as **critical**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank a vulnerability as critical, high, moderate or low.

Risk Assessment

We have identified and fixed several cross-site scripting (XSS) vulnerabilities in FishEye, which may affect FishEye instances. These vulnerabilities have security implications and are especially important for anyone running publicly accessible instances of FishEye.

- The attacker might take advantage of the vulnerability to steal other users' session cookies or other credentials, by sending the credentials back to the attacker's own web server.
- The attacker's text and script might be displayed to other people viewing a FishEye page. This is potentially damaging to your company's reputation.

You can read more about XSS attacks at [cgisecurity](#), [CERT](#) and other places on the web.

Vulnerability

All versions of FishEye are affected by these XSS vulnerabilities.

Affected FishEye Versions	Fix Availability	More Details	Severity
---------------------------	------------------	--------------	----------

All versions up to and including 2.2.1	2.2.3 only	An attacker could take advantage of this vulnerability to steal other users' session cookies or other credentials, or the attacker's text and script might be displayed to other people viewing a FishEye page.	Critical
--	------------	---	----------

Risk Mitigation

We strongly recommend upgrading your FishEye installation to fix these vulnerabilities. Please see the 'Fix' section below.

Fix

These issues have been fixed in FishEye 2.2.3 (see the [changelog](#)), which you can download from the [download centre](#).

Prevention of Brute Force Attacks

Severity

Atlassian rates this vulnerability as **moderate**, according to the scale published in [Severity Levels for Security Issues](#).

Risk Assessment

We have improved the security of the following areas in FishEye:

- Prevention of brute force attacks by requiring users to solve a [CAPTCHA](#) test after a maximum number of repeated login attempts.

Vulnerability

We have identified and fixed a problem where FishEye allows an unlimited number of repeated login attempts, potentially opening FishEye to a brute force attack. Details of this improvement are summarised below.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.2.1	2.2.3 only	FishEye allows an unlimited number of login attempts. This makes FishEye vulnerable to a brute force attack.	Moderate

Risk Mitigation

We recommend that you upgrade your FishEye installation to fix these vulnerabilities. Please see the 'fix' section [below](#).

You can also prevent brute force attacks by following our guidelines on [using Fail2Ban to limit login attempts](#).

Fix

This issue has been fixed in FishEye 2.2.3 (see the [changelog](#)). Later versions will include protection from this vulnerability. You can download FishEye 2.2.3 from the [download centre](#).

Changed Behaviour in FishEye

In order to fix these issues, we have changed FishEye's behaviour as follows:

- After three consecutive failed login attempts, FishEye will display a [CAPTCHA](#) form asking the user to enter a given word when attempting to log in again. This will prevent brute force attacks via the login screen. The number of failed attempts needed to trigger the [CAPTCHA](#) testing is configurable. For more information, see the documentation for [Brute force login protection](#).

In addition, after three consecutive failed login attempts via the FishEye remote API, an error message will be returned. Human intervention will then be required to reset that login account, i.e. solve the [CAPTCHA](#) test via the login screen.

Download Patches for Earlier FishEye / Crucible Versions

These patch releases contain security fixes, which apply to the shared FishEye architecture that is the basis of both FishEye and Crucible.

These patches fix the Admin Escalation vulnerability only. Please note that these patches are for specific older point versions of FishEye (2.1.4, 2.0.6 or 1.6.6). If you are running an earlier version than these, you will need to upgrade to a version specifically addressed by one of these patches. To update a more recent version of the product (2.1.5 through 2.2.1), please upgrade to FishEye 2.2.3 or later. Atlassian strongly recommends that you [upgrade to FishEye 2.2.3](#) or later.

[MD5 checksums](#) are provided to allow verification of the downloaded files.

Patch for FishEye / Crucible 2.1.4

File	FishEye / Crucible Version	Release Date	MD5 Checksum
fisheye-crucible-2.1.4-patch1.zip	2.1.4	4th May, 2010	6062fa2e1ad93729527357fb97b0d2ea

Patch for FishEye / Crucible 2.0.6

File	FishEye / Crucible Version	Release Date	MD5 Checksum
fisheye-crucible-2.0.6-patch1.zip	2.0.6	4th May, 2010	6aae75e2a5308121887bf9532473cf75

Patch for FishEye 1.6.6

File	FishEye Version	Release Date	MD5 Checksum
fisheye-1.6.6-patch1.zip	1.6.6	4th May, 2010	210ef3358aff83861733f8f22d331d7e

Patch for Crucible 1.6.6

File	Crucible Version	Release Date	MD5 Checksum
crucible-1.6.6-patch1.zip	1.6.6	4th May, 2010	48e8e8ada0ddb3fc8671459051df1120

 To acquire all of the fixes on this page, upgrade to [FishEye 2.2.3](#), which you can download from the [download centre](#).

Crucible FAQ

Crucible FAQ

Answers to frequently asked questions about configuring and using Crucible.

- [Crucible Resources](#)
- [General FAQs](#)
 - [Can I deploy Crucible or FishEye as a WAR?](#)
 - [Does Crucible support SSL \(HTTPS\)?](#)
 - [How do I force reviews to include SVN property changes?](#)
 - [How to Automate Daily Crucible Backups](#)
- [Licensing FAQ](#)
 - [What happens if I decide to stop using FishEye with Crucible](#)
 - [Do I need a FishEye licence to run Crucible?](#)
 - [Updating your Crucible license](#)
- [Support Policies](#)
 - [Bug Fixing Policy](#)
 - [New Features Policy](#)
 - [Security Bugfix Policy](#)
- [Troubleshooting](#)
 - [JIRA Integration Issues](#)
 - [Problems with very long comments and MySQL migration](#)
- [Contributing to the Crucible documentation](#)
- [Glossary](#)
 - [approve](#)
 - [authors in Crucible](#)
 - [code review](#)
 - [comment](#)
 - [creator](#)
 - [defect](#)
 - [moderator](#)
 - [participant](#)
 - [permission scheme](#)
 - [permissions in Crucible](#)
 - [projects in Crucible](#)
 - [review duration](#)
 - [reviewer](#)
 - [role](#)
 - [state](#)
 - [statement of objective](#)
 - [users in Crucible](#)
- [Collecting analytics in Crucible](#)

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

- [FishEye documentation](#)
- [FishEye FAQs](#)
- [Top Evaluator Questions](#)
 - [Can Crucible add support for new repositories?](#)
 - [Can I purchase Crucible on it's own?](#)
 - [Can I trial Crucible without FishEye?](#)
 - [How can I do reviews from the file system?](#)
 - [How does Crucible help enforce compliance and auditability?](#)
 - [How do I convince my team of the benefits of code review?](#)
 - [How do I do pre-commit reviews?](#)
 - [How do I raise defects in JIRA?](#)
 - [How do I review patch diffs?](#)
 - [What user permissions and review security is available?](#)

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

Crucible Resources

Resources for Evaluators

- [Free Trial](#)
- [Feature Tour](#)

Resources for Administrators

- [Crucible Knowledge Base](#)
- [Crucible FAQ](#)
- [Guide to Installing an Atlassian Integrated Suite](#)
- [The big list of Atlassian gadgets](#)

Downloadable Documentation

- [Crucible documentation in PDF, HTML or XML formats](#)

Plugins

- [Crucible Developer Documentation](#)
- [Add-ons for Crucible](#)

Support

- [Atlassian Support](#)
- [Support Policies](#)

Forums

- [Crucible Forum](#)
- [Crucible Developers Forum](#)

Mailing Lists

- Visit <http://my.atlassian.com> to sign up for mailing lists relating to Atlassian products, such as technical alerts, product announcements and developer updates.

Feature Requests

- [Issue Tracker and Feature Requests for Crucible](#)

General FAQs

Crucible General FAQs
• Can I deploy Crucible or FishEye as a WAR? — Unfortunately FishEye and Crucible cannot be deployed as a WAR. • Does Crucible support SSL (HTTPS)? • How do I force reviews to include SVN property changes? • How to Automate Daily Crucible Backups — Configuring Crucible backups is easy.

Can I deploy Crucible or FishEye as a WAR?

Unfortunately FishEye and Crucible cannot be deployed as a WAR. FishEye has some special needs for performance reasons that are not easily supported on third-party containers. Whilst this is an often requested feature, there are no immediate plans to provide a WAR version of FishEye or FishEye+Crucible. However the

upcoming separate edition of Crucible (i.e. without FishEye) may at some stage be available as a WAR.

Does Crucible support SSL (HTTPS)?

Crucible does not have any built-in support for running over SSL via the HTTPS protocol. However, it is possible to setup a proxy web server to forward requests to Crucible. Please see the page on [Integrating with Other Web Servers](#).

How do I force reviews to include SVN property changes?

Subversion (SVN) allows you to store arbitrary name/value pairs, called 'SVN properties', in association with files and directories. You can use these properties to store metadata, and Subversion also uses them, for example to store where code is branched from.

These name/value pairs can only be changed as part of a changeset or commit in Subversion. As such, you will have changesets with purely changes to SVN properties, or changesets with a mix of textual changes and changes to SVN properties.

Crucible 2.6 introduced [review creation without metadata](#) to prevent SVN properties from being included in reviews. This functionality was on by default.

For Crucible 3.0 and later, this functionality is disabled by default, and affects the following actions:

- Creating a review — The **Create Review** link is disabled in the activity stream, dashboard and changeset page for any changeset where all the file revisions only differ by SVN properties.
- Adding content to a review — SVN property-only changesets cannot be added to the review from the 'Add Content' dialog. Changesets with a mix of SVN property changes and textual changes can be added to the review, however only the file revisions with textual changes are added. File revisions with both textual changes and SVN property changes are always added.
Note that you can explicit add a particular file revision to a review, regardless of whether it is a textual change or metadata-only change.

The prime motivation behind metadata checking is to prevent the creation of reviews with a large number of files which have no effective content changes. Sometimes this sort of problem is seen when there is some wholesale SVN property change, for example setting the line endings on all files (`svn:eo1-style`). In that respect, this is a performance setting as it avoids the creation of large reviews. The properties themselves are still stored and managed in FishEye. There is no major performance impact on including properties in reviews.

Enabling this functionality can be done by starting up Crucible with the following system property:

```
-Dcrucible.detect.metadata.revision.changes=true
```

Crucible 3.0+ Changes

In Crucible 3.0 and later the default value of this setting was changed from true to false. Due to the phased nature of FishEye indexing, changesets are available for review much earlier in their overall processing. In particular changesets are available for review prior to having their line count and diff information processed. In this state all changes look somewhat like metadata only changes. Reviews created in this state with metadata detection enabled will cause the changes to be excluded from the review.

How to Automate Daily Crucible Backups

Configuring Crucible backups is easy. To set daily Crucible backups, open the administration page, click **Backup** (under 'System' on the left), and simply follow the instructions set out on the [Backing up and restoring Crucible data](#) page.

Licensing FAQ

Crucible Licensing FAQ

- [What happens if I decide to stop using FishEye with Crucible](#) — Crucible can be run as an application alone, without FishEye. However, if you decide to stop using FishEye with Crucible, you will lose certain functionality and will need to make configuration changes.
- [Do I need a FishEye licence to run Crucible?](#) — FishEye and Crucible are separate products. They can be run separately, and they can also be run together.
- [Updating your Crucible license](#)

What happens if I decide to stop using FishEye with Crucible

Crucible can be run as an application alone, without FishEye. However, if you decide to stop using FishEye with Crucible, you will lose certain functionality and will need to make configuration changes.

On this page:

- [How do I run Crucible without FishEye?](#)
- [How is Crucible without FishEye different from using Crucible with FishEye?](#)
 - [Conducting Reviews](#)
 - [Viewing Repositories/Files](#)
 - [Charts](#)

How do I run Crucible without FishEye?

- **Have a valid Crucible license but not a FishEye license**
To run Crucible without FishEye you need to have a **valid Crucible license but not a FishEye license** or if you want to disable FishEye enter Disabled in the license field. Crucible will actually use a "light" mode of FishEye that comes pre-bundled and does not need to be installed separately. For more information on Crucible with light FishEye, see [How is Crucible without FishEye different from using Crucible with FishEye?](#) below.
- **No need to reconnect your repositories**
Any repositories that you have currently defined in FishEye will not be visible in Crucible after removing FishEye (no more Source tab). The repositories however, will continue to update as usual without any intervention. You can add additional repositories as described in the [FishEye documentation](#). Note, all [repositories supported in FishEye](#) are supported in light FishEye.
 *Legacy "lightSCM" plugins, like the [Crucible Subversion SCM plugin](#), will still work. However, the functionality will be limited compared to using Crucible with light FishEye. See the [Can I still use lightSCM plugins with Crucible?](#) section below for more information.*
- **You need to reindex your repositories after removing a FishEye license**
When you remove a FishEye license, and operate with only a Crucible license, you need to reindex those repositories that were originally indexed under a FishEye license. You do not need to remove those repositories, you just need to run a re-index to create an index which is compatible with your new license setup.

How is Crucible without FishEye different from using Crucible with FishEye?

The following changes in functionality will occur if you use Crucible without FishEye (i.e. use Crucible with "light" FishEye).

Conducting Reviews

- When using [iterative reviews](#) in Crucible, you will not be prompted when a new version of a file is available.

Viewing Repositories/Files

- Files and changesets displayed in activity streams (e.g. the dashboard activity stream) will not render as links to the relevant files/changesets.

- You will not be able to see your content roots and repositories associated with projects.
- You will no longer be able to see [repository lists](#) and [browse repositories](#) using the 'Source' tab.

Charts

- You will not be able to [view charts](#) or [code metrics](#).

Do I need a FishEye licence to run Crucible?

FishEye and Crucible are separate products. They can be run separately, and they can also be run together.

We recommend that you run Crucible together with FishEye. If you choose to run Crucible alone without FishEye, you will have access to your repositories via the "light" FishEye implementation bundled with Crucible. However, a number of FishEye's advanced features will not be available to you, including [pre-caching repository content](#) (for improved performance), the ability to search and browse through repositories and FishEye's activity graphs.

For more information, please read the following FAQ: [What happens if I decide to stop using FishEye with Crucible](#)

Updating your Crucible license

When you upgrade or renew your Crucible license, you will receive a new license key – you'll need to update your Crucible server with the new license.

Note that you can access your current license, or obtain a new license, by going to my.atlassian.com.

Related pages:

- [Crucible and FishEye](#)
- [What happens if I decide to stop using FishEye with Crucible](#)

To update your Crucible license key:

1. Log in to Crucible [Admin area](#).
2. Click **System Info** (under 'System Settings').
3. Click **Edit License** and paste your new license key into the appropriate text box.
4. Paste your new license into this box. Obtain a new license by clicking my.atlassian.com in the 'Information' section.
5. Click **Update**.

Support

Information

New licenses can be acquired from my.atlassian.com.

Your Server ID is **BMRI-3AS3-HMMD-PGXX**

Licenses

FishEye License	Evaluation
Crucible License	<pre> AAAA9g0ODAAoPeNptkE9rwzAMxe/+FIadPWa7hzXgQ5eELRAnpcmgH15Uo62G1A3+E9Zvv3Yt dBs9C IT03k8PPfS7RJvDRAWnfJ7NeMZnNO96Kp64JAUG4+0Y7cGp3CdjtwNuMlpOMCQ4T0nu8acpl Kl6e xjnTAHirurH2hp0AfvjiA3siOnAGy6/R+uMvqPgDbdJ+i779eA/og 2KctP4TnA0X2CIEIIFdzOAiXbCy5kO/YS+KtSLXIVMLjrJ3rQu2PJ1vSZd2ahTsVqKZzGXklzz n +R1Vdzb3I+6TN7sIOD/v3wDPH51STAtAhRYkbBDq2j1NpXxLp5ws8aCIhAgBwIvAJceVTPzGB f4w g5FWvNY7c5adD+EX02co </pre>

Support Policies

Welcome to the support policies index page. Here, you'll find information about how Atlassian Support can help you and how to get in touch with our helpful support engineers. Please choose the relevant page below to find out more.

- [Bug Fixing Policy](#)
- [New Features Policy](#)
- [Security Bugfix Policy](#)

To request support from Atlassian, please raise a support issue in our online support system. To do this, visit support.atlassian.com, log in (creating an account if need be) and create an issue under Crucible. Our friendly support engineers will get right back to you with an answer.

Bug Fixing Policy

Summary

- Atlassian Support will help with workarounds and bug reporting.
- Critical bugs will generally be fixed in the next maintenance release.
- Non critical bugs will be scheduled according to a variety of considerations.



Raising a Bug Report

Atlassian Support is eager and happy to help verify bugs — we take pride in it! Please open a support request in our [support system](#) providing as much information as possible about how to replicate the problem you are experiencing. We will replicate the bug to verify, then lodge the report for you. We'll also try to construct workarounds if they're possible.

Customers and plugin developers are also welcome to open bug reports on our issue tracking systems directly. Use the appropriate project on <http://jira.atlassian.com> to report bugs for Atlassian products.

When raising a new bug, you should rate the priority of a bug according to our [JIRA usage guidelines](#). Customers [should watch](#) a filed bug in order to receive e-mail notification when a "Fix Version" is scheduled for

release.

How Atlassian Approaches Bug Fixing

Maintenance (bug fix) releases come out more frequently than major releases and attempt to target the most critical bugs affecting our customers. The notation for a maintenance release is the final number in the version (ie the 1 in 3.0.1).

If a bug is critical (production application down or major malfunction causing business revenue loss or high numbers of staff unable to perform their normal functions) then it will be fixed in the next maintenance release provided that:

- The fix is technically feasible (i.e. it doesn't require a major architectural change).
- It does not impact the quality or integrity of a product.

For non-critical bugs, the developer assigned to fixing bugs prioritises the non-critical bug according to these factors:

- How many of our supported configurations are affected by the problem.
- Whether there is an effective workaround or patch.
- How difficult the issue is to fix.
- Whether many bugs in one area can be fixed at one time.

The developers responsible for bug fixing also monitor comments on existing bugs and new bugs submitted in JIRA, so you can provide feedback in this way. We give high priority consideration to [security issues](#).

When considering the priority of a non-critical bug we try to determine a 'value' score for a bug which takes into account the severity of the bug from the customer's perspective, how prevalent the bug is and whether roadmap features may render the bug obsolete. We combine this with a complexity score (i.e. how difficult the bug is). These two dimensions are used when developers self serve from the bug pile.

Further reading

See [Atlassian Support Offerings](#) for more support-related information.

New Features Policy

Summary

- We encourage and display customer comments and votes openly in our issue tracking system, <http://jira.atlassian.com>.
- We do not publish roadmaps.
- Product Managers review our most popular voted issues on a regular basis.
- We schedule features based on a variety of factors.
- Our [Atlassian Bug Fixing Policy](#) is distinct from this process.
- Atlassian provides consistent updates on the top 20 issues.

How to Track what Features are Being Implemented

When a new feature or improvement is scheduled, the 'fix-for' version will be indicated in the JIRA issue. This happens for the upcoming release only. We maintain roadmaps for more distant releases internally, but because these roadmaps are often pre-empted by changing customer demands, we do not publish them.

How Atlassian Chooses What to Implement

In every [major release](#) we *aim* to implement highly requested features, but it is not the only determining factor. Other factors include:

- **Customer contact:** We get the chance to meet customers and hear their successes and challenges at Atlassian Summit, Atlassian Unite, developer conferences, and road shows.
- **Customer interviews:** All product managers at Atlassian do customer interviews. Our interviews are not simply to capture a list of features, but to understand our customers' goals and plans.
- **Community forums:** There are large volumes of posts on [answers](#), of votes and comments on jira.atlassian.com.

[an.com](#), and of conversations on community forums like groups on LinkedIn.

- **Customer Support:** Our support team provides clear insights into the issues that are challenging for customers, and which are generating the most calls to support
- **Atlassian Experts:** Our [Experts](#) provide insights into real-world customer deployments, especially for customers at scale.
- **Evaluator Feedback:** When someone new tries our products, we want to know what they liked and disliked and often reach out to them for more detail.
- **In product feedback:** The [JIRA Issue Collectors](#) that we embed our products for evaluators and our Early Access Program give us a constant pulse on how users are experiencing our product.
- **Usage data:** Are customers using the features we have developed?
- **Product strategy:** Our long-term strategic vision for the product.
- Please read our [post on Atlassian Answers](#) for a more detailed explanation.

How to Contribute to Feature Development

Influencing Atlassian's release cycle

We encourage our customers to vote on issues that have been raised in our public JIRA instance, <http://jira.atlassian.com>. Please find out if your request already exists - if it does, vote for it. If you do not find it you may wish to create a new one.

Extending Atlassian Products

Atlassian products have powerful and flexible extension APIs. If you would like to see a particular feature implemented, it may be possible to develop the feature as a plugin. Documentation regarding the [plugin APIs](#) is available. Advice on extending either product may be available on the user mailing-lists, or at [Atlassian Answers](#).

If you require significant customisations, you may wish to get in touch with our [partners](#). They specialise in extending Atlassian products and can do this work for you. If you are interested, please [contact us](#).

Further reading

See [Atlassian Support Offerings](#) for more support-related information.

Security Bugfix Policy

See [Security @ Atlassian](#) for more information on our security bugfix policy.

Troubleshooting

Crucible Troubleshooting

- [JIRA Integration Issues](#)
- [Problems with very long comments and MySQL migration](#) — Affects Version

Crucible Troubleshooting

The most common cause of FishEye/Crucible issues is an incorrect [symbolic setup](#) (trunk/branch/tag) for Subversion repositories. If you are using Subversion and your initial index is taking forever, double-check that your symbolic setup matches your repository.

FishEye runs with the default Java heap of 64 megabytes. This is sometimes problematic for FishEye, especially for Subversion repositories during the initial scan. You can give FishEye's JVM more memory by setting the `FISHEYE_OPTS` [environment variable](#).

Starting Crucible with the [command line options](#) `--debug --debug-perf` will print a lot of information to Crucible's logs. This can give you an insight into what is happening and possibly where you are stuck. Attach these logs along with your `config.xml` to an Atlassian support ticket, to speed up your [support request](#).

JIRA Integration Issues

Users are mapped to their own accounts when using Trusted Applications.

If you (or the general account used for JIRA access, if not using Trusted Applications) do not have the

permissions to carry out the JIRA actions linked from Crucible, an error will occur. Depending on the error returned from JIRA, Crucible may not display the error correctly or display it at all, simply reporting that "An error has occurred". To investigate what the error was, you can access the Crucible debug log, named `fisheye-debug.log.YYYY-MM-DD` under the `dist.inst/var/log` folder of your Crucible installation. In the debug log, look for the date and time when your error took place. Here, you will be able to follow the links and see what error the JIRA instance was producing by clicking through to JIRA.

If you are using JIRA 4.0 you will not be able to create subtasks in versions of Crucible prior to 2.0.5. If you are affected by this bug, please upgrade to at least 2.0.6 (2.0.5 is affected by another bug [CRUC-2471](#)).

Problems with very long comments and MySQL migration

Affects Version

This issue was introduced in Crucible 2.0 and fixed in Crucible 2.1.

Issue Symptoms

There is a known issue with Crucible 2.0.x and very long comments when migrating your database to MySQL. In some circumstances, this might result in truncation of very long comments, causing data loss.

Depending on your MySQL configuration, you may see an error message like this while migrating to MySQL, causing the migration to fail:

```
2009-07-16 16:56:12,390 ERROR [ThreadPool1] fisheye.app
com.cenqua.crucible.actions.admin.database.DBEditHelper - doGet -
Database migration failed:
java.sql.BatchUpdateException: Data truncation: Data too long for column
'cru_message' at row 1
java.sql.BatchUpdateException: Data truncation: Data too long for column
'cru_message' at row 1
```

 You may not see the message if you are running MySQL with default settings.

For more information, see the [JIRA issue](#).

Workaround

If your data contains very long comments or review descriptions (longer than 21,845 multibyte unicode characters), consider avoiding use of MySQL until you can upgrade the product. Alternatively, use PostgreSQL or the default (built-in) HSQLDB database.

This issue is now resolved. This issue was introduced in Crucible 2.0 and fixed in Crucible 2.1.

Requesting Support

If you require assistance in resolving the problem, please [raise a support request](#) under the Crucible project.

Contributing to the Crucible documentation

Would you like to share your Crucible hints, tips and techniques with us and with other Crucible users? We welcome your contributions.

On this page:

- [Updating the documentation](#)
 - [Getting permission to update the documentation](#)
 - [Our style guide](#)
 - [How we manage community updates](#)
- [Contributing documentation in other languages](#)

Updating the documentation

Have you found a mistake in the documentation, or do you have a small addition that would be so easy to add yourself rather than asking us to do it? You can update the documentation page directly

Getting permission to update the documentation

Please submit the [Atlassian Contributor License Agreement](#).

Our style guide

Please read our short [guidelines for authors](#).

How we manage community updates

Here is a quick guide to how we manage community contributions to our documentation and the copyright that applies to the documentation:

- **Monitoring by technical writers.** The Atlassian technical writers monitor the updates to the documentation spaces, using RSS feeds and watching the spaces. If someone makes an update that needs some attention from us, we will make the necessary changes.
- **Wiki permissions.** We use wiki permissions to determine who can edit the documentation spaces. We ask people to sign the [Atlassian Contributor License Agreement](#) (ACLA) and submit it to us. That allows us to verify that the applicant is a real person. Then we give them permission to update the documentation.
- **Copyright.** The Atlassian documentation is published under a Creative Commons CC BY license. Specifically, we use a [Creative Commons Attribution 2.5 Australia License](#). This means that anyone can copy, distribute and adapt our documentation provided they acknowledge the source of the documentation. The CC BY license is shown in the footer of every page, so that anyone who contributes to our documentation knows that their contribution falls under the same copyright.

Contributing documentation in other languages

Have you written a guide to Crucible in a language other than English, or translated one of our guides? Let us know, and we will link to your guide from our documentation.

RELATED TOPICS

[Author Guidelines](#)

[Atlassian Contributor License Agreement](#)

Glossary

Code review terminology can be confusing as there are many different words for the concepts, roles and process. Crucible has adopted the following terms (click for definitions):

[approve](#)

[authors in Crucible](#)

[code review](#)

[comment](#)

[creator](#)

[defect](#)

[moderator](#)

participant

permission scheme

permissions in Crucible

projects in Crucible

review duration

reviewer

role

state

statement of objective

users in Crucible

approve

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

authors in Crucible

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

Note: to map your repository username to your FishEye/Crucible username, see [Changing your User Profile](#).

code review

Without prejudice to 'code inspection', 'peer review' or a myriad of other terms, Crucible uses the phrase *code review* for simplicity.

See [Getting Started](#).

comment

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

See [Commenting on reviews](#).

creator

The *creator* is the person who [creates the review](#). In most cases this person will also act as [moderator](#).

defect

A *defect* is a comment flagged as something that requires addressing and includes optional defect classifications.

See [Flagging Defects](#) and [Customising the defect classifications](#).

moderator

The *moderator* is the person responsible for [creating](#) the review, [approving](#) the review, determining when reviewing is finished, [summarising](#) the outcomes and [closing](#) the review. By default, the moderator is the [creator](#). See also [author](#), the person whose changes to the code are to be reviewed.

participant

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

permission scheme

A *permission scheme* assigns particular [permissions](#) to any or all of the following:

- Particular [Users](#).
- Particular [Groups](#).
- All logged-in users.
- [Anonymous Users](#)
- People in particular [Review Roles](#), such as:
 - [Author](#)
 - [reviewer](#)
 - [creator](#)
 - [moderator](#)

The scheme's permissions will apply to all reviews belonging to the [project\(s\)](#) with which the scheme is associated.

You can create as many permission schemes as you wish. Each permission scheme can be associated with many projects or just one project, allowing you to tailor appropriate permissions for individual projects as required.

See [Creating a permission scheme](#).

permissions in Crucible

A *permission* is the ability to perform a particular action in Crucible, e.g. 'Create Review'.

Permissions are assigned to particular users, groups or [review roles](#) by means of [permission schemes](#).

The following permissions are available:

Permission	Description	Default Assignees
Abandon	Ability to abandon (i.e. cancel) a review.	Creator Author Mode rator
Approve	Ability to approve a review (i.e. issue it to the reviewers).	Creator Author Mode rator
Close	Ability to close a review once it has been summarised.	Creator Author Reviewer Moderator
Comment	Ability to add or remove a comment to or from a review.	Creator Author Reviewer Moderator
Complete	Ability of a reviewer to change their individual review status to Complete.	Reviewer
Create	Ability to create a review.	All logged-in users
Delete	Ability to delete a review.	Creator Author Mode rator
Edit Review Details	Ability to edit a review's details and change the set of revisions being reviewed.	Creator Author Reviewer Moderator

Re-Open	Ability to re-open a closed or abandoned review.	Creator Author Reviewer Moderator
Recover	Ability to resurrect an abandoned (i.e. cancelled) review.	Creator Author Reviewer Moderator
Reject	Ability to reject a review submitted for approval (i.e. prevent it from being issued to reviewers).	Creator Author Mode rator
Submit	Ability to submit a review for approval (i.e. request that the review be issued to the reviewers).	Creator Author Mode rator
Summarise	Ability to summarise a review. (Normally this would be done after all reviewers have completed their review.)	
Uncomplete	Ability of a reviewer to change their individual review status from Complete to Uncomplete.	Reviewer
View	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users Creator Author Reviewer Moderator

projects in Crucible

A Crucible *project* provides a way to group and manage related [reviews](#) – typically reviews that are all involved with the same software project. A Crucible project allows you to

- define default [moderators](#), [authors](#) and [reviewers](#) for the reviews in that project.
- define which people are eligible to be [reviewers](#) for the reviews in that project.
- use [permission schemes](#) to restrict who can perform particular actions (e.g. 'Create Review') in that project.

Every Crucible review belongs to a project. Each project has a *name* (e.g. ACME Development) and a *key* (e.g. ACME). The project key becomes the first part of that project's *review keys*, e.g. ACME-101, ACME-102, etc:

By default, Crucible contains one project. This default project has the key 'CR' and the name 'Default Project'. See [Creating a project](#).

review duration

The *review duration* is the period of time for which a review will run.

See [Editing a project](#).

reviewer

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

role

See [participant](#).

state

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

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

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

A review may also be in the following states:

Abandoned	This happens when a review is deleted.
Rejected	Any reviews that a moderator has rejected.
Needs Fixing	This means that the review state is not understood by Crucible, and indicates a programming or data issue. The review moderator can move the review into a known state if this happens.

statement of objective

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

users in Crucible

A *user* is a person using Crucible.

Collecting analytics in Crucible

We are continuously working to make Crucible better. Data about how you use Crucible helps us do that. We have updated our Privacy Policy so that we may collect usage data automatically, unless you disable collection. The data we collect includes information about the systems on which your installation of Crucible is operating, the features you use in Crucible, and your use of common IT terminology within the product. For more details, see our [Privacy Policy](#), in particular the 'Analytics Information from Downloadable Products' section.

See also our [End User Agreement](#).

How to change data collection settings?

You can opt in to, or out of, data collection at any time. A Crucible admin can change the data collection settings by going to **Analytics** (under 'Global Settings') in the Crucible admin area.

How is data collected?

We use the Atlassian Analytics plugin to collect event data in Crucible. Analytics logs are stored locally and then periodically uploaded to a secure location.