User Documentation for FishEye 3.0
Contents

FishEye Documentation Home ................................................. 6
Getting started ................................................................ 6
Supported platforms ............................................................. 7
End of Support Announcements for FishEye ............................. 12
  End of Support Announcement for IBM ClearCase .................. 14
  End of Support Announcement for Internally Managed Repositories 14
Installing FishEye on Windows .................................................. 16
Running FishEye as a Windows service .................................... 19
Installing FishEye on Linux and Mac ....................................... 22
Starting to use FishEye ......................................................... 25
Configuring JIRA Integration in the Setup Wizard ..................... 30
Using FishEye .................................................................... 39
  Using the FishEye screens .................................................. 39
  Browsing through a repository ......................................... 41
  Searching FishEye ........................................................... 43
  Viewing a file ................................................................. 48
    Viewing file content ..................................................... 49
    Viewing a file history .................................................... 52
    Viewing the changelog ................................................... 53
  FishEye Charts ............................................................... 55
  Using favourites in FishEye ................................................ 59
  Changeset Discussions ..................................................... 60
  Viewing the commit graph for a repository ............................ 60
  Viewing People’s Statistics ............................................... 66
Using smart commits ............................................................. 67
Changing your user profile .................................................... 72
  Re-setting your password ................................................ 76
Pattern matching guide ....................................................... 77
Date Expressions Reference Guide ......................................... 78
EyeQL Reference Guide ....................................................... 79
FishEye Release Notes .......................................................... 84
  FishEye upgrade guide ..................................................... 88
  FishEye 3.0 release notes ................................................ 92
  FishEye 2.10 release notes ............................................... 97
  FishEye 2.10 changelog .................................................. 100
  FishEye 2.10 upgrade guide ............................................ 104
  FishEye 2.9 release notes ................................................ 105
  FishEye 2.9 changelog ................................................... 109
  FishEye 2.9 upgrade guide ............................................. 111
  FishEye 2.8 release notes ................................................ 113
  FishEye 2.8 changelog ................................................... 116
  FishEye 2.8 upgrade guide ............................................. 126
  Upgrading from FishEye 2.7 with existing ClearCase Repositories 127
  FishEye 2.8 EAP Release Notes ......................................... 128
  FishEye 2.7 Release Notes .............................................. 128
  FishEye 2.7 Changelog .................................................. 132
How to reset the Administration Page password in FishEye or Crucible .......... 245
How do I configure an Outbound Proxy Server for FishEye ......................... 245
How to remove Crucible from FishEye 2.x or later ..................................... 245
How to run FishEye or Crucible on startup on Mac OS X ............................. 246
Integration FAQ ......................................................................................... 247
How do I disable the FishEye tab panel for non-code projects? ....................... 247
How do I enable debug logging for the JIRA FishEye plugin? ......................... 247
How do I uninstall the JIRA FishEye plugin? ............................................. 248
How is the Crucible review tab panel for the FishEye JIRA plugin populated? .. 248
What do I do if I discover a bug with the JIRA FishEye plugin? ...................... 249
Licensing FAQ ......................................................................................... 249
Are anonymous users counted towards FishEye's licence limits? ...................... 249
Restrictions on FishEye Starter Licenses .................................................... 249
Repository Management FAQ ..................................................................... 251
Setting the root location for internal repositories ...................................... 251
Migrating repositories to a changed root location ...................................... 252
Repairing a broken hierarchy of forks ....................................................... 252
Permanent authentication for Git repositories over HTTP(S) ...................... 253
Subversion FAQ ......................................................................................... 254
Configuring Start Revision based on date ................................................... 255
Errors 'SEVERE assert' or 'Checksum mismatch' ........................................ 255
FishEye fails to connect to the Subversion repository after a short time of successful operation. 255
How can FishEye help with merging of branches in Subversion? .................. 256
Subversion Changeset Parents and Branches .............................................. 256
SVN Authentication Issues ........................................................................ 256
What are Subversion root and tag branches? .............................................. 257
Why do I need to describe the branch and tag structure for Subversion repositories? 257
Why don't all my tags show up in FishEye? ............................................... 258
Support Policies ....................................................................................... 260
Bug Fixing Policy ..................................................................................... 260
How to report a Security Issue ................................................................... 261
New Features Policy .................................................................................. 262
Patch Policy ............................................................................................. 263
Security Advisory Publishing Policy .......................................................... 263
Security Patch Policy ................................................................................ 264
Severity Levels for Security Issues ............................................................ 265
Troubleshooting ....................................................................................... 266
After I commit a change to my CVS repository, it takes a long time before it appears in FishEye. . 266
FishEye freezes unexpectedly ..................................................................... 267
Generating a Thread DumpExternally .......................................................... 267
I have installed FishEye, and the initial scan is taking a long time. Is this normal? 268
I have installed FishEye, but there is no data in the Changelog. ..................... 268
Initial scan and page loads are slow on Subversion .................................... 269
It seems that FishEye's HTTP Header is Too Small .................................... 269
JIRA Integration Issues ............................................................................. 269
Message 'org.tigris.subversion.javahl.ClientException svn Java heap space' .... 269
On my Red Hat Linux system, after running for several days FishEye freezes and does not accept any more connections. 269
Problems with very long comments and MySQL migration .......................... 269
URLs with encoded slashes don't work, especially in Author constraints ......... 270
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributing to the FishEye Documentation</td>
<td>270</td>
</tr>
<tr>
<td>Tips of the Trade</td>
<td>271</td>
</tr>
<tr>
<td>FishEye Documentation in Other Languages</td>
<td>272</td>
</tr>
<tr>
<td>FishEye Resources</td>
<td>273</td>
</tr>
<tr>
<td>Glossary</td>
<td>274</td>
</tr>
</tbody>
</table>
FishEye Documentation Home

Getting started

- Installing FishEye on Windows
- Installing FishEye on Linux and Mac
- Starting to use FishEye
- Configuring repositories
- Using smart commits
- Viewing the changelog

Administering

- All administration topics
- Supported platforms
- Managing users and groups in FishEye
- Environment variables
- Managing your repositories
- Tuning FishEye performance

Extending FishEye...

- Integrating with JIRA
- APIs and add-ons

Latest release notes and changelog
Upgrading FishEye

Getting started

FishEye lets you view the contents of your Source Code Management (SCM) repositories in your web browser.

You can:

- view changesets, revisions, branches, tags, diffs and annotations.
- search everything – file names, commit messages, authors, text as well as the source code.
- visualise how source changes were introduced, what changed, when it changed, where it was changed, and who changed it.
- track activity in your source code repository.
- link specific source with related JIRA issues, Crucible code reviews, and Bamboo builds.
• get real-time notifications on code activity via email, RSS, or OpenSocial dashboards.
• construct your own sophisticated queries with EyeQL and integrate the results with other tools using the FishEye API.

Watch the video overview of FishEye's features.

To get started with FishEye:
1. Install and start FishEye on either Windows, or Linux and Mac.
2. Work through Starting to use FishEye.
3. Tell FishEye about your repositories.
4. Set up users and groups.

Supported platforms

This page shows the supported platforms for FishEye 3.0.x and its minor releases.

Key: = Supported; = Not Supported

<table>
<thead>
<tr>
<th>Java Version</th>
</tr>
</thead>
</table>
FishEye requires **Java Runtime** (JDK or JRE), version as noted. Pre-release/Early access versions of the Java Runtime are *not supported*.

You can download a Java Runtime for Windows/Linux/Solaris. On Mac OS X, the JDK is bundled with the operating system.

We highly recommend that you use the Oracle JVM (or use the default Mac OS X JVM), as other implementations have not been tested.

Please note:

- Once you have installed the JDK, you must set the `JAVA_HOME` environment variable. See [Installing FishEye on Windows](#) or [Installing FishEye on Linux and Mac](#).
- 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.
- If you intend to run FishEye as a Windows Service, using the Java Service Wrapper, we recommend that you use the Java JDK rather than the JRE so as to take advantage of the `-server` parameter.
- You'll need the JDK for the JSP source download.

Note also that a bug in Java 1.6.0_29 and above will prevent a connection to an external SQL Server 2008 database without an additional workaround.
Although FishEye 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 choose to run FishEye in a VM, please ensure that you choose a VM with good IO throughput.

<table>
<thead>
<tr>
<th>Databases</th>
<th>MySQL Enterprise Server 5.x</th>
<th>MySQL Community Server 5.x</th>
<th>The FishEye built-in database, running HSQLDB, is somewhat susceptible to data loss during system crashes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostgreSQL</td>
<td>8.2, 8.3, 8.4</td>
<td></td>
<td>External databases (such as MySQL) are generally more resistant to data loss during a system crash.</td>
</tr>
<tr>
<td>Oracle</td>
<td>11g</td>
<td></td>
<td>See the FishEye Database documentation for further details.</td>
</tr>
<tr>
<td>HSQLDB</td>
<td>Bundled; for evaluation use only</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Web Browsers</th>
<th>8.0, 9.0</th>
<th>6.0 and 7.0 are not supported</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Internet Explorer</td>
<td>Latest stable version supported</td>
<td>3.6, 4.0</td>
<td></td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td></td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>Safari</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chrome</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subversion

Server: 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7  
Client: SVNKit (default) or native client with JavaHL 1.6

Note that FishEye does not support SVN 1.7.x with the native client.  
Version 1.8 support is currently being assessed. FishEye does not yet work for file:// access to repositories created with Subversion 1.8.

### CVS (and CVSNT)

All versions

### Perforce

Client version 2007.3 or later

Support for internal hosting of Git repositories by FishEye will end on August 14th 2013, after which Git internal hosting will be removed from newer versions of FishEye.

Please see [End of Support Announcement for Internally Managed Repositories](#).

### Git

1.7.1.1 or later

Support for internal hosting of Git repositories by FishEye will end on August 14th 2013, after which Git internal hosting will be removed from newer versions of FishEye.

Please see [End of Support Announcement for Internally Managed Repositories](#).

### Mercurial

1.5.1 or later

Mercurial 2.1 has a bug that makes it incompatible with FishEye. Please use Mercurial 2.1.1 or later. Mercurial 2.6.x is currently not compatible. You should restart FishEye after upgrading Mercurial.

### Hardware requirements

FishEye should ideally run on a dedicated server. The most important aspect for a large-repository deployment will be I/O speed. You definitely want a fast local HDD for FishEye's cache. Note that NFS and SAN are not supported.

<table>
<thead>
<tr>
<th>Component</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1.8GHz or higher, a single core is sufficient. More cores or higher GHz will result in better load-handling ability.</td>
</tr>
<tr>
<td>RAM</td>
<td>1GB minimum, 2GB will provide performance &quot;headroom&quot;. Your Java heap should be sized at 512MB with the FISHEYE_OPTS environment variable, adjustable up to 1024MB depending on performance.</td>
</tr>
</tbody>
</table>
I/O

FishEye’s input/output is an important element of its overall performance. If FishEye accesses your repository remotely, make sure that the throughput is maximum and the latency minimum (ideally the servers are located in the same LAN, running at 100Mbps or faster).

While some of our customers run FishEye on SPARC-based hardware, Atlassian only officially supports FishEye running on x86 hardware and 64-bit derivatives of x86 hardware.

Disk Space Requirement Estimates

Disk space requirements for FishEye may vary due to a number of variables such as the repository implementation, file sizes, content types, the size of diffs and comments being stored. The following table contains some real-world examples of FishEye disk space consumption.

<table>
<thead>
<tr>
<th>Repository Technology</th>
<th>Commits</th>
<th>Codebase Size (HEAD of trunk)</th>
<th>FishEye Index Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subversion</td>
<td>14386</td>
<td>466 MB in 12151 files</td>
<td>647 MB</td>
</tr>
<tr>
<td>CVS</td>
<td>8210</td>
<td>115 MB in 11433 files</td>
<td>220 MB</td>
</tr>
</tbody>
</table>

These disk space estimates are to be used as a guideline only. We recommend you monitor the disk space that your FishEye instance uses over time, as needs for your specific environment may vary. It may be necessary to allocate more space than indicated here. Additionally, you can reduce disk space consumption by turning off diff storage in FishEye.

Deployment notes for version control systems

<table>
<thead>
<tr>
<th>Subversion (server)</th>
<th>FishEye can communicate with any repository running Subversion 1.1 or later.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subversion (client)</td>
<td>FishEye now bundles the SVNkit client, which becomes the default Subversion interface. An alternative is to use the native subversion client, using JavaHL bindings. Please see Subversion Client Setup for more information.</td>
</tr>
<tr>
<td>Perforce (client)</td>
<td>FishEye needs access to the p4 client executable. Due to some problems with earlier versions of the client, we recommend version 2007.3 or later.</td>
</tr>
<tr>
<td>CVS</td>
<td>If you are using CVS, FishEye needs read-access to your CVS repository via the file system. It does not support protocols such as pserver at the moment.</td>
</tr>
</tbody>
</table>

Support for other version control systems is planned for future releases. Let us know what SCM system you would like to see supported by creating a JIRA issue or adding your vote to an issue, if the request already exists.

WAR deployment

FishEye/Crucible is currently a Java program that runs on its own. 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 is bundled with the Crowd 2.4.1 client library, and supports the Crowd 2.4.x server.

Font size tips
(Except for Linux users.) For best results you may want to tweak your default monospace font and font-size. The default browser font is usually Courier New which can be hard to read in some browsers. We recommend choosing the same font you use in your IDE and selecting a font size approximately 2 points larger than your variable width font. Firefox, Internet Explorer and Safari all have excellent font rendering. It is worth taking some time to tweak your fonts for the best experience.

End of Support Announcements for FishEye

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

End of Support Matrix for FishEye

<table>
<thead>
<tr>
<th>Platform</th>
<th>FishEye End of Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL 5.0</td>
<td>January 2012</td>
</tr>
<tr>
<td>PostgreSQL 8.0 and 8.1</td>
<td>January 2012</td>
</tr>
<tr>
<td>IBM ClearCase (all versions)</td>
<td>4 April 2012 (announcement)</td>
</tr>
</tbody>
</table>

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

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

On this page (most recent announcements first):
- Deprecated Database Support for FishEye (4 October 2011)
- Deprecated Web Browsers for FishEye (21 March 2011)
- Deprecated Java Platforms for FishEye (21 March 2011)
- Deprecated SCM Repository Support for FishEye (4 April 2011)

Deprecated Database Support for FishEye (4 October 2011)

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

We will stop supporting older versions of databases as follows:
- For the next major version of FishEye, 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 FishEye. If you have questions or concerns regarding these announcements, please email eol-announcement at atlassian dot com.
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 FishEye after January 2012 will not support MySQL 5.0, PostgreSQL 8.0 or 8.1.
- As mentioned above, the releases of FishEye before January 2012 will contain support for MySQL 5.0 and PostgreSQL 8.0 and 8.1.

Deprecated Web Browsers for FishEye (21 March 2011)

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

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

- From FishEye 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 FishEye. If you have questions or concerns regarding this announcement, please email eol-announcement at atlassian dot com.

End of Life Announcement for Web Browser Support

<table>
<thead>
<tr>
<th>Web Browsers</th>
<th>Support End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Explorer 7</td>
<td>When FishEye 2.6 releases (target May 2011)</td>
</tr>
</tbody>
</table>

Internet Explorer 7 Notes:

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

Deprecated Java Platforms for FishEye (21 March 2011)

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

We will stop supporting the following Java Platforms:

- From FishEye 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 FishEye. If you have questions or concerns regarding this announcement, please email eol-announcement at atlassian dot com.

End of Life Announcement for Java Platform Support

<table>
<thead>
<tr>
<th>Java Platform</th>
<th>Support End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Java Platform 5 (JDK/JRE 1.5)  
When FishEye 2.6 releases (target May 2011)

Java Platform 5 End of Support Notes:

- FishEye 2.5 is the last version to officially support Java Platform 5 (JDK/JRE 1.5).
- FishEye 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 FishEye 2.6 release, Atlassian will not provide fixes in older versions of FishEye for bugs affecting Java Platform 5 (JDK/JRE 1.5).

Deprecated SCM Repository Support for FishEye (4 April 2011)

This section announces the end of Atlassian support for certain SCM repositories for FishEye. 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 FishEye.

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

<table>
<thead>
<tr>
<th>SCM Repository</th>
<th>Support End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM ClearCase (all versions)</td>
<td>4 April 2012</td>
</tr>
</tbody>
</table>

IBM ClearCase End of Support Notes:

- Atlassian intends to end of life IBM ClearCase functionality on 4 April 2012. The release of FishEye after 4 April 2012 will not contain any IBM ClearCase functionality.
- As mentioned above, the releases of FishEye 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 FishEye.

End of Support Announcement for IBM ClearCase

Support in FishEye for IBM ClearCase ended on April 4th 2012. FishEye 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 FishEye to support your existing installations with ClearCase. However, Atlassian will not be providing any ClearCase-related support for any FishEye version after 4 April 2012, and has removed all functionality related to ClearCase from FishEye 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 FishEye.

End of Support Announcement for Internally Managed Repositories

On August 14th 2013, we are ending support for internally managed repositories.

You can stay on older versions of FishEye to support your existing installations with Git repository management. However, Atlassian will remove all functionality related to repository management, from FishEye versions...
released after August 14th 2013. We are committed to helping our customers understand this decision and to assist you in migrating your repositories to one of the two other solutions offered by Atlassian if needed:

- **Stash** if you need to host your repositories behind your firewall
- **Bitbucket** if you prefer a SaaS hosting solution

### Why is repository management being removed?

FishEye was built to enable browsing, searching and visualising source code in various Version Control Systems. With many customers requesting repository management, we have decided to provide a solution on top of FishEye. However, the part of FishEye’s architecture that allows it to index different types of repositories and access your Subversion and Git repositories in one place, turned out to not be adequate for a repository management solution.

We have decided to focus on the core strengths of FishEye - browsing, searching and visualizing multiple source code management systems - and strengthen the product around these features. This also has enabled us to deliver a much more focused approach to Git repository management and offer a new solution -- **Atlassian Stash** – which was build from the ground-up with repository management as a focus.

Going forward FishEye will continue to deliver new features and enhancements to help users browse, search and visualize across different Version Control Systems including Git, Subversion, Mercurial, Perforce and CVS.

### My team manages Git repositories in FishEye, how do we migrate?

Here are suggestions to migrating your repositories to **Stash** or **Bitbucket**. The following steps will guide you through the commands that you need to run to migrate your FishEye hosted repositories to a different location.

We will assume that you already have a new repository ready to be used and that you have the latest local copy on your computer. In this case we will use a **Stash** example.

1. Create a new repository on the service you chose (Stash, BitBucket...)
2. Open your terminal and go to the local copy of the directory that you want to push.

   ```bash
   cd /path/to/myrepo
   ```

3. Update the address of your remote origin to point it a the new repository.

   ```bash
   git remote set-url origin ssh://git@stash.mycompany.com/MYPROJECT/myrepo.git
   ```

4. Push all the branches to the remote repository.

   ```bash
   git push --all origin
   ```

5. Push all the tags to the remote repository.

   ```bash
   git push --tags origin
   ```

At this stage all your local branches and tags should be present in your new repository and you can have the same development process as the one you had before.

### Index your newly created repository

7. **Index your newly created repository** with **FishEye** to be able to search, track and view report on your source.

Past this point the migration is complete – your repository should be hosted on a different service and indexed.
You will push and pull against your new service and FishEye will index the changes just like for any external repository.

**Installing FishEye on Windows**

Hey! We're going to install FishEye on Windows. There are a few steps involved, but we think you'll find it easy to follow along. If you are upgrading an existing installation, please refer to the FishEye upgrade guide 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 FishEye with, and that we recommend.

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

   - Running FishEye as a Windows service
   - Installing FishEye on Linux and Mac
   - Starting to use FishEye
   - Supported platforms
   - FishEye upgrade guide

2. **Check your version of Java**

   In a command prompt, run this:

   ```
   java -version
   ```

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

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

   Download and install the Java Platform JDK from Oracle's website.

   ![We recommend that the Java install path should not contain spaces, so don't install into C:\Program Files\Java]. Instead, use a path like C:\Java.

   Now try running `java -version` again to check the installation. The version of Java should be **1.6.0** or higher.

3. **Check that Windows can find Java**

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

   ```
   echo %JAVA_HOME%
   ```

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

   ✔ If you don't see a path without spaces...
If you see a path with spaces, like C:\Program Files\Java\, then sorry, but go back to 2. and reinstall Java to a location that doesn't have spaces.

If you don't see a path at all, or if you just see %JAVA_HOME%, then set JAVA_HOME as follows:

For Windows 7:

1. Go to Start, search for "sys env" and choose Edit the system environment variables.
2. Click Environment Variables, and then New under 'System variables'.
3. Enter "JAVA_HOME" as the Variable name, and the absolute path to where you installed Java as the Variable value. Don't use a trailing backslash. We recommend that JAVA_HOME should point to the Java executable in your PATH.
4. Now, in a new command prompt, try running '%JAVA_HOME%/bin/java -version'. You should see the same version of Java as you saw above.

4. Create a dedicated FishEye user (recommended)

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

- Should not have admin privileges.
- Should be a non-privileged user with read, write and execute access on the FishEye 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 FishEye user, ensure you are logged in as this user to complete the remaining instructions.

5. Now it's time to get FishEye

Download FishEye from the Atlassian download site.

Extract the downloaded file to an install location:

- Folder names in the path to your FishEye executable should not have spaces in them. The path to the extracted directory is referred to as the <FishEye home directory> in these instructions.
- If you expect to have a large number of users for this FishEye installation, and FishEye will be connected to an external database, consider installing FishEye on a different server from the one running the external database, for improved performance.

6. Tell FishEye where to store your data

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

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

Create your FishEye instance directory, and then tell FishEye where you created it by setting a FISHEYE_INST environment variable, as follows:

For Windows 7:

1. Go to Start, search for "sys env" and choose Edit the system environment variables.
2. Click Environment Variables, and then New under 'System variables'.
3. Enter "FISHEYE_INST" as the Variable name, and the absolute path to your new FishEye instance directory as the Variable value. Don't use a trailing backslash.
4. Now copy the newly extracted `<FishEye home directory> /config.xml file to the root of your new FishEye instance directory.

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

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 FishEye!

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

```
binary\start.bat
```

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

Enter your license, then an admin password, to finish the setup. Note that this password is for the ‘built-in’ FishEye 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.

8. Add repositories

Now you can tell FishEye about any existing repositories you have. Please read Starting to use FishEye for the details.

FishEye 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 FishEye, we suggest that you index a single project, so you can use FishEye 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.

9. Add users and groups

You will want to set up your users and groups in FishEye. You can add users directly to FishEye, or connect to an external user directory. Please read Starting to use FishEye for an introduction.

10. Set up your mail server

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

11. Connect to an external database (recommended)

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

12. Stop FishEye (optional)
In a command prompt, change directory to `<FishEye home directory>` and run this:

```
bin\stop.bat
```

13. Tuning FishEye performance

To get the best performance from your new FishEye installation, please consult Tuning FishEye performance. Running FishEye as a Windows service

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

The service wrapper provides the following benefits:

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

⚠️ Please note that:

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

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

Related pages:
- Installing FishEye on Windows

Installing the Java Service Wrapper

To install the Java Service Wrapper on Windows:

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


**3. Setting FishEye environment variables for Windows Services**

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

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

4. Set the FISHEYE_INST environment variable (and other FishEye-specific environment variables) in the `<FishEye home directory>\wrapper\conf\wrapper.conf` file, following the instructions below.

5. Install FishEye as a service as follows:
   a. Open an Administrator command prompt by searching for 'Command prompt' in the Windows Start menu, right-clicking on **Command Prompt** and then choosing **Run as administrator**.
   b. Change directory to `<FishEye home directory>\wrapper\bin` and run `Fisheye-Install-NTService.bat`. If you run into any problems starting the wrapper, you'll find its logs in `<FishEye home directory>\var\log\wrapper.log`.

6. Start the Fisheye service under 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 FishEye service in this way.

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

   **Setting FishEye environment variables for Windows Services**

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

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

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

```bash
# 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.password.file=. /wrapper/jmxremote.password
wrapper.java.additional.9=-Dwrapper.mbean.name="wrapper:type=Java Service Wrapper Control"

wrapper.java.additional.10=-XX:PermSize=256m
```

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

```bash
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 FishEye 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:

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

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

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

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

Troubleshooting

Extracting files from wrapper.zip

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

- Uncompressing wrapper.zip with Winzip or WinRar rather than using the Extract All command in the Windows right-click contextual menu.
- If the wrapper.zip filename appears green instead of black in Windows Explorer, decrypt it, prior to unzipping its contents, by right-clicking on the file, choose Properties, click the Advanced button, then clear the Encrypt contents to secure data checkbox.
Warning when using 64-bit Java JDK

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

```
WARNING - Unable to load the Wrapper's native library 'wrapper.dll'. The file is located on the path at the following location but could not be loaded:
C:\installs\service\fisheye28\wrapper\lib\wrapper.dll.

Please verify that the file is readable by the current user and that the file has not been corrupted in any way. System signals will not be handled correctly.
```

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

Wrapper configuration and "-server" parameter

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

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

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

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

**Installing FishEye on Linux and Mac**

Hey! We're going to install FishEye 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 are upgrading an existing installation, please refer to the FishEye upgrade guide 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 FishEye with, and that we recommend.

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

**Related pages:**

- Installing FishEye on Windows
- Starting to use FishEye
- Supported platforms
- FishEye upgrade guide

2. **Check your version of Java**

   In a terminal, run this:

   ```
   java -version
   ```
The version of Java should be 1.6.0 or higher.

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

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

3. Check that the system can find Java

In a terminal, run this:

```bash
echo $JAVA_HOME
```

You should see a path like `/System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Home/`.

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

<table>
<thead>
<tr>
<th>Linux</th>
<th>Mac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do either of the following:</td>
<td>Insert the following in your ~/.profile file:</td>
</tr>
</tbody>
</table>
| • If JAVA_HOME is not set, log in with 'root' level permissions and run:| JAVA_HOME="path/to/JAVA_HOME"
  export JAVA_HOME
  where path/to/JAVA_HOME may be like: /System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Home/
  • If JAVA_HOME needs to be changed, open the /etc/environment file in a text editor and modify the value for JAVA_HOME to:
  JAVA_HOME="path/to/JAVA_HOME"
  It should look like:
  JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Home/

Refresh your ~/.profile in the terminal and confirm that JAVA_HOME is set:

```bash
source ~/.profile
$JAVA_HOME/bin/java -version
```

You should see a version of Java that is 1.6.0 or higher, like this:

```
java version "1.6.0_24"
```

4. Create a dedicated FishEye user (recommended)

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

- Should not have admin privileges.
- Should be a non-privileged user with read, write and execute access on the FishEye 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 FishEye user, ensure you are logged in as this user to complete the remaining instructions.

5. Now it’s time to get FishEye

Download FishEye from the Atlassian download site.

Extract the downloaded file to an install location:

- Folder names in the path to your FishEye executable should not have spaces in them. The path to the extracted directory is referred to as the `<FishEye home directory>` in these instructions.
- If you expect to have a large number of users for this FishEye installation, and FishEye will be connected to an external database, consider installing FishEye on a different server from the one running the external database, for improved performance.

6. Tell FishEye where to store your data

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

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

Create your FishEye instance directory, and then tell FishEye where you created it by adding a FISHEYE_INST environment variable as follows:

<table>
<thead>
<tr>
<th>Linux</th>
<th>Mac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open the <code>/etc/environment</code> file in a text editor and insert: FISHEYE_INST=&quot;path/to/&lt;FishEye instance directory&gt;&quot;</td>
<td>Open the <code>~/.profile</code> file for the current user in a text editor and insert: FISHEYE_INST=&quot;path/to/&lt;FishEye instance directory&gt;&quot; export FISHEYE_INST</td>
</tr>
</tbody>
</table>

Now, copy the `<FishEye home directory> /config.xml to the root of the FISHEYE_INST directory, so that FishEye can start properly.

Also, 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 FishEye!

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

```
bin/start.sh
```

After a few moments, in a web browser on the same machine, go to `http://localhost:8060/` (or, from another machine, type `http://hostname:8060/`, where hostname is the name of the machine where you installed FishEye).
Enter your license, then an admin password, to finish the setup. Note that this password is for the 'built-in' FishEye admin user. You can log in as this user, if necessary, by clicking the Administration link in the page footer.

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

8. Add repositories

Now you can tell FishEye about any existing repositories you have. Please read Starting to use FishEye for the details.

FishEye 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 FishEye, we suggest that you index a single project, so you can use FishEye 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.

9. Add users and groups

You will want to set up your users and groups in FishEye. You can add users directly to FishEye, or connect to an external user directory. Please read Starting to use FishEye for an introduction.

10. Set up your mail server

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

11. Connect to an external database (recommended)

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

12. Stop FishEye (optional)

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

```
bin/stop.sh
```

13. Tuning FishEye performance

To get the best performance from your new FishEye installation, please consult Tuning FishEye performance.

Starting to use FishEye

This page will guide you through the basics of using FishEye. By the end of it you should be able to:

- Create accounts for your collaborators, and organize them into groups.
- Add repositories that need to be indexed and setup permissions.
- Use the Commit Graph to trace the history of your code

This page assumes that:

- You have installed and started the latest version of FishEye. See Installing FishEye on Linux and Mac or Installing FishEye on Windows for details.
- You are using a supported browser.
Create users in FishEye

FishEye doesn’t have any user accounts after you have installed it for the first time. You need to go to the Administration interface to add the first users of the system.

Click on the Administration link in the footer:

In the Users listing page click Add User to go to the user creation form:

Fill in the form and create your user:
From the User page you can click on Create another user to repeat this operation:

**User**

The user John Doe has been created.

Create another user.

**User Details**

Username: jdoe
Display name: John Doe
Email: john@doe.com
Type: built-in
Admin: false

**Group Details**

Groups: No groups.
Edit Groups

List of users

Return to the list of users

**Add a repository**

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

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

In the repository configuration, add the location of your repository. Fill in the authentication details if necessary.
Finally indicate whether or not you would like diff indexing should be turned on and if the repository should not be indexed right away.

Click **Add** to finish the process.
Once it's created you can click **Browse**, in the repository options menu, to access your repository.

You can now browse your files in FishEye, search through your code or track modifications via the commit graph.

### Configuring JIRA Integration in the Setup Wizard

This page describes the **Connect to JIRA** tab of the FishEye 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.
Connecting to JIRA in the Setup Wizard

To configure JIRA integration while running the FishEye setup wizard:

1. Configure the following setting in JIRA: Allow remote API access.
2. Enter the following information on the 'Connect to JIRA' step of the setup wizard:

<table>
<thead>
<tr>
<th>JIRA base URL</th>
<th>The web address of your JIRA server. Examples are: <a href="http://www.example.com:8080/jira/">http://www.example.com:8080/jira/</a> <a href="http://jira.example.com">http://jira.example.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>JIRA admin username</td>
<td>The credentials for a user with the 'JIRA System Administrators' global permission in JIRA.</td>
</tr>
<tr>
<td>JIRA password</td>
<td></td>
</tr>
<tr>
<td>FishEye base URL</td>
<td>JIRA will use this URL to access your FishEye server. The URL you give here will override the base URL specified in your FishEye administration console, for the purposes of the JIRA connection.</td>
</tr>
<tr>
<td>Groups to synchronize</td>
<td>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.</td>
</tr>
<tr>
<td>Admin Groups</td>
<td>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.</td>
</tr>
</tbody>
</table>

3. Click Connect to JIRA.
4. Finish the setup process.
### Troubleshooting

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.

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
</table>
| 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>`.<br>  
| 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.IllegalArgumentException`: 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 the ‘Delete’ link next to the application link that you want to delete.
   d. A confirmation screen will appear. Click the ‘Confirm’ button to delete the application link.
4. Remove the user management configuration from JIRA, if it exists:
   a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:
      - In JIRA 4.3: Click ‘Other Applications’ in the ‘Users, Groups & Roles’ section of the JIRA administration screen.
      - In JIRA 4.4: Select ‘Administration’ > ‘Users’ > ‘JIRA User Server’.
   b. Look for a link to your application. It will have a name matching this format:

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

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

- In JIRA 4.4: Select 'Administration' > 'Users' > 'JIRA User Server'.
- 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.
- In JIRA, click 'Delete' next to the application that you want to remove.
- Add the application link in JIRA again, so that you now have a two-way trusted link between JIRA and your application:
  - Click 'Add Application Link'. Step 1 of the link wizard will appear.
  - Enter the server URL of the application that you want to link to (the 'remote application').
  - Click the 'Next' button.
  - Enter the following information:
    - 'Create a link back to this server' – Tick this check box to add a two-way link between the two applications.
    - 'Username' and 'Password' – Enter the credentials for a username that has administrator access to the remote application.
      Note: These credentials are only used to authenticate you to the remote application, so that Application Links can make the changes required for the new link. The credentials are not saved.
    - 'Reciprocal Link URL’ – The URL you give here will override the base URL specified in your remote application's administration console, for the purposes of the application links connection. Application Links will use this URL to access the remote application.
  - Click the 'Next' button.
  - Enter the information required to configure authentication for your application link:
    - 'The servers have the same set of users' – Tick this check box, because the users are
the same in both applications.

- **These servers fully trust each other** – Tick this check box, because you trust the code in both applications and are sure both applications will maintain the security of their private keys.

For more information about configuring authentication, see Configuring Authentication for an Application Link.

- Click the 'Create' button to create the application link.

7. Configure a new connection for user management in JIRA:

a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:

- In JIRA 4.3: Click 'Other Applications' in the 'Users, Groups & Roles' section of the JIRA administration screen.
- In JIRA 4.4: Select 'Administration' > 'Users' > 'JIRA User Server'.

b. Add an application.

c. Enter the application name and password that your application will use when accessing JIRA.

d. Enter the IP address or addresses of your application. Valid values are:

- A full IP address, e.g. 192.168.10.12.
- A wildcard IP range, using CIDR notation, e.g. 192.168.10.1/16. For more information, see the introduction to CIDR notation on Wikipedia and RFC 4632.

- Save the new application.

8. Set up the JIRA user directory in the application.

- For Confluence:

  a. Go to the Confluence Administration Console.
  b. Click 'User Directories' in the left-hand panel.
  c. Add a directory and select type 'Atlassian JIRA'.
  d. Enter the following information:

    - Name – Enter the name of your JIRA server.
    - Server URL – Enter web address of your JIRA server. Examples:

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

    - Application name and Application password – Enter the values that you defined for Confluence in the settings on JIRA.
  
  e. Save the directory settings.
  
  f. Define the directory order by clicking the blue up- and down-arrows next to each directory on the 'User Directories' screen.

  For details see Connecting to Crowd or JIRA for User Management.

- For FishEye/Crucible:

  a. Click Authentication (under 'Security Settings').
  b. Click Setup JIRA/Crowd authentication. Note, if LDAP authentication has already been set up, you will need to remove that before connecting to JIRA for user management.
  c. Make the following settings:

<table>
<thead>
<tr>
<th>Authenticate against</th>
<th>Select a JIRA instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application name and password</td>
<td>Enter the values that you defined for your application in the settings on JIRA.</td>
</tr>
</tbody>
</table>
**JIRA URL**

The web address of your JIRA server. Examples:

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

**Auto-add**

Select [Create a FishEye user on successful login](#) so that your JIRA users will be automatically added as a FishEye user when they first log in.

**Periodically synchronise users with JIRA**

Select [Yes](#) to ensure that JIRA will synchronize all changes in the user information on a regular basis. Change the value for [Synchronise Period](#) if required.

**When Synchronisation Happens**

Select an option depending on whether you want to allow changes to user attributes from within FishEye.

**Single Sign On**

Select [Disabled](#). SSO is not available when using JIRA for user management and if enabled will make the integration fail.

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

- For Stash:
  a. Go to the Stash administration area.
b. Click [User Directories](#) in the left-hand panel.
c. [Add](#) a directory and select type [Atlassian JIRA](#).
d. Enter the following information:
   - **Name** – Enter the name of your JIRA server.
   - **Server URL** – Enter web address of your JIRA server. Examples:
     
     ```
     http://www.example.com:8080/jira/
     http://jira.example.com
     ```
   - **Application name** and **Application password** – Enter the values that you defined for Stash in the settings on JIRA.
e. Save the directory settings.
f. Define the directory order by clicking the blue up- and down-arrows next to each directory on the 'User Directories' screen.

   For details see [Connecting to JIRA for user management](#).
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.

**Using FishEye**

- **Using the FishEye screens**
  - Browsing through a repository
  - Searching FishEye
  - Viewing a file
    - Viewing file content
    - Using Side by Side Diff View
  - Viewing a file history
  - Viewing the changelog
- FishEye Charts
- Using favourites in FishEye
- Changeset Discussions
- Viewing the commit graph for a repository
- Viewing People’s Statistics
- Using smart commits
- Changing your user profile
  - Re-setting your password
- Pattern matching guide
- Date Expressions Reference Guide
- EyeQL Reference Guide

**Using the FishEye screens**

The sections below describe the different screens in FishEye and the information you can retrieve from them. Each page (tab) has a number of panes, and each pane is described separately.

**Header**

The header along the top of each FishEye page provides the following:

- The application navigator, at the left of the header, connects you directly to your other applications, such as JIRA and Bamboo. Admins can configure which apps appear in the navigator – just click **Application navigator** in the admin area.
- FishEye logo – click to go to the dashboard, to see your personal code commits, your reviews (if you are using Crucible) and your activity stream.
- **Repositories** — the list of all FishEye repositories. Click a repository name to browse the repository. A number of sub-tabs become available, as described below (see 'Repository sub-tabs' below).
- **Projects (when using with Crucible)** – a link to all projects (see the **Crucible documentation**). Logged-in users can see links to recently visited projects.
- **People** – tab to view statistics about committers to your FishEye repositories (see Viewing People’s...
Statistics). Logged in users can see links to users they have recently visited.

- **Reviews (when using with Crucible)** – go to your code reviews (see the Crucible documentation). Logged-in users can see links to recently visited reviews as well as to the Crucible ‘Inbox’ and ‘Outbox’.
- **Create review (when using with Crucible)** – click the down arrow to choose Create snippet.
- Click your avatar to change your user settings (see Changing your user profile).

**Repository sub-tabs**

Once you have selected a repository, you can navigate through it by selecting files and folders on the tree in the left navigation bar. The sub-tabs in the ‘Source’ tab change depending on whether you are viewing a repository or a file.

**Viewing a repository**

See Browsing through a repository.

**Viewing a file**

When you reach a source file, a summary page is shown, displaying recent revisions to the file.

The horizontal sub-tabs above the file provide different views for the file:

<table>
<thead>
<tr>
<th>Sub-tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revisions</td>
<td>Shows the latest revisions of the file. See Viewing a file history.</td>
</tr>
<tr>
<td>Files</td>
<td>Shows the contents of the directory.</td>
</tr>
<tr>
<td>Activity</td>
<td>Shows recent activity on the item. There are a number of sub-options here (see Viewing the changelog):</td>
</tr>
<tr>
<td></td>
<td>• All Activity — The default view, showing commits, reviews¹ and JIRA issues².</td>
</tr>
<tr>
<td></td>
<td>• Commits — Shows commits in the activity stream.</td>
</tr>
<tr>
<td></td>
<td>• Reviews¹ — Shows review activity in the activity stream.</td>
</tr>
<tr>
<td></td>
<td>• Scroll to Changeset — Opens the changeset ID specified in the text field (press Enter to carry out the action).</td>
</tr>
<tr>
<td></td>
<td>• Filter — Applies constraints to the current activity stream.</td>
</tr>
<tr>
<td></td>
<td>• Show Revisions — If this is selected, then changeset items are automatically expanded to show modified files.</td>
</tr>
<tr>
<td></td>
<td>• Earlier Activity (Left Arrow icon) — Loads a page of earlier activity.</td>
</tr>
<tr>
<td></td>
<td>• Later Activity (Right Arrow icon) — Loads a page of later activity.</td>
</tr>
<tr>
<td>Users</td>
<td>Shows the commit history of the different users that have committed changes on the item.</td>
</tr>
</tbody>
</table>

¹ If you are using Crucible  
² If you are using JIRA
Reports | Shows activity charts for the item. Various chart options can be selected in the left navigation bar (see FishEye Charts).
Source | Shows the contents of the file.
Query | Allows you to run an advanced search.

### Browsing through a repository

You can use FishEye to select a repository and browse through it. The repository view provides you with information about the files in the repository, activity occurring related to the repository and users committing to the repository. You can also generate charts and search for specific file revisions in the repository.

<table>
<thead>
<tr>
<th>On this page:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Browsing a repository</td>
</tr>
<tr>
<td>• Hiding empty directories and deleted files</td>
</tr>
<tr>
<td>• Watching a repository</td>
</tr>
</tbody>
</table>

#### Browsing a repository

**To browse a repository:**

1. Click **Repositories** in the header and choose either a recently viewed repository, or **All repositories**.
2. Click the name of a repository to view its contents. See the ‘Viewing a repository’ screenshot below.
3. If required, use the selector (under the repository or file name) to choose the branch or tag that you want to browse the changelog for.
4. You can view various information about the repository, as outlined below. If you navigate to a folder, the context of the information below will change. For example, if you navigate to a particular folder in the left navigation tree, the activity, files and users information, reports generated and search results will all relate to that folder.

   - A **greyed out item is either a deleted file or an empty/deleted (folder).**
   - **Activity** tab — View the commit, review and issue (requires JIRA) activity related to the repository/folder. The activity stream is similar to the changelog activity stream, see Viewing the changelog for more information.
   - **Commit Graph** tab — Visualise the repository, using the commit graph. See Viewing the commit graph for a repository for more information.
   - **Files** tab — View the contents of the repository/folder being viewed.
   - **Users** tab — View the commit history of the users that have committed changes to files in the repository/folder. See Viewing People’s Statistics for more information.
   - **Reports** — View activity charts for the repository/folder. See FishEye Charts for more information.
   - **Search** — Search the repository/folder. See Searching FishEye for more information.
5. Click any file, when browsing the repository, to view information about the file. See Viewing a file for more information.

**Screenshot: Browsing a repository**
Hiding empty directories and deleted files

FishEye tracks deleted files for your repository. Deleted files will be greyed out in your left-hand navigation tree. If all of the files in a directory are deleted, the empty directory will also be greyed out. Note, deleted files and empty directories are not removed from your left navigation.

You can choose to hide deleted files and empty directories in the left navigation tree when browsing through a repository, as described below.

To hide deleted directories/files in your navigation tree:

1. Click the Source tab and browse a repository.
2. In the left hand navigation panel, click the to show the dropdown menu:
   - Hide empty directories – hide all empty (greyed out) directories and their contents (i.e. deleted files and other empty directories).
   - Hide deleted files – hide deleted (i.e. greyed out) files. This does not affect directories.

Hence, if you choose to hide both empty directories and deleted files, you will only see files and directories that exist on the Head of that path. In repositories other than Subversion repositories, this could mean files/directories on any branch.

Screenshot: Hiding deleted files/empty directories
You can "watch" a repository in FishEye/Crucible. Watching the repository allows you to receive emails when changes are made to the repository. You can view all of your watches and configure the frequency of your watch emails in your user profile. See Changing your user profile for more information.

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

To watch a repository:

1. Navigate to the repository that you want to watch.
2. Choose Tools > Watch. (The watch icon becomes coloured, not grey).

To remove the watch, choose Tools > Watch again. (The watch icon becomes grey). You can also remove watches from within your user profile.

Searching FishEye

FishEye has a powerful search engine that allows you to find changesets, committers and files. There are two methods for searching in FishEye:

- **Quick Search** — The Quick Search allows you search all repositories connected to FishEye 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; files edited within the last twelve months are given greater weighting.
- **Advanced Search** — The Advanced Search allows you to search a single repository by entering search criteria against a range of fields, e.g. commit comments, file contents, etc. These parameters can be selected on the standard search interface or specified using the FishEye's custom query language: EyeQL. This search is more complex to use, however you can define more precise search criteria.

On this page:
- Using the Quick Search
- Using the Advanced Search
- Notes

Using the Quick Search

Before you begin:

- Cross-repository searching has a **100-repository limitation** on searches, to prevent it from becoming unresponsive on FishEye instances that have large numbers of repositories. This means that cross-repository Quick Search is not an exhaustive search if you have more than 100 repositories, as only the first 100 repositories (alphabetically, as defined in FishEye) are included. For faster responses, you should limit your search to a particular repository, if possible. FishEye will also limit the search to the specific repository that you are looking at, if you are already navigating within a specific repository
- The Quick Search will also return code reviews, if you are using **Crucible** with FishEye. For information on searching Crucible, see Searching Crucible in the Crucible documentation.

To search FishEye using the Quick Search:

1. Enter your search criteria in the search box in the FishEye header (i.e. Quick Nav). FishEye offers a number of parameters and functions that you can use to refine your expected results, see Refining your Quick Search Criteria below.
2. "Quick Nav" results will appear in a dropdown, as you type. "Quick Nav" will attempt to match against the file name, repository, committer and username.
   - If you want to use a quick nav result, use the up- and down-arrows on your keyboard and press enter or use your mouse to select the item.
   - If the quick nav results don't have what you are looking for, press enter to run a search. Ensure that no items in the dropdown are selected when you press enter.
3. The Quick Search results page will be displayed. You can filter your results further, as described in Filteri
ng 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 FishEye 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 FishEye.

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.

<table>
<thead>
<tr>
<th>Search Tool</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CamelCase Pattern Matching</td>
<td>Enter a CamelCase string pattern and FishEye find files and directories that match the pattern. This is a common search feature in many popular IDEs.</td>
<td>Search for BooQCTest and FishEye will return results like BooleanQueryCoordTest and BooleanQueryClassTest.</td>
</tr>
<tr>
<td>Path/File Pattern Matching</td>
<td>Enter a path/file pattern and FishEye will find files and directories that match the pattern.</td>
<td>Search for common/final/Actions and FishEye will return results like /src/common/eu/systemworks/specialprojects/final/Actions.java.</td>
</tr>
<tr>
<td>Field Handles</td>
<td>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.</td>
<td>Search for file:build.xml and FishEye will return files that have a name matching build.xml.</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>• file — Search against a file/directory name only.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• commit — Search against a commit message only.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• diff — Search against a diff (lines added/removed) only.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• content — Search against contents of a file only.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• committer — Search against a committer only.</td>
<td></td>
</tr>
<tr>
<td>Searching within a Directory (AntGlobs)</td>
<td>AntGlobs can be used in the Quick Search to help search within a specific directory.</td>
<td>Search for /src/**/gwt/*.xml and FishEye will return all files with a .xml suffix that are below both a src and a gwt directory. e.g. src/java/com/atlassian/fe cru/gwt/FecruCore.gwt.xml but not src/java/com/atlassi an/fecru/ApplicationConte xt.xml</td>
</tr>
<tr>
<td>Searching for Discrete Strings</td>
<td>Enter a specific string within quotation marks and FishEye will match against the exact string. Note, this search is not case-sensitive.</td>
<td>Enter &quot;update version in build&quot; and FishEye will only return results that match that exact string, i.e. it will not return a result with update build version or update version in file.</td>
</tr>
</tbody>
</table>

Filtering Quick Search Results

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

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repositories</td>
<td>Select or enter the name of the repository that you want to restrict your results to. For example, if you enter 'FE' then the search results page will refresh to display only files and directories in the 'FE' repository.</td>
</tr>
</tbody>
</table>

⚠️ If you are using Crucible with FishEye, there will be a projects dropdown in the 'Reviews' section. Selecting a Crucible project in this dropdown will not filter the FishEye search results. It is only used to filter reviews returned in the search results. See Searching Crucible.
| **Files and Directories** | Click this link to restrict your results to files and directories that have names that match the search criteria. |
| **Changeset Comments** | Click this link to restrict your results to changeset comments that match the search criteria. |
| **Diffs** | Click this link to restrict your results to diffs (lines added/removed) that match the search criteria. |
| **Content** | Click this link to restrict your results to files that have content that match the search criteria. |
| **Committers** | Click this link to restrict your results to committers that match the search criteria. |

### Using the Advanced Search

The Advanced Search can only be run against a specific repository, however you can specify more precise criteria against a number of fields for that repository.

**To search a FishEye repository using the Advanced Search:**

1. Navigate to the repository that you want to search, as described on Browsing through a Repository.
2. Click the 'Search' tab.
3. Enter your search criteria, as follows (Click the 'Switch to EyeQL Search'/Switch to Standard Search' link at the bottom of the 'Search Criteria' panel to switch between the two search methods):
   - **Standard Search** — Fill in the desired fields in the 'Search Criteria' panel. See the Specifying Search Criteria using the Standard Search Interface section below for more details.
   - **EyeQL Search** — Enter your "EyeQL" query. See the Specifying Search Criteria using EyeQL section below for more details.

### Specifying Search Criteria using the Standard Search Interface

The Advanced Search interface allows you to specify search criteria for multiple fields, order the results, group the results and choose the display fields in the results. You can link to the search results, as well as save the results to a CSV file.
Screenshot: Advanced Search Criteria — Standard Search Interface

Please note the following:

- **Contents of files** — Files must be non-binary, less than 5MB, and for svn repositories on trunk only. Only HEAD/tip revisions are searched. For older revisions, use the added/removed text search criteria.
- **File names/paths** — Antglobs can be used to specify the criteria for this field.

Specifying Search Criteria using EyeQL

The Advanced Search also allows you to run searches using FishEye’s powerful query language, EyeQL.

For information on how to construct an EyeQL query, see the EyeQL Reference Guide. If you haven’t built an EyeQL query before, we recommend that you use the Standard Search Interface interface to build your initial query, then switch to EyeQL to modify that query.
Searching Crucible *(Crucible documentation)*

### Viewing a file

You can search or browse your repositories in FishEye to view a specific file. FishEye provides information about the file history, file content and activity related to the file.

**To view a file:**

1. **Search,** or **browse** through a repository, to find the file.
2. Click the file name. The revision history for the file will be displayed. See the screenshot below.
3. View information about the file on these tabs:
   - **Activity** — Displays the commits and reviews activity related to the file. See **Viewing the changelog**.
   - **Revisions** — Displays the history of revisions for the file. See **Viewing a file history**.
   - **Users** — Displays commit histories for users who have committed changes to the file. See **Viewing people's statistics**.
   - **Reports** — Displays charts for the file activity. See **FishEye Charts**.
   - **Source** — Displays the annotated file contents. The raw file can be downloaded from this tab. See **Viewing file content**.

*Screenshot: Viewing a file (Revisions tab)*
Viewing file content

You can search or browse your repositories in FishEye to view a specific file. FishEye allows you to view and download the source code for the file. You can also view diffs between different revisions of the file and annotations.

To view the source code for a file:

1. Search, or browse through a repository, to find the file.
2. Click the file name. The revision history for the file will be displayed.
3. Click the **Source** tab.

<table>
<thead>
<tr>
<th>Displaying the diff</th>
<th>Select revision numbers (e.g. '107905') from the two revision dropdowns to display the diff for those two revisions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changeset</td>
<td>View the changeset that the revision was a part of.</td>
</tr>
<tr>
<td>Raw</td>
<td>Download the raw source code for the file.</td>
</tr>
<tr>
<td>Annotation Highlighting</td>
<td>Choose <strong>Age</strong>, <strong>Author</strong> or <strong>None</strong> to colour the annotations by age, author or remove highlighting respectively. The highlights are displayed over the revision numbers, next to the authors.</td>
</tr>
<tr>
<td>Columns</td>
<td>Select the columns to display: <strong>Author</strong>, <strong>Revision</strong> and <strong>Line Number</strong>.</td>
</tr>
<tr>
<td>Reviews</td>
<td>Select Create Review to create a Crucible review from the file. <em>(Requires Crucible)</em></td>
</tr>
</tbody>
</table>

**Screenshot: Viewing a file (Source tab)**
Using Side by Side Diff View

This page describes FishEye's innovative 'side-by-side diff' view that shows how a file's content has changed, compared on the left and right hand sides of the screen.

On this page:

- Opening the side-by-side diff view
- Understanding side-by-side diffs
- Alternative ways to open side-by-side diffs
  - From the FishEye Dashboard
  - From the Revisions History view

Opening the side-by-side diff view

To open FishEye's side by side diff view:

1. Open the source code view for the file in question.
2. Select a range of revisions to compare.
3. Choose View > Side by Side Diff.

Screenshot: Choosing a Revision Range for the Diff

The left and right panes of the side-by-side diff view can scroll independently, and the view stays anchored around a central point. Colour coding is used to illustrate where lines have been added (green highlights) and where lines have been removed (red highlights). Grey highlights indicate that a line's internal content has changed. Each addition or deletion is linked to the adjacent pane by a coloured triangle that shows the location...
of that change in the comparison file.

Screenshot: FishEye’s Side by Side Diff View

Understanding side-by-side diffs

Features of the side-by-side diff are referenced in the annotated screenshot below.

1. Added lines are highlighted green, displayed in the right hand pane.
2. Edited lines are highlighted grey, with minor sections highlighted red and green to show deletions and additions.
3. Deleted lines are highlighted red, displayed in the left hand pane.
4. Line numbers in the margin are permanent links ("permalinks") that can be sent to colleagues. When they open those links, the view will automatically open in side by side diff mode.

Screenshot: Elements of the Side by Side Diff View

Alternative ways to open side-by-side diffs

From the FishEye Dashboard

You can also open side by side diffs from the Dashboard screens, by clicking the 'Delta' triangle icon next to an item when it appears in the stream of events. This will open the file in the diff view. If you have currently selected side by side diff as the viewing mode, then the diff will automatically be displayed in that mode. If not, you can select side by side diff from the 'View' menu.
From the Revisions History view

When in the revisions view, you can show a diff by checking the boxes next to two revisions, then clicking the 'Diff' button in the top control bar. If you have currently selected side by side diff as the viewing mode, then the diff will automatically be displayed in that mode. If not, you can select side by side diff from the View menu.

You can also launch into a diff of the latest revision and the second most recent by clicking 'Latest Diff' in the top control bar.

Viewing a file history

You can view a specific file when browsing a repository. This allows you to see information about the file, including the history of file revisions.

To view the history of revisions for a file:

1. Log into FishEye/Crucible.
2. Search, or browse through a repository, to find the file.
3. Click the Revisions tab. The history of revisions for the file will be displayed. See the ‘File Revisions’ screenshot below.

<table>
<thead>
<tr>
<th>Diff 2 selected</th>
<th>Check boxes for two file revisions and then click to view the diff for those revisions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diff latest</td>
<td>View the diff of the two most recent file revisions.</td>
</tr>
<tr>
<td>Filter</td>
<td>View the file filter. Enter the desired fields to filter the file history results on.</td>
</tr>
<tr>
<td>Include other branches</td>
<td>Include revisions of the same file from other branches. A file can have many physical paths, all of which relate to the same filename in your project structure, or repository's logical structure. This applies to Subversion's branching and tagging directory structure in particular.</td>
</tr>
<tr>
<td>Show all details</td>
<td>Toggle expand or collapse of all file revisions to show additional information including the revision ID, parents and the branch where it is head, denoted with this graphic: HEAD.</td>
</tr>
</tbody>
</table>

See the ‘Overview of a File Revision’ diagram below for an explanation of the information provided about individual revisions.

Screenshot: File Revisions
Diagram: Overview of a file revision

Viewing the changelog

The changelog is a record of the commits and reviews for a repository, branch, directory or file. You can view the recent activity in the changelog when browsing a repository/branch/directory or viewing a file.

On this page:

- Viewing changelog activity
- Filtering commit activity in the changelog
- Watching the changelog activity

Viewing changelog activity

To view the changelog activity for a repository, branch, directory or file:

1. Browse to the desired repository, branch, directory or view the desired file.
2. If required, use the selector (under the repository or file name) to choose the branch or tag that you want to browse the changelog for.
3. Click the Activity tab. The recent changelog activity of your repository/branch/directory or file will be displayed.
4. Use the following functions of the Activity tab to see different aspects of the changelog activity:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>Click to show commits, reviews (requires integration with Crucible) and JIRA issues (requires integration with JIRA) activity in the activity stream.</td>
</tr>
<tr>
<td>Commits</td>
<td>Click to show only commits in the activity stream.</td>
</tr>
<tr>
<td>Reviews</td>
<td>Show only review activity. (Requires integration with Crucible)</td>
</tr>
</tbody>
</table>
Filter commits

See Filtering commit activity in the changelog (below) for more information.

Expand all

Show all modified files related to each changset.

Scroll to changeset

Type a changeset ID (e.g. 107856) and press Enter to scroll to the that changeset in the activity stream.

Screenshot: Viewing the changelog activity for a file

Filtering commit activity in the changelog

You can filter the commits that are displayed in the activity stream, that is, the commits in the All and Commits tabs of the Activity tab. Note that you cannot use the commits filter to filter reviews.

To filter commit activity:

1. Go to the Activity tab, as described above.
2. When viewing either the All or Commits tab, click Filter commits.
3. Enter filtering criteria for the commits to be displayed:

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Committer</td>
<td>Changesets checked in by the given committer/author.</td>
</tr>
<tr>
<td>Log Comment</td>
<td>Changesets where the commit comment matches the given text.</td>
</tr>
<tr>
<td>File Extension</td>
<td>Changesets that contain files with the specified file extension.</td>
</tr>
<tr>
<td>File Name</td>
<td>Changesets that contain a given file.</td>
</tr>
<tr>
<td>Start Date</td>
<td>Changesets created on or after that date. Must be of the form YYYY-MM-DDTHH:mm:ss, YYYY-MM-DD, YYYY-MM or YYYY (you can use '/' instead of '-').</td>
</tr>
</tbody>
</table>
End Date | Changesets created on or before that date. Must be of the form YYYY-MM-DDTHH:mm:ss, YYYY-MM-DD, YYYY-MM or YYYY (you can use ‘/’ instead of ‘-’).

4. Click **Apply**.
   - Use **Clear** to clear the filter.
   - Click **Filter commits** again to turn off the filter.

**Screenshot: Using the filter**

![Screenshot: Using the filter](image)

**Watching the changelog activity**

You can "watch" a changelog's activity stream in FishEye and Crucible. Watching the activity stream allows you to receive emails when updates occur in the activity stream. You can view all of your watches and configure the frequency of your watch emails in your user profile. See [Changing your user profile](#) for more information.

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

**To watch an activity stream:**

1. Navigate to the activity stream that you want to watch.
2. Choose **Tools > Watch**. The page will reload and a watch will be set up for the activity stream (the watch icon will now be coloured, not grey).
   - To remove the watch, from the activity stream, choose **Tools > Watch**. The watch will be removed. You can also remove watches from your user profile.

**FishEye Charts**

When browsing a repository, the **Reports** tab displays graphical information about the lines of code (LOC) committed to the repository, over time. The following options are available:

- **Charts**
- **Code Metrics**
- **Notes**

**Charts**

Click **Reports** and then **Charts** when browsing a repository to see charts of activity in the repository.

You can control the chart type that is displayed and various chart options. Click the cog icon on the left, select the required options, and click **Apply**. The available options include:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Explanation</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Branch/Tag selector</strong></td>
<td>Limits the chart to the selected branch/tag.</td>
<td>Any branch/tag from the current repository.</td>
<td>Displays the default/trunk.</td>
</tr>
<tr>
<td><strong>Chart type</strong></td>
<td>Changes the chart's presentation.</td>
<td>Area, line, pie or change* chart.</td>
<td>Area</td>
</tr>
<tr>
<td><strong>Author</strong></td>
<td>Limits the chart to show specific author(s).</td>
<td>Any author configured in the system.</td>
<td>All</td>
</tr>
<tr>
<td><strong>Extension</strong></td>
<td>Limits the chart to show specific file type(s).</td>
<td>Any file extension; e.g. '.java'.</td>
<td>All</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------</td>
<td>----------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td><strong>Subdirectory</strong></td>
<td>Limit the chart to a folder under the current branch. Files in the current directory are represented by an element labelled '.(this dir)'.</td>
<td>A single folder.</td>
<td>None (show all)</td>
</tr>
<tr>
<td><strong>Start Date</strong></td>
<td>Date of the earliest data to show.</td>
<td>Date in format YYYY-MM-DD.</td>
<td>None (show all)</td>
</tr>
<tr>
<td><strong>End Date</strong></td>
<td>Date of the latest data to show.</td>
<td>Date in format YYYY-MM-DD.</td>
<td>None (show all)</td>
</tr>
</tbody>
</table>

*The 'Change' chart displays the change in lines of code, for a specific date range, expressed as a line graph. For example, if the lines of code at the start date is 100, the start point will be zero and the rest of the graph shifted by 100 lines.

**Screenshot: FishEye custom chart settings**
Screenshot: FishEye per-author LOC chart

![LOC and Commit History - Path /, Branch (trunk), Show by author](chart1)

Screenshot: FishEye per-author LOC chart showing multiple authors

![LOC and Commit History - Path /, Branch (trunk), Show by author](chart2)

Screenshot: FishEye LOC chart by file extension
Per-Author Lines of Code Statistics

You can view per-author statistics for lines of code as a chart. This allows you to see how many lines of code were contributed to your project by each author, over time. You can easily view this information on the charts page. Note, if you are upgrading from a previous version of FishEye, you will need to re-index the repository in order to show the per-author information.

Code Metrics

A number of built-in reports are also provided:

**Screenshot: Commit Time/Volume**

**Screenshot: Top Committers**
Adding favourites

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>Hover the mouse cursor over their avatar or username. In the context menu, click <strong>Follow</strong>.</td>
</tr>
<tr>
<td>Changesets</td>
<td>Open the changeset and click the grey star icon next to its name, near the top of the screen.</td>
</tr>
<tr>
<td>Files or folders</td>
<td>Open the file or folder and click the grey star icon that appears next to its name. The name appears in the breadcrumb links at the top of the screen.</td>
</tr>
<tr>
<td>Repository</td>
<td>Click the <strong>Source</strong> tab and then the grey star icon that appears next to the name of the desired repository.</td>
</tr>
</tbody>
</table>

**Screenshot: Adding a repository to your favourites**
Managing favourites

To change the display name for, or remove, a favourite:

1. Click the Favourites menu (at the top of the FishEye screen, next to your username) and choose Manage favourites.
2. Click the yellow star beside the favourite, and either:
   a. edit the display name, and click Save label
   b. click Remove.

Due to FE-2348 you cannot currently rename favourite directories, users or committers

Screenshot: Renaming a favourite

Changeset Discussions

Please see the Crucible documentation for instructions on this feature.

Viewing the commit graph for a repository

The commit graph shows changesets in their respective branches, using configurable "swimlanes". This allows you to see key information such as branching and merging (and if you are using Git or Mercurial, you will be able to see anonymous branches as well).

The Highlight feature of the commit graph allows you to highlight different types of information in the swimlanes or changeset list:

- ancestors and descendants for a changeset
- commits with JIRA issues
- reviewed and unreviewed changesets.

For example, if you have the JIRA issues highlight active, clicking a changeset with a JIRA issue in the commit comment will show all other changesets with the same JIRA issue.

Before you begin:

- Subversion repositories currently do not show lines between branch swimlanes (i.e. merging). But in
some cases, FishEye might pick up associations based on SVN branch points.

- Some features of the commit graph are only available if you are using Crucible with FishEye. For details, see the description below.
- Some features of the commit graph are only available if you are using JIRA with FishEye. For details, see the description below.

On this page:

- Viewing the commit graph for a repository
- Highlighting the lineage of a changeset
- Highlighting JIRA issues
- Highlighting reviewed changesets
- Highlighting commits by an author
- Highlighting search results
- Viewing changesets across all branches
- Reordering swimlanes for Git repositories

Related pages:

- Subversion Changeset Parents and Branches
- What are Subversion root and tag branches?
- Perforce Changesets and Branches
- Using the FishEye screens
- Browsing through a repository
- JIRA Integration in FishEye

Viewing the commit graph for a repository

To view the commit graph for a repository:

1. Navigate to the desired repository, as described on Browsing through a repository.
2. Click the Commit Graph tab.

Selecting, or hovering on, a changeset (regardless of highlight) will display the following in the row for the changeset:

- an icon. Click this icon to see details for the changeset.
- a cog icon with a menu that allows you to see the changeset ID, view the full changeset, view the changeset in the activity stream, or to create a review for the changeset.
Highlighting the lineage of a changeset

Choose **Highlight > Lineage** to show the ancestor and descendant changesets for a selected changeset.

<table>
<thead>
<tr>
<th>Action</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click on a changeset in the changeset list</td>
<td>Highlights where a changeset comes from and where it propagates to, i.e. its ancestors and descendants.</td>
</tr>
<tr>
<td>Hover over a changeset in a swimlane</td>
<td>Displays the changeset number and all the branches that the changeset is referenced in. This will include branches that you may not have swimlanes displayed for.</td>
</tr>
</tbody>
</table>

Highlighting JIRA issues
Choose **Highlight > JIRA issues** to highlight all the changesets that have a JIRA issue key in the commit message.

This highlight type is only available if you have integrated FishEye with JIRA and linked your repository to a JIRA project.

<table>
<thead>
<tr>
<th>Action</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click on a changeset in the changeset list</td>
<td>Highlights all other changesets that have the same JIRA issue key in the commit message.</td>
</tr>
<tr>
<td>Hover over a changeset in a swimlane</td>
<td>Displays all branches that the changeset is referenced in, and all referenced JIRA issues.</td>
</tr>
</tbody>
</table>

---

Highlighted reviewed changesets

Choose **Highlight > Reviewed changesets** to highlight the changesets that have been reviewed (i.e. included in a Crucible review):

- **Red**: unreviewed, i.e. the changeset is associated with a review in the 'Dead' or 'Rejected' state, or no review is associated.
- **Yellow**: under review, i.e. the changeset is associated with a review not in the 'Dead', 'Rejected' or 'Closed' state.
- **Green**: reviewed, i.e. the changeset is associated with a review in 'Closed' state.

This highlight type is only available if you are using FishEye with Crucible.

<table>
<thead>
<tr>
<th>Action</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click on a changeset in the changeset list</td>
<td>Highlights the changesets that are part of the same review as the selected changeset.</td>
</tr>
<tr>
<td>Hover over a changeset in a swimlane</td>
<td>Displays all branches that the changeset is referenced in, and the Crucible review key.</td>
</tr>
</tbody>
</table>
### Highlighting commits by an author

Choose **Highlight > Author** to highlight all the changesets submitted by a particular author.

<table>
<thead>
<tr>
<th>Action</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click on a changeset in the changeset list</td>
<td>Highlights the changesets that were submitted by the same author.</td>
</tr>
<tr>
<td>Hover over a changeset in a swimlane</td>
<td>Displays the changeset number and all the branches that the changeset is referenced in.</td>
</tr>
</tbody>
</table>

### Highlighting search results

Choose **Highlight > Search** to highlight all the changesets where the commit message contains the search term.

<table>
<thead>
<tr>
<th>Action</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click on a changeset in the changeset list</td>
<td>Highlights the changesets that match the search term.</td>
</tr>
</tbody>
</table>
Hover over a changeset in a swimlane

Displays the changeset number and all the branches that the changeset is referenced in.

1. Hover over a changeset in a swimlane.
2. Displays the changeset number and all the branches that the changeset is referenced in.

Viewing changesets across all branches

The 'All Branches' mode allows you to view commit activity across all branches of a repository. In this mode, the swimlane headers are not displayed. However, you can hover over any changeset to display information about the changeset, as described in the 'Highlighting Information in the Commit Graph' section above.

To see all the repository's branches in the commit graph:

1. Click Select branches... when viewing the commit graph.
2. In the 'Select Branches' dialog, click Switch to all branches mode.

Reordering swimlanes for Git repositories

Reordering swimlanes is useful if you just want to show branches in a certain order. However, ordering swimlanes is vital for Git repositories, as it is the only way to determine which branch a commit is from.

When you view the commit graph for a Git repository, FishEye works from the leftmost swimlane to the right and, for each swimlane, checks if the commit is in that branch:

- If the commit is in the branch, a dot is shown representing the commit.
- If the commit is not in the branch, the dot for the commit is moved to the next column on the right.
For example, if the ‘master’ swimlane is to the left of another swimlane, e.g. ‘fisheye-2.6’ branch, there will be no changesets shown in the ‘fisheye-2.6’ swimlane, as all the commits will be picked up in the ‘master’ swimlane. However, if you move the ‘fisheye-2.6’ swimlane to the left of the ‘master’ swimlane, it will pick up all of the FishEye 2.6 commits.

For more information, read this Knowledge Base article: Ordering of Branches Important When Visualising Git Changeset

**Viewing People's Statistics**

To see charts and activity of everyone who commits code to your FishEye repositories, click the People tab at the top of the screen.

**Screenshot: List of all committers in FishEye**

The All Users screen shows all those with accounts on the system. You can see their commit history (expressed as a bar graph) and their total number of commits.

Click on a person's name to see detailed information about their additions to the repository, and issue updates. If you are using FishEye with Crucible and have JIRA integration set up, you can see their review activity.

**Screenshot: Statistics on a Person in FishEye**
Avatars

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.

Using smart commits

Smart commits allow repository committers to perform actions like transitioning JIRA issues or creating Crucible code reviews by embedding specific commands into their commit messages. Multiple smart commits can be used in one commit message, however they must be on separate lines. Note that smart commits don't provide for field-level updates in JIRA issues.

Note that Smart commits require the following:

- An application link must be configured between FishEye/Crucible and JIRA. See Adding an application link.
- If you have JIRA 5.0 or later, and the JIRA FishEye Plugin (at least version 5.0.10), a project/entity link is unnecessary. Otherwise, a project link must be configured between FishEye/Crucible and JIRA. See Adding project links between applications.
- Smart commits must be enabled in FishEye. See Enabling smart commits.

Transition your JIRA issues
Compatibility

- In order to use smart commits with JIRA you need to have the JIRA FishEye Plugin version 3.4.5 or above installed on your JIRA instance.
- Note that smart commits only support the default JIRA issue key format (that is, two or more uppercase letters, followed by a hyphen and the issue number, for example BAM-123).

On this page:

- Transition your JIRA issues
  - Basic command line syntax
  - Advanced command line syntax
  - Commands
- Integration with Crucible
  - Creating a review
  - Adding reviewers
  - Review objectives
  - Updating an existing review
- Linkers
- Error handling

Related pages:

- Enabling smart commits
- Configuring web hooks
- Transitioning JIRA issues
- Write your own smart commit
- Linkers

Basic command line syntax

The basic command line syntax for your commit comment is:

```<ISSUE_KEY> #<COMMAND> <optional COMMAND_PARAMETERS>```

Please note, commit commands cannot span more than one line (i.e. you cannot use carriage returns).

For example, if you include the following text in your commit message, FishEye will record 2 days and 5 hours of work against issue JRA-123, when you perform your commit:

```
JRA-123 #time 2d 5h
```

Please see the section below for further information on the command line parameters.

Advanced command line syntax

If you wish to perform multiple actions on issues, you can create composite commands by combining keywords, as described below. Please note, commit commands cannot span more than one line (i.e. you cannot use carriage returns).

- To perform multiple actions on a single issue:

```
<ISSUE_KEY> #<COMMAND1> <optional COMMAND1_PARAMETERS> #<COMMAND2> <optional COMMAND2_PARAMETERS> #<COMMAND3> <optional COMMAND3_PARAMETERS> etc
```

For example, if you include the following text in your commit message, FishEye will log 2 days and 5 hours of work against issue JRA-123, add the comment 'Task completed ahead of schedule' and resolve the
issue, when you perform your commit:

```
JRA-123 #time 2d 5h #comment Task completed ahead of schedule #resolve
```

- To perform a single action on multiple issues:

```
<ISSUE_KEY1> <ISSUE_KEY2> <ISSUE_KEY3> #<COMMAND> <optional COMMAND_PARAMETERS> etc
```

For example, if you include the following text in your commit message, FishEye will resolve issues JRA-123, JRA-234 and JRA-345, when you perform your commit:

```
JRA-123 JRA-234 JRA-345 #resolve
```

- To perform multiple actions on multiple issues:

```
<ISSUE_KEY1> <ISSUE_KEY2> <ISSUE_KEY3> #<COMMAND1> <optional COMMAND1_PARAMETERS> #<COMMAND2> <optional COMMAND2_PARAMETERS> #<COMMAND3> <optional COMMAND3_PARAMETERS> etc.
```

For example, if you include the following text in your commit message, FishEye will log 2 days and 5 hours of work against issues JRA-123, JRA-234 and JRA-345, add the comment 'Task completed ahead of schedule' to all three issues, and resolve all three issues, when you perform your commit:

```
JRA-123 JRA-234 JRA-345 #resolve #time 2d 5h #comment Task completed ahead of schedule
```

**Commands**

Note that you can see the custom commands available for use with smart commits by visiting the JIRA issue and seeing its available workflow transitions (in an issue, click View Workflow, near the issue status).
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>#time &lt;n&gt;w &lt;n&gt;d &lt;n&gt;h &lt;n&gt;m &lt;work log comment&gt;</code></td>
<td>Records time tracking information against an issue. Please note, time tracking must be enabled for your JIRA instance to use this command. Please check with your JIRA administrator, if you cannot record time tracking information against issues. <strong>Please note:</strong>&lt;br&gt;• Work log comments cannot be set using smart commits. See <a href="#">F E-3757</a></td>
<td><code>#time 1w 2d 4h 30m Total work logged</code> — this command would record 1 week, 2 days, 4 hours and 30 minutes against an issue, and add the comment 'Total work logged' in the Work Log tab of the issue.</td>
</tr>
<tr>
<td><code>#comment &lt;comment text&gt;</code></td>
<td>Records a comment against an issue.</td>
<td><code>#comment My comment</code> — this command would create the comment, &quot;My comment&quot;, against the issue.</td>
</tr>
<tr>
<td><code>#&lt;workflow command&gt; e.g. #resolve &lt;workflow&gt; &lt;comment text&gt;</code></td>
<td>Transitions an issue to a particular workflow state. Please see the documentation for Configuring Workflow in JIRA.</td>
<td><code>#close Fixed the issue</code> — this command would execute the 'Close Issue' workflow transition for an issue in the default JIRA workflow and adding the comment 'Fixed the issue'.&lt;br&gt;<code>#start</code> — this command would execute the 'Start Progress' workflow transition for an issue in the default JIRA workflow</td>
</tr>
</tbody>
</table>
FishEye will do prefix matching for issue transitioning. For example, if you have transition name with spaces, such as finish work then specifying #finish is sufficient. Hyphens replace spaces: #finish-work

FishEye will only execute issue transitions if there is no ambiguity in valid workflow transitions. Take the following example:
An issue has two valid transitions:
- Start
  - Progress
- Start Review

A smart commit with action #start is ambiguous as FishEye will not be able to determine which transition to execute. In order to execute one of these transitions, the smart commit specified will need to be fully qualified #start-review

Please note: If you want to resolve an issue using the #resolve command, you will not be able to set the resolution via smart commits.

Integration with Crucible

Please note that:
Each commit command in the commit message must not span more than one line (i.e. you cannot use carriage returns). You can use multiple commands in the same message as long as they are on separate lines.

Creating a review

With smart commits, it is also easy to create a Crucible review from a commit:

Fix a bug +review CR-TEST

The command "+review" tells FishEye to create a new review in the project CR-TEST with the content of the changeset. The review will be in a draft state unless the project has default reviewers or reviewers are explicitly mentioned. If you only have one project in Crucible, or a repository is a project's default repository, it is not necessary to mention the project key. Just use "fix a bug +review".

Adding reviewers

Reviewers can be added to a new review using a smart commit:

Fix a bug +review CR-TEST @jcage @skhan

That command will create a new review in PROJ and add the users jcage and skhan to the review. The review will be automatically started if reviewers are specified.

Note, you cannot add reviewers to existing reviews using smart commits.

Review objectives

When creating a new review using a smart commit the default project objectives are added to the review, and since Fishey/Crucible 2.10.2, the commit message is also copied to the review objectives.

Note that you cannot add arbitrary objectives to the review from the smart commit.

Updating an existing review

Often, reviews require rework and changes in response to comments left by the team. When committing these changes, adding the review key will iteratively add these new changes to the review:

Implement rework on past work +review CR-TEST-123

With this command FishEye will add the changeset content to the review CR-TEST-123.

Linkers

When using smart commits you can use linkers that create a hyperlink to the JIRA issue. See Linkers for more information.

Error handling

If there are any errors during the processing of smart commits, they will be logged to FishEye’s error console, as well as emailed to the actioning users. Please speak to your FishEye administrator about Configuring SMTP.

Changing your user profile
You can change FishEye (and Crucible) settings such as password, notifications, profile image and display settings.

To change your FishEye settings:

1. Log into FishEye.
2. Choose Settings from the User Menu (labelled with your username) at the top of the screen.
3. Update your user settings as required. Each tab is described in more detail below.
4. Click Close.

On this page:
- Display Settings
- Profile and Email
- Change Password
- Open Authentication (OAuth)
- Author Mapping
- Watches
- Reviews

Display Settings

<table>
<thead>
<tr>
<th>Display Settings</th>
<th>File history view mode</th>
<th>Default is Logical. In Subversion repositories, FishEye is able to show all operations on a single logical file spread across a number of physical paths - i.e. operations in different branches. When this is set to Logical, FishEye will show all the operations across all branches. In Physical mode, only the operations related to the physical path whose history is being viewed are shown.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timezone</td>
<td>Default is the timezone of the FishEye server.</td>
<td></td>
</tr>
<tr>
<td>Changelog</td>
<td>Changesets per page</td>
<td>The default is 30 per page.</td>
</tr>
<tr>
<td></td>
<td>Always expand changesets in stream</td>
<td>Default is Yes.</td>
</tr>
<tr>
<td>Diff view</td>
<td>Diff mode</td>
<td>Default is Unified.</td>
</tr>
<tr>
<td></td>
<td>Line wrapping</td>
<td>Default is None i.e. long lines will never word-wrap. Soft is when long lines will word-wrap.</td>
</tr>
<tr>
<td></td>
<td>Context lines</td>
<td>Default is 3. The number of lines to show (for context), if the diff contains more than three lines of code.</td>
</tr>
<tr>
<td>Source view</td>
<td>Default annotation mode</td>
<td>Default is Age.</td>
</tr>
<tr>
<td>Highlighting colours</td>
<td>The default scheme uses bright colours for highlighting diffs in the code. If you prefer more muted colours, select <strong>Classic (muted)</strong>.</td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Tab width</td>
<td>Default is 8. Can be changed to a number between 1 and 10.</td>
<td></td>
</tr>
</tbody>
</table>

### Profile and Email

<table>
<thead>
<tr>
<th><strong>Email settings</strong></th>
<th><strong>Display Name</strong></th>
<th>Name displayed for the user currently logged in.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Email Address</strong></td>
<td></td>
<td>The address to which all email notifications will be sent.</td>
</tr>
<tr>
<td><strong>Email Format</strong></td>
<td></td>
<td>Default is <strong>Text</strong>. Can be sent as <strong>HTML</strong>.</td>
</tr>
<tr>
<td><strong>Email watches</strong></td>
<td><strong>Send Watch Emails</strong></td>
<td>The frequency at which emails will be sent for watch notifications. <strong>Immediately</strong> is the default value. <strong>Daily</strong> sends a summary of changes.</td>
</tr>
<tr>
<td><strong>Profile Picture</strong></td>
<td><strong>Choose picture</strong></td>
<td>Upload an avatar image of your choice. This image will be displayed next to your username throughout FishEye/Crucible. Accepted formats are JPG, GIF and PNG. Image file size limit is 2Mb. Images will be automatically cropped on upload. This is disabled if avatars are served from an external server – see Configuring avatar settings.</td>
</tr>
</tbody>
</table>

### Change Password

Change your password from this tab, if required. Please note that passwords are case-sensitive.

⚠️ This tab is not displayed if your FishEye instance is connected to an external LDAP authentication source, such as LDAP. You will need to contact your administrator for assistance.

### Open Authentication (OAuth)

Configure your OAuth settings on this page. You can choose to allow gadgets/applications to access FishEye data using your account.

Read more about OAuth.

### Author Mapping

The **Author Mapping** tab allows you to make an association between you (as a logged-in user) and a committer, for each repository.
This is only necessary if the name or email of the user within FishEye is different from the committer name or email within the repository. By default, FishEye will automatically match users to committers where it can.

Watches

By adding a 'watch', you can ask to receive emails about changes made to the repository. Any watches that you have set up in FishEye/Crucible will be displayed on this tab. You can watch the dashboard activity stream, changelogs and repositories. Watching an activity stream/repository allows you to receive emails when updates occur. Note, the option to add a watch may only be available if the administrator has enabled watches for the repository.

You can delete any of your watches by clicking Delete next to the watch.

Reviews

This functionality is used by Crucible.

If the SMTP server is set up, then you will receive emails when different actions occur within Crucible.

You can change the options described below, to specify the stages at which emails will be sent.

<table>
<thead>
<tr>
<th>Auto-mark files as 'read'</th>
<th>Default is Yes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review Notifications Events</td>
<td>State change</td>
</tr>
<tr>
<td></td>
<td>Comment added</td>
</tr>
<tr>
<td></td>
<td>Comment reply added</td>
</tr>
<tr>
<td></td>
<td>Participant finished</td>
</tr>
<tr>
<td></td>
<td>General message</td>
</tr>
<tr>
<td></td>
<td>File revision added</td>
</tr>
<tr>
<td></td>
<td>Uncomplete review if defect is raised:</td>
</tr>
<tr>
<td></td>
<td>Uncomplete review if revision is added:</td>
</tr>
</tbody>
</table>
1. **My actions**  
   Default is **No**. If set to **Yes**, an email is sent every time you perform an action on a review.

---

⚠️ **Batch Notifications** will be sent out by Crucible every 30 minutes. All notifications will be rolled up into a single digest email.

---

**Screenshot: User Profile Settings**

---

**Re-setting your password**

If you need to reset your password, FishEye has an integrated mechanism to generate a new password and send it to the email address in your profile.

**To reset your password:**

1. On the log in screen, click the **Forgot your password?** link. The ‘Request New Password’ screen opens.
2. Fill out your username or email address and the **Captcha** step. That is, click in the form field labelled **Please enter the word as shown below** and type the graphical letters shown above the **Submit** button.
3. An email is then sent to the email address specified in your profile. When it arrives, click the link supplied to complete the password reset.
4. On the resulting web page, you will receive the message ‘A new password has been sent to your account.’
5. An email will arrive in your inbox, containing your new password.

**i** If you receive a password-reset email that you did not request, simply disregard it to continue using your current password.

---

**Screenshot: The Log In dialog**

---

Created by Atlassian in 2013. Licensed under a **Creative Commons Attribution 2.5 Australia License**.
FishEye supports a powerful type of regular expression for matching files and directories (same as the pattern matching in Apache Ant).

These expressions use the following wild cards:

<table>
<thead>
<tr>
<th>Wild Card</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Matches one character (any character except path separators)</td>
</tr>
<tr>
<td>*</td>
<td>Matches zero or more characters (not including path separators)</td>
</tr>
<tr>
<td>**</td>
<td>Matches zero or more path segments.</td>
</tr>
</tbody>
</table>

Remember that Ant globs match paths, not just simple filenames.

- If the pattern does not start with a path separator i.e. / or \, then the pattern is considered to start with /** /.
- If the pattern ends with / then ** is automatically appended.
- A pattern can contain any number of wild cards.

Also see the Ant documentation.

Examples
**Date Expressions Reference Guide**

FishEye supports a wide variety of date expressions. A date has the two possible general forms:

- `DATE[+-]TIMEZONE[+-]DURATION`, or
- `DATECONSTANT[+-]DURATION`.

The **TIMEZONE** and **DURATION** parts are both optional.

**TIMEZONE** can be an offset from GMT `HHMM` or `HH:MM`, or simply the letter `Z` to denote GMT. If no timezone is given, the FishEye server's configured timezone is used.

**DATE** can be either of the following:

<table>
<thead>
<tr>
<th>YYYY-MM-DDThh:mm:ss</th>
<th>Specifies a time and date (separated by a <code>T</code>). The seconds part may contain a fractional component. A <code>/</code> can be used instead of <code>-</code> as a separator.</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYYY-MM-DD</td>
<td>Specifies 00:00:00 on the given date. A <code>/</code> can be used instead of <code>-</code> as a separator.</td>
</tr>
</tbody>
</table>

**DATECONSTANT** can be any of:

<table>
<thead>
<tr>
<th>now</th>
<th>This very instant (at the time the expression was evaluated).</th>
</tr>
</thead>
<tbody>
<tr>
<td>today</td>
<td>The instant at 00:00:00 today. (server-time* or GMT)</td>
</tr>
<tr>
<td>todaygmt</td>
<td>The instant at 00:00:00 today. (server-time* or GMT)</td>
</tr>
<tr>
<td>thisweek</td>
<td>The instant at 00:00:00 on the first day of this week. Sunday is considered the first day. (server-time* or GMT)</td>
</tr>
<tr>
<td>thisweekgmt</td>
<td>The instant at 00:00:00 on the first day of this week. Sunday is considered the first day. (server-time* or GMT)</td>
</tr>
<tr>
<td>thismonth</td>
<td>The instant at 00:00:00 on the first day of this month. (server-time* or GMT)</td>
</tr>
<tr>
<td>thismonthgmt</td>
<td>The instant at 00:00:00 on the first day of this month. (server-time* or GMT)</td>
</tr>
<tr>
<td>thisyear</td>
<td>The instant at 00:00:00 on the first day of this year. (server-time* or GMT)</td>
</tr>
<tr>
<td>thisyeargmt</td>
<td>The instant at 00:00:00 on the first day of this year. (server-time* or GMT)</td>
</tr>
</tbody>
</table>
* The timezone used for server-time is part of the FishEye configuration

The syntax for DURATION is similar to the XML Schema duration type. It has the general form PnYnMnDTnHnMnS. See Section 3.2.6 of the XML Schema Datatypes document for more details.

**Examples**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005-01-02</td>
<td>The start of the day on January 1, 2005 (server's timezone)</td>
</tr>
<tr>
<td>2005-01-02-0500</td>
<td>The start of the day on January 1, 2005 at GMT offset -0500 (New York)</td>
</tr>
<tr>
<td>2005-01-02T12:00:00Z</td>
<td>Midday, January 1, 2005 GMT</td>
</tr>
<tr>
<td>today-P1D</td>
<td>Yesterday (start of day)</td>
</tr>
<tr>
<td>today+P1D</td>
<td>Start of tomorrow</td>
</tr>
<tr>
<td>thismonth-P1M</td>
<td>Start of last month</td>
</tr>
<tr>
<td>thisyear+P1Y</td>
<td>Start of next year</td>
</tr>
<tr>
<td>now-PT1H</td>
<td>One hour ago</td>
</tr>
<tr>
<td>now+PT1H2M3S</td>
<td>One hour, two minutes and three seconds from now</td>
</tr>
</tbody>
</table>

**EyeQL Reference Guide**

FishEye contains a powerful query language called EyeQL. EyeQL is an intuitive SQL-like language that allows you to write your own specific queries. See examples.

EyeQL allows you to perform complex searches either within the Advanced Search or incorporated in scripts when programming the FishEye API.

query:
```
select revisions
  (from (dir|directory) word)?
  (where clauses)?
  (order by date (asc | desc) ? )?
  Notes: asc produces 'ascending order'.
  desc produces 'descending order'.
  (group by (file|dir|directory|csid|changeset))? (return return-clauses)? (limit limit-args)?

clauses:
  clause ((or|and|,) clause)*
  Notes:
  and binds more tightly than or.
  ',' (comma) means 'and'.

clause:
  ( clauses )

not clause

path (not)? like word
Notes:
word is an Antglob.

**path =** word
Notes:
Defines an exact path without wildcards or variables. **path** must represent a complete (hard-coded) path.

**path !=** word
Notes:
Defines an exact path exclusion without wildcards or variables. **path** must represent a complete (hard-coded) path.

**date in ( ( [ ] dateExp, dateExp ( ) | ) )**
Notes: The edges are
inclusive if [ or ] is used.
exclusive if ( or ) is used.

**date dateop dateExp**
Notes:
**dateop** can be <, >, <=, >=, =, == or !=.

**author =** word

**author in (word-list)**

**comment matches word**
Notes:
Does a full-text search.

**comment = string**
Notes:
Matches **string** exactly.
Most comments end in a new line, so remember to add \n at the end of your string.

**comment =~ string**
Notes:
**string** is a regular expression.

**content matches word**
Notes:
Does a full-text search.
At this time searches are restricted to HEAD revisions.

**(modified|added|deleted)? on branch word**
Notes:
Selects all revisions on a branch.
**modified** excludes the branch-point of a branch.
**added** selects all revisions on the branch if any revision was added on the branch.
**deleted** selects all revisions on the branch if any revision was deleted on the branch.

**tagged op? word**
Notes:
If op can be <, >, <=, >=, =, == or !=.

op defaults to == if omitted.

These operators are 'positional' and select revisions that appear on, after, and/or before the given tag.

**between tags** tag-range

**after tag** word

**before tag** word

**is head (on word)?**

Notes:

This selects the top-most revision on any branch, if no branch is specified.

**is ( dead | deleted )**

Notes:

Means the revision was removed/deleted.

**is added**

Notes:

Means the revision was added (or re-added).

**csid = word**

Notes:

Selects all revisions for the given changeset ID.

**p4:jobid = word**

Notes: finds revisions whose Perforce jobid is word.

**p4:jobid =~ word**

Notes: finds revisions whose Perforce jobid matches regex word.

**reviewed**

Notes: *(applies to Crucible reviews)* alias for in or before any closed review.

**(in | before | in or before) review word**

**(in | before | in or before) any (review states)? review**

Notes:

word is a review key.

in selects reviewed revisions. If a review contains a diff, then only the most recent revision is considered in the review.

before selects all revisions in a file prior to the revision in the review.

review states is a comma-separated list of open, closed, draft.

**tag-range:**

(( | [ ) T1:word, T2:word ( | ] )

Notes:

A range of revisions between those tagged T1 and T2.

The edges are:

- inclusive if [ or ] is used.
- exclusive if ( or ) is used.

You can mix edge types. These are all valid: \((T1,T2), [T1,T2], (T1,T2) and [T1,T2)\).

**Having trouble with Subversion tags?** See How tags work in Subversion for more information.
word:
Any string, or any non-quoted word that does not contain white space or any other separators.

string:
A sequence enclosed in either " (double quotes) or ' (single quotes).
The following escapes work: \' \" \n \r \t \b \f.
Unicode characters can be escaped with \uXXXX.
You can also specify strings in 'raw' mode like r"foo". (Similar to Python's raw strings. See Python's own documentation).

dateExp:
See our Date Expressions Reference Guide for more information on date formats.

return-clauses:
return-clause (, return-clause)*
A return clause signifies that you want control over what data is returned/displayed.

return-clause:
(path | dir | directory | revision | author | date | comment | csid | isBinary | totalLines | linesAdded | line emoved | isAdded | isDeleted | isCopied | isMoved | tags | reviews | aggregate)
(as word)?
The attribute to return, optionally followed by a name to use for the column.

Notes: reviews applies to Crucible reviews.

aggregate-return-field:
(count(revisions ) | count( binary-field ) | count(distinct other-field ) | sum( numeric-field ) | average( numeric-field ) | max( numeric-field ) | min( numeric-field ) )
The aggregate field to return.

Notes:
binary-fields are isBinary, isAdded, isDeleted, isCopied, isMoved. e.g. count(isAdded) will return the number of added files.
numeric-fields are totalLines, linesAdded, linesRemoved.
other-field can be path, dir, author, date, csid, tags or reviews. e.g. count(distinct path) will return the number of unique paths. count(distinct tags) will return the number of unique tags.

If a group by is given, give sub-totals for each group.

With no group by clause, you can have:
• return normal columns
• return aggregates

With a group by changeset|csid clause:
• return normal columns
• return csid, comment, date, author, aggregates

With a group by file|path clause:
• return normal columns
• return path, aggregates

With a group by dir|directory clause:
• return normal columns
• return dir, aggregates
i.e. The EyeQL can contain a **returns** clause that contains all non-aggregate columns, or all aggregate columns. Non-aggregate and aggregate columns can only be mixed if the columns are unique for the grouping.

**limit-clause:**

```
( length | offset, length | length offset offset )
```

Notes: Limits the number of results to return. **offset** specifies the starting point of the truncated result set and **length** specifies the set length. **offset** is zero-based.

**Examples**

The following examples demonstrate using EyeQL to extract information from your repository.

**Find files removed on the Ant 1.5 branch:**

```sql
select revisions where modified on branch ANT_15_BRANCH and is dead group by changeset
```

**As above, but just return the person and time the files were deleted:**

```sql
select revisions where modified on branch ANT_15_BRANCH and is dead return path, author, date
```

**Find files on branch and exclude delete files:**

```sql
select revisions where modified on branch ANT_15_BRANCH and not is deleted group by changeset
```

**Find changes made to Ant 1.5.x after 1.5FINAL:**

```sql
select revisions where on branch ANT_15_BRANCH and after tag ANT_MAIN_15FINAL group by changeset
```

**Find changes made between Ant 1.5 and 1.5.1:**

```sql
select revisions where between tags (ANT_MAIN_15FINAL, ANT_151_FINAL] group by changeset
```

**As above, but show the history of each file separately:**

```sql
select revisions where between tags (ANT_MAIN_15FINAL, ANT_151_FINAL] group by file
```

**Find Java files that are tagged ANT_151_FINAL and are head on the ANT_15_BRANCH:** (i.e. files that haven’t changed in 1.5.x since 1.5.1)

```sql
select revisions from dir /src/main where is head and tagged ANT_151_FINAL and on branch ANT_15_BRANCH and path like *.java group by changeset
```

**Find changes made by conor to Ant 1.5.x since 1.5.0:**

```sql
select revisions where between tags (ANT_MAIN_15FINAL, ANT_154] and author = conor group by changeset
```

**Find commits that do not have comments:**

```sql
select revisions from dir / where comment = "" group by changeset
```

**Find the 10 most recent revisions:**

```sql
select revisions order by date desc limit 10
```

**Find the 5th, 6th & 7th revisions:**

```sql
select revisions order by date limit 4, 3
```

**Find commits between two dates:**

```sql
select revisions where date in [2008-03-08, 2008-04-08]
```

**Find revisions that do not have any associated review:**

```sql
select revisions where (not in any review)
Return number of matched revisions, the number of files modified, authors who modified code, changesets, tags, and reviews:

```sql
select revisions
where date in [ 2003-10-10, 2004-12-12 ]
return count(revisions), count(distinct path), count(distinct author),
count(distinct csid), count(distinct tags), count(distinct reviews)
```

As Sub-totals for each distinct changeset, Return csid, the author, date, comment, number of matched revisions, the number of files modified, the lines added/removed:

```sql
select revisions
where date in [ 2003-10-10, 2004-12-12 ]
group by changeset
return csid, author, date, comment, count(revisions), count(distinct path),
sum(linesAdded), sum(linesRemoved)
```

For each matched file, return the file name, number of matched revisions, the lines added/removed:

```sql
select revisions
where date in [ 2003-10-10, 2004-12-12 ]
group by file
return path, count(revisions), sum(linesAdded), sum(linesRemoved)
```

Show all the changesets with no review:

```sql
select revisions
from dir /
where not reviewed
group by changeset
return csid, author, count(revisions), comment
```

FishEye Release Notes

**FishEye 3.0**

30 May 2013

- Optimised indexing for new SVN repositories
- Commit graph highlighter
- Redesigned UX
- Platform upgrades: Jetty 8, Infinity 3 DB
- Small improvements
- More in the release notes and changelog

See the FishEye upgrade guide.

**FishEye 2.10**

15 January 2013

- Repository indexing REST API
- Bitbucket and GitHub polling integration
More in the release notes
FishEye 2.10 upgrade guide

FishEye 2.9

14 November 2012

- Simpler JIRA integration
- More JIRA data in FishEye
- Faster JIRA source tab
- More in the release notes
- FishEye 2.9 upgrade guide

FishEye 2.8

15 August 2012

- Mentions
- Shares
- Improved performance of the activity stream
- Support for Subversion 1.7
- End of life announcements
- More in the release notes
- FishEye 2.8 upgrade guide

FishEye 2.7

7 September 2011

- Managed Git Repositories
- Smart Commits
- Web Hooks
- JIRA FishEye Plugin Improvements
- Small Improvements
- More in the release notes
- FishEye 2.7 Upgrade Guide

FishEye 2.6

6 June 2011

- Repository Commit Graph
- User Management via JIRA
- Improved Quick Search
- Redesigned HTML Emails
- Dashboard and Navigation Improvements
- Improved Support for Git Branches
- Git Commit Authors include Email Address
- Mercurial Indexing Improvements
- More in the release notes
- FishEye 2.6 Upgrade Guide

FishEye 2.5

8 February 2011
• Search Revamp
• Redesigned Activity Stream
• Mercurial and Git Authentication
• RSS Improvements
• Universal Plugin Manager
• More in the release notes
• FishEye 2.5 Upgrade Guide

FishEye 2.4

20 October 2010
• Branch and Tag Selector
• File History Redesigned
• Easier Application Linking
• SSL Support
• User Interface Improvements
• Performance Improvements
• More in the release notes
• FishEye 2.4 Upgrade Guide

FishEye 2.3

26 May 2010
• Mercurial SCM Alpha
• New 'Aggregate' functions in EyeQL query language
• Revamped Installation Process
• More in the release notes
• FishEye 2.3 Upgrade Guide

FishEye 2.2

18 Feb 2010
• Enhanced Side-by-Side Diff View Mode
• Improved Quick Navigation
• Annotation Context Menu
• Code Copying
• ClearCase and Git Support Now Final
• Numerous improvements and bug fixes
• More in the release notes
• FishEye 2.2 Upgrade Guide

FishEye 2.1

12 November 2009
• Wiki Markup in Commit Messages
• Streamlined JIRA Integration
• FishEye Admin API
• History Page Performance Increases
• ClearCase Support Now in Beta
• More in the release notes
**FishEye 2.1 Upgrade Guide**

FishEye 2.0

*30 June 2009*

- Activity streams
- People statistics
- Favourites, bookmarks & saved search
- Enhanced JIRA integration
- New user interface
- Git beta
- More in the release notes
- FishEye 2.0 Upgrade Guide

FishEye 1.6

*23 September 08*

- FishEye search enhancements
- Multiple admin users
- Remote API improvements
- Changes to charts
- Perforce performance tweaks
- More in the release notes
- FishEye 1.6 Upgrade Guide

FishEye 1.5

*14 April 2008*

- Per-author lines of code statistics
- Charting improvements
- Customisable email templates
- More in the release notes
- FishEye 1.5 Upgrade Guide

FishEye 1.4

*5 December 2007*

- Enhancements to user management
- Crowd/SSO support
- Crucible integration
- Enhancements to JIRA plugin
- More in the release notes
- FishEye 1.4 Upgrade Guide

FishEye 1.3

*1 August 2007*

- Support for the Perforce version control system.
- SVN properties are now shown.
- Quicksearch now searches for changeset ids.
- New "mixed" chart on annotation pages, showing author-over-time breakdown.
- Side by Side diffs (1.3.1)
FishEye upgrade guide

This page describes how to upgrade to a new version of FishEye.

- For the latest FishEye release, see FishEye release notes.
- For production environments we recommend that you test the FishEye upgrade on a QA server before deploying to production.
- The first time you run a new version of FishEye, it will automatically upgrade its data. This may involve a complete re-index of your repository.

On this page:

- Upgrade steps
- Upgrading to FishEye 3.0
- Checking for known issues and troubleshooting the FishEye upgrade

Upgrade steps

This section provides general instructions for upgrading FishEye. See also the specific notes on this page for the version of FishEye you are upgrading to. We *strongly recommend* that you upgrade FishEye by following these steps:

- Back up your *entire* FishEye instance (see Backing up and restoring FishEye data), i.e.
  - If you are backing up your FishEye instance using the Admin interface, tick all of the 'Include' checkboxes (e.g. repository and application caches, plugins and their configuration data, SQL database, etc).
  - If you are backing up your FishEye instance using the command-line interface, do not use any exclusion options.
- Read the release notes and changelogs and the version-specific upgrade guide for the version you are upgrading to, as well as any versions you are skipping.
- Check the Supported platforms to ensure that your system meets the requirements for the new version.
- Check for known issues in the FishEye Knowledge Base.
- If you are using MySQL, read about the JDBC driver.

Your upgrade procedure depends on whether you are using a FISHEYE_INST directory (i.e. "FishEye instance"
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 <New FishEye home directory>/var and <New FishEye home directory>/cache directories.
6. Copy (or move) the <FishEye home directory>/var directory to <New FishEye home directory>.
Copy (or move) the <FishEye home directory>/cache directory to <New FishEye home directory>/cache.

Start FishEye/Crucible from the new installation by running <New FishEye home directory>/bin/run.sh. (Use run.bat on Windows.)

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

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>/var 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.

Upgrading to FishEye 3.0

Please also see the Upgrade steps section above.

Please note the following changes in FishEye 3.0:

Jetty 8

FishEye 3.0 now uses Jetty 8 as its web server and Java servlet container. This change should be completely transparent when you upgrade to FishEye 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
FishEye 3.0 now uses the InfinityDB 3.0 database internally to provide improved performance for concurrent access to FishEye. This change is transparent to users in all respects.

**Pipelined indexing**

FishEye 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 FishEye functionality, such as repository browsing, 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 FishEye logs and admin screens.

### Known issues (13 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>FE-4677</td>
<td>ERROR - Repository scanner paused.</td>
<td>Open</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4657</td>
<td>Linecount calculator issues</td>
<td>Open</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4719</td>
<td>Some custom JIRA issue key patterns not linked in activity stream</td>
<td>Open</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4698</td>
<td>Application Navigator uses Application URL instead of Display URL from Application Link</td>
<td>Open</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4674</td>
<td>Repository list empty until indexing completes and throws exception</td>
<td>Open</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4506</td>
<td>Fisheye can break quite badly if you create a repository with a long name (longer than 100 chars) and then try to add a committer mapping</td>
<td>Open</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4127</td>
<td>Reports UI - Cannot switch to Area chart from Line chart</td>
<td>Open</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4082</td>
<td>Change and Pie Charts</td>
<td>Open</td>
</tr>
</tbody>
</table>
Checking for known issues and troubleshooting the FishEye upgrade

If something is not working correctly after you have completed the steps above to upgrade your FishEye installation, please check for known FishEye 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 FishEye after we have released the software. In such cases we publish information about the known issues in the FishEye Knowledge Base. Please check the FishEye and Crucible Known Issues in the FishEye Knowledge Base and follow the instructions to apply any necessary patches if necessary.

- **Did you encounter a problem during the FishEye upgrade?** Please refer to the guide to troubleshooting upgrades in the FishEye 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.

### FishEye 3.0 release notes

30 May 2013

Today we're excited to release FishEye 3.0, which introduces huge indexing performance gains – especially for Subversion repositories – and search and code visualisation upgrades to help you track changes across your project. We think development teams will be able to work faster, every day.

If you are upgrading from an earlier version of FishEye, please read the FishEye upgrade guide.

The FishEye 3.0 changelog is at the bottom of this page.

Use SVN repositories sooner

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 FishEye, such as viewing recent changesets or commits in JIRA issues, up to 15X sooner than in FishEye 2.10. You can get on with your work, while FishEye quietly completes the fine details of indexing in the background.

[Read more...]
Commit graph highlighter

We’ve added two new highlight types to the commit graph for a repository:

- use the **Author** highlight type to see all the changesets submitted by a particular author.
- use the **Search** highlight type to see the changesets with commit messages that contain the search term.

Read more...

Redesigned user experience

We’ve made a few key design changes to help you work faster with your repositories:

**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, on the left of the header, connects you directly to your other applications, such as JIRA and Bamboo. Now you can switch between FishEye and JIRA – or any other Atlassian application – all from the FishEye 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 Source tab redesign**

We wanted to simplify the experience but still give you the important information you need to make decisions around your issues.
Small improvements

Faster browsing for large teams

FishEye now uses the InfinityDB 3.0 internal database to provide improved performance for concurrent access. Your team gets a better browsing experience!

Diff hunk shortcuts

We've changed the 'j' and 'k' keyboard shortcuts to now work with diff hunks in the changeset page, rather than with files, as previously.

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.

Updated Jetty 8 web server

FishEye 3.0 includes an upgrade from Jetty 6 to Jetty 8 as its web server and Java servelet container.

Change log

This section will contain information about the FishEye 3.0 minor releases as they become available. These releases will be free to all customers with active FishEye software maintenance.

If you are upgrading from an earlier version of FishEye, please read the FishEye upgrade guide.

The issues listed below are just the highlights of all those that have been resolved for the FishEye 3.0.x releases.

2 July 2013 - FishEye 3.0.2 (3 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚸</td>
<td>FE-4704</td>
<td>&quot;All repo&quot; user prefs cause problems on Oracle</td>
</tr>
<tr>
<td>🚸</td>
<td>FE-4687</td>
<td>NullPointerException in Svn2Scanner</td>
</tr>
<tr>
<td>🚸</td>
<td>FE-4682</td>
<td>Native SVN JavaHL broken</td>
</tr>
</tbody>
</table>
### 6 June 2013 - FishEye 3.0.1 (4 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-4683</td>
<td>NullPointerException when Repo incorrectly configured</td>
</tr>
<tr>
<td></td>
<td>FE-4679</td>
<td>ClassCastException with native SVN client</td>
</tr>
<tr>
<td></td>
<td>FE-4661</td>
<td>'No previous activity found' appearing twice on empty activity streams</td>
</tr>
<tr>
<td></td>
<td>FE-4658</td>
<td>Create/Edit Project is slow when there are many users</td>
</tr>
</tbody>
</table>

### 28 May 2013 - FishEye 3.0.0 (40 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-4544</td>
<td>Git client doc does not have any instructions</td>
</tr>
<tr>
<td></td>
<td>FE-4503</td>
<td>Doc Update: Update &quot;Using Dashboard&quot;</td>
</tr>
<tr>
<td></td>
<td>FE-4466</td>
<td>improve documentation for SVN symbolic rules</td>
</tr>
<tr>
<td></td>
<td>FE-4311</td>
<td>Update documentation to inform customers that FishEye won't work with SVN 1.7.x and native client.</td>
</tr>
<tr>
<td></td>
<td>FE-4512</td>
<td>provide the ability to get statistics of indexing state</td>
</tr>
<tr>
<td></td>
<td>FE-4456</td>
<td>Change semantics of j/k shortcuts in FishEye changeset page to cycle through diffs, rather than files</td>
</tr>
<tr>
<td></td>
<td>FE-3799</td>
<td>p4 support: Cleanup and Merge the p4 ancestry suppression flag to 2.6/7/default</td>
</tr>
<tr>
<td></td>
<td>FE-4695</td>
<td>Pre-commit iterative reviews</td>
</tr>
<tr>
<td></td>
<td>FE-4465</td>
<td>Add Application Navigator on BTF version</td>
</tr>
<tr>
<td></td>
<td>FE-4557</td>
<td>Documentation Should Indicate That 'populate' Means 'copy from commit message'</td>
</tr>
<tr>
<td>FE-4556</td>
<td>Indexing Hooks for Bitbucket and GitHub</td>
<td></td>
</tr>
<tr>
<td>FE-4452</td>
<td>Highlight row in Fisheye permission screen</td>
<td></td>
</tr>
<tr>
<td>FE-4450</td>
<td>Slow activity stream on Fisheye dashboard</td>
<td></td>
</tr>
<tr>
<td>FE-4083</td>
<td>Review smart commit should populate Crucible Objectives</td>
<td></td>
</tr>
<tr>
<td>FE-3611</td>
<td>Improve the way avatars are loaded to avoid unnecessary calls</td>
<td></td>
</tr>
<tr>
<td>FE-3045</td>
<td>Improve UX of Branch Selector</td>
<td></td>
</tr>
<tr>
<td>FE-1182</td>
<td>Server/Repo restart is not required after changing the poll period value</td>
<td></td>
</tr>
<tr>
<td>FE-490</td>
<td>Allow the poll period for a repository to have a configurable global default</td>
<td></td>
</tr>
<tr>
<td>FE-425</td>
<td>Make file status (modify, add, delete) stand out in changelog of JIRA plugin</td>
<td></td>
</tr>
<tr>
<td>FE-4639</td>
<td>Command-line application start fails to pull $FISHEYE_ARGS (Linux only)</td>
<td></td>
</tr>
<tr>
<td>FE-4634</td>
<td>Spurious 'branches' may appear within branches on the commit graph for SVN repos</td>
<td></td>
</tr>
<tr>
<td>FE-4625</td>
<td>Slow git output processing leads to long indexing time</td>
<td></td>
</tr>
<tr>
<td>FE-4598</td>
<td>&quot;Create crucible review&quot; missing in &quot;Source&quot; tab in JIRA, if also using Stash integration</td>
<td></td>
</tr>
<tr>
<td>FE-4590</td>
<td>Requests to dashboard and user list block when repositories are upgrading</td>
<td></td>
</tr>
<tr>
<td>FE-4567</td>
<td>Email validation fails if local-part includes a '#'</td>
<td></td>
</tr>
<tr>
<td>FE-4522</td>
<td>Allow FishEye notification to use the user's email address when it is set in the SMTP configuration</td>
<td></td>
</tr>
<tr>
<td>FE-4520</td>
<td>SvnClient instances are not</td>
<td></td>
</tr>
</tbody>
</table>
FishEye 2.10 release notes

15 January 2013

Atlassian is proud to present FishEye 2.10, which provides further improvements in performance.

- Visit our issue tracker to see the full list of improvements and bug fixes in FishEye and Crucible for this release.
- See the change log for FishEye 2.10.x minor releases.
- Upgrading from a previous version of FishEye. Upgrading FishEye should be fairly straight forward. We strongly recommend that you back up FishEye before upgrading. Please refer to the FishEye 2.10 Upgrade Guide for essential information about upgrading.
Known issues. Please check the important technical advisories on the front page of the Knowledge Base for information about any known issues for this release.

JIRA 5.0 integration. The features described below are supported by JIRA 5.0, or later, with the latest version of the JIRA FishEye plugin.

Highlights of this release:

- Repository Indexing REST API
- Bitbucket and GitHub polling integration
- Other announcements

Providing feedback:

Please log your votes and issues. They help us decide what needs doing, and are much appreciated!

1

Repository Indexing REST API

With FishEye 2.10, you can now optimize your instance by using the REST API to trigger indexing when it's necessary. Instead of configuring FishEye to poll each repository every minute, you can add a post-commit or post-receive hook to your repositories which will ping FishEye when new commits need to be indexed. Read more...

2

Bitbucket and GitHub polling integration

We have implemented indexing hooks for Bitbucket and GitHub. Just provide the details of your repository and FishEye will be pinged whenever new commits need to be indexed. Read more...
Other announcements

- Improved performance of the repositories listing in the administration interface.
- Smart commits can now populate review objectives. More...
- The CVS poll periods can be configured. More...

The FishEye 2.10 team

Development

Core team
Brendan Humphreys
Conor MacNeill
Geoff Crain
Lukasz Pater
Maciej Swinarski
Piotr wity wicicki
Richard Stephens
Tom Davies
Valery Trubnikov

Team lead
Nick Pellow

Product management
Sten Pittet

Project manager
Anton Mazkovoi
Support
Ajay Sridhar
Daniel Rohan
Felipe Kraemer
Gurleen Anand
Kah Loun Foong
Malik Mangier
Patrick Hill
Renan Battaglin
Rene Verschoor
Ricardo Martins

Others

Product marketing
Giancarlo Lionetti

Technical writing
Paul Watson

Operations
James Fleming

FishEye 2.10 changelog
This page will contain information about the FishEye 2.10 minor releases as these become available. Crucible license holders should also check the Crucible 2.10 changelog. These releases will be free to all customers with active FishEye software maintenance.

Don’t have FishEye 2.10 yet?
Take a look at all the features in the FishEye 2.10 release notes and see what you are missing out on!

Upgrading from a previous version of FishEye
If you are upgrading, please read the FishEye 2.10 upgrade guide.

06 June 2013 - Fisheye 2.10.5 (2 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>📜</td>
<td>FE-4658</td>
<td>Create/Edit Project is slow when there are many users</td>
<td>🔄 Closed</td>
</tr>
<tr>
<td>📜</td>
<td>FE-4625</td>
<td>slow git output processing leads to long indexing time</td>
<td>🔄 Closed</td>
</tr>
</tbody>
</table>
### 02 April 2013 - FishEye 2.10.4 (4 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>FE-4590</td>
<td>Requests to dashboard and user list block when repositories are upgrading</td>
<td>✅ Closed</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4567</td>
<td>Email validation fails if local-part includes a '#'</td>
<td>✅ Closed</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4522</td>
<td>Allow FishEye notification to use the user's email address when it is set in the SMTP configuration</td>
<td>✅ Closed</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4507</td>
<td>Migration/Integration with MySQL 5.6 fails with &quot;Specified key was too long&quot; error</td>
<td>✅ Closed</td>
</tr>
</tbody>
</table>

### 19 Mar 2013 - FishEye 2.10.3 (2 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>FE-4450</td>
<td>Slow activity stream on Fisheye dashboard</td>
<td>✅ Closed</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4479</td>
<td>FishEye Not Fully Compatible With Mercurial 2.5</td>
<td>✅ Closed</td>
</tr>
</tbody>
</table>

### 25 Feb 2013 - FishEye 2.10.2 (3 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>FE-4083</td>
<td>Review smart commit should populate Crucible Objectives</td>
<td>✅ Closed</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-4483</td>
<td>NullPointerException if Crucible project key contains &quot;-&quot;</td>
<td>✅ Closed</td>
</tr>
<tr>
<td>🔄</td>
<td>FE-3901</td>
<td>Rest transactions are rolled back after they've been committed</td>
<td>✅ Closed</td>
</tr>
</tbody>
</table>

### 29 Jan 2013 - FishEye 2.10.1 (1 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>FE-4424</td>
<td>Update on 'Store diff'</td>
<td>✅ Closed</td>
</tr>
</tbody>
</table>
## Documentation for FishEye 3.0

15 Jan 2013 - Fisheye 2.10.0 (29 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>FE-4191</td>
<td>Add built-in support for use of proxy to contact external plugins site</td>
<td>Closed</td>
</tr>
<tr>
<td>+</td>
<td>FE-4147</td>
<td>Extend REST API to include starting and stopping a repository and to allow for re-indexing</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4398</td>
<td>Checkout errors can terminate svn repo import process</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4361</td>
<td>[CVS] Allow pollPeriod to be configurable</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4227</td>
<td>'Download archive' (aka tarball): name of zip is truncated to 20 characters</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4416</td>
<td>repository refresh status button is broken</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4404</td>
<td>DiffTextCache files not deleted</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4399</td>
<td>CVS scanner activates repository on every poll, causing high native memory usage</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4394</td>
<td>The repositories admin page is slow with many repositories enabled</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4392</td>
<td>Svnkit fine logging causes very large debug logs</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4391</td>
<td>Validation of start-rev in admin is incorrect</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4371</td>
<td>Fisheye not handling empty git or hg changesets properly</td>
<td>Closed</td>
</tr>
<tr>
<td>⬆</td>
<td>FE-4368</td>
<td>Changeset Comment</td>
<td>Closed</td>
</tr>
</tbody>
</table>
FE-4357 Improve the way permissions are handled, as a lot of time is spent in user/repo permission checks, which should be cached

FE-4354 OSGI cache should be excluded from backup

FE-4351 Indexing stalls if parsing Perforce multi print output throws an exception

FE-4350 FishEye not fully compatible with Mercurial 2.3

FE-4349 Disable directory Listing

FE-4332 "The Compressed (zipped) Folder is invalid or corrupted" error in IE8 when downloading tarball from "Download Archive" link

FE-4317 Provide a mechanism to enable Jetty JMX Beans

FE-4315 Restore does not set key permissions

FE-4207 Fisheye fails to start if the FISHEYE_INST does not contain a config.xml

FE-4206 Product update check doesn't support proxy authentication

FE-4195 Non-Crucible users are shown create review links

FE-4126 "Select/deselect all users" Checkbox Not Visible
FishEye 2.10 upgrade guide

Below are some important notes on upgrading to FishEye 2.10. For details of the new features and improvements in this release, please read the FishEye 2.10 release notes.

Upgrade notes

FishEye 2.10

- As of FishEye 2.10 the RepositoryAdminService API enforces permission checking. If you’re writing a plugin that uses that API make sure to use ImpersonationService.doPrivilegedAction to execute the calls in privileged mode.
- Note that as of FishEye 2.9 the JDBC driver for MySQL is no longer bundled with FishEye/Crucible – due to license restrictions the MySQL JDBC driver cannot be bundled with FishEye/Crucible. Please download the driver from http://dev.mysql.com/downloads/connector/j/ and install it to the FISHEYE_INST/lib directory. For more information see Migrating to MySQL.

On this page:
- Upgrade notes
- Upgrade procedure
- Checking for known issues and troubleshooting the FishEye upgrade

Related pages:
- FishEye 2.10 release notes

Upgrade procedure

⚠️ Before you begin
- Test your upgrades in your test environment before rolling into production.
- Back up your entire FishEye instance (see Backing up and restoring FishEye data), i.e.
  - If you are backing up your FishEye instance using the Admin interface, tick all of the ‘Include’ checkboxes (e.g. repository and application caches, plugins and their configuration data, SQL database, etc).
  - If you are backing up your FishEye instance using the command-line interface, do not use
If you are already running a version of FishEye, please follow the instructions in the general FishEye upgrade guide.

Checking for known issues and troubleshooting the FishEye upgrade

If something is not working correctly after you have completed the steps above to upgrade your FishEye installation, please check for known FishEye issues and try troubleshooting your upgrade as described below:

- **Check for known issues.** Sometimes Atlassian finds out about a problem with the latest version of FishEye after the software is released. In such cases we publish information about the known issues in the FishEye Knowledge Base. Please check for any known issues in the FishEye Knowledge Base and follow the instructions to apply any necessary patches if necessary.

- **Did you encounter a problem during the FishEye upgrade?** Please refer to the guide to troubleshooting upgrades in the FishEye 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.

**FishEye 2.9 release notes**

14 November 2012

Atlassian is proud to present FishEye 2.9, which provides the simplest and most powerful integration with the latest version of JIRA that FishEye has ever had.

- Visit our issue tracker to see the full list of improvements and bug fixes in FishEye and Crucible for this release.

- See the change log for FishEye 2.9.x minor releases.

- **Upgrading from a previous version of FishEye.** Upgrading FishEye should be fairly straightforward. We strongly recommend that you back up FishEye before upgrading. Please refer to the FishEye 2.9 Upgrade Guide for essential information about upgrading.

- **Known issues.** Please check the important technical advisories on the front page of the Knowledge Base for information about any known issues for this release.

- **JIRA 5.0 integration.** The features described below are supported by JIRA 5.0, or later, with the latest version of the JIRA FishEye plugin.

Highlights of this release:

- Simpler JIRA integration
- More JIRA data in FishEye
- Faster JIRA source tab
- Other announcements

Providing feedback:

*Please log your votes and issues. They help us decide what needs doing, and are much appreciated!*

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
1

Simpler JIRA integration

With FishEye 2.9, linking to JIRA is just like Plug & Play. You simply need to set up an Application Link between your JIRA server and your FishEye instance to get all the power of JIRA / FishEye integration:

- View in JIRA the list of changesets corresponding to a specific issue
- Navigate to the related changesets from your issue
- Get the JIRA data corresponding to issues mentioned in your commit messages and your source in FishEye

You no longer need to create and maintain multiple links between projects in JIRA and repositories in FishEye. Maintaining your JIRA / FishEye integration has never been simpler! More...

2

More JIRA data in FishEye

Before FishEye 2.9 you had to create links from FishEye to particular JIRA projects in order to see your issue data in FishEye. Without doing this, you just wouldn’t see data for projects that were not linked to FishEye. Now with the FishEye 2.9 release, you no longer have to create those project-specific links because all of your JIRA data is accessible in FishEye as soon as you link your instances together. Not only will you save time from not having to administer all the separate FishEye/JIRA links, but your FishEye instance is now also smarter and gets data from all the projects existing on your JIRA server.

Note that project links remain available in FishEye 2.9, but they now act as a restriction on the integration. If you set up project links from FishEye to JIRA, only issues from those particular projects will be linked in FishEye.
Faster JIRA source tab

Part of the effort of revamping the JIRA integration was to improve the performance of the issue source tab, especially for large instances with multiple repositories, each with many changesets. The chart below shows the performance gain in FishEye as a result of the effort to make it scale well as your data grows.
Other announcements

- **Remote API setting always on**
  We removed the Remote API setting from the Server Settings page. From this release onwards the Remote APIs will always be accessible, to make the JIRA integration straightforward.

- **JIRA FishEye Plugin option "Disable for unmapped JIRA Projects" has been removed**
  As of this release, it no longer makes sense to disable the Source and Reviews tabs when Project Link are not configured. If you still wish to disable the Source and Reviews tabs for a specific project, for example for one that has no relation to source code, you should use project-level permissions. See How do I disable the FishEye tab panel for non-code projects?

### The FishEye 2.9 team

**Development**

**Core team**
- Brendan Humphreys
- Conor MacNeill
- Geoff Crain
- Lukasz Pater
- Maciej Swinarski
- Piotr wity wicicki
- Richard Stephens
- Tom Davies
- Valery Trubnikov

**Team lead**
- Nick Pellow

**Product management**
- Sten Pittet

**Project manager**
- Anton Mazkovoi

**Support**
- Ajay Sridhar
- Daniel Rohan
- Douglas Fabretti
- Felipe Kraemer
- Gurleen Anand
- Kah Loun Foong
- Leonardo Macedo
- Malik Mangier
- Patrick Hill
FishEye 2.9 changelog

This page will contain information about the FishEye 2.9 minor releases as these become available. Crucible license holders should also check the Crucible 2.9 changelog. These releases will be free to all customers with active FishEye software maintenance.

Don’t have FishEye 2.9 yet?

Take a look at all the features in the FishEye 2.9 release notes and see what you are missing out on!

Upgrading from a previous version of FishEye

If you are upgrading, please read the FishEye 2.9 upgrade guide.

On this page:
- From 2.9.1 to 2.9.2
- From 2.9.0 to 2.9.1
- FishEye 2.9.0

From 2.9.1 to 2.9.2

11 December 2012

This is a bug fix release.

JIRA Issues (4 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>📋</td>
<td>FE-4392</td>
<td>Svnkit fine logging causes very large debug logs</td>
</tr>
<tr>
<td>📋</td>
<td>FE-4391</td>
<td>Validation of start-rev in admin is incorrect</td>
</tr>
<tr>
<td>📋</td>
<td>FE-4368</td>
<td>Changeset Comment re-indexing</td>
</tr>
</tbody>
</table>
From 2.9.0 to 2.9.1

19 November 2012

This is a bug fix release.

JIRA Issues (1 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅</td>
<td>FE-4364</td>
<td>REST calls to /rest-service/repositories-v1/ and /rest-service-fe/repositories-v1/ fail when there are ClearCase repositories configured</td>
</tr>
</tbody>
</table>

FishEye 2.9.0

14 November 2012

JIRA Issues (17 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-4288</td>
<td>Profile viewing source files in FECRU with autolink enabled</td>
</tr>
<tr>
<td></td>
<td>FE-4194</td>
<td>Enable automatic repository-&gt;Jira project entity link creation</td>
</tr>
<tr>
<td></td>
<td>FE-4151</td>
<td>Every FeCru repo sends a REST call to Crowd with a getUserInfo function when the source tab is clicked in JIRA</td>
</tr>
<tr>
<td></td>
<td>FE-4150</td>
<td>Implement a REST endpoint that can return the results for all the repos at once. This will be used by the JIRA FishEye Plugin.</td>
</tr>
<tr>
<td></td>
<td>FE-2907</td>
<td>Branch selector should trim whitespace</td>
</tr>
<tr>
<td></td>
<td>FE-2639</td>
<td>Reduce memory usage of mysql backup process</td>
</tr>
<tr>
<td></td>
<td>FE-2476</td>
<td>Avoid spurious package scanner warnings in the debug log</td>
</tr>
</tbody>
</table>
FishEye 2.9 upgrade guide

Below are some important notes on upgrading to FishEye 2.9. For details of the new features and improvements in this release, please read the FishEye 2.9 release notes.

Upgrade notes

FishEye 2.9

- The JDBC driver for MySQL is no longer bundled with FishEye/Crucible – Due to license restrictions the MySQL JDBC driver cannot be bundled with FishEye/Crucible. Please download the driver from http://dev.mysql.com/downloads/connector/j/ and install it to the FISHEYE_INST/lib directory. For more information see Migrating to MySQL.
- The Connection pool manager has changed — FishEye and Crucible now use the BoneCP connection pool manager instead of the third-party c3p0 library previously used in FishEye/Crucible 2.8 and earlier.
versions. The BoneCP connection pool manager has been preconfigured in FishEye and Crucible to work out-of-the-box for most customers. Read Configuring the database connection pool for a description of the default settings and for instructions on how to override them.

- **The 'Suggest Reviews' function has been removed from Crucible** — From this release onwards, a Crucible user will no longer be prompted to add to an existing review when creating a review from a changeset. We have removed this functionality based on user feedback that it was confusing and not being used. This allows users who want to add changesets to an existing review to simply navigate to that review and use the 'Add Content' option. Users can also add commits to an existing review via commit messages. Removing ‘Suggest Reviews’ also improves performance when creating reviews, as Crucible does not have to search for and suggest reviews.

- **Remote API is now permanently enabled** — From this release of FishEye and Crucible onwards, the 'Remote API' setting in 'Server Settings' will not be configurable. The remote API is now always turned on, to make integration with JIRA and other services more straightforward.

---

### Upgrade notes

#### Upgrade procedure

**Before you begin**

- Test your upgrades in your test environment before rolling into production.
- Back up your *entire* FishEye instance (see Backing up and restoring FishEye data), i.e.
  - If you are backing up your FishEye instance using the Admin interface, tick all of the 'Include' checkboxes (e.g. repository and application caches, plugins and their configuration data, SQL database, etc).
  - If you are backing up your FishEye instance using the command-line interface, do not use any exclusion options.

If you are already running a version of FishEye, please follow the instructions in the general FishEye upgrade guide.

#### Checking for known issues and troubleshooting the FishEye upgrade

If something is not working correctly after you have completed the steps above to upgrade your FishEye installation, please check for known FishEye issues and try troubleshooting your upgrade as described below:

- **Check for known issues.** Sometimes Atlassian finds out about a problem with the latest version of FishEye after the software is released. In such cases we publish information about the known issues in the FishEye Knowledge Base. Please check the FishEye 2.8 known issues in the FishEye Knowledge Base and follow the instructions to apply any necessary patches if necessary.

- **Did you encounter a problem during the FishEye upgrade?** Please refer to the guide to troubleshooting upgrades in the FishEye 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.

**FishEye 2.8 release notes**

15th August 2012

Atlassian is proud to present FishEye 2.8, which provides a variety of social features as well as performance improvements.

See the change log for FishEye 2.8.x minor releases.

**Highlights of this release:**

- Mentions
- Shares
- Improved performance of the activity stream
- Support for Subversion 1.7
- End of life announcements

**Providing feedback:**

*Please log your votes and issues. They help us decide what needs doing, and are much appreciated!*

In FishEye 2.8 you can notify other users in a changeset discussion, review comment or snippet comment by using mentions. To mention someone, simply type `@` and then the person's name (not their username) and choose from the suggestions that FishEye offers. FishEye sends a notification to that person, so they know that you have mentioned them.
Shares

Share a changeset, a review, or a source repository with other members of your team, quickly and easily from where you are working. Just click the **Share** button at the top, enter their name, username or email address and add a cheery comment. FishEye sends an email.

![Share button in FishEye](image)

**3**

Improved performance of the activity stream

The activity streams throughout FishEye (such as the commits and review activity, and on the dashboard) have an improved user experience due to faster speed and infinite scroll (that replaces paging).

**4**

Support for Subversion 1.7

FishEye now supports Subversion 1.7 (although not with a native SVN client).
End of life announcements

- **ClearCase**
  As previously announced, IBM ClearCase is no longer supported in Fisheye 2.8.

- **Internally managed repositories**
  On August 13th 2013 we are ending support for internally managed repositories. Read more about this.

- **JIRA activity stream**
  We've removed JIRA information from the activity stream in order to simplify the user experience and improve the performance of FishEye. Please note that all other JIRA integration features still remain in FishEye.

The FishEye 2.8 team

Development

Core team
Geoff Crain
Tom Davies
Brendan Humphreys
Conor MacNeill
Richard Stephens

Team lead
Nick Pellow

Product management
Sten Pittet

Project manager
Anton Mazkovoi

Support
Ajay Sridhar
Armen Khachatryan
Daniel Rohan
Douglas Fabretti
Felipe Kraemer
Gurleen Anand
Renan Battaglin
Rene Verschoor
Zed Yap
Patrick Hill
**Others**

*Product marketing*
- Giancarlo Lionetti
- Jeff Park

*Quality assurance*
- George Filippoff
- Mark Hrynczak

*Technical writing*
- Paul Watson

*Operations*
- James Fleming

**FishEye 2.8 changelog**

This page will contain information about the FishEye 2.8 minor releases as these become available. Crucible license holders should also check the [Crucible 2.8 changelog](#). These releases will be free to all customers with [active FishEye software maintenance](#).

**Don’t have FishEye 2.8 yet?**

Take a look at all the features in the [FishEye 2.8 release notes](#) and see what you are missing out on!

[Download](#)

**Upgrading from a previous version of FishEye**

If you are upgrading, please read the [FishEye 2.8 upgrade guide](#).

---

**On this page:**
- FishEye 2.8.0

---

**From 2.8.1 to 2.8.2**

**5 October 2012**

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (6 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue</td>
<td>Summary</td>
<td>Assigned To</td>
<td>Status</td>
<td>Created</td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
preserved for synced users with mixed-case usernames

| FE-312 | Fisheye is indexing change set ids of Git submodules when it shouldn't be causing Brenda Humphreys [Atlassian] None | Fixed | Nov 02, 2010 | Nov 06, 2012 |

**From 2.8.0 to 2.8.1**

**29 August 2012**

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (5 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
FishEye 3.0

FE-449

Mention File Size Limit Settings Apply Only to Subversion Repositories

Author: Paul Watson [Atlassian]

Assignee: Ricardo Martins [Atlassian]

Status: Fixed

Created: Feb 13, 2013

Closed: Mar 22, 2013

FE-423

Update database encoding warning message to be more generic

Author: Gurleen Anand [Atlassian]

Assignee: Unassigned

Status: Fixed

Created: Aug 20, 2012

Closed: Aug 28, 2012

FE-425

Email Review option won't work

Author: Leonardo De Macedo [Atlassian]

Assignee: Nick Pellow [Atlassian]

Status: Fixed

Created: Aug 24, 2012

Closed: Oct 01, 2012

FishEye 2.8.0

15 August 2012

JIRA Issues (28 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Create Date</th>
<th>Updated Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket</td>
<td>Description</td>
<td>Assigned To</td>
<td>Resolution</td>
<td>Created</td>
<td>Closed</td>
<td>Details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------</td>
<td>--------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-4218</td>
<td>Always have resource monitor start with the FECRU server to log initial data</td>
<td>Daniel Rohan [Atlassian]</td>
<td>Fixed</td>
<td>Aug 07, 2012</td>
<td>Feb 11, 2013</td>
<td>Changed in admin UI due to radio button issues/changes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-4017</td>
<td>Add authenticated</td>
<td>Brenda Humph Issa</td>
<td>Fixed</td>
<td>Mar 26, 2012</td>
<td>Apr 05, 2012</td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Title</td>
<td>Assigned To</td>
<td>Resolution</td>
<td>Created</td>
<td>Updated</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------------------------</td>
<td>------------------------------------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-4090</td>
<td>Poor Dashboard page performance for logged in users</td>
<td>Unassigned</td>
<td>Fixed</td>
<td>May 08, 2012</td>
<td>Jun 12, 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------</td>
<td>---------------------------</td>
<td>------------------------</td>
<td>--------------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Branch Name</td>
<td>Description</td>
<td>Assignee</td>
<td>Status</td>
<td>Resolution</td>
<td>Created</td>
<td>Fixed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>-------------</td>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>---------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Implicit committter mappings are potentially expensively computed multiple times on every request.

Trying to browse a file name containing a + generates an error.

Overlapping Label Text in FishEye UI Using Internet Explorer (IE9)

hg indexing falls over on commits with very large commit message.

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Status</th>
<th>Fixed</th>
<th>Created</th>
<th>Resolved</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE-404</td>
<td>Allow specific location of which mime-types are to be treated as text</td>
<td>Unassigned</td>
<td>Conor MacNeill [Atlassian]</td>
<td>Closed</td>
<td>Fixed</td>
<td>Apr 11, 2012</td>
<td>May 14, 2012</td>
</tr>
<tr>
<td>FE-400</td>
<td>Consider removing Crowd Auth and LDAP Auth from dropdown when creating users in FishEye (was: Authentication Sync (or Re-Sync)) Deletes Users</td>
<td>Geoff Crain [Atlassian]</td>
<td>Daniel Rohan [Atlassian]</td>
<td>Closed</td>
<td>Fixed</td>
<td>Mar 16, 2012</td>
<td>May 03, 2012</td>
</tr>
</tbody>
</table>
FishEye 2.8 upgrade guide

Below are some important notes on upgrading to FishEye 2.8. For details of the new features and improvements in this release, please read the FishEye 2.8 release notes.

Upgrade notes

FishEye 2.8

Please be sure to read the End of Life Announcements for FishEye/Crucible 2.8. Most notably: Support for Clearcase repository types has been dropped.

On this page:

- Upgrade notes
- Upgrade procedure
- Checking for known issues and troubleshooting the FishEye upgrade

Related pages:

- FishEye 2.8 release notes

Before you begin

- Test your upgrades in your test environment before rolling into production.
- Back up your entire FishEye instance (see Backing up and restoring FishEye data), i.e.
  - If you are backing up your FishEye instance using the Admin interface, tick all of the 'Include' checkboxes (e.g. repository and application caches, plugins and their configuration data, SQL database, etc).
  - If you are backing up your FishEye instance using the command-line interface, do not use any exclusion options.
If you are already running a version of FishEye, please follow the instructions in the general FishEye upgrade guide.

Checking for known issues and troubleshooting the FishEye upgrade

If something is not working correctly after you have completed the steps above to upgrade your FishEye installation, please check for known FishEye issues and try troubleshooting your upgrade as described below:

- **Check for known issues.** Sometimes Atlassian finds out about a problem with the latest version of FishEye after the software is released. In such cases we publish information about the known issues in the FishEye Knowledge Base. Please check for FishEye 2.8 known issues in our JIRA issue tracker and follow the instructions to apply any necessary patches if necessary.

- **Did you encounter a problem during the FishEye upgrade?** Please refer to the guide to troubleshooting upgrades in the FishEye 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 2.7 with existing ClearCase Repositories**

*ClearCase Repositories*

FishEye 2.8 no longer supports ClearCase repositories. Please refer to the End of Support Announcement for IBM ClearCase.

If you have any ClearCase repositories present in your FishEye instance, FishEye 2.8 will log a warning and will not start. The intention behind this rather draconian behaviour is to give you the opportunity to backup and update your instance before FishEye performs the upgrade operations to convert your indexes and database into a 2.8 compatible format.

You can force FishEye to start with clearcase repositories disabled by setting the fisheye.clearcase.disable system property to true. Please see the documentation on setting up fisheye environment variables for more information.

For example, you can force startup by setting your FISHEYE_OPTS environment variable to

-Dfisheye.clearcase.disable=true

ie.:

```
$ export FISHEYE_OPTS="${FISHEYE_OPTS} -Dfisheye.clearcase.disable=true"
```

On Windows Server 2008 or Windows 7, you can configure this as follows:

1. Open the Start menu, right-click on "Computer" and select Properties
2. Click on "Advanced System Settings" on the left of the dialog
3. Click the "Environment Variables" button at the bottom of the Advanced tab
4. Add a FISHEYE_OPTS environment variable to the "System variables" section
FishEye 2.8 EAP Release Notes

FishEye & Crucible 2.8.0-m1

The 2.8.0-m1 EAP release is being provided to give early access to Subversion 1.7 support. The build contains the same features as FishEye/Crucible 2.7.14 with the updated version of svnkit necessary to work with Subversion 1.7.

We have chosen to provide this as an EAP release because we felt it was not appropriate to ship an update to a library so fundamental to FishEye's operation in a 2.7 point release.

Although this is an EAP release, the only difference between this build and the 2.7.14 release is the updated svnkit libraries. We believe, therefore, that this version can safely be deployed to production environments. As always, please take suitable backups before any such deployments. Please see the FishEye 2.7 Upgrade Guide for more information.

FishEye 2.7 Release Notes

7 September 2011

With great pleasure, Atlassian presents FishEye 2.7 featuring Git repository management, Smart Commits and Web Hooks.

Highlights of this Release:

- Managed Git Repositories
- Smart Commits
- Web Hooks
- JIRA FishEye Plugin Improvements
- Small Improvements

![Download latest version](download.png)
Thank you for all your issues and votes. Keep logging issues to help us keep improving!
Read the release notices for important information about this release.

Highlights of FishEye 2.7

1

Managed Git Repositories

Administrators can create and manage Git repositories directly from FishEye on their own servers. FishEye Git repository management gives you the flexibility to do the following:

- Create forks and clones of existing projects.
- Manage permissions easily with push and pull access per repository.
- Visualize the hierarchy of repository forks.
- Explore and visualize changes with the commit graph

More...

2

Smart Commits

Smart Commits allow repository committers to perform actions like transitioning JIRA issues or creating Crucit code reviews by embedding specific commands into their commit messages.

FishEye 2.7 comes bundled with support for creating and updating Crucible reviews, as well as transitioning,
commenting, and logging work to JIRA issues.

Placed into a commit message, the example below would close the JIRA issue BUG-123.

![Commit message example](image)

The following example would create and start a review for the current changeset in the project CR-TEST, as well as add the user jdoe as a reviewer:

![Review creation example](image)

For more details on using Smart Commits, please see the User's guide to Smart Commits.

JIRA issue transitioning requires at least version 3.4.5 of the JIRA FishEye plugin. It can be downloaded via the Atlassian plugin manager from within JIRA or manually downloaded from the Atlassian plugin exchange.

Smart Commits are extensible via plugins, allowing you to simply implement your own handlers to integrate with services used in your workplace. Please see the Smart Commit Tutorial for developer resources on creating a Smart Commit plugin.

By default, smart commits are disabled if the FishEye instance has any Mercurial or Git repositories. Please see the Smart Commits administrator's guide for more details.

More...
Web Hooks

FishEye has added a Web Hooks capability. A Web Hook is a form of event-notification via HTTP POST.

In the case of FishEye Web Hooks, developers can add URLs which will receive commit(s) data through POST requests. This can be used to easily create useful gadgets in various programming languages that will give push notification of commit changes.

More...

JIRA FishEye Plugin Improvements

The FishEye team are also proud to update the JIRA FishEye plugin to version 3.4.5. Upgrades to the plugin are available immediately via the Atlassian plugin manager from within JIRA or the Atlassian plugin exchange.

This update provides significant performance improvements, most notably asynchronous loading of the "Source" and "Reviews" tab in JIRA which allows the JIRA view issue page to load without waiting for the FishEye server to respond with data.

More...

Small Improvements

FishEye 2.7 also comes bundled with numerous other bug fixes and improvements, including:

- Syntax highlighting for Java 7, Groovy, Velocity and Scala.
- FishEye can now run on Java 7
- Improved user interface for the administration screens
- HEAD label in revisions page only appears for currently selected branch (or default branch if All is selected).
- Improved plugin points for developers

Visit our issue tracker to see the full list of improvements and bug fixes in FishEye and Crucible for this release.

Release Notices

- **Upgrading from a previous version of FishEye.** Upgrading FishEye should be fairly straightforward. We strongly recommend that you back up FishEye before upgrading. Please refer to the FishEye 2.7 Upgrade Guide for further essential information about your upgrade.
- **Known Issues.** Please check the important technical advisories on the front page of the Knowledge Base for information about any known issues for this release.

FishEye 2.7 Changelog

This page contains information about the FishEye 2.7 minor releases. Crucible license holders should also check the Crucible 2.7 Changelog. See the FishEye 2.7 Release Notes for details of what's new in 2.7.0.

⚠️ Please read the FishEye 2.7 Upgrade Guide before upgrading to any of the minor releases below.

On this page:

- From 2.7.13 to 2.7.14
  - JIRA Issues (5 issues)
    - From 2.7.12 to 2.7.13
    - JIRA Issues (27 issues)
    - From 2.7.11 to 2.7.12
    - JIRA Issues (7 issues)
    - From 2.7.10 to 2.7.11
    - JIRA Issues (19 issues)
    - From 2.7.9 to 2.7.10
    - JIRA Issues (3 issues)
    - From 2.7.8 to 2.7.9
    - JIRA Issues (11 issues)
    - From 2.7.7 to 2.7.8
    - JIRA Issues (7 issues)
    - From 2.7.6 to 2.7.7
    - JIRA Issues (7 issues)
    - From 2.7.5 to 2.7.6
    - JIRA Issues (14 issues)
    - From 2.7.4 to 2.7.5
    - JIRA Issues (15 issues)
    - From 2.7.3 to 2.7.4
    - JIRA Issues (9 issues)
    - From 2.7.2 to 2.7.3
    - JIRA Issues (2 issues)
    - From 2.7.1 to 2.7.2
JIRA Issues (6 issues)

- From 2.7.0 to 2.7.1

JIRA Issues (5 issues)

From 2.7.13 to 2.7.14

12 June 2012

This is a bug fix release. The complete list of issues is below.

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-410</td>
<td>FishEye 2.7.13 startup error when FISHE YE_IN ST have spaces</td>
<td>Nick Pellow [Atlassian]</td>
<td>Leo Leung</td>
<td>↑</td>
<td>Clos ed</td>
<td>Fixed</td>
<td>May 24, 2012</td>
<td>Jun 08, 2012</td>
</tr>
</tbody>
</table>
From 2.7.12 to 2.7.13

21 May 2012

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (27 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌐</td>
<td>FE-396</td>
<td>Emulate pre-2.7 behaviour: when an user is created by the user synchronisation, restore any deleted user with the same username</td>
<td>Brenda Humphreys [Atlassian]</td>
<td>Pierre-Etienne Poirot [Atlassian]</td>
<td>↑</td>
<td>Closed</td>
<td>Fixed</td>
<td>Feb 16, 2012</td>
<td>Aug 20, 2012</td>
</tr>
</tbody>
</table>
9 rver/max-threa
x-threads
increases from 20 to 150

FE-401 8 EyeQL Search Results not displaying author fields

FE-401 7 Add authenticated user in http response header for logging ability in reverse proxy

FE-398 9 Allow disabling of precise content Hash to csid mapping

FE-392 2 Add custom syntax highlighting for shell scripts

FE-349 5 com.atlassian.crowd.
<table>
<thead>
<tr>
<th>ID</th>
<th>Status</th>
<th>Description</th>
<th>Reporter</th>
<th>Assignee</th>
<th>Fixed Date</th>
<th>Created Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE-408</td>
<td>Closed</td>
<td>OperationFailedException if SSO is enabled with JIRA user management</td>
<td>Leo Macedo [Atlassian]</td>
<td>Leo Macedo [Atlassian]</td>
<td>May 01, 2012</td>
<td>Jun 27, 2012</td>
</tr>
<tr>
<td>FE-406</td>
<td>Closed</td>
<td>CC Indexing can fetch version 0 and index it even if</td>
<td>Conor MacNeill [Atlassian]</td>
<td>Conor MacNeill [Atlassian]</td>
<td>Apr 17, 2012</td>
<td>Apr 18, 2012</td>
</tr>
</tbody>
</table>
before the start date.

### FE-404
- **4**
  - **Lack of user names element in security config can stop AJPAuth working**
  - Created by [Atlassian](https://www.atlassian.com)
  - Fixed Date: Apr 11, 2012 to Apr 15, 2012

### FE-404
- **3**
  - **Handle NPE caused by change set without a position value in Commit Graph**
  - Created by [Atlassian](https://www.atlassian.com)
  - Fixed Date: Apr 10, 2012 to Apr 15, 2012

### FE-403
- **6**
  - **P4 GetLatestRevision will fail for paths with no revisions - leads to excessive logging**
  - Created by [Atlassian](https://www.atlassian.com)
  - Fixed Date: Apr 05, 2012 to Apr 10, 2012

### FE-403
- **4**
  - **Copy from a file which existed**
  - Created by [Atlassian](https://www.atlassian.com)
  - Fixed Date: Apr 03, 2012 to Apr 10, 2012
FE-400 3 JIRA Application Password Field Not Refreshing with Apply Button

Brendan Humphreys [Atlassian]  Daniel Rohan [Atlassian]

Clos ed Fixed Mar 15, 2012 Apr 04, 2012

FE-398 6 An svn copy from an unindexed part of a repo to an indexed part causes an error: com.ce nqua.fisheye.rep.DbException: Attempt to load isDeleted from nonexistent revid = -1

Tom Davies [Atlassian]  Tom Davies [Atlassian]


FE-398 5 Display the SSID above the license textarea

Richard Stephens [Atlassian]  Stefan Sorin Nicolin


FE-398 4 JIRA user

Brendan Sridhar

Clos ed Fixed Mar 06, 2012 Feb 19, 2013

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
management fails if there are no groups selected to sync.

**FE-397 9**

Missing label for "Allow remote API calls" option in FishEye UI.

- **Summary**: Missing label for "Allow remote API calls" option in FishEye UI.
- **Assignee**: Richard Stephens [Atlassian]
- **Reported By**: Sarah Maddox [Atlassian]
- **Status**: Fixed
- **Closed Date**: Feb 27, 2012
- **Resolved Date**: Apr 23, 2012

**FE-389 9**

fisheye doesn't handle p4 branch specs in ways that p4 supports.

- **Summary**: fisheye doesn't handle p4 branch specs in ways that p4 supports.
- **Assignee**: Geoff Crain [Atlassian]
- **Reported By**: Geoff Crain [Atlassian]
- **Status**: Fixed
- **Closed Date**: Jan 18, 2012
- **Resolved Date**: May 01, 2012

**FE-387 0**

Removed license gets replaced by "Disabled", preventing Update.

- **Summary**: Removed license gets replaced by "Disabled", preventing Update.
- **Assignee**: Conor MacNeill [Atlassian]
- **Reported By**: Ajay Sridhar [Atlassian]
- **Status**: Fixed
- **Closed Date**: Dec 29, 2011
- **Resolved Date**: Sep 07, 2012

**FE-373 2**

Switching from tag to trunk results in a

- **Summary**: Switching from tag to trunk results in a
- **Assignee**: Conor MacNeill [Atlassian]
- **Reported By**: Rene Verschoor [Atlassian]
- **Status**: Fixed
- **Closed Date**: Sep 21, 2011
- **Resolved Date**: Apr 18, 2012
From 2.7.11 to 2.7.12

16 April 2012

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (7 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
From 2.7.10 to 2.7.11

27 February 2012

This is a bug fix release. The complete list of issues is below.

Error rendering macro 'jiraissues': Unable to determine if sort should be enabled.

From 2.7.9 to 2.7.10

20 January 2012

This is a bug fix release. The complete list of issues is below.

<table>
<thead>
<tr>
<th>JIRA Issues (3 issues)</th>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
stream in JIRA
Activity Stream gadget on dashboard when have Repository with no description

<table>
<thead>
<tr>
<th>FE-390</th>
<th>Issue Transition controls don't appear when closing a review</th>
<th>Unassigned</th>
<th>None</th>
<th>Fixed</th>
<th>Jan 19, 2012</th>
<th>Jan 19, 2012</th>
</tr>
</thead>
</table>


**From 2.7.8 to 2.7.9**

**19 January 2012**

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (11 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
FishEye wrongly removes admin privileges for a Crowd group when Crowd is not up when FishEye is starting up.

- [FE-389](#) 6  
  **Webwork 2 vulnerability**
  - Nick Pellow [Atlassian]
  - Alex Wei [Atlassian]
  - Vitaly Osipov [Atlassian]
  - Paul Watson [Atlassian]
  - Conor MacNeill [Atlassian]

- [FE-388](#) 8  
  **FishEye does not correctly handle moved tags**
  - Conor MacNeill [Atlassian]
  - Conor MacNeill [Atlassian]

- [FE-388](#) 4  
  **Default UserManager. getUsersInGroup incorrectly acquires a write lock instead of a read lock**
  - Brenda Humphreys [Atlassian]
  - Brenda Humphreys [Atlassian]

- [FE-388](#) 0  
  **P4 client**
  - Brenda Humphreys [Atlassian]
  - Brenda Humphreys [Atlassian]

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
### FE-387

**8** When processing performance file revisions, only branch specs that have been added or modified since the last restart are considered.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Assigned To</th>
<th>Status</th>
<th>Fixed By</th>
<th>Fixed</th>
<th>Created</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE-387</td>
<td>When processing performance file revisions, only branch specs that have been added or modified since the last restart are considered.</td>
<td>Anna Buttfield [Atlassian]</td>
<td>Fixed</td>
<td>Anna Buttfield [Atlassian]</td>
<td>Jan 08, 2012</td>
<td>Jan 09, 2012</td>
<td></td>
</tr>
</tbody>
</table>

### FE-387

**5** SVN property changes with content lines containing certain strings break diff parsing.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Assigned To</th>
<th>Status</th>
<th>Fixed By</th>
<th>Fixed</th>
<th>Created</th>
<th>Closed</th>
</tr>
</thead>
</table>

### FE-386

**2** JS errors in ie8.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Assigned To</th>
<th>Status</th>
<th>Fixed By</th>
<th>Fixed</th>
<th>Created</th>
<th>Closed</th>
</tr>
</thead>
</table>

### FE-386

**1** lock mismatches.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Assigned To</th>
<th>Status</th>
<th>Fixed By</th>
<th>Fixed</th>
<th>Created</th>
<th>Closed</th>
</tr>
</thead>
</table>
From 2.7.7 to 2.7.8

30 November 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (7 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-384</td>
<td>crowd admin groups get removed on startup</td>
<td>Geoff Crain</td>
<td>Geoff Crain</td>
<td>↓</td>
<td>Clos ed</td>
<td>Fixed</td>
<td>Nov 27, 2011</td>
<td>Nov 29, 2011</td>
</tr>
<tr>
<td></td>
<td>FE-381</td>
<td>Clicking on repository in FishEye Administration generates error dialog in IE8</td>
<td>Anna Buttfeld</td>
<td>Felipe Kraemer</td>
<td>↓</td>
<td>Clos ed</td>
<td>Fixed</td>
<td>Nov 04, 2011</td>
<td>Jan 17, 2012</td>
</tr>
<tr>
<td>Issue</td>
<td>Title</td>
<td>Description</td>
<td>Assigned</td>
<td>Resolution</td>
<td>Created</td>
<td>Updated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------------</td>
<td>----------</td>
<td>------------</td>
<td>---------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-374</td>
<td>Changing Block Size triggers a reindex + restart, while only a restart is needed</td>
<td>Pierre-Etienne Poirot [Atlassian]</td>
<td>Rene Verschoor [Atlassian]</td>
<td>Closed</td>
<td>Sep 26, 2011</td>
<td>Nov 24, 2011</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From 2.7.6 to 2.7.7

21 November 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (7 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-383</td>
<td>In SVN, if a tag is created and the identified parent change set id is</td>
<td>Anna Buttfeld [Atlassian]</td>
<td>Anna Buttfeld</td>
<td></td>
<td>Clos ed</td>
<td>Fixed</td>
<td>Nov 16,</td>
<td>Nov 17,</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>not indexed by FishEye the tag won't show up in the branch</td>
<td>[Atlassian]</td>
<td>[Atlassian]</td>
<td></td>
<td></td>
<td></td>
<td>2011</td>
<td>2011</td>
</tr>
<tr>
<td>Issue</td>
<td>Title</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-383 3</td>
<td>Performance jobids are not indexed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-383 2</td>
<td>Links in RSS feed do not use absolute URLs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-382 6</td>
<td>Jetty log messages are not reported anywhere</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-382 2</td>
<td>mime type mapping for file names rather than suffixes removes first character of candidate file name before matching with key filenames</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Owner</th>
<th>Assigned To</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
When navigating forward in time in activity stream, items are shown earliest to latest down the page.

Browse /diff menu inactive looks active and vice-versa.

From 2.7.5 to 2.7.6

8 November 2011

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (14 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-381</td>
<td>Shutdown the thread pool once the background upgrade tasks have been completed</td>
<td>Pierre-Etienne Poirot [Atlassian]</td>
<td>Pierre-Etienne Poirot [Atlassian]</td>
<td>↓</td>
<td>Clos ed</td>
<td>Fixed</td>
<td>Nov 03, 2011</td>
<td>Nov 03, 2011</td>
</tr>
<tr>
<td>FE-379</td>
<td>Chang e authent ication provide rs to return Comm unicatio n Except io n when appropri ate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pierre-Etienne Poirot [Atlassian]</td>
<td>Pierre-Etienne Poirot [Atlassian]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Closed</td>
<td>Fixed</td>
<td>Oct 20, 2011</td>
<td>Nov 01, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FE-377</th>
<th>Add option to enable client authent ication SSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pierre-Etienne Poirot [Atlassian]</td>
<td>Pierre-Etienne Poirot [Atlassian]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FE-361</th>
<th>Improv e the way avatars are loaded to avoid une nece ssary calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna Buttfield [Atlassian]</td>
<td>Renan Battaglini [Atlassian]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FE-382</th>
<th>If the total length of the branch names selecte d for the commit graph is too long, the spinner never</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unassigned</td>
<td>Tom Davies [Atlassian]</td>
</tr>
<tr>
<td>Closed</td>
<td>Fixed</td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
do not deactivate users if the error is a communication problem

<table>
<thead>
<tr>
<th>Issue</th>
<th>Title</th>
<th>Description</th>
<th>Poster</th>
<th>State</th>
<th>Fixed</th>
<th>Date</th>
</tr>
</thead>
</table>
## From 2.7.4 to 2.7.5

### 21 October 2011

This is a bug fix release. The complete list of issues is below.

### JIRA Issues (15 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
ancestry suppression flag to default

### FE-376

**User** Michael Heemskerk
**URL** [Atlassian](https://confluence.atlassian.com/display/FE-376)
**Status** Fixed
**Created** Oct 06, 2011
**Closed** Oct 11, 2011
**Summary** FishEye logs lots of ApplicationPermissionException when synchronising Crowd users that have no permission on FishEye / Crucible

Pierre-Etienne Poirot
[Atlassian](https://confluence.atlassian.com/display/FE-376)

### FE-378

**User** Pierre-Etienne Poirot
**URL** [Atlassian](https://confluence.atlassian.com/display/FE-378)
**Status** Fixed
**Created** Oct 19, 2011
**Closed** Oct 19, 2011
**Summary** Git does not report an error when creating a managed repository over an existing one

Pierre-Etienne Poirot
[Atlassian](https://confluence.atlassian.com/display/FE-378)

### FE-377

**User** Pierre-Etienne Poirot
**URL** [Atlassian](https://confluence.atlassian.com/display/FE-377)
**Status** Fixed
**Created** Oct 11, 2011
**Closed** Oct 12, 2011
**Summary** Fix documentation of the
Search results page fails to render for EyeQL queries > 4000 characters on Oracle.

Paths under tag directories are displayed as deleted / empty directories (greyed out)

Repository path, includes, excludes and hidden directories should be copied to newly created forks
d with "application/xml" as the content type

| FE-369 | When pushing to a Git managed repo while another client is cloning, the client which is cloned freezes | Pierre-Etienne Poirot [Atlassian] | Pierre-Etienne Poirot [Atlassian] | ↑ | Clos ed | Fixed | Sep 06, 2011 | Oct 09, 2011 |

From 2.7.3 to 2.7.4

20 September 2011

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (9 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
Exception while loading user preferences from database, causing login failures.

FE-374 7 If you edit and then save a managed repository in Internet Explorer, the text "fork of null" will appear and the forks view will no longer be available.

FE-374 2 After a user has been deleted and then added again, subseq
FE-374 Manual Auth Re-Sync button doesn't do anything
Tom Davies [Atlassian] Phil Davies
Fixed Sep 23, 2011 Sep 27, 2011

FE-373 Log files not included in support zip
Matthew Watson Watson [Atlassian]
Closed Sep 23, 2011 Jan 16, 2012

FE-373 Commit events fired on Git repos when branches are added/removed
Seb Ruiz [Atlassian] Matt Ryall [Atlassian]
Closed Sep 22, 2011 Sep 27, 2011

FE-373 Error upon exceeding license limit with Crowd user management
Tom Davies [Atlassian] Gurleen Anand [Atlassian]
Closed Sep 21, 2011 Sep 27, 2011

FE-371 Hg authentication fails with special character in
Closed Sep 13, 2011 Jan 16, 2012
From 2.7.2 to 2.7.3

20 September 2011

This is a bug fix release. The complete list of issues is below.

### JIRA Issues (2 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Create Date</th>
<th>Update Date</th>
</tr>
</thead>
</table>

From 2.7.1 to 2.7.2

19 September 2011

This is a bug fix release. The complete list of issues is below.

### JIRA Issues (6 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Create Date</th>
<th>Update Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt=" " /></td>
<td>FE-361 6</td>
<td>Change</td>
<td>Conor MacNei</td>
<td>Felipe Cuozzo</td>
<td>👻</td>
<td>🍃 Closed</td>
<td>Fixed</td>
<td>Aug 02, 2011</td>
<td>Jan 16, 2012</td>
</tr>
</tbody>
</table>

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
FE-3718 Eclipse Git integration (egit) fails to clone from managed Git repository
Unassigned Tim Pettersen [Atlassian]

FE-3711 Some Directory links in Quick Search have double encoded URLs
Jason Hinch [Atlassian] Jason Hinch [Atlassian]

FE-3696 Web Hooks should be deleted when deleting a repository
Tom Davies [Atlassian] Seb Ruiz [Atlassian]

FE-3693 Pagination numbers at bottom
Jonathan Poh [Atlassian] Rene Verschoor [Atlassian]
From 2.7.0 to 2.7.1

9 September 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (5 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
**FishEye 2.7 Upgrade Guide**

Below are some important notes on upgrading to FishEye 2.7. For details of the new features and improvements in this release, please read the FishEye 2.7 Release Notes.

### On this page:
- Upgrade Notes
  - FishEye 2.7
- Upgrade Procedure
- Checking for Known Issues and Troubleshooting the FishEye Upgrade

#### Upgrade Notes

**FishEye 2.7**

- When FishEye 2.7 is upgraded, it will run a one-time upgrade task that will create a new index of file revisions, and minor database indexing tasks. These should have no functional impact and should complete in a few minutes.
- FishEye 2.7 adds ActiveObjects support, which allows plugins to store configuration data in the database. Any already scheduled backups will not have plugin data included during backups. Please modify your
scheduled backup configuration appropriately to include ActiveObjects backups.

- If you are using Custom authentication, FishEye 2.7's managed repositories have changed the responsibilities of the com.cenqua.fisheye.user.plugin.FishEyeAuthenticator class. For external repositories the hasPermissionToAccess() method is still invoked to check whether a user may access a particular repository. For internal or managed repositories, hasPermissionToAccess() is used to determine whether a user has both read and write access to the repository.

Upgrade Procedure

⚠️ Before you begin
- Test your upgrades in your test environment before rolling into production.
- Back up your entire FishEye instance (see Backing up and restoring FishEye data), i.e.
  - If you are backing up your FishEye instance using the Admin interface, tick all of the 'Include' checkboxes (e.g. repository and application caches, plugins and their configuration data, SQL database, etc).
  - If you are backing up your FishEye instance using the command-line interface, do not use any exclusion options.

If you are already running a version of FishEye, please follow the instructions in the general FishEye Upgrade Guide.

Checking for Known Issues and Troubleshooting the FishEye Upgrade

If something is not working correctly after you have completed the steps above to upgrade your FishEye installation, please check for known FishEye issues and try troubleshooting your upgrade as described below:

- **Check for known issues.** Sometimes Atlassian finds out about a problem with the latest version of FishEye after the software is released. In such cases we publish information about the known issues in the FishEye Knowledge Base. Please check the FishEye 2.7 Known Issues in the FishEye Knowledge Base and follow the instructions to apply any necessary patches if necessary.

- **Did you encounter a problem during the FishEye upgrade?** Please refer to the guide to troubleshooting upgrades in the FishEye 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.

**RELATED TOPICS**

FishEye 2.7 Release Notes

FishEye 2.6 Release Notes

**6 June 2011**

With great pleasure, Atlassian presents FishEye 2.6 featuring the new repository commit graph, user management via JIRA and a host of DVCS improvements.

**Highlights of this Release:**

- Repository Commit Graph
- User Management via JIRA
- Improved Quick Search
- Redesigned HTML Emails
- Dashboard and Navigation Improvements
- Improved Support for Git Branches
- Git Commit Authors include Email Address
- Mercurial Indexing Improvements
And Even More Improvements

Responding to your Feedback:

⭐ Over 269 votes satisfied

Thank you for all your issues and votes. Keep logging issues to help us keep improving!

Read the release notices for important information about this release.

Highlights of FishEye 2.6

Repository Commit Graph

FishEye 2.6 introduces a powerful visualisation tool to help you understand your repositories better – the commit graph. The commit graph shows changesets in their respective branches, using configurable "swimlanes". At a glance, you will be able to see key information such as branching and merging. If you are using Git or Mercurial, you will also be able to see anonymous branches.

Commit graph highlights provide further context, allowing you to highlight changesets in the same branch, commits with JIRA issues, and reviewed/unreviewed changesets. Clicking a changeset with the appropriate highlight active (selected in the 'Highlight' dropdown) shows you related changesets, such as changesets with the same lineage, the same JIRA issue or same Crucible review.
User Management via JIRA

You can now use the same set of users in FishEye and JIRA, and manage your users and groups in JIRA. The FishEye setup wizard gives you the opportunity to configure a JIRA connection quickly and automatically, using the most common options. You can further configure the JIRA connections via the FishEye administration screens.
Improved Quick Search

You'll be able to find information faster using FishEye's improved Quick Search. The Quick Search now pattern matches against CamelCase strings for files and directories, just like many popular IDEs. The search itself provides quicker and more accurate results. The search results are also easier to work with – the new user interface has a cleaner look and feel and features links to default/trunk for files, pop-up summaries for JIRA issues in commit messages and more.

More...

Redesigned HTML Emails

A 20% time project has resulted in email notifications getting a dramatic facelift. Gone are the dreary old HTML emails, replaced by much better-looking ones. You'll see the new emails in action for changeset notifications.
Dashboard and Navigation Improvements

The Dashboard and Header have been tweaked to simplify the user interface in this release. You won't have to click different tabs to find the activity stream, as it will always be displayed. We've moved the other functions to the header to remove the clutter and provide you with a more streamlined view.

More...

Improved Support for Git Branches

We've improved the support for Git branches in FishEye. Previously, if FishEye detected a commit on a branch (other than the 'master' branch), it would be considered as part of that branch only, even if the branch was later merged back to the 'master' branch. In this release, FishEye will correctly consider the ancestry of a changeset. For example, if a branch 'fisheye-2.6' is merged back to the 'master' branch, then all changesets that were seen as part of the 'fisheye-2.6' branch will also be considered to be part of the 'master'.

The activity stream for a branch will now show all commits that contribute to (i.e. are in the ancestry of) the head of the branch. You can also see the improved Git branch support on the new commit graph and on the view changeset screen.
Git Commit Authors include Email Address

FishEye now includes the user email address for Git commit authors, in the same style as command line tools. The committer will be automatically mapped to the FishEye user with the same email address. This means that you won't have to set up explicit user mappings where the email address is the same.

Please note, if you are using FishEye with a starter license, this improvement may cause you to exceed the user limit for your license. See the FishEye 2.6 Upgrade Guide before upgrading for details.

Mercurial Indexing Improvements

We have changed the way that we index Mercurial. Indexing times for merge commits should be almost twice as fast. Commits with many file changes (especially binary file changes) will be even faster. The absolute change in indexing speed will vary depending on the content of your repository. As a reference, the indexing time for our repository of 24,000 commits improved from 18 hours to 6.5 hours.

And Even More Improvements

Visit our issue tracker to see the full list of improvements and bug fixes in FishEye and Crucible for this release.

Release Notices

- **Upgrading from a previous version of FishEye.** Upgrading FishEye should be fairly straightforward. We strongly recommend that you back up FishEye before upgrading. Please refer to the FishEye 2.6 Upgrade Guide for further essential information about your upgrade.

- **Known Issues.** Please check the important technical advisories on the front page of the Knowledge Base.
FishEye 2.6 Changelog

This page contains information about the FishEye 2.6 minor releases. See the FishEye 2.6 Release Notes for details of what's new in 2.6.0.

Please read the FishEye 2.6 Upgrade Guide before upgrading to any of the minor releases below.

On this page:
- From 2.6.6 to 2.6.7
- From 2.6.5 to 2.6.6
- From 2.6.4 to 2.6.5
- From 2.6.3 to 2.6.4
- From 2.6.2 to 2.6.3
- From 2.6.1 to 2.6.2
- From 2.6.0 to 2.6.1

From 2.6.6 to 2.6.7

31 January 2012

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (1 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>

From 2.6.5 to 2.6.6

2 September 2011

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (10 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Case</td>
<td>Summary</td>
<td>Assignee</td>
<td>Resolution</td>
<td>Status</td>
<td>Created</td>
<td>Updated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>----------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------------</td>
<td>---------</td>
<td>----------</td>
<td>----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-365</td>
<td>Issue</td>
<td>Summary</td>
<td>Assignee</td>
<td>Resolution</td>
<td>Created</td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>---------</td>
<td>----------</td>
<td>------------</td>
<td>---------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From 2.6.4 to 2.6.5

24 August 2011

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (1 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>

From 2.6.3 to 2.6.4

22 August 2011

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (7 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>📋</td>
<td>FE-3800</td>
<td>StoreDiff and Loc can't be</td>
<td>Conor MacNei</td>
<td>Conor MacNei</td>
<td>↓</td>
<td>🌴 Closed</td>
<td>Fixed</td>
<td>Aug 08, 2011</td>
<td>Oct 24, 2011</td>
</tr>
</tbody>
</table>
### Issue FE-365

**Number**: 5  
**Title**: IllegalStateException and IllegalArgumentException exceptions in activity streams  
**Assignee**: Geoff Crain  
**Status**: Closed  
**Priority**: Fixed  
**Categories**:  
**Created**: Aug 18, 2011  
**Updated**: Aug 29, 2011

### Issue FE-365

**Number**: 4  
**Title**: JIRA FishEye plugin not working - Error handling trusted application authentication attempt: BAD_SIGNATURE  
**Assignee**: Tim Pettersen  
**Status**: Closed  
**Priority**: Fixed  
**Categories**:  
**Created**: Aug 17, 2011  
**Updated**: Jan 24, 2012

### Issue FE-363

**Number**: 9  
**Title**: FishEye Crowd integration doesn't work when the group name has a space  
**Assignee**: Geoff Crain, Ajay Sridhar  
**Status**: Closed  
**Priority**: Fixed  
**Categories**:  
**Created**: Aug 10, 2011  
**Updated**: Aug 18, 2011

### Issue FE-363

**Number**: 6  
**Title**: Empty "tag"  
**Assignee**: Geoff Crain, Nick Pellow  
**Status**: Closed  
**Priority**: Fixed  
**Categories**:  
**Created**: Aug 09, 2011  
**Updated**: Aug 18, 2011
From 2.6.2 to 2.6.3

9 August 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (5 issues)
<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-360</td>
<td>Setting a user's authentication type to other than 'built in' when 'built in' is in use causes NPE when editing that user's details</td>
<td>Tom Davies [Atlassian]</td>
<td>Tom Davies [Atlassian]</td>
<td>↓</td>
<td>⬤ Closed</td>
<td>Fixed</td>
<td>Jul 26, 2011</td>
<td>Aug 09, 2011</td>
</tr>
</tbody>
</table>
FE-359
HG: Commit with no commit message break indexing

From 2.6.1 to 2.6.2
20 July 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (18 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>------</td>
<td>----------------------------------------------------------</td>
<td>-------------------------</td>
<td>----------------------------</td>
<td>-------</td>
<td>-------</td>
<td>---------------</td>
<td>---------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Summary</td>
<td>Assignee(s)</td>
<td>Resolution</td>
<td>Created</td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
<td>------------</td>
<td>---------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-3559</td>
<td>FishEye can't scan repository. &quot;Cannot create a position that is between two identical positions&quot;</td>
<td>Michael Heemskerk [Atlassian] Nick Pellow [Atlassian]</td>
<td>Closed</td>
<td>Jul 05, 2011</td>
<td>Aug 17, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>Issue</td>
<td>Description</td>
<td>Author(s)</td>
<td>Status</td>
<td>Fixed</td>
<td>Date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-352</td>
<td>8</td>
<td>NPE with CVS repo when doing a Lucene index upgrade (when there is some strange headless state)</td>
<td>Joe Xie [Atlassian]</td>
<td></td>
<td></td>
<td>Jun 16, 2011</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-350</td>
<td>7</td>
<td>Polling mechanism skips repositories when there are a lot of them</td>
<td>Tom Davies [Atlassian]</td>
<td></td>
<td></td>
<td>Jun 14, 2011</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-347</td>
<td>8</td>
<td>Git diff processing fails for diffs with ^M character</td>
<td>Unsigned</td>
<td></td>
<td></td>
<td>Jun 02, 2011</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From 2.6.0 to 2.6.1

22 June 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (11 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>[Atlassian]</td>
<td>[Atlassian]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>FE-354 3</td>
<td>Nullpointer exception in yours showcase / sandbox</td>
<td>Unassigned</td>
<td>Jens Bormüller [avono AG]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>----------</td>
<td>--------------------------------------------------</td>
<td>------------</td>
<td>---------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Clos ed</td>
<td>Jun 29, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Jul 06, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FE-352 9</td>
<td>NPE during EyeQL when review has null description</td>
<td>Adam Ahmed [Atlassian]</td>
<td>Adam Ahmed [Atlassian]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Clos ed</td>
<td>Jun 18, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Jun 19, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FE-351 3</td>
<td>No Horizontal Scroll Bar on Any Non-G WT Admin Pages</td>
<td>Seb Ruiz [Atlassian]</td>
<td>Tom Davies [Atlassian]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Clos ed</td>
<td>May 26, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Jun 29, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FE-351 0</td>
<td>Login screen after setup has incomplete wording</td>
<td>Seb Ruiz [Atlassian]</td>
<td>Sarah Maddox [Atlassian]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Clos ed</td>
<td>Jun 02, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Jun 20, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FE-350 5</td>
<td>Alerts boxes show up on every page for IE 7 saying that the browse</td>
<td>Jason Hinch [Atlassian]</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Clos ed</td>
<td>Jun 01, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Jun 14, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Title</td>
<td>Assignee</td>
<td>Resolution</td>
<td>Status</td>
<td>Created</td>
<td>Updated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
<td>-------------------</td>
<td>------------</td>
<td>-------------</td>
<td>-----------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-346</td>
<td>'Save' button</td>
<td>Tom Davies</td>
<td>Fixed</td>
<td>Closed</td>
<td>May 26, 2011</td>
<td>Oct 17, 2011</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FishEye 2.6 Upgrade Guide

Below are some important notes on upgrading to **FishEye 2.6**. For details of the new features and improvements in this release, please read the **FishEye 2.6 Release Notes**.

**On this page:**
- Upgrade Notes
  - FishEye 2.6
- Upgrade Procedure
- Checking for Known Issues and Troubleshooting the FishEye Upgrade

**Upgrade Notes**

**FishEye 2.6**

- **Internet Explorer 7 and Java Platform 5 (JDK/JRE 1.5) no longer supported** — As per the End of Support Announcements for FishEye (published previously), we are no longer supporting Internet Explorer 7 and Java Platform 5 (JDK/JRE 1.5) in this release. See Supported platforms page for the full list of supported platforms for FishEye.

- **Starter Licence holders who use Git** — The inclusion of the user's email address in Git committer names allows committers to be automatically mapped to FishEye users. However, if your repository has the same committer using multiple email addresses in their commits, this change means you will get extra committers (as uniquely identified by the combination of username and email address). This may cause you to exceed the committer limit allowed by your license, which would stop FishEye from indexing your repository. To correct this, you need to rewrite the history of your repository to remove these duplicates, so that each committer has only one unique email address. You can do this using the `git-filter-branch` command as detailed in this knowledge base article.

- **Set aside additional time for the upgrade** — When you upgrade to FishEye 2.6, FishEye will run a number of upgrade tasks and trigger an automatic upgrade of the metadata index. While this is happening, FishEye (and Crucible) can be be accessed, however no new changes in the repositories will be detected until the upgrade is complete. Hence, you should set aside time for these activities to complete. The time taken will depend on the complexity of your code base, as well as the type of repository that you are using. See the following repository-specific notes:
| Subversion | The upgrade tasks for Subversion repositories will take significantly longer in this release, than in previous releases. In previous releases (e.g. FishEye 2.5), the upgrade tasks only required a short amount of time to complete, relative to the time taken to upgrade the metadata index. In this release, the upgrade tasks and metadata index upgrade take approximately equal amounts of time. Hence, the overall upgrade may take up to twice as long, compared to prior releases. Our Subversion repository for JIRA development, which is quite complex, took about three hours to complete both the upgrade tasks and metadata index upgrade. |
| CVS, Perforce, ClearCase | As per Subversion, the upgrade tasks for CVS and Perforce will take proportionally longer to complete in this release, hence the overall upgrade will take longer. However, the tasks are not as complex as the Subversion upgrade tasks, so you shouldn’t notice a dramatic increase in time taken. The upgrade task for ClearCase is not complex at all, so you shouldn’t notice a significant increase in time taken for the overall upgrade. |
| Git, Mercurial | The upgrade tasks for Git & Mercurial repositories will not take long to complete, perhaps a few minutes. You will still need to set aside time for the metadata index upgrade though. |

To reduce the time taken for the upgrade tasks, you can configure additional update threads to allow repositories to be upgraded in parallel (otherwise, repositories will be upgraded one at a time). Please ensure that you have the compute capacity to run multiple threads before attempting this.

**Upgrade Procedure**

**Before you begin**
- Test your upgrades in your test environment before rolling into production.
- Back up your *entire* FishEye instance (see [Backing up and restoring FishEye data](#)), i.e.
  - If you are backing up your FishEye instance using the Admin interface, tick all of the 'Include' checkboxes (e.g. repository and application caches, plugins and their configuration data, SQL database, etc).
  - If you are backing up your FishEye instance using the command-line interface, do not use any exclusion options.

If you are already running a version of FishEye, please follow these instructions on [FishEye upgrade guide](#).

**Checking for Known Issues and Troubleshooting the FishEye Upgrade**

If something is not working correctly after you have completed the steps above to upgrade your FishEye installation, please check for known FishEye 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 FishEye after we have released the software. In such cases we publish information about the known issues in the FishEye Knowledge Base. Please check the [FishEye 2.6 Known Issues](#) in the FishEye Knowledge Base and follow the instructions to apply any necessary patches if necessary.

• **Did you encounter a problem during the FishEye upgrade?** Please refer to the guide to troubleshooting upgrades in the FishEye 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.

**RELATED TOPICS**

FishEye 2.6 Release Notes
FishEye 2.5 Release Notes

8 February 2011
With great pleasure, Atlassian presents **FishEye 2.5**, providing vastly improved source code searching and DVCS authentication.

**Highlights of this Release:**
- Search Revamp
- Redesigned Activity Stream
- Mercurial and Git Authentication
- RSS Improvements
- Improved Header
- And Even More Improvements

**Responding to your Feedback:**
⭐ Over 150 votes satisfied

[DOWNLOAD latest version]

- Thank you for all your issues and votes. Keep logging issues to help us keep improving!
- Read the release notices for important information about this release.

**Highlights of FishEye 2.5**

Search Revamp

FishEye's Quick Search functionality has been redesigned from the ground up. Among the improvements:

- Faster search breaking out results by content type
- Improved layout for easier scanning
- Smarter search results with results prioritized to the top by modified time
- For each path, view the top three most recent branches with a modification of that file

Read more about Quick Search...
Redesigned Activity Stream

We have spent a considerable amount of time improving the FishEye activity stream:

- Cleaner visual design
- Larger avatars
- Improved scanability
- De-cluttering of UI elements, showing actions on hover
- Space-saving condensed mode so you can see more changesets on the same page
- Embedded images are thumbnailed

Read more about the changelog...
Mercurial and Git Authentication

We have added http(s) and ssh authentication for Git and Mercurial repositories, allowing you to more easily browse your private repositories hosted on popular sites such as GitHub and BitBucket. FishEye now supports http authentication with passwords and ssh using public/private keypairs. You can let FishEye generate a pair and upload the public key to your hosting provider or upload a private key (passphraseless only) to FishEye if you prefer.

Read more about Authentication...

New Repository - Page 2 of 3

Mercurial Connection Details

Repository Location: 

Mercurial Authentication

Authentication Style: Generate key pair for ssh

Generate a key pair, then upload the public key to your repository server. Note: SSH keys cannot be used for http(s) urls.

Private Key: Private key has been generated by FishEye

Public Key: 

Copy this public key to your repository server

Remove

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
RSS Improvements

FishEye 2.5 introduces some fine tuning for rendered RSS feeds. RSS content is easier to digest (better titles) and allows you to consume FishEye activity through your favourite RSS reader. We render wiki-markup in RSS which will allow you to click through links and view images inline.

Improved Header

FishEye 2.5 tracks locations that you have recently visited, and provides quick and easy access to navigate to these resources. The five most recently viewed repositories, projects, users and reviews (if Crucible is installed) are available from the header drop down links as shown:

And Even More Improvements

- The Universal Plugin Manager
- Complete branch selector support for Subversion repositories
- Oracle support when FishEye is teamed with Crucible
Visit our issue tracker to see the full list of improvements and bug fixes in FishEye and Crucible for this release.

FishEye 2.5 Changelog

This page contains information about the FishEye 2.5 minor releases.

⚠️ Please read the FishEye 2.5 Upgrade Guide before upgrading to any of the minor releases below.

On this page:
- From 2.5.6 to 2.5.7
- From 2.5.5 to 2.5.6
- From 2.5.4 to 2.5.5
- From 2.5.3 to 2.5.4
- From 2.5.2 to 2.5.3
- From 2.5.1 to 2.5.2
- From 2.5.0 to 2.5.1

From 2.5.6 to 2.5.7

22 June 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (4 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>🕵️‍♂️</td>
<td>FE-346</td>
<td>'Save'</td>
<td>Tom</td>
<td>Tom</td>
<td>▼</td>
<td>🟢 Closed</td>
<td>Fixed</td>
<td>May</td>
<td>Oct 17,</td>
</tr>
</tbody>
</table>
3 button on Database Configuration, Edit Admin page doesn't save the new config

From 2.5.5 to 2.5.6

24 May 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (6 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>📝</td>
<td>FE-344</td>
<td>Do not throw exception when git content cannot be fetched.</td>
<td>Conor MacNeill</td>
<td>Conor MacNeill</td>
<td>↓</td>
<td>Clos ed</td>
<td>Fixed</td>
<td>May 16, 2011</td>
<td>May 19, 2011</td>
</tr>
<tr>
<td>📝</td>
<td>FE-343</td>
<td>Prevent SSH from dropping down to password auth when using SSH keys.</td>
<td>Conor MacNeill</td>
<td>Conor MacNeill</td>
<td>↓</td>
<td>Clos ed</td>
<td>Fixed</td>
<td>May 12, 2011</td>
<td>May 17, 2011</td>
</tr>
</tbody>
</table>
view:
Clicking refresh causes same status message to repeat

[Atlassian]

Conor MacNeill [Atlassian]

Fixed
May 17, 2011
May 17, 2011

FE-344
4
Handle unversioned element in isactivity

[Atlassian]

Conor MacNeill [Atlassian]

FE-343
6
500 error in the search (java.lang.Null Pointer Exception)

Pierre-Etienne Poirot [Atlassian]

Fixed
May 11, 2011
May 26, 2011

FE-334
5
ClearCase: custom element types are treated as binary files.

Conor MacNeill [Atlassian]

Michael Heemskerk [Atlassian]

Fixed
Dec 21, 2010
May 26, 2011

From 2.5.4 to 2.5.5

11 May 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (9 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Create</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Status</td>
<td>Description</td>
<td>Created</td>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-379 7</td>
<td>Closed</td>
<td>XSS vulnerability in user's profile display name</td>
<td>Oct 24, 2011</td>
<td>Nov 30, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-343 2</td>
<td>Closed</td>
<td>Perforce escape for * is case sensitive</td>
<td>May 10, 2011</td>
<td>May 10, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-302 8</td>
<td>Closed</td>
<td>Gutter renderer breaks the source layout</td>
<td>Apr 18, 2011</td>
<td>May 09, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FE-302 6</td>
<td>Closed</td>
<td>Search hint: when a hint is specified, clicking on a different search type does</td>
<td>Apr 17, 2011</td>
<td>Apr 18, 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
not do anything

FE-302 5 FishEye watch notifications can potentially not send change sets due to long running incremental slurps


FE-302 3 Annot page - resizing window in FF causes double scrollbars and/or a really long page.


FE-302 1 AJP authentication regression


From 2.5.3 to 2.5.4

11 April 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (8 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍃</td>
<td>FE-301</td>
<td>add</td>
<td>Adam</td>
<td>Geoff</td>
<td>↑</td>
<td>Clos</td>
<td>Fixed</td>
<td>Apr 06, 2011</td>
<td>Apr 06, 2011</td>
</tr>
</tbody>
</table>

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
<table>
<thead>
<tr>
<th>Issue ID</th>
<th>Title</th>
<th>抱歉</th>
<th>Assignee</th>
<th>Product</th>
<th>Version</th>
<th>Created</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE-338</td>
<td>IE8 source view window resize bug</td>
<td>未解决</td>
<td>None</td>
<td></td>
<td></td>
<td>2011</td>
<td>2011</td>
</tr>
<tr>
<td>FE-301</td>
<td>Diff View - deplink to a line using an 'l' hash (e.g., #98) causes js error</td>
<td>未解决</td>
<td>Adam Ahmed [Atlassian]</td>
<td></td>
<td></td>
<td>2011</td>
<td>2011</td>
</tr>
<tr>
<td>FE-300</td>
<td>clearcase ucm: we aren't storing</td>
<td>未解决</td>
<td>Geoff Crain [Atlassian]</td>
<td></td>
<td></td>
<td>2011</td>
<td>2011</td>
</tr>
</tbody>
</table>
the latest possible activity time, so it's possible to get a lot of unnecessary data.

**FE-299**

FeCru TransactionTemplate implementation should close the Hibernate Session opened during the transaction.

**FE-175**

In diff view, selecting a context clears any ignore settings, and vice versa.

---

**From 2.5.2 to 2.5.3**

**8 March 2011**

This is a bug fix release. The complete list of issues is below.

**JIRA Issues** (12 issues)
<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
null pointer exception in group edit action.

The database is not Unicode. Crucible requires that the database use a case sensitive collation "to reflect the current check: "The database is not Unicode. Crucible requires ..."

FE-307 5 NullPointer Exception in GroupEditAction

Pierre-Etienne Poirot [Atlassian] None

Closed Fixed Mar 20, 2011 May 09, 2011

FE-305 2 Recent visited tabs spins forever when REST end points are disabled

Pierre-Etienne Poirot [Atlassian] None

Closed Fixed Mar 15, 2011 May 09, 2011

FE-299 clearca Geoff Geoff


Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
<table>
<thead>
<tr>
<th>#</th>
<th>Case</th>
<th>Description</th>
<th>Assigned To 1</th>
<th>Assigned To 2</th>
<th>Status</th>
<th>Fixed</th>
<th>Date</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td>base: use the latest repository activity when indexing branches</td>
<td>Crain [Atlassian]</td>
<td>Crain [Atlassian]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
indexing error ‘No such file or directory’

FE-297
Go-To-Change set doesn’t work

Matthe w Watson [Atlassi an]
Erik van Zijst [Atlassi an]

↑ Clos ed Fixed Mar 06, 2011 Mar 24, 2011

FE-296
Deadlock in Jetty connection timeout

Joe Xie [atlassi an]
Gurleen Anand [Atlassi an]


FE-295
LineDecor ator. décorateLine(int) never call

Tom Davi es [Atlassi an]
Julien Hoarau [Old]


From 2.5.1 to 2.5.2

8 March 2011

This is a bug fix release. The complete list of issues is below.

JIRA Issues (10 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FE-325</td>
<td>Add option to disable git head detection and tag processing</td>
<td>Conor MacNei ll [Atlassi an]</td>
<td>Conor MacNei ll [Atlassi an]</td>
<td>↓</td>
<td>Clos ed</td>
<td>Fixed</td>
<td>Feb 03, 2011</td>
<td>May 09, 2011</td>
</tr>
</tbody>
</table>
FishEye doesn't index project components that aren't components of the project's PVOB.

FE-315 7  Git isn't detecting binary files at all  Michael Heems kerk [Atlassian]  None  Clos ed  Fixed  Feb 02, 2011  May 09, 2011

FE-312 3  Anchoring Patch to Git Repo Causes NPE  Geoff Crain [Atlassian]  None  Clos ed  Fixed  Feb 02, 2011  May 09, 2011


Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
<table>
<thead>
<tr>
<th>Issue #</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE-296</td>
<td>Fixed</td>
<td>ClearCase: cleartoolk message is missed by FishEye and as a result indexing stops in error for known 'error' conditions</td>
</tr>
<tr>
<td>FE-295</td>
<td>Fixed</td>
<td>P4 Unicode Handling</td>
</tr>
<tr>
<td>FE-294</td>
<td>Fixed</td>
<td>java.lang.IllegalStateException - No match found exception thrown when the clearca</td>
</tr>
</tbody>
</table>
### From 2.5.0 to 2.5.1

**10 February 2011**

This is a bug fix release. The complete list of issues is below.

#### JIRA Issues (17 issues)

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Summery</th>
<th>Assignee</th>
<th>Reporter</th>
<th>Priority</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
</table>
CRUC-5599  Upgrade scripts do not always create the same schema as scripts to create a new database

CRUC-5597  The "Tabs" on the statistics panel of the "Project View" do not work in JIRA 4.2.3 and JIRA 4.3

CRUC-5594  Adding an AppLink to a "standalone Crucible" creates a FishEye configuration that is unusable.
CRUC-5574  Update "refresh cache" links to synch form App Links as well.

CRUC-5526  FishEye instance gets re-created on JIRA restart

CRUC-5475  Create func tests for all search result pages
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-5183</td>
<td>Extend OAuth services to REST resources (such as gadgets)</td>
<td>Joe Xie [atlassian]</td>
<td>Joe Xie [atlassian]</td>
<td>Fixed</td>
<td>Dec 13, 2010</td>
<td>Feb 13, 2011</td>
</tr>
</tbody>
</table>
FishEye 2.5 Upgrade Guide

Below are some important notes on upgrading to FishEye 2.5. For details of the new features and improvements in this release, please read the FishEye 2.5 Release Notes.

On this page:
- Upgrade Notes
- Upgrade Procedure
- Checking for Known Issues and Troubleshooting the FishEye Upgrade

Upgrade Notes

- **FishEye 2.5:**
  - This release will trigger an automatic upgrade of the metadata index the first time FishEye is started. FishEye is still usable while this task runs in the background, however the search functionality will not contain all results. A warning will appear in the UI until the upgrade has complete. Please also see How do I Avoid Long Reindex Times When I Upgrade?
  - Some RSS readers may duplicate items when first fetching feeds after upgrading because RSS titles have changed.

Upgrade Procedure

⚠️ **Before you begin**
- Test your upgrades in your test environment before rolling into production.
- Back up your *entire* FishEye instance (see Backing up and restoring FishEye data), i.e.
  - If you are backing up your FishEye instance using the Admin interface, tick all of the 'Include' checkboxes (e.g. repository and application caches, plugins and their configuration data, SQL database, etc).
  - If you are backing up your FishEye instance using the command-line interface, do not use any exclusion options.

If you are already running a version of FishEye, please follow these instructions on FishEye upgrade guide.

Checking for Known Issues and Troubleshooting the FishEye Upgrade

If something is not working correctly after you have completed the steps above to upgrade your FishEye installation, please check for known FishEye 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 FishEye after we have released the software. In such cases we publish information about the known issues in the FishEye Knowledge Base. Please check the [FishEye 2.5 Known Issues](https://confluence.atlassian.com/display/FISH/Eye/FishEye+2.5+Known+Issues) in the FishEye Knowledge Base and follow the instructions to apply any necessary patches if necessary.

• **Did you encounter a problem during the FishEye upgrade?** Please refer to the guide to [troubleshooting upgrades](https://confluence.atlassian.com/display/FISH/Eye/FishEye+Knowledge+Base) in the FishEye Knowledge Base.

• If you encounter a problem during the upgrade and cannot solve it, please create a [support ticket](https://confluence.atlassian.com/display/FISH/Eye) and one of our support engineers will help you.

**RELATED TOPICS**

- [FishEye 2.5 Release Notes](https://confluence.atlassian.com/display/FISH/Eye/FishEye+2.5+Release+Notes)
- [Security advisories](https://confluence.atlassian.com/display/FISH/Eye/Security+advisories)

**Finding and reporting a security vulnerability**


**Publication of Bamboo security advisories**


**Severity levels**

Atlassian’s scale for measuring security issues is detailed in [Severity Levels for Security Issues](https://confluence.atlassian.com/display/FISH/Eye/Severity+Levels+for+Security+Issues).

**Our patch policy**

Atlassian’s approach to releasing patches is detailed in our [Security Patch Policy](https://confluence.atlassian.com/display/FISH/Eye/Security+Patch+Policy).

**Security advisories**

- [FishEye and Crucible Security Advisory 2012-08-21](https://confluence.atlassian.com/display/FISH/Eye/FishEye+and+Crucible+Security+Advisory+2012-08-21)
- [FishEye and Crucible Security Advisory 2012-05-17](https://confluence.atlassian.com/display/FISH/Eye/FishEye+and+Crucible+Security+Advisory+2012-05-17)
- [FishEye and Crucible Security Advisory 2012-01-31](https://confluence.atlassian.com/display/FISH/Eye/FishEye+and+Crucible+Security+Advisory+2012-01-31)
- [FishEye and Crucible Security Advisory 2011-11-22](https://confluence.atlassian.com/display/FISH/Eye/FishEye+and+Crucible+Security+Advisory+2011-11-22)
- [FishEye and Crucible Security Advisory 2011-05-16](https://confluence.atlassian.com/display/FISH/Eye/FishEye+and+Crucible+Security+Advisory+2011-05-16)
- [FishEye and Crucible Security Advisory 2011-01-12](https://confluence.atlassian.com/display/FISH/Eye/FishEye+and+Crucible+Security+Advisory+2011-01-12)
- [FishEye Security Advisory 2010-10-20](https://confluence.atlassian.com/display/FISH/Eye/FishEye+Security+Advisory+2010-10-20)
- [FishEye Security Advisory 2010-06-16](https://confluence.atlassian.com/display/FISH/Eye/FishEye+Security+Advisory+2010-06-16)
- [FishEye Security Advisory 2010-05-04](https://confluence.atlassian.com/display/FISH/Eye/FishEye+Security+Advisory+2010-05-04)

**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.atlassi](http://support.atlassi)
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](https://confluence.atlassian.com/display/FESECURITY/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](https://jira.atlassian.com/browse/FE-4222) - Authenticate to see issue details and [CRUC-6188](https://jira.atlassian.com/browse/CRUC-6188) - Authenticate to see issue details.

The table below describes the FishEye and/or Crucible versions and the specific functionality affected by the vulnerabilities.

<table>
<thead>
<tr>
<th>FishEye and/or Crucible Vulnerability</th>
<th>Affected versions</th>
<th>Fixed Version</th>
<th>Issue Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevation of privileges</td>
<td>2.5.x or earlier</td>
<td>2.5.9</td>
<td><a href="https://jira.atlassian.com/browse/FE-4222">FE-4222</a> - Authenticate to see issue details</td>
</tr>
<tr>
<td></td>
<td>2.6.x</td>
<td>2.6.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.7.x</td>
<td>2.7.15, 2.8.0</td>
<td><a href="https://jira.atlassian.com/browse/CRUC-6188">CRUC-6188</a> - Authenticate to see issue details</td>
</tr>
</tbody>
</table>

**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 - Authenticate to see issue details

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.

<table>
<thead>
<tr>
<th>FishEye and Crucible Component</th>
<th>Affected FishEye and Crucible Versions</th>
<th>Fixed Version</th>
<th>Issue Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebWork 2</td>
<td>2.0 - 2.7.8</td>
<td>2.6.7</td>
<td>FE-3891</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.7.9</td>
<td></td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>FishEye/Crucible Feature</th>
<th>Affected FishEye/Crucible Versions</th>
<th>Fixed Version</th>
<th>Issue Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>FishEye user profile - display name</td>
<td>2.5.4 and earlier</td>
<td>2.5.5</td>
<td>FE-3797</td>
</tr>
<tr>
<td>FishEye user profile - snippets in a user's comment</td>
<td>2.5.4 and earlier</td>
<td>2.5.5</td>
<td>FE-3798</td>
</tr>
</tbody>
</table>

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 of 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.

<table>
<thead>
<tr>
<th>FishEye/Crucible Feature</th>
<th>Affected FishEye/Crucible Versions</th>
<th>Fixed Version</th>
<th>Issue Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>FishEye changeset tooltips</td>
<td>FishEye 2.4.6 to 2.5.6</td>
<td>FishEye 2.5.7</td>
<td>FE-3811</td>
</tr>
<tr>
<td>Crucible review tooltips</td>
<td>Crucible 2.4.6 to 2.5.6</td>
<td>Crucible 2.5.7</td>
<td>CRUC-5811</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>FishEye/Crucible Feature</th>
<th>Affected FishEye/Crucible Versions</th>
<th>Issue Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crucible snippets</td>
<td>Crucible 2.4.5 to 2.5.0</td>
<td>CRUC-5734</td>
</tr>
<tr>
<td>Crucible author mapping</td>
<td>Crucible 2.4.5 to 2.5.0</td>
<td>CRUC-5735</td>
</tr>
<tr>
<td>Crucible changeset comments in search results</td>
<td>Crucible 2.3.0 to 2.5.0</td>
<td>CRUC-5736</td>
</tr>
<tr>
<td>Crucible comments search</td>
<td>Crucible 2.2.6 to 2.5.0</td>
<td>CRUC-5737</td>
</tr>
<tr>
<td>FishEye/Crucible dashboard - review activity</td>
<td>FishEye/Crucible 2.2.8 to 2.5.2</td>
<td>FE-3031</td>
</tr>
<tr>
<td>FishEye/Crucible reviews list</td>
<td>FishEye/Crucible 2.2.8 to 2.5.2</td>
<td>FE-3032</td>
</tr>
</tbody>
</table>

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.
### 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**

There are no patches available to fix this vulnerability. You must upgrade your FishEye/Crucible installation.
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.

**FishEye Security Advisory 2010-10-20**

This advisory announces a number of security vulnerabilities in earlier versions of FishEye that we have found and fixed in FishEye 2.4 and FishEye 2.3.7. In addition to releasing FishEye 2.4 and FishEye 2.3.7, we also provide a patch for the vulnerabilities mentioned below. You will be able to apply this patch to existing installations of FishEye 2.3.6. However, we recommend that you upgrade to FishEye 2.4 to fix these vulnerabilities.

In this advisory:

- XSS Vulnerabilities
  - Severity
  - Risk Assessment
  - Vulnerabilities
  - Risk Mitigation
  - Fix
- Available Patches
  - Step 1 of the Patch Procedure: Install the Patch

**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 instances, including publicly available instances.

- An attacker might take advantage of an XSS vulnerability to steal the current session of a logged-in user.
- XSS vulnerabilities potentially allow an attacker to embed their own JavaScript into a FishEye page. An attacker's text and script might be displayed to other people viewing the page.

You can read more about XSS attacks at cgisecurity, CERT and other places on the web.

**Vulnerabilities**

The table below describes the parts of FishEye affected by the XSS vulnerabilities.

<table>
<thead>
<tr>
<th>FishEye Feature</th>
<th>Affected FishEye Versions</th>
<th>Issue Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Metrics Plugin</td>
<td>2.0.x to 2.3.6 inclusive</td>
<td>CRUC-4572</td>
</tr>
<tr>
<td>FishEye Revision ID Parameters on Annotated Views</td>
<td>2.3.0 to 2.3.6 inclusive</td>
<td>CRUC-4641</td>
</tr>
</tbody>
</table>
Risk Mitigation

We recommend that you upgrade your FishEye installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable the 'Code Metrics Plugin' via the Administration Console ('Plugins' menu item under 'Systems Settings') to mitigate the Code Metrics Plugin XSS vulnerability. There is no mitigation for the FishEye Revision ID Parameters on Annotated Views XSS vulnerability.

Fix

FishEye-only installations:
FishEye 2.4 (recommended) and FishEye 2.3.7 fix these issues. For a full description of the FishEye 2.4 release, see the release notes. You can download FishEye 2.4 from the download centre. You can download FishEye 2.3.7 from the download centre archives.

If you cannot upgrade to FishEye 2.4/2.3.7, you can patch your existing installation using the patch listed below.

FishEye+Crucible installations:
Crucible 2.4 (recommended) and Crucible 2.3.7 fix these issues. For a full description of the Crucible 2.4 release, see the release notes. You can download Crucible 2.4 from the download centre. You can download Crucible 2.3.7 from the download centre archives.

If you cannot upgrade to Crucible 2.4/2.3.7, you can patch your existing installation using the patch listed below.

Available Patches

If for some reason you cannot upgrade to FishEye 2.4/2.3.7 or Crucible 2.4/2.3.7, you can apply the following patch to fix the vulnerabilities described in this security advisory.

Step 1 of the Patch Procedure: Install the Patch

A patch is available for FishEye/Crucible 2.3.6 only.

The patch addresses the following issue:

- XSS vulnerability in the code metrics plugin (CRUC-4572).
- XSS vulnerability in revision ID parameters on annotated views (CRUC-4641).

1. Shut down FishEye.
2. Back up your FishEye instance.
3. Download the patch, fisheye-2.3.6-security-patch.zip.
4. Expand the zip file into <fisheye_install_dir>, overwriting the existing files.
   The patch will overwrite your 'plugins/bundled-plugins.zip' file as well as some class files.
5. Restart FishEye.

FishEye Security Advisory 2010-06-16

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.

<table>
<thead>
<tr>
<th>Affected FishEye Versions</th>
<th>Fix Availability</th>
<th>More Details</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All versions up to and including 2.3.2.</td>
<td>2.3.3 update, also available as patches for 2.3.2 and 2.2.3.</td>
<td>This vulnerability allows a motivated attacker to call remote code on the host server.</td>
<td>Critical</td>
</tr>
</tbody>
</table>

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](https://www.opensymphony.com/xwork/) 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**

<table>
<thead>
<tr>
<th>File</th>
<th>FishEye / Crucible Version</th>
<th>Release Date</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>fisheye-crucible-2.3.2-patch1.zip</td>
<td>2.3.2</td>
<td>16th June, 2010</td>
<td>6fe98db821a6d26f26907688af2ccd84</td>
</tr>
</tbody>
</table>

**Patch for FishEye / Crucible 2.2.3**

<table>
<thead>
<tr>
<th>File</th>
<th>FishEye / Crucible Version</th>
<th>Release Date</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>fisheye-crucible-2.2.3-patch1.zip</td>
<td>2.2.3</td>
<td>16th June, 2010</td>
<td>6fe98db821a6d26f26907688af2ccd84</td>
</tr>
</tbody>
</table>

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.

**FishEye Security Advisory 2010-05-04**

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.

<table>
<thead>
<tr>
<th>Affected FishEye Versions</th>
<th>Fix Availability</th>
<th>More Details</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All versions up to and including 2.2.1</td>
<td>2.2.3 update, also available as patches for certain versions, listed on this page.</td>
<td>This vulnerability allows a motivated attacker to perform admin actions.</td>
<td>Critical</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Affected FishEye Versions</th>
<th>Fix Availability</th>
<th>More Details</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All versions up to and including 2.2.1</td>
<td>2.2.3 only</td>
<td>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.</td>
<td>Critical</td>
</tr>
</tbody>
</table>

**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.
# 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

<table>
<thead>
<tr>
<th>File</th>
<th>FishEye / Crucible Version</th>
<th>Release Date</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>fisheye-crucible-2.1.4-patch1.zip</td>
<td>2.1.4</td>
<td>4th May, 2010</td>
<td>6062fa2e1ad937295273 57fb97b0d2ea</td>
</tr>
</tbody>
</table>
Patch for FishEye / Crucible 2.0.6

<table>
<thead>
<tr>
<th>File</th>
<th>FishEye / Crucible Version</th>
<th>Release Date</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>fisheye-crucible-2.0.6-patch1.zip</td>
<td>2.0.6</td>
<td>4th May, 2010</td>
<td>6aae75e2a5308121887b f9532473cf75</td>
</tr>
</tbody>
</table>

Patch for FishEye 1.6.6

<table>
<thead>
<tr>
<th>File</th>
<th>FishEye Version</th>
<th>Release Date</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>fisheye-1.6.6-patch1.zip</td>
<td>1.6.6</td>
<td>4th May, 2010</td>
<td>210ef3358aff83861733f8 f22d331d7e</td>
</tr>
</tbody>
</table>

Patch for Crucible 1.6.6

<table>
<thead>
<tr>
<th>File</th>
<th>Crucible Version</th>
<th>Release Date</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>crucible-1.6.6-patch1.zip</td>
<td>1.6.6</td>
<td>4th May, 2010</td>
<td>48e8e8ada0dd93fc8671 459051df1120</td>
</tr>
</tbody>
</table>

To acquire all of the fixes on this page, upgrade to FishEye 2.2.3, which you can download from the download centre.

FishEye FAQ

Answers to frequently asked questions about configuring and using FishEye.

- **Top Evaluator Questions**
  - How do I fix problems with indexing my repository?
  - How do I setup JIRA integration?
  - How do I setup LDAP or external user management?
  - How do I speed up slow CVS updates?
  - How do I start FishEye as a Windows service?
  - How do I view changesets and diffs?
  - How is FishEye licensed?
  - What kind of search capabilities does FishEye have?
  - What programming languages are supported?

- **CVS FAQ**
  - How does FishEye calculate CVS changesets?
  - How is changeset ancestry implemented for CVS?

- **Example EyeQQL Queries**
  - How do find changes made to a branch after a given tag?
  - How do I filter results?
  - How do I find changes between two versions, showing separate histories?
  - How do I find changes made between two version numbers?
  - How do I find commits without comments?
  - How do I find files on a branch, excluding deleted files?
  - How do I find files removed from a given branch?
  - How do I find revisions made by one author between versions?
  - How do I select the most recent revisions in a given branch?
  - How do I show all changesets which do not have reviews?
FishEye Developer FAQ

General FAQ
- About database encoding
- About the Lines of Code Metric
- Cannot View Lines of Code Information in FishEye
- Finding your Server ID
- How Do I Archive a Branch within Perforce
- How do I Avoid Long Reindex Times When I Upgrade?
- Mercurial Known Issues
- Ordering of Branches Important When Visualising Git Changesets
- Perforce Changesets and Branches
- What SCM systems are supported by FishEye?
- Automating Administrative Actions in Fisheye

Installation & Configuration FAQ
- Can I deploy FishEye or Crucible as a WAR?
- Does Fisheye support SSL (HTTPS)?
- Improve FishEye scan performance
- Migrating FishEye Between Servers
- Setting up a CVS mirror with rsync
- What are the FishEye System Requirements?
- How to reset the Administration Page password in Fisheye or Crucible
- How Do I Configure an Outbound Proxy Server for FishEye
- How to remove Crucible from FishEye 2.x or later
- How to run Fisheye or Crucible on startup on Mac OS X

Integration FAQ
- How do I disable the FishEye tab panel for non-code projects?
- How do I enable debug logging for the JIRA FishEye plugin?
- How do I uninstall the JIRA FishEye plugin?
- How is the Crucible review tab panel for the FishEye JIRA plugin populated?
- What do I do if I discover a bug with the JIRA FishEye plugin?

Licensing FAQ
- Are anonymous users counted towards FishEye's licence limits?
- Restrictions on FishEye Starter Licenses

Repository Management FAQ
- Setting the root location for internal repositories
- Migrating repositories to a changed root location
- Repairing a broken hierarchy of forks
- Permanent authentication for Git repositories over HTTP(S)

Subversion FAQ
- Configuring Start Revision based on date
- Errors 'SEVERE assert' or 'Checksum mismatch'
- FishEye fails to connect to the Subversion repository after a short time of successful operation.
- How can FishEye help with merging of branches in Subversion?
- Subversion Changeset Parents and Branches
- SVN Authentication Issues
- What are Subversion root and tag branches?
- Why do I need to describe the branch and tag structure for Subversion repositories?
- Why don't all my tags show up in FishEye?
• Support Policies
  • Bug Fixing Policy
  • How to Report a Security Issue
  • New Features Policy
  • Patch Policy
  • Security Advisory Publishing Policy
  • Security Patch Policy
  • Severity Levels for Security Issues
• Troubleshooting
  • After I commit a change to my CVS repository, it takes a long time before it appears in FishEye.
  • FishEye freezes unexpectedly
  • Generating a Thread Dump Externally
  • I have installed FishEye, and the initial scan is taking a long time. Is this normal?
  • I have installed FishEye, but there is no data in the Changelog.
  • Initial scan and page loads are slow on Subversion
  • It seems that FishEye's HTTP Header is Too Small
  • JIRA Integration Issues
  • Message 'org.tigris.subversion.javahl.ClientException svn Java heap space'
  • On my Red Hat Linux system, after running for several days FishEye freezes and does not accept any more connections.
  • Problems with very long comments and MySQL migration
  • URLs with encoded slashes don't work, especially in Author constraints
• Contributing to the FishEye Documentation
• Tips of the Trade
• FishEye Documentation in Other Languages
• FishEye Resources

Do you have a question, or need help with FishEye? Please create a support request.

CVS FAQ

FishEye CVS FAQ

• How does FishEye calculate CVS changesets? — FishEye's goal is to allow changesets to be seen as a consistent stream of atomic commits. Revisions are collated into the same changeset provided that:
  • They have the same commit comment.
  • They are by the same author.
  • They are on the same branch.
  • The changeset does not span more than 10 minutes.
  • The same file does not appear in a changeset more than once.

How is changeset ancestry implemented for CVS?

About Changeset Ancestry in FishEye

When FishEye indexes a CVS repository, it synthesizes a changeset identifier to group file-level changes into a single consistent changeset. The grouping is described in this FAQ: How does FishEye calculate CVS changesets?
Changeset ancestry was added in FishEye 2.6. Changeset ancestry refers to the linking of a changeset to a preceding/parent changeset(s). This allows you to view the development progress of your repository using the Commit Graph (see Viewing the Commit Graph for a Repository).

Changeset Ancestry for CVS

For CVS repositories, changeset ancestry is implemented, as follows:

- For all but the first change on a branch, FishEye chooses the most recent change on that branch as the parent changeset.
- For the first change on a branch, FishEye examines the branchpoints of all files in the branch and chooses the latest changeset that affected any such files as the parent changeset.

This approach ensures that a branch, whose first change is to a file which is very old, is not considered to have been branched at the time that file was last changed. It is considered to be branched at the last change to the repository instead.

Example EyeQL Queries

<table>
<thead>
<tr>
<th>EyeQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>How do find changes made to a branch after a given tag?</td>
</tr>
<tr>
<td>How do I filter results?</td>
</tr>
<tr>
<td>How do I find changes between two versions, showing separate histories?</td>
</tr>
<tr>
<td>How do I find changes made between two version numbers?</td>
</tr>
<tr>
<td>How do I find commits without comments?</td>
</tr>
<tr>
<td>How do I find files on a branch, excluding deleted files?</td>
</tr>
<tr>
<td>How do I find files removed from a given branch?</td>
</tr>
<tr>
<td>How do I find revisions made by one author between versions?</td>
</tr>
<tr>
<td>How do I select the most recent revisions in a given branch?</td>
</tr>
<tr>
<td>How do I show all changesets which do not have reviews?</td>
</tr>
</tbody>
</table>

For more information on using EyeQL, see the Reference guide.

How do find changes made to a branch after a given tag?

Find changes made to Ant 1.5.x after 1.5 FINAL:

```
select revisions where on branch ANT_15_BRANCH and after tag ANT_MAIN_15FINAL group by changeset
```

How do I filter results?

This query, finds files removed on the Ant 1.5 branch, but just returns the person and time the files were deleted:

```
select revisions where modified on branch ANT_15_BRANCH and is dead return path, author, date
```

How do I find changes between two versions, showing separate histories?

As above, but show the history of each file separately:
How do I find changes made between two version numbers?
Find changes made between Ant 1.5 and 1.5.1:

```
select revisions where between tags (ANT_MAIN_15FINAL, ANT_151_FINAL] group by changeset
```

How do I find commits without comments?
Using the Advanced Search and EyeQL you can find commits that do not have comments using the following query:

```
select revisions from dir / where comment = "" group by changeset
```

How do I find files on a branch, excluding deleted files?
Find files on branch and exclude delete files:

```
select revisions where modified on branch ANT_15_BRANCH and not is deleted group by changeset
```

How do I find files removed from a given branch?
Find files removed on the Ant 1.5 branch:

```
select revisions where modified on branch ANT_15_BRANCH and is dead group by changeset
```

How do I find revisions made by one author between versions?
Find changes made by conor to Ant 1.5.x since 1.5.0:

```
select revisions where between tags (ANT_MAIN_15FINAL, ANT_154] and author = conor group by changeset
```

How do I select the most recent revisions in a given branch?
Find Java files that are tagged ANT_151_FINAL and are head on the ANT_15_BRANCH: (i.e. files that haven’t changed in 1.5.x since 1.5.1)
How do I show all changesets which do not have reviews?

The following query will return any changesets that have not been reviewed.

```sql
select revisions where (not in any review)
```

FishEye Developer FAQ

This page contains answers to frequently asked questions posed by FishEye developers. For detailed information about developing in FishEye, see the Fisheye Developer documentation.

Feel free to comment, make submissions, or pose your own question on FishEye Development here.

- **Q:** I'm getting the error "API access is disabled" as a response from http://fisheye/api/rest/repositories on my installation. How do I enable the API as a Fisheye administrator?
  - **A:** A toggle to enable the API under "Server Settings" in the web admin interface existed in versions prior to 2.9 (see Configuring the FishEye web server for more details).
  - See the FishEye 2.9 Release Notes

- **Q:** Is there any way to return unique results from an EyeQL query?
  - **A:** It is not currently possible to return unique results.
  - An improvement request exists: FE-1136. Your vote and comments on that issue are appreciated.

General FAQ

### FishEye General FAQ

- About database encoding
- About the Lines of Code Metric
- Cannot View Lines of Code Information in FishEye
- Finding your Server ID
- How Do I Archive a Branch within Perforce
- How do I Avoid Long Reindex Times When I Upgrade? — If reindexing your repository takes longer than you can allow, you can use a temporary copy of your repository and FishEye instance to reduce downtime during the reindexing process.
- Mercurial Known Issues
- Ordering of Branches Important When Visualising Git Changesets
- Perforce Changesets and Branches
- What SCM systems are supported by FishEye?
- Automating Administrative Actions in Fisheye

### About database encoding

It is possible to have files in your repository whose names differ only in case, e.g. `Foo.java` and `foo.java`. Hence, your database will need to use rules for comparing string values which recognise that upper and lower
case letters are different, that is, the database should use 'case sensitive collation'.

If your database was originally configured to use case-insensitive and/or non-UTF8 collation, FishEye will display the following message at the bottom of your screen:

“Your database is not using a case sensitive UTF8 encoding for character fields.”

The following sections provide instructions for changing your database collation for each database type supported by FishEye and Crucible.

On this page:
- MySQL
- Oracle
- PostgreSQL
- SQL Server

Related pages:
- Migrating to an external database

MySQL

⚠️ Please take a backup of your database before changing its collation.

To change your collation to utf8_bin you need to change your database's default collation, but as this only affects newly created tables you will also need to change the collation on the table for which case sensitivity is critical.

Change your database’s collation

Use the ALTER DATABASE command, as follows:

```
alter database character set utf8 collate utf8_bin;
```

Change collation for the CRU_STORED_PATH table

Use the ALTER TABLE command, as follows:

```
alter table cru_stored_path convert to character set utf8 collate utf8_bin;
```

Oracle

Oracle collation encoding must be configured when installing the database server. It cannot be configured on a per database level. When installing Oracle, you should select the AL32UTF8 encoding.

PostgreSQL

⚠️ Please take a backup of your database before changing its collation.

If you have created your PostgreSQL database with the incorrect encoding, you will need to dump your database, drop it, create a new database with the correct encoding and reload your data.
You can do this using the standard database migration procedure — instead of migrating from HSQLDB to PostgreSQL, you migrate from a PostgreSQL database with the incorrect encoding to one created with the correct encoding.

SQL Server

⚠️ Please take a backup of your database before changing its collation.

Unfortunately, changing the database collation for an existing SQL Server database (even using the ALTER DATABASE ... COLLATE... statement) does not change the collation for existing objects stored in the database. See http://blogs.msdn.com/b/qingsongyao/archive/2011/04/04/do-not-alter-database-collation-in-your-server.aspx for an explanation of this.

The recommended route for changing the collation for SQL Server is to migrate to a new database that has the correct collation configuration. You can do this using the standard database migration procedure — instead of migrating from HSQLDB to SQL Server, you migrate from a SQL Server database with the incorrect collation to one created with the correct collation.

The correct collation to use when you create the new SQL Server database is Latin1_General_CS_AS.

About the Lines of Code Metric

This page contains information about the Lines of Code metric and how it is processed and represented by FishEye.

On this page:
- Definition
- Disadvantages
- LOC in FishEye
- User-Specific LOC

Definition

Lines of Code or LOC (also known as Source Lines of Code - SLOC) is a quantitative measurement in computer programming for files that contains code from a computer programming language, in text form. The number of lines indicates the size of a given file and gives some indication of the work involved.

LOC is literally the count of the number of lines of text in a file or directory. In FishEye, blank lines and comment lines are counted toward the total lines of code.

LOC for a file/directory is the total number of lines in the relevant files, while LOC for an author is the number of lines blamed on that author. Neither of these should ever be less than zero. However, the change in LOC over a period of time can be negative if there was a net reduction in the LOC over the period.

Disadvantages

While it can be useful, LOC has some well documented disadvantages. Keep these disadvantages and limitations in mind when using LOC in your work environment.

In addition, the nature of branching in SCM applications means that calculating a LOC value for a whole project is not possible. A naive summation of the LOC of all the branches will give a meaningless number that jumps every time a branch is copied to create a new branch. Thus, in FishEye we usually look at the LOC of the trunk, unless we can infer from the context that another branch is more appropriate.

LOC in FishEye

FishEye calculates the LOC for the trunk only. For SVN repositories it can calculate LOC for a branch if it is "trick
ed” to see the branch as part of the trunk. FishEye also calculates the LOC for each user, unless that facility is turned off in the repository (see Store Diff Info). The LOC count will include all files except those identified by the SCM as binary.

FishEye presents LOC data as charts of the change in LOC over time, and as informational statistics in various places:

- **Chart pages**
  The best way to explore the evolution of LOC in your project is the LOC chart report where you can easily filter the LOC by branch, author, file extension and date range. Here you can investigate what caused a particular spike in the LOC charts, or find the user whom has the most lines of code blamed on them and how this has changed over time.

- **Repository-specific activity pages**
  These show trunk LOC statistics for the repository, limited to the directory being viewed and its subdirectories. The LOC charts show the LOC for the directory, using trunk LOC unless the directory can be identified as a branch.

- **User pages**
  Here, the statistics pane in the sidebar shows the trunk LOC blamed on the user for the all repositories that have user-specific LOC enabled. The chart shows the trunk LOC from all the repositories that the user has contributed to.

- **The global User List page**
  This shows the trunk LOC for all users from the repositories that have user-specific LOC enabled. Repository-specific user lists (in repositories that have user-specific LOC enabled) show the trunk LOC for the users and committers, limited to the directory being viewed and its subdirectories.

- **Project pages**
  This shows a chart of the LOC for all associated repository paths, and statistics include the trunk LOC for those paths.

**User-Specific LOC**

The evolution of user-specific LOC over subsequent commits can appear at first glance to be counter-intuitive. It is important to keep in mind that the LOC for a given user is the number of lines in the repository that were last changed by them (as calculated by Fisheye).

A couple of simple examples:

- Alice adds a files with 30 lines to the SCM. Her LOC for this file is now 30. She then edits the file, deletes 10 lines and adds 20 (+20 -10). Her LOC is now 40, as is the LOC of the file.

- Alice adds a files with 30 lines to the SCM. Her LOC for this file is now 30. Now Bob edits the file, deletes 10 lines and adds 20 (+20 -10). Alice now has LOC of 20, because Bob deleted 10 lines that were blamed on her, and Bob has LOC of 20, from the 20 lines he added. The total LOC is still 40.

A user can have LOC on a branch that they have never committed on, if something that has been blamed on them is copied. For example, a developer may have never committed to a particular branch, but FishEye may still report a lot of LOC for them in that area.

One current limitation of FishEye’s user-specific LOC calculation is the handling of merging. For example, if a file has been changed on both trunk and branch, and the changes made on the branch are merged to trunk, the changes made on branch will generally be blamed on the person who did the merge; not the person who made the change.

**Cannot View Lines of Code Information in FishEye**

**Symptoms**
The LOC (Lines of Code) information in FishEye cannot be seen, for example in charts or when viewing the statistics for a user.

ℹ️ See About the Lines of Code Metric for more information about the usage of the LOC (Lines of Code) metric in FishEye.

**Cause**

There are four possible causes:

- LOC data will not be shown for users if the Store diff info setting is disabled. If a page is being viewed in FishEye that relates to a particular user or committer, and the Store Diff Info setting is disabled, no LOC information for the user will be visible.
- LOC data is currently not supported for Mercurial repositories.
- LOC data is currently not supported for Git repositories.
- The SVN repository is indexing branches only.

**Resolution**

Cannot view LOC information for specific users or committers:

- Enable the Store Diff Info setting for the repository in Administration > Repository Settings > Repositories, click on the repository name, and then "SCM Details". A full re-index needs to be performed on the repository after enabling this setting for FishEye to collect the diff information for all revisions in the repository. Please note that the Store Diff Info setting is always enabled for CVS repositories.

Cannot view LOC information for Mercurial repositories:

- There is an outstanding feature request for LOC support in Mercurial.

Cannot view LOC information for Git repositories:

- There is an outstanding feature request for LOC support in Git.

The SVN repository is indexing branches only:

- FishEye can calculate LOC for a branch if it is "tricked" to see the branch as part of the trunk.

**Finding your Server ID**

Your Server ID can be found in your FishEye administration console, as described below.

**To find your Server ID:**

1. Navigate to FishEye's administration console.
2. Click 'Sys-Info/Support' under the 'System settings' section. The 'Server ID' for your FishEye server will be displayed in the 'License' section.
   🔄 The Server ID should match the one set for your license. You can check this at http://my.atlassian.com

**How Do I Archive a Branch within Perforce**

In SVN, a branch exists as a separate directory. However in Perforce, files are given a label to identify them as belonging to the branch. Thus it may not be possible to download the branch as a tarball via FishEye.

You may be able to download the branch as a tarball, depending on your structure:

⚠️ If it is not a single folder, then it is not possible to download the tarball in your perforce repository.

1. In FishEye, navigate to your perforce repository.
2. In the Constraint section on the left, select the branch. This will return the directories that belong to that branch.
3. If it is one single folder, download the tarball of it. Under **constraint and sub directories**, there is a panel tarball giving options on how to download the directory.

**How do I Avoid Long Reindex Times When I Upgrade?**

**Mitigating Lengthy Reindex Times**

If reindexing your repository takes longer than you can allow, you can use a temporary copy of your repository and FishEye instance to reduce downtime during the reindexing process.

> **Warning:** Most upgrades (even major ones) do not require a reindex. If an upgrade is required, this will always be explicitly mentioned in the Upgrade Guide for that release.

**Reindexing With a Temporary Copy of Your FishEye Instance**

To reindex a temporary copy of your FishEye instance:

1. Make a copy of your FishEye instance to another server. See ‘How to Make a Temporary Copy of Your FishEye Instance’ below for instructions.
2. Upgrade the temporary FishEye, then start it up, connected to your repository. It will automatically begin the scanning process.
   
   > **Info:** If you are concerned about the repository being overloaded by the scanning process, you can make a copy of that as well. See ‘How to Make a Temporary Copy of Your Repository’ below for instructions. If you do that, you must edit the `config.xml` of your temporary FishEye instance to point to your temporary repository.
3. The copied instance will run its course without affecting your production instance.
   
   a. Shutdown both your servers completely.
   b. Make a backup of your `FISHEYE_INST` directory.
   c. Replace the `FISHEYE_INST/var/cache` directory on live FishEye with the `FISHEYE_INST/var/cache` from your test server.
   d. Download the latest FishEye/Crucible from the [Atlassian downloads](https://confluence.atlassian.com/display/FE/Atlassian+downloads).
   e. Follow the instructions in the Upgrade Guide to upgrade to the new version.
4. The scan of the temporary FishEye instance (and repository, if you copied that also) is complete. You’re now free to delete the temporary copy(s).

**How to Make a Temporary Copy of Your FishEye Instance?**

To make a copy of your FishEye instance, follow the instructions for [Migrating FishEye Between Servers](https://confluence.atlassian.com/display/FE/Migrating+FishEye+Between+Servers).

**How to Make a Temporary Copy of Your Repository?**

To make a copy of your repository use `rsync` (for CVS repositories in the Linux environment) or `svnsync` documentation (for Subversion only).

**How to Reindex a Single Repository on a Test Server?**

If you need to reindex your repository on your production system but don’t want to burden your production server, carry out the following steps:

1. Install another instance of FishEye on a test server (the same FishEye version as the one you are using).
2. Add a repository to FishEye with the exact same name and details as that referenced by the production server.
3. Let it finish indexing. Go to **Administration > View Repository List > Stop** (shown next to the name of your repository) and disable on both production and test.
4. Copy over the FISHEYE_INST/var/cache/REPO directory on the production FishEye with the FISHEYE_INST/var/cache/REPO directory from the test server.

5. Trigger a review revision data re-index: **Administration > Repository > Maintenance > Review-Revision Data Index.**

**i** For this procedure, neither server needs to be shut down.

**Mercurial Known Issues**

- **CRUC-3474:** If a file is removed and then another file is copied or moved over the same file within one commit, the ancestor revision is miscalculated and can result in errors in "diff to previous".
- **CRUC-3470:** Permission changes (and prop changes in repos converted from svn) may result in revisions that have no ancestors - subsequent changes will consider it's parent revision to be their parent revision.
- **CRUC-3468:** Scanning repositories converted from svn (especially using hgsubversion) can result in commits that take a long time to scan (due to the changes produced by merges from other branches).

**Ordering of Branches Important When Visualising Git Changesets**

FishEye 2.6 introduced the repository commit graph. The commit graph allows you to visualise changesets in their branches by showing them in configurable branch "swimlanes". One of the ways in which you can configure the commit graph is by reordering the swimlanes. Reordering swimlanes is useful for non-Git repositories, if you want to show branches in a certain order. However, ordering swimlanes is vital for Git repositories, as it is the only way of determining which branch a commit is displayed in, when a commit belongs to multiple branches.

**Git Branches and Changesets in FishEye**

Before considering how Git repositories are visualised in the commit graph, it is important to understand how FishEye relates Git changesets to branches.

In FishEye 2.6 and later, FishEye considers the ancestry of a Git changeset when determining which branch it is a part of. Branches can effectively be considered as pointers to changesets. Hence, merging and branching can change the branches that a changeset is considered part of.

For example, if a branch 'fisheye-2.6' is merged back to the 'master' branch, then all changesets that were seen as part of the 'fisheye-2.6' branch only will also be considered to be part of the 'master' (e.g. the changeset will be seen as part of 'master' and 'fisheye-2.6' in the activity stream).

**Viewing Git Changesets and Branches in the Commit Graph**

The previous section describes how a changeset can be associated with multiple branches, due to its ancestry. Instead of showing the changeset in every branch swimlane on the commit graph, FishEye represents these changesets as described below.

When you view the commit graph for a Git repository, FishEye works from the leftmost swimlane to the right doing the following:

- For each swimlane, FishEye checks if the commit is in that branch. If the commit is in the branch, a dot is shown representing the commit.
- If the commit is not in the branch, the dot for the commit is moved to the next column on the right.

For example, if the 'master' swimlane is to the left of another swimlane, e.g. 'fisheye-2.6' branch, there will be no changesets shown in the 'fisheye-2.6' swimlane, as all the commits will be picked up in the 'master' swimlane. However, if you move the 'fisheye-2.6' swimlane to the left of the 'master' swimlane, it will pick up all of the FishEye 2.6 commits.

This allows you to visually isolate changesets in the desired branches by reordering the swimlanes. For example, if you want to see the lineage of a branch, 'fisheye-2.6', but not 'fisheye-2.5' after both branches have previously been merged to 'master', you could arrange your swimlanes to 'fisheye-2.6', 'master', 'fisheye-2.5'
from left to right. You will be able to see the 'fisheye-2.6' changesets and where the merge back to 'master' was made. The 'fisheye-2.5' changesets will just be seen as part of the 'master' branch.

Screenshots below: Example of how ordering swimlanes affects the branches that changesets are displayed on (click to view full-size images)

Perforce Changesets and Branches

Why does FishEye say this changeset is on more than one branch? Why does that changeset have multiple parents?

Perforce allows a single changeset to include files on multiple branches, so FishEye marks those changesets as being on all of the branches involved.

When a changeset is on multiple branches FishEye may consider it to have multiple parents from the different branches.

FishEye does not track merges in Perforce, so merges are not shown in the graph.

Changeset branches and parents are only annotations at the changeset level --- the individual file revisions will each only have a single branch and at most one parent.

What SCM systems are supported by FishEye?

To see the list of SCM systems that is supported by FishEye, see Supported platforms.

Automating Administrative Actions in Fisheye

With some command line scripting and a tool like wget, and Live HTTP Headers for firefox you can automate actions. In this example, Fisheye will automatically rescan revision properties of an SVN when the commit message is updated to reference a new JIRA issue.

1. Enable live HTTP headers in firefox, then perform the action you want to perform automatically via the Fisheye Administration UI.
2. In the live HTTP headers window you should see some output similar to the following:
3. The important parts are the URL I’ve highlighted above (http://erdinger.sydney.atlassian.com/fisheye/admin/indexMaint.do) and any GET/POST parameters (startRev=0&endRev=58&rep=2&startRev=0&endRev=58&rep=2&endRev=58&rep=2&startRev=0&endRev=58&rep=2)

4. Now we can construct a script with `wget` to automate this:

```bash
wget --keep-session-cookies --save-cookies cookie.txt http://erdinger.sydney.atlassian.com/fisheye/admin/login.do
--post-data="origUrl=&adminPassword=admin"
wget --load-cookies cookie.txt
```

With that you could generate a `post-revprop-change` hook in `svn` that will update the repositories automatically.

**Installation & Configuration FAQ**
## FishEye Installation & Configuration FAQ

- Can I deploy FishEye or Crucible as a WAR? — Unfortunately FishEye and Crucible can not be deployed as a WAR.
- Does Fisheye support SSL (HTTPS)?
- Improve FishEye scan performance — Background information
- Migrating FishEye Between Servers — This page describes the process for migrating FishEye between servers.
- Setting up a CVS mirror with rsync — In situations where running FishEye on the same server as your CVS repository is not practical or possible, you can use the Linux utility `rsync` to mirror the CVS repository contents onto the FishEye server.
- What are the FishEye System Requirements?
- How to reset the Administration Page password in Fisheye or Crucible
- How Do I Configure an Outbound Proxy Server for FishEye
- How to remove Crucible from FishEye 2.x or later
- How to run Fisheye or Crucible on startup on Mac OS X

## Can I deploy FishEye or Crucible as a WAR?

Unfortunately FishEye and Crucible can not 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 Fisheye support SSL (HTTPS)?

FishEye has built-in SSL support from FishEye 2.4 onwards. Read FishEye SSL configuration for more information.

### Improve FishEye scan performance

This page contains information about improving the performance of FishEye repository scans.

#### Background information

When you add a repository, FishEye needs to perform a once-off scan through the repository to build up its initial index and cache. This scan can take some time. Until this scan is complete, you may find that some data is not displayed. As a guide, FishEye should be able to process about 100KB-200KB per second on an averaged-size PC. If FishEye is accessing the repository over the network (e.g. over a NFS mount), then you should expect the initial scan to take longer.

#### General improvements

You can increase the speed of your scans using the following options:

- If your repository is non-local, set up a local repository mirror on the FishEye server. This will provide a major speed boost for anyone scanning a repository across a network.
- Exclude unused file types, unused directories and specific large files from FishEye.

#### Improve update performance during initial scan

One option is break large repositories into multiple smaller repositories. While this technique will not improve the overall initial scan time, it allows for all fully scanned repositories to be updated while the initial scan is still being performed on those remaining.

In FishEye 1.3.4 and later, the initial and incremental scans happen in separate, single threads. So, splitting the repositories will allow incremental scans to run concurrently alongside the initial scans. You may also wish to split projects into separate repositories, since permissions in FishEye are applied on a per-repository basis.
Improving initial scan performance for an SVN repository

The http/s protocol has the slowest performance during the initial scan. The svn protocol (svn://) is faster and the file protocol (file:///) is the fastest.
Therefore if you find your initial scan takes an extended amount of time (more than a day or two), you should consider switching over from the http/s protocol to the svn or file protocol to define the location of your SVN repository. (Use svnsync to mirror the repository onto the fisheye server, so that you can access it with the file protocol.)

E.g. Switch from
https://example.com/svn/project/
to
svn://example.com/svn/project/
or
file:///home/user/some/location/svn/project

In order for SVN protocol to work you need to have set up an svnserv based server.
More information on how to troubleshoot SVN indexing related issues can be found here.

Performance support

If you have implemented at least one of the suggestions above but are still experiencing slow performance, ask us for help:

1. First read the Tuning FishEye performance document.
2. Turn on debug logging using the command line debug flag.
3. Allow FishEye to continue its initial scan overnight.
4. Create a new support request in the FishEye project, including your server environment and log files with the problem description.

Migrating FishEye Between Servers

This page describes the process for migrating FishEye between servers.

If you have defined the FISHEYE_INST Environment Variable, then upgrades and migrations of your FishEye instance will be relatively simpler.

If you have defined FISHEYE_INST

1) Shut down your current FishEye server completely.
2) Copy the FISHEYE_INST directory to your destination server.
3) Copy and set up all of your Environment Variables from your source server to your destination server (remembering to set up your FISHEYE_INST directory to point to the location where you copied the data to in Step 2).
4) Install FishEye on your destination server.
5) Start FishEye. It should pick up your environment variables, and from that access your FISHEYE_INST directory, which contains your configuration.

If you have not defined FISHEYE_INST but would like to set it up

1) Shut down your current FishEye server completely.
2) Copy the following three items into to a new folder on your destination server (for example, fisheye_inst):
   • <FishEye home directory>/config.xml
   • <FishEye home directory>/var
3) Copy and set up all of your Environment Variables from your source server to your destination server. In addition to this, set up the FISHEYE_INST environment variable as follows, replacing the /path/to/fisheye_inst with the fully qualified path to the fisheye_inst folder you set up in Step 2:

```bash
export FISHEYE_INST=/path/to/fisheye_inst
```

4) Install FishEye on your destination server.

5) Start FishEye. It should pick up your environment variables, and from that access your FISHEYE_INST directory, which contains your configuration.

### External Databases

Please note that the steps above will only move your FishEye configuration data. If you are using an external database, that configuration will assume that your database is accessible from the new server.

However if you want to make a copy of your server (You may want to create a copy, if you want a development server), then you will need to make a copy of your external database, and edit the FISHEYE_HOME/config.xml and update the database details to point to the copied database.

If you wish to migrate both FishEye and your external database to a different server, see Backing up and restoring FishEye data for more information.

### Setting up a CVS mirror with rsync

In situations where running FishEye on the same server as your CVS repository is not practical or possible, you can use the Linux utility 'rsync' to mirror the CVS repository contents onto the FishEye server. This is possible because CVS data is stored in a reasonably simple form in the file system.

We recommend this to achieve best performance when FishEye and CVS cannot be hosted on the same machine.

This workaround requires the ability to SSH into both machines. Linux and Mac OS X operating systems have rsync built in. For Windows, you will need to install rsync.

*Diagram: A scenario where rsync is required*
To set up a CVS mirror with rsync:

1. You will need to set up a local directory on the FishEye server for the mirrored CVS content, ensuring that this server has ample disk space to store the current CVS database and any future space requirements.
2. We will refer to the CVS instance (on your CVS Server,) as <CVS_HOME> and the new 'mirror directory' (on your FishEye server) as <MIRROR_HOME>.
3. Type the following rsync command on the console at the command-line of the FishEye server:

   ```
   rsync --backup <CVS_HOME> <MIRROR_HOME>
   ```

   A real-world example would look something like this:

   ```
   rsync --backup user@cvs_server:/CVS_server/path/to/instance/datastore/FishEye/cvs-mirror
   ```

4. Schedule the rsync command to run regularly with a cron job. Running hourly is a good default interval. Under Windows, use a native task scheduler.
5. With the cron job active, you will have established rsync to run an hourly comparison of the two directories and copy any changes across to the mirror directory as they occur. Note that running the rsync process will impact the FishEye server's performance (and also the CVS server's) to a certain degree.
6. In the FishEye admin interface, add the local 'mirror directory' as a new CVS repository and run the initial scan. As this is local data on the same file system, FishEye's scanning of this data will be optimal.
7. Adjust the FishEye Updater Full Scan period to one hour (the default is 15 minutes).
8. The rsync configuration is now complete. Monitor the disk space on both servers to ensure there is
adequate headroom for the mirroring process.

For more information on the syntax for rsync, visit the rsync home page.

**What are the FishEye System Requirements?**

**Visit the FishEye Supported platforms.**

**How to reset the Administration Page password in FishEye or Crucible**

If you have forgotten or misplaced the password for the Admin page (http://<FECRU_URL>:<FECRU_PORT>/admin), you can reset it manually.

To manually reset the admin password, please edit your FISHEYE_INST/config.xml file (make a backup as well before editing).

You will see something like:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<config control-bind="127.0.0.1:8059" version="1.0"
admin-hash="352353256326369233A801FC3">
```

To reset the password to "admin", please change the admin-hash value so that it appears as admin-hash="21232F297A57A5A743894A0E4A801FC3"

```xml
<?xml version="1.0" encoding="UTF-8"?>
<config control-bind="127.0.0.1:8059" version="1.0"
admin-hash="21232F297A57A5A743894A0E4A801FC3">
```

Restart FishEye for this to take effect. You should now be able to access the FishEye Admin page (http://<FECRU_URL>:<FECRU_PORT>/admin) with the password "admin". Please change this password immediately from the Admin area: click Change Admin password (under "Security Settings").

**See also** Best practices for FishEye configuration.

**How Do I Configure an Outbound Proxy Server for FishEye**

The Java Virtual Machine provides support for outbound proxy servers. To take advantage of this some additional parameters need to be passed to the JVM via the FISHEYE_OPTS environment variable:

```bash
export FISHEYE_OPTS="-Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=8080 -Dhttp.nonProxyHosts=*.foo.com=localhost -Dhttp.proxyUser=username -Dhttp.proxyPassword=password"
```

See Environment variables for instructions on how to set these parameters.

**How to remove Crucible from FishEye 2.x or later**

☐ Use the same process to remove FishEye and keep Crucible; just remove the FishEye license instead.

Want to know What happens if I decide to stop using FishEye with Crucible?

To remove Crucible from a FishEye installation:

1. Go to the Admin area (choose Administration from your user menu) and click System Info (under "System Settings").
2. Click Edit License (in the "License" section).
3. Remove the Crucible license from this screen:

4. Click **Update** to save. Crucible will now be disabled and you will no longer see any reference to it in the application.

How to run Fisheye or Crucible on startup on Mac OS X

This article is only provided as a guide and has only been tested on Mac OS X 10.5

- launchd does not provide support for service dependencies so if you are using an external database, this may not work for you. Fisheye assumes the database is available when it starts, and the startup scripts will not wait for the database to become available.

You need to create a .plist file to create items that will start at boot time. Please refer to the following page for details:

- Here is an example .plist that should work for Fisheye/Crucible:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.atlassian.fisheye</string>
  <key>ProgramArguments</key>
  <array>
    <string>/path/to/FISHEYE_HOME/bin/run.sh</string>
  </array>
  <key>OnDemand</key>
  <false/>
  <key>RunAtLoad</key>
  <true/>
  <key>StandardOutPath</key>
  <string>/path/to/FISHEYE_INST/var/log/fisheye.out</string>
  <key>StandardErrorPath</key>
  <string>/path/to/FISHEYE_INST/var/log/fisheye.out</string>
  <key>EnvironmentVariables</key>
  <dict>
    <key>FISHEYE_INST</key>
    <string>/path/to/FISHEYE_INST/</string>
    <key>FISHEYE_OPTS</key>
    <string>-Xms512m -Xmx1024m -XX:MaxPermSize=128m</string>
  </dict>
</dict>
</plist>

Customise the /path/to/FISHEYE_INST/ and /path/to/FISHEYE_HOME/ with the FISHEYE_INST and FISHEYE_HOME directories respectively and make any required modifications to FISHEYE_OPTS. Save the file as com.atlassian.fisheye.plist and then try and load it with:

```
[amyers@erdinger:fecru-2.1.3]$ launchctl load com.atlassian.fisheye.plist
[amyers@erdinger:fecru-2.1.3]$ launchctl start com.atlassian.fisheye
```

Fisheye should now start up and you should be able to access it via your web browser.

**Integration FAQ**

The following FAQs relate to integrating FishEye with other Atlassian applications. You may also wish to refer to Troubleshooting JIRA FishEye Plugin in the FishEye Knowledge Base.

- How do I disable the FishEye tab panel for non-code projects?
- How do I enable debug logging for the JIRA FishEye plugin?
- How do I uninstall the JIRA FishEye plugin?
- How is the Crucible review tab panel for the FishEye JIRA plugin populated?
- What do I do if I discover a bug with the JIRA FishEye plugin?

**How do I disable the FishEye tab panel for non-code projects?**

By removing all users and groups from the View Issue Source Tab permission in your project's permission scheme, the Source and Reviews tabs will be effectively disabled for that project.

**How do I enable debug logging for the JIRA FishEye plugin?**

For Plugin versions prior to 3.0

You can enable DEBUG logging for the JIRA FishEye plugin temporarily (until JIRA is shutdown) by accessing
the following URL in a web browser when logged in as an administrator:

```
http://YOUR-JIRA-BASE-URL/secure/admin/ConfigureLogging.jspa?loggerName=com.atlassian.jira.ext.fisheye&levelName=DEBUG
```

To enable debug logging across JIRA restarts, update the **CLASS-SPECIFIC LOGGING LEVELS** section of your `WEB-INF/classes/log4j.properties` file by inserting the following:

```
log4j.category.com.atlassian.jira.ext.fisheye = DEBUG, console, filelog
log4j.additivity.com.atlassian.jira.ext.fisheye = false
```

and restarting JIRA.

**For Plugin versions 3.0+**

You can enable DEBUG logging for the JIRA FishEye plugin temporarily (until JIRA is shutdown) by accessing the following URL in a web browser when logged in as an administrator:

```
http://YOUR-JIRA-BASE-URL/secure/admin/ConfigureLogging.jspa?loggerName=com.atlassian.jirafisheyeplugin&levelName=DEBUG
```

To enable debug logging across JIRA restarts, update the **CLASS-SPECIFIC LOGGING LEVELS** section of your `WEB-INF/classes/log4j.properties` file by inserting the following:

```
log4j.category.com.atlassian.jirafisheyeplugin = DEBUG, console, filelog
log4j.additivity.com.atlassian.jirafisheyeplugin = false
```

and restarting JIRA.

**How do I uninstall the JIRA FishEye plugin?**

**Plugin versions prior to 3.0**

To uninstall, simply remove the `jira-fisheye-plugin-*.jar` from your `WEB-INF/lib` directory.

**Plugin version 3.0+**

To uninstall, simply remove any `jira-fisheye-plugin-*.jar` jars from your `WEB-INF/classes/com/atl
assian/jira/plugin/atlassian-bundled-plugins.zip` and `%JIRA_HOME%/plugins/installed-plugins` directory.

**How is the Crucible review tab panel for the FishEye JIRA plugin populated?**

The Crucible 'Reviews' tab is populated differently depending on the version of the FishEye JIRA plugin you are using:

- **Prior to Version 2.3** of the plugin, a review is displayed on the Crucible tab panel if (and only if) it contains a revision that is displayed on the FishEye tab panel. Having the issue key in the description will not automatically link it to the issue.
- **In Version 2.3+**, you can configure the Crucible tab panel to use the method described above and/or search for issue keys in the title/description of reviews.

If you associate reviews with JIRA issues from the Crucible user interface, they will not necessarily show
What do I do if I discover a bug with the JIRA FishEye plugin?

If you need assistance or think you’ve encountered a bug with the JIRA FishEye plugin, please raise an issue in the JIRA FishEye Plugin project.

Licensing FAQ

### Are anonymous users counted towards FishEye’s licence limits?

The short answer is no. If you are using FishEye in your organisation but most users require only anonymous access (that is, you have not set access restrictions on the content in your repositories), then an unlimited number of anonymous users can be accommodated regardless of the FishEye licence you are using.

Users accessing FishEye anonymously are, for all intents and purposes, unlimited users.

However, if your users require permissions and controlled access to specific content in your repositories, then they will need to log in to FishEye. Hence, these users will need to create accounts and will be factored into the licence limit.

### Restrictions on FishEye Starter Licenses

This page explains the limitations of the FishEye Starter license and provides general information about using this license in production.

**On this page:**

- What is a Starter License?
- What are the Starter License restrictions?
- What happens if I exceed the Starter License limits?
  - Evaluate
  - Upgrade
  - Reconfigure your repository
- Frequently Asked Questions

**What is a Starter License?**

Starter licenses are low-cost licenses that allow small teams to make use of Atlassian products (see more information). FishEye Starter licenses were introduced with the release of FishEye 2.0.5 (October 2009).

**What are the Starter License restrictions?**

**FishEye Starter Licenses** are restricted to no more than:

- 10 users in FishEye
- 10 committers total per repository
- 5 repositories

**What happens if I exceed the Starter License limits?**

If you have more than five repositories, FishEye will prevent more than five repositories from being enabled at any given time. Administrators can control which five repositories are enabled.
If you exceed more than ten committers in a repository, a warning will appear at the top of pages for the entire system, stating the following:

**NOTE:** This repository, (repository-name) has more than ten committers which exceeds the limits for your Starter license. Indexing has stopped. To fix this, you can 'Evaluate', 'Upgrade' or 'Reconfigure your repository'.

**Screenshot: Warning for Starter License Limits**

The links in this warning will lead you to the following solutions:

**Evaluate**

30-day evaluation licenses are available to allow you to try out FishEye and other Atlassian products. You can select a license that allows more users than your current license.

**Upgrade**

You can upgrade your license at any time (via the Atlassian online store), which will remove the committer and repository limits which apply to the Starter License. Please ensure to restart your repository, after the license upgrade, to ensure the changes are picked up for the new committer limit.

**Reconfigure your repository**

This option lets you configure your repository to remain within the limits of the Starter License. You can take the following actions to reduce the scope of FishEye's indexing:

1. **Change the repository definition to look at a subset of your repository**
   Typically this involves setting the path within your repository that you wish FishEye to index. Read more.

2. **Exclude parts of the repository**
   You can exclude portions of your repository that you are not interested in. Committers that are active in only these areas will not appear in FishEye and not be included in the committer count. Read more.

3. **Set a starting point**
   Some of the FishEye SCM integrations allow you to configure a starting revision from which to start indexing. All commits prior to this starting point are not included in FishEye and do not contribute to the committer count. Read more.

4. **Map Committers**
   If your developers have not correctly configured their committer names for Git or Mercurial, they may have committed with multiple identities. It is then possible to remap these to correct the problem. See the knowledge base article, Git or Hg Repository exceeds number of allowed Committers.

Once you have reconfigured your repository, you will need to re-index the repository, allowing you to remain under the limits of the Starter license.

**Frequently Asked Questions**

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
</table>

Created by Atlassian in 2013. Licensed under a Creative Commons Attribution 2.5 Australia License.
<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>What happens when the 11th unique committer is encountered during indexing?</td>
<td>For all SCMs other than CVS FishEye will index all revisions up to but not including the revision that introduced the 11th committer. Since CVS is indexing is file-by-file based, FishEye will index files until it reaches the committer limit. Remaining files in the repository are not indexed. This means only files which have been indexed will be displayed in changesets and changesets may be incomplete.</td>
</tr>
<tr>
<td>What happens when a FishEye instance with a Starter license is started, using existing indexes with more than five repositories?</td>
<td>FishEye will only start indexing on the first five repositories. An administrator can use admin UI to adjust which repositories are enabled and hence control the five repositories that are started. FishEye should then be restarted.</td>
</tr>
<tr>
<td>What happens when a FishEye instance with a Starter license is started, using existing indexes with one or more repositories with more than ten committers?</td>
<td>FishEye will display all information currently indexed but for each repository that has reached the ten committer limit, no further revisions will be indexed.</td>
</tr>
<tr>
<td>What happens on upgrade from a Starter license, if indexing has been paused due to the committer limit being reached?</td>
<td>On restart of FishEye, indexing will resume for all repositories. Each repository can restarted individually to avoid restarting FishEye. Due to the nature of CVS indexing, we recommend reindexing any CVS repositories which have reached the committer limit prior to the license upgrade.</td>
</tr>
<tr>
<td>What happens when upgrading from a Starter license, when repositories have not started due to the repository limit being reached?</td>
<td>On restart of FishEye, all enabled repositories will start. Each repository can restarted individually to avoid restarting FishEye.</td>
</tr>
<tr>
<td>What happens if my evaluation license has expired and I upgrade to a Starter license, however I have exceeded the Starter license limitations?</td>
<td>As described above, a maximum of five repositories will start and for any repository with more than 10 committers, no further indexing will occur. All existing indexed content is retained and can be viewed.</td>
</tr>
<tr>
<td>What happens when downgrading to a Starter license, where the repository limit has been exceeded?</td>
<td>A maximum of five of your configured repositories will start running. The remainder will not start but will continue to be available.</td>
</tr>
<tr>
<td>What happens when downgrading to a Starter license, where the committer limit has been exceeded for one or more repositories?</td>
<td>No further indexing will occur for the repositories where the committer limit has been exceeded.</td>
</tr>
</tbody>
</table>

**Repository Management FAQ**

<table>
<thead>
<tr>
<th>FishEye Repository Management FAQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Setting the root location for internal repositories</td>
</tr>
<tr>
<td>• Migrating repositories to a changed root location</td>
</tr>
<tr>
<td>• Repairing a broken hierarchy of forks</td>
</tr>
<tr>
<td>• Permanent authentication for Git repositories over HTTP(S)</td>
</tr>
</tbody>
</table>

**Setting the root location for internal repositories**

If you have already created internal repositories, you must **migrate them when you update the**
For security reasons, you can not change the path that internal repositories are created under via the FishEye administration UI. Instead, you must edit the config.xml file directly.

Steps to update your repositories root:

1. Shut down your FishEye server.
3. Edit your config.xml and change the repositories-root attribute of the <repository-management> element to the new location. This must be an absolute path to the directory where you wish to store internal repositories created by FishEye.
4. If you are migrating existing repositories, follow the steps at Migrating repositories to a changed root location.
5. Migrate any existing backup procedures that you have set up for your managed repositories.
6. Restart your FishEye server. The new root will be automatically created if it doesn't already exist.

Related Links

- Setting the root location for internal repositories
- Migrating repositories to a changed root location
- Repairing a broken hierarchy of forks
- Permanent authentication for Git repositories over HTTP(S)

Migrating repositories to a changed root location

Before you restart your FishEye server after modifying the repositories-root attribute of the <repository-management> element in your config.xml file, you will need to migrate any existing repositories to the new location.

Steps to migrate your existing managed repositories:

1. Follow the steps at Setting the root location for internal repositories up until the reference to migrating existing repositories.
2. Copy all of the directories in your current repositories-root (by default, this is in the %FISHEYE_INST%/manged_repos directory) to the new location that you have specified.
3. Open your config.xml file for editing.
4. Update the location attribute of each <git> element that is the child of a managed <repository> element to reference the updated repository location under the new repositories root. Managed <repository> elements have the attribute managed="true".
5. Continue following the steps at Setting the root location for internal repositories.

Related Links

- Setting the root location for internal repositories
- Migrating repositories to a changed root location
- Repairing a broken hierarchy of forks
- Permanent authentication for Git repositories over HTTP(S)

Repairing a broken hierarchy of forks

If you see the error message "Failed to render fork tree for repository 'my-fork': The repository 'my-repository' does not exist but is referenced in the 'parent'..."
attribute of another repository." on your repository's **Forks** tab...

![Repository Forks Example](image)

...or see one of the following errors in your logs...

```
[java] 2011-09-06 12:24:51,093 WARN - The repository 'my-repository' does not exist but is referenced in the 'parent' attribute of another repository.
```

```
[java] 2011-09-06 12:24:51,093 WARN - Repository 'my-fork' has 'my-repository' set as parent, but no repository found named 'my-repository'
```

...you may need to repair some parent attribute references in your config.xml file.

**Steps to repair your config.xml file:**

1. Stop your FishEye instance.
3. Grep or otherwise search your config.xml for references to the repository that is missing (the name *my-repository* is used in the example above)
4. From the results of the above search, remove any `parent` attributes that reference the missing repository, e.g. `<repository parent="my-repository" name="my-fork">` should be modified to read `<repository name="my-fork">`.
5. Restart your FishEye instance and browse to the **Forks** tab for the effected repository.

**Related Links**

- Setting the root location for internal repositories
- Migrating repositories to a changed root location
- Repairing a broken hierarchy of forks
- Permanent authentication for Git repositories over HTTP(S)

**Permanent authentication for Git repositories over HTTP(S)**

Currently, FishEye only supports HTTP or HTTPS for pushing and pulling from managed Git repositories. Git has no method of caching the user's credentials, so you need to re-enter them each time you perform a clone, push or pull.

Fortunately, there is a mechanism that allows you to specify which credentials to use for which server: the `.netrc` file.

**Warning!**

Git uses a utility called **cURL** under the covers, which respects the use of the `.netrc` file. Be aware that other applications that use curl to make requests to servers defined in your `.netrc` file will also now be
1. Create a file called `.netrc` in your home directory (`~/.netrc`). Unfortunately, the syntax requires you to store your passwords in plain text - so make sure you modify the file permissions to make it readable only to you.

2. Add credentials to the file for the server or servers you want to store credentials for, using the format below. You may use either IP addresses or hostnames, and you do not need to specify a port number, even if you’re running FishEye on a non-standard port.

   ```
   machine fisheye1.mycompany.com
   login myusername
   password mypassword
   machine fisheye2.mycompany.com
   login myotherusername
   password myotherpassword
   ```

3. And that’s it! Subsequent `git clone`, `git pull` and `git push` requests will be authenticated using the credentials specified in this file.

**Windows**

1. Create a text file called `_netrc` in your home directory (e.g. `c:\users\kannonboy\_netrc`). Curl has problems resolving your home directory if it contains spaces in its path (e.g. `c:\Documents and Settings\kannonboy`). However, you can update your `%HOME%` environment variable to point to any old directory, so create your `_netrc` in a directory with no spaces in it (for example `c:\curl-auth\`) then set your `%HOME%` environment variable to point to the newly created directory.

2. Add credentials to the file for the server or servers you want to store credentials for, using the format from the Linux or OSX section above.

**Related Links**

- Setting the root location for internal repositories
- Migrating repositories to a changed root location
- Repairing a broken hierarchy of forks
- Permanent authentication for Git repositories over HTTP(S)

**Subversion FAQ**

**Warning!**

Curl will not match the machine name in your `.netrc` if it has a username in it, so make sure you edit your `.git/config` file in the root of your clone of the repository and remove the user and '@' part from any clone url’s (url fields) that look like `https://user@machine.domain.com/...` so instead they look like `http://machine.domain.com/...`
## FishEye Subversion FAQ

- **Configuring Start Revision based on date** — For Subversion repositories Fisheye has the ability to configure a Start Revision parameter to allow you to only index content from a given point in your repository.

- **Errors 'SEVERE assert' or 'Checksum mismatch'** — SVNKit may have problems with older version Subversion servers - versions 1.1.x and prior.

- **FishEye fails to connect to the Subversion repository after a short time of successful operation.** — On Unix systems, the `svn://` protocol is usually handled by `inetd` or `xinetd`. These daemons apply, by default, a connection per second limit to incoming connections. Any connections above this rate are rejected by the server.

- **How can FishEye help with merging of branches in Subversion?** — In merge management, the main advantages of FishEye come from its search functionality. If you record the revisions merged when you check in a merge result, you can find this information in FishEye easily for the next merge operation.

- **Subversion Changeset Parents and Branches**

- **SVN Authentication Issues** — If multiple repositories have been defined in FishEye for the same SVN Server and those repositories use different credentials, FishEye may not use the correct credentials.

- **What are Subversion root and tag branches?**

- **Why do I need to describe the branch and tag structure for Subversion repositories?** — In Subversion, branches and tags are defined by convention, based on their path within a repository, and not directly defined by the repository. A few different layout alternatives are commonly used. It is also possible that you are using your own custom layout. As a result you need to describe to FishEye which paths in your repository are used as branches and tags.

- **Why don't all my tags show up in FishEye?**

### Configuring Start Revision based on date

For Subversion repositories Fisheye has the ability to configure a **Start Revision** parameter to allow you to only index content from a given point in your repository.

Quite often users will find it helpful to index from a revision on a given date. For example, you may want to only index SVN data in the past year. To determine the revision based on date, you can use the following command:

```bash
svn log -r {YYYY-MM-DD}:HEAD <SVN_URL> -l 1
```

The output of this command will the revision number closest to the date that you provide.

### Errors 'SEVERE assert' or 'Checksum mismatch'

When using SVNKit, you may see errors in the FishEye log such as 'SEVERE: assert #B' or 'Checksum mismatch'.

SVNKit may have problems with older version Subversion servers - versions 1.1.x and prior. If this is the case you should either use the native JavaHL layer or upgrade your Subversion server to a more recent version.

### FishEye fails to connect to the Subversion repository after a short time of successful operation.

If you use the `svn://` protocol to access a Subversion repository, you may notice that FishEye fails to connect to the repository after a short time of successful operation.

On Unix systems, the `svn://` protocol is usually handled by `inetd` or `xinetd`. These daemons apply, by default, a connection per second limit to incoming connections. Any connections above this rate are rejected by the server.
Two options for fixing this problem:

- Ask your system administrator increase the connection rate allowed for the svn connection by updating the xinetd configuration, or
- Specify a connection per second limit in your FishEye repository definition, to prevent FishEye from exceeding the xinetd limits.

**How can FishEye help with merging of branches in Subversion?**

FishEye gives you a logical view of your branched files so you can see activity on a single file across multiple branches/trunk.

In merge management, the main advantages of FishEye come from its search functionality. If you record the revisions merged when you check in a merge result, you can find this information in FishEye easily for the next merge operation.

As an example, let's say you have a branch `dev` created at revision 1300 from `trunk`. Development has proceeded on both `trunk` and `dev`. At some point you wish to add the latest `trunk` changes into the `dev` branch. Let's say that is at revision 1400. When you check in the results of this merge, you would use some standard format checkin comment such as:

```
merge from trunk to dev 1300:1400
```

When you come to do the next merge, say at revision 1500, you can use FishEye search to find this checkin comment and know what the starting point for the merge should be. You can then check this in as:

```
merge from trunk to dev 1400:1500
```

Merges back to `trunk` from the `dev` branch are managed in the same way.

**Subversion Changeset Parents and Branches**

*Why do some changesets have more than one branch? Why do these changesets have more than one parent?*

In Subversion, a single changeset can have files and directories that are on different branches, as defined by the SVN tag and branch structure. In this situation, the changeset is considered to be on all of the branches of its constituent file revisions. If a changeset is on more than one branch it can have a parent changeset of each of its branches, giving the changeset multiple parents.

FishEye does not track SVN merges, so merges are not indicated on the graph.

*When I create a complex branch, how does FishEye determine which is its parent changeset? When I create a complex tag, how does FishEye decide which changeset to tag?*

In Subversion, a *simple* branch or tag is created by copying a source directory, e.g. copying `"/trunk"` to `"/branches/branch1"` or `"/tags/tag1"`. The tag or branch is considered *complex* if a part of the copied directory is replaced with another version, e.g. `"/trunk"` is copied to `"/tags/tag1"`, and then `"branches/branch1/dirl"` is copied to `"/tags/tag1/dirl"`.

For the purpose of the commit graph, FishEye looks at where the root directory was copied from, to determine where the branch or tag originated. In the example above, the label `"tag1"` would be applied to the latest changeset on trunk when the tag was created, even though part of the tag was copied from branch1. This only affects the annotation of the changeset, not the file revisions that are tagged — the tagged file revisions are still those on `trunk` or `branch1` as appropriate.

**SVN Authentication Issues**

If multiple repositories have been defined in FishEye for the same SVN Server and those repositories use different credentials, FishEye may not use the correct credentials.

FishEye does not control directly when authentication information is used to access Subversion repositories. It
delegates this operation to the JavaHL layer in use. JavaHL will then ask FishEye to supply credentials when required, using a callback. The default JavaHL layer shipped with FishEye, SVNKit, can cache credentials at the server level rather than at the repository level.

When experiencing this problem, FishEye can be configured to use the native JavaHL implementation, which will correctly apply the appropriate credentials.

The simplest solution is to have the same credentials for accessing the Subversion Server.

Alternatively, SVNKit can be tricked into thinking that different servers are being used. For each connection to a repository a hostname in the hosts file can be defined. All these entries then point to the same IP address of the SVN Server, but to SVNKit they look like different servers, thus bypassing the problem.

Example hosts entries (replace the IP address with the address of the SVN Server):

```
123.45.6.78  account1
123.45.6.78  account2
```

Replace these new server names in the SVN URLs:

```
http://account1/svn/project-a/
http://account2/svn/project-b/
```

**What are Subversion root and tag branches?**

FishEye identifies branches and tags in your Subversion repository by applying your specified SVN tag and branch structure.

**The "root:" branch**

Any files or directories that fall outside the tag and branch structure are identified as being on the special branch, "root:". Some directories will almost always fall outside this structure. In general, root directories of branches are considered to be on the "root:" branch. This means that any changeset in which a branch is created is considered to be on branch, "root:". Additionally, any files or directories that fall outside the defined structure will be assigned branch, "root:". If you're seeing a lot of files and changesets on "root:" you may need to update your branch and tag structure in FishEye and rescan your repository, or exclude parts of your repository that don't follow your defined structure.

**"tag:" branches**

When FishEye detects that a tag has been created, it looks at the files that were tagged and adds the tag as an annotation to those file revisions. No file revisions are created at this point.

If a tag is modified after it has been created and committed, FishEye promotes the tag to a branch to preserve the history of the modification. For example, a user may create the tag "build1" by copying "trunk" to "tags/build1". If they then modify contents of tags/build1, a new branch "tag:build1" will be created for the modification.

**Why do I need to describe the branch and tag structure for Subversion repositories?**

In Subversion, branches and tags are defined by convention, based on their path within a repository, and not directly defined by the repository. A few different layout alternatives are commonly used. It is also possible that you are using your own custom layout. As a result you need to describe to FishEye which paths in your repository are used as branches and tags.
It is very important that you correctly define in FishEye the layout you are using. If you do not, FishEye will not know which paths represent tags and branches. This will prevent FishEye from relating different versions of the same logical file across separate paths within your repository. It will also mean that FishEye's cache will be much larger as each tagged path will be indexed separately. This will result in an increase in the initial scan time and may reduce runtime performance.

If you are having trouble using Subversion tags, see How tags work in Subversion.

Why don’t all my tags show up in FishEye?

This page gives a detailed technical explanation of why certain issues affect Subversion users.

On this page:

- Introduction
- How Subversion Processes Tags and Branches
- An Example from a Live Subversion Repository
- Avoid Modifications in the Tag Area
- Conclusion

Introduction

When accessing Subversion via FishEye, you may see references to tags in the branches drop-down menu. In the example below, we can see tag1 and tag3 in the drop-down menu but not tag2:

Screenshot: The Branches Drop-Down Menu in FishEye

In actual fact, the branches drop-down menu shows only branch names. It does not show tags, but in some instances FishEye will synthesise a branch name to record certain operations. To understand how this occurs, you will need some background knowledge on Subversion tagging (introduced in the following segments of this page).

How Subversion Processes Tags and Branches

In Subversion, tags are only a convention and are typically the result of a copy operation from the trunk to a tag area in the tags directory. When FishEye processes this copy operation, it recognises that the destination is a tag directory and tags the source file on trunk with the name of the tag.

i.e. FishEye is interpreting the Subversion copy to a tag directory as a tagging operation on the trunk files.

For regular changes in your Subversion repository, FishEye records each change against a branch where the change took place. If, however, after tagging, you make a change to a file in the tagged area, you are making a change outside trunk or a recognized branch. FishEye records such changes by creating an artificial branch name and associating that branch name with the change. The branch name is derived from the tag name by prepending "tag:" (in other words, the characters "tag:" appear as the
first part of the name). The same thing will occur if you create a new file in the tagged area which does not come from an existing branch or trunk.

This is the reason you see some of your tags in the branch drop down. It means that for those tags, you have made a modification after the tagging operation.

An Example from a Live Subversion Repository

For example, consider tag4 in this screenshot:

Screenshot: Subversion Tag Changes in FishEye

There are two changes here. The first creates the tag and the second adds a new file in the tagged area. This will result in the creation of an artificial branch, called "tag:tag4" within FishEye.

Avoid Modifications in the Tag Area

In general, it's not good practice to make changes in the tag areas of a Subversion repository. Such changes can easily get lost if they are not applied to trunk or a current branch. It is preferable to make the change in trunk or a branch and then create a new tag to capture the update. Nevertheless, since Subversion tagging is merely a convention, this is sometimes convenient. FishEye handles this situation as described above.

Conclusion

In general a lot of systems have a large number of tags which would make the drop-down unworkable. This is the reason the tag field is a text-entry box below the branch drop-down menu in FishEye.

Since tags and branches are based on location convention in Subversion, the constraint is less effective than on other SCMs. You can always see the tag or branch you are interested in, based on its location in the repository. For example, the subdirectory list here shows all tags:

Screenshot: Subdirectory Listing in FishEye
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
- How to Report a Security Issue
- New Features Policy
- Patch Policy
- Security Advisory Publishing Policy
- Security Patch Policy
- Severity Levels for Security Issues

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 FishEye. 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 http://jira.atlassian.com for the stand-alone products and http://studio.atlassian.com for JIRA Studio and
Atlassian OnDemand.

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.

How to Report a Security Issue

Finding and Reporting a Security Vulnerability

If you find a security bug in the product, please open an issue on http://jira.atlassian.com in the relevant project.

- Set the priority of the bug to 'Blocker'.
- Provide as much information on reproducing the bug as possible.
- Set the security level of the bug to 'Developer and Reporters only'.

All communication about the vulnerability should be performed through JIRA, so that Atlassian can keep track of the issue and get a patch out as soon as possible.

If you discover a security vulnerability, please attempt to create a test case that proves this vulnerability locally before opening either a bug or a support issue. When creating an issue, please include information on how the vulnerability can be reproduced; see our Bug Fixing Policy for general bug reporting guidelines. We will prioritise fixing the reported vulnerability if your report has information on how the vulnerability can be exploited.

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 our Feature Request process.
- Atlassian provides consistent updates on the top 20 feature/improvement requests (in our issue tracker systems).

How to Track what Features are Being Implemented

When a new feature or improvement is scheduled, the ‘fix-for’ version will be indicated in the JIRA issue. This happens for the upcoming release only. We maintain roadmaps for more distant releases internally, but because these roadmaps are often pre-empted by changing customer demands, we do not publish them.

How Atlassian Chooses What to Implement

In every major release we aim to implement highly requested features, but it is not the only determining factor. Other factors include:

- **Customer contact:** We get the chance to meet customers and hear their successes and challenges at Atlassian Summit, Atlassian Unite, developer conferences, and road shows.
- **Customer interviews:** All product managers at Atlassian do customer interviews. Our interviews are not simply to capture a list of features, but to understand our customers’ goals and plans.
- **Community forums:** There are large volumes of posts on answers, of votes and comments on jira.atlassian.com, and of conversations on community forums like groups on LinkedIn.
- **Customer Support:** Our support team provides clear insights into the issues that are challenging for customers, and which are generating the most calls to support.
- **Atlassian Experts:** Our Experts provide insights into real-world customer deployments, especially for customers at scale.
- **Evaluator Feedback:** When someone new tries our products, we want to know what they liked and disliked and often reach out to them for more detail.
- **In product feedback:** The JIRA Issue Collectors that we embed our products for evaluators and our Early Access Program give us a constant pulse on how users are experiencing our product.
- **Usage data:** Are customers using the features we have developed?
- **Product strategy:** Our long-term strategic vision for the product.

How to Contribute to Feature Development

Influencing Atlassian’s release cycle

We encourage our customers to vote on feature requests in JIRA. The current tally of votes is available online in our issue tracking system, http://jira.atlassian.com. Find out if your improvement request already exists. If it does, please vote for it. If you do not find it, create a new feature or improvement request online.

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.

Patch Policy

Atlassian will only provide software patches in extremely unusual circumstances. If a problem has been fixed in a newer release of the product, Atlassian will request that you upgrade your instance to fix the issue. If it is deemed necessary to provide a patch, a patch will be provided for the current release and the last maintenance release of the last major version only.

Patches are issued under the following conditions:

- The bug is critical (production application down or major malfunction causing business revenue loss or high numbers of staff unable to perform their normal functions).
- A patch is technically feasible (i.e., it doesn't require a major architectural change)
- The issue is a security issue, and falls under our Security Patch Policy.

Atlassian does not provide patches for non-critical bugs.

Provided that a patch does not impact the quality or integrity of a product, Atlassian will ensure that patches supplied to customers are added to the next maintenance release. Customers should watch a filed bug in order to receive e-mail notification when a "Fix Version" is scheduled for release.

Patches are generally attached to the relevant http://jira.atlassian.com issue.

Further reading

See Atlassian Support Offerings for more support-related information.

Security Advisory Publishing Policy

Publication of Security Advisories

When a critical severity security vulnerability in an Atlassian product is discovered and resolved, Atlassian will inform customers through the following mechanisms:

- We will post a security advisory in the latest documentation of the affected product at the same time as releasing a fix for the vulnerability.
- We will send a copy of all posted security advisories to the Technical Alerts' mailing list for the product concerned.
  Note: To manage your email subscriptions and ensure you are on this list, please go to my.atlassian.com and click ‘Communications Centre’ near the top right of the page.
- If the person who reported the vulnerability wants to publish an advisory through some other agency, such as CERT, we will assist in the production of that advisory and link to it from our own.

If you want to track non-critical severity security vulnerabilities, you need to monitor the issue trackers for the relevant products on http://jira.atlassian.com. For example, https://jira.atlassian.com/browse/JRA for JIRA and https://jira.atlassian.com/browse/CONF for Confluence. Security issues in trackers will be marked with a “security” label. All security issues will be listed in the release notes of the release where they have been fixed, similar to other bugs.

One of the ways to monitor updates to security issues is subscribing to the results of a sample search via email.
Further reading

See Atlassian Support Offerings for more support-related information.

Security Patch Policy

Product Security Patch Policy

Atlassian makes it a priority to ensure that customers' systems cannot be compromised by exploiting vulnerabilities in Atlassian products.

Scope

This page describes when and how we release security patches and security upgrades for our products. It does not describe the whole of disclosure process that we follow. It also excludes OnDemand, since OnDemand will always be patched by Atlassian without additional notifications.

Critical vulnerabilities

When a **Critical** security vulnerability is discovered by Atlassian or reported by a third party, Atlassian will do all of the following:

- Issue a new, fixed release for the current version of the affected product as soon as possible, usually in a few days.
- Issue a binary patch for the current release.
- Issue a binary patch for the latest maintenance release of the previous version of the product.
- Patches for older versions or releases normally will not be issued.

Patches will be attached to the relevant JIRA issue. You can use these patches as a "stop-gap" measure until you upgrade your installation in order to fully fix the vulnerability.

Non-critical vulnerabilities

When a security issue of a **High**, **Medium** or **Low** severity is discovered, Atlassian will do all of the following:

- Include the fix into the next scheduled release, both for the current and previous maintenance versions.
- Where practical, provide new versions of plugins or other components of the product that can be upgraded independently.

You should upgrade your installation in order to fix the vulnerability.

Other information

Severity level of vulnerabilities is calculated based on [Severity Levels for Security Issues](https://confluence.atlassian.com/display/FISH/Security+Patch+Policy).

Visit our general [Atlassian Patch Policy](https://confluence.atlassian.com/display/FISH/Security+Patch+Policy) as well.

Examples

**Example 1:** A critical severity vulnerability is found in a (hypothetical current release) JIRA 5.3.2. The last bugfix release in 5.2.x branch was 5.2.3. In this case, a patch will be created for 5.3.2 and 5.2.3. In addition, new bugfix releases, 5.3.3 and 5.2.4, which are free from this vulnerability, will be created in a few days.

**Example 2:** A high or medium severity vulnerability is found in the same release as in the previous example. The fix will be included into the currently scheduled releases 5.3.3 and 5.2.4. Release schedule will not be brought forward and no patches will be issued. If the vulnerability is in a plugin module, then a plugin upgrade package may still be supplied.

Further reading
See Atlassian Support Offerings for more support-related information.

Severity Levels for Security Issues

Severity Levels

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 learn more about CVSS at FIRST.org web site.

CVSS scores are mapped into the following severity ratings:

- Critical
- High
- Medium
- Low

An approximate mapping guideline is as follows:

<table>
<thead>
<tr>
<th>CVSS score range</th>
<th>Severity in advisory</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 2.9</td>
<td>Low</td>
</tr>
<tr>
<td>3 – 5.9</td>
<td>Medium</td>
</tr>
<tr>
<td>6.0 – 7.9</td>
<td>High</td>
</tr>
<tr>
<td>8.0 – 10.0</td>
<td>Critical</td>
</tr>
</tbody>
</table>

Below is a summary of the factors which illustrate types of vulnerabilities usually resulting in a specific severity level. Please keep in mind that this rating does not take into account details of your installation.

Severity Level: Critical

Vulnerabilities that score in the critical range usually have most of the following characteristics:

- Exploitation of the vulnerability results in root-level compromise of servers or infrastructure devices.
- The information required in order to exploit the vulnerability, such as example code, is widely available to attackers.
- Exploitation is usually straightforward, in the sense that the attacker does not need any special authentication credentials or knowledge about individual victims, and does not need to persuade a target user, for example via social engineering, into performing any special functions.

For critical vulnerabilities, is advised that you patch or upgrade as soon as possible, unless you have other mitigating measures in place. For example, if your installation is not accessible from the Internet, this may be a mitigating factor.

Severity Level: High

Vulnerabilities that score in the high range usually have some of the following characteristics:

- The vulnerability is difficult to exploit.
- Exploitation does not result in elevated privileges.
- Exploitation does not result in a significant data loss.

Severity Level: Medium

Vulnerabilities that score in the medium range usually have some of the following characteristics:

- Denial of service vulnerabilities that are difficult to set up.
- Exploits that require an attacker to reside on the same local network as the victim.
• Vulnerabilities that affect only nonstandard configurations or obscure applications.
• Vulnerabilities that require the attacker to manipulate individual victims via social engineering tactics.
• Vulnerabilities where exploitation provides only very limited access.

Severity Level: Low

Vulnerabilities in the low range typically have very little impact on an organisation's business. Exploitation of such vulnerabilities usually requires local or physical system access.

Further reading

See Atlassian Support Offerings for more support-related information.

Troubleshooting

<table>
<thead>
<tr>
<th>FishEye Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>After I commit a change to my CVS repository, it takes a long time before it appears in FishEye.</strong> — If you do not have a CVSROOT/history file, then a commit will not appear in FishEye until after FishEye has done a periodic full scan of your repository. You can configure the period of this scan in the Admin pages.</td>
</tr>
<tr>
<td><strong>FishEye freezes unexpectedly</strong> — If your FishEye 2.0 or 2.0.1 instance freezes unexpectedly, this could be caused by a known issue with FishEye and MySQL database technology.</td>
</tr>
<tr>
<td><strong>Generating a Thread Dump Externally</strong> — If FishEye stops responding or is showing poor performance, providing thread dumps to support can help diagnose the problem.</td>
</tr>
<tr>
<td><strong>I have installed FishEye, and the initial scan is taking a long time. Is this normal?</strong> — As a guide, FishEye should be able to process about 100KB-200KB per second on an averaged-size PC. If FishEye is accessing the repository over the network (e.g. over a NFS mount), then you should expect the initial scan to take longer.</td>
</tr>
<tr>
<td><strong>I have installed FishEye, but there is no data in the Changelog.</strong> — When you add a repository, FishEye needs to scan through the repository to build up its index and cache. This scan can take some time. Until this scan is complete, you may find that some data is not displayed.</td>
</tr>
<tr>
<td><strong>Initial scan and page loads are slow on Subversion</strong> — It's possible that you've mis-configured your tag and branch structure and caused FishEye to process some or all files as trunk files. You should recheck your tag configuration.</td>
</tr>
<tr>
<td><strong>It seems that FishEye's HTTP Header is Too Small</strong></td>
</tr>
<tr>
<td><strong>JIRA Integration Issues</strong></td>
</tr>
<tr>
<td><strong>Message 'org.tigris.subversion.javahl.ClientException svn Java heap space'</strong> — The Java heap space needs to be increased to an acceptable size. See the FishEye Tuning documentation for more information.</td>
</tr>
<tr>
<td><strong>On my Red Hat Linux system, after running for several days FishEye freezes and does not accept any more connections.</strong> — On some Linux systems (particularly RH9), there are socket problems between the JVM and the kernel. To fix this, you need to set the LD_ASSUME_KERNEL environment variable before starting FishEye.</td>
</tr>
<tr>
<td><strong>Problems with very long comments and MySQL migration</strong> — There is a known issue with FishEye 2.0.x and very long comments when migrating your database to MySQL.</td>
</tr>
<tr>
<td><strong>URLs with encoded slashes don't work, especially in Author constraints</strong> — If the author names in your repository contain slashes or back-slashes, and you are using Apache, you may run into a problem where these characters are incorrectly escaped.</td>
</tr>
</tbody>
</table>

FishEye Knowledge Base

See the troubleshooting guides and technical announcements in the FishEye knowledge base.

After I commit a change to my CVS repository, it takes a long time before it appears in FishEye.
If possible, FishEye will monitor and parse the CVSROOT/history file in your repository to quickly work out what has changed. You may want to check with your CVS administrator to ensure this feature of CVS is turned on.

If you do not have a CVSROOT/history file, then a commit will not appear in FishEye until after FishEye has done a periodic full scan of your repository. You can configure the period of this scan in the Admin pages.

**FishEye freezes unexpectedly**

**Issue Symptoms**

If your FishEye 2.0 or 2.0.1 instance freezes unexpectedly, this could be caused by a known issue with FishEye and MySQL database technology.

This issue manifests itself in some FishEye pages returning a server timeout error. To identify the issue, check the FishEye error log. For this issue, the following output will appear in the error log:

```
2009-07-15 15:34:45,555 ERROR [btpool0-519] fisheye.app HibernateUtil-commitTransaction - Commit fail msg-0:Could not execute JDBC batch update
2009-07-15 15:34:45,556 ERROR [btpool0-519] fisheye.app HibernateUtil-commitTransaction - Commit fail msg-1:Lock wait timeout exceeded; try restarting transaction
... ...
Caused by: java.sql.BatchUpdateException: Lock wait timeout exceeded; try restarting transaction
at com.mchange.v2.c3p0.impl.NewProxyPreparedStatement.executeBatch(NewProxyPreparedStatement.java:1723)
at org.hibernate.jdbc.BatchingBatcher.doExecuteBatch(BatchingBatcher.java:48)
at org.hibernate.jdbc.AbstractBatcher.executeBatch(AbstractBatcher.java:246)
... 163 more
```

The FishEye error log can be found under `FISHEYE_INST/var/log/fisheye-error.log.YYYY-MM-DD`

See the [JIRA issue](#) for more information.

**Workaround**

Until the issue is solved, the suggested course of action is to restart your FishEye instance. This will return FishEye to normal operation.

The FishEye development team is actively working on a solution and this be part of an upcoming point release of FishEye.

**Requesting Support**

If you require assistance in resolving the problem, please raise a support request under the FishEye project.
Generating a Thread Dump Externally

If FishEye stops responding or is showing poor performance, providing thread dumps to support can help diagnose the problem.

ℹ️ If you were asked by Atlassian technical support to create the thread dump, please take 4 thread dumps with a time interval in between (eg. 30 secs) so we can see some patterns. Attach the log file to the support ticket.

Generating a Thread Dump for Windows

If you are not running FishEye via run.bat, click on the console and press <CTRL>+BREAK

Generating a Thread Dump on Linux, including Solaris and other Unices

Find the process ID of the JVM and use the `ps` command to get list of all processes:

```bash
kill -3 <pid>
```

Note: This will not kill your server (so long as you included the "-3" option, no space in between). The thread dump will be printed to FishEye's standard output (fisheye.out).

If you have trouble generating the thread dumps with this method, then use the method “Generating a Thread dump for Windows” as they can also apply for linux, etc.

`jstack`

Sun JDK 1.5 and above ship with native tool called `jstack` to perform thread dump. To use the tool find the Process ID and execute the command:

```bash
jstack <ProcessID>
```

Output

Standard logging for FishEye is sent to the `FISHEYE_INST/var/log/fisheye-debug.log.*` files, in the `FISHEYE_INST` directory. Thread dumps are an exception since they dump the threads of the entire application server - they'll appear in the `FISHEYE_INST/var/log/fisheye.out` file. You can search for the term "thread dump" in the log file for the beginning of the dump.

Thread Dump Tools

- Samurai
- Thread Dump Analyzer TDA TDA 2.2 Final can be obtained from java.net

I have installed FishEye, and the initial scan is taking a long time. Is this normal?

When you add a repository, FishEye needs to scan through the repository to build up its index and cache. This scan can take some time. Until this scan is complete, you may find that some data is not displayed.

As a guide, FishEye should be able to process about 100KB-200KB per second on an averaged-size PC. If FishEye is accessing the repository over the network (e.g. over a NFS mount), then you should expect the initial scan to take longer.

For more details, see Improve FishEye scan performance.

I have installed FishEye, but there is no data in the Changelog.

When you add a repository, FishEye needs to scan through the repository to build up its index and cache. This scan can take some time. Until this scan is complete, you may find that some data is not displayed.
As a guide, FishEye should be able to process about 100KB-200KB per second on an averaged-size PC. If FishEye is accessing the repository over the network (e.g. over a NFS mount), then you should expect the initial scan to take longer.

**Initial scan and page loads are slow on Subversion**

**Background Information**

When you add a repository, FishEye needs to perform a once-off scan through the repository to build up its initial index and cache. This scan can take some time. Until this scan is complete, you may find that some data is not displayed. As a guide, FishEye should be able to process about 100KB-200KB per second on an averaged-size PC. If FishEye is accessing the repository over the network (e.g. over a NFS mount), then you should expect the initial scan to take longer. Read on if your scan appears to be considerably slower than expected.

**Solutions**

It's possible that you've mis-configured your tag and branch structure and caused FishEye to process some or all files as trunk files. You should recheck your tag configuration.

If that fails, then the Atlassian support team will be happy to help you with this issue. Please sign up for a support account if you don't have one already, then login and create a FishEye support request.

Users with very large or non-local repositories may be able to improve their FishEye scan performance.

**It seems that FishEye's HTTP Header is Too Small**

If you are receiving errors about FishEye's HTTP header being too small, it is adjustable. See the page on Setting JVM System Properties for instructions.

**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 FishEye, an error will occur. Depending on the error returned from JIRA, FishEye 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 FishEye debug log, named fisheye-debug.log.YYYY-MM-DD under the dist.inst/var/log folder of your FishEye installation and 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.

**Message 'org.tigris.subversion.javahl.ClientException svn Java heap space'**

When adding a new repository and on the initial scan, you may receive messages similar to this in the logs: org.tigris.subversion.javahl.ClientException: svn: Java heap space

The Java heap space needs to be increased to an acceptable size. See the FishEye Tuning documentation for more information.

**On my Red Hat Linux system, after running for several days FishEye freezes and does not accept any more connections.**

On some Linux systems (particularly RH9), there are socket problems between the JVM and the kernel. To fix this, you need to set the LD_ASSUME_KERNEL environment variable before starting FishEye.

Add the following code to the script that starts FishEye:

```bash
export LD_ASSUME_KERNEL=2.4.1
```

**Problems with very long comments and MySQL migration**
**Issue Symptoms**

There is a known issue with FishEye 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 configuration, you may see an error message like this while migrating to MySQL, causing the migration to fail:

```plaintext
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](https://jira.atlassian.com/browse/CRU-121).

**Workaround**

If your data contains very long comments or review descriptions (longer than 21,845 multibyte unicode characters), consider avoiding use of MySQL until this issue is resolved fully. Alternatively, use PostgreSQL or the default (built-in) HSQLDB database.

The FishEye developers are actively working on a solution to this problem and it will be addressed in an upcoming FishEye point release.

**Requesting Support**

If you require assistance in resolving the problem, please [raise a support request](https://jira.atlassian.com/secure/CreateRequest.jspa) under the FishEye project.

**URLs with encoded slashes don’t work, especially in Author constraints**

If the author names in your repository contain slashes or back-slashes, and you are using Apache, you may run into a problem where these characters are incorrectly escaped. By default Apache explicitly forbids encoded slashes or back-slashes in URLs. You can change this behavior with the following `httpd.conf` directive:

```plaintext
AllowEncodedSlashes On
```

This directive is documented [here](https://httpd.apache.org/docs/2.2/mod/core.html#allowencodedslashes).

**Contributing to the FishEye Documentation**

Would you like to share your FishEye hints, tips and techniques with us and with other FishEye users? We welcome your contributions.

**On this page:**

- Blogging your Technical Tips and Guides – Tips of the Trade
- Contributing Documentation in Other Languages
- Updating the Documentation Itself
  - Getting Permission to Update the Documentation
  - Our Style Guide
  - How we Manage Community Updates
Blogging your Technical Tips and Guides – Tips of the Trade

Have you written a blog post describing a specific configuration of FishEye or a neat trick that you have discovered? Let us know, and we will link to your blog from our documentation. More....

Contributing Documentation in Other Languages

Have you written a guide to FishEye 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. More....

Updating the Documentation Itself

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.

RELATED TOPICS

Tips of the Trade
Author Guidelines
Atlassian Contributor License Agreement

Tips of the Trade

Below are some links to external blog posts and articles containing technical tips and instructions on setting up and using FishEye. This page presents an opportunity for customers and community authors to share information and experiences.

The references here are specific to FishEye and are technical 'how to' guides written by bloggers who use FishEye. For more general information on source repository insight, best practices and business cases, please refer to the Atlassian website.

⚠️ Please be aware that these are external blogs and articles.
Most of the links point to external sites, and some of the information is relevant to a specific release of
FishEye. Atlassian provides these links because the information is useful and relevant at the time it was written. Please check carefully whether the information is still relevant when you read it, and whether it is relevant to your version of FishEye. Unless explicitly stated, Atlassian does not offer support for third-party extensions or plugins. The information in the linked blog posts has not been tested or reviewed by Atlassian. We recommend that you test all solutions on a test server before trying them on your production site.

On this page:

- Using a reverse proxy to Fisheye from Tomcat 5+ – Blog
- Diary of a FishEye Hacker – Part 1 and Part 2

### Technical Tips

**Using a reverse proxy to Fisheye from Tomcat 5+ – Blog**

- By: Dieter Wimberger, on his Coalevo blog.
- About: How to use a reverse proxy to Fisheye from Tomcat 5.
- Date and FishEye version: 21 Jan 2008; FishEye 1.4.2.
- Related documentation: Integrating with Other Web Servers.

### Plugin Development

**Diary of a FishEye Hacker – Part 1 and Part 2**

- By: Dan Hardiker, on the Adaptavist blog
- About: How to write a FishEye reporting plugin that provides a framework for alternative views on the repository data
- Date and FishEye version: 6 May 2008; FishEye 1.5
- Related documentation: FishEye Development Hub

✅ Have you written a technical tip for FishEye?

Add a comment to this page, linking to your blog post or article. We will include it if the content fits the requirements of this page.

✅ Feedback?

Your first port of call should be the author of the linked blog post. If you want to let us know how useful (or otherwise) a linked post is, please add a comment to this page.

### Other Sources of Information

- FishEye documentation
- Atlassian website
- Atlassian Answers
- Atlassian Blog
- FishEye plugins

**FishEye Documentation in Other Languages**

Below are some links to FishEye documentation written in other languages. In some cases, the documentation may be a translation of the English documentation. In other cases, the documentation is an alternative guide written from scratch in another language. This page presents an opportunity for customers and community authors to share documentation that they have written in other languages.

⚠️ Please be aware that these are external guides.
Documentation for FishEye 3.0

Most of the links point to external sites, and some of the information is relevant to a specific release of FishEye. Atlassian provides these links because the information is useful and relevant at the time it was written. Please check carefully whether the information is still relevant when you read it, and whether it is relevant to your version of FishEye. The information in the linked guides has not been tested or reviewed by Atlassian.

On this page:

- No guides yet

Adding Your Own Guide to this Page

Have you written a guide for FishEye in another language? Add a comment to this page, linking to your guide. We will include it if the content fits the requirements of this page.

Giving Feedback about One of the Guides

If you have feedback on one of the guides listed above, please give the feedback to the author of the linked guide.

If you want to let us know how useful (or otherwise) one of these guides is, please add a comment to this page.

Other Sources of Information

FishEye documentation
Atlassian website
Atlassian blog
FishEye plugins

FishEye Resources

Resources for Evaluators

- Free Trial
- Feature Tour

Resources for Administrators

- FishEye Knowledge Base
- FishEye FAQ
- Tips of the Trade
- Guide to Installing an Atlassian Integrated Suite
- The big list of Atlassian gadgets

Downloadable Documentation

- FishEye documentation in PDF, HTML or XML formats

Plugins

- FishEye Developer Documentation
- Add-ons for FishEye

Support
Glossary

Code repository or SCM (Source Code Management) software terminology can be confusing. This page provides definitions for some of the most commonly used terms.

On this page:

- Branch
- Changeset
- Commit
- Committer
- CSID
- Head
- Linker
- Repository
- SCM
- Slurp
- Tag
- Trunk

FishEye and its documentation uses the following terms:

Branch

A branch is an independent stream of work in a repository. For example, you might copy a set of files in the repository into a new branch, where you can carry on with a separate stream of work without cluttering up the main production area on trunk.

Different SCMs handle branching and merging in different ways. The common aspects allow users to create a branch in which to make changes which do not affect the files in other branches and the trunk development stream. These changes can then be merged into the trunk in a controlled fashion when a development unit of work is complete. Branches can also be used for experimental changes so that these do not affect the main development.

Changeset

A changeset is a collection of changes to files in a repository which are committed to the repository in a single operation with a single commit message. Not all SCMs support atomic commit operations. For these SCMs, FishEye will determine the file revisions which make up the changeset using a reliable heuristic (set of rules).

Different SCMs use different terms for the concept of a changeset, such as "changelist", which is generally
interchangeable with changeset.

Commit

A commit is a single entry of content (usually source code) into a repository. It can be a single file or comprised of multiple file versions.

Committer

A committer is a user of an SCM repository who is adding content to the repository (where it will be permanently archived). Typically, a committer is a programmer who is committing source code but SCMs can also store other files such as documents, images and schematics.

CSID

An abbreviation for ‘Changeset ID’, this is a code that is used to reference every set of files that is committed to a repository. For example, if you commit three files to a repository, they are collectively a changeset, and will share the one CSID. CSIDs normally appear as a number, for example ‘31905’. In FishEye, CSIDs appear as links that you can click to visit the 'Changeset View', which shows a list of the files in the left column, and the file contents or diffs in the right hand pane. In some circumstances you can hover your mouse over the CSID to see the 'Changeset Hover' dialog, which displays the commit message, author, timestamp and files in the changeset.

Head

The "head" revision is the latest change to be made to a file in either a trunk or a branch part of a repository.

Linker

FishEye can render issue IDs or Bug IDs that appear in commit messages or comments as hyperlinks to the relevant issue/bug in your issue/bug tracker. The "linker" patterns used to detect the ID substrings can be configured separately for each repository. See Linkers.

Repository

A repository is a area managed by an SCM where you store a set of related files, typically from a project or set of projects. The SCM is responsible for version controlling the files in the repository and maintaining their change history. A repository will contain the trunk and all branches for the files of the various projects. A single SCM instance can typically house multiple repositories.

SCM

SCM (Source Code Management) software is a category of computer software that archives complex sets of files, for example, all the source code comprised in a large multi-part software project. SCM systems keep copies of every version of every file, allowing you to completely restore and build any version of the software from any point in time.

Committers typically add new versions of code to the SCM once it is tested and approved by peer code review or quality assurance.

Each instance of an SCM can host multiple repositories.

Slurp

"Slurping" is a term that is synonymous with "repository scanning". FishEye must do intensive scans of the contents of repositories (SCM systems) in order to provide its lightning-fast web-based browsing of their
contents. This can be referred to as a *slurp*, or *slurping*.

**Tag**

In SCM terminology, a "tag" is a label that is added to a number of files, to capture their collective state at a particular moment in time. A typical tag would be a specific software version number, that could be referenced to see all the files that belong to a specific version build of a software project.

**Trunk**

In SCM terminology, the "trunk" is the central part of the version control "tree". For example, you might copy a set of files in the repository into a new branch, where you can do new experimental work without cluttering up the main production area on trunk.