



Documentation for FishEye 2.10

Contents

Getting started	8
Supported platforms	8
End of Support Announcements for FishEye	12
End of Support Announcement for IBM ClearCase	15
End of Support Announcement for Internally Managed Repositories	15
Installing FishEye on Windows	17
Running FishEye as a Windows service	20
Installing FishEye on Linux and Mac	23
Starting to use FishEye	27
Configuring JIRA Integration in the Setup Wizard	32
Using FishEye	40
Using the FishEye Screens	41
Browsing through a repository	43
Searching FishEye	46
Viewing a File	51
Viewing File Content	52
Using Side by Side Diff View	53
Viewing a File History	55
Viewing the changelog	56
FishEye Charts	59
Using favourites in FishEye	63
Changeset Discussions	64
Viewing the commit graph for a repository	64
Viewing People's Statistics	69
Using smart commits	70
Changing your User Profile	75
Re-setting your password	79
Pattern matching guide	80
Date Expressions Reference Guide	81
EyeQL Reference Guide	82
Administering FishEye	87
Managing your repositories	88
Adding an external repository	91
CVS	91
Git	92
Mercurial	95
Perforce	97
Subversion	100
SVN fisheye.access	104
SVN tag and branch structure	105
Adding an Internal Repository	113
Enabling Repository Management in FishEye	114
Creating Git Repositories	116
Forking Git Repositories	118
Deleting a Git Repository	121
Setting up a Repository Client	121
CVS Client	121
Git Client	121
Mercurial Client	121
Perforce client	122
Subversion Client	123

Native Subversion Client	123
SVNkit Client	125
Re-indexing your repository	125
Repository Options	127
Authentication	129
Commit Message Syntax	135
Configuring repository details	137
Hidden directories	139
Include Exclude Paths	141
Indexer	144
Linkers	145
Operations	150
Permissions	150
Properties	153
Store diff info	154
Tarball settings	155
Text file size limiting	157
Updater	158
Watches	160
Managing users and groups in FishEye	162
Creating a user	163
Creating a group	164
Adding a user to a group	164
Associating a group with a repository	165
Configuring avatar settings	166
Configuring anonymous access	167
Editing a user's details	167
Changing a user's password	168
Renaming a user	168
Deleting or deactivating a user	169
Configuring External Authentication Sources	170
Loading Users from Active Directory, LDAP, Crowd or JIRA	170
Synchronising With Large User Repositories	171
LDAP Authentication	172
How to connect FishEye to Active Directory or another LDAP server	174
JIRA and Crowd Authentication	178
Custom authentication	180
Host-Based Authentication	181
AJPv13 Authentication	183
Understanding authentication	184
Forcing Imported Usernames to be Lowercase	185
Configuring SMTP	186
Customising email notifications	186
Freemarker Data Model for Email Templates	188
Migrating to an external database	190
Migrating to MySQL	190
Migrating to Oracle	194
Migrating to PostgreSQL	196
Migrating to SQL Server	199
Backing up and restoring FishEye data	201
Backing Up and Restoring Managed Repositories	210
Managing add-ons	211
Integrating FishEye with Atlassian applications	212
JIRA Integration in FishEye	214
User Management Limitations and Recommendations	218
Configuring Application Links	219
Adding an Application Link	220

Configuring Authentication for an Application Link	222
Configuring Basic HTTP Authentication for an Application Link	225
Configuring OAuth Authentication for an Application Link	226
Configuring Trusted Applications Authentication for an Application Link	228
Deleting an Application Link	230
Editing an Application Link	231
Making an application link the primary link	232
Relocating an Application Link	232
Upgrading an Application Link	233
Configuring Project Links across Applications	235
Trusted Applications	237
System Settings	237
FishEye folder layout	238
Command-Line Options	239
Configuring Web Hooks	242
Enabling Smart Commits	246
Configuring commit hooks	248
Setting the REST API token	251
Configuring ViewVC Compatibility	252
Customising the Welcome Message	253
Software Update Notifications	255
Enabling Access Logging in FishEye	255
Setting JVM System Properties	257
System Properties	257
FishEye and Crucible Cookies	259
Configuring the FishEye web server	264
FishEye SSL configuration	269
Integrating with other web servers	273
Configuring indexing	275
Optimising the Search Indexes on Restart	276
Customising FishEye's look and feel	276
Environment variables	277
Configuring Outgoing Web Proxy Support for FishEye or Crucible	281
Configuring FishEye Security	282
Best practices for configuring FishEye security	283
Brute force login protection	284
Configuring CAPTCHA	285
Granting administrator privileges	285
Configuring public signup	286
Tuning FishEye performance	287
Best practices for FishEye configuration	289
Separating FishEye instance data from application data	290
Contacting Support	291
Finding your FishEye or Crucible Support Entitlement Number (SEN)	294
Configuring the database connection pool	295
FishEye Release Notes	298
FishEye upgrade guide	300
FishEye release summary	302
FishEye 2.10 release notes	304
FishEye 2.10 changelog	307
FishEye 2.10 upgrade guide	312
FishEye 2.9 release notes	313
FishEye 2.9 changelog	317
FishEye 2.9 upgrade guide	322
FishEye 2.8 release notes	323
FishEye 2.8 changelog	327
FishEye 2.8 upgrade guide	334

Upgrading from FishEye 2.7 with existing ClearCase Repositories	335
FishEye 2.7 Release Notes	336
FishEye 2.7 Changelog	340
FishEye 2.7 Upgrade Guide	368
FishEye 2.6 Release Notes	369
FishEye 2.6 Changelog	375
FishEye 2.6 Upgrade Guide	386
FishEye 2.5 Release Notes	387
FishEye 2.5 Changelog	392
FishEye 2.5 Upgrade Guide	405
FishEye 2.4 Release Notes	406
FishEye 2.4 Changelog	411
FishEye 2.4 Upgrade Guide	436
FishEye 2.3 Release Notes	437
FishEye 2.3 Changelog	440
FishEye 2.3 Upgrade Guide	444
Older release notes	445
FishEye 2.2 Release Notes	445
FishEye 2.2 Changelog	448
FishEye 2.2 Upgrade Guide	449
FishEye 2.1 Release Notes	450
FishEye 2.1 Changelog	453
FishEye 2.1 Upgrade Guide	454
FishEye 2.0 Release Notes	454
FishEye 2.0 Changelog	462
FishEye 2.0 Upgrade Guide	464
FishEye 2.0 Beta Release Notes	465
Upgrading to the FishEye 2.0 Beta	469
FishEye 2.0 Beta Reviewer's Guide	470
JIRA Integration in FishEye 2.0 Beta	471
Git Alpha in FishEye 2.0 Beta	475
FishEye 1.6 Release Notes	476
FishEye 1.6 Changelog	483
FishEye 1.6 Upgrade Guide	484
FishEye 1.5 Release Notes	485
FishEye 1.5 Changelog	487
FishEye 1.5 Upgrade Guide	488
FishEye 1.4 Release Notes	489
FishEye 1.4 Changelog	492
FishEye 1.4 Upgrade Guide	493
FishEye 1.3 Release Notes	494
FishEye 1.3 Changelog	495
FishEye 1.3 Upgrade Guide	497
Security advisories	498
FishEye and Crucible Security Advisory 2012-08-21	498
FishEye and Crucible Security Advisory 2012-05-17	500
FishEye and Crucible Security Advisory 2012-01-31	501
FishEye and Crucible Security Advisory 2011-11-22	502
FishEye and Crucible Security Advisory 2011-05-16	504
FishEye and Crucible Security Advisory 2011-01-12	506
FishEye Security Advisory 2010-10-20	509
FishEye Security Advisory 2010-06-16	511
FishEye Security Advisory 2010-05-04	513
FishEye FAQ	517
CVS FAQ	519
How does FishEye calculate CVS changesets?	519
Example EyeQL Queries	519

How do I find changes made to a branch after a given tag?	520
How do I filter results?	520
How do I find changes between two versions, showing separate histories?	520
How do I find changes made between two version numbers?	520
How do I find commits without comments?	520
How do I find files on a branch, excluding deleted files?	521
How do I find files removed from a given branch?	521
How do I find revisions made by one author between versions?	521
How do I select the most recent revisions in a given branch?	521
How do I show all changesets which do not have reviews?	521
FishEye Developer FAQ	521
General FAQ	522
About database encoding	522
About the Lines of Code Metric	524
Cannot View Lines of Code Information in FishEye	525
Finding your Server ID	526
How Do I Archive a Branch within Perforce	526
How do I Avoid Long Reindex Times When I Upgrade?	526
Mercurial Known Issues	528
Ordering of Branches Important When Visualising Git Changesets	528
Perforce Changesets and Branches	529
What SCM systems are supported by FishEye?	529
Automating Administrative Actions in Fisheye	529
Installation & Configuration FAQ	530
Can I deploy FishEye or Crucible as a WAR?	531
Does Fisheye support SSL (HTTPS)?	531
Improve FishEye scan performance	531
Migrating FishEye Between Servers	532
Setting up a CVS mirror with rsync	533
What are the FishEye System Requirements?	535
How to reset the Administration Page password in Fisheye or Crucible	535
How Do I Configure an Outbound Proxy Server for FishEye	535
How to remove Crucible from FishEye 2.x or later	535
How to run Fisheye or Crucible on startup on Mac OS X	536
Integration FAQ	537
How do I disable the FishEye tab panel for non-code projects?	538
How do I enable debug logging for the JIRA FishEye plugin?	538
How do I uninstall the JIRA FishEye plugin?	538
How is the Crucible review tab panel for the FishEye JIRA plugin populated?	539
What do I do if I discover a bug with the JIRA FishEye plugin?	539
Licensing FAQ	539
Are anonymous users counted towards FishEye's licence limits?	539
Restrictions on FishEye Starter Licenses	539
Repository Management FAQ	542
Setting the root location for internal repositories	542
Migrating repositories to a changed root location	542
Repairing a broken hierarchy of forks	543
Permanent authentication for Git repositories over HTTP(S)	544
Subversion FAQ	545
Configuring Start Revision based on date	545
Errors 'SEVERE assert' or 'Checksum mismatch'	545
FishEye fails to connect to the Subversion repository after a short time of successful operation. ..	546
How can FishEye help with merging of branches in Subversion?	546
Subversion Changeset Parents and Branches	546
SVN Authentication Issues	547
What are Subversion root and tag branches?	547
Why do I need to describe the branch and tag structure for Subversion repositories?	548

Why don't all my tags show up in FishEye?	548
Support Policies	551
Bug Fixing Policy	551
How to Report a Security Issue	552
New Features Policy	553
Patch Policy	554
Security Advisory Publishing Policy	554
Security Patch Policy	555
Severity Levels for Security Issues	556
Troubleshooting	557
After I commit a change to my CVS repository, it takes a long time before it appears in FishEye. .	558
FishEye freezes unexpectedly	558
Generating a Thread Dump Externally	559
I have installed FishEye, and the initial scan is taking a long time. Is this normal?	560
I have installed FishEye, but there is no data in the Changelog.	560
Initial scan and page loads are slow on Subversion	560
It seems that FishEye's HTTP Header is Too Small	560
JIRA Integration Issues	560
Message 'org.tigris.subversion.javahl.ClientException svn Java heap space'	561
On my Red Hat Linux system, after running for several days FishEye freezes and does not accept any more connections.	561
Problems with very long comments and MySQL migration	561
URLs with encoded slashes don't work, especially in Author constraints	562
FishEye Resources	562
Glossary	563
Contributing to the FishEye Documentation	565
Tips of the Trade	566
FishEye Documentation in Other Languages	567

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 2.10.x** and its minor releases.

Key:  = Supported;  = Not Supported

<i>Java Version</i>		

JRE / JDK	 1.6  1.7	FishEye requires Java Runtime (JDK or JRE), version as noted. Pre-release/Early access versions of the Java Runtime are <i>not supported</i>. 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.
Operating Systems		
Microsoft Windows		FishEye is a pure Java application and should run on any platform provided the requirements for the JRE or JDK are satisfied.
Linux		
Apple Mac OS X		

Databases		
MySQL	✔ MySQL Enterprise Server 5.x ✔ MySQL Community Server 5.x For 5.0, version must be 5.0.21 or later For 5.1, version must be 5.1.10 or later	The FishEye built-in database, running HSQLDB, is somewhat susceptible to data loss during system crashes. External databases (such as MySQL) are generally more resistant to data loss during a system crash. See the FishEye Database documentation for further details.
PostgreSQL	✔ 8.2, 8.3, 8.4	
Oracle	✔ 11g	
SQL Server	✔ 2005, 2008, 2008 R2	
HSQLDB	✔ Bundled; for evaluation use only	
Web Browsers		
Microsoft Internet Explorer	✔ 8.0, 9.0 ✖ 6.0 and 7.0 are not supported	
Mozilla Firefox	✔ Latest stable version supported ✔ 3.6, 4.0	
Safari	✔ Latest stable version supported ✔ 4, 5	
Chrome	✔ Latest stable version supported	
Version Control Systems		
Subversion	✔ 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	
CVS (and CVSNT)	✔ All versions	
Perforce	✔ Client version 2007.3 or later	

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. 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.

Component	Specifications
CPU	1.8GHz or higher, a single core is sufficient. More cores or higher GHz will result in better load-handling ability.
RAM	1GB minimum, 2GB will provide performance "headroom". Your Java heap should be sized at 512MB with the FISHEYE_OPTS environment variable, adjustable up to 1024MB depending on performance.
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).

i 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.

Repository Technology	Commits	Codebase Size (HEAD of trunk)	FishEye Index Size

Subversion	14386	466 MB in 12151 files	647 MB
CVS	8210	115 MB in 11433 files	220 MB

 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

Subversion (server)	FishEye can communicate with any repository running Subversion 1.1 or later.
Subversion (client)	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.
Perforce (client)	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.
CVS	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.

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

(Especially for Linux users.) For best results you may want to tweak your default monospace font and font-size. The default browser font is usually Courier New which can be hard to read in some browsers. We recommend choosing the same font you use in your IDE and selecting a font size approximately 2 points larger than your variable width font. Firefox 3, Internet Explorer 7 and Safari all have excellent font rendering. It is worth taking some time to tweak your fonts for the best experience.

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

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

The table above summarises information regarding the end of support announcements for upcoming 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@atlassian.com.

Database	Support End Date
MySQL 5.0	January 2012
PostgreSQL 8.0 and 8.1	January 2012

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

- Atlassian intends to end of life support for MySQL 5.0, PostgreSQL 8.0 and 8.1 in January 2012. The release of 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 eo1-announcement@atlassian.com.

End of Life Announcement for Web Browser Support

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

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 eo1-announcement@atlassian.com.

End of Life Announcement for Java Platform Support

Java Platform	Support End Date
Java Platform 5 (JDK/JRE 1.5)	When 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 `eo1-announcement@atlassian.com`.

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

IBM ClearCase End of Support Notes:

- Atlassian intends to end of life IBM ClearCase functionality on 4 April 2012. The release of 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 feature.
- 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...)

This repository will be used as the new remote for your development.

2. Open your terminal and go to the local copy of the directory that you want to push.

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

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

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

4. Push all the branches to the remote repository.

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

5. Push all the tags to the remote repository.

```
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.

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 by FishEye as an external Git repository.

8. Delete your old managed repository

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.

Related pages:

- [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](#).

 *The Java install path should not contain spaces, so don't install into C:\Program Files\Java\.*
Instead, use a path like C:\Java.

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

3. Check that Windows can find Java

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

```
echo %JAVA_HOME%
```

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

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

- If you see a path with spaces, like `C:\Program Files\Java\`, then sorry, but go back to 2. and reinstall Java to a location that doesn't have spaces.
- If you don't see a path at all, or if you just see `%JAVA_HOME%`, then set `JAVA_HOME` as follows:

For Windows 7:

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

4. Create a dedicated 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.

i 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:

```
bin\start.bat
```

After a few moments, in a web browser on the same machine, go to <http://localhost:8060/> (or, from another machine, type <http://hostname:8060/>, where hostname is the name of the machine where you 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 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, <FishEye home directory>\wrapper\bin, etc and NOT <FishEye home directory>\wrapper\wrapper, <FishEye home directory>\wrapper\wrapper\bin. The location of the wrapper directory is important.
3. Tell the wrapper where to find the Java JDK by editing the <FishEye home directory>\wrapper\conf\wrapper.conf file, replacing this:

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

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

```
# Java Application

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

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

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

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

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

4. Set the FISHEYE_INST environment variable (and other 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>\var\log\wrapper.log.

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

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

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

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

Note that the the -Xrs options should be used when running 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:

```
# 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:

```
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...**

Linux	Mac
Do either of the following: - If JAVA_HOME is not set, log in with 'root' level permissions and run: <pre>echo JAVA_HOME="path/to/JAVA_HOME" >> /etc/environment</pre> where path/to/JAVA_HOME may be like: /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: <pre>JAVA_HOME="path/to/JAVA_HOME"</pre> It should look like: <pre>JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Home/</pre>	Insert the following in your ~/.profile file: <pre>JAVA_HOME="path/to/JAVA_HOME" export JAVA_HOME</pre> where path/to/JAVA_HOME may be like: /System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Home/ Refresh your ~/.profile in the terminal and confirm that JAVA_HOME is set: <pre>source ~/.profile \$JAVA_HOME/bin/java -version</pre> You should see a version of Java that is 1.6.0 or higher, like this: <pre>java version "1.6.0_24"</pre>

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:

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

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, 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

Assumptions

This page assumes that:

- You have installed and started the latest version of FishEye.
- You are using a [supported browser](#).

On this page:

- [Assumptions](#)
- [Install FishEye](#)
- [Create users in FishEye](#)
- [Add a repository](#)

Related pages:

- [Installing FishEye on Windows](#)
- [Installing FishEye on Linux and Mac](#)
- [Supported platforms](#)
- [Managing your repositories](#)
- [Setting up your Users and Security](#)

Install FishEye

Get FishEye running on your computer. See the details here:

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

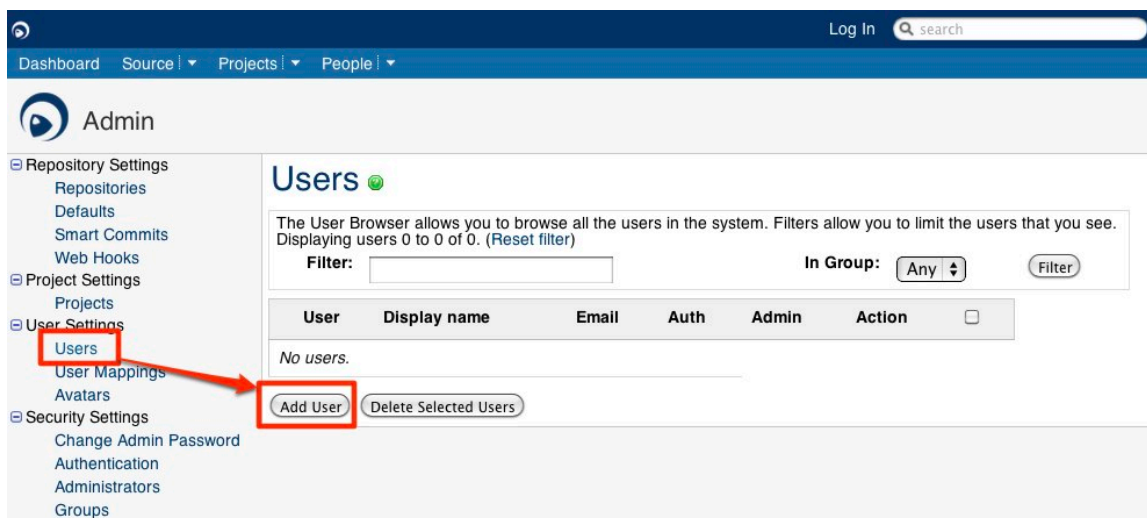
Create users in FishEye

FishEye doesn't have any users created when you install 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

Add New User

Information about the new user

Username

Display name

Email

Auth Type

Password

Confirm Password

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**

[Edit](#) | [Change Password](#) | [Rename](#) | [Delete](#)

Group Details

Groups **No groups.**

[Edit Groups](#)

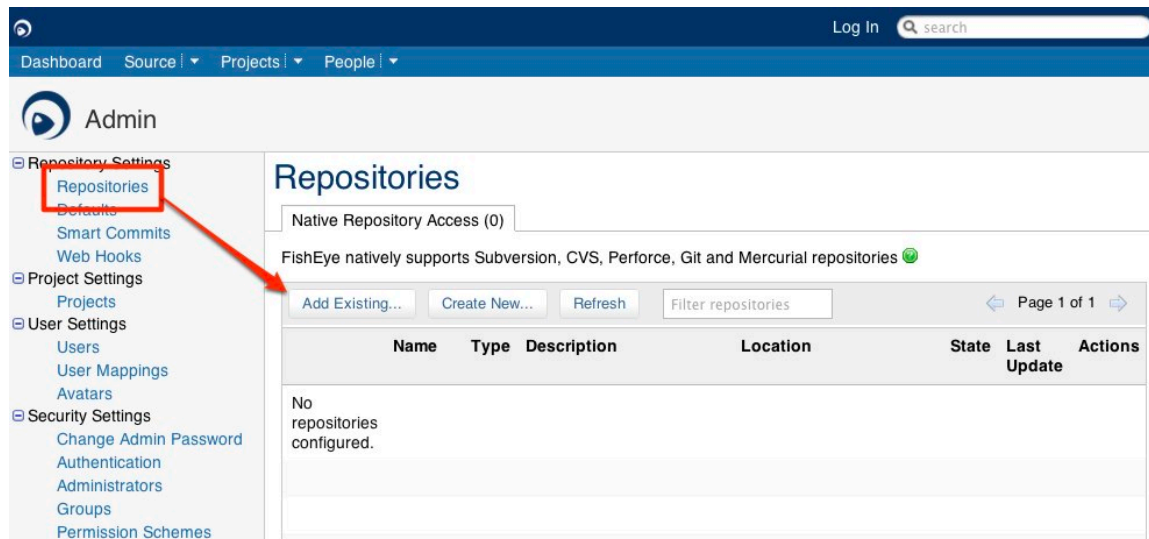
List of users

[Return to the list of users](#)

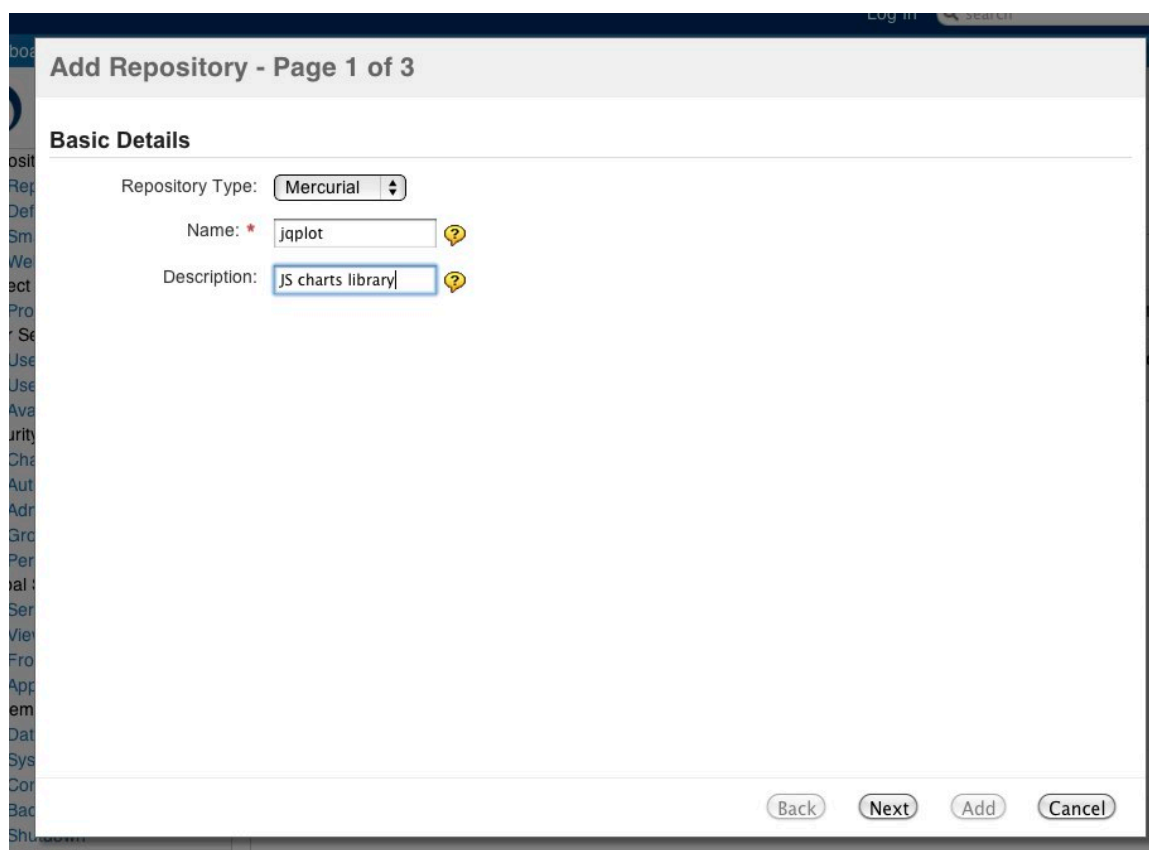
Add a repository

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

Add Repository - Page 2 of 3

Mercurial Connection Details

Repository Location: * ?

Mercurial Authentication

Authentication Style: ?

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

☐ Show advanced settings

Finally indicate whether or not you would like diff indexing should be turned on and if the repository should not be indexed right away.

Click **Add** to finish the process.

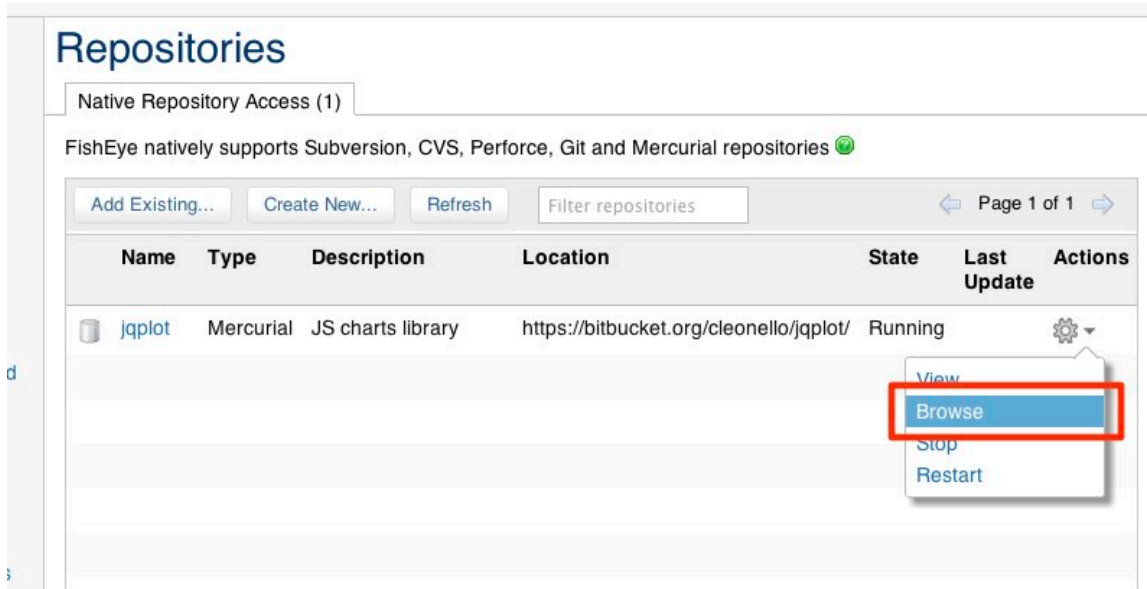
Add Repository - Page 3 of 3

Final Settings

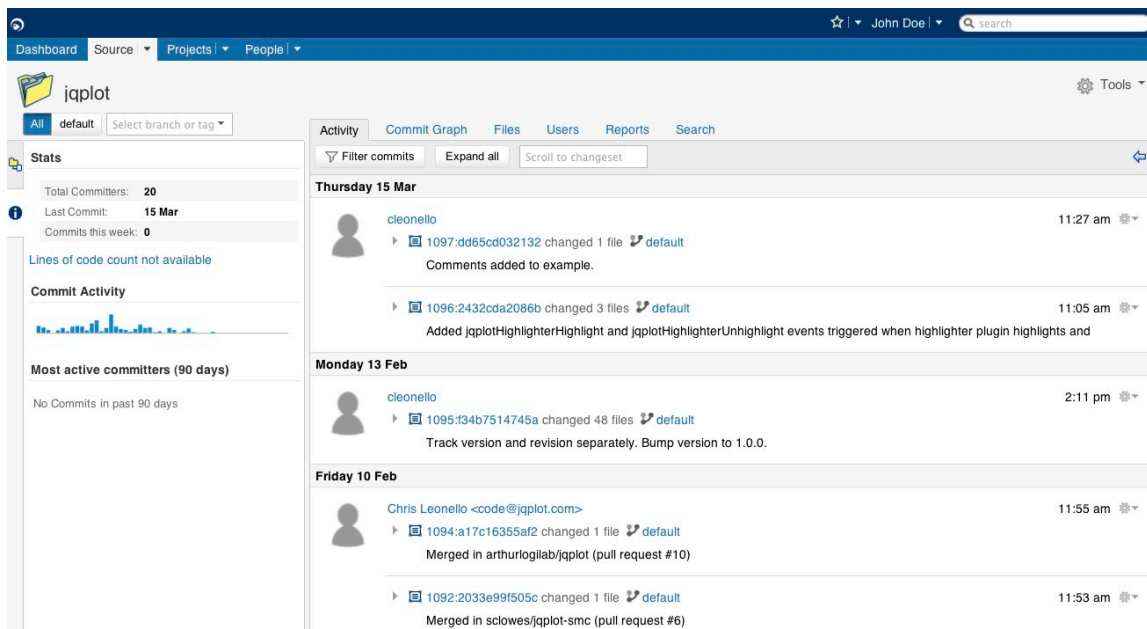
Store Diff Info: ☐ ?

Enable Repository After Adding: ☒ ?

Once created you can click on **Browse** from 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.

On this page:

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

Related pages:

- [Starting to use FishEye](#)
- [JIRA Integration in FishEye](#)
- [Connecting to JIRA for user management](#)
- [User Management Limitations and Recommendations](#)

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:

JIRA base URL	The web address of your JIRA server. Examples are: http://www.example.com:8080/jira/ http://jira.example.com
JIRA admin username	The credentials for a user with the 'JIRA System Administrators' global permission in JIRA.
JIRA password	
FishEye base URL	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.
Groups to synchronize	Click Advanced Options to see this field. Select at least one JIRA group to synchronize. The default group is <code>jira-users</code> . JIRA will synchronize all changes in the user information on a regular basis. The default synchronization interval is 1 hour.
Admin Groups	Click Advanced Options to see this field. Specify a JIRA group whose members should have administrative access to FishEye/Crucible. The default group is <code>jira-administrators</code> .

3. Click **Connect to JIRA**.
4. Finish the setup process.

Troubleshooting

▼ Click to see troubleshooting information...

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

Symptom	Cause	Solution
The setup wizard displays one of the following error messages: - Failed to create application link from JIRA server at <URL> to this <application> server at <URL>. - Failed to create application link from this <application> server at <URL> to JIRA server at <URL>. - Failed to authenticate application link from JIRA server at <URL> to this <application> server at <URL>. - Failed to authenticate application link from <application> server at <URL> to this JIRA server at <URL>.	The setup wizard failed to complete registration of the peer-to-peer application link with JIRA. JIRA integration is only partially configured.	Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.

The setup wizard displays one of the following error messages: - Failed to register <application> configuration in JIRA for shared user management. Received invalid response from JIRA: <response> - Failed to register <application> configuration in JIRA for shared user management. Received: <response>	The setup wizard failed to complete registration of the client-server link with JIRA for user management. The peer-to-peer link was successfully created, but integration is only partially configured.	Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.
The setup wizard displays the following error message: Error setting Crowd authentication	The setup wizard successfully established the peer-to-peer link with JIRA, but could not persist the client-server link for user management in your <code>config.xml</code> 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: <code>java.lang.IllegalStateException: Could not create the application in JIRA/Crowd (code: 500)</code>. 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:

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

Or:

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

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

- Go to the following URL in your browser:

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

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

For example:


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

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

Solution 2: Removing a Partial Configuration – The Longer Way

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

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

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

For example:

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

Or:

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

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

- Go to the following URL in your browser:

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

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

For example:

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

- The application links manifest will appear. Check the application ID in the `<id>` element.
 - c. In JIRA, click **Delete** next to the application that you want to remove.
6. Add the application link in JIRA again, so that you now have a two-way trusted link between JIRA and your application:
 - a. Click **Add Application Link**. Step 1 of the link wizard will appear.
 - b. Enter the **server URL** of the application that you want to link to (the 'remote application').
 - c. Click the **Next** button.
 - d. Enter the following information:
 - **Create a link back to this server** – Tick this check box to add a two-way link between the two applications.
 - **Username** and **Password** – Enter the credentials for a username that has administrator access to the remote application.
Note: These credentials are only used to authenticate you to the remote application, so that Application Links can make the changes required for the new link. The credentials are not saved.
 - **Reciprocal Link URL** – The URL you give here will override the base URL specified in your remote application's administration console, for the purposes of the application links connection. Application Links will use this URL to access the remote application.
 - e. Click the **Next** button.
 - f. Enter the information required to configure authentication for your application link:
 - **The servers have the same set of users** – Tick this check box, because the users are the same in both applications.
 - **These servers fully trust each other** – Tick this check box, because you trust the code in both applications and are sure both applications will maintain the security of their private keys.
For more information about configuring authentication, see [Configuring Authentication for an Application Link](#).
 - g. Click the **Create** button to create the application link.
 7. Configure a new connection for user management in JIRA:
 - a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:
 - In JIRA 4.3: Click **Other Applications** in the **Users, Groups & Roles** section of the JIRA administration screen.
 - In JIRA 4.4: Select **Administration** > **Users** > **JIRA User Server**.
 - b. **Add** an application.
 - c. Enter the **application name** and **password** that your application will use when accessing JIRA.
 - d. Enter the **IP address** or addresses of your application. Valid values are:
 - A full IP address, e.g. 192.168.10.12.
 - A wildcard IP range, using CIDR notation, e.g. 192.168.10.1/16. For more information, see the introduction to [CIDR notation on Wikipedia](#) and [RFC 4632](#).
 - **Save** the new application.
 8. Set up the JIRA user directory in the application.

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


```
http://www.example.com:8080/jira/
http://jira.example.com
```
 - **Application name** and **Application password** – Enter the values that you defined for Confluence in the settings on JIRA.
 - e. Save the directory settings.
 - f. Define the **directory order** by clicking the blue up- and down-arrows next to each directory on the '**User Directories**' screen.

For details see [Connecting to Crowd or JIRA for User Management](#).
- For FishEye/Crucible:
 - a. Click **Authentication** (under 'Security Settings').
 - b. Click **Setup JIRA/Crowd authentication**. Note, if LDAP authentication has already been set up, you will need to remove that before connecting to JIRA for user management.
 - c. Make the following settings:

Authenticate against	Select a JIRA instance
Application name and password	Enter the values that you defined for your application in the settings on JIRA.
JIRA URL	The web address of your JIRA server. Examples: <pre>http://www.example.com:8080/jira/ http://jira.example.com</pre>
Auto-add	Select Create a FishEye user on successful login so that your JIRA users will be automatically added as a FishEye user when they first log in.
Periodically synchronise users with JIRA	Select Yes to ensure that JIRA will synchronize all changes in the user information on a regular basis. Change the value for Synchronise Period if required.
When Synchronisation Happens	Select an option depending on whether you want to allow changes to user attributes from within FishEye.

Single Sign On	Select Disabled . SSO is not available when using JIRA for user management and if enabled will make the integration fail.

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


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

Notes

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

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

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 the FishEye screen will remain the same as you browse through the different screens. You can do the following,

- Click the **'Dashboard'** tab to see your personal code commits; your reviews (if you are using [Crucible](#)); and your issues (if you are using [JIRA](#)).
- Click the **'Source'** tab to see the following sub-tabs:
 - **'Repositories'** — the list of all FishEye repositories. Click a repository name to [browse the repository](#). A number of sub-tabs are then available as described below (see *'Repository Sub-Tabs'*).
 - **'Activity'** — sub-tabs allow you to see the following across all repositories, for all users: code commits; reviews (if you are using [Crucible](#)); and issues (if you are using [JIRA](#)).
- Click the arrow next to the **'Source'** tab to open the dropdown menu. *Logged-in users* will be able to see a list of links to repositories they have recently visited.
- *If you are using Crucible:* Click the **'Projects'** tab to see a list of all projects (see the [Crucible documentation](#)). Click the arrow to open the dropdown menu. *Logged-in users* will be able to see a list of links to projects they have recently visited.
- Click the **'People'** tab to view statistics about committers to your FishEye repositories (see [Viewing People's Statistics](#)). Click the arrow to open the dropdown menu. *Logged in users* will be able to see a list of links to user pages they have recently visited.
- *If you are using Crucible:* Click the **'Reviews'** tab to go to your code reviews (see the [Crucible documentation](#)). Click the arrow to open the dropdown menu. *Logged-in users* will be able to see a list of links to reviews they have recently visited as well as links to the Crucible *'Inbox'* and *'Outbox'*.
- Click the star icon to view your favourite repositories, folders and/or files (see [Using favourites in FishEye](#)).
- Click your name 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:

Sub-Tab Name	Description
Revisions	Shows the latest revisions of the file. See Viewing a File History .
Files	Shows the contents of the directory.
Activity	Shows recent activity on the item. There are a number of sub-options here (see Viewing the changelog): • All Activity — The default view, showing commits, reviews¹ and JIRA issues². • Commits — Shows commits in the activity stream. • Reviews¹ — Shows review activity in the activity stream. • Scroll to Changeset — Opens the changeset ID specified in the text field (press Enter to carry out the action). • Filter — Applies constraints to the current activity stream. • Show Revisions — If this is selected, then changeset items are automatically expanded to show modified files. • Earlier Activity (Left Arrow icon) — Loads a page of earlier activity. • Later Activity (Right Arrow icon) — Loads a page of later activity. ¹ If you are using Crucible ² If you are using JIRA
Users	Shows the commit history of the different users that have committed changes on the item.
Reports	Shows activity charts for the item. Various chart options can be selected in the left navigation bar (see FishEye Charts).
Source	Shows the contents of the file.
Query	Allows you to run an advanced search .

Screenshot: The Repositories View

Repositories

Native Repository Access (8) Plugins (5)

FishEye and Crucible natively supports Subversion, CVS, ClearCase, Perforce, Git and Mercurial repositories 🟢

Name	Type	Description	Location	State	Last Update
AGMP	Subversion	a good marketing play	https://studio.atlassian.com/svn/AGMP	Disabled	
applinks	Subversion		https://studio.atlassian.com/svn/APL	Disabled	
atlassian	Subversion		https://svn.atlassian.com/svn/public/atlassian	Running	58 seconds ago
BM	Subversion	Build Monitor	https://labs.atlassian.com/svn/BM	Disabled	
CFS	Subversion		https://studio.plugins.atlassian.com/svn/CFS	Running	50 seconds ago
CGIT	Subversion	Git LightSCM Plugin	https://studio.plugins.atlassian.com/svn/CGIT	Running	50 seconds ago
CLOV	Subversion	Clover	https://studio.atlassian.com/svn/CLOV	Running	28 seconds ago
FE	Subversion	FishEye	https://studio.atlassian.com/svn/FE	Running	5 seconds ago

Info
Indexing has commenced for repository FE

Add...

Screenshot: The Activity View

My Reviews

Inbox (12)

- To Review (4)
- To Summarize (1)
- In Draft (7)
- Require My Approval (0)

Outbox (0)

- Out For Review (0)
- Completed (0)

Archive (5)

- Closed (5)
- Abandoned (0)

My Snippets

- My Open Snippets (1)
- My Snippets (1)

Committer Mappings

0 committer mappings

All Commits Reviews Issues **Expand all** ☒ Show My Activity

Today



Jake
1 min ago

CR-FE-5390 summarized and closed

hey



Seb Ruiz
1 min ago

CR-FE-5599 summarized and closed



Nick Pellow
1 min ago

Re: CR-FE-5593 replied to Tim Pettersen

done. am going to get this merged to default.

25776:e29071439734 changed 1 file 2.6-FECRU-938-review-key-search 1 min ago

FECRU-938: ensure search for review-key doesn't redirect user. even on the review page with a single result.

+5 -0 /test/plugins/fisheye/.../functest/search/ReviewDetailsSearchFuncTest.java



Jason Hinch
1 min ago

CR-FE-5390 finished reviewing

Screenshot: Repository Sub-Tabs — Viewing a File

FE-hg
build.xml

All default Select branch or tag

- idea/
- settings/
- admin/
- amps-jira-plugin/
- artwork/
- build-dependencies/
- bundled-plugins/
- dbdriver/
- dependencies/
- doc/dac/
- etc/
- fe-jar/
- fe-vs-highlighter-plugin/
- fecru-applinks-webitem-plugin/src/
- fecru-gadgets-plugin/src/
- fecru-rest-plugin/
- fisheye-feedback-plugin/
- lib/
- maven-dependencies/
- plugins/

Activity Revisions Users Reports Source

Diff 2 selected Diff latest Filter Include other branches Show all details 1/50

1476 revisions

06 Jun



Pierre-Etienne Poirot
26271:671adca58a4a on 20pc-FECRU-718-user-mgmt

FECRU-729: use a different constant for devmode, as the current one impact all the plugins

HEAD Source | Diff | Changeset
+1 -1

06 Jun



Tom Davies
26259:34b5d64024ee on 2.7-davies-FECRU-660-upgrade-to-platform-2.10

FECRU-660: upgrade to atlassian platform 2.10

HEAD Source | Diff | Changeset
+5 -6

06 Jun



Jason Hinch
26230:4a05bd93675a on 2.6-FECRU-1054-Quarantine-annotation

FECRU-1054: add Quarantine support to testet aka unit, integration and crucible tests

HEAD Source | Diff | Changeset
+1 -0

06 Jun



Michael Studman
26236:48f27811a53 on 2.6-FECRU-1025-gwt-2.3-upgrade

FECRU-1025: Upgrade GWT to 2.3

HEAD Source | Diff | Changeset
+2 -1

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.

On this page:

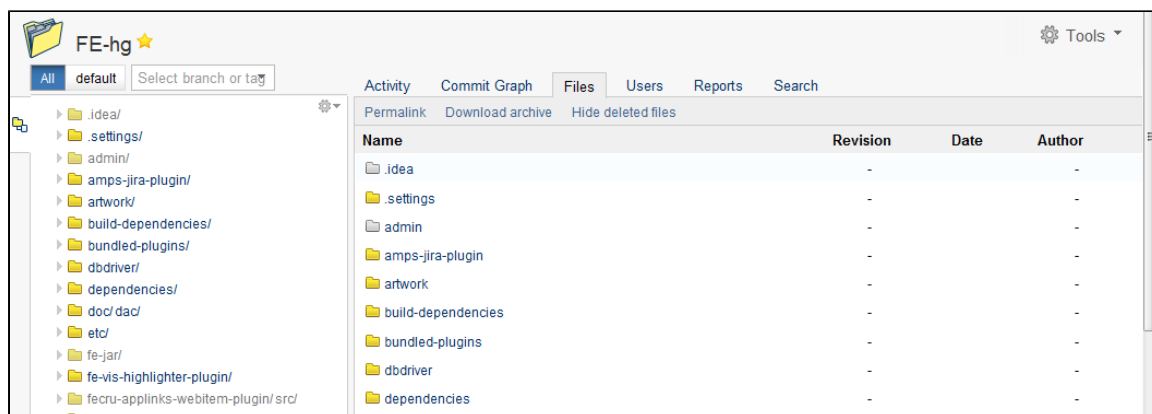
- [Browsing through a repository](#)
- [Hiding empty directories and deleted files](#)
- [Watching a repository](#)

Browsing through a repository**To browse through a repository:**

1. Click the **'Source'** tab. The **'Repositories'** view will be displayed, showing summary information if you have multiple repositories set up.
2. Click the desired repository to view its contents. See the ['Viewing a repository'](#) screenshot below.
3. (optional) Select the [branch](#) or [tag](#) of the repository that you want to view information for, using the branch/tag selector. Only information related to the selected branch/tag will be displayed, when browsing your repositories.
 - Click the tab for the default branch (e.g. 'trunk') to view only information related to it. The default branch (e.g. 'trunk') is determined by FishEye.
 - Click the  icon and enter the desired branch/tag, to view information related to the branch/tag.
 - Click the 'All' tab to view information related to all branches/tags.



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.
 - **'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.

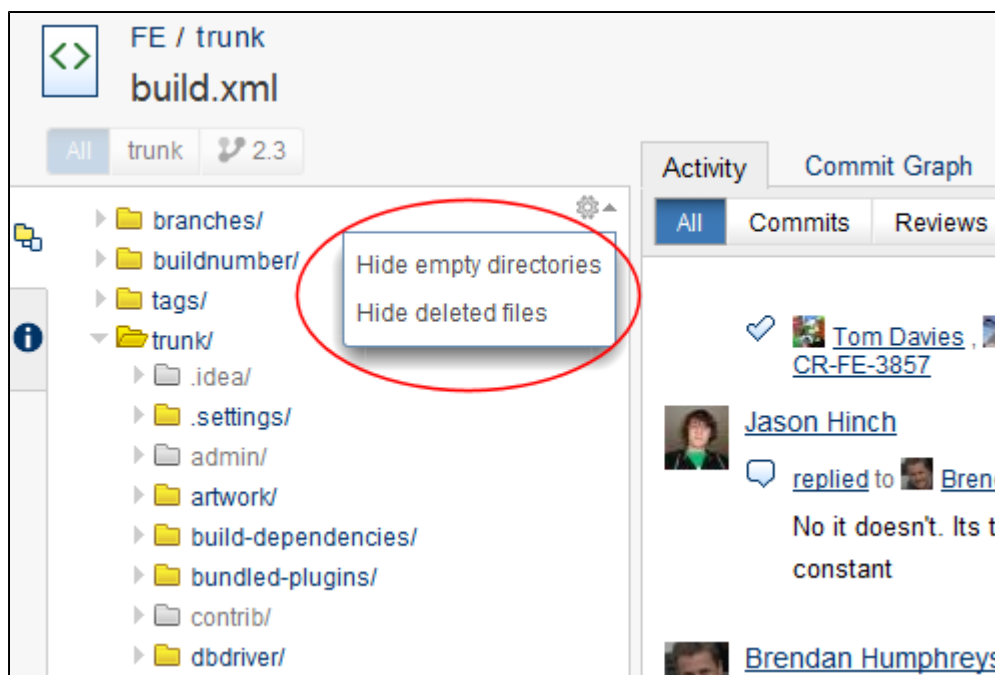
You can choose to hide deleted files and empty directories in the left navigation tree when browsing through a repository, as described below.

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:
 - Click **'Hide empty directories'** to hide all empty (greyed out) directories and their contents (i.e. deleted files and other empty directories).
 - Click **'Hide deleted files'** to 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



Watching a repository

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. Click the **'Tools'** menu and click **'Watch'**. The page will reload and a watch will be set up for the repository

(the watch icon will be coloured, not grey).

- If you want to remove the watch, navigate the repository, click the '**Tools**' menu and click '**Watch**'. The watch will be removed (the watch icon will be coloured, not grey).
-  You can also remove watches via 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

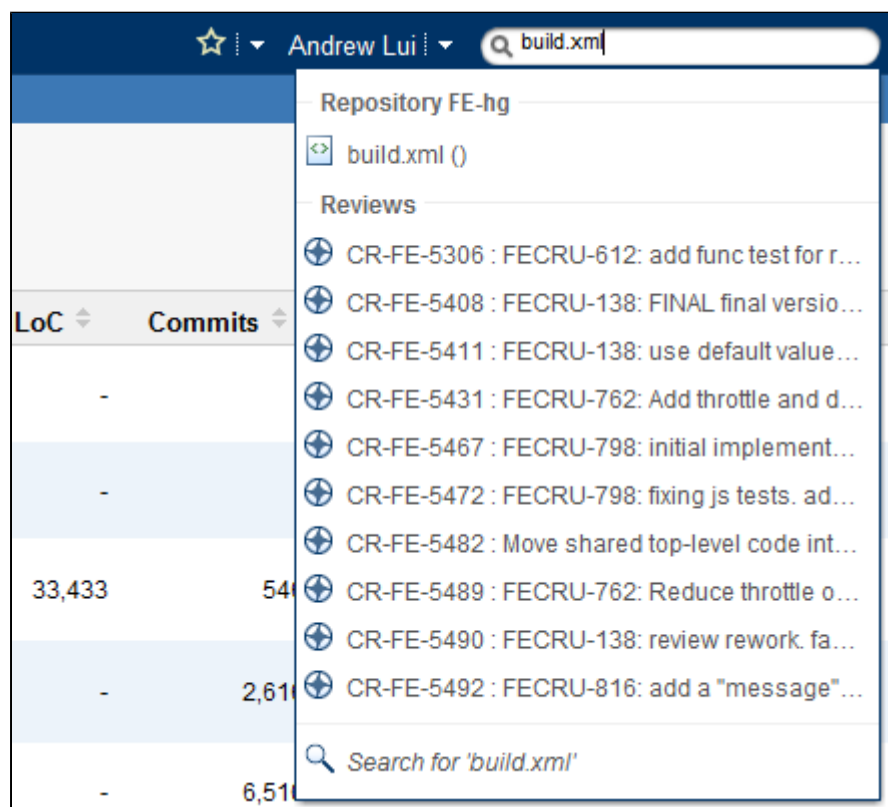
Before you begin:

- 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 [Filtering Quick Search Results](#) below. Results are sorted by relevance and boosted if they were edited recently. A maximum of 10 results are displayed per page.
 - If you have **integrated your 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.



Screenshot above: Quick Search displaying "quick nav" matches

Refining your Quick Search Criteria

The FishEye Quick Search has a number of powerful tools that you can use to refine your search criteria before executing the search.

Search Tool	Description	Example
CamelCase Pattern Matching	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.	Search for BooQCTest and FishEye will return results like BooLeanQueryCoordTest and BooleanQueryClassTest.
Path/File Pattern Matching	Enter a path/file pattern and FishEye will find files and directories that match the pattern.	Search for common/final/Actions and FishEye will return results like /src/common/eu/systemworks/specialprojects/final/Actions.java.

Field Handles	Use a field handle in your criteria to restrict your search to a particular field. Note, you cannot have multiple field handles in a query. • <code>file</code> — Search against a file/directory name only. • <code>commit</code> — Search against a commit message only. • <code>diff</code> — Search against a diff (lines added/removed) only. • <code>content</code> — Search against contents of a file only. • <code>committer</code> — Search against a committer only.	Search for <code>file:build.xml</code> and FishEye will return files that have a name matching <code>build.xml</code> .
Searching within a Directory (AntGlobs)	AntGlobs can be used in the Quick Search to help search within a specific directory.	Search for <code>/src/**/gwt/*.xml</code> and FishEye will return all files with a <code>.xml</code> suffix that are below both a <code>src</code> and a <code>gwt</code> directory. e.g. <code>src/java/com/atlassian/fecru/gwt/FecruCore.gwt.xml</code> but not <code>src/java/com/atlassian/fecru/ApplicationContext.xml</code>
Searching for Discrete Strings	Enter a specific string within quotation marks and FishEye will match against the exact string. Note, this search is not case-sensitive.	Enter "update version in build" and FishEye will only return results that match that exact string, i.e. it will not return a result with <code>update build version</code> or <code>update version in file</code> .

Filtering Quick Search Results

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

Filter	Description
Repositories	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.  <i>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.</i>

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.

Search Criteria

☐ Search HEAD only

Comments:

Contents:

Added Text:

Removed Text:

Author(s):
 Brendan Humphreys (b)
 conor
 daniel
 matt

File name
(Ant-globs):

Branch:

Tag:

After:

Before
(YYYY-MM-DD):

Group by:

Display data:

☒ Path ☒ Revision ☒ Author ☒ Date
☒ Changeset ID ☐ Comment ☐ Total Lines
☐ Lines Added ☐ Lines Removed ☐ Tags
☐ Reviews

[Switch to EyeQL Search](#)

Screenshot: Advanced Search Criteria — Standard Search Interface

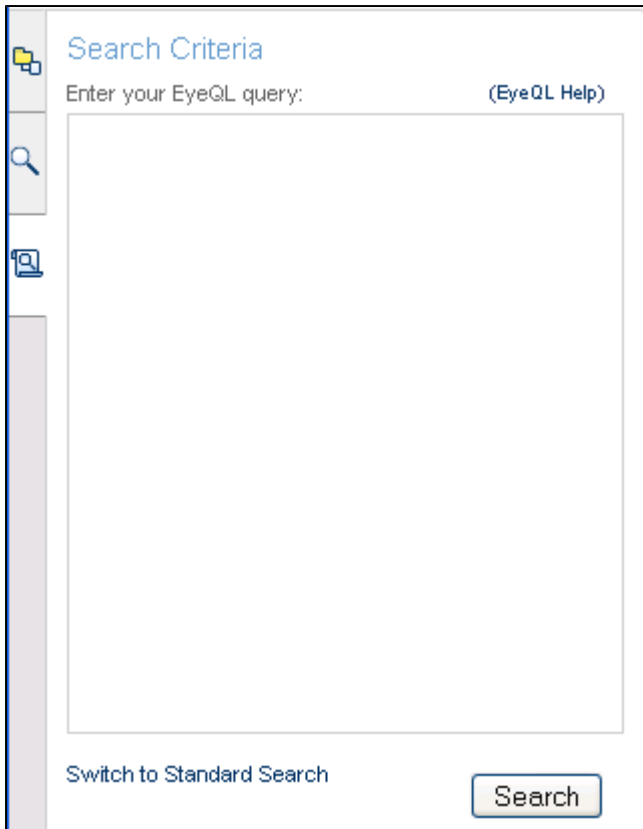
Please note the following:

- **'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.



Screenshot above: Advanced Search Criteria — EyeQL Interface

Notes

Related Topics

[Searching Crucible](#) (Crucible documentation)

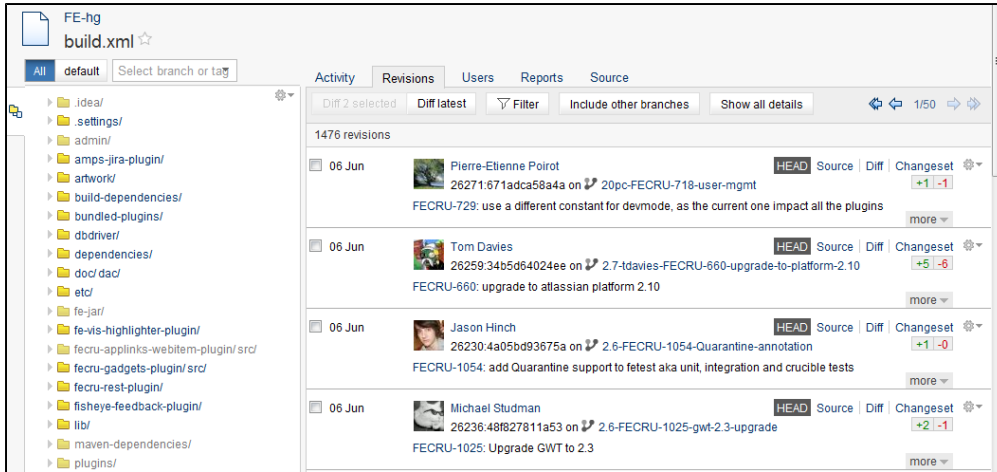
Viewing a File

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 for a file or browse through a repository to find the file.
2. Click the file name. The file will be displayed, showing the revision history. See screenshot below.
3. View information about the file:
 - **'Activity'** tab — View the changelog for the file. This is the commits, reviews and issues (requires JIRA) activity related to the file. See [Viewing the changelog](#) for more information.
 - **'Revisions'** tab — View the history of revisions for the file. See [Viewing a File History](#) for more information.
 - **'Users'** tab — View the commit history of the users that have committed changes to the file. See [Viewing People's Statistics](#) for more information.
 - **'Reports'** tab — View activity charts for the file. See [FishEye Charts](#) for more information.
 - **'Source'** tab — View the annotated file contents. The raw file can be downloaded via this tab. See [Viewing File Content](#) for more information.

Screenshot: Viewing a File (Revisions)



Related Topics

- [Viewing File Content](#)
- [Viewing a File History](#)
- [Viewing the changelog](#)

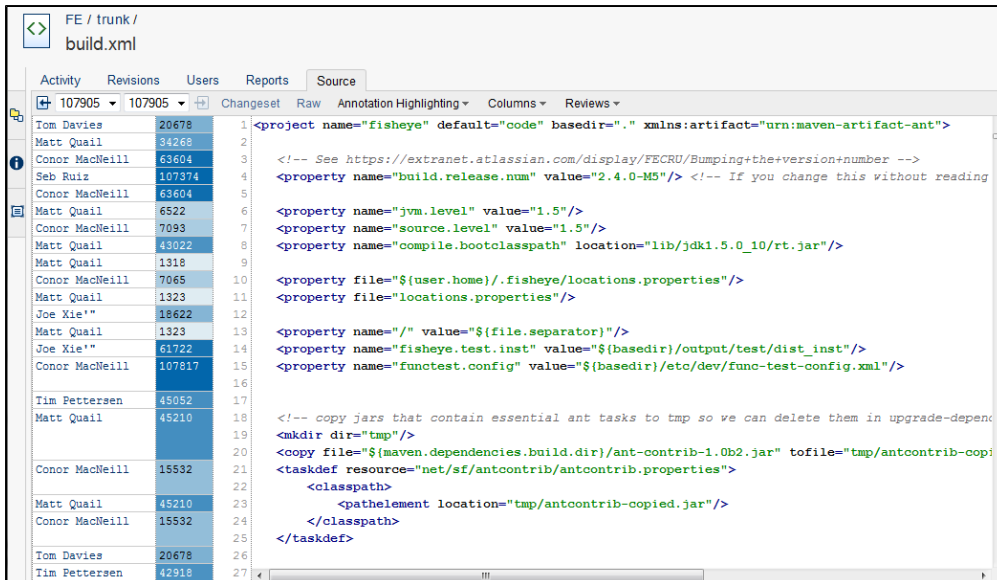
Viewing File Content

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 for a file or browse through a repository to find the file.
2. Click the file name. The file will be displayed, showing the revision history.
3. Click the **'Source'** tab. The source code for the file will be displayed.
 - Select the revision numbers (e.g. '107905') from the two revision dropdowns to [display the diff](#) between the two revisions. By default, the latest revision of the source code is displayed (without any diff).
 - Click the **'Changeset'** link to view the changeset that the revision was a part of.
 - Click the **'Raw'** link to download the raw source code for the file.
 - Click the **'Annotation Highlighting'** dropdown and select **'Age'**, **'Author'** or **'None'** to colour the annotations by age, author or remove highlighting respectively. The highlights are displayed over the revision numbers, next to the authors.
 - Click the **'Columns'** dropdown and select the columns to display: **'Author'**, **'Revision'** and **'Line Number'**.
 - Click the **'Reviews'** dropdown and select **'Create Review'** to create a Crucible review from the file.

Screenshot: Viewing a File (Source)



Using Side by Side Diff View

This page contains instructions for FishEye's innovative '**side by side diff**' view. This is an assistive source code viewing mode where you can see how a file's content has changed, compared on the left and right hand sides of the screen.

On this page:

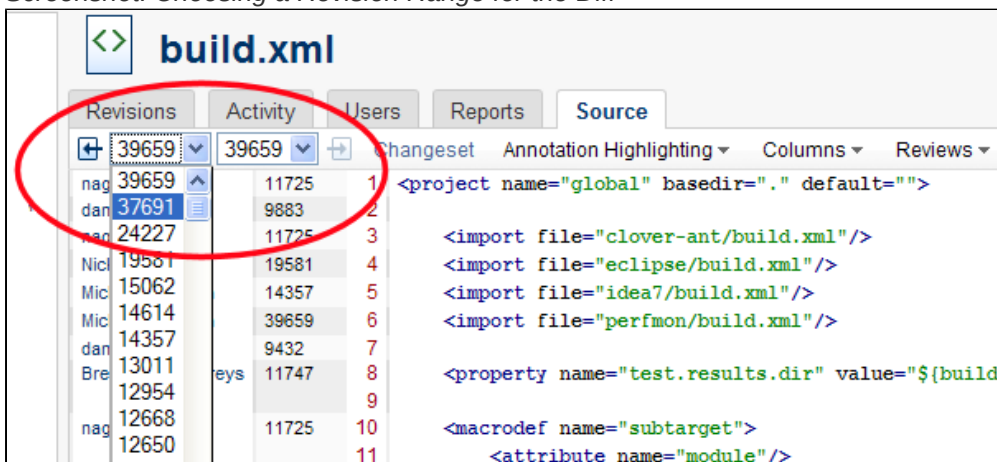
- [Opening side by side diff view](#)
- [Understanding side by side diffs](#)
- [Alternative ways to open side by side diffs](#)
 - [From the FishEye Dashboard](#)
 - [From the Revisions History view](#)

Opening side by side diff view

To open FishEye's side by side diff view:

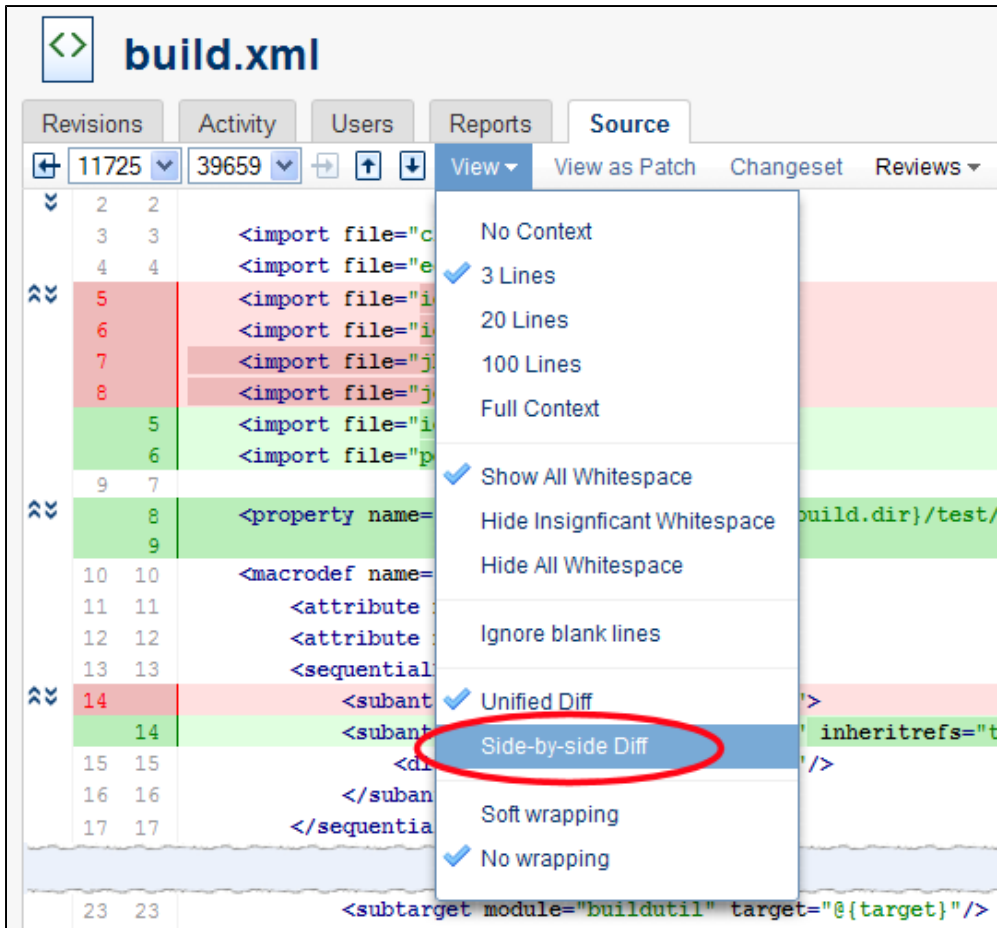
1. Open the source code view for the file in question.
2. Select a range of revisions to compare between.

Screenshot: Choosing a Revision Range for the Diff



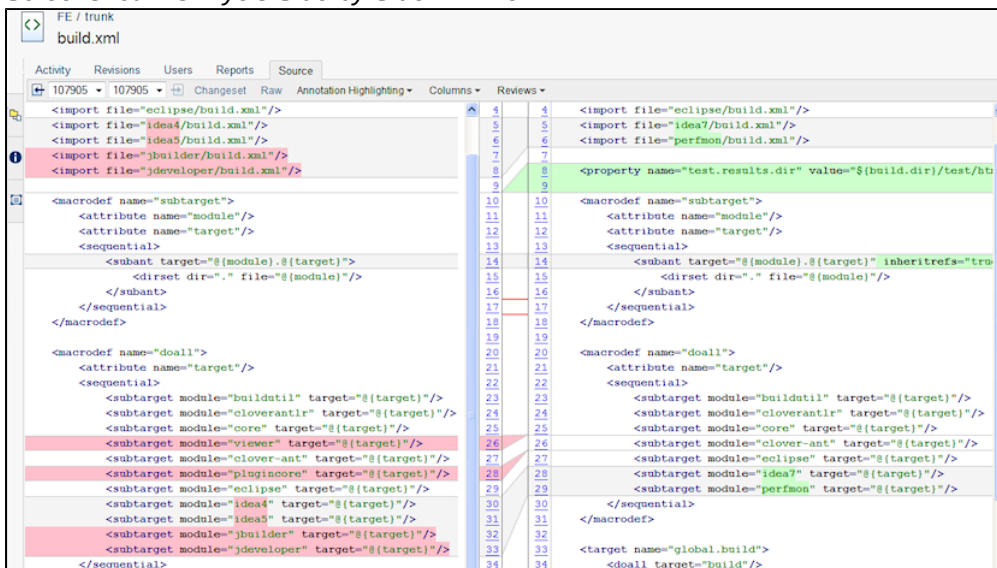
3. Click the '**View**' menu, then select '**Side by Side Diff**'.

Screenshot: Choosing Side by Side Diff Mode



4. **Side by side diff view** opens. The left and right panes 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 opposite window by a coloured triangle that links to the location of that change in the counterpart file.

Screenshot: FishEye's Side by Side Diff View



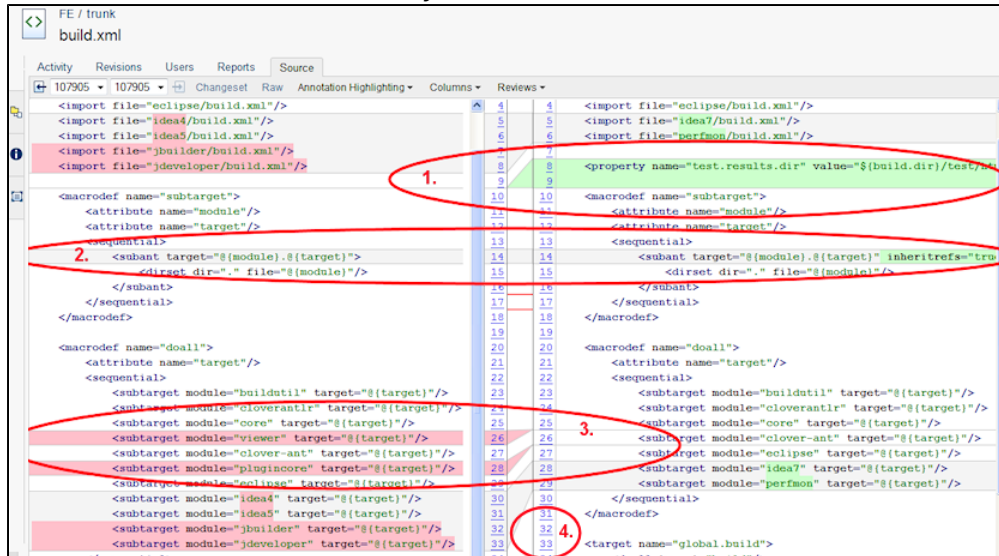
Understanding side by side diffs

Features of the side by side diff screen 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. Gutter line numbers are permanent links ("permalinks") that can be saved and 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. The Dashboard will be displayed.
2. Search for the desired file or browse to it as follows:
 - a. Click the **'Source'** tab in the header at the top of your screen. The list of repositories set up in your FishEye instance will be displayed.
 - b. Click the name of the repository that your file is located in.
 - c. Browse to the file using the tree in the left menu and click the file name.
3. Click the **'Revisions'** tab. The history of revisions for the file will be displayed. See the ['File Revisions'](#) screenshot below.
 - Tick the checkboxes next to two file revision and click **'Diff 2 selected'** to [view the diff](#) of the two selected file revisions. Click **'Diff latest'** to [view the diff](#) of the two most recent file revisions.

- Click **'Filter'** to view the file filter. Enter the desired fields to filter the file history results on.
- Click **'Include other branches'** to include revisions of the same file on 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.
- Click **'Show all details'** to expand all file revisions to show additional information including the revision ID, parents and the branch where it is head, denoted with this graphic: **HEAD**.
- See the ['Overview of a File Revision'](#) diagram below for information about the individual revisions.

Screenshot: File Revisions

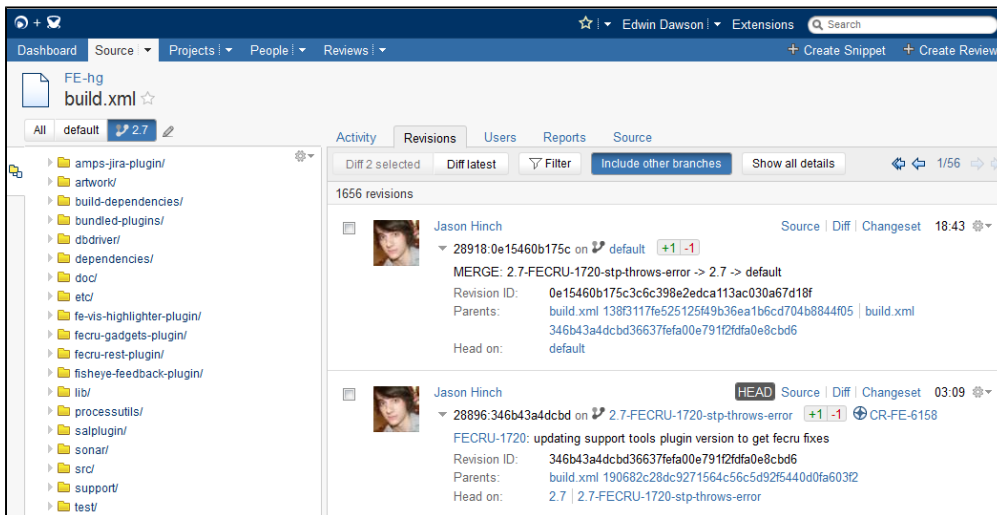
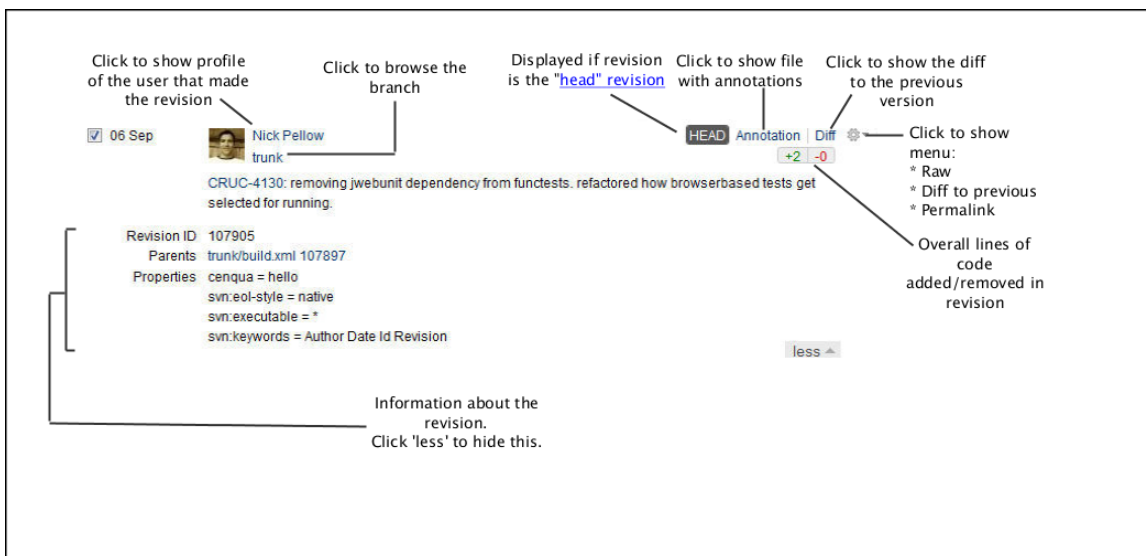


Diagram: Overview of a File Revision



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 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:

All	Click to show commits, reviews (requires integration with Crucible) and JIRA issues (requires integration with JIRA) activity in the activity stream.
Commits	Click to show only commits in the activity stream.
Reviews	Show only review activity. (Requires integration with Crucible)
Filter commits	See Filtering commit activity in the changelog (below) for more information.
Expand all	Show all modified files related to each changeset.
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

The screenshot shows the FishEye web interface. The top navigation bar includes links for Dashboard, Source, Projects, People, and Reviews. The main content area is titled 'FE-hg build.xml' and shows the 'Activity' tab selected. The activity stream displays a list of recent changesets, including commits and reviews, with details such as the user, changeset ID, and description. The sidebar on the left shows a file tree structure for the repository.

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:

Committer	Changesets checked in by the given committer/author.
Log Comment	Changesets where the commit comment matches the given text.
File Extension	Changesets that contain files with the specified file extension.
File Name	Changesets that contain a given file.
Start Date	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 '-').
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

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**' sub-tab in the right-hand column 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

You can view chart information controlled by various criteria. Simply select the desired constraints and click the '**Apply**' button.

Setting	Explanation	Values	Default
Branch/Tag selector	Limits the chart to the selected branch/tag.	Any branch/tag from the current repository.	Displays the default/trunk.
Author	Limits the chart to show specific author(s).	Any author configured in the system.	All
Extension	Limits the chart to show specific file type(s).	Any file extension; e.g. ' .java '.	All
Chart type	Changes the chart's presentation.	Area, line, pie or change* chart.	Area
Show by	Secondary data by which to refine the chart.	Subdirectory, author or extension.	None
Start Date	Date of the earliest data to show.	Date in format YYYY-MM-DD.	None (show all)
End Date	Date of the latest data to show.	Date in format YYYY-MM-DD.	None (show all)
Y Axis	Choosing ' Tight ' zooms in the charts view to the limits of the range that the data covers. Only applies to Line charts.	Full or Tight	Tight
Sub Directories	Limit the chart to a folder under the current branch. Files in the current directory are represented by an element labelled ' .(this dir) '.	A single folder.	None (show all)

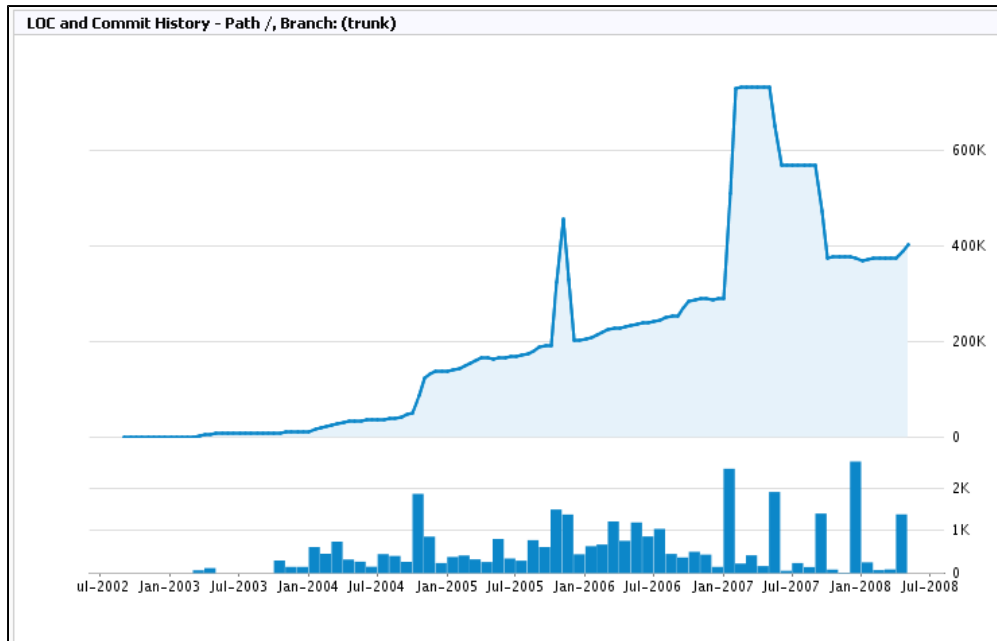
 *The '**change**' chart displays the change in lines of code, for a specific date range, expressed as a line graph. For example, if the lines of code at the start date is 100, the start point will be zero and the rest of the graph shifted by 100 lines.

To return to the default chart settings, click the '**Clear**' button.

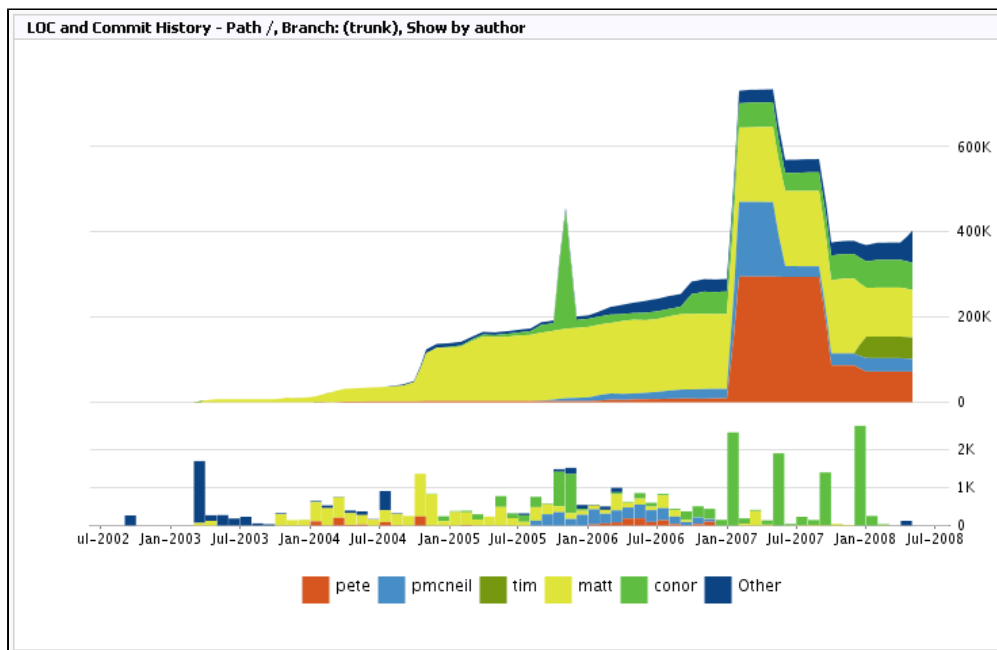
Screenshot: FishEye custom chart settings

The screenshot shows the 'Chart types' and 'Chart options' settings in FishEye. At the top, there are tabs for 'All' and 'trunk', and a dropdown for 'Select branch or tag'. Under 'Chart types', there are radio buttons for 'Area', 'Line' (selected), 'Pie', and 'Change'. Under 'Chart options', there is a 'Break down by:' section with radio buttons for 'None' (selected), 'Subdirectory', 'Author', 'User', and 'Extension'. Below this is a 'Date Range:' section with 'Start Date:' and an empty input field, followed by a colon and another empty input field. There is an 'Author(s):' section with a list box containing '(any)', 'Adam Ahmed (aahmed)', 'Anna Buttfield (abuttfield)', 'Ben Speakmon (bspeakmon)', and 'Brendan Humphreys (bhumphreys)'. Below that is a 'File Extension:' section with a list box containing '(any)', 'None', '.MF', '.TXT', and '.cfs'. An 'Apply' button is at the bottom right.

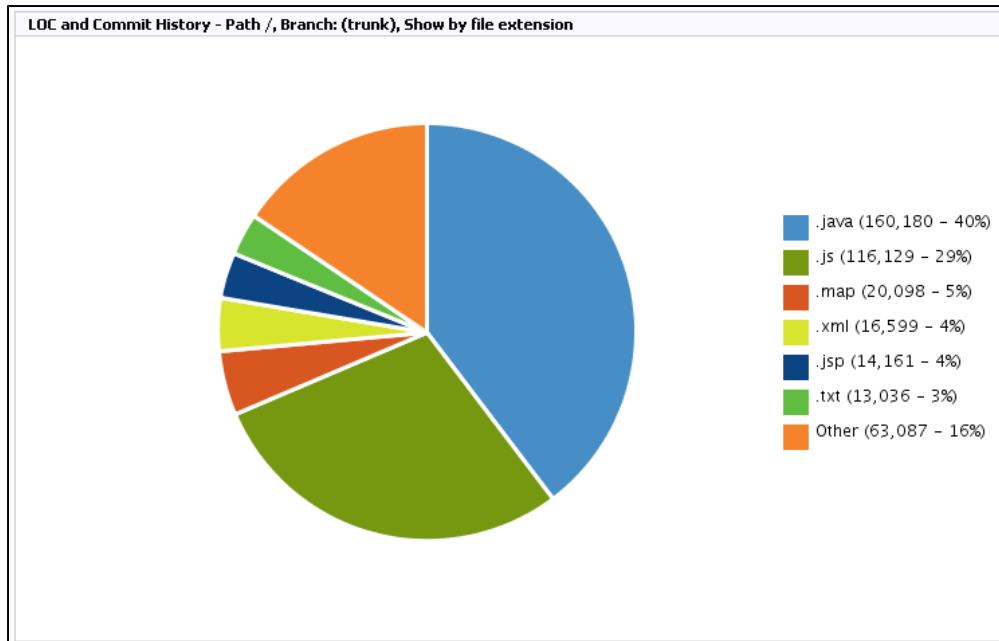
Screenshot: FishEye per-author LOC chart



Screenshot: FishEye per-author LOC chart showing multiple authors



Screenshot: FishEye LOC chart by file extension



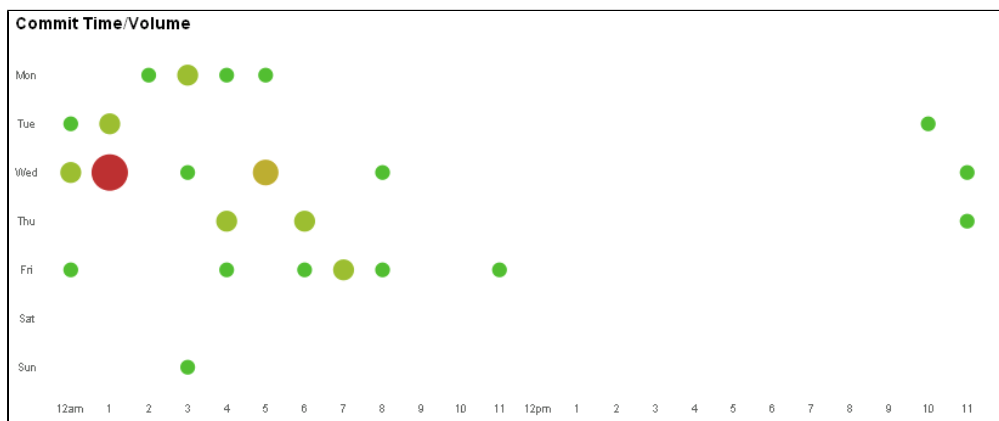
Per-Author Lines of Code Statistics

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.

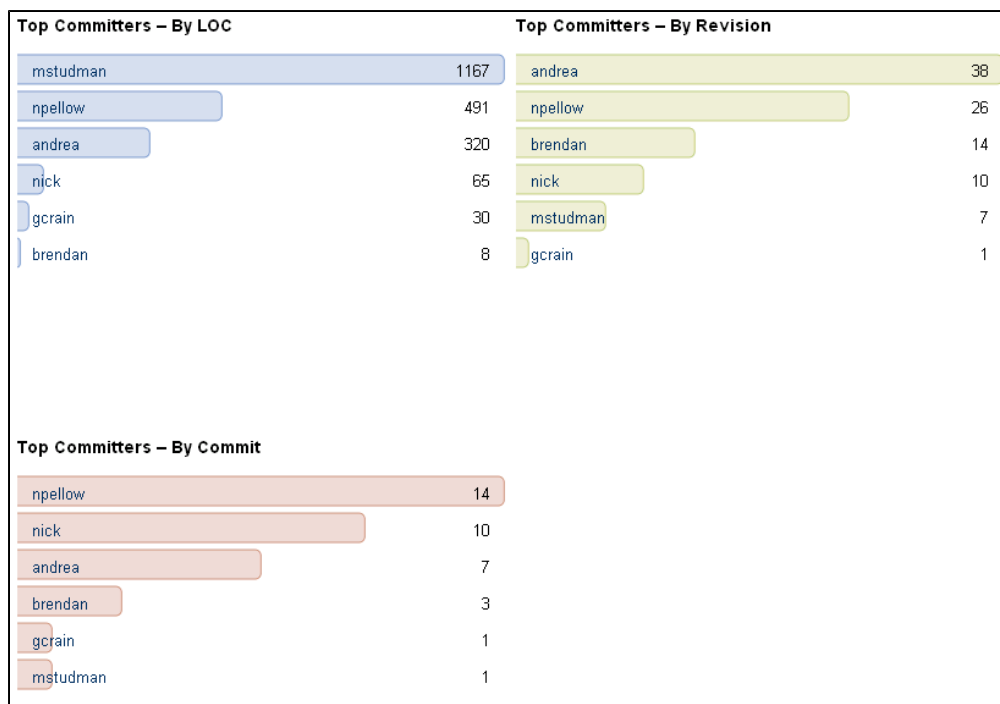
Code Metrics

A number of built-in reports are also provided:

Screenshot: Commit Time/Volume



Screenshot: Top Committers



Notes

Related Topics

[Browsing through a repository](#)

Using favourites in FishEye

FishEye allows you to add changesets, files, people and repositories as favourites. You can view your favourites, or see a stream of all activity relating to your favourites. We suggest that you select items that you are currently working on as favourites, to create a more relevant personalised view.

You can always view your favourites from the menu at the top of the screen, next to your username.

i If you are using Crucible, you can also add [code reviews](#) to your favourites.

On this page:

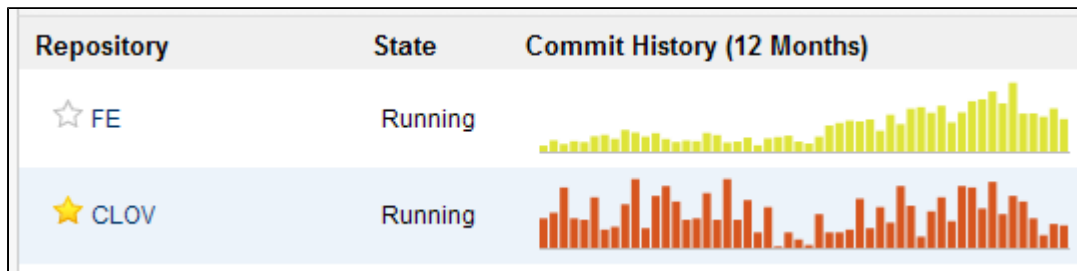
- [Adding favourites](#)
- [Managing favourites](#)

Adding favourites

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

People	Hover the mouse cursor over their avatar or username. In the context menu, click Follow .
Changesets	Open the changeset and click the grey star icon next to its name, near the top of the screen.
Files or folders	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.
Repository	Click the Source tab and then the grey star icon that appears next to the name of the desired repository.

Screenshot: Adding a repository to your favourites



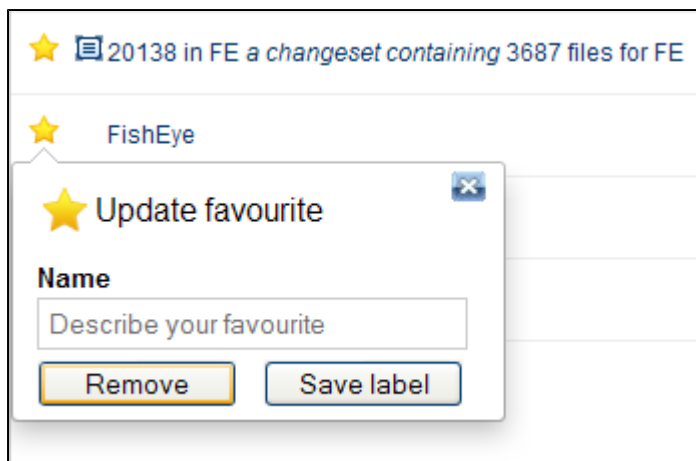
Managing favourites

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 (if you are using Git or Mercurial, you will be able to see anonymous branches as well). You can also use highlights to identify changesets in the same branch, commits with JIRA issues, and reviewed/unreviewed changesets. Clicking a changeset with the appropriate highlight active shows you related changesets, such as changesets with the same lineage, the same JIRA issue or same Crucible review.

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:

- [Before you begin](#)
- [Browsing to the commit graph for a repository](#)
- [Highlighting information in the commit graph](#)
- [Viewing changesets across all branches](#)
- [Reordering swimlanes for Git repositories](#)

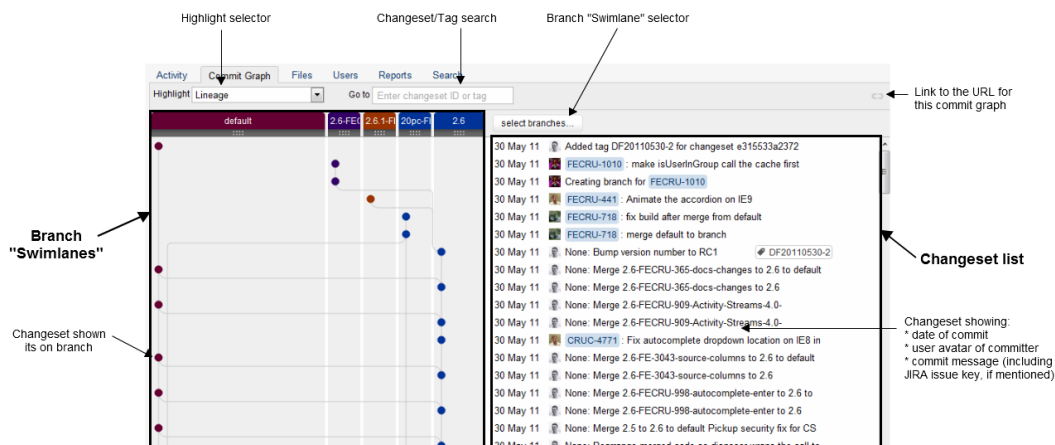
Related pages:

- [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](#)

Browsing to the commit graph for a repository

To view the commit graph for a repository:

- Navigate to the desired repository, as described on [Browsing through a repository](#).
- Click the **'Commit Graph'** tab.
- The commit graph for the repository will be displayed.



Annotated screenshot above: Commit graph for a repository – default view

Highlighting information in the commit graph

The **'Highlight'** dropdown of the commit graph allows you to highlight different types of information in the swimlanes. When a highlight is active, you can also select a changeset in the changeset list to show related changesets. For example, if you have the 'JIRA Issues' highlight active, selecting a changeset with a JIRA issue (in the commit comment) will show which other changesets have the same JIRA issue.

Selecting a changeset (regardless of highlight) will display the following in the changeset list:

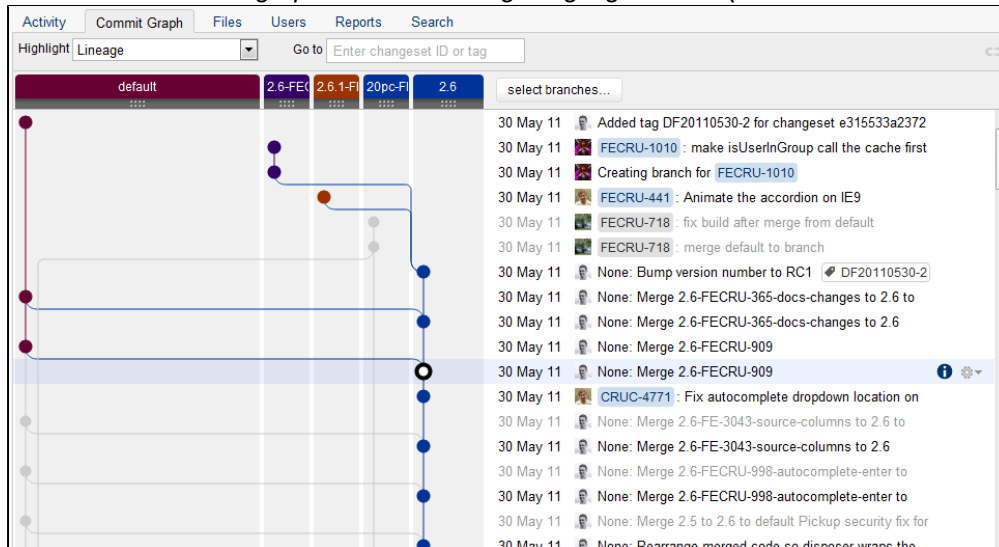
- Display an  icon next to the changeset. Clicking this changeset will display a changeset summary
- Display dropdown menu (accessible via a cog icon) allowing you to view the full changeset, view the changeset in the activity stream or create a review for the changeset.

In this section:

- [Highlighting the lineage of a changeset](#)
- [Highlighting JIRA issues](#)
- [Highlighting reviewed changesets](#)

Highlighting the lineage of a changeset

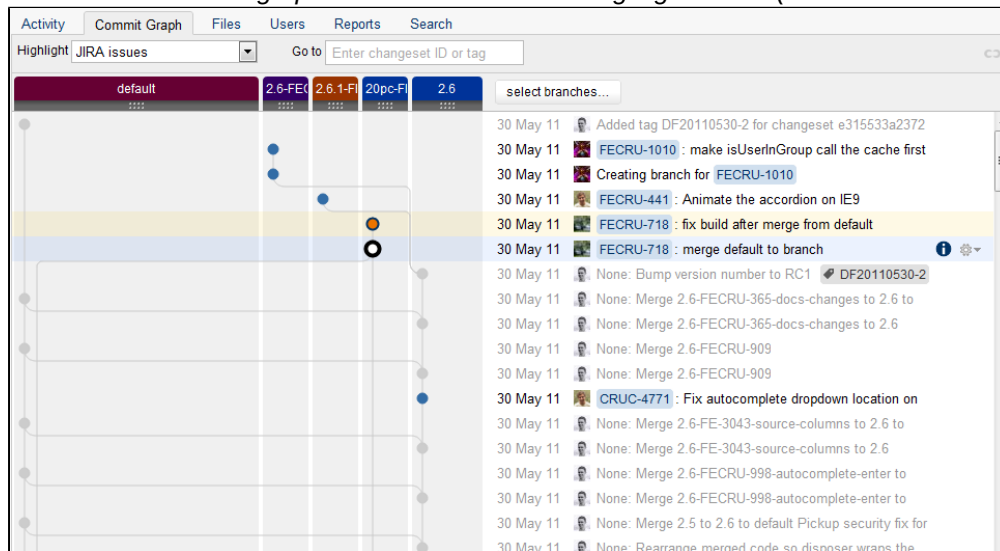
Action	Behaviour
Select the 'Lineage' highlight	Colours the changesets in the swimlanes according to which branch they are in.
Select a changeset in the changeset list with the 'Lineage' highlight active	Shows where a changeset comes from and where it propagates to, i.e. its ancestors and descendants.
Mouseover a changeset in the swimlanes with the 'Lineage' highlight active	Shows a pop-up displaying all branches that the changeset is referenced in. This will include branches that you may not have swimlanes displayed for.

Screenshot: Commit graph with the 'Lineage' highlight active (with mouseover on a changeset)**Highlighting JIRA issues**

This highlight is only available if you have [integrated FishEye with JIRA](#) and [linked your repository to a JIRA project](#).

Action	Behaviour
Select the 'JIRA issues' highlight	Colours the changesets in the swimlanes that have a JIRA issue key in the commit message.
Select a changeset in the changeset list with the 'JIRA issues' highlight active	Colours the changesets in the swimlanes that have the same JIRA issue key in the commit message, as the changeset selected.
Mouseover a changeset in the swimlanes with the 'JIRA issues' highlight active	Shows a pop-up displaying all branches that the changeset is referenced in and all referenced JIRA issues.

Screenshot: Commit graph with the 'JIRA Issues' highlight active (with mouseover on a JIRA issue key)

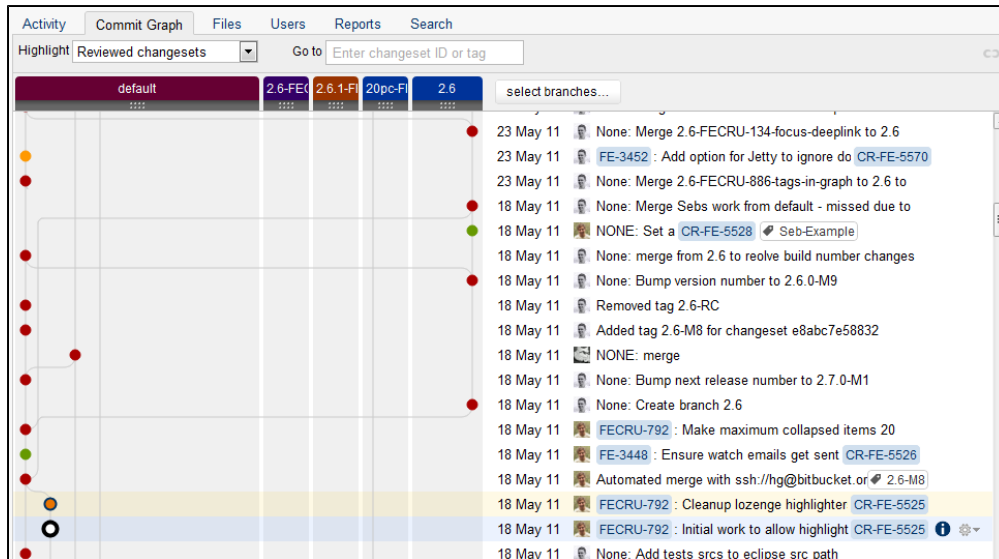


Highlighting reviewed changesets

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

Action	Behaviour
Select the 'Reviewed changesets' highlight	Colours the changesets in the swimlanes that have been reviewed (i.e. included in a Crucible review), as follows: * Red dot: Unreviewed changeset, i.e. associated with review in 'Dead' or 'Rejected' state or no review associated. * Yellow dot: Changeset under review, i.e. associated with review not in 'Dead', 'Rejected' or 'Closed' state. * Green dot: Changeset reviewed, i.e. associated with review in 'Closed' state.
Select a changeset in the changeset list with the 'Reviewed changesets' highlight active	Colours the changesets in the swimlanes that are part of the same review as the changeset selected.
Mouseover a changeset in the swimlanes with the 'Reviewed changesets' highlight active	Shows a pop-up displaying all branches that the changeset is referenced in and the Crucible review key.

Screenshot: Commit graph with the 'Reviewed changesets' highlight active (with mouseover on a changeset)



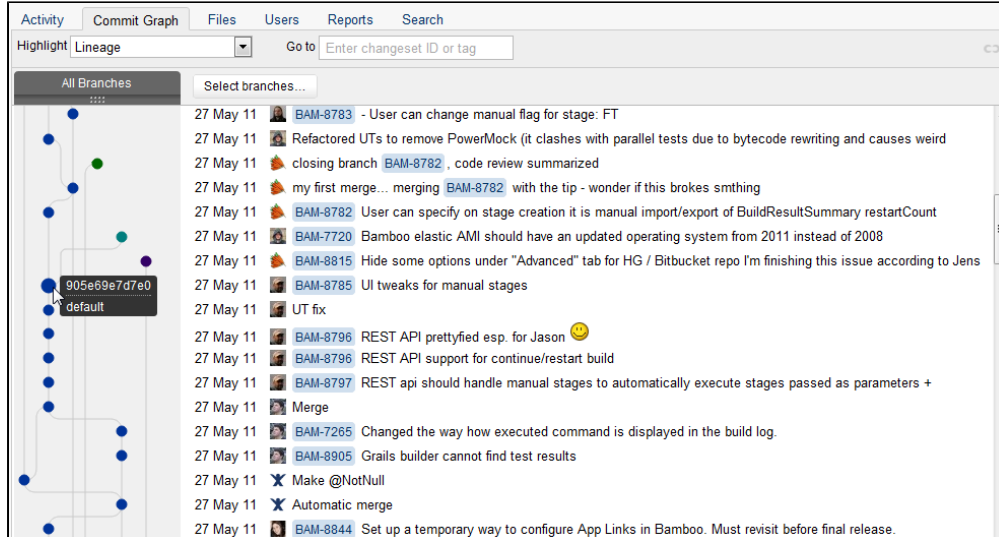
Viewing changesets across all branches

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 mouseover any changeset to display information about the changeset, as described in the 'Highlighting Information in the Commit Graph' section above.

To view the 'All Branches' mode of the commit graph for a repository:

1. Click the **'select branches...'** button when viewing the commit graph.
2. In the 'Select Branches' pop-up, click **'Switch to all branches mode'**.

Screenshot: Commit graph – '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 of determining 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 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.

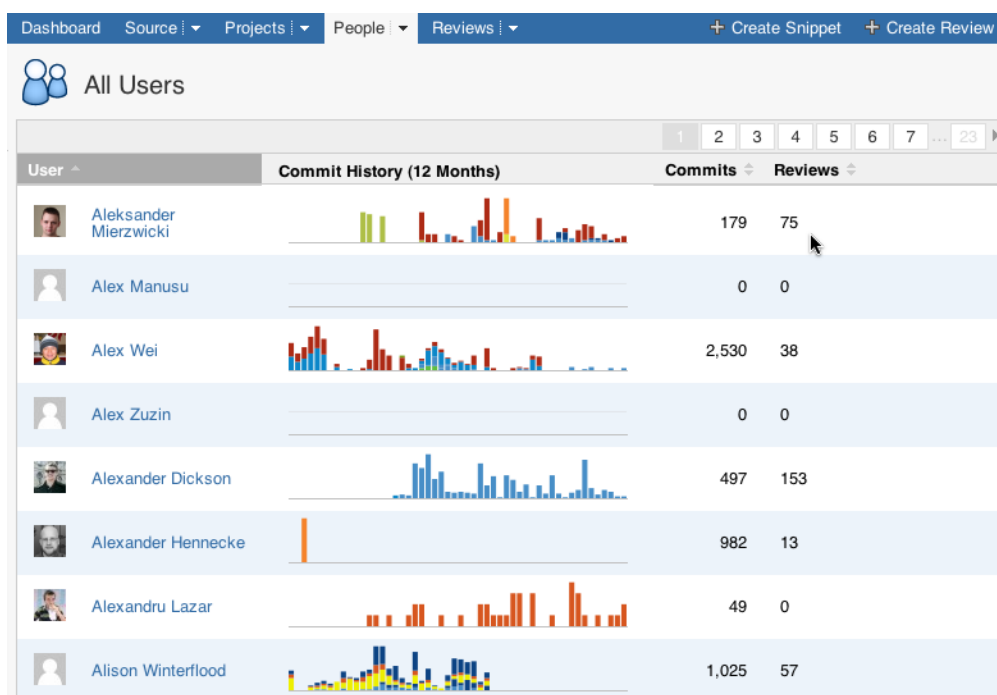
For example, if the 'master' swimlane is to the left of another swimlane, e.g. 'fisheye-2.6' branch, there will be no changesets shown in the 'fisheye-2.6' swimlane, as all the commits will be picked up in the 'master' swimlane. However, if you move the 'fisheye-2.6' swimlane to the left of the 'master' swimlane, it will pick up all of the FishEye 2.6 commits.

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.

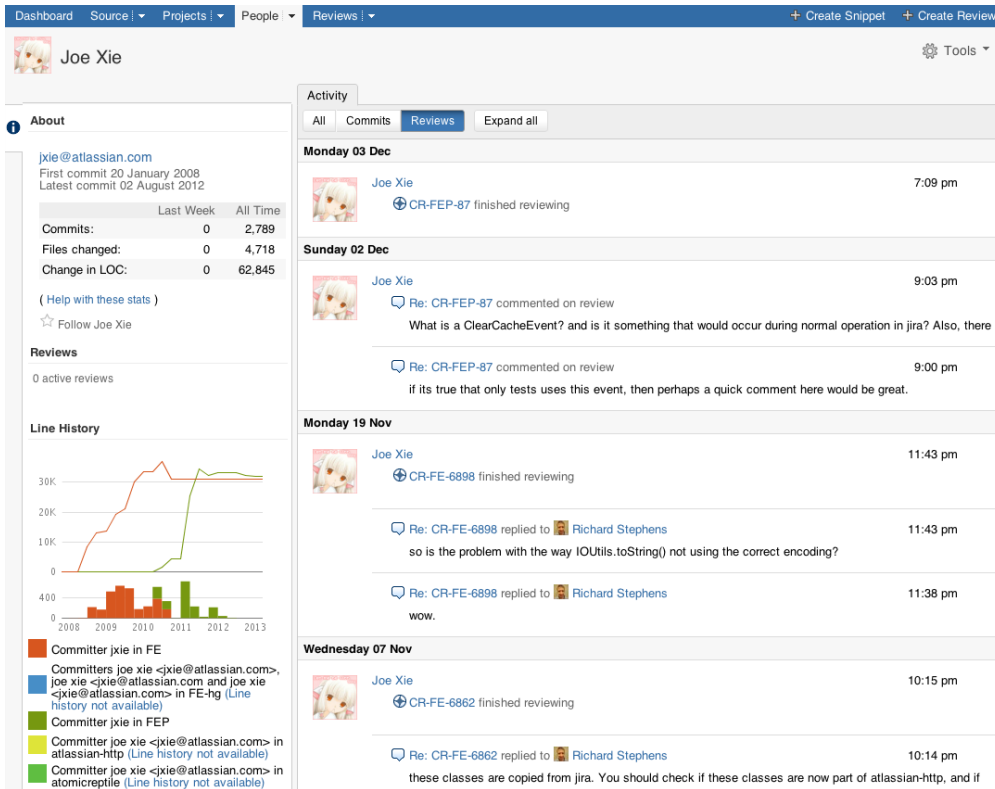
Screenshot: List of all committers in FishEye



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



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

i If you are using [Crucible](#), statistics for each person's code reviews are also available.

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

i 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](#)
 - [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
```

i 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).



Online banking / EBANK-7

A blocker

[Edit](#)
[Assign](#)
[Assign To Me](#)
[Comment](#)
[More Actions](#)
[Resolve Issue](#)
[Close Issue](#)
[Workflow](#)

Details

Type: Bug
 Priority: Blocker
 Affects Version/s: None
 Labels: None

Status: Open
 (View Workflow)
 Resolution: Unresolved
 Fix Version/s: None

Command	Command Parameters	Description	Example

#time	<n>w <n>d <n>h <n>m <work log comment> <i>where <n> is a user-specified time period.</i>	This command 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. <i>Please note:</i> • <i>Work log comments cannot be set using smart commits. See FE-3757.</i>	#time 1w 2d 4h 30m Total work logged — this command would record 1 week, 2 days, 4hours and 30 minutes against an issue, and add the comment 'Total work logged' in the Work Log tab of the issue.
#comment	<comment text>	This command records a comment against an issue.	#comment My comment . — this command would create the comment, "My comment", against the issue.
#<workflow command> e.g. #resolve	<workflow> <comment text>	This command transitions an issue to a particular workflow state. Please see the documentation for Configuring Workflow in JIRA.	#close Fixed the issue — this command would execute the 'Close Issue' workflow transition for an issue in the default JIRA workflow and adding the comment 'Fixed the issue'. #start — this command would execute the 'Start Progress' workflow transition for an issue in the default JIRA workflow

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.

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

Display Settings	File history view mode	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.
	Timezone	Default is the timezone of the FishEye server.
Changelog	Changesets per page	The default is 30 per page.
	Always expand changesets in stream	Default is Yes .
Diff view	Diff mode	Default is Unified .
	Line wrapping	Default is None i.e. long lines will never word-wrap. Soft is when long lines will word-wrap.
	Context lines	Default is 3. The number of lines to show (for context), if the diff contains more than three lines of code.
Source view	Default annotation mode	Default is Age .
	Highlighting colours	The default scheme uses bright colours for highlighting diffs in the code. If you prefer more muted colours, select Classic (muted) .

	Tab width	Default is 8. Can be changed to a number between 1 and 10.

Profile and Email

Email settings	Display Name	Name displayed for the user currently logged in.
	Email Address	The address to which all email notifications will be sent.
	Email Format	Default is Text . Can be sent as HTML .
Email watches	Send Watch Emails	The frequency at which emails will be sent for watch notifications. Immediately is the default value. Daily sends a summary of changes.
Profile Picture	Choose picture	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.

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](#), [changes](#) 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.

Auto-mark files as 'read'	Default is Yes .	
Review Notifications Events	State change	Default is Immediate . A Crucible review moves through different states e.g: 'Draft', 'Under Review'. An email is sent when the state changes.
	Comment added	Default is Immediate . An email is sent when a comment is added to a review.
	Comment reply added	Default is Immediate . An email is sent (to the Moderator only) when any reviewer has completed their review .
	Participant finished	Default is Immediate . An email is sent when a reviewer is added or removed from a review, after it has gone into the 'Under Review' state .
	General message	Default is Immediate . An email is sent when a reviewer is added or removed from a review, after it has gone into the 'Under Review' state .
	File revision added	Default is Immediate .
	Uncomplete review if defect is raised:	Default is Yes . This allows reviews to be resurrected automatically to deal with new code or defects.
	Uncomplete review if revision is added:	Default is Yes . This allows reviews to be resurrected automatically to deal with new code or defects.

	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 e-mail.

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

Login Required

Username:

Password:

[Forgot your password?](#)

Screenshot: The Request New Password screen

Request New Password

To have a new password generated and sent to you, please enter either your username or email address below.

Username:

OR

Email:

Please enter the word as shown below:

1 0 0 i n s

Pattern matching guide

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:

?	Matches one character (any character except path separators)
*	Matches zero or more characters (not including path separators)
**	Matches zero or more <i>path segments</i> .

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

*.txt	Matches /foo.txt and /bar/foo.txt but not /foo.txtty or /bar/foo.txtty/
/*.txt	Matches /foo.txt but not /bar/foo.txt
dir1/file.txt	Matches /dir1/file.txt, /dir3/dir1/file.txt and /dir3/dir2/dir1/file.txt
**/dir1/file.txt	Same as above.
/**/dir1/file.txt	Same as above.
/dir3/**/dir1/file.txt	Matches /dir3/dir1/file.txt and /dir3/dir2/dir1/file.txt but not /dir3/file.txt, /dir1/file.txt
/dir1/**	Matches all files under /dir1/

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:

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

DATECONSTANT can be any of:

now	This very instant (at the time the expression was evaluated).
today todaygmt	The instant at 00:00:00 today. (server-time* or GMT)
thisweek thisweekgmt	The instant at 00:00:00 on the first day of this week. Sunday is considered the first day. (server-time* or GMT)

thismonth thismonthgmt	The instant at 00:00:00 on the first day of this month. (server-time* or GMT)
thisyear thisyeargmt	The instant at 00:00:00 on the first day of this year. (server-time* or GMT)

* 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

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

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:

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* | *linesRemoved* | *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 column. 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:

```
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:

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

Find files on branch and exclude delete files:

```
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:

```
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:

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

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

```
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)

```
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:

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

Find commits that do not have comments:

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

Find the 10 most recent revisions:

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

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

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

Find commits between two dates:

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

Find revisions that do not have any associated review:

```
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:

```
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:

```
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:

```
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:

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

Administering FishEye

You can access the Administration Console at <http://HOSTNAME:8060/admin/>.

The FishEye Administration Console allows you to administer your FishEye instance and manage your repositories. You will also want to read about the [command-line options](#) for controlling FishEye.

 You can disable FishEye's Administration Console by setting `admin-hash=""` in the `<config>` element of `config.xml` before starting FishEye.

We recommend that you run FishEye as a user that has only **read** access to your repository.

If you're running FishEye on Windows you might want to install it as a [service](#) so that FishEye starts up automatically during start-up. See [Run FishEye as a Windows Service](#).

If you intend to use FishEye with [Subversion](#), please ensure you read the [Supported Platforms](#), [Subversion client setup](#), and [granting permission to FishEye](#) to scan your repository.

If you intend to use FishEye with [Perforce](#), please ensure you read the [Perforce client setup](#).

FishEye runs its own HTTP web server, and additionally listens on a socket for administration/shutdown commands. These default to `:8060` and `127.0.0.1:8059` respectively. You can change both of these addresses by editing `config.xml`. You'll need to restart FishEye after doing that.

If you need to reset the administrator password, delete the `admin-hash` attribute in the `<config>` element. You will be prompted to enter an administrator password next time you start FishEye. See [How to reset the Admin password in FishEye or Crucible](#).

One of your first steps will be to [add a repository](#).

Topics

- [Managing your repositories](#)
- [Managing users and groups in FishEye](#)
- [Configuring External Authentication Sources](#)
- [Configuring SMTP](#)
- [Migrating to an external database](#)
- [Backing up and restoring FishEye data](#)
- [Managing add-ons](#)
- [Integrating FishEye with Atlassian applications](#)
- [System Settings](#)
- [Configuring FishEye Security](#)
- [Tuning FishEye performance](#)
- [Contacting Support](#)
- [Configuring the database connection pool](#)

Managing your repositories

You can view the repositories currently set up for your FishEye/Crucible instance in the Administration Console. The list of repositories also provides functions to start/stop repository scans, disable/enable repositories, delete repositories and configure various repository settings. You can also add a new repository from the repository list page.

On this page:

- [Viewing repositories](#)
- [Adding a new repository](#)
 - [External Repositories](#)
 - [Internal Repositories](#)
- [Accessing FishEye repository options](#)

Viewing repositories**To view the repositories set up for your FishEye/Crucible instance:**

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link. The 'Repositories' screen of the administration console will be displayed, with the 'Native' tab showing.
2. Click any of the tabs on the 'Repositories' screen to view the relevant repositories. The number next to the tab title indicates how many of that type of repository are listed under the tab. In most instances, the '**Native**' tab will contain the majority of your repositories.
3. Configure/View the details for individual repositories, as follows:
 - Click the name of the repository to view the repository options. See [Repository Options](#).
 - Click the cog icon  to view the following repository operations in a dropdown menu. Click any option in the dropdown menu to run the operation.
 - '**View**' — View the repository options. See [Repository Options](#).
 - '**Browse**' ('Running' repositories only) — Browse a repository. See [Browsing through a Repository](#).
 - '**Stop**' ('Running' repositories only) — Stop access to the repository and stop the [repository scan](#), if a scan is in progress. See [Operations](#).
 - '**Restart**' ('Running' repositories only) — Allow access to the repository and resume the [repository scan](#), if necessary. See [Operations](#).
 - '**Enable**' ('Disabled' repositories only) — Enable the disabled repository. See [Operations](#).
 - '**Delete**' ('Disabled' repositories only) — **External repositories:** Delete the *FishEye profile* for the disabled repository. See [Operations](#).
 - '**Delete**' ('Disabled' repositories only) — **Internal repositories:** Deletes the *actual repository data* for the disabled repository. See [Operations](#).
 - '**Application Links**' — Configure [application links](#) for the repository.
 - Click **Add** at the bottom of the list of repositories to [add a new repository](#).
 - Click the refresh icon  to refresh the list of repositories. See [Updater](#) for information about polling the repositories.

 If you want to configure the default options for all repositories, click **Defaults** in the left menu. See [Repository Options](#) for more information.

Screenshot: The Repository List in the FishEye Administration Screen

Repositories

Native Repository Access (8)Plugins (5)

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

Name	Type	Description	Location	State	Last Update
AGMP	Subversion	a good marketing ploy	https://studio.atlassian.com/svn/AGMP	Disabled	
applinks	Subversion		https://studio.atlassian.com/svn/APL	Disabled	
atlassian	Subversion		https://svn.atlassian.com/svn/public/atlassian	Running	58 seconds ago
BM	Subversion	Build Monitor	https://labs.atlassian.com/svn/BM	Disabled	
CFS	Subversion		https://studio.plugins.atlassian.com/svn/CFS	Running	50 seconds ago
CGIT	Subversion	Git LightSCM Plugin	https://studio.plugins.atlassian.com/svn/CGIT	Running	50 seconds ago
CLOV	Subversion	Clover	https://studio.atlassian.com/svn/CLOV	Running	28 seconds ago
FE	Subversion	FishEye	https://studio.atlassian.com/svn/FE	Running	5 seconds ago

Info

Indexing has commenced for repository FE

Add...

Adding a new repository

The following pages contain information on adding a new repository in FishEye/Crucible:

External Repositories

- [Adding an external repository](#)
 - [CVS](#)
 - [Git](#)
 - [Mercurial](#)
 - [Perforce](#)
 - [Subversion](#)
 - [SVN fisheye.access](#)
 - [SVN tag and branch structure](#)
 - [How Tags Work in Subversion](#)

Internal Repositories

- [Adding an Internal Repository](#)
 - [Enabling Repository Management in FishEye](#)
 - [Creating Git Repositories](#)
 - [Forking Git Repositories](#)
 - [Deleting a Git Repository](#)

Accessing FishEye repository options

The following pages contain information on configuring the options for a repository in FishEye/Crucible:

- [Repository Options](#)
 - [Authentication](#)
 - [Commit Message Syntax](#)
 - [Configuring repository details](#)
 - [Hidden directories](#)
 - [Include Exclude Paths](#)
 - [Indexer](#)
 - [Linkers](#)
 - [Operations](#)
 - [Permissions](#)
 - [Properties](#)
 - [Store diff info](#)
 - [Tarball settings](#)

- [Text file size limiting](#)
- [Updater](#)
- [Watches](#)

Adding an external repository

This page describes how to add a new external repository to FishEye. Once a repository has been added, further configuration options are available, depending upon the type of repository.

 The information on this page [is restricted](#) in FishEye OnDemand.

 Please note, when the repository is first added, FishEye will build an index and cache. This begins when you first **enable** a repository, and may take some time to complete.

To add a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click **Repositories** (under 'Repository Settings').
3. Click **Add Existing...**
4. Complete the 'Basic Details' form by choosing from **Repository Type**, and entering a name and description for the repository. Click **Next**.
5. Configure the repository settings specific to the repository type. See the following topics for more information:
 - [CVS](#)
 - [Git](#)
 - [Mercurial](#)
 - [Perforce](#)
 - [Subversion](#)
6. Click the **Add** to finish..

CVS

When [adding](#) or [configuring](#) a CVS (Concurrent Versions System) repository, you can:

- Define repository details, **as described below**.
- Set FishEye's [repository options](#).

There are also the following CVS-specific actions:

- [Updater](#).
- [Indexer](#).

Known Limitations

- To add a CVS repository, FishEye must have file system access to the repository. If you cannot install fisheye on the same server as CVS, then use [rsync](#) to mirror the repository.
- Currently, FishEye does not handle the \$Log RCS expansion keyword correctly. Some diff results (and line numbers in diffs) may appear incorrect in files where \$Log is used.

CVS Repository Details

Name	A name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters. Use 'cvs' if you can't think of a better name.

Description	A short description of this repository.
CVS dir	The path to the CVS repository. This is often /usr/local/cvsroot. This is a path in the server's file system.
Charset	The character set used to interpret and display text files.
Enable immediately	Controls whether FishEye will immediately enable this repository, which starts the initial scan. If you wish to do some further configuration before the scan starts, then select 'No'. You can enable a repository later from the Repository List.

Screenshot: Adding a CVS Repository

Add Repository

Name:

Description:

Repository type: Concurrent Versions System (CVS) ▼

CVS dir:

Charset: default (ISO-8859-1) ▼

Enable immediately: ☒ Yes ☐ No

Git

When adding or managing a Git repository, you can do the following:

- Define repository details, as described below.
- Set FishEye's [repository options](#).

FishEye interacts with Git repositories by executing the Git command in a separate process. Hence, the server running FishEye needs to have Git installed. FishEye indexes Git repositories by making a private, bare clone of your repository within FishEye's cache area. It uses this private clone for most Git operations.

Requirements

- The server running FishEye must have Git (version 1.7.1.1 or later) installed.
 - ***i** FishEye should work with later versions of Git and there are some minor features of later versions which FishEye is able to take advantage of.*

On this page:

- [Requirements](#)
- [Git Repository Details](#)
 - [Step 1](#)
 - [Step 2](#)
 - [Step 3](#)
- [Limitations](#)

Related pages:

- [Authentication](#)
- [Git Client Configuration](#)
- [Crucible Repository Configuration](#)
- [General Crucible Configuration](#)

Git Repository Details**Step 1**

Field	Description
Repository Type	Select 'Git'.
Name	Enter a name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters. This will be repository name in FishEye.
Description	Enter a short description of this repository.

Step 2

Field	Description
Repository Location	Enter the URL describing the Git repository location. FishEye will clone this repository for indexing purposes. You can use any URL recognised by Git itself. FishEye recognizes the following URL's: <pre>git://server_name[:port]/path_to_repository http://server_name[:port]/path_to_repository https://server_name[:port]/path_to_repository ssh://server_name[:port]/path_to_repository file://[hostname]/path_to_repository</pre> Do not use spaces in your URL.

Path	(optional) Enter the path within the Git repository that you want FishEye to index. This lets you limit FishEye to indexing a subset of the complete Git repository.
Block Size	(optional) Enter how many commits you want FishEye to process in one batch. Larger values require more memory and increase the amount of work FishEye commits to the database in a single operation.
Command Timeout	(optional) Enter the time that a single Git command is allowed to take to execute. Any command that exceeds this time is terminated and the operation will fail.
Rename Detection	(optional) Select which Git rename detection strategy FishEye will use to detect copy and move operations within the repository. Please refer to the Git documentation for more information .

Step 3

Field	Description
Store Diff Info	Tick this checkbox, if you want FishEye to cache information about file diffs in its database. This is required for some FishEye features. See Configuring Repository Details for more information on this setting.
Enable Repository After Adding	Tick this checkbox, to enable the repository after adding (i.e. when you click the 'Add' button).

Screenshots: Adding a Git Repository

The image shows two screenshots of the FishEye 'New Repository' wizard. The left screenshot, labeled 'Step 3', shows the 'Final Settings' page. It has two checkboxes: 'Store Diff Info' and 'Enable Repository After Adding', both of which are checked. The right screenshot, labeled 'Step 2', shows the 'Git Connection Details' page. It has a 'Repository Location' field with the value 'git://github.com:origin.git', a 'Path' field, a 'Block Size' field, a 'Command Timeout' field set to '1h', and a 'Rename Detection' dropdown menu set to 'copies'. Both screenshots have 'Back', 'Next', 'Add', and 'Cancel' buttons at the bottom.

Step 3

Step 2

Limitations

When FishEye indexes a Git repository, it indexes by the available branches. As it processes the commits on a

branch, FishEye will assign the commit to the branch it first sees the commit on. Commits are only indexed once so if a commit belongs to multiple branches, the commit will not indexed against subsequent branches.

Mercurial

When adding or managing a Mercurial repository, you can do the following:

- Define repository details, as described below.
- Set FishEye's [repository options](#).

Requirements

- **Please use Mercurial 2.1.1 or later.** Mercurial 2.1 has a [bug](#) that makes it incompatible with FishEye. (You should restart FishEye after upgrading Mercurial.)
- Before you add your Mercurial repositories, you will need to [set your executable location](#) for hg (Mercurial).

On this page:

- [Requirements](#)
- [Mercurial Repository Details](#)
 - [Step 1](#)
 - [Step 2](#)
 - [Step 3](#)
- [Notes](#)

Related pages:

- [Authentication](#)
- [Mercurial Client Configuration](#)
- [Configuring Crucible repositories](#)
- [General Crucible Configuration](#)

Mercurial Repository Details

Name	A name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters.
Description	A short description of this repository.
Repository Type	Select ' Mercurial ' from the drop-down list.
Enable Immediately	Defaults to ' Yes '; set to ' No ' if desired.
Location	The URL describing the Mercurial repository location. FishEye will clone this repository for indexing purposes. ⓘ This URL must be compatible with the hg clone command.
Block Size	Controls how many commits FishEye will process in one batch. ⓘ This setting appears under ' Advanced '.

Step 1

Field	Description
Repository Type	Select 'Mercurial'.
Name	Enter a name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters. This will be repository name in FishEye.
Description	Enter a short description of this repository.

Step 2

Field	Description
Repository Location	Enter the URL describing the Mercurial repository location. FishEye will clone this repository for indexing purposes.  This URL must be compatible with the hg clone command.
Authentication Style	Choose the desired authentication style for your repository — 'No authentication', 'Generate key pair for SSH', 'Upload private key for SSH' or 'Password for http(s)'.
Block Size	(optional) Enter how many commits you want FishEye to process in one batch. Larger values require more memory and increase the amount of work FishEye commits to the database in a single operation.
Command Timeout	(optional) Enter the time that a single Mercurial command is allowed to take to execute. Any command that exceeds this time is terminated and the operation will fail.

Step 3

Field	Description
Store Diff Info	Tick this checkbox, if you want FishEye to cache information about file diffs in its database. This is required for some FishEye features. See Configuring Repository Details for more information on this setting.
Enable Repository After Adding	Tick this checkbox, to enable the repository after adding (i.e. when you click the ' Add ' button).

Screenshots: Adding a Mercurial Repository

The image shows two screenshots of the FishEye 'New Repository' wizard.
Step 1: Basic Details (Page 1 of 3)
 - Repository Type: Mercurial
 - Name: ExampleRepo
 - Description: An example Mercurial repo
 - Buttons: Back, Next, Add, Cancel
 - Link: Show advanced settings
Step 2: Mercurial Connection Details (Page 2 of 3)
 - Repository Location: ssh://hg@bitbucket.engineers.example.com
 - Authentication Style: No authentication
 - Mercurial Options: Block Size, Command Timeout
 - Buttons: Test Connection, Back, Next, Add, Cancel
 - Link: New Repository

Step 1

Step 2

Notes

- **Submitting Feedback** — We're very interested in your feedback regarding this feature. The best place for submitting feedback is the [FishEye forums](#).

Perforce

When adding or managing a Perforce repository, you must:

- Define repository details, as described below.
- Set FishEye's [repository options](#).
- Set up a [Perforce client](#).

Perforce Repository Details

Name	A name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters. Use 'perforce' if you can't think of a better name.
Description	A short description of this repository.
Perforce Host	The name of the server which provides the Perforce repository.
Port	The port the server is listening on. This field is optional. FishEye will default to the standard Perforce port (1666) if you do not specify a value here.
Path	The path within the Perforce depot that you wish to have FishEye index. You would normally put the depot path here, e.g. //depot/ but you may also use a more specific path to restrict FishEye to a subset of the depot. The Perforce wildcards '*' and '...' are not supported.

Block Size	Controls how many changelists FishEye will fetch from the depot in one batch. Larger values can reduce the time it takes for FishEye to scan your repository for changes, but use more memory. The default is 400.
Filelog limit	FishEye uses the P4 filelog command to gather information about the files in changesets. The list of files generated can be very large. Setting a limit here will cause FishEye to batch up filelog operations into groups. This is useful with some versions of the Perforce client which may have trouble with large output. In general you should only set this field if you have a 2005 client or earlier. Lower values will degrade scanning performance.
P4 Operation Timeout	Sets the timeout value that FishEye imposes on P4 operations. Operations which exceed this value are terminated. The default for most operations is 10 minutes. Use the following terms to specify particular units of time: s, mi, h, d, w, mo, y (for seconds, minutes, hours, days, weeks, months and years respectively). For example, 10s.
Throttle connections-per-sec	If set, this allows FishEye to throttle how many connections it makes per second to the Perforce server. The default is blank (do not throttle). You may enter fractional values such as 2.5.
Charset	The character set used to interpret and display text files.
Unicode Server	This field indicates whether the Perforce Server is running in internationalised mode.
Skip Labels	When true, FishEye will not scan Perforce Labels for FishEye tag information.
Case Sensitive	This field indicates whether the Perforce Server metadata is case sensitive. You should set this to 'false' for servers running on Windows platforms.
Disable Multiple Print	When FishEye needs file content from Perforce, it uses a p4 print operation. Normally FishEye will request multiple files in one operation but this can cause problems in some Perforce instances. Set this value to true to disable printing multiple files in one operation.

Start Revision	If you wish to set this, please enter a changelist number. If set, the revision number from which FishEye will start indexing the repository. The default is to start scanning from the first revision in the repository.
Initial Import	When a Start Revision is set, this setting controls how FishEye establishes the initial state of the repository. When true, FishEye will import the repository content as it existed one revision prior to the start revision. FishEye will create a single synthetic revision to hold the initial state. The comment associated with this revision will be 'Created by FishEye for initial repository import'. False means that FishEye will only process the revisions from the start revision onwards. The repository state prior to this revision is ignored.
Username/Password	The credentials to use if your repository requires authentication.
Store Diff Info	Enable this option if you are using the Subversion or Perforce SCM systems and making use of per-author line counts. Otherwise, enabling this option is not necessary. Read more information
Enable immediately	Controls whether FishEye will immediately enable this repository, which starts the initial scan. If you wish to do some further configuration before the scan starts, then select 'No'. You can enable a repository later from the Repository List.

Screenshot: Adding a Perforce Repository

Add Repository ⓘ

Name*:

Description:

Repository type:

Enable immediately: ☒ Yes ☐ No

Perforce Connection ⓘ

Perforce Host*:

Port:

Path*:

Perforce Options ⓘ

Block Size:

Filelog Limit:

P4 Operation Timeout:

Throttle connections-per-sec:

Charset:

Unicode Server: ☐ Yes ☒ No

Skip Labels: ☐ Yes ☒ No

Case Sensitive: ☒ Yes ☐ No

Disable Multiple Print: ☐ Yes ☒ No

Start Revision:

Initial Import: ☐ Yes ☒ No

Username:

Password:

Store Diff Info: ☒ Yes ☐ No

Subversion

When adding or managing a SVN (Subversion) repository, you can:

- Define repository details, as described below.
- Set FishEye's [repository options](#).
- Set up a [Subversion client](#).
- Grant [permission to FishEye](#) to scan your repository.
- Set up the correct [branch and tag structure](#).

There are also the following SVN-specific actions:

- [Updater](#)
- [Indexer](#)
- [Store diff info](#)

 It is particularly important that you set up the correct branch and tag structure for your Subversion repositories. If FishEye does not know which files are tags and branches, it will treat all files as trunk files. This can significantly increase the effective size of your repository. **This will increase initial scan time and impact runtime performance.** Please refer to the instructions on [tag and branch configuration](#).

In the majority of cases, indexing a small repository shouldn't take hours, and certainly not days. However, if you have a giant repository, have a slow remote host, you're using HTTP or HTTPS protocols, or if there is a problem with the symbolic setup of your repository, it could potentially take hours or even days. If in doubt, schedule the indexing to run over a weekend or extended maintenance period.

 Using the 'file://' protocol to access your Subversion repository can be much faster than the other network protocols if your repository is on a local disk on the same server as FishEye.

Knowledge Base

You may find some useful information in the [Knowledge Base](#) too.

SVN Repository Details

Detail	Description
Repository type	The type of repository - select "Subversion"
Name	A name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters. Use 'svn' if you can't think of a better name.
Description	A short description of this repository.
SVN URL	The Subversion Repository Root URL to your repository, such as svn://svn.foo.com or file:///var/svn . If you are not sure what your repository root is, check the section below "Finding your Repository Root". (Please note that file protocol performs the fastest followed by svn and lastly by http/s. Therefore where possible please use the file protocol.)
Path	The sub-tree within your repository that FishEye should display. If this value is '.' (or empty), then the whole repository will be shown.
Username/Password	The credentials to use if your repository requires authentication.
Store Diff Info	Enable this option if you are using the Subversion or Perforce SCM systems and making use of per-author line counts. Otherwise, enabling this option is not necessary. Read more information

Block Size	Controls how many revisions FishEye will pull down from the repository in one batch. Larger values can reduce the time it takes for FishEye to scan your repository for changes, but use more memory. Smaller values can reduce the amount of memory FishEye uses during scans. The default is 400.
Svn Operation Timeout	Sets the timeout value that FishEye imposes on Subversion operations. Operations which exceed this value are terminated. The default for most operations is 1 hour. It can be changed to a different interval, for example: "2 days", "10 hours", or "20 minutes".
Throttle connections-per-sec	If set, this allows FishEye to throttle how many connections it makes per second to the SVN server. Many systems use <code>inetd/xinetd</code> to service the <code>svnserve</code> protocol. <code>xinetd</code> has, by default, an incoming connection limit which can cause FishEye to disrupt other <code>svnserve</code> -based connections. The default is blank (do not throttle).
Charset	The character set used to interpret and display text files.
Access Code	The access code for the <code>fisheye.access</code> property on the server. See also Subversion fisheye.access .
MD5 Access Code	The MD5 sum of the above Access Code. See also Subversion fisheye.access . (This field only appears if Access Code is set.)
Set Access Property Command	The Subversion command to set the <code>fisheye.access</code> property to grant FishEye access if necessary. See also Subversion fisheye.access . (This field only appears if Access Code is set.)
Start Revision	If set, the revision number from which FishEye will start indexing the repository. The default is to start scanning from the first revision in the repository.
Initial Import	When a Start Revision is set, this setting controls how FishEye establishes the initial state of the repository. 'Do not import' means that FishEye will only process the revisions from the start revision onwards. The repository state prior to this revision is ignored. 'Import without tag information' means that FishEye will import the repository content as it existed one revision prior to the start revision. FishEye will create a single synthetic revision to hold the initial state. The comment associated with this revision will be 'Created by FishEye for initial repository import'. Tags created prior to the start revision are ignored.

Follow Base Moves	If you have changed the location of your repository in your Subversion server over time, set this option to true. Doing so will cause FishEye to index your repository's full history across all its locations within your server.
trunk/branch/tag structure	Determines how FishEye attempts to understand the tag and branch structure of your Subversion repository. Read more information .
Enable immediately	Controls whether FishEye will immediately enable this repository, which starts the initial scan. If you wish to do some further configuration before the scan starts, then select 'No'. You can enable a repository later from the repository list.



If FishEye returns the error 'Authentication Cancelled', this may simply indicate an incorrect username or password.

Finding your Repository Root.

Run the following command:

```
svn info SVN_URL
```

Where SVN_URL is the complete URL of the repository you want to add.

You will get something like the following:

```
>svn info http://svn.example.com/svn5/

Path: svn5
URL: http://svn.example.com/svn5/
Repository Root: http://svn.example.com/
Repository UUID: ce062a09-193b-427a-a7b3-a85007076e5d
Revision: 83
Node Kind: directory
Last Changed Author: ryan
Last Changed Rev: 83
Last Changed Date: 2009-05-07 10:48:41 +1000 (Thu, 07 May 2009)
```

Next to "Repository Root" is the URL you should define as your repository root. The path will be whatever is remaining.

Screenshots: Adding a SVN repository

Add Repository - Page 1 of 3

Add Repository - Page 2 of 3

Basic Details

Repository Type:
Name: *
Description:

SVN Connection Details

SVN URL: *
Path:
Username:
Password:

Add Repository - Page 3 of 3

Final Settings

Store Diff Info: ☒
Enable Repository After Adding: ☐

SVN fisheye.access

The `fisheye.access` property allows an administrator/committer to control FishEye access to a directory in the repository. FishEye queries this property to decide whether it will continue to access the repository. If the property exists, but does not match that configured in FishEye, FishEye will immediately disconnect from the repository.

⚠ By default, FishEye will have access to your repository and you do not need to configure this access mode. It is only necessary if you identify a need to restrict FishEye access to your repository (this is generally not required).

Setting FishEye Access Mode

FishEye can operate in one of three access modes:

Mode	Access	Subversion repository property: <code>fisheye.access</code>
Allow	Any FishEye server	'allow' or no property set
Access Code	Only FishEye servers configured with the correct Access Code	e.g. 'md5:dc0c08df1f3e80b599c90f53d7dd05ec'
Deny	No FishEye server	'deny'

If you would like to restrict FishEye access to your repository, you must set the `fisheye.access` property. This property must be set on the 'URL + path' you have configured in FishEye.

Setting an Access Code

The repository must be configured with the MD5 sum of the Access Code that is configured in FishEye. The MD5 sum and even the `svn` command to set the property will be generated for you by FishEye when you configure the repository using the FishEye Administration page. See [Subversion repository details](#).

For example, if you have configured FishEye with a URL of `svn://foo.com/`, a path of `.` and an Access Code of 'fisheye', then you would need to do something like this:

```
$ svn checkout -N svn://foo.com/ tmpworkspace
$ cd tmpworkspace
$ svn propset fisheye.access "md5:4d0c5db8382f80c58e7b0619ae5767a7" .
$ svn commit -m "grant fisheye access"
```

Denying Access to all FishEye Instances

To deny all FishEye instances access to the repository, it must be configured with the `fisheye.access` property of 'deny'.

For example, if you have configured FishEye with a URL of `svn://foo.com/` and a path of `.` (or you have left path empty), then you would need to do something like this:

```
$ svn checkout -N svn://foo.com/ tmpworkspace
$ cd tmpworkspace
$ svn propset fisheye.access "deny" .
$ svn commit -m "disable fisheye access"
```

If you configured a path of `some/dir` then use:

```
$ svn checkout -N svn://foo.com/some/dir tmpworkspace
$ cd tmpworkspace
$ svn propset fisheye.access "deny" .
$ svn commit -m "disable fisheye access"
```

SVN tag and branch structure

This page contains an explanation of how Subversion works, how FishEye interacts with it, and examples of how to configure FishEye to work with Subversion according to your needs.

Overview

Subversion, one of the most popular source code management systems in use today, applies an interesting technology for creating branches and tags. In a Subversion repository, branches and tags can be easily copied or duplicated — this is allowed by creating a form of pointer or reference from one location to another, avoiding the need to duplicate a lot of information. The disadvantage of this is that Subversion repositories can be confusing to administer at times and its internal complexity can be problematic for applications such as FishEye that need to finely process its contents. As a result, FishEye may require some in-depth configuration with Subversion.

 If you are following strict conventions in organising your Subversion repository (as laid out in the [Subversion Red Book](#)), FishEye has automatic presets which are compatible with your repository.

Jump straight to the [examples](#), if you're already very familiar with Subversion and FishEye. Otherwise, please read on.

On this page:

- [Overview](#)
- [Introduction to Subversion conventions](#)
- [Choosing the correct configuration method](#)
- [Custom layouts](#)
 - [How to set a custom layout](#)
- [Examples](#)
 - [Ideal configuration example](#)
 - [Custom example 1](#)
 - [Custom example 2](#)
 - [Example from a FishEye customer](#)
- [How Subversion works](#)
- [Frequently Asked Questions](#)

Introduction to Subversion conventions

The most common project structure in Subversion is to use a project structure and have trunk, branches and tags directories for each project, like so:

```
/project1/trunk/...  
/project1/branches/b1  
/project1/branches/b2  
/project1/branches/...  
  
/project1/tags/t1  
/project1/tags/t2  
/project1/tags/...
```

If you point your FishEye repository at a single project root, say /project1/, then to FishEye, the directory structure looks like this:

```
/trunk  
/branches  
/tags
```

In this case, you have one single project within the repository root. On the other hand, if you point FishEye at the repository root, you will have multiple projects visible in your FishEye view.

i Note that this example follows common conventions for repository layout, as outlined in the [Subversion Red Book](#). If you follow these conventions, your FishEye configuration will be simpler.

Choosing the correct configuration method

Unsure as to whether you can use a preset symbolic rule, or whether you need to write your own custom symbolic rules? Refer to the following:

1. If you point FishEye at your project root:

```
/PROJECT1
```

and you have the following structure at this level of the repository:

```
trunk
branches
tags
```

-then use FishEye's built-in rules for easy, one-step configuration.

2. If you point FishEye at the repository root, you have multiple projects and each project has the following structure:

```
trunk
branches
tags
```

leading to this structure in each project:

```
/PROJECT1/trunk
/PROJECT1/branches
/PROJECT1/tags
```

```
/PROJECT2/trunk
/PROJECT2/branches
/PROJECT2/tags
```

(and so on), then use FishEye's built-in rules for easy, one-step configuration.

3. Otherwise, you have a [custom layout](#).

Custom layouts

i [Regular expressions](#) are required knowledge for creating custom repository layouts in FishEye.

You will now need to describe to FishEye how to recognise the paths in your repository. What you are telling FishEye is which paths within the repository are related, i.e. which are operations on the same file in different branches and which are tags of a file. You must also tell FishEye how to determine the branch name or the tag name. Most custom layouts are variations on the one of the two standard layouts described above. The best approach to creating your custom configuration is to use one of the appropriate entries from the drop down list. This can serve as a template for you, which you can then customize. Once you have selected the appropriate template, select the "Custom" entry from the drop down list. Now, you will be able to edit the entries (or add entries).

See the [examples](#) below for more information.

How to set a custom layout

Using [Regular Expressions](#), you can describe any custom tag or branch structure that you have. You should use one of the common layouts (from the drop down list) as a basis, firstly select it, then select 'Custom' to edit or add rules.

When looking at a file on a branch, or a file that was tagged, FishEye needs to determine a name for the branch or tag. FishEye does this by matching a regular expression against the file's path, and extracting the name based upon the match. FishEye also needs a name for files on the trunk. In effect, this is the name of the trunk 'branch'.

For any file that matches a trunk/branch/tag regular expression, a logical path is calculated. Two different files with the same logical path are considered to be related. For example, using the second type of common repository layout:

- The file `project1/trunk/dir1/foo.txt` would have a logical path of `project1/dir1/foo.txt`.
- The file `project1/tags/BUILD123/dir1/foo.txt` would have a logical path of `project1/dir1/foo.txt` and the name of the tag would be `project1-BUILD123`.
- Both these files have the same logical path, and so are considered related. By looking at the revision where the directory-copy for `project1/tags/BUILD123/dir1/foo.txt` occurred, FishEye can determine to what revision the tag `project1-BUILD123` applies.

You can add as many rules as you need. For any given file, the first rule that matches is used.

Rule	Description
Regex	The regular expression used to match against the start of the path. The trailing part of the path that does not match the regex is called the tail.
Name	An expression used to extract a tag or branch name from the regex.
Logical Path Prefix	This is an expression used to construct the logical path. The logical path is the concatenation of the result of this expression, and the tail of the regex.

See the [examples](#) below for more information.

Examples

These examples show the regular expressions used for some custom configurations. If you need more information on how these examples work, please see [SVN tag and branch structure](#) on this page.

- [Ideal configuration example](#)
This shows a best-case near "zero configuration" project structure that is instantly compatible with FishEye.
In this case, you have trunk, branches and tags as the base folders in your repository.
- [Custom example 1](#)
This shows a custom project structure and the symbolic rules required.
- [Custom example 2](#)
This shows another kind of custom project structure and the symbolic rules required.
- [FishEye customer](#) example.

This is a real-world configuration used by a FishEye customer.

Ideal configuration example

If your repository is organised in this way, simply select the **In-Built symbolic rules** option. FishEye will then be fully connected to your repository (you do not need to write a regular expression, or choose anything from a list).

Project Structure

```
/PROJECT1/trunk/
/PROJECT1/branches/branchname
/PROJECT1/tags/tagname
```

 Note that this example follows common conventions, as outlined in the [Subversion Red Book](#).

Custom example 1

 Whenever you have a custom project structure in Subversion, you will need to write a [regular expression](#).

Say you have an additional directory you use for tagging releases, which is different from the everyday tags you create in the tags directory:

Project Structure

```
/trunk/
/branches/branchname
/tags/tagname
/releases/releasename
```

Symbolic Rules

Regular Expression	Name	Logical Path Prefix
trunk(/ \$)	trunk	N/A
branches/([^\s/]+)	\${1}	N/A

(tags releases) /([^\/]++)	\${2}	N/A

Custom example 2

Whenever you have a custom project structure in Subversion, you will need to write a [regular expression](#).

In this example, there is a "core" project area and then a number of separate plugins. the core contains its own trunk/branches/tags structure while the plugins are in a named directory which contains their trunk/branches/tags directory. We want to have the core and all the plugins visible in a single FishEye repository.

Project Structure

```

/core/trunk/
/core/branches/
/core/tags/

/plugins/plugin1/trunk/
/plugins/plugin1/branches/
/plugins/plugin1/tags/

/plugins/plugin2/trunk/
/plugins/plugin2/branches/
/plugins/plugin2/tags/

```

Symbolic Rules

Applies to	Regular Expression	Name	Logical Path Prefix
trunk	core/trunk(/ \$)	trunk	core
trunk	plugins/([^\/]++)/trunk(/ \$)	trunk	plugin_\${1}
branches	core/branches/([^\/]++)	core-\${1}	core

branches	plugins/ ([^\s/]+)/ branches /([^\s/]+)	\${1}-\${2} }	plugin_\${1}
tags	core/tag s/([^\s/]+)	core-\${1} }	core
tags	plugins/ ([^\s/]+)/ tags/([^\s/]+)	\${1}-\${2} }	plugin_\${1}

In this example, the Logical Path Prefix has been configured to distinguish files with the same name in different plugins. For example, the file build.xml may exist in all plugins but such files are not related even though they have the same name. The Logical Path Prefix is used to tell FishEye to which "logical group" the files belong.

Example from a FishEye customer

This is real-world example from a FishEye customer. This is a slightly non-standard project structure. The correct symbolic rules for this project structure are shown below:

Project Structure

/trunk/PROJECT1

/branches/PROJECT1/branchname

/tags/PROJECT1/tagname

Symbolic Rules

Regular Expression	Name	Logical Path Prefix
trunk/([^\s/]+)	\${1}	N/A

<code>branches/([^\s/]+)</code>	<code>/\${1}-\${2}</code>	N/A
<code>tags/([^\s/]+)/([^\s/]+)</code>	<code>/\${1}-\${2}</code>	N/A

How Subversion works

Since tags and branches are implemented via directory copies in Subversion, they are not really first-class concepts. This means that FishEye has to determine branch and tag information by examining the paths involved in Subversion operations and matching these against branch and tag conventions used in the repository. Since these conventions are not fixed, you may need to tell FishEye what conventions you use in your repository. By default FishEye has some inbuilt rules which handle the most common conventions typically used in most Subversion sites. If, however, you've decided to use a custom convention, you can define custom rules to describe what your tag/branch structure looks like. These settings can be edited on the 'Add Repository' or 'Edit Repository' pages in the FishEye Administration pages.

The symbolic setup tells FishEye how to classify each path it encounters as it indexes the repository. Each path is classified as either a trunk, branch, tag or root path. The trunk, branch and tag categories are the normal conventions used in SCMs. The root category is used when a path does not match any of the given trunk/branch/tag settings and is mostly treated in the same way as trunk paths. For example, the branches directory itself does not belong to the trunk, a particular branch or a tag and is classified as a root path.

 The symbolic settings do not exclude any paths from consideration by FishEye. To exclude paths you should set up appropriate 'allow' rules. If your symbolic setup does not match a path, that path will be classified as a root path and processed by FishEye accordingly.

 If you change these trunk/branch/tag settings, you would normally perform a complete re-scan of the repository to ensure FishEye's index is consistent with the settings. FishEye will suggest this when you make changes and you can also do this manually from the [Indexer](#) option. If you don't want to re-index, you can also choose to ignore this suggestion.

For more information on tag and branch layout, see [Repository Layout](#) in the Subversion documentation, or [How Tags Work in Subversion](#) for more background information.

Frequently Asked Questions

What is the logical path prefix?

The first part of a path, where you want to constrain to a specific origin for the paths in context.

What is the logical tail?

The last part of the path is the logical tail, that specifies the exact folders in context.

How does the logical tail affect the search?

The logical tail could be used to select every occurrence of a file called `build.xml`, for example, or every folder of a given name near the end of the path.

How Tags Work in Subversion

If you are having trouble finding out the correct Tag or Branch names to use under Subversion (especially when

searching or creating EyeQL queries), the steps on this page will help.

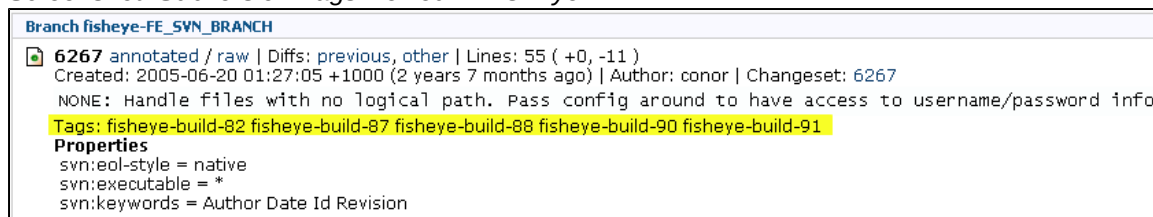
Discovering your Subversion Tag & Branch Structure

1. Open a file in your repository using FishEye. Look for a file that has a long lifetime (for example, your build script).
2. Look at the '**Tags**' section to see examples of tags in your system. There are a few different conventions, but the tags shown will give you clues as to how your repository custodians do things.
A typical tag looks like this:

```
fisheye-build-82
```

3. Copy the tag(s) you may need from the examples you can see in FishEye's view.
4. Enter the tags exactly as they appear into your EyeQL queries or (FishEye Search).
5. Test the outcome.

Screenshot: Subversion Tags viewed in FishEye



How tag names are constructed

Hyphens are the default method for separating the tag elements, but tags in your FishEye instance may be different.

Your tag structure depends on several things:

- The symbolic structure of your repository
- The way your FishEye instance was set up
- Your organisation's convention for naming things in the repository
- The configurable character that separates parts of the tag name.

This can sometimes make it confusing to guess what the tag structure could be, when you are searching in FishEye or using an EyeQL query. Additionally, the separator between tag elements can be configured as hyphen, colon, or other punctuation marks.

Note that slashes cannot be used in a tag name – these are converted to colons by default. For example if your symbolic setup would give you a tag name like the following:

```
path/project/fisheye-build-82
```

then your tag should look like this:

```
path:project: fisheye-build-82
```

For more information, see [SVN tag and branch structure](#).

Adding an Internal Repository

 **Support for repository management within FishEye will end on August 14th, 2013**

You will continue to be able to browse, search and visualise externally hosted Git repositories through FishEye. [Learn more about this change.](#)

As of version 2.7, Fisheye has the capability to create and host internal Git repositories. This feature is called **Repository Management**.

For instructions, see the following pages:

- [Enabling Repository Management in FishEye](#)
- [Creating Git Repositories](#)
- [Forking Git Repositories](#)
- [Deleting a Git Repository](#)

To setup the authentication of your credentials over Git, see the following page: [Permanent authentication for Git repositories over HTTP\(S\)](#)

Enabling Repository Management in FishEye

 **Support for repository management within FishEye will end on August 14th, 2013**

You will continue to be able to browse, search and visualise externally hosted Git repositories through FishEye. [Learn more about this change.](#)

Repositories can be created inside the FishEye interface. These repositories are then hosted by FishEye.

1. Start by [downloading and installing FishEye](#) on your system.

The installation instructions will be displayed on the download page.

2. Check your Git version

To use Git repository management, you will need to have your Git binary path and your repository root set up in FishEye.

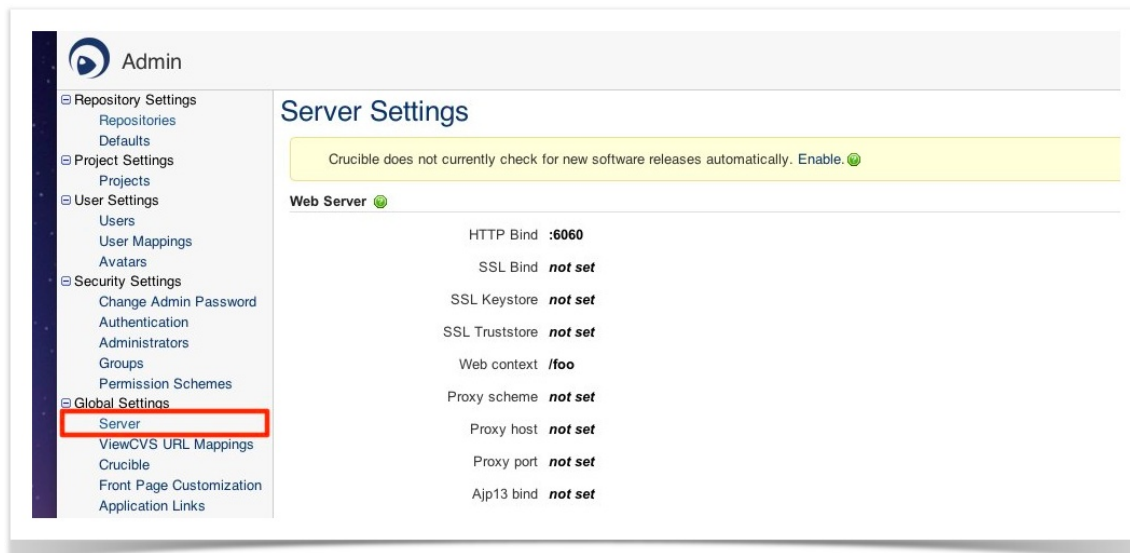
Make sure that you are using Git 1.7.1.1 or later.

You can download the latest version of Git here if necessary: <http://git-scm.com/download>

3. Open the FishEye administration screen

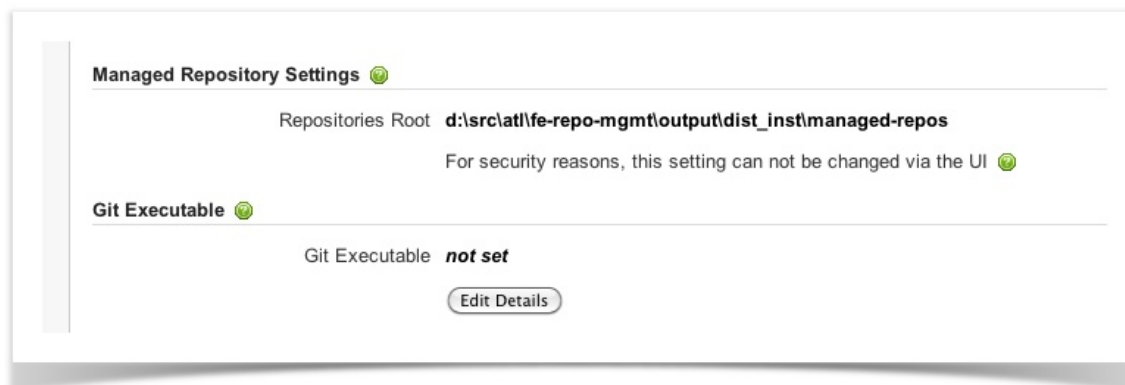
Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.

Open the '**Server administration**' page.



4. Edit Git settings

Edit your Git repository settings and set up your Git binary path if it is not already configured.



Running Apache?

If the server you've installed FishEye on is already serving Git via a web server such as Apache, make sure that any existing rules don't conflict with FishEye's Git serving URL: `http://<FishEye base URL>/git/<repository name>`. FishEye will handle authenticating and serving your Git repositories for you.

Setting up Git on your computer

1. Download and install the [latest version of Git](#).
2. Set up your username and email address.

Your commits will be tracked using the username and email specified in the Git configuration. To configure these, execute the command below replacing the username and email address with your own:

```
git config --global user.name "John Doe"
git config --global user.email "jdoe@company.com"
```

Creating a Git Repository

See [Creating Git Repositories](#).

Working with your new repository

Import your project to Git

If your project is not already controlled by Git just execute the following commands:

```
cd existing-project
git init
git add .
git commit -m "First commit"
git remote add origin http://myserver/my/repo/url
git push origin master
```

At this stage all your project files should have been added to your repository in FishEye

Already using Git for your project?

In case your project is already controlled by Git the follow steps will allow you to add FishEye as a new remote url that you can push to:

```
cd existing-project
git remote add fisheye <repo_url>
git push fisheye master
```

i The name "fisheye" used in this example is arbitrary. You can use any name which is not already used by a remote Git repository.

Forking Repositories

See [Forking Git Repositories](#).

i Future Updates Planned for This Feature

Several customers have requested that Atlassian update this feature to allow non-admin users to create repositories.

We're happy to report that allowing users to create their own repositories is something that Atlassian is already planning on doing for a FishEye release following version 2.7. Enabling repository management for administrators is just the first step FishEye will take in this direction.

Related Links

- [Enabling Repository Management in FishEye](#)
- [Creating Git Repositories](#)
- [Forking Git Repositories](#)
- [Deleting a Git Repository](#)

Creating Git Repositories

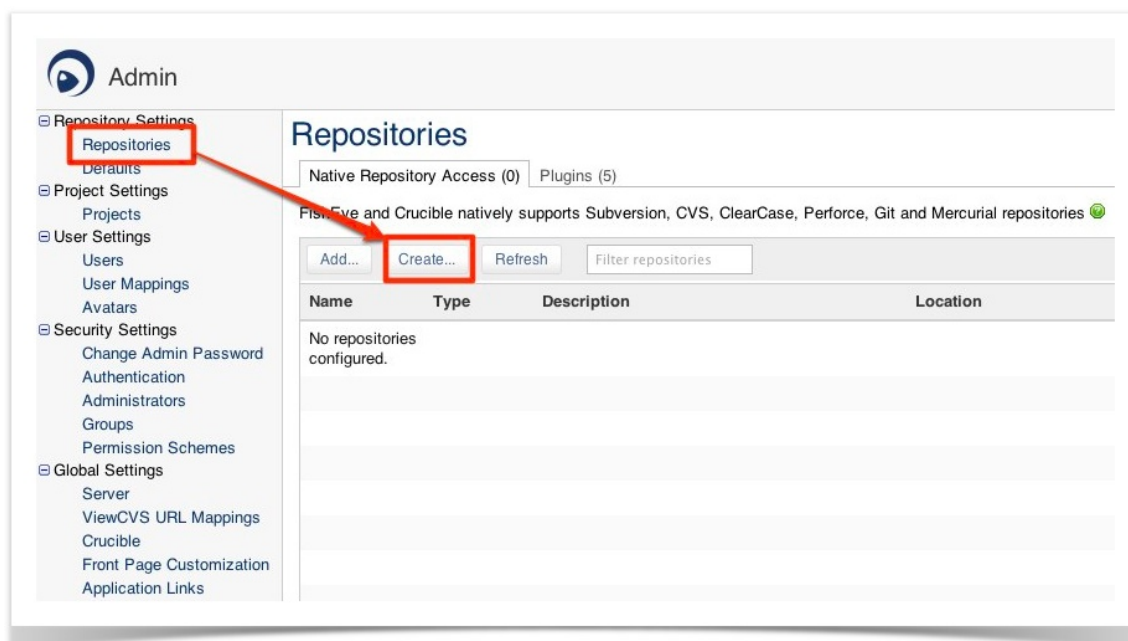
❌ **Support for repository management within FishEye will end on August 14th, 2013**

You will continue to be able to browse, search and visualise Git repositories externally. [Learn more about this change.](#)

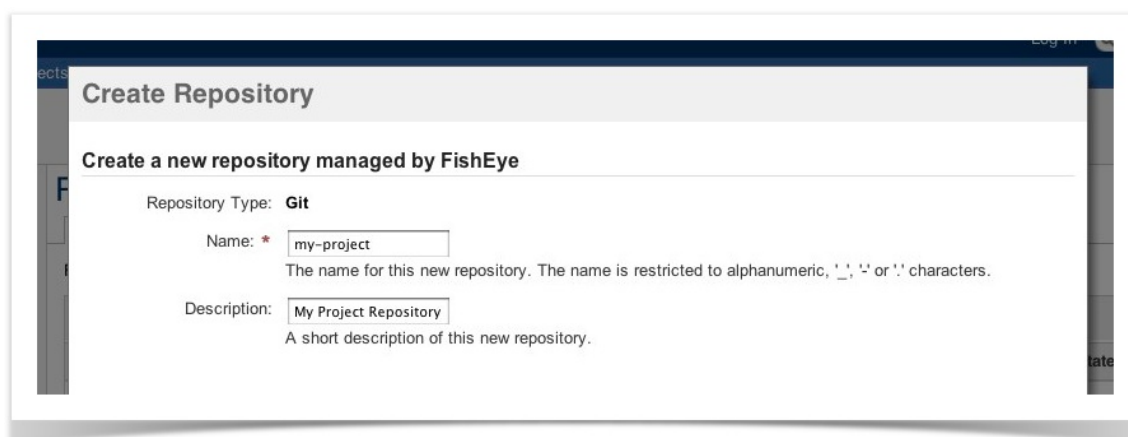
Creating a repository

Creating a new Git repository is a straight-forward process.

1. Go to the repositories section of the admin interface and click on **Create**.

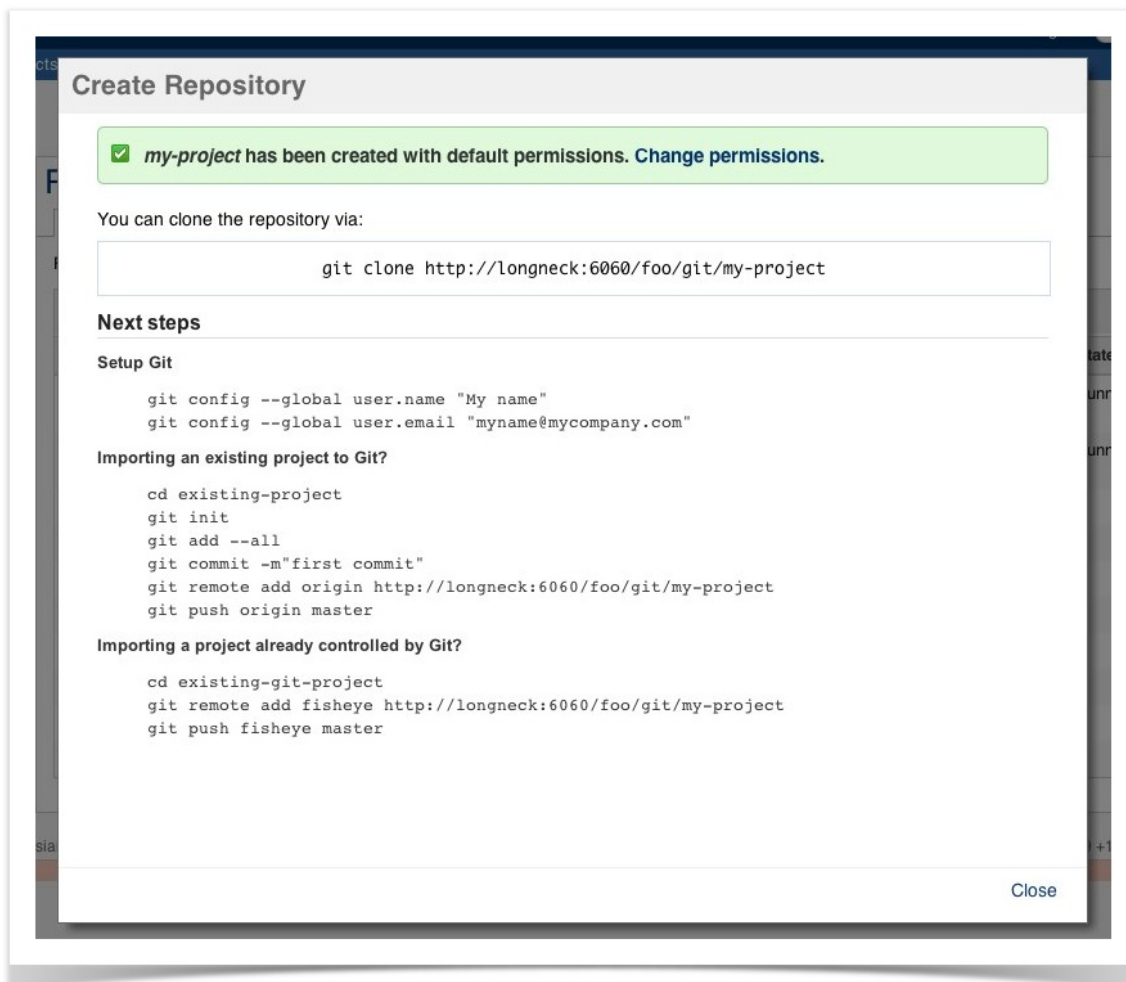


2. Specify the name and description for the new repository.



❌ Only alphanumeric characters, and the following punctuation characters are accepted in the name: underscore '_', hyphen '-' and full-stop / period '.' .

3. Click **Create** at the bottom to add your new repository.



The repository URL can also be found on the repository page.

Related Links

- [Enabling Repository Management in FishEye](#)
- [Creating Git Repositories](#)
- [Forking Git Repositories](#)
- [Deleting a Git Repository](#)
- [Permanent authentication for Git repositories over HTTP\(S\)](#)

Forking Git Repositories

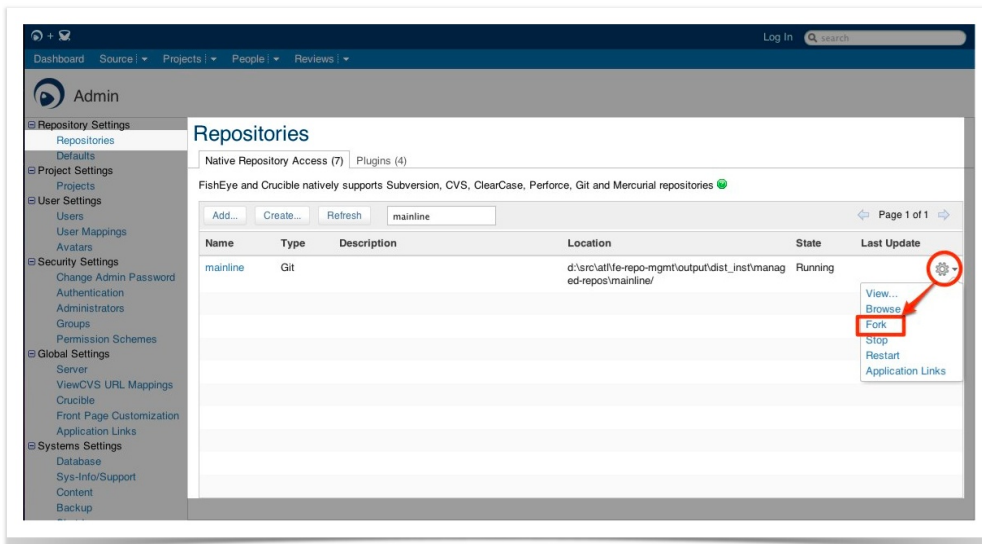
Support for repository management within FishEye will end on August 14th, 2013

You will continue to be able to browse, search and visualise externally hosted Git repositories through FishEye. [Learn more about this change.](#)

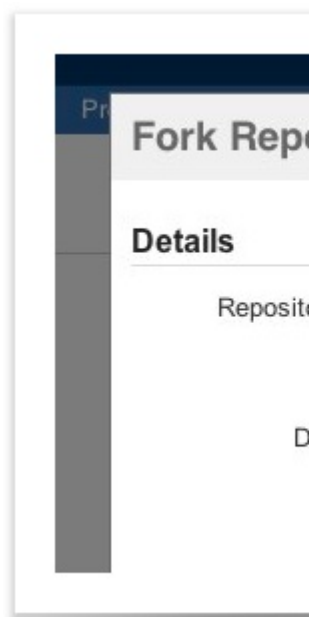
Once you have created a Git repository in FishEye, you can create copies which will be linked to it as "children" of the repository. This is what we call a *fork*. ([learn more about forks](#)).

To create a fork, follow the steps below.

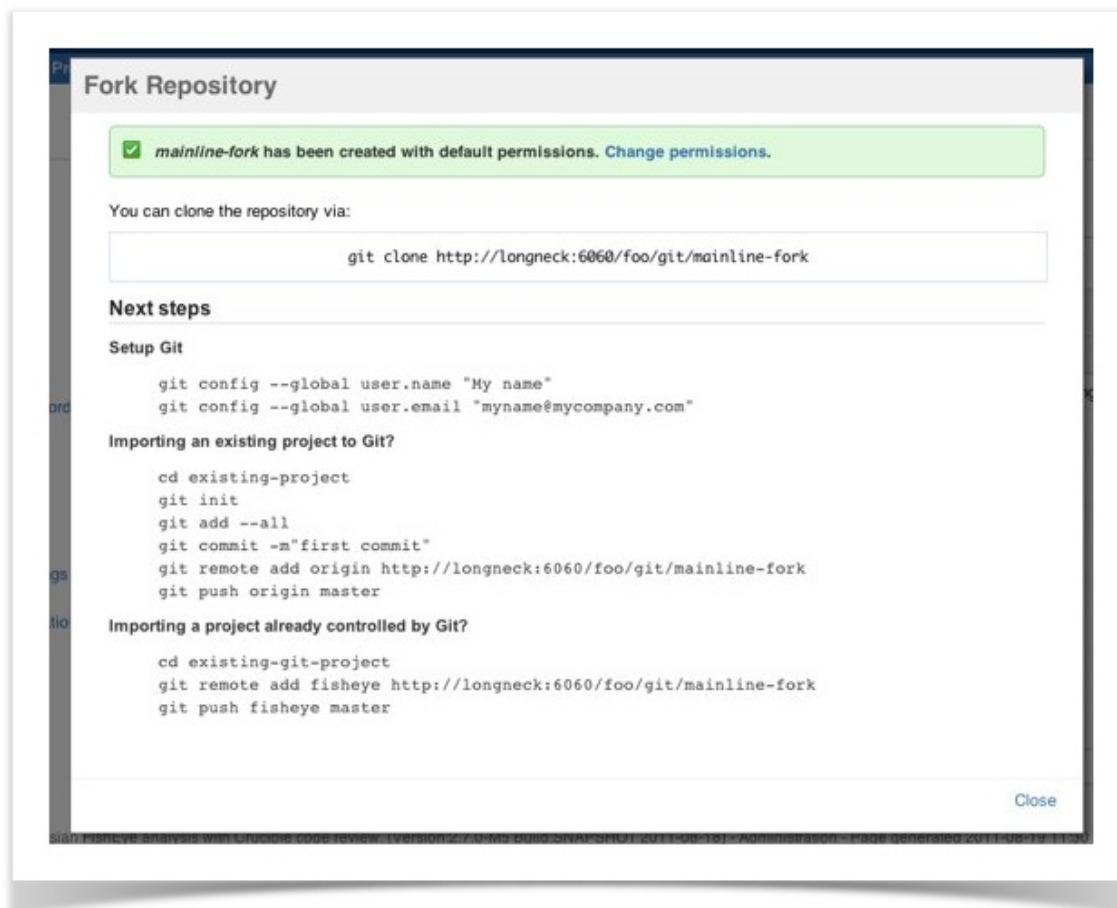
1. Open the cog menu in the repositories section of the admin interface and click **Fork**.



2. Fill in the form, and click **Fork**.

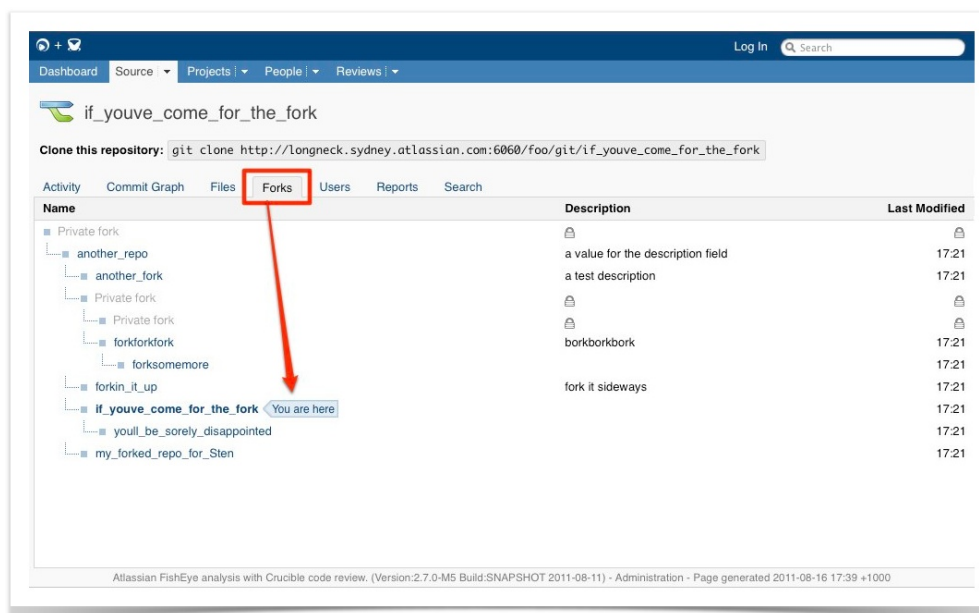


3. FishEye will fork the repository.

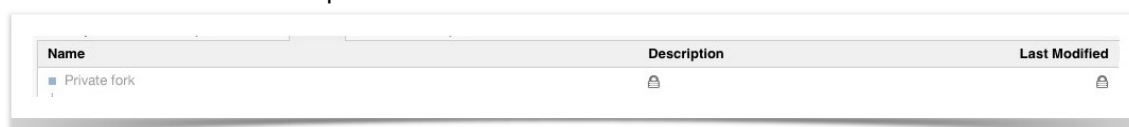


Viewing Fork Hierarchies

To visualize the relationships between the different forks you just need to go to the *forks* tab of the repository.



If you do not have the permission to access a repository, you will see a line containing the mention *Private fork* and a lock icon on the description and modification date.



Related Links

- [Enabling Repository Management in FishEye](#)
- [Creating Git Repositories](#)
- [Forking Git Repositories](#)
- [Deleting a Git Repository](#)
- [Permanent authentication for Git repositories over HTTP\(S\)](#)

Deleting a Git Repository

Internal repositories can be deleted through the regular FishEye repository management interface, as explained in [Managing your repositories](#), but when an attempt is made to delete the repository, a dialog confirms the action due to the fact that the repository itself will be destroyed (rather than just the FishEye index information, as for external repositories).

i If you delete a fork from a hierarchy, its children will be re-parented to the parent of the deleted repository's parent. If the parent is at the top level of the hierarchy, then the children will be moved to the top level.

Setting up a Repository Client

- [CVS Client](#)
- [Git Client](#)
- [Mercurial Client](#)
- [Perforce client](#)
- [Subversion Client](#)

CVS Client

FishEye supports CVS repository access, but this does not require installation of a separate client.

As FishEye reads your CVS repository directly from the local file system, using a CVS repository is a 'zero-installation' scenario. Once you have set up FishEye on the system where your CVS repository resides, FishEye should be able to access it automatically once you have configured it.

See [Configuring FishEye to access your CVS repository](#) for more information.

Git Client

i From FishEye 2.7, FishEye has the [capability to create Git repositories](#), which are then hosted in FishEye.

This new functionality is significantly different from the features described on this page, which refer to adding *externally hosted repositories* to FishEye's list of things to scan.

[Git 1.7.1.1](#) is supported at this stage.

Configuration

FishEye requires Git 1.7.1.1 or later. At present, you need to have the Git executables in the path of the user used to run your FishEye instance.

Related Links

- [Git Configuration](#)
- [Crucible Repository Configuration](#)
- [General Crucible Configuration](#)

Mercurial Client

Please see the [Supported Platforms](#) page for information about the versions of Mercurial that are supported by FishEye.

Configuration

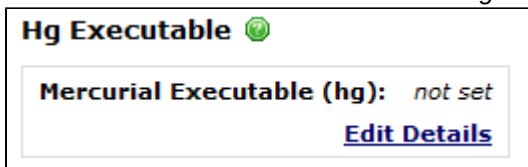
To set the location of the Mercurial executable, carry out the following steps:

1. Open the FishEye Admin screen.
2. Select '**Server Settings**' under '**Global Settings**' in the left navigation bar.
3. Click '**Edit Details**' under '**Hg Executable**' in the configuration dialog.
4. Enter the path to your hg (Mercurial) binary executable.
5. Click '**Update**'.
6. Restart your FishEye server.

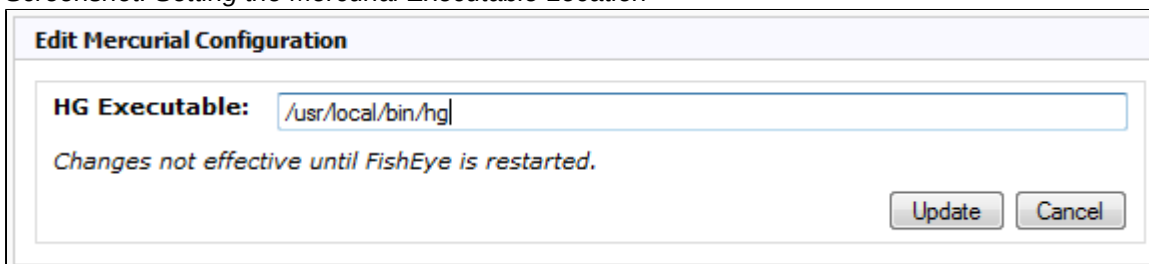
Following this process, you will be able to add your Mercurial repository to FishEye.

 FishEye requires Mercurial 1.5.1 or later.

Screenshot: Mercurial Executable Settings



Screenshot: Setting the Mercurial Executable Location



Submitting Feedback

We're very interested in your feedback. The best place for submitting feedback is the [FishEye forums](#).

Related Links

- [Mercurial Configuration](#)
- [Crucible Repository Configuration](#)
- [General Crucible Configuration](#)

Perforce client

FishEye can communicate with any Perforce server, but it needs to use the P4 command-line client to do so.

By default, FishEye looks for the p4 executable in the current path. To specify the exact path of the p4 executable, click **Server** (under 'Global Settings') in the FishEye Admin area, and look for 'Perforce client'.

Files incorrectly considered binary

Some users have reported errors where FishEye considers some files to be binary when they are not. It appears this may be a limitation of earlier P4 clients. If you can upgrade to a recent P4 client (2006.1 onwards), this will fix this issue. You do not need to update the P4 Server. Please note that the repository will need to be reindexed in Fisheye after upgrading the P4 client for the changes to be picked up.

If you are unable to upgrade to a recent P4 client, the [Repository Details](#) page in FishEye allows you to set a limit on the size of filelog commands sent to the server. Setting this to something around 100 will fix the issue. It will, however, also impact performance significantly.

Subversion Client

FishEye can communicate with any Subversion server running version 1.1 or later, but it needs to use a Subversion *client* to do so.

The SVNKit client is included in the current FishEye package (version 1.4.2 onwards). This is the default client for interfacing with Subversion, is generally the easiest to use, requires "zero-installation" and will be used automatically unless another client is specified. The SVNKit client is recommended for most users.

The alternative is the native client, which should only be used if the SVNKit client is unsuitable.

- [Native Subversion Client](#)
- [SVNKit Client](#)

 Using the 'file://' protocol to access your Subversion repository can be much faster than the other network protocols. We recommend using the 'file://' protocol if possible.

 Using Subversion 1.3 or later is strongly recommended. Versions prior to 1.3 are no longer supported by the Subversion project. They will work with FishEye, but you may want to consider upgrading to a supported version.

Native Subversion Client

Native Client

FishEye can use a native Subversion client installed on your system, but your client needs to be version 1.2 or later, and **must include the JavaHL bindings**. FishEye can use all of the protocols supported by your native client.

The JavaHL bindings include a Java `.jar` file, typically named `javasvnhl.jar`, and a dynamic library such as `libsvnjava-1.so` or `libsvnjavahl-1.dll`. FishEye must be configured so it can find both the `.jar` and the dynamic library.

If the JavaHL dynamic library is in your library path (such as `%PATH%` on Windows), then FishEye will automatically find it. Otherwise you can tell FishEye where it is, or set the [FISHEYE_LIBRARY_PATH](#) environment variable before starting FishEye.

Acquiring native Subversion libraries for your operating system

Pre-compiled native clients are available for most platforms. The [Subversion download page](#) links to platform specific distributions. **Ensure you get the binary that includes JavaHL bindings**, as well as the standard

package. Also ensure that the versions of the JavaHL and standard packages match.

- **Subversion for Windows:**

To install Subversion for Windows, visit [this page](#)

You need to download the standard package as well as the **JavaHL** version. The standard package is named **svn-X.Y.Z-setup.exe** and the JavaHL installer file is named **svn-win32-X.Y.Z_javahl.zip** where 'X.Y.Z' refers to the version number (for example, svn-win32-1.4.6_javahl.zip at the time of writing).

- **Subversion for Fedora Linux:**

For Linux systems using the yum package manager (such as Fedora Core 3 and above) you can type the following:

```
yum install subversion-javahl
```

at the Linux command line interface to install the JavaHL bindings for Subversion. Note that this will also install the standard Subversion library, which is required.

- **Subversion for Ubuntu and Debian Linux:**

For Linux systems using the apt-get package manager (such as Debian and Ubuntu) you can type the following:

```
apt-get install libsvn-javahl
```

at the Linux command line interface to install the JavaHL bindings for Subversion. Note that this will also install the standard Subversion library, which is required.

Native Client Configuration

There are two ways you can configure the path to your Subversion client: Via the FishEye user interface, or by editing the `config.xml` configuration file.

Configuring your Native Client in the FishEye User Interface

You can configure your Subversion client in the FishEye Administration screens: click **Server** (under 'Global Settings') and look for the 'Subversion client' section.

JAR	The path to the JavaHL .jar.
Dynamic library	The path to the dynamic library, if it is not already on your system's library path. ⚠ Due to a bug in earlier versions of the JavaHL bindings, setting this value is ineffective unless you are using a Subversion client 1.2.3 or later.

Configuring your Native Client in the FishEye Configuration File

An alternative method to using the FishEye user interface is editing the `<svn-config>` section of your `config.xml`. If you change these settings, you need to restart FishEye.

Windows Platform Example (change path locations as required)

```
<svn-config jar="C:\subversion\lib\svn-javahl.jar"  
jniLib="C:\subversion\lib\libsvnjavahl-1.dll"/>
```

Mac OS X Platform Example (change path locations as required)

```
<svn-config jar="/opt/subversion/lib/svn-javahl/svn-javahl.jar"  
jniLib="/opt/subversion/lib/libsvnjavahl-1.dylib"/>
```

Linux Platform Example (change path locations as required)

```
<svn-config jar="/usr/share/subversion/lib/svn-javahl.jar"  
jniLib="/usr/lib/libsvnjavahl-1.so"/>
```

Performance Impact of Native Client

In general, using the JavaHL libraries should improve the speed of FishEye's indexing when compared to SVNKit, primarily because of the advantage of native code. However, some customers have experienced more peak memory usage (some more than 2GB) during indexing when using JavaHL implementations due to greedier processing. This is generally not a major concern, but your own performance requirements may vary.

If you need to allocate more than 2gb for the indexing to complete, please run Fisheye with a 64-bit JDK

SVNKit Client

SVNKit is included in the current FishEye package, and is the default library for interfacing with Subversion. It is generally the easiest to use, and will be used automatically unless another library is specified.

See [Configuring Subversion repositories](#) for more information.

 SVNKit supports the 'file://' protocol for FSFS repositories only.

SVNKit sometimes has problems working with Subversion servers which are running older versions, such as 1.1.x. If you see exceptions such as those listed below in FishEye's log file, you will need to either swap to [use the native client](#) or upgrade your Subversion server to version 1.3 or later.

Example exceptions:

- [SEVERE: assert #B](#)
- [Checksum mismatchwhile reading representation:](#)

Other Subversion Clients

Native Subversion Clients are also supported by FishEye as an alternative to SVNKit. This process requires additional configuration. Read the [Native Subversion Client instructions](#) for more information.

Re-indexing your repository

Re-indexing involves FishEye performing an intensive scan of the repository contents, so it can more quickly display repository data. Re-indexing may be required in a number of situations, such as after server maintenance, changes in your repository, major setting changes and upgrades to FishEye.

i Due to the complexity of SCM repositories, an indexing scan may take many hours or even days to complete. During this time, your FishEye users may be inconvenienced as they will not be able to view the repository being indexed. Furthermore, Crucible users will not be able to carry out reviews unless you have Crucible set to [store all revisions offline](#).

i How do I mitigate long re-index times when upgrading? [See the FAQ page](#) on this topic.

Performing a manual re-index

Settings and administrative operations for repository indexing are located on the [Indexer](#) page.

To access indexing functions for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the name of the repository.
3. Click **Maintenance** in the left-hand panel.
4. Choose from the following indexing options:

Repository Source Index	
Run Incremental Index	If the repository has already been indexed, this will start an incremental scan, otherwise an "initial" scan. This is especially useful if you have not set the repository to poll automatically, or it is set with a long poll (interval) period.
Re-index	Delete the existing cache and re-index the repository from scratch. For large or slow repositories this may take some time, during which some functionality will be unavailable. This action will also restart the repository.
Review-Revision Data Index	<i>(Crucible only)</i> Re-index all the Crucible review data associated with the current repository.
Line Count Data Index	Re-index the linecount data used to generate the LOC (Lines Of Code) charts. The linecount data will be recalculated in daily buckets based on the server timezone.
Subversion Non-Version Properties (revprops)	<i>(Subversion repositories only)</i> Re-scan the non-versioned properties of the repository (e.g. check-in comments if so enabled) and amend the entry in FishEye and Crucible accordingly. Set the revision numbers to scan from and to. In Subversion it is possible to enable non-versioned properties (e.g. the check-in comments) to be updated by committers. When this happens, FishEye will not automatically pick up the updates. By rescanning specific revisions, FishEye will rescan the non-versioned properties and amend the entry in FishEye accordingly.

Screenshot: FishEye Index Maintenance menu

Repository "applinks"

Summary

Maintenance

SCM Details

Updates

Linkers

Permissions

Include/Exclude Paths

Hidden Directories

Tarballs

Commit Messages

Other Settings

Indexing Status

Current Activity: **Idle**

Status Message: **N/A**

Last Index Update: **10 seconds ago**

Repository Source Index

Run Incremental Index: ?

Re-index: ?

Review-Revision Data Index

Re-index: ?

Line Count Data Index

Re-index: ?

Subversion Non-Versioned Properties (revprops)

Re-scan: from revision to revision ?

Test Connection

Save

Close Without Saving

Repository Options

You can configure options for repositories that affect settings like caching, permissions, polling period, etc. Some configuration options are only defined for individual repositories, whereas others affect all repositories (defaults).

On this page:

- [Configuring Options for a Specific Repository](#)
- [Configuring Options Affecting All Repositories](#)

Configuring Options for a Specific Repository

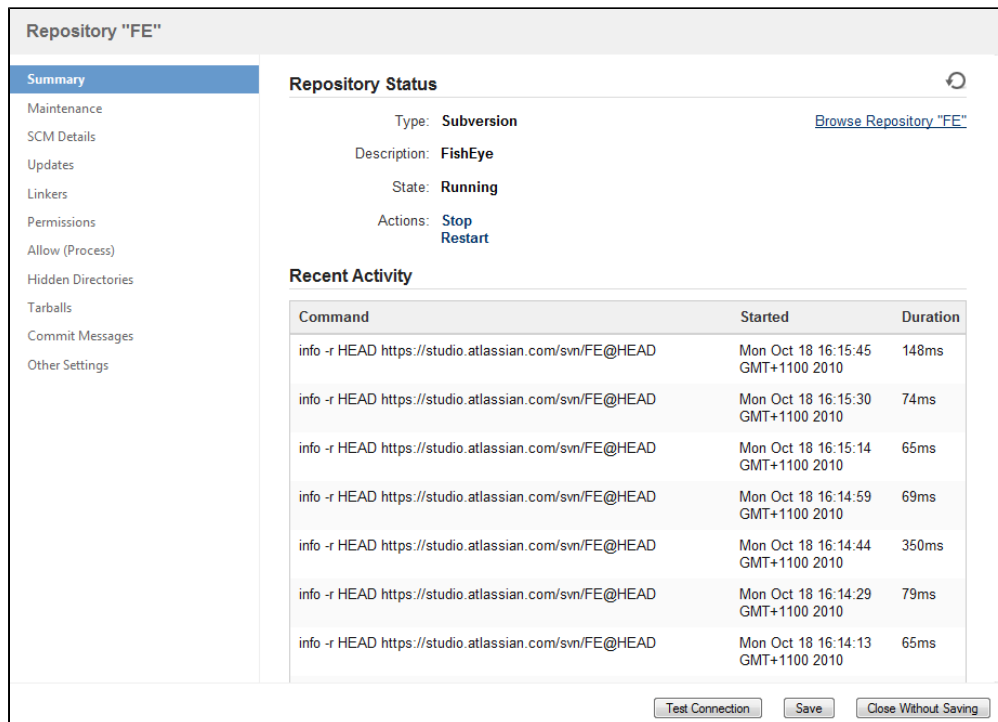
To configure options for a specific repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the name of the repository in the list of repositories to manage the settings for that repository. The repository options for the repository will be displayed with the 'Summary' page shown.
 - Click the relevant link in the left menu to configure the desired options. None of the fields will be editable, unless you click the '**Edit**' button at the bottom of the page:
 - Summary — View the repository type (e.g. Subversion) and description. You can also perform common repository operations from this screen, see [Operations](#).
 - Maintenance — See [Indexer](#).
 - SCM Details — See [Configuring repository details](#).
 - Updates — See [Updater](#).
 - Linkers — See [Linkers](#).
 - Permissions — See [Permissions](#).
 - Allow (Process) — See [Include Exclude Paths](#).
 - Hidden Directories — See [Hidden directories](#).
 - Tarballs — See [Tarball settings](#).
 - Commit Messages — See [Commit Message Syntax](#).
 - Other Settings — See [Watches](#) and [Properties](#).
 - Click the '**Test Connection**' button to test FishEye's connection to the repository.
 - Click the '**Save**' button to save your changes and close the repository options window.

- Click the '**Close Without Saving**' button to abandon any changes you made and close the repository options window.

⚠ Some changes will require the repository to be restarted, while others will require the repository to be re-indexed. FishEye will advise you if this is the case when you make the change. You can restart a repository from the '**Repositories**' screen, by using the '**Restart**' function for the desired repository. See [Managing your repositories](#) for further details.

Screenshot: Configuring repository options



The screenshot shows the 'Repository "FE"' configuration page. On the left is a sidebar with a 'Summary' tab selected, and other tabs like 'Maintenance', 'SCM Details', 'Updates', 'Linkers', 'Permissions', 'Allow (Process)', 'Hidden Directories', 'Tarballs', 'Commit Messages', and 'Other Settings'. The main content area is titled 'Repository Status' and includes a refresh icon. It displays the following information:

- Type: Subversion
- Description: FishEye
- State: Running
- Actions: Stop, Restart

Below this is a 'Recent Activity' table:

Command	Started	Duration
info -r HEAD https://studio.atlassian.com/svn/FE@HEAD	Mon Oct 18 16:15:45 GMT+1100 2010	148ms
info -r HEAD https://studio.atlassian.com/svn/FE@HEAD	Mon Oct 18 16:15:30 GMT+1100 2010	74ms
info -r HEAD https://studio.atlassian.com/svn/FE@HEAD	Mon Oct 18 16:15:14 GMT+1100 2010	65ms
info -r HEAD https://studio.atlassian.com/svn/FE@HEAD	Mon Oct 18 16:14:59 GMT+1100 2010	69ms
info -r HEAD https://studio.atlassian.com/svn/FE@HEAD	Mon Oct 18 16:14:44 GMT+1100 2010	350ms
info -r HEAD https://studio.atlassian.com/svn/FE@HEAD	Mon Oct 18 16:14:29 GMT+1100 2010	79ms
info -r HEAD https://studio.atlassian.com/svn/FE@HEAD	Mon Oct 18 16:14:13 GMT+1100 2010	65ms

At the bottom of the page are three buttons: 'Test Connection', 'Save', and 'Close Without Saving'.

Configuring Options Affecting All Repositories

To configure options affecting all repositories:

- Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
- Click the '**Defaults**' link under the '**Repository Settings**' section in the '**Admin**' menu. The 'Repository Defaults' screen will display with the 'Linkers' tab shown.
- Click the relevant tab to configure the desired repository default setting:
 - Linkers — See [Linkers](#).
 - Permissions — See [Permissions](#).
 - Allow (Process) — See [Include Exclude Paths](#).
 - Hidden Directories — See [Hidden directories](#).
 - Tarballs — See [Tarball settings](#).
 - Commit Messages — See [Commit Message Syntax](#).
 - Other Settings — See [Watches](#) and [Properties](#).
- Click the '**Save changes**' button to save your changes. If you need to revert any changes, click the '**revert**' link next to the '**Save changes**' button.

Screenshot: Configuring default repository options

Repository Defaults

Linkers Permissions Allow (Process) Hidden Directories Tarballs Commit Messages Other Settings

FishEye and Crucible can search for text patterns that appears in commit messages or comments and generate hyperlinks for them. This is often used to link bug IDs to your issue tracking system. 🟢

Description	Type	Reg. Exp.	
project code linker for Edgy JIRA	Simple	(?i)(CLEP CCD CLN CBL CRU SHAJ)-\d+	⚙️
Atlassian-internal JIRA linker for the "CENQUA" project only	Simple	(?i)CENQUA-\d+	⚙️
sac issue linker	Simple	(?i)(FSH CRC)-\d+	⚙️
Add...			

Save changes or revert

Related Topics

- [Authentication](#)
- [Commit Message Syntax](#)
- [Configuring repository details](#)
- [Hidden directories](#)
- [Include Exclude Paths](#)
- [Indexer](#)
- [Linkers](#)
- [Operations](#)
- [Permissions](#)
- [Properties](#)
- [Store diff info](#)
- [Tarball settings](#)
- [Text file size limiting](#)
- [Updater](#)
- [Watches](#)

Authentication

SCMs typically support simple username and password combinations for authentication. However, in FishEye and Crucible, more complex authentication styles are supported for the following repository types:

- Mercurial
- Git

The configuration of these authentication styles in FishEye is described below. Note, the authentication style can also be configured as part of a particular repository's definition.

i Although Subversion supports more complex authentication styles like ssh and ssl certificate authentication, these are not currently configurable in FishEye. When using the bundled svnkit library, these are typically configured using Java property definitions.

On this page:

- [Authentication Styles](#)
 - [No Authentication](#)
 - [Generate Key Pair for SSH](#)
 - [Upload Private Key for SSH](#)
 - [Password](#)
 - [Security Considerations when using SSH Key Pairs](#)
 - [Using ssh-agent](#)

- [SSH Connectivity Tools under Windows](#)

Authentication Styles

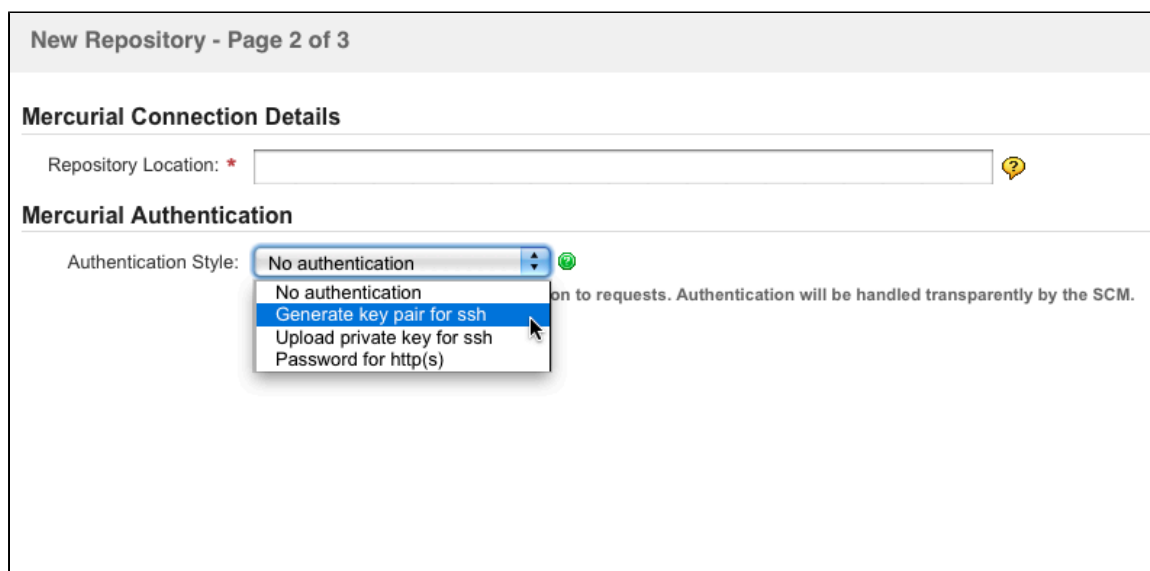
The authentication styles are configured as part of a particular repository definition.

There are four authentication styles for a repository:

- [No Authentication](#) — Used where no authentication is required to access a repository.
- [Generate Key Pair for SSH](#) — A SSH key is generated and managed by FishEye to access the repository.
- [Upload Private Key for SSH](#) — A SSH key is created and managed externally to FishEye.
- [Password for http\(s\)](#) — This is for the username / password style of authentication with http and https URLs (also known as basic-auth).

When a FishEye repository type supports these authentication styles, you will see an Authentication section during repository configuration (e.g. the "Mercurial Authentication" section in the screenshot below). This section will be available when initially [defining a repository configuration](#) and also in the "SCM Details" section when [maintaining a repository](#).

Screenshot: Selecting an Authentication Style



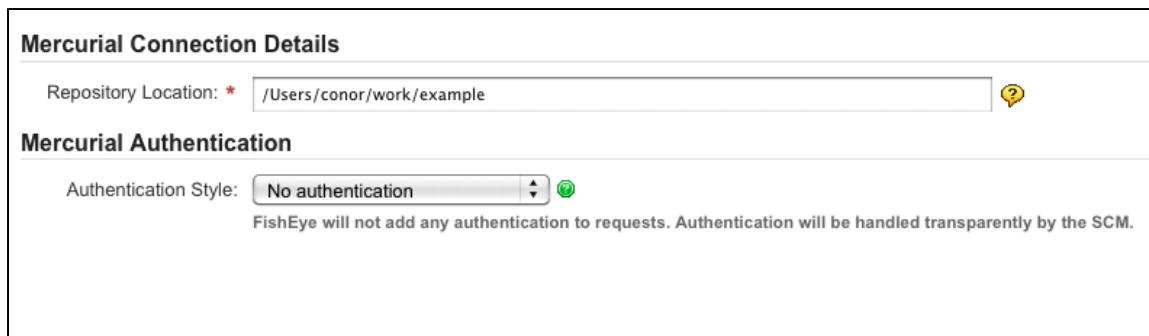
No Authentication

This is, the simplest authentication style. It means that FishEye can connect to the repository using Git or Mercurial without any credentials or authentication. If you have supplied credentials via another mechanism (such as ssh-agent or a default passphraseless private key for a ssh connection) Git or Mercurial will use this automatically.

This option is appropriate for,

1. public repositories which grant read-only access to anonymous users, or
2. internal repositories on a secure network which allow anonymous reads to network users, or
3. repositories which are accessible to the FishEye server on the local filesystem.

Screenshot: No Authentication selected for a local repository



Mercurial Connection Details

Repository Location: * ?

Mercurial Authentication

Authentication Style: ?

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

FishEye's access to repositories is always read-only, so it can easily be used with repositories which only require authentication and authorisation for write operations.

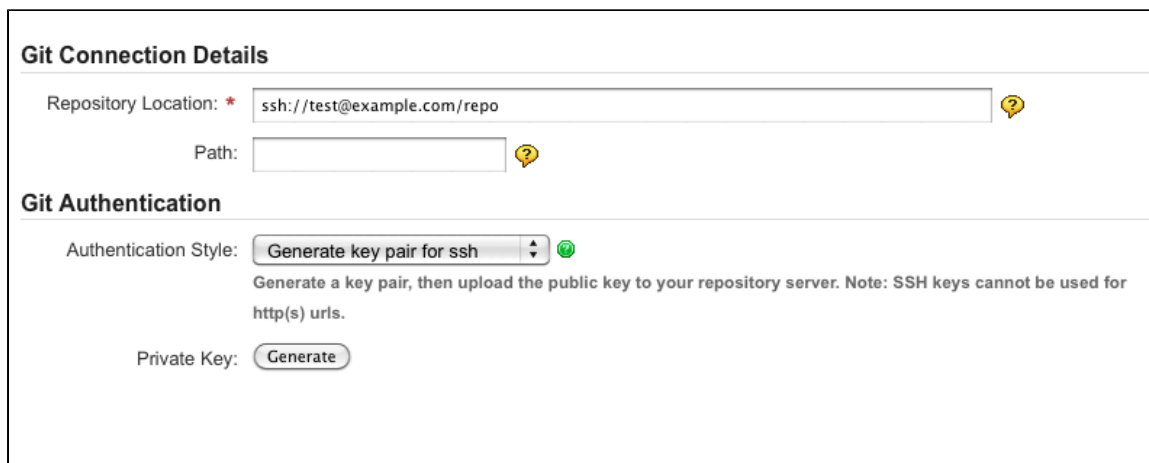
- If you specify "No Authentication" but enter a repository URL which includes a username and password, FishEye will remove the password from the URL and set the authentication style to "Password", storing the password specified in the password field.

Generate Key Pair for SSH

This is the most secure option for ssh access, as the private key is never transmitted across the network. If you want to manage repository access using SSH keys, you can choose this style of authentication to have FishEye generate and manage the public/private key pair for you.

Click the '**Generate**' button to generate the key pair (see "Generating a SSH key pair" screenshot below). FishEye will generate and store the public and private key pair. The key is specific to the repository being indexed.

Screenshot: Generating a SSH key pair



Git Connection Details

Repository Location: * ?

Path: ?

Git Authentication

Authentication Style: ?

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:

The public key will be displayed to allow you to copy it to your repository server and to associate the key with your user account (see "Public Key of Generated Key displayed" screenshot below). The private key is stored by FishEye and never exposed to users or administrators.

Screenshot: Public Key of Generated Key displayed

Git 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:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGCpD/T5PEvaW4q1Nyvob0ZHI4wnv32WIHLU7iwwAZ
sOZumylj+0jM
/yWgECK6d6nDAcdldOoqjjoVgl7q1FojAStuMx+1blzZYY0cY+YAO4gLtz1DPAXHivWLRy5inZT
RjDYmQTgsvhi1d5Jrl3L68d0icsxh/Yk6CoabAYIHvOQ== gittest on FishEye at
http://raleigh.sydney.atlassian.com:8080
```

Copy this public key to your repository server

[Remove](#)

If you wish to change the key, you can remove the existing key by clicking the '**Remove**' link and then generate a new key.

When using SSH keys, you will typically specify a username as part of the URL you use to access the repository.

Public hosting systems such as Bitbucket and GitHub provide simple web-based mechanisms for associating public keys with your account. For these systems, a generic username is used in the repository URL and it is the key that determines the account. See the screenshots below for examples of how to associate keys with Bitbucket and GitHub accounts.

Screenshot: Key management on Bitbucket

SSH Keys

Your SSH Public keys – Help: [Using SSH](#)

» ssh-rsa AAAAB3NzaC1yc2EAAAABIw atlassian.com

[Add key](#)

Or you can upload your public key file here.

[Browse...](#) [Upload key](#)

Screenshot: Key management on GitHub

SSH Public Keys

[Job Profile](#)

conor test (edit)

Title

Key

or [cancel](#)

Our RSA fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48

Upload Private Key for SSH

If you are using SSH keys for repository access and already have an SSH key or you would prefer to manage your SSH keys yourself, this authentication style allows you to upload the private key to FishEye. Please note however, that FishEye can only use passphrase-less SSH keys (To vote for support for private keys with passphrases, see <http://jira.atlassian.com/browse/CRUC-5579>).

 Generating a key within FishEye, as described [above](#), is **our preferred approach** to using SSH keys. We believe it is advisable to use a private key for a single purpose. Different access needs should use different keys. This option should only be used if you must use an existing key.

If you choose to use this option, you will be transmitting your private key across the network to your FishEye server. We **strongly recommend** that you [enable https for FishEye](#) before you do this.

The private key is uploaded by your web browser file upload operation. As soon as FishEye completes the file upload, it verifies that the provided key is in fact a valid private SSH key and that it does not have a passphrase. If you wish to change the stored key you must remove the current key by clicking the '**Remove**' link and then upload the new key.

Screenshot: Private Key Upload

Mercurial Connection Details

Repository Location: 

Mercurial Authentication

Authentication Style: 

Warning: you should understand the risks of uploading private keys. Note: SSH keys cannot be used for http(s) urls.

Private Key: Private key has been uploaded
[Remove](#)

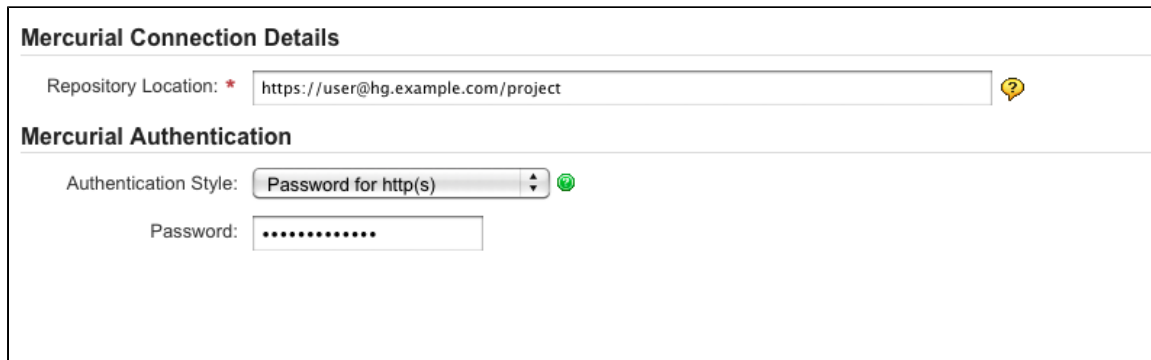
 **Tip:** If you are uploading under OSX and using 10.6 or later, you can press command-shift-period to display

hidden files and directories. This makes it easier to access ssh key files stored in their default .ssh directory

Password

This authentication style is used when using http/https to access your repository with a username and password. For these repositories, the repository URL includes the username for the account and the password is configured in the **'Password'** field (see "Password Authentication" screenshot below).

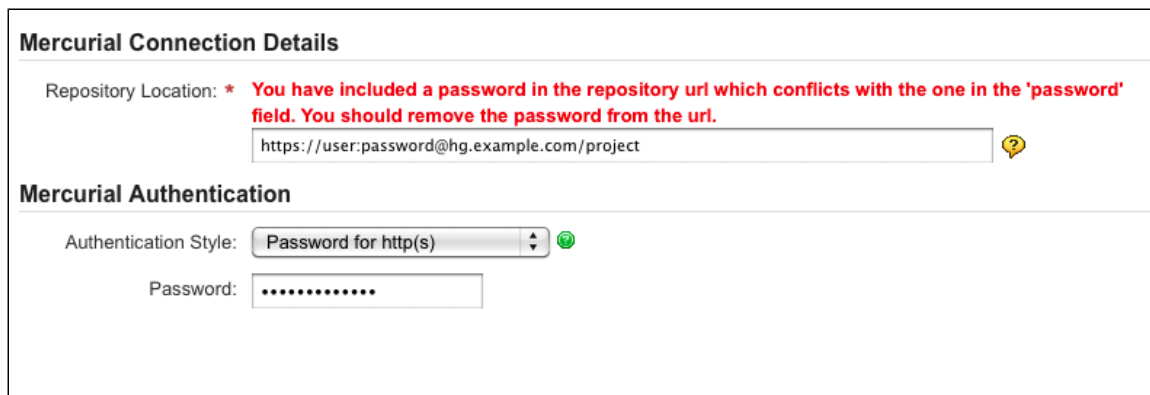
Screenshot: Password Authentication



The screenshot shows a web form titled "Mercurial Connection Details". It has two sections. The first section, "Mercurial Connection Details", contains a "Repository Location:" label with a red asterisk, followed by a text input field containing "https://user@hg.example.com/project" and a yellow question mark icon. The second section, "Mercurial Authentication", contains an "Authentication Style:" label followed by a dropdown menu set to "Password for http(s)" with a green checkmark icon. Below this is a "Password:" label followed by a text input field filled with ten dots.

It is possible to specify both the username and password in a repository URL. If you do this FishEye will warn you to remove the password from the URL definition, as it would conflict with the password specified in the password field. When using password authentication, FishEye will always manage the password separately from the repository URL.

Screenshot: Do not provide password in URL



The screenshot shows the same "Mercurial Connection Details" form. In the "Repository Location:" section, the text input field now contains "https://user:password@hg.example.com/project". Above the input field, a red warning message reads: "You have included a password in the repository url which conflicts with the one in the 'password' field. You should remove the password from the url." The "Mercurial Authentication" section remains the same, with "Authentication Style:" set to "Password for http(s)" and a "Password:" field with ten dots.



- For git repositories, when using http basic authentication, git must be compiled with libcurl support. If this is not the case, authentication will fail with a message similar to: **error: git was compiled without libcurl support.** You will need to update your git installation or select an alternative authentication style.
- Password authentication using ssh is not currently supported by FishEye. You must use a key when using ssh.

Notes

Security Considerations when using SSH Key Pairs

The `Generate Key Pair for SSH` option is the safest option to use, as private keys are never transmitted across the network. If an attacker gains access to your private key, they have access to all the services that allow that keypair.

Since FishEye only requires read access to your accounts, it is best to create a user that only has read access to the repositories you want to index and add the public key to their account settings. This way, even if the private key is compromised, while your source can be accessed, you can easily revoke that user's access and/or remove their public keys and the attacker will not be able to change or modify any of your data.

Using ssh-agent

If you are using a Unix style system with SSH key authentication and do not want to perform key management in FishEye at all, it is possible to launch FishEye in the context of an ssh-agent process. When FishEye launches sub-processes to interact with the repository, the ssh command in those sub-processes will inherit access to the ssh keys in the ssh-agent process. This approach would allow you to use ssh keys with passphrases. Please refer to your operating system documentation for more information on ssh-agent.

SSH Connectivity Tools under Windows

If your FishEye installation is using Windows, you must use [OpenSSH](#) for SSH connectivity, rather than alternatives like TortoiseHg or Putty.

Download and install [Git for Windows](#), as this includes OpenSSH. Be careful about the options you choose during the install:

- When asked how to adjust your PATH environment, select the option 'Run Git and included Unix tool from the Windows Command prompt'. This will ensure that Git and its associated tools (including OPENSSH) are available to the FishEye server via the PATH environment variable.
- When asked for the SSH executable, select the option 'Use OpenSSH'. The FishEye server doesn't work with alternatives like TortoiseHg or Putty.

Commit Message Syntax

FishEye can render Wiki Markup in commit messages. This page contains instructions for turning this setting on and off. You can set this option as a global default, as well as individually per-repository.

 When using Crucible, you can also have [Wiki Markup rendering in review comments](#) and review descriptions.

On this page:

- [Configuring Commit Message Syntax for a Specific Repository](#)
- [Configure Commit Message Syntax for All Repositories](#)

Configuring Commit Message Syntax for a Specific Repository

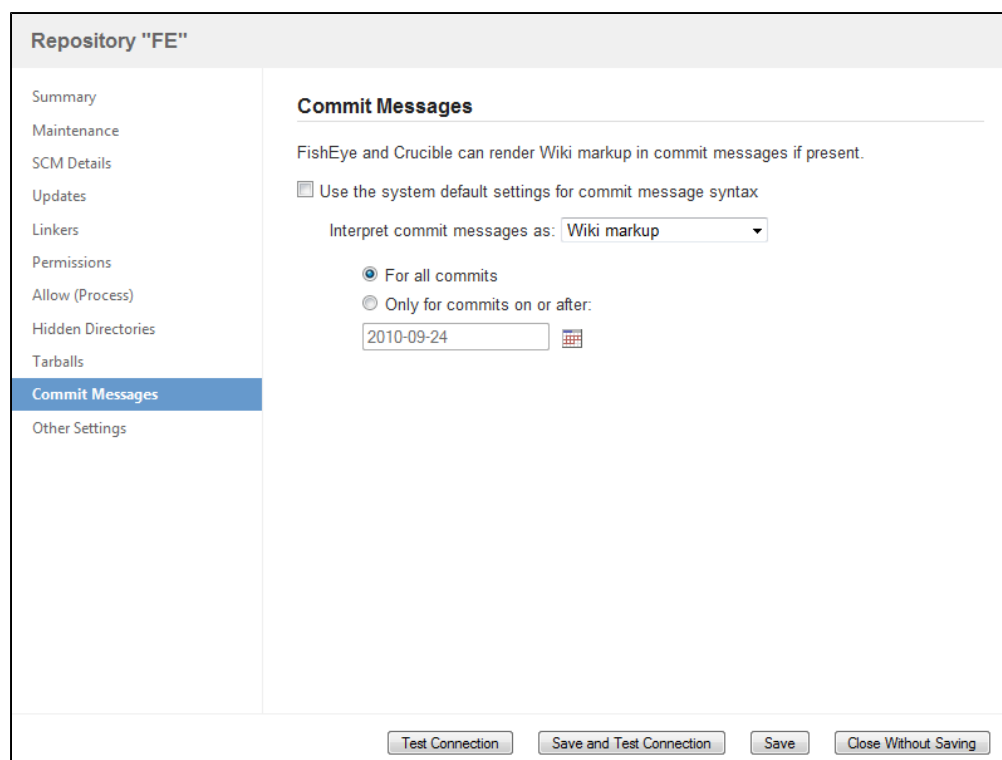
To configure commit message syntax for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the '**Repositories**' link. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the '**Name**' column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
4. Click the '**Commit Messages**' tab. The 'Commit Messages' screen will be displayed (see screenshot below).
5. Update the commit message syntax for this repository as follows:
 - Tick the '**Use the system default settings for commit message syntax**', if you want to use the system default settings for commit message syntax. The following fields will only be enabled if you do not tick this checkbox:
 - Select '**Interpret commit messages as**' to be either 'Plain text' or 'Wiki markup'. This determines whether wiki markup is rendered in commit messages or not.
 - Choose whether you want commit message syntax to be applied to all commits or commits

after a specified date.

6. Save your changes.

Screenshot: Configuring commit message syntax for a specific repository



The screenshot shows the 'Repository "FE"' configuration page. On the left is a sidebar with a list of tabs: Summary, Maintenance, SCM Details, Updates, Linkers, Permissions, Allow (Process), Hidden Directories, Tarballs, **Commit Messages** (highlighted), and Other Settings. The main content area is titled 'Commit Messages' and contains the following text: 'FishEye and Crucible can render Wiki markup in commit messages if present.' Below this is a checkbox labeled 'Use the system default settings for commit message syntax'. Underneath is a label 'Interpret commit messages as:' followed by a dropdown menu currently set to 'Wiki markup'. There are two radio button options: 'For all commits' (which is selected) and 'Only for commits on or after:'. The second option has a text input field containing '2010-09-24' and a small calendar icon to its right. At the bottom of the page are four buttons: 'Test Connection', 'Save and Test Connection', 'Save', and 'Close Without Saving'.

Configure Commit Message Syntax for All Repositories

To configure commit message syntax for all repositories:

1. Click the '**Defaults**' link. The 'Repository Defaults' will be displayed.
2. Click the '**Commit Messages**' tab. The 'Commit Messages' screen will be displayed (see screenshot below).
3. Update the commit message syntax for this repository as follows:
 - Select '**Interpret commit messages as**' to be either 'Plain text' or 'Wiki markup'. This determines whether wiki markup is rendered in commit messages or not. The default is 'Plain text'.
 - Choose whether you want commit message syntax to be applied to all commits or commits after a specified date.
4. Save your changes.

Screenshot: Configuring commit message syntax for all repositories

Repository Defaults

[Linkers](#)
[Permissions](#)
[Allow \(Process\)](#)
[Hidden Directories](#)
[Tarballs](#)
[Commit Messages](#)
[Other Settings](#)

FishEye and Crucible can render Wiki markup in commit messages if present. 

Interpret commit messages as: Wiki markup

☒ For all commits
☐ Only for commits on or after: 2010-09-24 

[or revert](#)

Configuring repository details

You can define the **SCM Details** for a repository when [adding](#) or [managing](#) a repository. The SCM details will vary depending on the type of repository that you are adding, e.g. the SCM details for a Subversion repository differ to the SCM details for a Git repository.

On this page:

- [Configuring the SCM details for a repository](#)
- [Notes](#)

Related pages:

- [Adding an external repository](#)
- [Repository Options](#)

Configuring the SCM details for a repository

To configure the SCM details for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **'Repositories'** link. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the **'Name'** column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
(You can also click the cog icon () and click **'View'** from the dropdown menu to open the Repository Options)
4. Click **'SCM Details'** in the left menu. The details for your repository will be displayed.
5. Click the **'Edit'** button at the bottom of the dialogue. The 'SCM Details' will refresh with the fields editable.
6. Update the fields as desired.
 - **'Description'** — Enter a description for the repository. This is used to describe the repository in FishEye screens.
 - **'Store Diff Info'** — Tick this checkbox, if you want FishEye to cache information about file diffs in its database. This is required for some FishEye features. This setting will be enabled by default for new repositories. See the [Notes](#) section below for important information on this setting.
 - The rest of the fields will vary depending on the SCM. See the following topics:
Note, the topics below are for adding a repository, however the fields are the same.
 - [CVS](#)
 - [Git](#)
 - [Mercurial](#)
 - [Perforce](#)

- [Subversion](#)

7. Click the '**Apply and Test Connection**' button to test your changes.

- If you are happy with your changes, click the '**Save**' button to save your changes.
- If you need to abandon your changes, click the '**Revert**' button.

Screenshot: Configuring the SCM details for a Subversion repository

Repository "FE"

Summary

Maintenance

SCM Details

Updates

Linkers

Permissions

Allow (Process)

Hidden Directories

Tarballs

Commit Messages

Other Settings

Basic Details

Description: FishEye

Store Diff Info: ☒

SVN Connection Details

SVN URL: * https://studio.atlassian.com/

Path: FE

Username: tpetersen

Password:

SVN Options

Block Size: 1000

Svn Operation Timeout:

Throttle Connections/s: 0.0

Charset: default (UTF-8)

Access Code:

Start Revision:

Initial Import: Do not import

Follow Base Moves: ☐

SVN Symbolic Rules

Use Built-in Symbolic Rules: ☐

And then Apply the Following Rules:

/trunk/..., /branches/NAME/..., /tags/NAME/...

Customise

	Regex	Name	Logical Path Prefix
Trunk	trunk/(/ \$)	trunk	
Branches	branches/([^\s/]+)	\${1}	
Tags	tags/([^\s/]+)	\${1}	

Test Path

Test Connection

Edit

Close

Notes

About the 'Store Diff Info' setting

Enabling the 'Store Diff Info' setting means that FishEye will store information about file diffs in its database, i.e. FishEye will store summary of what lines are added and removed between subsequent versions of the same file. You will still be able view file diffs regardless of whether this setting is enabled or disabled.

⚠ Please note, you need to perform a **full re-index of your repository after enabling this setting, for FishEye to collect the diff information for all revisions in your repository.**

Please also take note of the following information about this setting:

- Disabling the 'Store Diff Info' setting will **disable [per-author line graphs](#)**.
- Disabling the 'Store Diff Info' setting also removes the author/revision information in the diff display.
- Diff information is always stored for [CVS repositories](#). Note, that a full re-index is required to enable per

author charts after upgrading from FishEye 1.4.3 or earlier.

- Enabling the 'Store Diff Info' setting will allow FishEye to perform text searches of lines added and removed, in addition to the text search of the trunk head.
- Enabling the 'Store Diff Info' setting for [Perforce repositories](#) will force FishEye to make extra requests to your depot in order to collect the diffs. This may substantially increase the time it takes to scan your repository (scan times for other repositories, like CVS and Subversion, are not affected by the 'Store Diff Info' setting). If your Perforce repository was created before FishEye 1.5, this setting will be disabled by default.

Hidden directories

You can configure unused (deprecated) directories as 'hidden' in FishEye. Hidden directories will not be shown to users in the FishEye user interface, unless a user has specifically elected to view hidden directories in their [user profile](#). This can be useful if you have old directories that you don't want cluttering the screens by default. Please note, FishEye will still index and cache these directories.

 Note, this administration option does not affect the user's option of hiding **empty** directories and deleted files, when browsing a repository. See [Browsing through a Repository](#) for details.

On this page:

- [Configuring hidden directories](#)
 - [Configuring hidden directories for a specific repository](#)
 - [Configure hidden directories for all repositories](#)
- [Adding a hidden directory](#)

Configuring hidden directories

Configuring hidden directories for a specific repository

To configure hidden directories for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **Repositories** link. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the 'Name' column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
4. Click the **Hidden Directories** tab. The 'Hidden Directories' screen will be displayed (see screenshot below).
5. Configure the hidden directories for the repository as desired:
 - Click **Add** to add a new hidden directory pattern. See the [Adding a Hidden Directory](#) section below for further instructions.
 - Click the cog icon (⚙) next to a hidden directory pattern and then **Edit** to edit the pattern. The editable 'Hidden Directory' dialogue is identical to the 'Add Hidden Directory' dialogue. See the [Adding a Hidden Directory](#) section below for further instructions.
 - Click the cog icon (⚙) next to a hidden directory pattern and then **Delete** to delete the hidden directory pattern.
6. Save your changes.

Configure hidden directories for all repositories

To configure hidden directories for all repositories,

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **Defaults** link. The 'Repository Defaults' will be displayed.
3. Click the **Hidden Directories** tab. The 'Hidden Directories' screen will be displayed (see screenshot

below).

4. Configure the hidden directories for all repositories as desired:
 - Click the '**Add...**' link to add a new hidden directory pattern. Hidden directory patterns for your repository defaults will be inherited by all repositories. See the [Adding a Hidden Directory](#) section below for further instructions.
 - Click the cog icon (⚙️) next to a hidden directory pattern and then **Edit** to edit the pattern. The editable 'Hidden Directory' dialogue is identical to the 'Add Hidden Directory' dialogue. See the [Adding a Hidden Directory](#) section below for further instructions.
 - Click the cog icon (⚙️) next to a hidden directory pattern and then **Delete** to delete the hidden directory pattern.
5. Save your changes.

Screenshot: Configuring hidden directories for all repositories

Repository Defaults

Linkers Permissions Allow (Process) **Hidden Directories** Tarballs Commit Messages Other Settings

Any directory marked as hidden will not be displayed by FishEye and Crucible in its user interface (although it will still be indexed and cached). This is useful to hide unused, deprecated or erroneously committed directories. 🧠

Pattern	Case Sensitive
No hidden directories defined.	
Add...	

Save changes or revert

Adding a hidden directory

To add a new hidden directory,

1. Navigate to the hidden directories for a specific repository or the repository defaults, as described [above](#), and click **Add** to add a new hidden directory pattern. The 'Add Hidden Directory' dialogue will be displayed.
2. Complete the fields on the dialogue as follows:
 - **Pattern** — Enter the pattern for directories that you want to be hidden in FishEye.
 - **Case Sensitive** — By default, hidden directory patterns are case-sensitive but they can be configured to be case-insensitive. If your repository is set to be case-insensitive ([Perforce](#)) then hidden directory patterns will also be case-insensitive, regardless of how the individual patterns are configured.
3. Click **Add** to add the hidden directory pattern.

Screenshot: Adding a new hidden directory pattern

Include Exclude Paths

By default, FishEye will cache and index your whole repository, and present all of this information to users. You can allow FishEye to process certain parts of your repository and/or disallow it from accessing other parts, by configuring inclusion/exclusion patterns for FishEye to follow. These patterns are set in the 'Include/Exclude Paths' repository option.

On this page:

- [Configuring Include/Exclude Paths](#)
 - [Configuring Include/Exclude for a Specific Repository](#)
 - [Configure Include/Exclude for All Repositories](#)
- [Adding an Include](#)
- [Adding an Exclude](#)
- [Notes](#)
 - [Include/Exclude Processing](#)
 - [About Setting the Repository Root](#)

Configuring Include/Exclude Paths

You can configure the include/exclude paths option for a specific repository or configure linkers for all repositories.

Configuring Include/Exclude for a Specific Repository

To configure the include/exclude paths option for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the '**Repositories**' link. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the '**Name**' column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
4. Click the '*Include/Exclude Paths' tab. The 'Include/Exclude Paths' screen will be displayed (see screenshot below).
5. Configure the includes/excludes for the repository as desired:
 - Click the '**Add...**' link to add a new include/exclude pattern. See the [Adding an Include](#) and [Adding an Exclude](#) sections below for further instructions.
 - Click the cog icon (⚙️) next to a linker and click '**Edit**' from the dropdown menu to edit an include/exclude pattern. The editable 'Linker' dialogue is identical to the 'Add a Link' dialogue. See the [Adding an Include](#) and [Adding an Exclude](#) sections below for further instructions.
 - Click the cog icon (⚙️) next to a linker and click '**Delete**' from the dropdown menu to delete the include/exclude pattern.
6. Save your changes.

Configure Include/Exclude for All Repositories

To configure the allow (process) for all repositories:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the '**Defaults**' link. The 'Repository Defaults' will be displayed.
3. Click the '**Include/Exclude Paths**' tab. The 'Include/Exclude Paths' screen will be displayed (see screenshot below).
4. Configure the includes/excludes for all repositories as desired:
 - Click the '**Add...**' link to add a new include/exclude pattern. See the [Adding an Include](#) and [Adding an Exclude](#) sections below for further instructions.
 - Click the cog icon (⚙️) next to a linker and click '**Edit**' from the dropdown menu to edit an include/exclude pattern. The editable 'Linker' dialogue is identical to the 'Add a Link' dialogue. See the [Adding an Include](#) and [Adding an Exclude](#) sections below for further instructions.
 - Click the cog icon (⚙️) next to a linker and click '**Delete**' from the dropdown menu to delete the include/exclude pattern.
5. Save your changes.

Screenshot: Configuring Include/Exclude Paths for all repositories

Repository Defaults

[Linkers](#)
[Permissions](#)
[Allow \(Process\)](#)
[Hidden Directories](#)
[Tarballs](#)
[Commit Messages](#)
[Other Settings](#)

By default, FishEye and Crucible will cache and index your whole repository, and present all of this information to users. Configure includes and excludes to control which parts of your repository will be accessed. 🌱

Includes

Tree	Case Sensitive
No includes defined. Everything will be included by default.	
Add...	

Excludes

Pattern	Case Sensitive
/CVSROOT/	false ⚙️
Add...	

[Save changes](#)
[or revert](#)

Adding an Include

The '**Includes**' subsection of the '**Include/Exclude Paths**' screen defines what subtrees of your repository FishEye will index. FishEye defaults to including 'everything'. If you specify some 'include' directories, then FishEye will process only those directories (and all their subdirectories). For instance, you might want to do this to limit FishEye to the subset of active projects in your repository. Each include specifies the path to a subtree to be processed. Paths are expressed relative to the repository root configured in the repository configuration.

📘 Includes do not support [Antglobs](#). They are also defined relative to where FishEye connects to your repository. To match the 'tags' subtree, simply use /tags. Please note that [Excludes](#) can still make use of Antglobs.

To add a new include:

1. Navigate to the "Include/Exclude Paths" for a specific repository or the repository defaults, as described [above](#), and click the 'Add...' link to add a new include. The 'Add an Include' dialogue will be displayed.
2. Complete the fields on the dialogue as follows:
 - **'Path Prefix'** — Enter the tree of your repository that you want to include. See the [example](#) below.
 - **'Case Sensitive'** — Tick this checkbox if you want your include to be case-sensitive. By default, Includes are case-sensitive but they can be configured to be case-insensitive. If your repository is set to be case-insensitive ([Perforce](#)) then Includes will also be case-insensitive, regardless of how the individual Includes are configured.

Example:

- Including directories:

```
PROJECT1
```

The code above **includes** /PROJECT1 and all its children (sub-directories and their contents). You could specify /PROJECT1/ and /PROJECT2/ to include both of these directories in FishEye's indexing.

Adding an Exclude

The '**Excludes**' subsection of the '**Include/Exclude Paths**' screen allows you to specifically exclude files and directories from those which have been included in indexing. FishEye will not process these files and directories. Each exclude is an [Antglob Pattern](#).

To add a new exclude:

1. Navigate to the "Include/Exclude Paths" for a specific repository or the repository defaults, as described [above](#), and click the 'Add...' link to add a new exclude. The 'Add an Exclude' dialogue will be displayed.
2. Complete the fields on the dialogue as follows:
 - **'Pattern'** — Enter the pattern for directories/files that you want to exclude. See the [examples](#) below.
 - **'Case Sensitive'** — Tick this checkbox if you want your exclude to be case-sensitive. By default, Excludes are case-sensitive but they can be configured to be case-insensitive. If your repository is set to be case-insensitive ([Perforce](#)) then Excludes will also be case-insensitive, regardless of how the individual Excludes are configured.

Examples:

- Excluding directories:

```
/PROJECT2/
```

The code above **excludes** /PROJECT2 and all its children (sub-directories and their contents).

- Excluding file types:

```
**/*.OBJ
```

The code above **excludes** any OBJ (object) files.

 Changes to Includes and Excludes do not take effect until the repository is restarted. If you do not re-index when changing the includes and excluded, files and directories which have been indexed prior to the update will remain visible in FishEye.

Notes

Include/Exclude Processing

When processing includes and excluded, FishEye merges the includes and excludes from the repository itself with those from the repository defaults. The repository's specific includes and excludes take priority over those of the repository defaults. Once merged, FishEye processes include definitions first and then excludes. If there are any includes defined a path must match at least one of those includes to be considered. If there are no include patterns defined, all paths are considered to be included. Once includes have been processed, a path which is a candidate for processing is tested against any defined excludes. If the path matches any of the exclude patterns, the path is excluded and not included in FishEye.

About Setting the Repository Root

When you are setting the Allow (Process), you should be aware that the options on this page only act on the parts of the repository that lie under the level of the repository root, which you configure as a directory location in your repository. In other words, FishEye can only access directories "lower" than the repository root. For example, consider a repository with the following structure:

```
/CORE/2007/LEGACY/  
/CORE/2008/PROJECT1/  
/CORE/2008/PROJECT2/
```

In this case, you could set the repository root (or 'Path') to be /CORE/2008/. In that situation, you would be able to include or exclude the /PROJECT1/ and /PROJECT2/ directories, but the /CORE/2007/LEGACY/ directory would not be available. To have FishEye index all of the directories in this repository, you would need to set the repository root path to be /CORE/. Then, you could use the includes and excludes to add and remove directories under /CORE/ from FishEye's scan. For more information, see the [Subversion configuration page](#) and read the 'Path' options.

Indexer

You can manually perform a number of indexing functions on your repositories in FishEye.

 Note, an indexing scan for a repository can take hours or possibly days, depending on the size of the repository, network speed, machine performance and other factors.

To access indexing functions for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the name of the repository.
3. Click **Maintenance** in the left-hand panel.
4. Choose from the following indexing options:

Repository Source Index	

Run Incremental Index	If the repository has already been indexed, this will start an incremental scan, otherwise an "initial" scan. This is especially useful if you have not set the repository to poll automatically, or it is set with a long poll (interval) period.
Re-index	Delete the existing cache and re-index the repository from scratch. For large or slow repositories this may take some time, during which some functionality will be unavailable. This action will also restart the repository.
Review-Revision Data Index	(Crucible only) Re-index all the Crucible review data associated with the current repository.
Line Count Data Index	Re-index the linecount data used to generate the LOC (Lines Of Code) charts. The linecount data will be recalculated in daily buckets based on the server timezone.
Subversion Non-Version Properties (revprops)	(Subversion repositories only) Re-scan the non-versioned properties of the repository (e.g. check-in comments if so enabled) and amend the entry in FishEye and Crucible accordingly. Set the revision numbers to scan from and to. In Subversion it is possible to enable non-versioned properties (e.g. the check-in comments) to be updated by committers. When this happens, FishEye will not automatically pick up the updates. By rescanning specific revisions, FishEye will rescan the non-versioned properties and amend the entry in FishEye accordingly.

Screenshot: FishEye Index Maintenance menu

Repository "aplinks"

Summary
Maintenance
SCM Details
Updates
Linkers
Permissions
Include/Exclude Paths
Hidden Directories
Tarballs
Commit Messages
Other Settings

Indexing Status

Current Activity: **Idle**

Status Message: **N/A**

Last Index Update: **10 seconds ago**

Repository Source Index

Run Incremental Index: ?

Re-index: ?

Review-Revision Data Index

Re-index: ?

Line Count Data Index

Re-index: ?

Subversion Non-Versioned Properties (revprops)

Re-scan: from revision to revision ?

Test Connection
Save
Close Without Saving

Linkers

FishEye can render any issue IDs or Bug IDs that appear in commit messages or comments as hyperlinks. Users can click the links to quickly navigate to the relevant issue/bug in your issue/bug tracker. This link

rendering relies on FishEye detecting certain substrings in commit messages and comments. You can configure the "linker" patterns that FishEye uses to detect these substrings for each repository in FishEye.

Please note, if you have set up [JIRA integration](#) between your FishEye instance and a JIRA server, you will need to disable linkers for that JIRA server. Otherwise, the linkers will override the integration features described on that page (which includes linkers).

On this page:

- [Configuring linkers](#)
 - [Configuring linkers for a specific repository](#)
 - [Configure linkers for all repositories](#)
- [Adding a linker](#)
- [Example linkers](#)
 - [JIRA examples](#)
 - [Bamboo examples](#)
 - [Bugzilla examples](#)
- [Configuring the SyntaxDef field](#)

Configuring linkers

You can configure linkers for a specific repository or configure linkers for all repositories.

Configuring linkers for a specific repository

To configure the linkers for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **'Repositories'** link. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the **'Name'** column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
4. Click the **'Linkers'** tab. The 'Linkers' screen will be displayed (see screenshot below).
5. Configure the linkers for the repository as desired:
 - Click the **'Add...'** link to add a new linker. See the [Adding a Linker](#) section below for further instructions.
 - Click the cog icon (⚙️) next to a linker and click **'Edit'** from the dropdown menu to edit a linker. The editable 'Linker' dialogue is identical to the 'Add a Link' dialogue, except that you cannot change the link type. See the [Adding a Linker](#) section below for further instructions.
 - Click the cog icon (⚙️) next to a linker and click **'Delete'** from the dropdown menu to delete the linker.
6. Save your changes.

Configure linkers for all repositories

To configure the linkers for all repositories:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **'Defaults'** link. The 'Repository Defaults' will be displayed, with the **'Linkers'** tab shown (see screenshot below).
3. Configure the linkers for all repositories as desired:
 - Click the **'Add...'** link to add a new linker. Any new linkers added in the repository defaults will be inherited by all repositories. See the [Adding a Linker](#) section below for further instructions.
 - Click the cog icon (⚙️) next to a linker and click **'Edit'** from the dropdown menu to edit a linker. The

editable 'Linker' dialogue is identical to the 'Add a Link' dialogue, except that you cannot change the link type. See the [Adding a Linker](#) section below for further instructions.

- Click the cog icon (⚙️) next to a linker and click **Delete** from the dropdown menu to delete the linker.

4. Save your changes.

Screenshot: Viewing linkers for all repositories

Repository Defaults

Linkers Permissions Allow (Process) Hidden Directories Tarballs Commit Messages Other Settings

FishEye and Crucible can search for text patterns that appears in commit messages or comments and generate hyperlinks for them. This is often used to link bug IDs to your issue tracking system. 🐞

Description	Type	Reg. Exp.
project code linker for Edgy JIRA	Simple	(?i)(CLEP CCD CLN CBL CRU SHAJ)-\d+
Atlassian-internal JIRA linker for the "CENQUA" project only	Simple	(?i)CENQUA-\d+
sac issue linker	Simple	(?i)(FSH CRC)-\d+
Add...		

Save changes or revert

Adding a linker

To add a new linker:

1. Navigate to the linkers for a specific repository or the repository defaults, as described [above](#), and click the **Add...** link to add a new linker. The 'Add a Linker' dialogue will be displayed.
2. Complete the fields on the dialogue as follows:
 - **Linker Type** — Select 'Simple' or 'Advanced'. Choose 'Simple' unless you are an experienced developer.
 - **Description** — Enter a description for your linker.
 - **Regular Expression** ('Simple' Linker Type) — Enter the regular expression defining the pattern that FishEye will look for when rendering links. FishEye uses the [Java regular expression language](#), which is based on Perl 5 regular expressions. You can test your regular expressions on this [online test page](#).

✓ **Tip:** If you want your regex to be case insensitive, put `(?i)` at the start of the regex.
 - **Href** ('Simple' Linker Type) — Enter the destination of the link. For example, `http://jira.mycompany.com/browse/${0}`
 - **SyntaxDef** ('Advanced' Linker Type) — See [Configuring the SyntaxDef Field](#) section below.
3. Click the **Add** button to add the linker.
4. Save your changes.

✓ **Tip:** See the [Example Linkers](#) section below for examples of linkers to JIRA and Bugzilla servers.

Screenshot: Adding a new linker

Add a Linker

Linker Type:

Description:

Regular Expression:

Href:

Example linkers

Here are some examples of simple linkers.

JIRA examples

⚠ Do not create linkers for a JIRA server, if you have already set up [JIRA integration](#) between your FishEye instance and that JIRA server.

- To link any occurrence of a JIRA-style issue to JIRA:

```
Regex: [a-zA-Z]{2,}-\d+
Href: http://jirahost:8080/browse/${0}
```

The regular expression above matches any sequence of two or more alphabetical characters, followed by a dash, followed by a number, which comprise the format of JIRA issue IDs (such as AB-123 or ABC-123 or ABCDE-123). Replace `jirahost` with the hostname of the desired JIRA instance.

- To link a specific set of JIRA projects (e.g. JRA, CONF and CRUC) to a JIRA instance:

```
Regex: (JRA|CONF|CRUC)-\d+
Href: http://jirahost:8080/browse/${0}
```

The regular expression above matches only specific JIRA issue keys with any number, like JRA-123 or CONF-123 or CRUC-123. Replace `jirahost` with the hostname of the desired JIRA instance.

Bamboo examples

- To link to specific Bamboo builds:


```
Regex: (ABC)-[a-zA-Z]+\d+
Href: http://bamboohost/browse/${0}
```

The regular expression above matches Bamboo build IDs like ABC-MAIN-123 or ABC-BRANCH-123. These will then be made links to the build reports in your Bamboo instance. Replace bamboohost with the hostname of the desired bamboo instance.

Bugzilla examples

- To link bug numbers that occur at the start of a line to Bugzilla:

```
Regex: ^BUG: (\d+)
Href: http://bugzilla/bugzilla/show_bug.cgi?id=${1}
```

- To link bug numbers that occur after the word bug and optionally whitespace, ":" or "#" (e.g. Bug123, bug:123, or BUG #123):

```
Regex: (?i)bug[#\s|:]*(\d+)
Href: http://bugzilla/bugzilla/show_bug.cgi?id=${1}
```

The regular expressions above matches Bugzilla bug IDs. These will then be made links to build reports in your Bamboo instance.

Configuring the SyntaxDef field

 This is an advanced feature, intended for use by experienced developers only.

Please note;

- The '**description**' is optional.
- You will want to define the three region properties as in the example below.

This example matches numbers that appear after a PR: and might be separated by commas or 'and':

```
PR: 123
PR: 123 456
PR: 123, 456
PR: 123, 456 and 789
```

```
<syntaxdef>
/PR:\s*((\d+\s*(and|,)?\s*)+)/i : {
  nummatcher(${1});
}
context nummatcher {
  /\d+/ : {
    region {
      href="http://issues.apache.org/bugzilla/show_bug.cgi?id=${0}";
    }
  }
}
}
```

Understanding the SyntaxDef example:

- The first regex matches means: match "PR:" followed by a sequence of numbers ("d+") separated by whitespace, 'and' or commas.
- The "context" means in the above match, link each number individually.

Operations

You can carry out operations such as stopping a repository scan, restarting a repository scan, disabling it, etc, via the [repositories list](#) or [repository summary](#) screens.

To access the operations for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
 2. Click the **Repositories** link. The list of repositories set up in your FishEye instance will be displayed.
 3. Click the name of the repository, (under the **Name** column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
 4. A number of repository operations will be available next to the **Actions** field:
 - **Stop** ('Running' repositories only) — Stop the repository.
 - **Restart** ('Running' repositories only) — Restart the repository.
 - **Enable** ('Disabled' repositories only) — Enable the repository.
-  More repository actions are available via the [Repositories list](#). See [Managing your repositories](#).

Permissions

You can control access to specific repositories or all repositories in FishEye. Access can be restricted by user groups. You can also allow anonymous access, so that anyone can browse your repositories. For more information, see [Associating a group with a repository](#).

On this page:

- [Configuring permissions for a specific repository](#)
- [Configuring permissions for all repositories](#)

Configuring permissions for a specific repository

You can configure permissions for a specific repository, as described in this section. You can also configure permissions for all repositories at once, [as described below](#).

To configure the permissions for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.

2. Click **Repositories** (under 'Repository Settings').
3. Click the name of the repository (in the 'Name' column).
4. Click **Permissions** in the left-hand panel. See the screenshot below.
5. Configure permissions for the repository as required:

Repository defaults	Select Use repository defaults to have access to this repository controlled by the default permission settings for all repositories .
Anonymous access	Select the Can Read permission for the 'Anonymous' group at the top of the list.
User group access	Select Can Read for the groups to which you wish to grant that permission.

6. Click **Save**.

Screenshot: Configuring permissions for a specific repository

Repository "CLOV"

Summary
Maintenance
SCM Details
Updates
Linkers
Permissions
Include/Exclude Paths
Hidden Directories
Tarballs
Commit Messages
Other Settings

Access by Built-in Groups

☐ Use repository defaults. [Configure defaults.](#)

	Can Read
Anonymous	☐
All logged-in users	☐
KLDevToolsSupport	☒
beac-users	☐
caviar-sync	☐
crucible-users-non-crowd	☐
atlassian-build-admins	☐
atlassian-cfo	☐
atlassian-cpp-team	☐
atlassian-crowd	☐

Test Connection
Save
Close Without Saving

Configuring an LDAP restriction

If the LDAP Authentication type is configured in your instance, you can optionally restrict a repository based on an LDAP query.

When an LDAP Restriction is configured for a repository, then the logic to determine whether you can read from a repository is:

Anon access is allowed:

YES

Logged in AND...

all logged in users are allowed & no

LDAP Restriction defined?

YES

authorised by LDAP Restriction?

YES

authorised by group membership?

YES

Otherwise:

NO

Configuring permissions for all repositories

You can configure permissions for all repositories at once, as described in this section. You can also configure permissions for a specific repository, as described above.

To configure the permissions for all repositories:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click **Defaults** (under 'Repository Settings').
3. Click **Permissions** in the left-hand panel.
4. Configure anonymous access and group permissions for all repositories at once as required:

Anonymous access	Select the Can Read or Can Write permissions for the 'Anonymous' group at the top of the list.
User group access	Select Can Read or Can Write for the groups to which you wish to grant those permissions. Note that the Can Write permission only applies to Git repositories that are managed by FishEye and Crucible, and is not used for any other repositories.

5. Remember to click **Save changes** at the bottom of the screen.

Screenshot: Configuring permissions for all repositories

Repository Defaults

[Linkers](#)
[Permissions](#)
[Include/Exclude Paths](#)
[Hidden Directories](#)
[Tarballs](#)
[Commit Messages](#)
[Other Settings](#)

Access by Built-in Groups

You can instruct FishEye and Crucible to restrict the groups of users who can access the repositories.

	Can Read	Can Write
Anonymous	☐	☐
All logged-in users	☒	☐
KLDevToolsSupport	☒	☐
beac-users	☒	☐
caviar-sync	☒	☐
crucible-users-non-crowd	☒	☐
fisheye3	☒	☐

Properties

You can configure a number of default or repository-specific properties for FishEye, including enabling/disabling changeset discussions and enabling/disabling the changelog calendar. A default property is inherited by all repositories. A default property may be overridden at the repository level.

On this page:

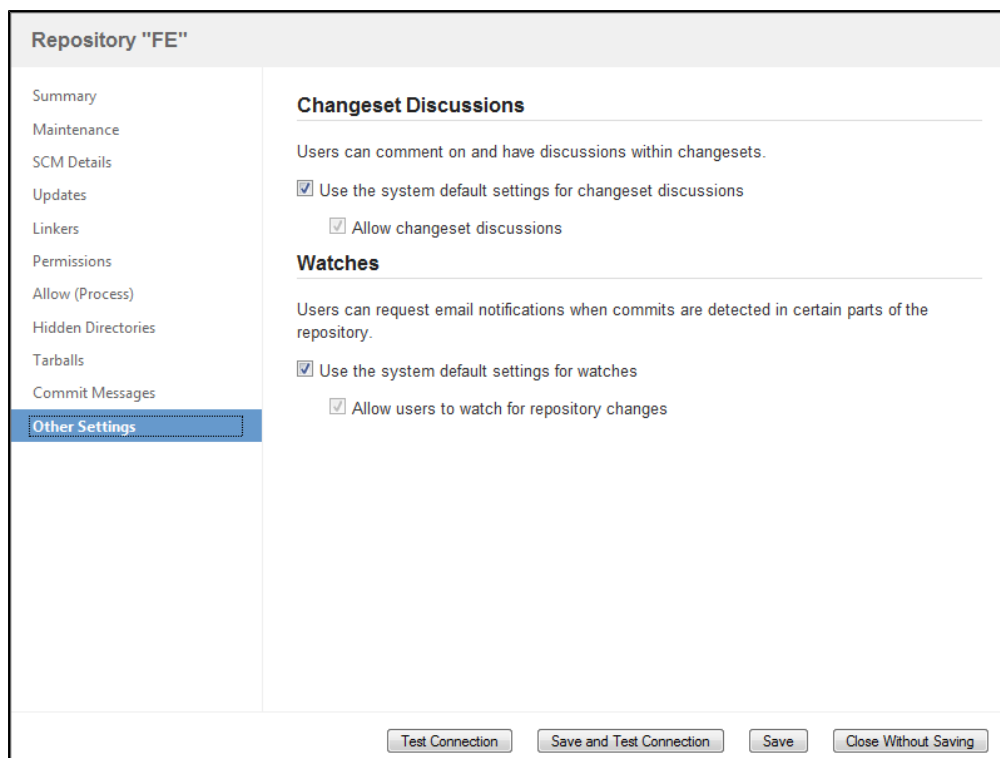
- [Configuring properties for a specific repository](#)
- [Configure properties for all repositories](#)

Configuring properties for a specific repository

To configure properties for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **Repositories** link. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the **Name** column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
4. Click the **Other Settings** tab. The other settings screen will be displayed (see screenshot below).
5. Update the settings for this repository as desired:
 - Tick the **Use the system default settings for changeset discussions**, if you want to use the system default settings for changeset discussions, i.e. if you enable/disable changeset discussions in your repository defaults, changeset discussions will be enabled/disabled for this repository respectively.
 - Tick the **Allow changeset discussions** checkbox, if you want to allow users to be able to comment on and have discussions in changesets. This checkbox will only be enabled if the **Use the system default settings for changeset discussions** checkbox is not ticked.
 - Configure settings for watches. See [Watches](#) for more information.
6. Save your changes.

Screenshot: Configuring properties for a specific repository



Repository "FE"

- Summary
- Maintenance
- SCM Details
- Updates
- Linkers
- Permissions
- Allow (Process)
- Hidden Directories
- Tarballs
- Commit Messages
- Other Settings**

Changeset Discussions

Users can comment on and have discussions within changesets.

- ☒ Use the system default settings for changeset discussions
- ☒ Allow changeset discussions

Watches

Users can request email notifications when commits are detected in certain parts of the repository.

- ☒ Use the system default settings for watches
- ☒ Allow users to watch for repository changes

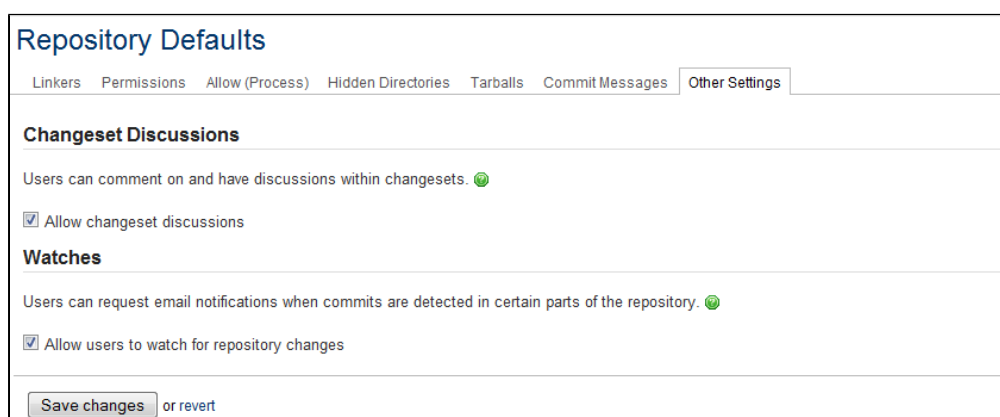
Test Connection Save and Test Connection Save Close Without Saving

Configure properties for all repositories

To configure properties for all repositories:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **Defaults** link. The 'Repository Defaults' will be displayed.
3. Click the **Other Settings** tab. The other settings screen will be displayed (see screenshot below).
4. Update the settings for all repositories as desired:
 - Tick the **Allow changeset discussions** checkbox, if you want to allow users to be able to comment on and have discussions in changesets. This checkbox will only be enabled if the **Use the system default settings for changeset discussions** checkbox is not ticked.
 - Configure settings for watches. See [Watches](#) for more information.
5. Save your changes.

Screenshot: Configuring properties for all repositories



Repository Defaults

Linkers Permissions Allow (Process) Hidden Directories Tarballs Commit Messages **Other Settings**

Changeset Discussions

Users can comment on and have discussions within changesets. 🟢

- ☒ Allow changeset discussions

Watches

Users can request email notifications when commits are detected in certain parts of the repository. 🟢

- ☒ Allow users to watch for repository changes

Save changes or revert

Store diff info

The Store Diff Info setting is configured as part of the general repository details. See [Configuring repository details](#) for more information.

About the 'Store Diff Info' Setting

Enabling the 'Store Diff Info' setting means that FishEye will store information about file diffs in its database, i.e. FishEye will store summary of what lines are added and removed between subsequent versions of the same file. You will still be able view file diffs regardless of whether this setting is enabled or disabled.

 Please note, you need to perform a **full re-index** of your repository after enabling this setting, for FishEye to collect the diff information for all revisions in your repository.

Please also take note of the following information about this setting:

- Disabling the 'Store Diff Info' setting will **disable** [per-author line graphs](#).
- Disabling the 'Store Diff Info' setting also removes the author/revision information in the diff display.
- Diff information is always stored for [CVS repositories](#). Note, that a full re-index is required to enable per author charts after upgrading from FishEye 1.4.3 or earlier.
- Enabling the 'Store Diff Info' setting will allow FishEye to perform text searches of lines added and removed, in addition to the text search of the trunk head.
- Enabling the 'Store Diff Info' setting for [Perforce repositories](#) will force FishEye to make extra requests to your depot in order to collect the diffs. This may substantially increase the time it takes to scan your repository (scan times for other repositories, like CVS and Subversion, are not affected by the 'Store Diff Info' setting). If your Perforce repository was created before FishEye 1.5, this setting will be disabled by default.

Tarball settings

FishEye contains a feature that will build an archive of a directory tree. This feature is disabled by default. The 'Tarball Settings' repository option allows you to customise tarball settings in the Repository Defaults and on a per-repository basis. You can set a limit on the number of files that a tarball can contain, as well as selectively disable the creation of tarballs for certain directories or directory trees.

On this page:

- [Configuring tarball settings for a specific repository](#)
- [Configure tarball settings for all repositories](#)

Configuring tarball settings for a specific repository

To configure tarball settings for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the '**Repositories**' link. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the '**Name**' column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
4. Click the '**Tarballs**' tab. The 'Tarballs' screen will be displayed (see screenshot below).
5. Tick the '**Allow users to download repository trees as tarballs**' checkbox, if you want to allow users to download repository trees as tarballs. The following field will be enabled:
 - '**Restrict the number of files in tarballs to**' — Choose either 'No limit', or choose 'No more than <number> files' and specify the number of files.
6. Configure '**Tarball Excludes**' for your repository. The creation of tarballs will not be permitted for directories/directory trees that are excluded:
 - Click the '**Add...**' link to add a new tarball exclude. Enter the '**Base Directory**' and select whether to '**Exclude subdirectories**' in the dialogue that displays.
 - Click the cog icon (⚙) next to a tarball exclude and click '**Edit**' from the dropdown menu to edit the pattern. The editable Tarball dialogue is identical to the 'Add an Exclude' dialogue. See the [Adding a Tarball Exclude](#) section below for further instructions.
 - Click the cog icon (⚙) next to a tarball exclude and click '**Delete**' from the dropdown menu to delete the tarball exclude.

7. Save your changes.

Screenshot: Configuring watches for a specific repository

Configure tarball settings for all repositories

To configure watches for all repositories,

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **Defaults** link. The 'Repository Defaults' will be displayed.
3. Click the **Tarballs** tab. The 'Tarballs' screen will be displayed (see screenshot below).
4. Tick the **'Allow users to download repository trees as tarballs'** checkbox, if you want to allow users to download repository trees as tarballs. The following field will be enabled:
 - **'Restrict the number of files in tarballs to'** — Choose either 'No limit', or choose 'No more than <number> files' and specify the number of files.
5. Configure **'Tarball Excludes'** for your repository. The creation of tarballs will not be permitted for directories/directory trees that are excluded:
 - Click the **'Add...'** link to add a new tarball exclude. Enter the **'Base Directory'** and select whether to **'Exclude subdirectories'** in the dialogue that displays.
 - Click the cog icon (⚙) next to a tarball exclude and click **'Edit'** from the dropdown menu to edit the pattern. The editable Tarball dialogue is identical to the 'Add an Exclude' dialogue. See the [Adding a Tarball Exclude](#) section below for further instructions.
 - Click the cog icon (⚙) next to a tarball exclude and click **'Delete'** from the dropdown menu to delete the tarball exclude.
6. Save your changes.

Screenshot: Configuring tarball settings for all repositories

Repository Defaults

[Linkers](#)
[Permissions](#)
[Allow \(Process\)](#)
[Hidden Directories](#)
[Tarballs](#)
[Commit Messages](#)
[Other Settings](#)

FishEye and Crucible allows users to download an tar archive file (tarballs) of directory trees.

☐ Allow users to download repository trees as tarballs

Restrict the number of files in tarballs to:

☒ No limit
 ☐ No more than: files

Tarball Excludes:

Base Directory	Exclude Subdirs
No tarball excludes configured.	
Add...	

[or revert](#)

Text file size limiting

How Text File Size Limits are Applied

Content Indexing

FishEye maintains several indexes which are used to provide fast searching of files in the repository. One of these indexes is used for searching file content. FishEye places a number of restrictions on this index to keep it manageable and to maintain its usefulness in searching.

- FishEye only indexes text files
- Only the HEAD revision of any file is maintained in the index. New revisions of a file will replace older revisions in the index.
- FishEye only indexes files which are below a size limit
- For most repository types, FishEye only indexes the main branch (i.e. trunk in Subversion, MAIN in CVS, etc)

The limits applied to the content index do not affect the indexing of files in the other indexes maintained by FishEye, notably the metadata index which indexes the file against the commit metadata (comment, author, date, etc). All other FishEye operations are thus unaffected by the application of the content indexing limit.

The default file size limit is 5MB and, in FishEye 2.6, this limit was made configurable in the admin user interface. Changing this limit only affects indexing operations. Files which have already been indexed are not affected - i.e. they are not removed from the content index. Only files which are indexed after the change are affected by the changed limit. A re-index is necessary to apply the new limit to all files in the repository.

When a file is not indexed, FishEye will log a warning message on the console:

```
Not indexing contents of HEAD of /path/to/file.txt because its size (6M) is too large (limit is 5M)
```

Large File Viewing

In FishEye 2.7, for Subversion repositories, the file size limit is also applied to the UI treatment of text files. Any files which exceed the file size limit have the following additional restrictions:

- The file content is not viewable in the FishEye and Crucible user interfaces.
- The number of lines in the file are not counted towards the Lines of Code (LoC) totals
- Added and Removed text is not indexed and, consequently, is not searchable.

Such files are now marked internally as "too large" within FishEye, and rendered similarly to binary files. A link is available to download the file content and in Crucible file-level review comments can be made.

As above, changing the file size limit does not affect files which have already been indexed. Such files will continue to be treated according to the setting in effect when the file was originally indexed. A re-index is necessary to apply a new limit to files which have already been indexed. This can mean that a large file appears to abruptly acquire the "too large" designation when a new revision is added after upgrading FishEye, or after the file size limit has been changed.

The application of this limit allows FishEye to avoid certain Subversion operations which are resource intensive (typically I/O, memory or time and often all three). Such large text files are typically not practical to view in FishEye and review within Crucible in any case.

Choosing a value for the text file limit

FishEye applied the original 5MB limit as a reasonable compromise for what would typically be the largest text files in a repository. Increasing this value will allow you to index and see the content for larger text files at the cost of increased resource usage.

Updater

You can configure how often FishEye/Crucible checks if there have been new commits to the repository. In most cases, you will not need to change the default settings for this "updater" which polls your repository.

To configure the updater for a repository,

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click **Repositories**. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the 'Name' column in the list of repositories).
4. Click the **Updates** tab.
5. Check **Use the system default settings for updates** to use the default updater settings. In most cases, you can leave this ticked. If you want to customise the updater settings, clear the checkbox and complete the following fields, as required:

Disable polling	If you disable repository polling, FishEye/Crucible will only pick up new commits to this repository if you scan it manually. You can perform a manual scan of the repository via the command line or by using the ' Scan For Updates ' function on the repository 'Maintenance' screen.
Polling Interval	Specify how often FishEye should check for new commits to the repository. Use the following terms to specify particular units of time: s, m, h, d, w, mo, y (for seconds, minutes, hours, days, weeks, months and years respectively). For example, 10s. The default value for this field is 60 seconds.

6. The following additional settings can be configured for [CVS](#) repositories. You should not have to change the default values in most cases:

History File	The location of the CVS history file. If you use a relative path, it must be relative to the CVS directory specified for this repository. The default value for this field is <code>./CVSR00T/history</code> .
Full Scan Interval	How often FishEye will perform a full scan of the repository. Use the following terms to specify particular units of time: second, minute, hour, day, week, month, year. For example, <code>10second</code> . If you do not specify a unit of time, the default unit is days. For example, <code>15min</code> . Set this field to '0' to disable the periodic full scan (you can still manually perform a full scan via the command line). The default value for this field is 15 minutes.
Strip Prefix	The prefix to strip off files found in the history file. This will make the files relative to this repository's CVS directory. You <i>must</i> configure a strip prefix if the CVS directory specified is not the root of the CVS repository. For example, your CVS is located at <code>/usr/local/cvsroot</code> , but you have specified <code>/usr/local/cvsroot/foo/bar</code> as the CVS directory of this repository. In this case, you will need to set the history file as <code>../../../../CVSR00T/history</code> and specify a strip prefix of <code>foo/bar</code> .

 FishEye will monitor your CVS history file `CVSR00T/history` to determine what has changed in your repository. FishEye will also periodically scan the whole repository. CVS is not always configured to create a history file. Talk to your CVS administrator if you need assistance.

7. Click **Save**.
8. Restart FishEye, if you have changed any values. Your repositories will be scanned in order (depending on the number of threads you have configured) once FishEye starts up. The polling interval and full scan period (CVS only) will then be determined from the time that this initial scan is complete. For example, if you have set your polling interval to one hour, your next scan will begin one hour after your initial scan is complete.

Screenshot: Repository Updates Settings

Watches

On this page:

- [Configuring watches for a specific repository](#)
- [Configure watches for all repositories](#)

FishEye has a "watch" notification system that allows users to receive email notifications when commits are detected. Users can opt in for these notifications by "[watching](#)" a particular repository/activity stream. You can configure FishEye to enable or disable watches for all users for a specific repository or for all repositories.

⚠ Please note, if you want to enable watches for your repositories, you must configure a valid [SMTP server](#).

Configuring watches for a specific repository

To configure watches for a repository:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click **Repositories**. The list of repositories set up in your FishEye instance will be displayed.
3. Click the name of the repository, (under the 'Name' column in the list of repositories). The Repository Options for the repository will be displayed in a dialogue.
4. Click the **Other Settings** tab. The 'Watches' section will be shown on the screen that is displayed (see screenshot below).
5. Configure the watches for the repository as desired:
 - **Use the system default settings for watches** — Check this if you want to use the system default settings for watches, i.e. if you enable/disable watches in your repository defaults, watches will be enabled/disabled for this repository respectively.
 - **Allow users to watch for repository changes** — Check or clear this to enable/disable watches for the repository. This checkbox will only be enabled if **Use the system default settings for watches** is not checked.
6. Save your changes.

Screenshot: Configuring watches for a specific repository

Repository "FE"

Summary
Maintenance
SCM Details
Updates
Linkers
Permissions
Allow (Process)
Hidden Directories
Tarballs
Commit Messages
Other Settings

Changeset Discussions

Users can comment on and have discussions within changesets.

☒ Use the system default settings for changeset discussions

☒ Allow changeset discussions

Watches

Users can request email notifications when commits are detected in certain parts of the repository.

☒ Use the system default settings for watches

☒ Allow users to watch for repository changes

Test Connection Save and Test Connection Save Close Without Saving

Configure watches for all repositories**To configure watches for all repositories:**

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click the **Defaults** link. The 'Repository Defaults' will be displayed.
3. Click the **Other Settings** tab. The 'Watches' section will be shown on the screen that is displayed (see screenshot below).
4. Configure the watches for all repositories as desired:
 - **Allow users to watch for repository changes** — Check or clear this to enable or disable watches for all repositories that been configured to use the system default settings for watches.
5. Save your changes.

Screenshot: Configuring watches for all repositories

Repository Defaults

Linkers Permissions Allow (Process) Hidden Directories Tarballs Commit Messages **Other Settings**

Changeset Discussions

Users can comment on and have discussions within changesets. 🟢

☒ Allow changeset discussions

Watches

Users can request email notifications when commits are detected in certain parts of the repository. 🟢

☒ Allow users to watch for repository changes

Save changes or revert

Managing users and groups in FishEye

You can implement access control using a set of 'built-in' users stored in the FishEye database, or you can have FishEye look in an [external authentication source](#) for users, passwords and permissions.

Note that anonymous access to FishEye is allowed by default. You can disable [anonymous access](#) at a global level and per repository.

To configure external authentication settings, click **Authentication** (under 'Security Settings') in the Admin area. See [Configuring External Authentication Sources](#) for more information.

To set up built-in users, choose from the pages below for specific information on settings:

- [Creating a user](#)
- [Creating a group](#)
- [Adding a user to a group](#)
- [Associating a group with a repository](#)
- [Configuring avatar settings](#)
- [Configuring anonymous access](#)
- [Editing a user's details](#)
- [Deleting or deactivating a user](#)

 FishEye provides a pluggable architecture to allow arbitrary forms of authorisation and authentication.

Screenshot: Authentication Settings

Admin Menu	Authentication Settings																																																				
Repository Settings - Repository List (new...) - Repository Defaults - Review Coverage Project Settings - Project List (new...) Global Settings - Server Settings Security - Users - User Mapping - Avatar Settings - Groups - Administrators - ViewCVS URL Mappings - Change Admin Password - Customize Crucible Defect Classifications - Permission Schemes - Trusted Applications - JIRA Servers - Customize Front Page System - Database Configuration - Sys-Info/Support - Content - Backup - Plugins - Shutdown	Permissions Summary <table border="1"> <thead> <tr> <th></th> <th>Allow anon</th> <th>Groups</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>Global:</td> <td>NO (Turn On)</td> <td></td> <td></td> </tr> <tr> <td>Crucible:</td> <td>NO (Turn On)</td> <td></td> <td></td> </tr> <tr> <td>Repository Default:</td> <td>NO</td> <td>none set</td> <td>Edit</td> </tr> <tr> <td colspan="4">Per-repository:</td> </tr> <tr> <td>AGMP:</td> <td>NO (default)</td> <td>default</td> <td>Edit</td> </tr> <tr> <td>applinks:</td> <td>NO (default)</td> <td>default</td> <td>Edit</td> </tr> <tr> <td>atlassian:</td> <td>NO (default)</td> <td>default</td> <td>Edit</td> </tr> <tr> <td>BM:</td> <td>NO (default)</td> <td>default</td> <td>Edit</td> </tr> <tr> <td>CLOV:</td> <td>NO</td> <td>atlassian-dev atlassian-staff KLDevToolsSupport</td> <td>Edit</td> </tr> <tr> <td>contrib:</td> <td>NO (default)</td> <td>default</td> <td>Edit</td> </tr> <tr> <td>Critter:</td> <td>NO (default)</td> <td>default</td> <td>Edit</td> </tr> <tr> <td>FE:</td> <td>NO</td> <td>atlassian-dev atlassian-staff crucible-users-non-crowd KLDevToolsSupport</td> <td>Edit</td> </tr> </tbody> </table> Built-in Public Signup : OFF (Turn On) Use CAPTCHA : ON (Turn Off) User managed mappings : ON (Turn Off) Username Option Force Lowercase Username : OFF (Turn On) Email Visibility User email visibility: ☒ Public ☐ Hidden ☐ Visible to logged in users only Note: emails which are visible are obfuscated and will not be harvested by web spiders. Crowd Authentication Application Name: cenqua-dogfood-fisheye Crowd URL: http://extranet-crowd.private:8080/crowd/services/ Auto-add users into FishEye: true Single sign on (SSO) enabled: false Synchronise Period: false Re-Sync Edit Remove		Allow anon	Groups	Action	Global:	NO (Turn On)			Crucible:	NO (Turn On)			Repository Default:	NO	none set	Edit	Per-repository:				AGMP:	NO (default)	default	Edit	applinks:	NO (default)	default	Edit	atlassian:	NO (default)	default	Edit	BM:	NO (default)	default	Edit	CLOV:	NO	atlassian-dev atlassian-staff KLDevToolsSupport	Edit	contrib:	NO (default)	default	Edit	Critter:	NO (default)	default	Edit	FE:	NO	atlassian-dev atlassian-staff crucible-users-non-crowd KLDevToolsSupport	Edit
	Allow anon	Groups	Action																																																		
Global:	NO (Turn On)																																																				
Crucible:	NO (Turn On)																																																				
Repository Default:	NO	none set	Edit																																																		
Per-repository:																																																					
AGMP:	NO (default)	default	Edit																																																		
applinks:	NO (default)	default	Edit																																																		
atlassian:	NO (default)	default	Edit																																																		
BM:	NO (default)	default	Edit																																																		
CLOV:	NO	atlassian-dev atlassian-staff KLDevToolsSupport	Edit																																																		
contrib:	NO (default)	default	Edit																																																		
Critter:	NO (default)	default	Edit																																																		
FE:	NO	atlassian-dev atlassian-staff crucible-users-non-crowd KLDevToolsSupport	Edit																																																		

Creating a user

There are two types of user accounts:

- 'Built-in' user accounts — these are stored in the local database for FishEye/Crucible.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if configured. See [Configuring External Authentication Sources](#).



Note regarding external directories:

- New users can only be added if they already exist in the external directory. Your external directory will not be modified.
- If you have enabled 'auto-add' for your [external directory](#), users who don't exist locally in FishEye/Crucible will be automatically added the first time they log in to Crucible.

To add a new user:

- In the Admin area, click **Users** under 'User Settings'.
- Click **Add user** at the bottom of the screen.
- Configure the following settings:

Username	Type the user's login name. You can use the following characters: • letters and numbers • hyphen ('-') • underscore ('_') • 'at' sign ('@')
Display name	The user's display name appears in the user interface.
Email	The email address is where the user will receive notifications .
Auth Type	Select either built-in or the appropriate external directory where the user identity will be stored.
Password	(For built-in users only) ⓘ The user can easily change their own password later.

4. Click **Add**.

Add New User ⓘ

Information about the new user

Username

Display name

Email

Auth Type ⓘ

Password

Confirm Password

Creating a group

There are two types of FishEye user groups:

- 'Built-in' groups — these are stored in the FishEye database.
- 'External' groups — these are stored in an external directory (e.g. Crowd), if any are configured. See [Configuring External Authentication Sources](#).

ⓘ Note that 'external' groups can only be created in your external directory.

To add a 'built-in' group:

1. Click **Groups**, under 'Security Settings', in the Admin area.
2. At the bottom of the screen, type the name of your new group into **Group name** and click **Add**.
3. Your new group will now appear in the list of groups.
 - ⓘ To add users to your new group, see [Adding a user to a group](#).

Adding a user to a group

There are two types of FishEye user groups:

- 'Built-in' groups — these are stored in the FishEye database.
- 'External' groups — these are stored in an external directory (e.g. Crowd), if any are configured. See [Configuring External Authentication Sources](#).

i Note that for external directories and external groups, users can only be added to external groups from within the directory, so the method described below will not apply. However if you want to add external users to internal (built in) groups, then you can using the below.

To add a user to a 'built-in' group:

1. In the Admin area, click **Users** under 'User Settings'.
2. Locate the user and click their username link, under the 'User' column.
i If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click **Filter**.
3. Click **Edit Groups**, under 'Group Details'. See [Editing a User's Details](#).
4. Select group names, and use the **Join** and **Leave** buttons, to add groups to, or remove groups from, the 'Groups' column.
5. Click **Back to username** to save the changes.

Screenshot: Edit User Groups

Edit User Groups: matt

Available Groups	Groups
Join >> team-2	<< Leave team-1 team-3

[Back to matt](#)

Associating a group with a repository

Associating a group with a repository means that only members of that group will be able to access the repository. For more information, please see [Understanding authentication](#).

To associate a group with a repository:

1. Log in to FishEye as an administrator.
2. Click on **Repositories** (under 'Repository Settings') in the left-hand navigation panel.
3. Click on the name of a repository.
4. Click on **Permissions** in the left-hand navigation panel.
5. Select the groups you wish to associate with the repository.
6. Click **Save**.

Repository "CLOV"

Summary
Maintenance
SCM Details
Updates
Linkers
Permissions
Include/Exclude Paths
Hidden Directories
Tarballs

Access by Built-in Groups

☐ Use repository defaults. [Configure defaults.](#)

	Can Read
Anonymous	☐
All logged-in users	☐
KLDevToolsSupport	☒
beac-users	☐

Configuring avatar settings

Every user with a FishEye/Crucible account will have an avatar image displayed next to their name throughout the application. By default, each user will have a plain grey avatar image. Note, you cannot disable avatars.

You can configure whether to use FishEye/Crucible's own avatars or avatars from an external server (e.g. <http://www.gravatar.com>).

In both cases, users can manually upload their own images to use as avatars.

To configure avatar settings:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click **Avatars** under the 'User Settings' section.
3. Choose the desired avatar configuration for your FishEye/Crucible instance:

Internal	Use FishEye/Crucible's default avatars, "Charlitors" (Semi-unique avatars resembling charlie) or Joe Doe. We recommend this option if you are running FishEye/Crucible behind a firewall. Users will also be able to upload their own images to use as avatars via their user profile, (see Changing your User Profile). User-defined avatars will override default avatars.
External	Use avatars hosted by an external server (e.g. http://www.gravatar.com). Users will also be able to upload their own images to use as avatars via their user profile. See Changing your User Profile . User-defined avatars will override avatars from an external server.
Url	The address of the external avatar server e.g. <code>http://example.com/avatar/</code> Use the same protocol that is used to access FishEye. That is, if FishEye uses the secure HTTP protocol then you need to use 'https' when specifying your external server. If you are using the Gravatar service, we recommend that you use the following URL: <code>https://secure.gravatar.com/avatar/</code>

Suffix	(<i>optional</i>) Specify any parameters supported by the external server. For example, if you wish to control the type of images to be displayed, Gravatar accepts "r=g" ("rating=general"). The suffix must be in the form of a URL query string, starting with an ampersand, e.g. "&r=g"

Note that if you want to use an external avatar server and are using a web server in front of FishEye, you will need to set your [site URL and proxy host settings](#).

4. Click **Save changes**.

Avatars


☐ Internal
Use FishEye and Crucible's internal avatar server.


☒ External
Choose an avatar server that FishEye and Crucible can use to get custom user avatars.

Url:
Enter the URL of the external avatar server that uses the Gravatar URL format. E.g. http://gravatar.com/avatar/

Suffix:
Enter any query string suffix supported by the external server. E.g. Gravatar accepts "r=g" ("rating=general").

or

Configuring anonymous access

Anonymous access to FishEye is allowed by default. You can disable anonymous access at the following levels:

- Globally by turning off 'Global Anonymous Access' on the Administration, Authentication page. When this setting is off, an anonymous user just sees a log in page. When it is on they see an activity stream of all repositories which are visible to anonymous users.
- For all repositories which are using the default Security settings (using the [Repository defaults](#)),
- or for particular repositories.

Projects and Reviews

If you are using FishEye with Crucible, anonymous access is also subject to individual project's permissions (see [Creating a permission scheme](#)).

To configure anonymous access globally:

1. Click the user menu in the FishEye/Crucible header, and choose **Administration**. You will need to be logged in as an [administrator](#) to see this link.
2. Click **Defaults** under 'Repository Settings'. The 'Repository Defaults' screen will be displayed.
3. Click the **Permissions** tab.
4. Either check or clear the option *Can Read* in the Anonymous row.

To configure anonymous access per-repository:

See the steps detailed in [Configuring Permissions for a Specific Repository](#).

Editing a user's details

There are two types of user accounts:

- 'Built-in' user accounts — these are stored in the local database for FishEye/Crucible.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if configured. See [Configuring External Authentication Sources](#).

To edit a user's details:

1. In the Admin area, click **Users** under 'User Settings'.
2. Locate the user and click the corresponding **Edit** link.
 If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click **Filter**.
3. You can update the following fields:

Display name	To change the user's login name, see Renaming a user .
Email	This is the address to which notifications are sent.
Auth Type	Select either built-in or the name of the appropriate external directory where the user is stored. By changing the ' Auth Type ', you move the user to a different directory.
Active Crucible user	Check to allow the user the have access to Crucible functionality.

4. Click **Apply**.

Changing a user's password

There are two types of user accounts:

- 'Built-in' user accounts — these are stored in the local database for FishEye/Crucible.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if configured. See [Configuring External Authentication Sources](#).

 You can only change the passwords of 'built-in' users.

To change a user's password:

1. In the Admin area, click **Users** under 'User Settings'.
2. Locate the user and click their username.
 If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click **Filter**.
3. Click **Change Password**.
4. Enter, and confirm, the new password.
5. Click **Apply**.

Renaming a user

There are two types of user accounts:

- 'Built-in' user accounts — these are stored in the local database for FishEye/Crucible.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if configured. See [Configuring External Authentication Sources](#).

 Renaming a user changes their login name. To change their Display Name, please see [Editing a User's Details](#).

To rename a user:

1. In the Admin area, click **Users** under 'User Settings'.
2. Locate the user and click their username.
 If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click **Filter**.
3. Click **Rename**.
4. Enter the new username
5. Click **Rename**.

Deleting or deactivating a user

To revoke a user's access to FishEye, you need to delete the user's account.

To revoke a user's access to Crucible, you can either:

- delete the user's account; or
- deactivate the user's account (this has the advantage of being easy to undo if required).

Note that the number of users granted by your FishEye license may be different from the number of users granted by your Crucible license.

To deactivate a user's Crucible account:

1. Click **Users** on the 'Admin Menu'.
2. Locate the user and click the corresponding **'Edit'** link.
 If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click the **'Filter'** button.
3. The 'Edit User' screen will be displayed. Deselect the **Active Crucible user** check-box.
4. Click **Apply**.

To delete a user's FishEye (and Crucible) account:

1. Click **Users** on the 'Admin Menu'.
2. The 'User Browser' screen will be displayed (see screenshot below), showing a list of users.
3. Locate the user and click the corresponding **Delete** link.
 If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click **Filter**.

Screenshot: Authentication Settings

User browser

The User Browser allows you to browse all the users in the system. Filters allow you to limit the users that you see.
Displaying users 1 to 6 of 6. ([Reset filter](#))

Email contains:
In Group: Any

User	Display name	Email	Auth	Action
brendan	Brendan Humphreys	brendan@cenqua.com	built-in	Edit Delete Logout
conor	Conor MacNeill	conor@cenqua.com	built-in	Edit Delete Logout
matt	Matt Quail	matt@cenqua.com	built-in	Edit Delete Logout
nick	Nick Pellow	nick@cenqua.com	built-in	Edit Delete Logout
pete	Peter Moore	pete@cenqua.com	built-in	Edit Delete Logout
pmcneil	Peter McNeil	pmcneil@cenqua.com	built-in	Edit Delete Logout

[Add User](#)

Configuring External Authentication Sources

Although FishEye always maintains a list of users internally, you can have FishEye authenticate and authorise users against one or more external authentication sources.

Be aware that you can [force usernames to become lowercase](#) on import.

Single External Authentication Source

To set an external authentication source, click '**Authentication**' on the Admin menu. You can only set up one external authentication source and each repository has the choice of authenticating against that or the internal repository. To change the external authentication source, you will need to remove the settings that are already configured, by clicking the '**Remove**' link. You will then see the option to add a different authentication source.

FishEye currently supports:

- [LDAP Authentication](#)
- [Host-Based Authentication](#) – Using PAM on Linux/Solaris/OS-X and Local/Domain Accounts on Windows
- [AJPv13 Authentication](#)
- [Custom authentication](#)
- [JIRA Authentication](#) – Using Atlassian's JIRA to manage your users and groups

Multiple External Authentication Sources

The recommended approach to authenticating against more than one authentication source is to implement single signon (SSO). You can integrate FishEye with an existing SSO solution, or add [Atlassian Crowd](#) integration to combine users and authentication from multiple external user repositories. Crowd connectors are already available for all Atlassian products, LDAP, Active Directory and Subversion:

- [Crowd Authentication](#)
- [Custom authentication](#)

Loading Users from Active Directory, LDAP, Crowd or JIRA

FishEye and Crucible can load an external user base stored in Active Directory, Atlassian Crowd, Atlassian JIRA, or any LDAP-compatible user repository.

1. Configuring your External Source

You can only have one external user repository connected to FishEye/Crucible. If you need multiple repositories,

you can use Atlassian Crowd to collate users from multiple sources and then connect FishEye/Crucible to Atlassian Crowd.

You must first configure FishEye/Crucible to connect to your external user repository, which may be LDAP-based (including Active Directory), Atlassian Crowd or Atlassian JIRA. Follow the steps in the documentation pages listed below and then return to step 2 on this page.

- [LDAP Authentication](#)
- [JIRA and Crowd Authentication](#)

2. Loading External Users

To load users from Atlassian Crowd or Atlassian JIRA:

1. Click '**Authentication**' on the admin menu.
2. Select one or more **groups to synchronise**. For JIRA, the default is `jira-users`.
3. Click '**Apply changes**'.
4. Select '**Yes**' next to the option '**Synchronise users with JIRA/Crowd**'.
5. Click '**Apply**' to save the changes.

As soon as you click '**Apply**', users are immediately added to the user list.

Notes

- Users with identical names to existing users will not be imported.
- In the users list, you can check whether each user is from the local database or loaded externally.
- The '**synchronise period**' option allows you to set the time interval when Fisheye or Crucible will synchronise with the external directory. You can use values such as '**1m**' for one minute, '**1h**' for one hour, and so on.
- If you have a large user repository, then please read this document on best practices for [Synchronising With Large User Repositories](#).

Synchronising With Large User Repositories

When FishEye synchronises with external user repositories (LDAP, AD or Crowd) it adds all visible users in the repository, even if you do not want all these users to have access to FishEye.

For example you may have restricted [viewing permissions to your repositories](#) to a single LDAP group (using a filter). However the synchronise button will import all your LDAP users, because each of these users will be able to log into fisheye, even though they will not have permission to view any repository.

This is unavoidable, as users may create another repository in the future or a repository with no group based restrictions.

Also if you have a large number of users in your LDAP, then a synchronise needs to check if for each user within Fisheye, to see if they still exist in LDAP and for each user in LDAP to see if they exist in Fisheye. This can be quite taxing when you have a large user base. It will also require more memory and could lead to

OutOfMemoryError: Java Heap Space errors.

Additionally, when setting up external authentication for large repositories: set "[Synchronise users with LDAP](#)" to no. Next turn on "Auto-add" users to on, so that any External user that matches your defined user filter who logs into FishEye, will be added as a user (NOTE: Auto-add does not mean any user will have access to FishEye, only users that can successfully authenticate against your LDAP server, using the settings you have defined within FishEye, will be added. They first must pass the authentication before they will be added).

You should always try to restrict your [user filter](#) when setting up your LDAP authentication to only match the small subset of users that should have access to Fisheye.

LDAP Authentication

This page explains the settings for LDAP authentication and their parameters.

On this page:

- [Global Settings](#)
- [Per-Repository Settings](#)
 - [Examples:](#)
- [Active Directory](#)

Global Settings

Global LDAP settings are:

URL	The URL of the LDAP server, e.g. <code>ldap://localhost:389</code> .
Base DN	The base search space for users, e.g. <code>dc=example, dc=com</code>
User Filter	The LDAP search for locating users, e.g. <code>uid=\${USERNAME}</code> . The <code>\${USERNAME}</code> variable is expanded to the username of the individual being authenticated. You can use a more complicated LDAP filter to allow only a subset of users, such as: <code>(&(uid=\${USERNAME})(group=fisheye))</code> .
UID Attribute	The name of the username attribute in objects matching the filter.
Email attribute	Optional. The name of an attribute giving the user's email address.
Cache TTL (positive)	How long FishEye should cache permission checks. Example values are: 0 secs, 5 mins.
Auto-add	FishEye can automatically create a user it has not previously encountered if the user can successfully authenticate against LDAP.
Initial bind DN and password	Optional. If your LDAP server does not allow anonymous bind, then you need to specify a user FishEye can use to do its initial bind.
Synchronise users with Crowd	Optional. Sets whether users will be loaded from an external directory.

Per-Repository Settings

You can give FishEye an LDAP filter that will be used to check if a user has access to individual repositories.

To specify this filter for a repository, open the [Permission settings](#) of the repository, and check the *Apply LDAP restriction* option. Then just below the checkbox, type the filter string:

LDAP restriction	An LDAP filter used to check if a given user can access a given repository, e.g. <code>(&(uid=\${USERNAME})(group=\${REP}))</code> . When an user attempts to access the repository, FishEye will perform a search in LDAP to retrieve all the entities matching the LDAP filter. This search will start at the <i>Base DN</i> of your LDAP's global settings .
Match Type	One of 'user' (default) or 'any'. This setting modifies how the search results are interpreted. • If set to 'user', then the first search result must match the exact Distinguished Name (DN) of the current user. Only if it does match, then that user is allowed to access to the repository. • If set to 'any', then if there is one or more search results, the current user is allowed to access to the repository.

 When writing your LDAP filter, the following expressions are available:

- `${USERNAME}` will be replaced by the username of the logged user
- `${DN}` will be replaced by the [Distinguished Name](#) of the logged user
- `${REP}` will be replaced by the name of the repository the logged user is trying to access

Examples:

- To allow a user to access a repository only if he or she is part of the group *AuthorizedGroup*, specify an LDAP filter such as: `(&(uniqueMember=${USERNAME})(cn=AuthorizedGroup)(objectClass=groupofuniquenames))` and set the Match Type as *Any*.
- To allow more than a group of users to access a repository, specify an LDAP filter such as: `(&(uniqueMember=${USERNAME})(|(cn=AuthorizedGroup1)(cn=AuthorizedGroup2))(objectClass=groupofuniquenames))` and set the Match Type as *Any*.

Testing the LDAP restriction

To test the LDAP restriction, you can use [Apache Directory Studio](#):

- add a connection to your LDAP server (if you use Active Directory, use the following [instructions](#))
 - in the *LDAP* menu, click on *New Connection*
 - enter the same URL and port as you did for FishEye
 - click on *Next* and enter the same Bind DN and password as in [Global settings](#) (see *Initial bind DN and password* above)
 - click on *Finish*
- in the LDAP menu, and click on *New Search*
- in the Search dialog,
 - select the connection to your LDAP server
 - enter the same *Base DN* you configured in [Global settings](#) next to the *Search Base* textbox
 - enter the *LDAP filter* next to the *Filter* textbox (if the filter contains `${USERNAME}` or `${REP}`, replace them by the username or repository name you want to test)
 - select Subtree in the *Scope* section
 - click on *Search*

Active Directory

To have FishEye connect to an Active Directory server, use settings such as the following:

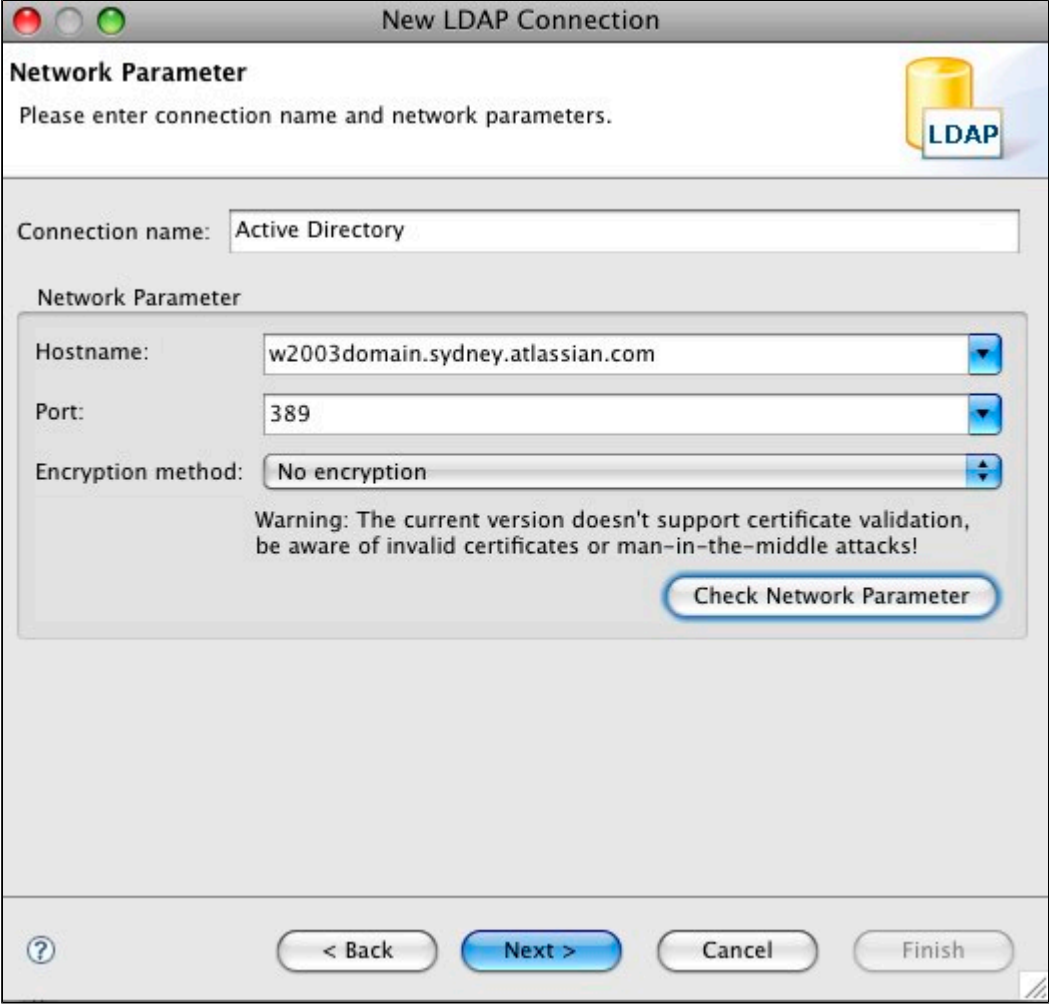
URL	ldap://HOSTNAME:389
Base DN	DC=corp, DC=example, DC=com
User Filter	(sAMAccountName=\${USERNAME})
UID Attribute	sAMAccountName
Email attribute	mail
Initial bind DN	corp.example.com/Users/SomeUser

How to connect FishEye to Active Directory or another LDAP server

-  These instructions target Active Directory, but you should be able to use the same process for other LDAP servers.

You use Active Directory (AD) for managing your users and you would like to connect FishEye to AD to authenticate users. For many administrators this can be confusing because AD tries to hide the complexities of LDAP from the user.

1. Download and install [Apache Directory Studio](#).
2. Once installed Apache Directory Studio go into File -> New -> LDAP Connection and follow the wizard. The details you enter in step 2 will be the initial bind user / password for FishEye.



The image shows a 'New LDAP Connection' dialog box with a title bar containing standard window controls. The main area is titled 'Network Parameter' and includes a sub-header 'Please enter connection name and network parameters.' with an LDAP icon. A text field for 'Connection name:' contains 'Active Directory'. Below this is a 'Network Parameter' section with three fields: 'Hostname:' with 'w2003domain.sydney.atlassian.com', 'Port:' with '389', and 'Encryption method:' with 'No encryption'. A warning message is displayed below these fields, and a 'Check Network Parameter' button is to its right. The bottom of the dialog features a help icon, a '< Back' button, a highlighted 'Next >' button, a 'Cancel' button, and a 'Finish' button.

New LDAP Connection

Network Parameter
Please enter connection name and network parameters.

Connection name: Active Directory

Network Parameter

Hostname: w2003domain.sydney.atlassian.com

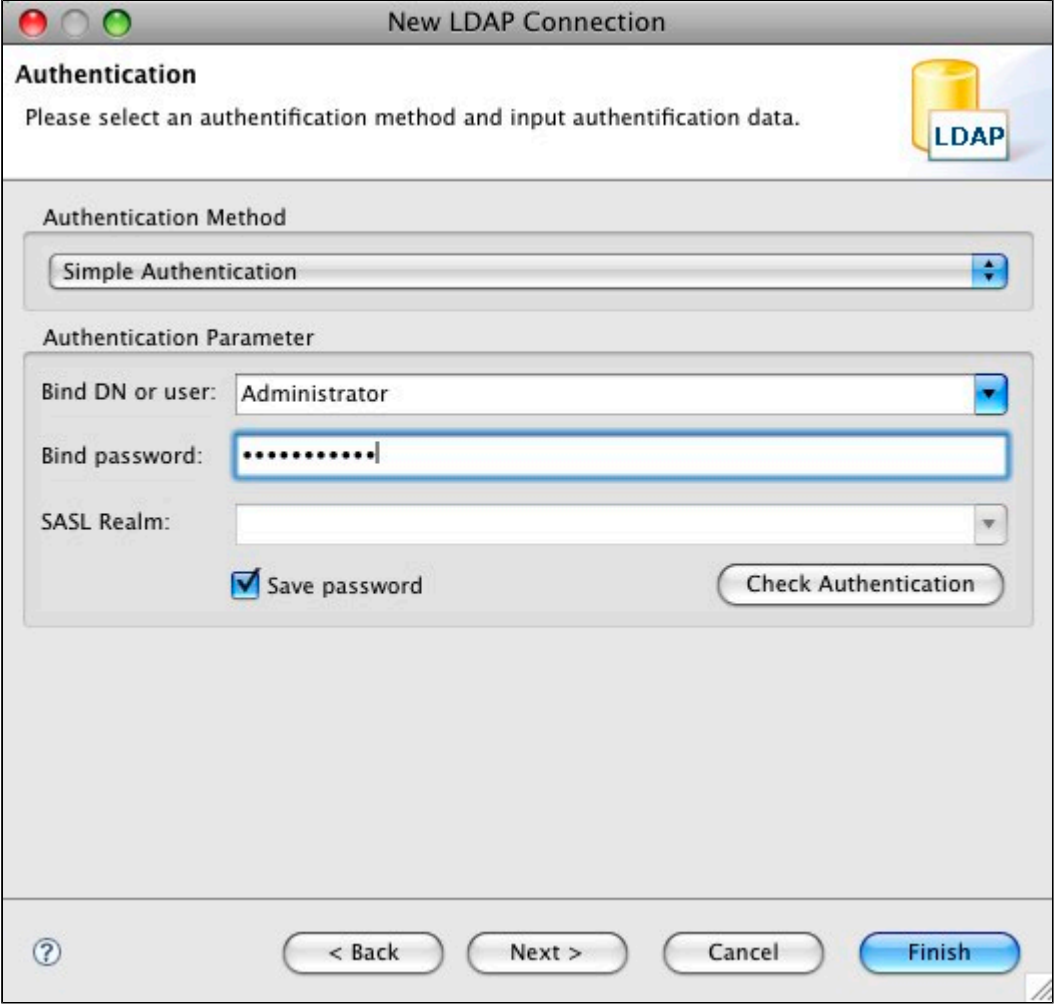
Port: 389

Encryption method: No encryption

Warning: The current version doesn't support certificate validation, be aware of invalid certificates or man-in-the-middle attacks!

Check Network Parameter

? < Back Next > Cancel Finish



The image shows a 'New LDAP Connection' dialog box with a title bar containing standard macOS window controls. The main area is titled 'Authentication' and includes a sub-header 'Please select an authentication method and input authentication data.' and an LDAP icon. Below this, the 'Authentication Method' is set to 'Simple Authentication'. The 'Authentication Parameter' section contains three fields: 'Bind DN or user' with the value 'Administrator', 'Bind password' with masked characters, and 'SASL Realm' which is empty. A 'Save password' checkbox is checked, and a 'Check Authentication' button is present. At the bottom, there is a help icon, and navigation buttons for '< Back', 'Next >', 'Cancel', and 'Finish'.

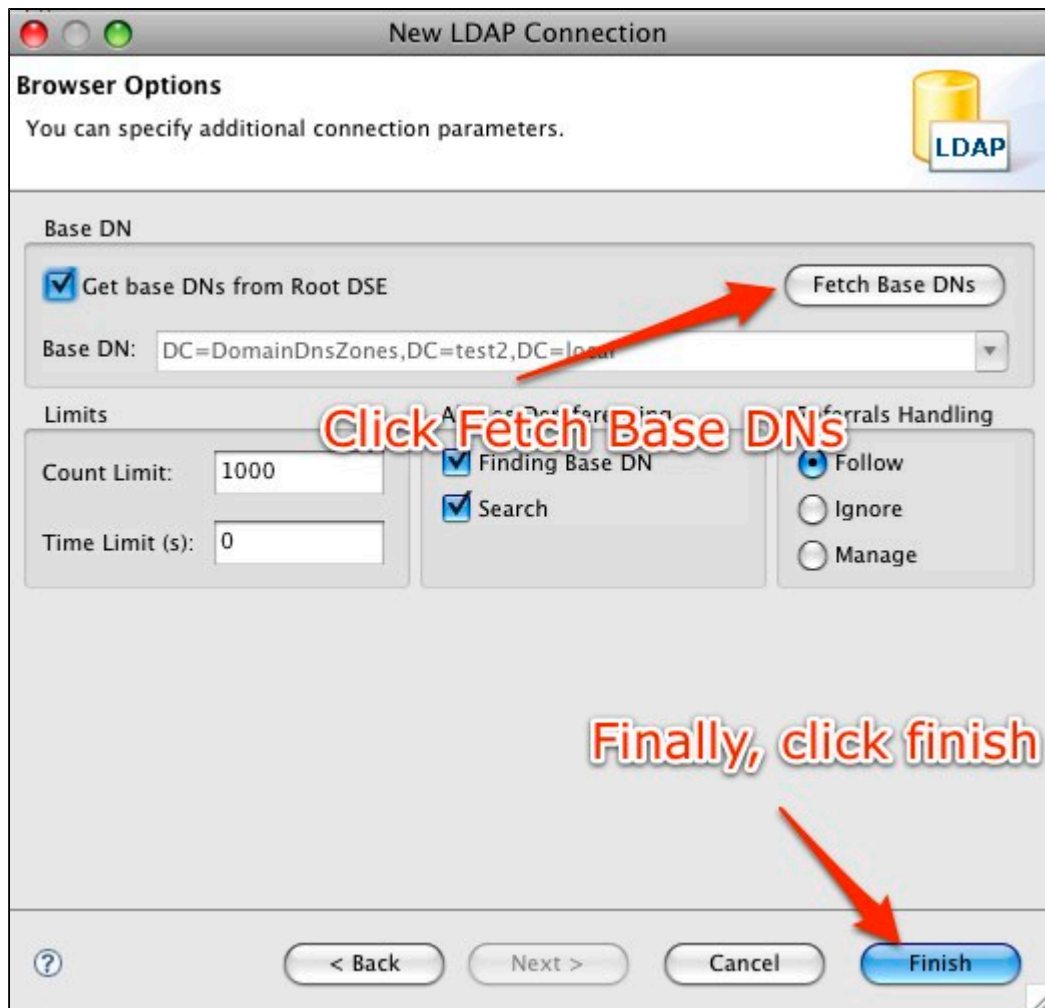
New LDAP Connection

Authentication
Please select an authentication method and input authentication data.

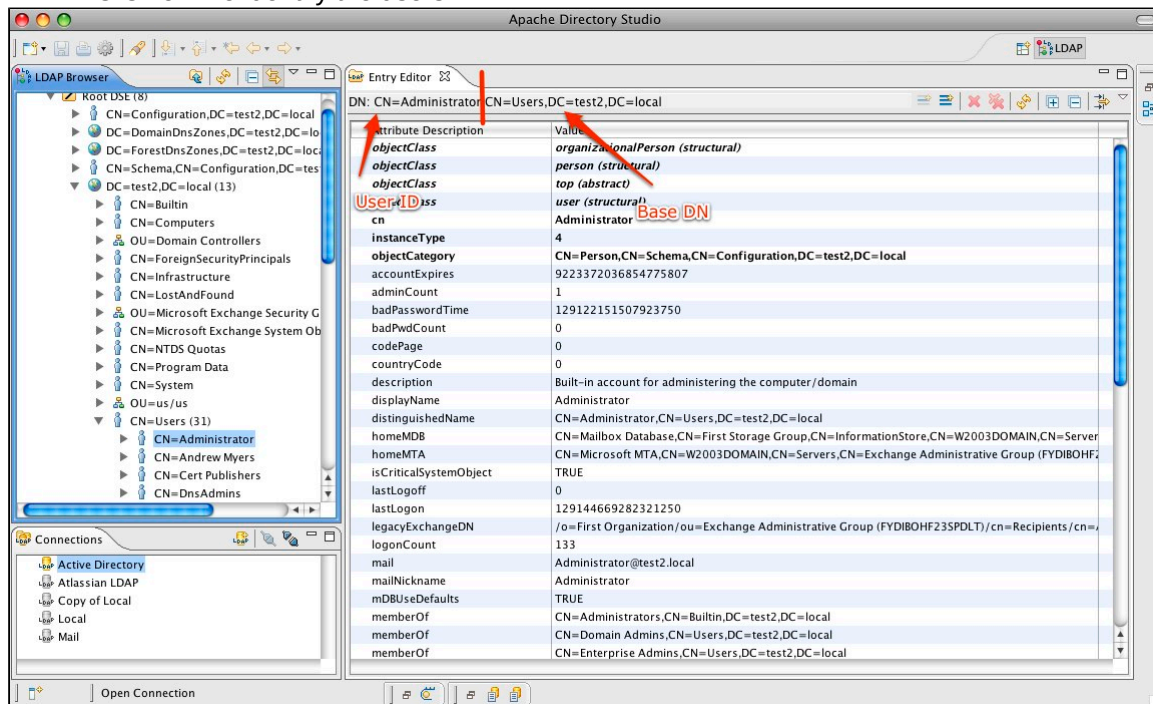
Authentication Method
Simple Authentication

Authentication Parameter
Bind DN or user: Administrator
Bind password:
SASL Realm:
☒ Save password Check Authentication

? < Back Next > Cancel Finish



3. Now we have a connection to AD. Browse to the location where your users are stored. In my case its under CN=Users, DC=test2, DC=local. Here I select a user Administrator. At the very top we see the DN. This is how we identify the users.



In this instance the DN is CN=Administrator, CN=Users, DC=test2, DC=local, so the Base DN for FishEye would be CN=Users, DC=test2, DC=local. CN=Administrator identifies the user, so the user filter would be (CN=\${USERNAME}).

i For Active Directory: Windows logins use sAMAccountName as the login username, so it is more correct to use (sAMAccountName=\${USERNAME}) as the User ID filter if you want to match the FishEye/Crucible logins to Windows logins.

4. For the UID attribute, email attribute and the display name attribute, I can get that information from the right hand pane above. In my case UID attribute is cn, the email attribute is mail and the display name attribute is displayName.
5. With all that information we are now ready to connect FishEye to AD. To summarise we have:

LDAP URL	ldap://w2003domain.sydney.atlassian.com:389
Base DN	CN=Users,DC=test2,DC=local
User Filter	(CN=\${USERNAME})
UID Attribute	cn
Email Attribute	mail
Display Name Attribute	displayName
Initial Bind user	Administrator
Initial bind password	password

6. Placing that information into FishEye:

LDAP Authentication settings 

URL: ldap://w2003domain.sydney.atlassian.com:389

Base DN: CN=Users,DC=test2,DC=local

User Filter: (CN=\${USERNAME})

UID attribute: cn

Email attribute: mail

Display name attribute: displayName

Cache TTL (positive): 5 minutes

Auto-add users into FishEye: true

Initial bind user: Administrator

Synchronise Period: false

[Re-Sync *](#) | [Edit](#) | [Test](#) | [Remove](#)

* Synchronising with a large user base can lead to [Performance Issues](#)

Useful References

- [LDAP Authentication](#)
- http://www.computerperformance.co.uk/Logon/LDAP_attributes_active_directory.htm

JIRA and Crowd Authentication

Integrating FishEye with JIRA for User Management

You can manage your FishEye/Crucible users in JIRA.

This option is available only with JIRA 4.3 and later.

If you want to manage your FishEye users using JIRA, you need to configure a connection in JIRA and then set up the user directory in FishEye. Both steps are described below.

2.1. To configure a new connection for user management in JIRA:

1. Log in to JIRA as a user with the 'JIRA System Administrators' global permission.
2. Click **Administration** on the JIRA top navigation bar.
 - a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:
 - In JIRA 4.3: Click '**Other Applications**' in the '**Users, Groups & Roles**' section of the JIRA administration screen.
 - In JIRA 4.4: Select '**Administration**' > '**Users**' > '**JIRA User Server**'.
 - b. **Add** an application.
 - c. Enter the **application name** and **password** that your application will use when accessing JIRA.
 - d. Enter the **IP address** or addresses of your application. Valid values are:
 - A full IP address, e.g. 192 . 168 . 10 . 12.
 - A wildcard IP range, using CIDR notation, e.g. 192 . 168 . 10 . 1/16. For more information, see the introduction to [CIDR notation on Wikipedia](#) and [RFC 4632](#).
 - **Save** the new application.

2.2. To set up the JIRA user directory in FishEye:

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

Authenticate against	Select a JIRA instance
Application name and password	Enter the values that you defined for your application in the settings on JIRA.
JIRA URL	The web address of your JIRA server. Examples: <pre>http://www.example.com:8080/jira/ http://jira.example.com</pre>
Auto-add	Select Create a FishEye user on successful login so that your JIRA users will be automatically added as a FishEye user when they first log in.
Periodically synchronise users with JIRA	Select Yes to ensure that JIRA will synchronize all changes in the user information on a regular basis. Change the value for Synchronise Period if required.
When Synchronisation Happens	Select an option depending on whether you want to allow changes to user attributes from within FishEye.

Single Sign On

Select **Disabled**. SSO is not available when using JIRA for user management and if enabled will make the integration fail.

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

For further information, see the JIRA documentation here: [Allowing Other Applications to Connect to JIRA for User Management](#)

Integrating FishEye with Crowd

Atlassian's [Crowd user management application](#) can be integrated with FishEye.

You set the basic Crowd properties, such as the application name, password and URL, using the FishEye administration screens, as described in [Integrating Crowd with FishEye](#) in the *Crowd Administrator's Guide*.

You can also fine tune your Crowd integration by overriding the default Crowd properties, such as the session validation interval and SSO cookie name, by manually editing the `config.xml` file in your FishEye installation directory. See [Integrating Crowd with FishEye](#) for details.

Custom authentication

To implement an arbitrary form of authentication and authorisation for FishEye you need to provide a class which extends `com.cenqua.fisheye.user.plugin.AbstractFishEyeAuthenticator`. You can find more information about custom FishEye authorisation in the [online javadocs](#) and the [library jar](#).

For FishEye to use the authenticator, it must be compiled, placed in a jar archive and then put in the `$FISHEYE_INST/lib` directory. If other third-party libraries are required by your authenticator, they must also be in the `$FISHEYE_INST/lib` directory.

Global configuration

After implementing a custom authenticator, the next step is to configure FishEye to use it.

Click **Setup Custom authentication** on the FishEye **Security** page, in the Admin area.

You will be presented with a form containing the following fields to be set:

Classname	The fully qualified class name of your <code>AbstractFishEyeAuthenticator</code> , e.g. <code>com.cenqua.fisheye.user.plugin.ExampleFishEyeAuthenticator</code> .
Cache TTL (positive)	How long FishEye should cache permission checks. Example values are: 0 secs, 5 mins.
Auto-add	FishEye can automatically create a user it has not previously encountered if the user can successfully authenticate against your authenticator.

Properties	Any properties your authenticator requires. These will be passed to its <code>init()</code> method. This field should comply with the <code>java.util.Properties</code> format. Example: <pre># comments name1=value1 name2=value2</pre>

Per-repository constraint configuration

You may also require a per-repository constraint to restrict access to specific repositories using your custom authenticator. If a custom authenticator is set, then the [Permissions Summary table](#) will display the constraint per repository and a link to enable you to edit it.

When using FishEye's built-in groups in conjunction with a custom authenticator, a user will have access to a repository if:

1. the user is in an allowed group for the repository
2. or the `hasPermissionToAccess` method of the custom authenticator returns true.

 The 'Authentication Test' page allows you to enter a user's credentials and to test the user's authentication. It will also test which repositories the user is authorised to access.

Host-Based Authentication

Host-based authentication uses the user account mechanism of the underlying operating system on which FishEye is running.

 FishEye currently supports PAM-based authentication on **32-bit** Linux/Solaris/OS-X, and NT-based authentication on Windows. An outstanding feature request for 64-bit support is located at [FE-683](#)

Group Restrictions

FishEye can be configured to check if a user belongs to a group (or groups) before allowing access.

You can list one group name, or join several group names into a boolean expression like `group1 & (group2 | group3)`.

If your group name contains spaces or non-ASCII characters, then you need to use quotes. For example: "Power Users" | Administrators.

Windows

 If you are using Active Directory, you can configure FishEye to [use LDAP](#) as an alternative to host-based authentication.

If the computer FishEye is running on is **not** a member of a domain, then the Domain attribute is ignored.

When the computer is a member of a domain, you need to enter the full DNS name of the domain (e.g. `corp.example.com`). If you enter the short version of the domain (e.g. `corp`), then group-based restrictions may fail.

Once you have configured your settings, we recommend you use the '**Test**' function to ensure your access control behaves correctly.

PAM

On Linux, Solaris and OS-X, host-based authentication uses PAM (Pluggable Authentication Modules) to check users' passwords.

FishEye needs to be configured with the *service name* to use when conversing with PAM. You can create a new service name in the PAM configuration (typically `/etc/pam.conf` or `/etc/pam.d/`), or configure FishEye to use an existing service name (such as `other`, `login` or `xscreensaver`).

Some general operating-system specific tips are given below, but you should consult the PAM documentation for your operating system.

Once you have configured your settings, we recommend you use the '**Test**' function to ensure your access control behaves correctly.

Linux

On many Linux distributions, you may need to create a `/etc/pam.d/fisheye` file containing:

```
auth        required    pam_stack.so service=system-auth
```

Mac OS-X

On a default OS-X installation, you may need to create a `/etc/pam.d/fisheye` file containing:

```
auth        sufficient   pam_securityserver.so
auth        required     pam_deny.so
```

Solaris

If you are using the default `pam_unix_auth` PAM configuration on Solaris, then you may need to add a line like this to your `/etc/pam.conf` file:

```
fisheye auth requisite    pam_authtok_get.so.1
fisheye auth required     pam_unix_auth.so.1
```

If you test this and it does not work, it is probably because when using `pam_unix_auth` on Solaris, the process doing the password check needs read access to `/etc/shadow`.

Giving the FishEye process read access to this file may solve this problem, but using permissions other than `0400` for `/etc/shadow` is not recommended. You should discuss this with your system administrators first, and possibly change to a PAM module other than `pam_unix_auth`.

Global Settings

Global settings are:

Domain/Service name	Windows: the name of the domain. Leave blank to use the local computer. PAM: The service name in your PAM configuration to use. If blank, <code>fisheye</code> is used.
Required group:	The group or groups a user must belong to in order for them to be able to log in.
Cache TTL (positive)	How long FishEye should cache permission checks. Example values are: 0 secs, 5 mins.
Auto-add	FishEye can automatically create a user it has not previously encountered if the user can successfully authenticate with the host.

Per-Repository Settings

You can give FishEye a group restriction that will be used to check if a user has access to individual repositories. You can specify this per repository, or just specify it in the repository defaults:

Required Group	A group (or groups) used to check if a given user can access a given repository. For example: <code>cvsusers & cvs\${REP}</code> The <code>\${REP}</code> variable is replaced with the name of the repository in question.

AJPv13 Authentication

AJP authentication expects requests to be pre-authenticated via an external server before arriving at FishEye.

Typically, this would be a web server (e.g. Apache) configured to perform password and role checking for a given URL. If successful, the server forwards the request to the FishEye server via the AJPv13 protocol.

FishEye Configuration

For FishEye to use AJP authentication, the following two values must be configured:

- The AJP Bind Address must be set per FishEye instance. See also [Server Settings](#).
- The user's Auth Type must be set to 'ajp'.

Apache Configuration

mod_proxy_ajp

To configure Apache Httpd to forward all requests to `/fisheye` to a FishEye instance with an AJP Bind address of `localhost:8009`, on the same machine, ensure that the `proxy_ajp_module` apache module is enabled, and add these lines to the `VirtualHost` you wish to serve FishEye through:

```
ProxyPass /fisheye/ ajp://localhost:8009/fisheye/ nocanon
```

mod_jk

To configure Apache Httpd so that all requests to Apache Httpd for the path `/fisheye` are forwarded to a

FishEye instance on the same machine with an AJP Bind Address of `localhost:8009`.

Add these lines to your apache configuration:

```
LoadModule jk_module modules/mod_jk.so

JkWorkersFile /path/to/workers.properties
JkLogFile /var/log/mod_jk.log
JkLogLevel debug
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
JkMount /fisheye/* worker1
```

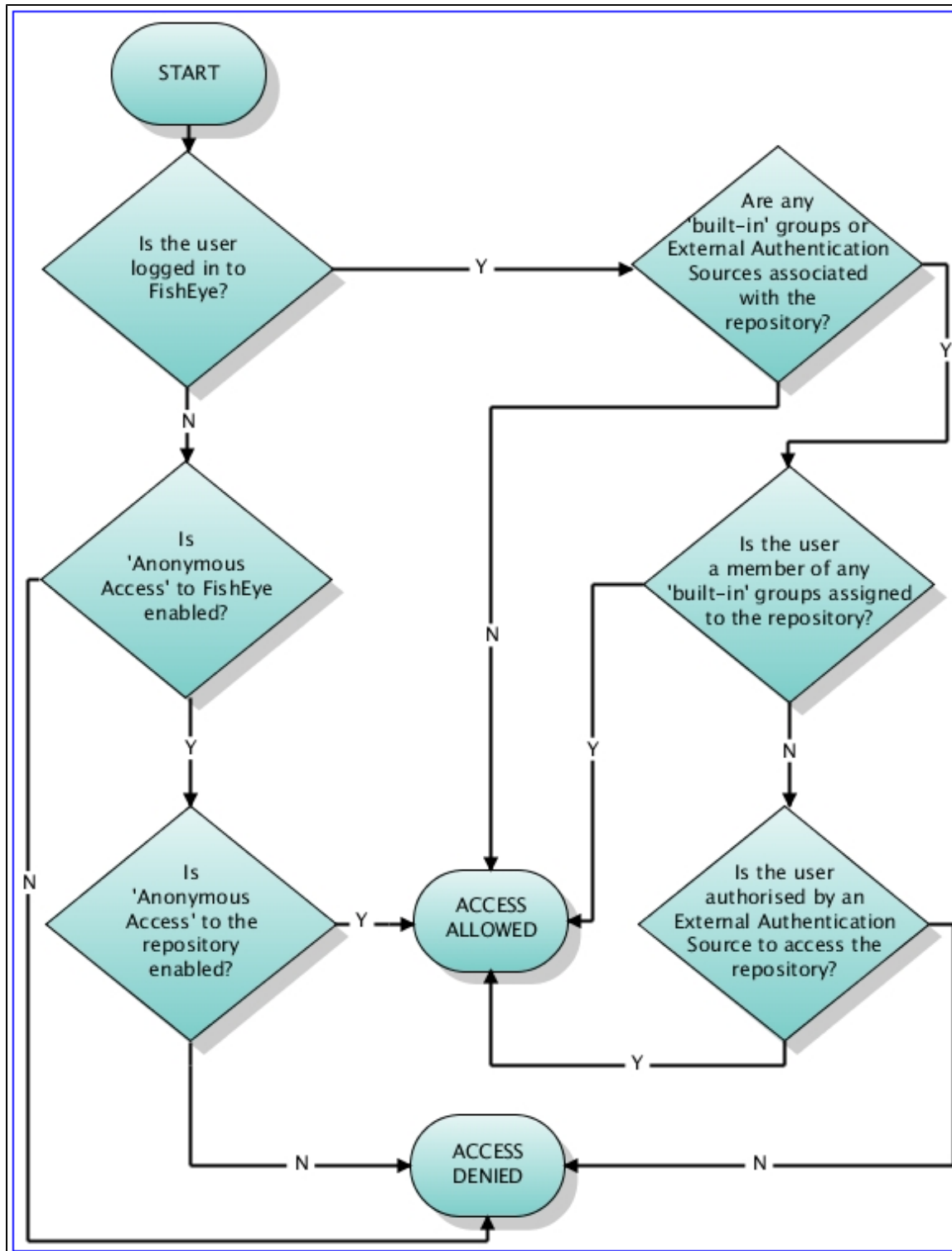
Then create a file under `/path/to/workers.properties` and add:

```
worker.list=worker1
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
```

Understanding authentication

The following flowchart shows how to determine whether a user is allowed to access a FishEye repository:

Screenshot: FishEye Authentication Flowchart



Forcing Imported Usernames to be Lowercase

When importing users from external authentication sources, you can set FishEye to force the usernames to become lowercase. This solves an issue with some sources adding duplicate users to the FishEye database.

To force lowercase usernames, carry out the following steps:

1. Log into FishEye's Admin Interface.
2. Under **Authentication settings**, the option **Force Lowercase Username** can be toggled on and off.
3. With this setting switched **On**, when new users are added to FishEye from an external authentication source, all usernames will be converted to lowercase.

Screenshot: Forcing Lowercase Usernames

Username OptionForce Lowercase Username: OFF (On) 

Configuring SMTP

To configure SMTP settings:

1. Go to the Admin screen, click **Server** (under 'Global Settings') and look for the 'Mail Server' section.
2. Click **Edit config**.
3. You can enter the following parameters:

Send emails from	Select either Server address , or the User address , which selects the email address in the user's profile. <i>(Note: this only applies to Crucible notifications. FishEye will always use the Server Address.)</i>
From Address	The from address used when FishEye sends an email, e.g. fisheye-noreply@example.com
SMTP Host name	The host name of the SMTP server.
Enable debug	Optional. Turn this on to enable debug logging from the mail server. Useful in tracking down mail server connectivity problems.
SMTP Port	Optional. The port to connect to on the SMTP host.
Use SSL/TLS	Optional, defaults to 'False'. This turns on Secure Sockets Layer/Transport Layer Security security for mail servers that require it, or use it by default.
Username & Password	Optional. Username and password for authenticated SMTP access.

4. Click **Save**.

Once you have configured SMTP, you can click **Send test email** in the 'Mail Server' section to confirm the SMTP connectivity.

Customising email notifications

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

Editing FishEye email templates

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

There are sets of templates for both HTML and plain-text emails, as listed in the table below. Note that these templates do not support embedding full diffs into notifications. They are only for changing the appearance and

order of certain content inside the messages.

You can edit templates in any text editor.

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

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

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

Advanced editing of FishEye email templates

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

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

FishEye email template file list

The following template files for FishEye notifications are stored in the FISHEYE_HOME/template/fisheye/ folder.

	Location	Template filename	Purpose
Common	FISHEYE_HOME/template/fisheye/html/	changeset-mail-subject.ftl	Subject template
	FISHEYE_HOME/template/shared/html/	changeset-header.ftl	
	FISHEYE_HOME/template/shared/	util.ftl	
HTML	FISHEYE_HOME/template/fisheye/html/	changeset-mail.ftl	HTML email template
		fisheye-layout.ftl	
		password-reset.ftl	
		plugin-notification.ftl	
		verify-password-reset.ftl	
Plain text	FISHEYE_HOME/template/fisheye/plain/	changeset-mail.ftl	Plain text email template
		fisheye-layout.ftl	
		fisheye-support-request.ftl	
		password-reset.ftl	
		plugin-notification.ftl	

		verify-password-reset.ftl	

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

Freemarker Data Model for Email Templates

Customising FishEye email templates with Freemarker

This page lists the Freemarker data-model for FishEye email templates. See the [Freemarker documentation](#) for instructions on Freemarker syntax. Use the templates that ship with FishEye as a guide to the properties available on each object.

These templates are used to send both batch (e.g. daily) and immediate emails. The template has access to the **changesets** variable which contains the list of changesets to send.

The default FishEye email templates make use of various data model objects, listed below.

Here is a simple example that prints out each revision in each changeset.

```
[#list changesets as cs]
${cs.id}
Author: ${cs.author}
Comment: ${cs.comment}
Files:
[#list cs.revisionInfos as rev]
  ${rev.path} ${rev.revision}
[/#list]
[/#list]
```

Primary Data Model Objects

Object name	Function	Notes
repname	The name of the repository	
siteurl	Base URL of the FishEye instance	
timezone	The time zone as configured in admin	
watchpath	The path for this watch	
changesets	A list of changesets	

The syntax to use the data model object '**repname**' as an example, is as follows:

```
${repname}
```

Changeset objects

The **changesets** list will contain multiple changesets for batch (e.g. daily) notifications and one element for immediate notifications.

These changeset objects have the following properties:

Object name	Function	Notes
comment	The commit comment	Belongs to a <i>changeset</i>
author	Author of the commit	Belongs to a <i>changeset</i>
dateValue	the date of the commit	Belongs to a <i>changeset</i>
revisionInfos	A list of revisions for the changeset	Belongs to a <i>changeset</i>

For example, to iterate through all the changesets notifications, you would use the following:

```
[#list changesets as cs]
  ${cs.id} ${cs.author}
[/#list]
```

Revision objects

Object name	Function	Notes
path	The path of the file	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>
revision	The revision number	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>
binary	Boolean indicating whether file is binary	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>
dead	Boolean indicating whether file is deleted	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>
move	Boolean indicating whether file is moved	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>
copy	Boolean indicating whether file is copied	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>
added	Boolean indicating whether file is added	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>
linesAdded	Number of lines added	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>

linesRemoved	Number of lines removed	Belongs to a <i>revisionInfo</i> , under a <i>changeset</i>

Migrating to an external database

This page provides an overview of connecting FishEye to an external database. Atlassian recommends that you use an external database for production installations, for the following reasons:

- **Improved protection against data loss:** The FishEye built-in database, which runs [HSQLDB](#), is susceptible to data loss during system crashes. External databases are generally more resistant to data loss during a system crash. HSQLDB is not supported in production environments and should only be used for evaluation purposes.
- **Performance & scalability:** If you have a large number of users on your FishEye instance, running the database on the same server as FishEye may slow it down. When using the embedded database, the database will always be hosted and run on the same server as FishEye.
- **Data stored in the FishEye database:** The FishEye database stores information besides the cache for repository scans. Specifically, user data and user preferences information.

On this page:

- [Supported databases](#)
- [Support for other databases](#)
- [Notes](#)

Related pages:

- [Configuring the database connection pool](#)

Supported databases

You can use a number of alternatives to the built-in HSQLDB database for storing FishEye and Crucible's relational data. The supported databases versions are listed on the [Supported platforms](#) page.

The pages linked below detail the steps required to migrate to an external database:

- [Migrating to MySQL](#)
- [Migrating to Oracle](#)
- [Migrating to PostgreSQL](#)
- [Migrating to SQL Server](#)

Support for other databases

If you are using another database product that you would like to see supported, please create a [JIRA issue](#) for it under the Crucible project.

Notes

FishEye uses Read Committed transaction isolation. There is no requirement to configure this explicitly when setting up an external database - FishEye will configure the transaction isolation when connecting to the database.

Migrating to MySQL

This page describes how to use FishEye/Crucible with both MySQL Enterprise Server and MySQL Community Server.

To switch to a MySQL database, install MySQL and then follow the steps below. Please note that during the

migration of database servers, the FishEye/Crucible instance will not be available to users or to external API clients.

⚠ Note that for FishEye 2.9+, the JDBC driver for MySQL is *not bundled* with FishEye/Crucible (due to [licensing restrictions](#)).

Prerequisites

Before you begin:

1. Install a *supported version* of MySQL. Check the [Supported platforms](#) page for the exact versions that are supported.
2. Download and install the JDBC driver, if necessary. Note that for FishEye 2.9+, the JDBC driver for MySQL is *not bundled* with FishEye/Crucible (due to [licensing restrictions](#)).
 - a. Download the **MySQL Connector/J JDBC driver** from the [MySQL download website](#).
 - b. Expand the downloaded **zip/tar.gz file**.
 - c. Copy the `.jar` file to your `FISHEYE_INST/lib` directory. If the `lib/` directory doesn't already exist, you will need to create it.
 - d. Restart FishEye/Crucible.

On this page:

- [Prerequisites](#)
- [Step 1. Create a MySQL Database](#)
- [Step 2. Configure FishEye/Crucible to use MySQL, and Migrate Data](#)

Related pages:

- [Migrating to PostgreSQL](#)
- [Migrating to Oracle](#)
- [Migrating to SQL Server](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Step 1. Create a MySQL Database

1. Create a UTF-8 Database:

```
CREATE DATABASE fisheye CHARACTER SET utf8 COLLATE utf8_bin;
```

2. You will also need to set the Server Characterset to utf8. This can be done by adding the following in `my.ini` for Windows or `my.cnf` for other operating systems (create the file at `/etc/my.cnf` if it doesn't already exist). It has to be declared in the Server section, which is the section after `[mysqld]`:

```
[mysqld]
character-set-server=utf8
```

3. Use the **status** command to verify database character encoding information.
First select the fisheye database: `mysql> use fisheye;`
Then issue the `mysql> status;` command:

Screenshot: Using the MySQL Status Command

```
mysql> status
-----
/Applications/xampp/xamppfiles/bin/mysql Ver 14.12 Distrib 5.0.67, for apple-da
rwin8.11.1 (i686) using EditLine wrapper

Connection id:          5764
Current database:
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.0.67 Source distribution
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    utf8
Db characterset:        utf8
Client characterset:    latin1
Conn. characterset:     latin1
UNIX socket:            /Applications/xampp/xamppfiles/var/mysql/mysql.sock
Uptime:                 29 days 1 hour 34 min 56 sec

Threads: 6  Questions: 49238134  Slow queries: 0  Opens: 460445  Flush tables: 1
Open tables: 4  Queries per second avg: 19.607
-----

mysql> 
```

4. Create a user that can log in from the host that FishEye/Crucible is running on and make sure that the user has full access to the newly created database. In particular, the user should be allowed to create and drop tables, indexes and other constraints.

For instance, when FishEye/Crucible and MySQL run on the same machine (accessible through localhost), issue the following commands (replacing username and password with the appropriate values):

```
mysql> grant all on fisheye.* to 'username'@'localhost' identified
by 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

Step 2. Configure FishEye/Crucible to use MySQL, and Migrate Data

In order to migrate to a different database backend, you must create a backup of sql data, configure the database and finally import the data via a backup restoration process. This can be done from either the FishEye/Crucible administration console, which streamlines the process, or via the command line tool which FishEye/Crucible provides.

From FishEye/Crucible's Administration

1. Navigate to the Database page in FishEye/Crucible's Administration console
2. Then click **'Test Connection'** to verify that FishEye/Crucible can log in to the database:
3. Select MySQL from the database type
4. Fill in the appropriate fields, replacing the host, port, database name, username and password as required
5. Click on Test Connection to validate the values

Screenshot: Testing the Connection

Database Configuration

To switch to a different database, specify the database's configuration settings in the form below and use the Test Connection button to verify that the database can be used. We are currently connected to the Oracle database at jdbc:oracle:thin:@localhost:1521:xe.

Changes to connection pool settings will take effect after the next restart before the new settings will take effect.

Type: Built-In
PostgreSQL
Oracle
MySQL

Driver Location: User Supplied - FISHEYE_INST/lib

URL: jdbc:mysql://localhost/dbname

User Name: username

Password: *****

Minimum Pool Connections: 5

Maximum Pool Connections: 20

Parameters: useUnicode=true
characterEncoding=UTF8
connectionCollation=UTF8_bin

Test Connection
Cancel Save & Migrate Save

If this fails, verify that you have the MySQL JDBC driver .jar file in the classpath (see [Prerequisites](#) section above for instructions on how to install the driver). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

6. Click **Save & Migrate** to start the migration process.

During the migration process (which will take several minutes, depending on the size of your database and network throughput), the product will be inaccessible to users and external API clients. Users will see a maintenance screen that informs them of the process. Should the migration fail for any reason, FishEye/Crucible will not switch to the new database and report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration.

Screenshot: Migrating the Database

Migrating Database

 **Migrating database...**

From: jdbc:hsqldb:file:/Users/ervzijst/cenqua/demotion/var/data/crddb/crucible
To: jdbc:mysql://localhost/crucible

```

Started
Creating table definitions...
1 rows written, 0 tables completed.
9 rows written, 1 tables completed.
2328 rows written, 2 tables completed.
7327 rows written, 2 tables completed.
11143 rows written, 3 tables completed.
12676 rows written, 3 tables completed.
12740 rows written, 4 tables completed.
22739 rows written, 4 tables completed.
30492 rows written, 5 tables completed.
30493 rows written, 5 tables completed.
30497 rows written, 6 tables completed.
  
```

Ok

From the command line

1. Create a backup of the sql data from the FishEye/Crucible instance. Read [Backing up and restoring FishEye data](#) and [Backing up and restoring Crucible data](#) for information on how to create a backup.
2. Run the following command from the bin directory in FISHEYE_INST

```
$ ./fisheyectl.sh restore --sql \  
    --file /path/to/backup.zip \  
    --dbtype mysql \  
    --jdbcurl jdbc:mysql://hostname/dbname \  
    --username crucible \  
    --password password
```

3. When the import is complete, FishEye/Crucible can be started and will use MySQL.

Migrating to Oracle

To switch to an Oracle database, install Oracle and follow the steps below. Please note that during the migration of database servers, the FishEye/Crucible instance will not be available to users or to external API clients.

i Oracle support for FishEye/Crucible and Crucible was introduced in version 2.5.0. In order to migrate to Oracle, your instance must be *currently running* at least version 2.5. If you are running an older version, then you will be required to first upgrade FishEye/Crucible and then migrate.

On this page:

- [Step 1. Install and Create a Oracle Database](#)
- [Step 2. Configure FishEye/Crucible to use Oracle, and Migrate Data](#)

Related pages:

- [Migrating to MySQL](#)
- [Migrating to PostgreSQL](#)
- [Migrating to SQL Server](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Step 1. Install and Create a Oracle Database

1. The JDBC drivers for Oracle are bundled with FishEye/Crucible. Skip to step 2 if this meets your needs. If you want to install a specific, different version of the bundled JDBC driver, download the download the Oracle JDBC driver .jar file from the Oracle website (<http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>) and copy the .jar file to your FISHEYE_INST/lib directory (create the lib/ directory if it doesn't already exist). Move the existing JDBC .jar file to another location (and back it up). Restart FishEye/Crucible to have it pick up the new driver.
2. Because creating a database with Oracle is a complex process, we recommend speaking to your resident DBA for creation of a new database for usage with Crucible. We *highly* recommend installing Oracle with the AL32UTF8 encoding otherwise you may see encoding issues in the product.

Step 2. Configure FishEye/Crucible to use Oracle, and Migrate Data

In order to migrate to a different database backend, you must create a backup of sql data, configure the database and finally import the data via a backup restoration process. This can be done from either the

FishEye/Crucible administration console, which streamlines the process, or via the command line tool which FishEye/Crucible provides.

From FishEye/Crucible's Administration

1. Navigate to the Database page in FishEye/Crucible's Administration console
2. Then click '**Test Connection**' to verify that FishEye/Crucible can log in to the database:
3. Select Oracle from the database type
4. Fill in the appropriate fields, replacing the host, port, database name, username and password as required
5. Click on Test Connection to validate the values

Screenshot: Testing the Connection

Database Configuration 🟢

To switch to a different database, specify the database's configuration settings in the form below and use the Test Connection button to verify that the database can be used. We are currently connected to the PostgreSQL database at jdbc:postgresql://localhost/crucible.

Changes to connection pool settings will take effect after the application is restarted before the new settings will take effect.

Type: Built-In
PostgreSQL
Oracle
MySQL

Driver Location:

URL:

User Name:

Password:

Minimum Pool Connections:

Maximum Pool Connections:

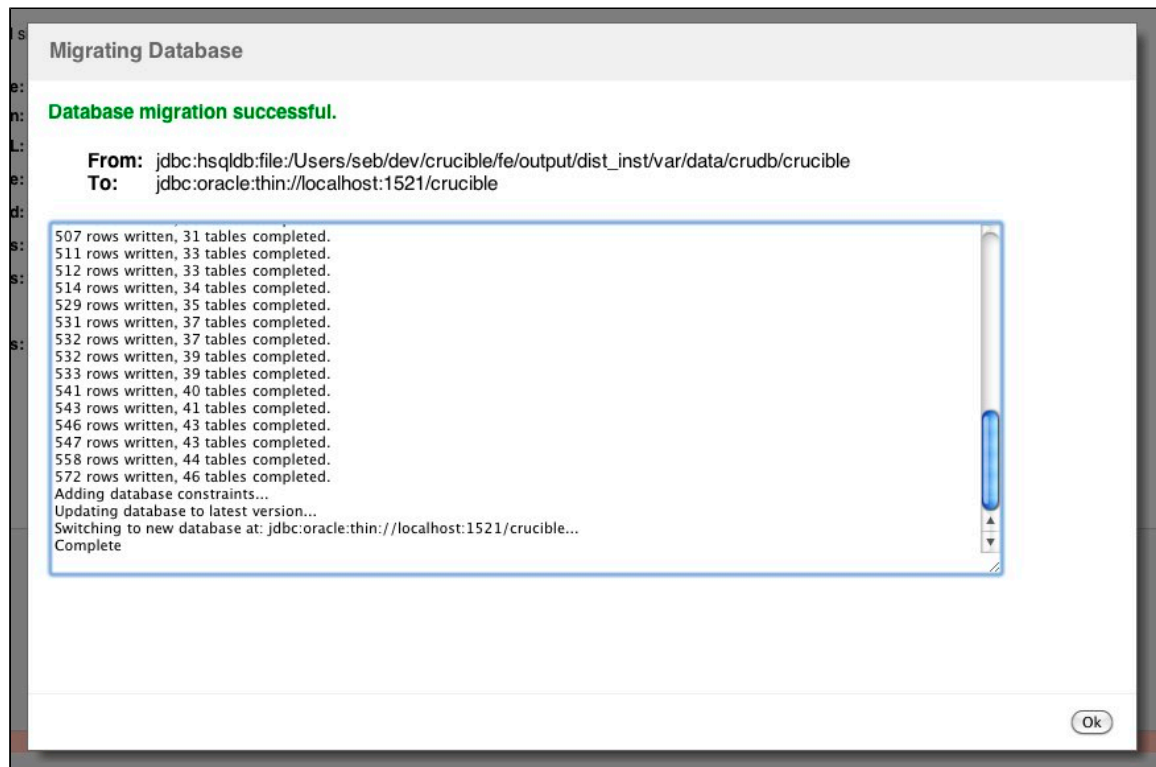
Parameters:

Buttons: Cancel, Save & Migrate, Test Connection, Save

If this fails, verify that you have the Oracle JDBC driver .jar file in the classpath (by placing the .jar file in FISHEYE_INST/lib). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

6. Click '**Save & Migrate Data**' to start the migration process.

During the migration process (which will take several minutes, depending on the size of your database and network throughput), the product will be inaccessible to users and external API clients. Users will see a maintenance screen that informs them of the process. Should the migration fail for any reason, FishEye/Crucible will not switch to the new database and report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration.

Screenshot: Migrating the Database**From the command line**

1. Create a backup of the **sql data** from the FishEye/Crucible instance. Information on how to create a backup can be found at [Backing up and restoring FishEye data](#) \ [Backing Up and Restoring Crucible Data](#)
2. Run the following command from the bin directory in FISHEYE_INST

```
$ ./fisheyectl.sh restore --sql \  
    --file /path/to/backup.zip \  
    --dbtype oracle \  
    --jdbcurl jdbc:oracle:thin:@hostname:port:dbname \  
    --username crucible \  
    --password password
```

3. When the import is complete, FishEye/Crucible can be started and will use Oracle.

Migrating to PostgreSQL

To switch to a PostgreSQL database, install PostgreSQL and follow the steps below. Please note that during the migration of database servers, the FishEye/Crucible instance will not be available to users or to external API clients.

On this page:

- [Step 1. Install and Create a PostgreSQL Database](#)
- [Step 2. Configure FishEye/Crucible to use PostgreSQL, and Migrate Data](#)

Related pages:

- [Migrating to MySQL](#)
- [Migrating to Oracle](#)
- [Migrating to SQL Server](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Step 1. Install and Create a PostgreSQL Database

1. The JDBC drivers for PostgreSQL are bundled with FishEye/Crucible. Skip to [Step 2](#) if this meets your needs. If you want to install a specific, different version of the bundled JDBC driver, download the PostgreSQL JDBC driver .jar file from the [PostgreSQL website](#) and copy the .jar file to your FISHEYE_INST/lib directory (create the lib/ directory if it doesn't already exist). Move the existing JDBC .jar file to another location (and back it up). Restart FishEye/Crucible to have it pick up the new driver.
2. Create a new database user (replacing 'username' and 'password' with the appropriate values):

```
$ psql
> create user username password 'password';
```

3. Create a UTF-8 database and make the newly created user the owner:

```
> create database crucible ENCODING 'UTF-8' OWNER username;
```

4. Make sure the user has full access to the database:

```
> grant all on database crucible to username;
```

Step 2. Configure FishEye/Crucible to use PostgreSQL, and Migrate Data

In order to migrate to a different database backend, you must create a backup of sql data, configure the database and finally import the data via a backup restoration process. This can be done from either the FishEye/Crucible administration console, which streamlines the process, or via the command line tool which FishEye/Crucible provides.

From FishEye/Crucible's Administration

1. Navigate to the Database page in FishEye/Crucible's Administration console.
2. First click **Edit** and then click **Test Connection** to verify that FishEye/Crucible can log in to the existing database.
3. Select PostgreSQL from the database **Type**.
4. Fill in the appropriate fields:
 - a. **Driver Location**: either your own PostgreSQL JDBC or the **Bundled** one that came with FishEye
 - b. **URL**: create this field by replacing the host, port, and database name with your own (i.e. jdbc:postgresql://localhost:5432/<dbname> e.g. jdbc:postgresql://localhost:5432/crucible)
 - c. **Username**: your DB username
 - d. **Password**: your DB password
 - e. **Minimum Pool Connections**: 5 is the default
 - f. **Maximum Pool Connections**: 20 is the default

- g. **Parameters:** (one per line)
 - i. useUnicode=true
 - ii. characterEncoding=UTF8

5. Click **Test Connection** to validate the values.

Screenshot: Testing the Connection

Database Configuration 🟢

To switch to a different database, specify the database's configuration settings in the form below and use the Test Connection button to verify that the database can be used. We are currently connected to the PostgreSQL database at jdbc:postgresql://localhost:5432/crucible.

Changes to connection pool sizes require a full restart before the new settings will take effect.

Type: **PostgreSQL** (dropdown menu showing Built-In, PostgreSQL, Oracle, MySQL)

Driver Location: **PostgreSQL** (dropdown menu)

URL: **jdbc:postgresql://localhost:5432/crucible**

User Name: **crucible**

Password: *********

Minimum Pool Connections: **5**

Maximum Pool Connections: **20**

Parameters: **useUnicode=true
characterEncoding=UTF8**

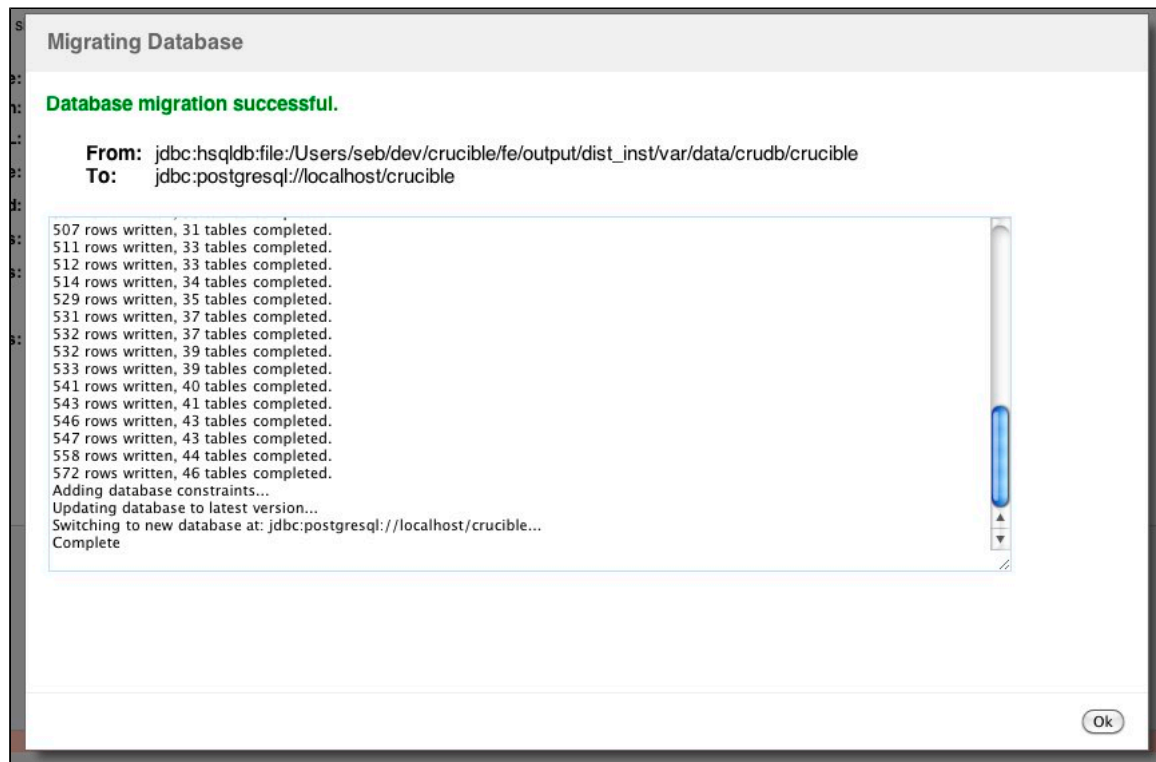
Buttons: Cancel, Save & Migrate, Test Connection, Save

If this fails, verify that you have the PostgreSQL JDBC driver .jar file in the classpath (by placing the .jar file in FISHEYE_INST/lib). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

6. Click **Save & Migrate** to start the migration process.

During the migration process (which will take several minutes, depending on the size of your database and network throughput), the product will be inaccessible to users and external API clients. Users will see a maintenance screen that informs them of the process. Should the migration fail for any reason, FishEye/Crucible will not switch to the new database and report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration.

Screenshot: Migrating the Database



From the command line

1. Create a backup of the sql data from the FishEye/Crucible instance. Information on how to create a backup can be found at [Backing up and restoring FishEye data](#) / [Backing up and restoring Crucible data](#).
2. Run the following command from the bin directory in FISHEYE_INST:

```
$ ./fisheyectl.sh restore --sql \  
    --file /path/to/backup.zip \  
    --dbtype postgresql \  
    --jdbcurl jdbc:postgresql://hostname/dbname \  
    --username crucible \  
    --password password
```

3. When the import is complete, FishEye/Crucible can be started and will use PostgreSQL.

Migrating to SQL Server

To migrate FishEye/Crucible to an SQL Server database, install SQL Server and follow the steps below.

Before You Begin

Check that you are using version of SQL Server that is supported for use with FishEye. See [Supported platforms](#).

An existing Java bug prevents connection with Java 1.6.0_29 and above (including Java 1.7.0). Read more about the issue and possible workarounds [here](#).

Step 1. Install and Create an SQL Server Database

See the [SQL Server Online resources \(MSDN\)](#) for instructions on how to install and create an SQL Server

database.

Please note the following FishEye/Crucible-specific information when installing and creating an SQL Server database:

- The JDBC jtds drivers for SQLServer are bundled with Crucible. We do not support using the Microsoft distributed jdbc driver.
- Your database server must be configured to use the Latin1_General_CS_AS collation set.

On this page:

- [Before You Begin](#)
- [Step 1. Install and Create an SQL Server Database](#)
- [Step 2. Configure FishEye/Crucible to Use SQL Server and Migrate Data](#)

Related pages:

- [Migrating to MySQL](#)
- [Migrating to PostgreSQL](#)
- [Migrating to SQL Server](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Step 2. Configure FishEye/Crucible to Use SQL Server and Migrate Data

In order to migrate to a different database backend, you must create a backup of SQL data, configure the database and finally import the data via a backup restoration process. This can be done from either the Crucible administration console, which streamlines the process, or via the command line tool which Crucible provides. These two methods are described below. The following resources may be of interest:

- [Backing up and restoring FishEye data](#)
- [Backing up and restoring Crucible data](#)
- [SQL Server Online resources \(MSDN\)](#)

[[Configuring and Migrating via FishEye/Crucible's Administration Console](#)] [[Configuring and Migrating via the Command Line](#)]

Configuring and Migrating via FishEye/Crucible's Administration Console

Before you begin:

- Note, during the migration process (which will take several minutes, depending on the size of your database and network throughput), **the FishEye/Crucible instance will be inaccessible to users and external API clients**. Users will see a maintenance screen that informs them of the process.
- If you are attempting a migration after a previous migration has failed, you must drop all tables, indexes and constraints before attempting a new migration. This is because the destination database may contain data from the previous migration attempt.
- Verify that you have the jtds JDBC driver .jar file in the classpath (by placing the .jar file in FISHEYE_INST/lib).
- Ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

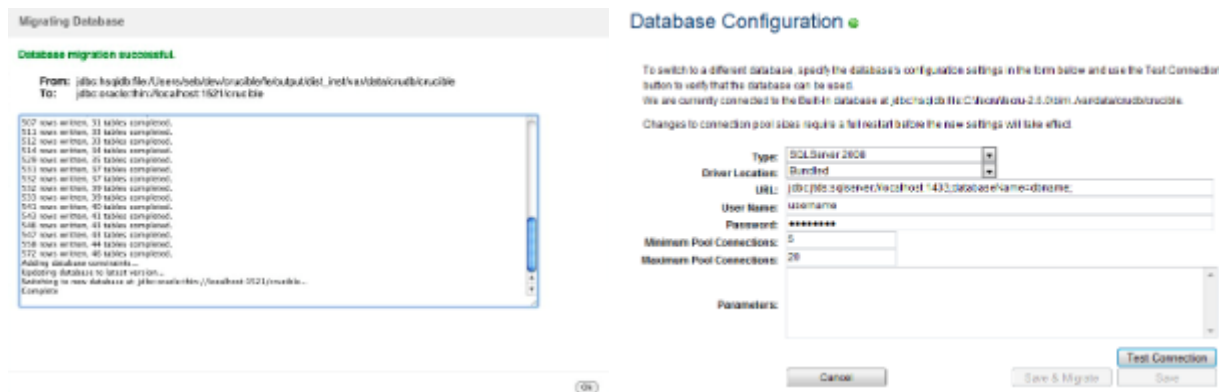
To configure FishEye/Crucible to use SQL Server and migrate data using the Administration Console:

1. Navigate to the '**Database**' page in FishEye/Crucible's Administration console.

- Configure FishEye/Crucible to use SQL Server, as follows:
 - Select '**SQLServer**' from the '**Type**' dropdown.
 - Complete the appropriate fields, replacing the '**URL**' (host, port and database name), '**User Name**' and '**Password**' as required.

NOTE: The default SQL server instance listens on port 1433. If your instance is not the default, use the port number that is associated with your particular instance.

e.g. **URL**: jdbc:jtds:sqlserver://localhost:1433;databaseName=*your database name here*;
- Click '**Test Connection**' to verify that FishEye/Crucible can log in to the database (see 'Testing the Connection' screenshot below).
- Click '**Save & Migrate Data**' to start the migration process (see 'Migrating the Database' screenshot below). If the migration fails, FishEye/Crucible will not switch to the new database and will report the problems encountered.



2. Migrating Data

1. Configuring the Database

Screenshots above: Configuring FishEye/Crucible to use SQL Server and migrating data (click to view full-size images)

Configuring and Migrating via the Command Line

To configure FishEye/Crucible to use SQL Server and migrate data using the Command Line:

- Create a backup of the sql data from the FishEye/Crucible instance. Information on how to create a backup can be found at [Backing up and restoring FishEye data](#) \ [Backing up and restoring Crucible data](#)
- Run the following command from the bin directory in FISHEYE_INST:

```
$ ./fisheyectl.sh restore --sql \
  --file /path/to/backup.zip \
  --dbtype sqlserver2008 \
  --jdbcurl
"jdbc:jtds:sqlserver://hostname:port;databaseName=dbName;" \
  --username crucible \
  --password password
```

- When the import is complete, FishEye/Crucible can be started and will use SQLServer.

[[Configuring and Migrating via FishEye/Crucible's Administration Console](#)] [[Configuring and Migrating via the Command Line](#)]

Backing up and restoring FishEye data

FishEye data can be backed up from the admin interface or command line. This page contains the command syntax, options and the required procedure to backup and restore your FishEye instance.

 If you are migrating to a different machine, please follow the instructions on our [Migrating FishEye Between Servers](#) page.

Backing up FishEye data

Using the admin interface to backup

1. Navigate to the FishEye Admin area (click **Administration** in the footer of any FishEye page).
2. Click **Backup** (under 'System Settings').
3. On the Backup screen, **File Path** indicates where the backup file (in .zip format) will be stored. You can edit the name of the backup file.
4. Choose the components, described in the table [below](#), that should be included in the backup file. Note that the backup will always include the configuration data (config.xml), your license file and the FishEye user data.
 -  Repository and application caches contain temporary data stored from repository scans and library caches that improve startup time. Both will be recreated automatically by re-scanning the source repositories, so the backup files can be significantly reduced by excluding these (if the cost of re-scanning is acceptable).
5. Click **Create Backup Now**.

On this page:

- [Backing up FishEye data](#)
 - [Using the admin interface to backup](#)
 - [Using the command line to backup](#)
 - [Scheduling FishEye backups](#)
- [Restoring FishEye data](#)
 - [Using the command line to restore FishEye data](#)
 - [Restore command line options](#)
 - [Migrating backup data](#)

Related pages:

- [Migrating FishEye Between Servers](#)
- [Migrating to an external database](#)
- [Troubleshooting Databases](#)

Component	Purpose
ActiveObjects	Data that is stored by plugins
Repository and application caches	The cache contains data that reflects the state of FishEye's repositories. Without it, FishEye must re-scan its repositories after a backup is restored. The cache also contains OSGI library data that increases startup time. These too can be excluded and will be generated automatically when the application is started.
Plugins	Plugins are 3rd-party extensions that you may have installed, and configuration for all plugins (this includes configuration for Crucible's set of standard plugins).

SQL Database	Refers to the SQL content database (used by both FishEye and Crucible and containing all user profile data, reviews and their comments).
Web templates	In this context, these are custom freemarker templates that you or your users have created. They live in FISHEYE_INST/template.
Uploaded files	In this context, uploads refers to files which are added to Crucible via the web interface (such as patch file reviews). It also includes each repository-backed file that went under review, when Crucible is configured to make a local copy of every reviewed file .

Screenshot: The FishEye Backup screen

Backup

FishEye and Crucible will be unavailable until the backup completes

Depending on the number of repositories and reviews, the backup may take a long time to complete and may require large amounts of disk space (especially when you are including the cache). Please make sure you have enough disk space before proceeding.

Your managed repositories will not be backed up!

FishEye does not automatically backup content stored in your managed repositories. Please ensure that a backup process which is satisfactory to your requirements is in place. [Read more](#)

Create a new backup

Filename

Specify the filename on the server where FishEye and Crucible should export.
This file will be created in /opt/j2ee/domains/atlassian.com/eac-fisheye/data/backup

Include

- ☒ ActiveObjects
- ☒ repository and application caches
- ☒ plugins and their configuration data
- ☒ SQL database
- ☒ web templates
- ☒ uploaded files and local copies of files under review

Choose the components that FishEye and Crucible should include in the backup.

or [Manage Scheduled Backups](#)

Using the command line to backup

i Your FishEye instance must be running during the backup.

1. Open a terminal or command prompt on the FishEye server computer.
2. Navigate to the <FishEye home directory>/bin/ directory.
3. Run the backup command on the command line using any desired options (described in the table [below](#)).
By default, all backup components, except repository cache data, is included in the backup.

The backup is created as a new Zip archive file and placed in the FISHEYE_INST/backup/ directory.

i Note that if your FishEye instance uses a custom FISHEYE_INST directory, make sure the environment variable is properly set when running the backup command.

Option	Switch	Default setting
Quiet mode	-q, --quiet	No

Output filename	-f, --file	FISHEYE_INST/backup/ is the default directory.
Compression level	-c, --compression	Yes (6)
Anonymise	-a, --anonymise	No
Cache Backup	--cache	No. By default, the cache data is excluded from backups.

Advanced backup command line settings

In some cases it might be preferable to only backup a limited set of items. This could be useful when your instance uses an external database such as MySQL or PostgreSQL and your DBA has already configured automatic backups in the database. The commands below allow this.

Option	Switch	Description	Default
Exclude Plugins	--no-plugins	Excludes plugins from the backup.	No. By default, plugins are included in every backup.
Exclude Templates	--no-templates	Excludes templates from the backup.	No. By default, templates are included in every backup.
Exclude Uploads	--no-uploads	Excludes uploaded files (such as patch reviews, stored in Crucible's internal database) from the backup.  This item only applies when using Crucible with FishEye.	No. By default, uploads are included in every backup.
Exclude SQL Database	--no-sql	Excludes the SQL content database used by both FishEye and Crucible.	No. By default, this data is included in every backup.

Backup command line examples

These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyectl.bat` and use the correct slashes. Run the command from the <FishEye home directory>/bin/ directory.

The basic syntax of the backup command is as follows:


```
$ ./fisheyectl.sh backup [OPTIONS]
```

To see inline help for all backup options, run the following command in the <FishEye home directory>/bin/ directory:

```
$ ./fisheyectl.sh backup --help
```

Backing up with compression of 9, quiet mode and setting an output location:

```
$ ./fisheyectl.sh backup --compression 9 -q -f  
/application_backups/fisheye/20090215.zip
```

Backup including cache data (also includes all default components):

```
$ ./fisheyectl.sh backup --cache
```

Known limitations

Please note that the below limitations are common for any Java-based backup tool.

Archives Containing Over 65535 Files

Versions of Java earlier than v1.6 (b25) are incapable of handling zip files that contain more than 65,535 files. The solution for this problem is to either upgrade to a version of Java later than v1.6 (b25), or ensure that the archive does not exceed the threshold (contains less than 65,535 files). The FishEye cache (not included in backups by default) can be a contributor of many small files. Hence, exclude the cache from backups if this is likely to be a concern.

Archives Larger Than 4GB

Java has trouble reading and writing zip files that are larger than 4GB. As of release 1.5 Java appears capable of reliably creating archives that are over 4GB, but remains unable to extract them. For details see [Sun's bug report](#). Also be aware of the fact that some file systems (including [FAT32](#)) have trouble with files larger than 4GB.

As a workaround, make sure you do not create archives that are larger than 4GB. The FishEye cache (not included in backups by default) can be a contributor of a lot of small files (although these tend to compress very well). If you still want to archive everything and end up with an archive that is too large, consider creating separate backups for the FishEye cache and uploaded files respectively.

Scheduling FishEye backups

To set a schedule for automatic backups:

1. Go to the Admin area and click **Backup** (under 'System Settings').
2. Click **Manage Scheduled Backups** at the bottom of the page.
3. Click **Edit**, set the desired options, then click **Save**.

The options for scheduled backups are detailed in the table below:

Option name	Description	Allowed Values
Disable Scheduled Backups	Stops regular backups from taking place.	On (disabled) or Off (enabled)
Backup path	The path where the backup .zip file will be stored.	Any system or network path that FishEye or Crucible can access.
Backup file prefix	Characters that will be added to the beginning of the backup file name.	Any string of characters that can be used as part of a filename on the local operating system.
Backup file date pattern	Sets a date for the next (or initial) backup to take place.	Any valid date in the format yyyy_MM_dd (year, month, day of the month).
Backup frequency	Sets how often the backup will take place.	Can be set to every day , every Sunday, Monday to Friday and first day of the month .
Backup time (HH:mm)	The time when the backup will take place.	Any valid 24-hour time in the format HH : mm (hours, minutes).
Include	Specifies which items must be included in the backups (these components are explained at the top of this page).	As per the options for regular on-demand backup (These components are explained at the top of this page).

Screenshot: Scheduling backups in FishEye

Scheduled Backups

 Please choose the schedule for your backups carefully. FishEye and Crucible will be unavailable for the duration of the backup process.

 Your managed repositories will not be backed up!
 FishEye does not automatically backup content stored in your managed repositories. Please ensure that a backup process which is satisfactory to your requirements is in place. [Read more](#)

Schedule backups

Disable scheduled backups ☐ Switch scheduled backups on and off.

Backup directory **/opt/j2ee/domains/atlassian.com/eac-fisheye/data/backup**
The location to store the backups. Edit config.xml to change this setting.

Backup file prefix
The prefix for the generated backup file.

Backup file date pattern
The date/time format you specify will be added to the file prefix to differentiate between backup files.

Backup frequency How often do you want FishEye and Crucible to run a backup.

Backup Time (HH:mm)
What time would you like FishEye and Crucible to run the backup, e.g. 17:30.
 Note that this is 24 hour notation in local server time. Current local server time is: 02:04.

Include ☐ ActiveObjects
☒ repository and application caches
☒ plugins and their configuration data
☒ SQL database
☒ web templates
☒ uploaded files and local copies of files under review
Choose the components that FishEye and Crucible should include in the backups.

 Be aware that scheduled backups can fill up disks unless you regularly move or delete old archives.

Restoring FishEye data

Using the command line to restore FishEye data

There is currently no way to restore a backup from the admin interface because FishEye must be shut down during a data restore.

 Restoring a backup will irreversibly overwrite the data of your installation with the data from the backup archive. Note that you cannot restore data into versions of FishEye which are older than the version that created the backup.

1. [Install FishEye](#) into a new, empty directory (this must be the same version that the backup was created from, or later).
2. Make sure the FishEye instance is not running.
3. Open a command line interface on the FishEye server computer.
4. Run the restore command on the command line with any desired options.
5. The specified elements will be restored.
6. Start the FishEye instance.
7. When using FishEye integrated with Crucible, you will need to re-index your repositories after restoring data, unless the backup archive was created with the `--cache` option.

Restore command line options

By default, the restore process will restore all items found in the backup archive (so if you included the caches using the `--cache` option, these will automatically be restored). However, you can specify a partial restore, by explicitly specifying the item names on the command line.

If you are using an external database (as opposed to the default HSQL database), make sure the JDBC driver file is present in the `FISHEYE_INST/lib` directory when running restore.

Furthermore, if you are restoring to a new FishEye instance and home directory, and select `--dbtype` of `mysql`, you must download the JDBC driver and accept the license agreement before proceeding with the restore.

The options available for use with the restore command are listed in the following table:

Option	Switch	Description
Choose file to restore from	<code>-f PATH/FILENAME,</code> <code>--file PATH/FILENAME</code>	(Required) Restore the backup from PATH/FILENAME.
Repository and application caches	<code>--cache</code>	Restore the repository cache backup.
Plugins	<code>--plugins</code>	Restore 3rd-party plugins and their configuration data.
Web templates	<code>--templates</code>	Restore freemarker templates from the backup (the restored instance will use the built-in templates).

Uploaded files	--uploads	Restore uploads (e.g. patch files uploaded into Crucible and contents of files under review).  This item only applies when using Crucible with FishEye.
SQL Database	--sql	Restore the SQL database containing user profiles, reviews and review comments.
ActiveObjects	--ao	Restore ActiveObjects data stored by plugins.
List backup contents	-l, --list	List the contents of the backup file, and exit.
Set database type	-t, --dbtype	SQL database type ('mysql', 'postgresql', 'oracle', 'sqlserver2005', 'sqlserver2008', 'hsql'). Only required when restoring to a database location different to that used at backup time.
Set JDBC URL	-j, --jdbcurl	JDBC URL of the SQL database. Only required when restoring to a database location different to that used at backup time (not applicable for 'hsql').
Set JDBC username	-u, --username	JDBC username of the SQL database. Only required when restoring to a database location different to that used at backup time (not applicable for 'hsql').
JDBC password	-p, --password	JDBC password of the SQL database. Only required when restoring to a database location different to that used at backup time (not applicable for 'hsql').
JDBC class	-d, --driver	Optionally, specify the JDBC driver class name needed to access the SQL database. Only required when restoring to a database location different to that used at backup time and when using a different JDBC driver than the standard driver associated with the database specified though - -d b type. (Not applicable for 'hsql'.)

JDBC driver source	<code>-s, --driver-source</code>	Optionally, specify the JDBC driver file. The default ('bundled') is to use the bundled JDBC driver file; 'user' = use the driver from the FI SHEYE_INST/lib directory.
Suppress output	<code>-q, --quiet</code>	Suppress the output messages from the restore program on the command line.
Overwrite the existing DB	<code>--force</code>	Overwrite the existing database, without warning.
Don't batch SQL inserts	<code>--no-batch-sql</code>	Do not batch SQL inserts. This can be useful when diagnosing errors.
Display the help	<code>-h, --help</code>	Display the help, and exit.

 These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyectl.bat` and use the correct slashes. Run the command from the <FishEye home directory>/bin/ directory.

The basic syntax of the restore command is as follows:

```
$ ./fisheyectl.sh restore -f /path/to/backup_2009-10-02_1138.zip [OPTIONS]
```

To see inline help for all backup options, run the following command in the <FishEye home directory>/bin/ directory:

```
$ ./fisheyectl.sh restore --help
```

Restoring a backup with cache data (also restores all default components):

```
$ ./fisheyectl.sh restore --cache
```

Migrating backup data

When the process restores a SQL database, it looks at the configuration data (`config.xml`) included in the backup archive to learn which database product was used and how to connect to it. When FishEye uses the built-in HSQLDB database (which is the default), the restored instance will also use that.

However, when the restored instance will use a different database than the backed up instance (for instance, HSQLDB was used at the time the backup was created, but it needs to be restored on MySQL), you should use the command line options to point the process to the new database.

Command line example: migrating backup data to MySQL

 These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyectl.bat` and use the correct slashes. Run the command from the `<FishEye home directory>/bin/` directory.

Restoring to a FishEye instance that uses a different database (ensure the MySQL driver jar file is present in the `FISHEYE_INST/lib` directory)

```
$ ./fisheyectl.sh restore \  
--username john \  
--password smith \  
--jdbcurl jdbc:mysql://localhost:3306/crucible \  
--dbtype mysql \  
--file /path/to/backup_2009-10-02_1138.zip
```

Backing Up and Restoring Managed Repositories

Backing up Managed Repositories

Managed repositories are **not** backed up by FishEye's built-in [backup and restore](#) feature. Instead, you will need to design and implement your own backup procedure.

All managed repositories are stored under a single directory in FishEye, which should be backed up regularly along with a copy of your `config.xml`.

By default, this directory lives here:

FISHEYE_INST/managed-repos

This location can be configured by modifying your `config.xml` file (see [Setting the root location for internal repositories](#) for more details on changing this value).

To back up the `managed-repos` directory, you need to create a snapshot of the `managed-repos` directory (including all subdirectories), then back up that snapshot. Note that the directory structure under the `managed-repos` directory **must** be preserved in the snapshot. Your `config.xml` file should also be backed up at the same time.

Creating this snapshot is an operating system-specific task, e.g.:

- On Windows, a batch script copying the directory can be written and scheduled periodically (**All Programs > Accessories > System Tools > Task Scheduler**). There are also various utilities available to simplify this (eg. [PicoZip](#)).
- On Linux/Solaris, it is best to write a small shell script, placed in `/etc/cron.daily`, backing up files to a directory like `/var/backup/fisheye-repos`. It is best to copy an existing script in `/etc/cron.daily` to ensure local conventions (file locations, lockfiles, permissions) are adhered to.

Restoring Managed Repositories from Backup

Warning!

Restoring a backup requires you to revert your FishEye configuration to the point where the backup was made, and may require a [full re-index](#) of your repository, which may take a long time. Restoring a managed repository backup should only be done in a case of critical data loss.

To restore a backed up snapshot:

1. Stop your FishEye instance.
2. Make a backup of the current `config.xml` file and `managed-repositories` directory, in case something goes wrong during the restore process.
3. Replace the current `config.xml` file with the `config.xml` file from your backup.
4. Replace the contents of your `managed-repos` directory with the contents of your backup snapshot.
5. Restart your FishEye server.
6. Perform a [manual re-index](#) for each managed repository that was restored.

Note, it is possible to perform a manual restore for a single repository at a time by copying a single repository from your backup to your `managed-repositories` directory and starting a re-index.

Managing add-ons

An add-on is an installable component that supplements or enhances the functionality of FishEye in some way. For example, the [FishEye Release Report](#) is a free add-on that presents a release report of the new commits in a branch for a given release (based on the Commit Graph), including all commits in feature branches that were merged to the release branch.

FishEye comes with many pre-installed add-ons (called system add-ons). You can install more add-ons, either by acquiring the add-on from the [Atlassian Marketplace](#) or by uploading it from your file system. This means that you can install add-ons that you have developed yourself. For information about developing your own add-ons for FishEye, see the [FishEye developer's guide](#).

On this page:

- [About the Universal Plugin Manager \(UPM\)](#)
- [Administering Add-ons in FishEye](#)

i You may notice that the terms 'add-on' and 'plugin' both appear in the Atlassian documentation and tools. While the terms are often used interchangeably, there is a difference. A plugin is a type of add-on that can be installed into an Atlassian host application. Plugins are what developers create with the Atlassian SDK. But there are other types of add-ons as well. For example, the JIRA client is an add-on that runs as a separate program rather than as a plugin to JIRA. This documentation uses the term 'add-on' most often.

About the Universal Plugin Manager (UPM)

You administer add-ons for FishEye using the Universal Plugin Manager (UPM). The UPM is itself an add-on that exposes add-on administration pages in the FishEye Administration Console. UPM works across Atlassian applications, providing a consistent interface for administering add-ons in FishEye, Crucible, Confluence, JIRA, Stash and Bamboo.

UPM comes pre-installed in recent versions of all Atlassian applications, so you do not normally need to install it yourself. However, like other add-ons, the UPM software is subject to regular software updates. Before administering add-ons in FishEye, therefore, you should [verify your version](#) of the UPM and update it if needed.

Administering Add-ons in FishEye

You can update UPM, or any add-on, from the UPM's own add-on administration pages. Additionally, you can perform these tasks from the UPM administration pages:

- Install or remove add-ons
- Configure add-on settings
- Discover and install new add-ons from the [Atlassian Marketplace](#)
- Enable or disable add-ons and their component modules

For information on performing these add-on administration tasks, see the [Universal Plugin Manager documentation](#).

Integrating FishEye with Atlassian applications

You can integrate FishEye with the following Atlassian applications:

	When FishEye is integrated with JIRA , you can: • Use smart commits to transition JIRA issues See User management limitations and recommendations. When Crucible is integrated with JIRA, you can: • Use JIRA integration in Crucible reviews • Transition JIRA issues i Your user tiers don't need to match between JIRA and FishEye/Crucible in order to integrate them. JIRA users that are not FishEye users will see the same view as FishEye users within JIRA, but will not be able to log in to FishEye to view the source/reviews.	When JIRA is integrated with FishEye, you can: • View an issue's FishEye changesets • Browse a project's FishEye changesets • Add the FishEye Charts Gadget to your JIRA dashboard • Add the FishEye Recent Changesets Gadget to your JIRA dashboard • View an issue's Crucible reviews • Browse a project's Crucible reviews • Add the Crucible Charts Gadget

	When FishEye is integrated with Crucible, you can: <i>In FishEye:</i> • Use smart commits to create Crucible reviews, add reviews to new reviews and update an existing review. <i>In Crucible:</i> • When using Iterative reviews in Crucible, you will be prompted when a new version of a file is available. • Files and changesets displayed in activity streams (e.g. the dashboard activity stream) render as links to the relevant files/changesets. • See your content roots and repositories associated with projects. • See repository lists and browse repositories using the Source tab. • View charts or code metrics. See also What happens if I decide to stop using FishEye with Crucible. i You'll need the same number (or higher) of users in FishEye as Crucible.
	See: • Integrating Crowd with Atlassian FishEye • Integrating Crowd with Atlassian Crucible
	When FishEye is integrated with your Bamboo continuous integration server, you can view the code changes that triggered a build. When a build fails due to a compilation error or failed test, you can explore the failed build in FishEye and jump directly into the changeset that broke the build. You can view the history of that changeset to see what the author was trying to fix, take advantage of the the side-by-side diff view to analyze the change and then open the correct files in your IDE. For more details see Viewing the Code Changes that Triggered a Build.

Screenshots: Code changes listed in a Bamboo build (above), and viewed in FishEye (below)

The screenshot shows the 'History' tab in FishEye. At the top, there's a red header with 'History' and a navigation bar with icons. Below, there's a sub-navigation bar with 'Summary', 'Tests', 'Changes', 'Artifacts', 'Logs', and 'Metadata'. The 'Changes' tab is selected, displaying 'Code Changes'. A message states 'Code has been updated by James Dumay.' Below this, the change is titled 'Angry Nerds'. It shows a commit by James Dumay on 05 Dec 2011, 2:16:12 PM, with commit ID be7df50cf5bd20cf8... The commit message is 'Add main'. The changes listed are: 'cake (version be7df50cf5bd20cf8839ac92c143ab5a580ba64c) (diffs)' and 'main.c (version be7df50cf5bd20cf8839ac92c143ab5a580ba64c) (diffs)'.

The screenshot shows the 'Source' tab in FishEye. The top navigation bar includes 'Dashboard', 'Source', 'Projects', 'People', and 'Reviews'. The 'Source' tab is selected, showing the project 'AngryNerds' and the commit 'be7df50'. The file 'main.c' is selected, showing a code review by James Dumay. The review shows a diff for 'main.c' (+7 lines). The code snippet is:


```
1 #include <stdio.h>
2
3
4 int main() {
5     printf("Hello, world!");
6     return 0;
7 }
```

 The footer of the page reads: 'Atlassian FishEye Git analysis with Crucible code review. (Version:2.7.8 Build:20111130052542 2011-11-30) - Administration - Page generated 2011-12-05 15:12 +1100'.

Trusted applications

A 'trusted application' is an application that FishEye will allow to access specific functions in FishEye on behalf of any user, without the user logging in to FishEye. You can configure Trusted Applications authentication between FishEye, Crucible, JIRA (3.12 and later) and Confluence (2.7 and later).

Using the bundled Application Links plugin, you can configure application links with Trusted Applications authentication as well as other types of authentication.

For further instructions, please see [Configuring application links](#).

JIRA Integration in FishEye

[JIRA](#) is Atlassian's issue tracking and project management application.

When FishEye is [integrated with JIRA](#), you can:

- [Use smart commits in FishEye to transition JIRA issues](#)
- [View a JIRA issue's FishEye changesets](#)

- [Browse a JIRA project's FishEye changesets](#)
- [Add the FishEye Charts Gadget to your JIRA dashboard](#)
- [Add the FishEye Recent Changesets Gadget to your JIRA dashboard](#)

This page describes how to integrate JIRA with FishEye.

Initial configuration

You should check that JIRA is configured to accept remote API calls; in JIRA, go to **Administration > System > General Configuration** and turn on **Accept remote API calls** (under 'Options'). A restart is not required.

See [JIRA options](#) for more detail.

On this page:

- [Initial configuration](#)
- [Quick setup with the FishEye Setup Wizard](#)
- [JIRA integration within FishEye](#)
 - [1. Configuring the application links with JIRA](#)
 - [2. Connecting to JIRA for user management](#)
- [Project links](#)
- [Notes](#)

Related pages:

- [User Management Limitations and Recommendations](#)
- [Configuring Application Links](#)
- [Enabling Smart Commits](#)

Quick setup with the FishEye Setup Wizard

When you are installing FishEye, the setup wizard allows you to configure the JIRA connection automatically. This is a quick way of setting up your JIRA integration with the most common options. See [Configuring JIRA Integration in the Setup Wizard](#).

If you did not set up the JIRA integration during the FishEye setup, you can configure the integration via the FishEye administration screens as described below.

JIRA integration within FishEye

You can further configure the JIRA connections using the FishEye administration screens. There are two parts to the integration process:

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

1. Configuring the application links with JIRA

Below are instructions for creating a two-way trusted applications link between FishEye and JIRA. We assume that your FishEye and JIRA servers are using the same set of users. If not, please refer to [Adding an Application Link](#) for information about the various options available.

To add a two-way trusted applications link between FishEye and JIRA:

Go to your FishEye administration screen and add the application link:

1. Click **'Add Application Link'**. Step 1 of the link wizard will appear.
2. Enter the **server URL** of the application that you want to link to (the 'remote application').
3. Click the **'Next'** button.
4. 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.
5. Click the **'Next'** button.
6. 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](#).
7. Click the **'Create'** button to create the application link.

2. Connecting to JIRA for user management

This option is available only with JIRA 4.3 and later.

If you want to manage your FishEye users using JIRA, you need to configure a connection in JIRA and then set up the user directory in FishEye. Both steps are described below.

2.1. To configure a new connection for user management in JIRA:

1. Log in to JIRA as a user with the 'JIRA System Administrators' global permission.
2. Click **Administration** on the JIRA top navigation bar.
 - a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:
 - In JIRA 4.3: Click **'Other Applications'** in the **'Users, Groups & Roles'** section of the JIRA administration screen.
 - In JIRA 4.4: Select **'Administration' > 'Users' > 'JIRA User Server'**.
 - b. **Add** an application.
 - c. Enter the **application name** and **password** that your application will use when accessing JIRA.
 - d. Enter the **IP address** or addresses of your application. Valid values are:
 - A full IP address, e.g. 192.168.10.12.
 - A wildcard IP range, using CIDR notation, e.g. 192.168.10.1/16. For more information, see the introduction to [CIDR notation on Wikipedia](#) and [RFC 4632](#).
 - **Save** the new application.

2.2. To set up the JIRA user directory in FishEye:

1. Go to your FishEye administration screen.
 - a. Click **Authentication** (under 'Security Settings').
 - b. Click **Setup JIRA/Crowd authentication**. Note, if LDAP authentication has already been set up, you will need to remove that before connecting to JIRA for user management.

c. Make the following settings:

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

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

Project links

We do not recommend the use of project links with FishEye 2.9 and later, if you have [JIRA 5.0 or later](#) as well as the latest version of the [JIRA FishEye Plugin](#). This is because application links now provide all of the functionality previously available with project links. However, project links are retained for the following reasons:

- Setting up project links provides a way to restrict the scope of JIRA searches to a specific FishEye repository.
- Legacy configurations can continue to use project links without any need for changes.
- Third-party plugins may continue to rely on project links for their functionality.

See [Configuring Project Links across Applications](#).

Notes

- **JIRA requires FishEye to manually refresh repository cache when repository changes are made** — When a repository is removed, or when there has been any change in FishEye repositories, JIRA does not update the FishEye repository list cache automatically. You must manually refresh the repository list

cache. This is done in JIRA: 'Administration' > 'FishEye Configuration' > 'Refresh Cache' link (next to 'Repository List Cache')

- **If FishEye/Crucible will be running on the same machine as JIRA** (already installed), you need to ensure that the URL paths are different for FishEye/Crucible and JIRA. Change the default FishEye/Crucible path as follows:
 - Edit the `config.xml` in your FishEye/Crucible data directory.
 - Add the context attribute to the `web-server` element:

```
<web-server context="/fisheye">
```

User Management Limitations and Recommendations

This page describes the optimal configurations and limitations that apply when you are connecting FishEye to JIRA for user management.

On this page:

- [Recommendations for Connecting to JIRA for User Management](#)
 - [Single Sign-On Across Multiple Applications is Not Supported](#)
 - [Custom Application Connectors are Not Supported](#)
 - [Custom Directories are Not Supported](#)
 - [Optimal Number of Users and Applications](#)
 - [Recommendations](#)

Recommendations for Connecting to JIRA for User Management

Please consider the following limitations and recommendations when connecting to a JIRA server for user management.

Single Sign-On Across Multiple Applications is Not Supported

When you connect to JIRA for user management, you will not have single sign-on across the applications connected in this way. JIRA, when acting as a directory manager, does not support SSO.

Custom Application Connectors are Not Supported

JIRA, Confluence, FishEye, Crucible and Bamboo can connect to a JIRA server for user management. Custom application connectors will need to use the new REST API.

Custom Directories are Not Supported

Earlier versions of JIRA supported OSUser Providers. It was therefore possible write a special provider to obtain user information from any external user directory. This is no longer the case.

Optimal Number of Users and Applications

Please consider the following limitations when connecting to a JIRA server for user management:

- Maximum 500 users.
- Maximum 5 connected applications.

Recommendations

Your environment	Recommendation

If all the following are true: • You have fewer than 500 users. • You want to share user and group management across just a few applications, such as one JIRA server and one Confluence server, or two JIRA servers. • You do not need single sign-on (SSO) between JIRA and Confluence, or between two JIRA servers. • You do not have custom application connectors. Or, if you do have them, you are happy to convert them to use the new REST API. • You are happy to shut down all your servers when you need to upgrade JIRA.	Your environment meets the optimal requirements for using JIRA for user management.
If one or more of the following are true: • You have more than 500 users. • You want to share user and group management across more than 5 applications. • You need single sign-on (SSO) across multiple applications. • You have custom applications integrated via the Crowd SOAP API, and you cannot convert them to use the new REST API. • You are not happy to shut down all your servers when you need to upgrade JIRA.	We recommend that you install Atlassian Crowd for user management and SSO.
If you are considering creating a custom directory connector to define your own storage for users and groups...	Please see if one of the following solutions will work for you: • If you have written a custom provider to support a specific LDAP schema, please check the supported LDAP schemas to see if you can use one of them instead. • If you have written a custom provider to support nested groups, please consider enabling nested groups in the supported directory connectors instead. • If you have written a custom provider to connect to your own database, please consider loading the data into the application's database instead. • If you need to keep the custom directory connection, please consider whether Atlassian Crowd meets your requirements. See the documentation on Creating a Custom Directory Connector.

Related Topics

[JIRA Integration in FishEye](#)

Configuring Application Links

An application link is a trust relationship between two applications. Linking two applications allows you to share information and to access one application's functions from within the other.

For example, you may want to set up a trust relationship between FishEye and two JIRA servers.

Configure Application Links 							 Add Application Link	
Name	Application	Application URL	Incoming Authentication 	Outgoing Authentication 	Primary	Actions		
 JDOG	JIRA	https://jdog.atlassian.com/secure/Dashboard.jspa	Trusted Applications	none		Configure Delete Make Primary		
 Your Company JIRA	JIRA	http://localhost:8080	Trusted Applications	Trusted Applications		Configure Delete		

Screenshot above: Application links to two JIRA servers

Related Topics

- [Adding an Application Link](#)
- [Configuring Authentication for an Application Link](#)
- [Deleting an Application Link](#)
- [Editing an Application Link](#)
- [Making an application link the primary link](#)
- [Relocating an Application Link](#)
- [Upgrading an Application Link](#)
- [Configuring Project Links across Applications](#)
- [Trusted Applications](#)

Adding an Application Link

This page describes how to add a new application link in FishEye. The process for adding an application link is different depending on whether or not the application you are linking FishEye to supports Atlassian's Application Links.

If you are linking FishEye to an application that does not have Application Links, you will need to do additional configuration in that application. This is because Application Links in FishEye will not be able to automatically configure authentication in your remote application.

On this page:

- [Before You Begin](#)
- [Adding an Application Link to an Application that Supports Application Links](#)
- [Adding an Application Link to an Application that Does Not Support Application Links](#)

Before You Begin

Check the following settings. This is required for synchronisation to work correctly:

- Make sure that the base URL is set correctly in FishEye. See [Configuring the FishEye web server](#).
- Make sure that the base URL is set correctly in the application which you intend to link to. See the appropriate instructions for [JIRA](#), [Crucible](#), [Bamboo](#), [Confluence](#).

Adding an Application Link to an Application that Supports Application Links

To link to an application that supports Application Links:

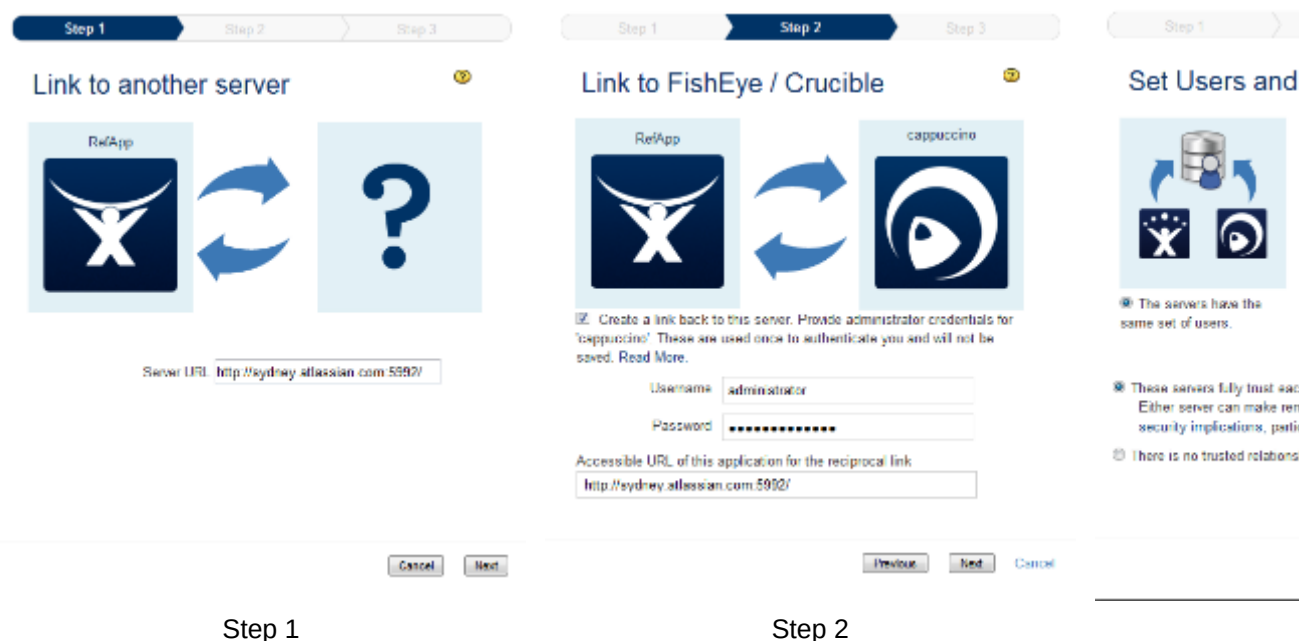
1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click '**Add Application Link**'. Step 1 of the link wizard will appear.
3. Enter the **server URL** of the application that you want to link to (the 'remote application').
4. Click the '**Next**' button. Step 2 of the link wizard will appear.
5. Enter the following information:
 - '**Create a link back to this server**' – Tick this check box if you want to create a two-way link between the remote application and your application. If you want to do this, you will need to enter

the username and password of an administrator for 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.
6. Click the **'Next'** button. Step 3 of the link wizard will appear.
 7. Enter the information required to configure authentication for your application link:
 - **'The servers have the same set of users'** or **'The servers have different sets of users'** – Select one of these options depending on how you manage users between the two applications.
 - **'These servers fully trust each other'** – Tick this check box if you know that the code in both applications will behave itself at all times and are sure each application will maintain the security of its private key.

For more information about configuring authentication, see [Configuring Authentication for an Application Link](#).
 8. Click the **'Create'** button to create the application link.



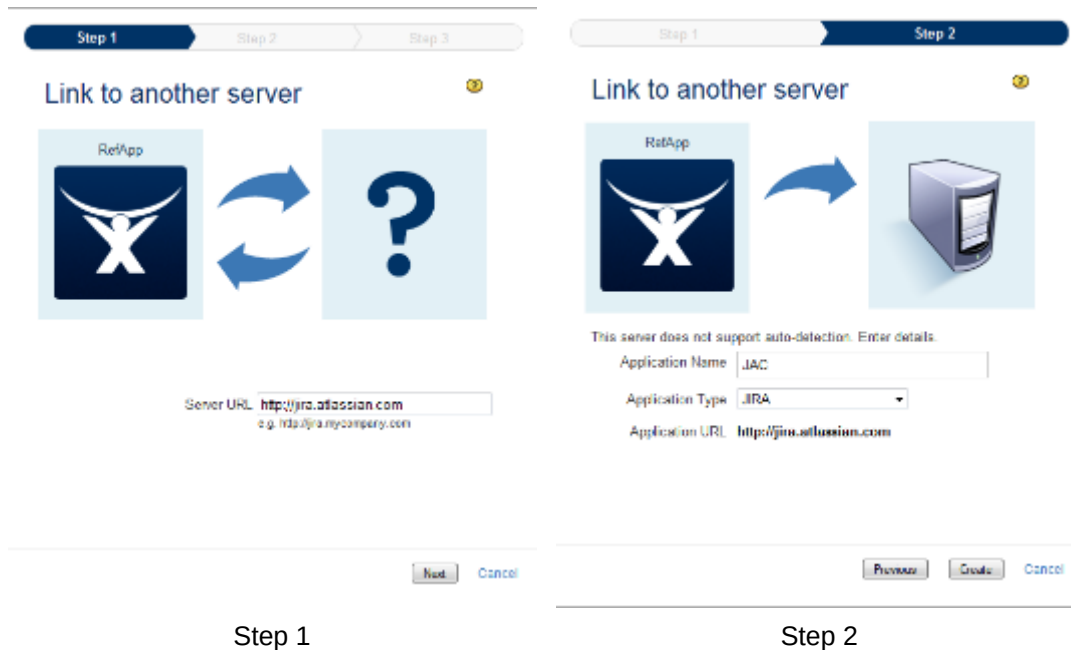
Screenshots above: Adding an application link to an application that supports Application Links (click to view full-sized images)

Adding an Application Link to an Application that Does Not Support Application Links

To link to an application that does not support Application Links:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click **'Add Application Link'**. Step 1 of the 'Link to another server' dialogue will be displayed.
3. Enter the server URL of the application that you want to link to, in the **'Server URL'** field. Click the **'Next'** button. Step 2 of the 'Link to another server' dialogue will be displayed.
4. Fill out the fields, as follows:
 - **'Application Name'** — Enter the name by which this remote application will be referred to, in your application.
 - **'Application Type'** — Select the type of application that you are linking to: Bamboo, FishEye/Crucible, JIRA, Confluence, Subversion.
 - **'Application URL'** — This will be set to the server URL you entered in the previous step and will not be editable.

5. Click the **Create** button to create the application link. The 'Configure Application Links' page will be displayed, listing all of the application links that have currently been set up for your application including the one you just added.
6. Configure the desired authentication type (Trusted Applications, OAuth, basic HTTP, none) for your new application link. See [Configuring Authentication for an Application Link](#).
7. In your application that does not support Application Links, configure the same type of authentication that you configured for your application link's *outgoing* authentication (in the previous step). For example, if you configured outgoing Trusted Applications authentication in your Application-Links-enabled application, you also need log into your non-Application-Links application and manually configure Trusted Applications (see the relevant administrator's documentation for the application).



Screenshots above: Adding an application link to an application that does not support Application Links (click to view full-sized images)

Related Topics

[Making an application link the primary link](#)
[Configuring Authentication for an Application Link](#)
[Configuring Project Links across Applications](#)

Configuring Authentication for an Application Link

When you configure authentication for an application link, you are defining the level of trust between FishEye and the application that it is linked to.

On this page:

- [Choosing Authentication for an Application Link](#)
- [Security Implications for each Authentication Type](#)
- [About Primary Authentication Types](#)
- [About Impersonating and Non-Impersonating Authentication Types](#)
- [Incoming and Outgoing Authentication](#)

Choosing Authentication for an Application Link

The level of authentication that you should configure for your application link depends on a number of factors.

- Do the two applications trust each other? In other words, are you sure that the code in the application will behave itself at all times and that the application will maintain the security of its private key?
- Do the two applications share the same user base?

- Do you have administrative access to the application you are linking to?

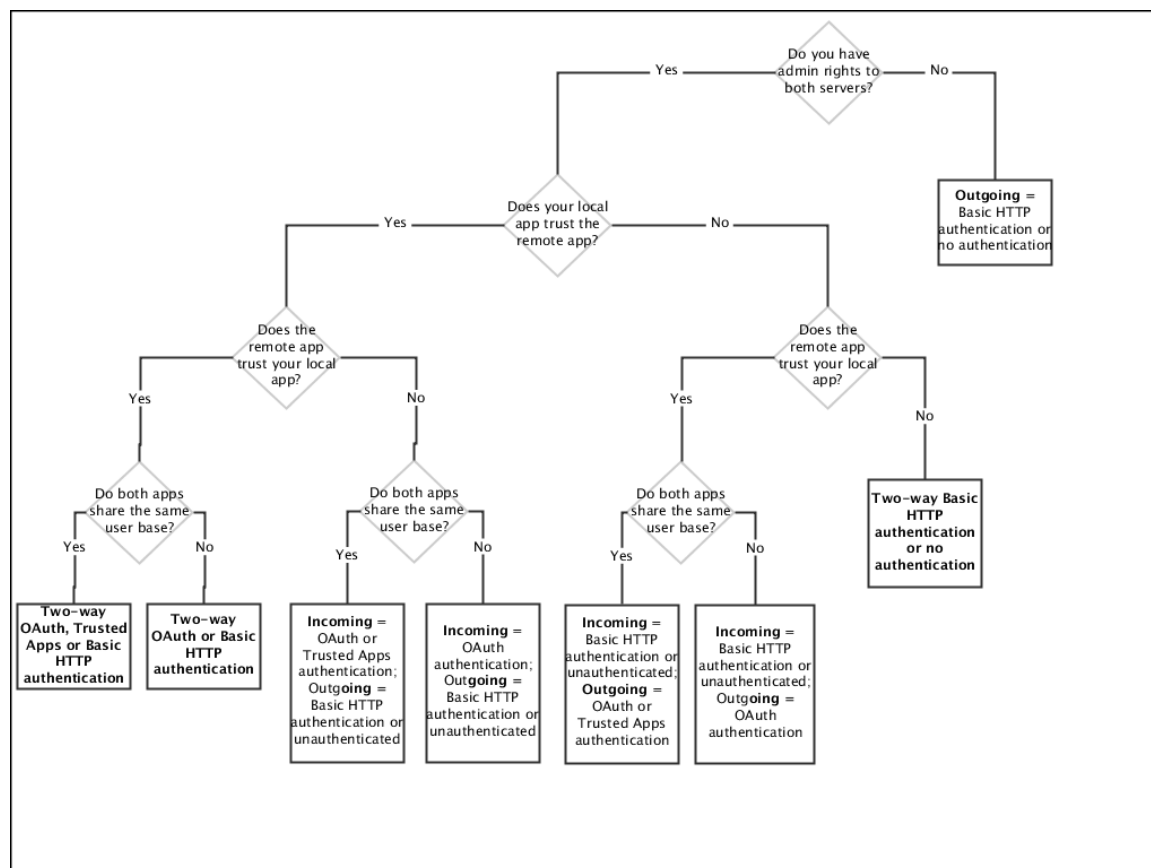
Common scenarios include:

- If the two applications you are linking **trust each other** and **share the same user base**, configure **two-way authentication using Trusted Applications** for both incoming and outgoing authentication. For example, you may link your internal FishEye server to an internal JIRA server.
- If the two applications you are linking **trust each other** but **do not share the same user base**, configure **two-way authentication using OAuth** for both incoming and outgoing authentication. For example, you may link your internal FishEye server to an external (customer-facing) JIRA server.
- If you **do not have administrative rights to the application that you are linking to** (for example, linking to a public FishEye server), configure a **one-way outgoing link** authenticated using **basic HTTP authentication** or **do not configure any authentication** for the link. For example, you may link your external FishEye server to a partner organisation's FishEye server. An unauthenticated link will still allow the local application to render hyperlinks to the remote application or query anonymously-accessible APIs.

The flowchart below provides a guide to what authentication you should configure for your application link.

Read the following topics for information on how to configure authentication for an application link:

- [Configuring Basic HTTP Authentication for an Application Link](#)
- [Configuring OAuth Authentication for an Application Link](#)
- [Configuring Trusted Applications Authentication for an Application Link](#)



Flowchart above: Determining what authentication to configure for an Application Link

Security Implications for each Authentication Type

If you configure **Trusted Applications authentication** for your application (meaning that your servers have the same set of users and they fully trust each other) 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.

If you configure **OAuth authentication** for your application (meaning that your servers have different sets of users and they fully trust each other) please be aware of the following security implications:

- Adding an OAuth consumer requires the transmission of sensitive data. To prevent '[man-in-the-middle attacks](#)', it is recommended that you **use SSL** for your applications while configuring OAuth authentication.
- Do not link to an application using OAuth authentication, unless you **trust all code in the application** to behave itself at all times. OAuth consumers are a potential security risk to the applications that they are linked to.

Step 1

Step 2

Step 3

Set Users and Trust



☒ The servers have the same set of users.



☐ The servers have different sets of users.

☒ These servers fully trust each other.
Either server can make remote requests as any user. [Read more on the security implications](#), particularly if you have public signup enabled.

☐ There is no trusted relationship between these servers.

Previous

Create

Cancel

Screenshot above: Configuring authentication during application link setup

About Primary Authentication Types

You can configure multiple authentication types for each application link. When a feature makes a request using an Application Link, it will use one of the configured authentication types. If more than one authentication type is configured, it will by default use the authentication type that is marked as the primary authentication type. The

default authentication type is indicated by the green tick  next to the authentication type on the list application link screen.

You **cannot** configure which authentication type is the primary authentication type. The primary authentication type is determined automatically by Application Links and depends on a weight defined by each authentication type method. However, every feature that uses Application Links can also choose to use a specific authentication type and might not use the default primary authentication type.

About Impersonating and Non-Impersonating Authentication Types

Applications Links allows you to configure 'impersonating' and 'non-impersonating' authentication types:

- **Impersonating authentication types** make requests on behalf of the user who is currently logged in. People will see only the information that they have permission to see. This includes [OAuth](#) and [Trusted Applications](#) authentication.
- **Non-impersonating authentication types** always use a pre-configured user when making a request. Everyone logged into the system will see the same information. This includes [basic HTTP authentication](#).

Incoming and Outgoing Authentication

When you configure authentication for an application link, you are defining the level of trust between the two linked servers. When configuring a link from one application to another, you can set up:

- **Incoming authentication** (authentication of requests coming from a linked application into this application).
- **Outgoing authentication** (authentication of requests sent from this application to a linked application).

Configuring Basic HTTP Authentication for an Application Link

This page describes how to configure **Basic HTTP authentication** for incoming and/or outgoing authentication for an application link.

Basic HTTP authentication allows FishEye to provide a specific set of user credentials to a remote application and vice versa. Once authenticated, one application can access specified functions on the other application on behalf of that user. For example, if you supply the credentials of a FishEye administrator on your FishEye server to a remote application, the remote application will be able to access all functions on your FishEye server that the FishEye administrator can access.

This method of authentication relies on the connection between FishEye and the remote application being secure. We recommend that you use [Trusted Applications authentication](#) or [OAuth authentication](#) for your application link instead, if possible.

On this page:

- [Before You Begin](#)
- [Configuring Basic HTTP Authentication for Outgoing Authentication](#)
- [Configuring Basic HTTP Authentication for Incoming Authentication](#)

Before You Begin

- The instructions assume that **both of the applications that you are linking have the Application Links plugin installed**. If the remote application that you are linking to supports Basic HTTP authentication, but does not have the Application Links plugin installed, you will need to configure Basic HTTP authentication from within the remote application. See the relevant administrator's documentation for the application. This is in addition to configuring the outgoing/incoming authentication for the application link described below.
- You must be a FishEye administrator to configure Basic HTTP authentication for an application link.

Configuring Basic HTTP Authentication for Outgoing Authentication

Configuring **outgoing basic http authentication** will allow FishEye to trust a remote application. This allows the

remote application to access specified functions in FishEye.

To configure basic http authentication for an outgoing application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure authentication for.
3. Click the '**Outgoing Authentication**' tab. The outgoing authentication page will be displayed.
4. Click the '**Basic Access**' tab.
5. Click the '**Configure**' button and enter the credentials (username and password) that the remote application will use to log into your application .
6. Click the '**Apply**' button to save your changes.

Configuring Basic HTTP Authentication for Incoming Authentication

Configuring **incoming basic http authentication** will allow the remote application that you are linking to, to trust FishEye. FishEye will be able to access specified functions on the remote application.

To configure basic http authentication for an incoming application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure authentication for.
3. Click the '**Incoming Authentication**' tab. The incoming authentication page will be displayed.
4. Click the '**Basic Access**' tab.
5. Click the '**Configure**' button and enter the credentials (username and password) that the your application will use to log in to the remote application.
6. Click the '**Apply**' button to save your changes.

Related Topics

[Configuring OAuth Authentication for an Application Link](#)

[Configuring Trusted Applications Authentication for an Application Link](#)

Configuring OAuth Authentication for an Application Link

This page describes how to configure **OAuth** for outgoing authentication and/or incoming authentication for an application link.

[OAuth](#) is a protocol that allows a web application to share data/resources with any other OAuth-compliant application. These applications could be another web application (such as a JIRA site or an [iGoogle](#) home page), a desktop application or a mobile device application, provided that they are accessible from within your network or available on the Internet.

A typical scenario is to set up an application link between two applications which trust each other, have the Application Links plugin installed, but do not share the same set of users. In this case, you would configure OAuth for both outgoing and incoming authentication.

Key OAuth Terminology:

- **Service provider** – An application that shares ('provides') its resources.
- **Consumer** – An application that accesses ('consumes') a service provider's resources.
- **User** – An individual who has an account with the service provider.

On this page:

- [Before You Begin](#)
- [Configuring OAuth for Outgoing Authentication](#)
- [Configuring OAuth for Incoming Authentication](#)

Before You Begin

- Adding an OAuth consumer requires the transmission of sensitive data. To prevent '[man-in-the-middle](#)' [attacks](#), it is recommended that you **use SSL** for your applications while configuring OAuth authentication.
- Do not link to an application using OAuth authentication, unless you **trust all code in the application** to behave itself at all times. OAuth consumers are a potential security risk to the applications that they are linked to.
- The instructions assume that **both of the applications that you are linking have the Application Links plugin installed**. If the remote application supports OAuth but does not have the Application Links plugin installed, you will need to configure OAuth from within the remote application (see the administrator's documentation for the application) in addition to configuring the outgoing/incoming authentication for the application link as described below.
- You must be a FishEye administrator to configure OAuth authentication for an application link.

Configuring OAuth for Outgoing Authentication

Configuring **outgoing OAuth authentication** will allow FishEye to access data in a remote application on behalf of a user. In other words, FishEye can access specified functions in the remote application.

To configure OAuth authentication for an outgoing application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure OAuth for.
3. Click the '**Outgoing Authentication**' tab. The outgoing authentication page will be displayed.
4. Click the '**OAuth**' tab.
5. If you are not currently logged in to the remote application (or you logged in to the remote application under a variant of the application's hostname, such as the IP address), a login dialogue will display.
 - Enter the '**Username**' and '**Password**' for the remote server, not your local server, and click the '**Login**' button. The remote server needs to learn the identity of your local server for the OAuth protocol to work and your admin credentials are used to store your local server's public key on the remote server. If you are already logged into your remote server, then the appropriate changes can be made without having to log in again.
6. Click the '**Enable**' button to enable OAuth authentication for the outgoing link. Your application will be automatically set up to be the 'consumer' and the remote application as a 'service provider'.

Configuring OAuth for Incoming Authentication

Configuring **incoming OAuth authentication** will allow the remote application to access data in FishEye.

To configure OAuth authentication for an incoming application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure OAuth for.
3. Click the '**Incoming Authentication**' tab. The incoming authentication page will be displayed.
4. Click the '**OAuth**' tab.

5. Click the **'Enable'** button to enable OAuth authentication for the incoming link. The remote application will be automatically set up to be the 'consumer' and your local application as a 'service provider'.

Related Topics

[Configuring Basic HTTP Authentication for an Application Link](#)

[Configuring Trusted Applications Authentication for an Application Link](#)

Configuring Trusted Applications Authentication for an Application Link

This page describes how to configure **Trusted Applications** for outgoing and/or incoming authentication for an application link.

Trusted Applications authentication allows one application to gain access to specified functions within another application on behalf of any user, without the user having to log in to the second application.

A typical scenario is to set up an application link between two applications which trust each other, have the same set of users and have the application links plugin installed. In this case, you would configure Trusted Applications for both outgoing and incoming authentication.

On this page:

- [Before You Begin](#)
- [Configuring Trusted Applications for Outgoing Authentication](#)
- [Configuring Trusted Applications for Incoming Authentication](#)

Before You Begin

- 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.
- The instructions below assume that **both of the applications that you are linking have the Application Links plugin installed**. If the remote application supports Trusted Applications but does not have the Application Links plugin installed, you will need to configure Trusted Applications from within the remote application (see the relevant administrator's documentation for the application) in addition to configuring the outgoing/incoming authentication for the application link as described below.
- You must be a FishEye administrator to configure Trusted Applications authentication for an application link.

Configuring Trusted Applications for Outgoing Authentication

Configuring **outgoing Trusted Applications authentication** will allow the remote application to trust FishEye. In other words, FishEye will be able to access specified functions and data on the remote application.

To configure Trusted Applications authentication for an outgoing application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the **'Configure'** link next to the application link that you want to configure Trusted Applications authentication for.
3. Click the **'Outgoing Authentication'** tab. The outgoing authentication page will show, with the **'Trusted Applications'** tab displayed.
4. If you are not currently logged into the remote application (or you logged into the remote application under a variant of the application's hostname, e.g. the IP address), a login dialogue will display.
 - Enter the **'Username'** and **'Password'** for the remote server, (not your local server), and click the **'Login'** button. You need to enter the credentials for the remote server, as the remote server needs

to be instructed to trust your local server for the Trusted Applications protocol to work. If you are already logged into your remote server, then the appropriate changes can be made without having to log in again.

5. Configure the settings for the Trusted Applications authentication:

- **'IP Patterns'** — Enter the IP addresses (IPv4 only) from which the remote application will accept requests (this effectively is the IP address your local server). You can specify wildcard matches by using an asterisk (*), e.g. '192.111.*.*' (note, you cannot use netmasks to specify network ranges). If you are entering multiple IP addresses, separate them with commas or spaces.

 *Please note, if you are setting up Trusted Applications between two applications that both have the Application Links plugin installed, you can leave this field blank (or explicitly use *.*.*.*). However, if your remote application does not have the Application Links plugin installed and you are configuring the IP Patterns in the remote application (not the Application Links plugin), **you must not leave this field blank nor use *.*.*.***. Failure to configure IP address restrictions in this scenario is a security vulnerability, allowing an unknown site to log into your site under a user's login ID.*

Consider the following scenarios, if you want to limit access by using this field:

- If your local application is using a proxy server, you need to add the proxy server's IP address to this field.
- If your local application is a clustered instance of Confluence, you need to configure the remote server to accept requests from each cluster node. If you do not set up each node appropriately, your Confluence users may not be able to view any information from the remote server. You can set this up by either specifying each individual IP address for each node of the cluster (e.g. 172.16.0.10, 172.16.0.11, 172.16.0.12), or specifying the IP address for the clustered Confluence instance using wildcards (e.g. 172.16.0.*).
- **'URL Patterns'** — Enter the URLs in the remote application that your local application will be allowed to access. Each URL corresponds to a particular application function. Enter one URL per line, as follows:
 - If your remote application is JIRA, enter the following URL Patterns: /plugins/servlet/streams, /sr/jira.issueviews:searchrequest, /secure/RunPortlet, /rest, /rpc/soap
 - If your remote application is Confluence, enter the following URL Patterns: /plugins/servlet/streams, /plugins/servlet/applinks/whoami
- **'Certificate Timeout (ms)'** — Enter the certificate timeout. The default is 10 seconds. The certificate timeout is used to prevent replay attacks. For example, if a Trusted Applications request is intercepted and (maliciously) re-sent, the application will be able to check when the request was first sent. If the second request is sent more than 10 seconds (or whatever the certificate timeout is set to) after the initial request, it will be rejected. Please note, you should not have to change the default value of this field for most application links. Note that the certificate timeout relies on the clocks on both servers being synchronised.

6. Click the **'Apply'** button to save your changes.

Configuring Trusted Applications for Incoming Authentication

Configuring **incoming Trusted Applications authentication** will allow FishEye to trust the remote application. The remote application will be able to access specified functions and data on FishEye.

To configure Trusted Applications authentication for an incoming application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the **'Configure'** link next to the application link that you want to configure Trusted Applications authentication for.
3. Click the **'Incoming Authentication'** tab. The incoming authentication page will show, with the **'Trusted Applications'** tab displayed.
4. The tab will show whether Trusted Applications is currently enabled or not. Use the **'Modify'** or **'Configure'**

e' button to configure Trusted Applications. The Trusted Applications configuration settings will be displayed:

- **'IP Patterns'** — Enter the IP addresses (IPv4 only) from which our application will accept requests. You can specify wildcard matches by using an asterisk (*), e.g. '192.111.*.*' (note, you cannot use netmasks to specify network ranges). If you are entering multiple IP addresses, separate them with commas or spaces.

 *Please note, if you are setting up Trusted Applications between two applications that both have the Application Links plugin installed, you can leave this field blank (or explicitly use *.*.*.*). However, if your remote application does not have the Application Links plugin installed and you are configuring the IP Patterns in the remote application (not the Application Links plugin), **you must not leave this field blank nor use *.*.*.***. Failure to configure IP address restrictions in this scenario is a security vulnerability, allowing an unknown site to log into your site under a user's login ID.*

Consider the following scenarios, if you want to limit access by using this field:

- If the remote application is using a proxy server, you need to add the proxy server's IP address to this field.
- If the remote application is a clustered instance of Confluence, you need to accept requests from each cluster node. If you do not specify each node's address, Confluence users may not be able to view any data from your application. You can set this up by either specifying each individual IP address for each node of the cluster (e.g. 172.16.0.10, 172.16.0.11, 172.16.0.12), or specifying the IP address for your clustered Confluence instance using wildcards (e.g. 172.16.0.*).
- **'URL Patterns'** — Enter the local URLs that the remote application will be allowed to access. Each URL corresponds to a particular application function. Enter one URL per line, as follows:
 - If your local application is JIRA, enter the following URL Patterns — /plugins/servlet/streams, /sr/jira.issueviews:searchrequest, /secure/RunPortlet, /rest, /rpc/soap
 - If your local application is Confluence, enter the following URL Patterns — /plugins/servlet/streams, /plugins/servlet/applinks/whoami
- **'Certificate Timeout (ms)'** — Enter the certificate timeout. The default is 10 seconds. The certificate timeout is used to prevent replay attacks. For example, if a Trusted Applications request is intercepted and (maliciously) re-sent, the application will be able to check when the request was first sent. If the second request is sent more than 10 seconds (or whatever the certificate timeout is set to) after the initial request, it will be rejected. Please note, you should not have to change the default value of this field for most application links. Note that the certificate timeout relies on the clocks on both servers being synchronised.

5. Click the **'Apply'** button to save your changes.

Related Topics

[Configuring Basic HTTP Authentication for an Application Link](#)

[Configuring OAuth Authentication for an Application Link](#)

Deleting an Application Link

Deleting an application link stops the two applications from sharing information. You will no longer be able to make requests from one application to the other. This means that certain integration features may not work.

If you have set up application links to multiple servers of the same application type, for example you have linked FishEye to multiple JIRA servers, deleting the [primary link](#) will mean that another of the links will be made the primary link.

Deleting an application link will also [delete all project links](#) set up for that application link.

To delete an application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the

administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.

2. Click the '**Delete**' link next to the application link that you want to delete. A confirmation screen will be displayed.
3. Click the '**Confirm**' button to delete the application link.

RELATED TOPICS

[Editing an Application Link](#)

[Relocating an Application Link](#)

Editing an Application Link

You can change the details, such as the application name and display URL, for an existing application link.

To edit an application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to edit the details for. The application details for the application link will be displayed.
3. Update the application details as desired. Please note, you cannot update the Application Type nor the Application URL.
 - '**Application Name**' — Update this field to change the display name for the application that you are linking to.
 - '**Display URL**' — This URL is used when displaying links to the application in the browser. When creating the application link, you may have used a URL that is not accessible to other users, such as an internal IP address. If so, you can change the display URL to an address in a domain that is accessible to other users.
4. Click the '**Update**' button to save your changes.

Configure JAC

Application Details	Application Name JAC
Outgoing Authentication	Application Type JIRA
Incoming Authentication	Application URL http://jira.atlassian.com This is the URL used to connect to the application.
	Display URL http://jira.atlassian.com The display URL is used when rendering links to the application.

Screenshot above: Editing an application link

Related Topics

[Configuring Authentication for an Application Link](#)
[Making an application link the primary link](#)

Making an application link the primary link

If you have set up application links to more than one of the same application type, for example you have linked FishEye to two JIRA servers, then one of the servers will be marked as the 'primary' link. This means that any outgoing requests will be directed to the primary link's application.

For example, if you have linked a FishEye server to two JIRA servers with two-way authentication for both links, you can nominate an application link to one of the JIRA servers as the primary link. Every time FishEye requests JIRA information, it will request it from the primary link's JIRA server. Both JIRA servers can still make requests of the FishEye server.

Please see [Configuring Project Links across Applications](#) for information on how primary project links also influence the information shared between servers.

On this page:**To make an application link the primary link:**

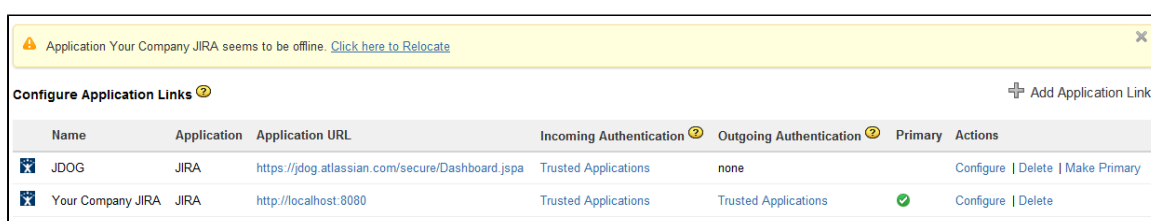
1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the **'Make Primary'** link next to the application link that you want to make the primary link. A '✓' symbol will display in the 'Primary' column next to the application link.
 ⓘ The 'Primary' column and 'Make Primary' link will only display if you have set up application links to more than one of the same application type, e.g. you have linked your application to two JIRA servers.

Relocating an Application Link

This page describes how to change the location of an application link. You will need to relocate an application link if the target application has moved to a new address.

To relocate an application link:

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. If the remote application for an application link cannot be reached by your application, the **'List Application Links'** page will display a warning message (see 'Relocate Link - Warning Message' screenshot below).
3. If your remote application has been moved to a different address (rather than just being offline temporarily), click the **'Relocate'** link in the warning message (see 'Relocate Link - Updating URL' screenshot below).
4. Enter the new URL for the remote application of your application link and click **'Relocate'**.
5. You will need to confirm the relocation, if the new URL cannot be contacted. Otherwise, the application link will be updated.



Name	Application	Application URL	Incoming Authentication	Outgoing Authentication	Primary	Actions
JDOG	JIRA	https://jdog.atlassian.com/secure/Dashboard.jspa	Trusted Applications	none		Configure Delete Make Primary
Your Company JIRA	JIRA	http://localhost:8080	Trusted Applications	Trusted Applications	✓	Configure Delete

Screenshot above: Relocate link – The warning message



Screenshot above: Relocate link – Updating the URL

Related Topics

[Making an application link the primary link](#)

Upgrading an Application Link

This page describes how to upgrade an existing application link. You may want to upgrade an application link in either of the two situations below:

- Your local FishEye application has been upgraded from a version that does not include Application Links to a version that does.
- Your remote application has been upgraded to a version that includes Application Links. For example, you had set up an application link to a JIRA 4.2 site (does not include Application Links) and have since upgraded to JIRA 4.3 (includes Application Links).

On this page:

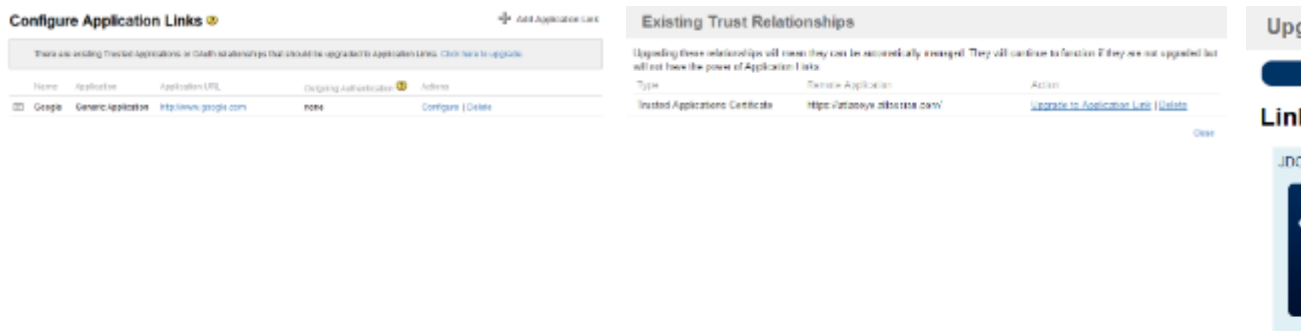
- [Upgrading an Application Link \(Local App Upgraded to Include Application Links\)](#)
- [Upgrading an Application Link \(Remote App Upgraded to Include Application Links\)](#)

Upgrading an Application Link (Local App Upgraded to Include Application Links)

When you upgrade from a FishEye version that does not include Application Links to a version that does, you will have the option of converting any Trusted Applications or OAuth links to Application Links. The advantage of converting your links to Application Links is that link configuration will be simplified in future.

To upgrade an application link when FishEye has been upgraded to include Application Links:

1. After your FishEye upgrade, navigate to the administration console.
2. Click '**Application Links**'. The 'Configure Application Links' screen will show the following message: "There are existing Trusted Applications or OAuth relationships that should be upgraded to Application Links. **Click here to upgrade.**"
3. Click the '**Click here to upgrade**' link. The 'Existing Trust Relationships' screen will appear, showing all Trusted Applications and OAuth relationships that can be upgraded to Application Links.
4. Click the '**Upgrade to Application Link**' link next to the desired trust relationship. The 'Upgrade to Application Link' wizard will appear.
5. Complete the wizard. The process will be similar to adding a new link (described on [Adding an Application Link](#)), except that most fields should be pre-filled.



Step 1

Step 2

Screenshots above: Upgrading an application link for your local application

Upgrading an Application Link (Remote App Upgraded to Include Application Links)

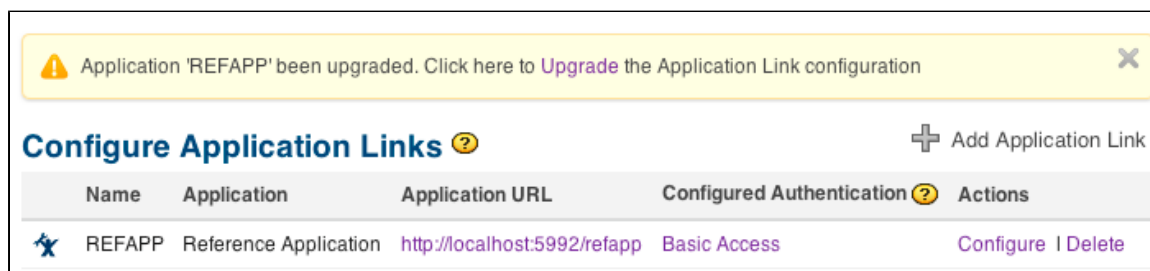
When an application link is created between a version of FishEye that supports Application Links, and a remote application that does not (either a non-Atlassian product, or an older version of an Atlassian product that did not ship with Application Links), this link is configured to run in 'legacy mode'. While there is no distinguishable difference to a user, connection and configuration without Application Links is a little different. For example:

- Setting up OAuth requires manual configuration by the administrator. If your applications support Application Links, exchange of the consumer keys and public keys is done automatically.
- The Trusted Applications protocol (Atlassian-specific) will not be available for authentication.

If you upgrade your remote application to a version that does support Application Links, the application link will continue to work. Upgrading your link may simplify link configuration and make additional authentication protocols available (as mentioned above).

To upgrade an application link when your remote application has been upgraded to include Application Links:

1. After you have upgraded your remote application to a version that includes Application Links, go to the administration console of your local application. A warning will be displayed, requesting that you upgrade the link to full Application Links mode.
2. Click '**Upgrade**' in the warning message to start the upgrade wizard. Note the following:
 - You will be prompted to make your application link a reciprocal link. You will need to provide administrator credentials for your remote application, if you choose to do so.
 - If you make your application link a reciprocal link, you will also be able to make reciprocal links for your project links. For example, you may be able to link your JIRA project to a FishEye repository and also make a link from your FishEye repository back to the JIRA project.



Screenshot above: Upgrading an application link for remote application

Step 1 **Step 2** Step 3

Link to Reference Application ?

RefApp - erik

REFAPP

☒ Create a link back to this server. Provide administrator credentials for 'REFAPP'. These are used once to authenticate you and will not be saved. [Read More.](#)

Username

Password

☒ For each existing Charlie link, create an incoming link from the remote application.

Accessible URL of this application for the reciprocal link

Previous Next Cancel

Screenshot above: Upgrading an application link wizard

Related Topics

[Adding an Application Link](#)

[Configuring Authentication for an Application Link](#)

Configuring Project Links across Applications

We no longer recommend the use of project links with FishEye 2.9 and later. This is because application links now provide all of the functionality previously available with project links.

However, project links are retained for the following reasons:

- Setting up project links provides a way to restrict the scope of JIRA searches, which can provide performance benefits.
- Legacy configurations can continue to use project links without any need for changes.

- Third-party plugins may continue to rely on project links for their functionality.

On this page:

- [Adding a project link](#)
- [Deleting a project link](#)
- [Making a project link the primary link](#)

Related pages:

- [Adding an Application Link](#)

Adding a project link

In the following example, we'll create an **two-way project link** between a JIRA project (project key is 'MYPROJECT') and a FishEye repository (repository key is 'MYREPO'). We assume that you have already [added an application link](#).

To link a FishEye project to a project in another application:

1. Go to the FishEye administration console and find the repository that you want to link from.
2. Click the  icon and choose **Application Links**.
3. The instructions for adding a project link will vary depending on whether the target application has the Application Links functionality installed:
 - If the target application has Application Links:
 - a. Click **'Add Link'**. A dropdown menu will appear listing the applications you have already linked to.
 - b. In the dropdown menu, click the application that contains the project you want to link to. For example, if you want to link to a specific JIRA project, click the JIRA site that contains that project. If you want to link to a Confluence space, click the Confluence site that contains that space.
 - c. Click one of the options on the 'Authorization required' screen:
 - **'Authorize'** — Click this option if you want to grant your project authorised access to the target project. The target application will open in a new window, so that you can log in and authorise access.
 - **'Skip – your access is anonymous'** — Click this option if you only want to allow anonymous access to the target project.
 - d. In the **'Name or Key'** field, enter the name/key of the project in the remote application that you want to link to. For example, if you want to link to a JIRA project, enter the project key. If you want to link to a Confluence space, enter the space key.
 - e. Click the **'Create'** button to create the project link.
 - If the target application does not have Application Links:
 - a. Click **'Add Link'**. A dropdown menu will display listing the applications you have already linked to.
 - b. In the dropdown menu, click the application that contains the project you want to link to. For example, if you want to link to a specific JIRA project, click the JIRA site that contains that project. If you want to link to a Confluence space, click the Confluence site that contains that space.
 - c. In the **'Key'** field, enter the name/key of the project in the remote application that you want to link to. For example, if you want to link to a JIRA project, enter the project key. If you want to link to a Confluence space, enter the space key.
 - d. (optional) Enter the alias for the project in the **'Alias'** field. This is the display name for the project in your administration console.
 - e. Click the **'Create'** button to create the project link.

Deleting a project link

If you have set up multiple project links to the same application, for example you have linked a FishEye project to multiple JIRA projects, deleting the [primary link](#) will mean that another of the links will be made the primary link.

To delete a project link:

1. Go to the FishEye administration console and find the repository that you want to link from.
2. Click the  icon and choose **Application Links**.
3. Click **Delete** next to the link that you want to delete.
4. Click **Confirm** to delete the link.

Making a project link the primary link

If you have set up project links to more than one project in the same application, for example you have linked your FishEye project to two JIRA projects, then one of the project links will be marked as the primary link. All outgoing requests will be directed to the primary link.

For example, if you have a FishEye project that is linked to two JIRA projects, you can nominate the link to one of the JIRA projects as the primary link. Every time FishEye requests JIRA information, it will request it from the primary link's JIRA project. Both JIRA projects can still request information from the FishEye project.

To make a project link the primary link:

1. Go to the FishEye administration console and find the repository that you want to link from.
2. Click the  icon and choose **Application Links**.
3. Click the **'Make Primary'** link in the **'Action'** column for the project link that you want to make the primary link. A  symbol will display in the 'Primary' column next to the link.

Note: The 'Primary' column and 'Make Primary' link will appear only if you have set up multiple project links to the same application, for example you have linked a Confluence space to a number of JIRA projects.

Trusted Applications

A **'trusted application'** is an application that FishEye will allow to access specific functions in FishEye on behalf of any user, without the user logging in to FishEye. You can configure Trusted Applications authentication between FishEye, Crucible, JIRA (3.12 and later) and Confluence (2.7 and later).

Using the bundled Application Links plugin, you can configure application links with Trusted Applications authentication as well as other types of authentication.

For further instructions, please see [Configuring Application Links](#).

Related Topics

[Configuring Application Links](#)

[JIRA Integration in FishEye](#)

System Settings

- [FishEye folder layout](#)
- [Command-Line Options](#)
- [Configuring Web Hooks](#)
- [Enabling Smart Commits](#)
- [Configuring commit hooks](#)
- [Configuring ViewVC Compatibility](#)
- [Customising the Welcome Message](#)
- [Software Update Notifications](#)
- [Enabling Access Logging in FishEye](#)
- [Setting JVM System Properties](#)

- [FishEye and Crucible Cookies](#)
- [Configuring the FishEye web server](#)
- [Configuring indexing](#)
- [Optimising the Search Indexes on Restart](#)
- [Customising FishEye's look and feel](#)
- [Environment variables](#)
- [Configuring Outgoing Web Proxy Support for FishEye or Crucible](#)

FishEye folder layout

By default, FishEye will run a self-contained instance directory within the <FishEye home directory> – the directory where you unzip the package. The FishEye instance directory is where your FishEye data is stored.

For a production environment, 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.

You should create your FishEye instance directory, and then tell FishEye where you created it by setting a FISH EYE_INST environment variable. See [Installing FishEye on Windows](#) and [Installing FishEye on Linux and Mac](#) for information about setting the FISHEYE_INST environment variable.

Default FishEye folder layout

The default FishEye folder layout includes the FishEye instance data inside the <FishEye home directory>. This layout is not recommended for production environments.

<FishEye home directory>/config.xml	Configuration file.
<FishEye home directory>/var/	Directory under which FishEye stores its data.
<FishEye home directory>/var/data/	Persistent data.
<FishEye home directory>/var/cache/	Caches and indexes.
<FishEye home directory>/var/log/	Log files.
<FishEye home directory>/var/tmp/	Temporary files.
<FishEye home directory>/cache/	Caches and indexes. (and also in FISHEYE_HOME/var/cache)
<FishEye home directory>/bin/	Scripts for controlling FishEye.
<FishEye home directory>/lib/	FishEye's dependent libraries.
<FishEye home directory>/syntax/	Syntax highlighting definitions.
<FishEye home directory>/ ...	Remainder omitted for brevity.

Recommended FishEye folder layout

The FishEye folder layout recommended for production environments has the FishEye instance directory (as defined by `FISHEYE_INST`) in a completely separate location from the `<FishEye home directory>`.

<code>\$FISHEYE_INST/config.xml</code>	Configuration file.
<code>\$FISHEYE_INST/var/</code>	All persistent and most temporary data .
<code>\$FISHEYE_INST/cache/</code>	Caches and indexes (and also in <code>\$FISHEYE_INST/var/cache</code>)
<code>\$FISHEYE_INST/lib/</code>	Site-specific Java libraries (.jars) that FishEye should load on startup. (Do not copy the dependent <code><FishEye home directory>/lib/</code> files into here.)
<code>\$FISHEYE_INST/syntax/</code>	Site-specific syntax highlighting definitions.
<code>\$FISHEYE_INST/system.properties</code>	Used for setting system properties within the FishEye JVM, which may sometimes be useful for support purposes. (Note: the other way to set properties is using <code>FISHEYE_OPTS</code> , e.g.: <code>export FISHEYE_OPTS=-DpropertyName=value</code>)
<code><FishEye home directory>/lib/</code>	FishEye's dependent libraries.
<code><FishEye home directory>/syntax/</code>	FishEye bundled highlighting definitions.
<code><FishEye home directory>/bin/</code>	
<code><FishEye home directory>/ ...</code>	Remaining files are found under <code><FishEye home directory>/</code> .

On this page:

- [Default FishEye folder layout](#)
- [Recommended FishEye folder layout](#)

Command-Line Options

A FishEye instance can be managed using the `fishyeectl` script. Before running this script you need to ensure that you have set the [JAVA_HOME](#) environment variable, or that `java` is on the path.

Unix usage:

```
/FISHEYE_HOME/bin/fisheyeectl.sh command [options]
```

Windows usage:

```
\FISHEYE_HOME\bin\fishyeectl.bat command [options]
```

The *command* parameter can be one of `run`, `start` or `stop` (see below). You can also find convenience scripts for running each of these commands, such as `run.sh` or `run.bat`.

run

The `run` command starts FishEye. This command runs FishEye in the foreground. It does not fork a background process.

start

The `start` command has the same options as `run`, but starts FishEye in the background.

Windows: FishEye will be run in a separate `cmd.exe` window.

Unix: FishEye will be run with `nohup` and the console output will be redirected to `$FISHEYE_INST/var/log/fisheye.out`.

Options (for both `run` and `start`):

<code>--config path</code>	Load configuration from the file at <code>path</code> . Default is <code>\$FISHEYE_INST/config.xml</code> .
<code>--quiet</code>	Do not print anything to the console.
<code>--debug</code>	Print extra information to the debug log.
<code>--debug-perf</code>	Print performance-related information to the debug log.

The following options are currently available, but will be removed at a later date:

<code>--Xtab-width nchars</code>	Specifies the number of spaces to use to represent a tab character. The default is 8.
<code>--Xdisable-dirtree-empty-checks</code>	When rendering the directory tree on some pages, FishEye calculates if each directory subtree is empty. For massive repositories, this calculation can cause the page to take a long time to render. This option disables the calculation that determines emptiness.  If you start FishEye with this flag, then empty directories will no longer show up as grey (i.e. deleted) and the option to hide them will also disappear
<code>--Xdisable-content-indexing</code>	Disable the generation of a full-text index for file content. This prevents further indexing, but does not delete any existing full-text indexes. FishEye will not warn you if you specify this option but still try to do a content search. This option is useful if you do not use content search and you are finding FishEye is taking a long time to index your content.

--Xenable-git-content-hash-resolving-heuristic	When several git revisions have the same hash (for instance if changes have been reverted and then reapplied) just use the most recent revision. This may perform better, but may show incorrect revision numbers in File tab of the repository browser.

stop

The stop command stops a running FishEye instance.

Options:

--config path	Load configuration from the file at path. Default is \$FISHEYE_INST/config.xml.

fullscan

Usage:

```
fishyeectl fullscan [options] [repname ...]
```

The fullscan command requests a full scan of the given repositories, or all repositories if no repository name is given

Options:

--config path	Load configuration from the file at path. Default is \$FISHEYE_INST/config.xml.

rescan

Usage:

```
fishyeectl rescan [options] repname start end
```

Requests a rescan of the given repository between two specified revision ids.

Note: this operation is not supported by CVS repositories.

Options:

--config path	Load configuration from the file at path. Default is \$FISHEYE_INST/config.xml.

reindex

Usage:

```
fishyeectl reindex [options] [repname ...]
```

Requests a full reindex of the given repositories, or all repositories if no repository name is given.

Options:

<code>--config path</code>	Load configuration from the file at path. Default is \$FISHEYE_INST/config.xml.

scannow

The scannow command requests an incremental scan from the command line.

Usage:

```
fishyeectl scannow -s [repname ...]
```

Options:

<code>-s</code>	Run synchronously; i.e. the command will not terminate until the requested indexing operation is complete.

- You can set the poll period to '**Never**' in which case the repo will be viewable from the web UI but the repository will not be polled.
- Repositories may be polled in the demand using this code:

```
fishyeectl.sh scannow [repname ...]
```

This will perform a one-time indexing operation.

Note: scannow can also be used on a repository which has a poll period.

backup

Usage:

```
fishyeectl backup -f [filename]
```

The backup command creates a zip archive containing important FishEye configuration files.

Options:

<code>-f filename</code>	Store the backup.zip to filename. Default is \$FISHEYE_INST/backup/backup_yyyyddMMHHmmss.zip.

Configuring Web Hooks

Web Hooks allow you to post changeset information to a specific URL. For each commit indexed by FishEye, the changeset data will be sent in JSON format to all configured URLs using a POST request.

You can use this feature to build integration with applications such as issue trackers, continuous integration solutions or notification systems.

- [Adding Web Hooks](#)
- [Disabling Web Hooks](#)
- [Demonstration of Web Hooks](#)
- [Web Hook Data](#)
- [Sample Web Hook Scripts](#)
 - [Python](#)

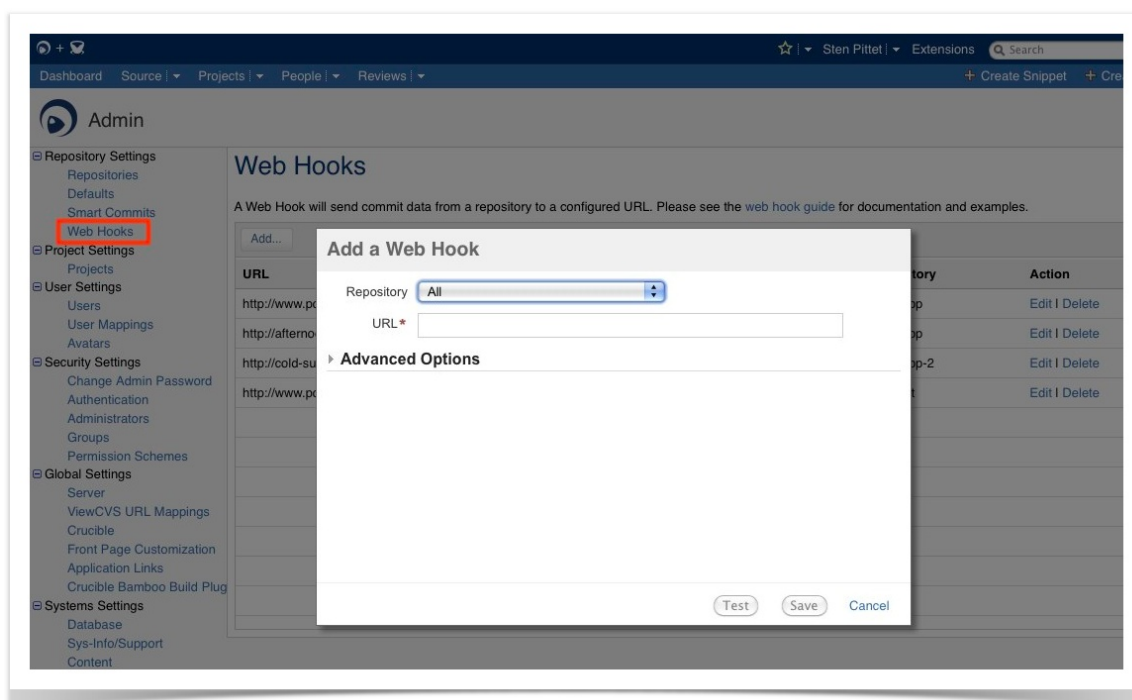
- [Ruby and Sinatra](#)

⊖ Be aware that Web Hooks can expose private commit information. For example, using a public service could expose your code. Any user with administrative privileges in FishEye can create a Web Hook for any repository.

Adding Web Hooks

Web Hooks can be added or edited in from the configuration page accessible from the Administration interface.

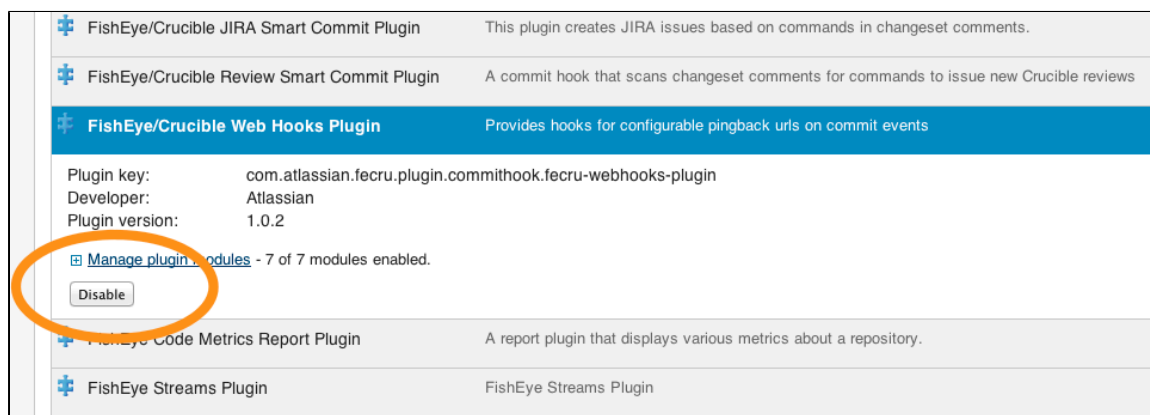
Click on the **Add** button to create a new Web Hook. Choose a repository and enter your URL. Click **Test** and **Save** to verify and store your configuration. Web Hooks also support basic authentication, which can be configured in the **Advanced Options**.



Disabling Web Hooks

To disable Web Hooks, go to the Administration interface > Plugins

Show **System Plugins** and navigate to the **FishEye/Crucible Web Hooks Plugin** and click on the **Disable** button.



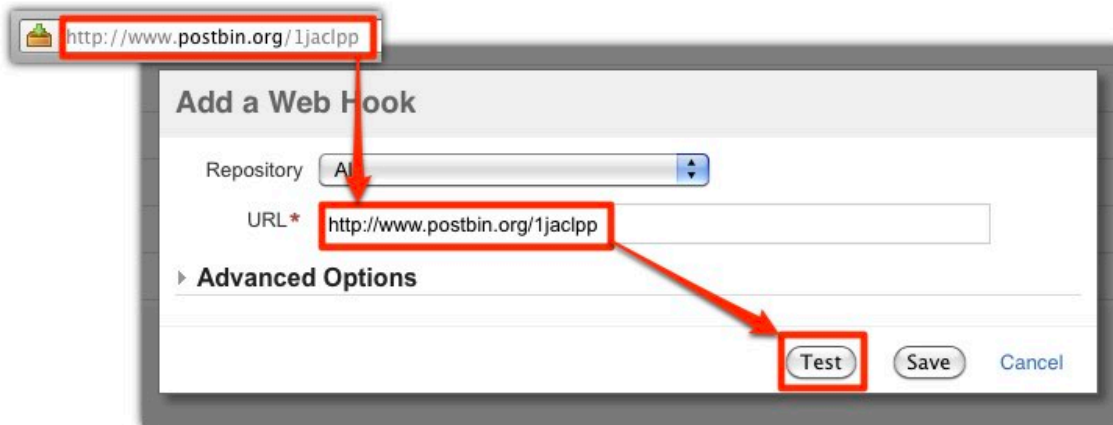
Demonstration of Web Hooks

[PostBin](http://www.postbin.org) is a service for debugging web hooks.

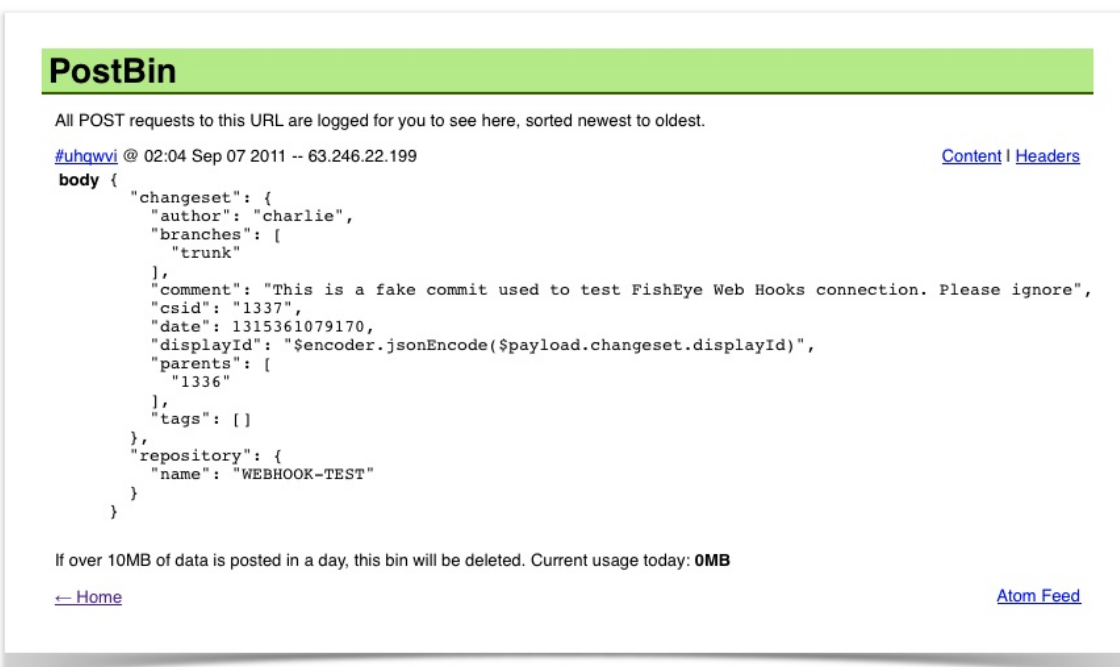
⊖ This is a public service and we do not recommend testing private repositories with public services as your code can be exposed.

Start by going to <http://www.postbin.org> and clicking on the *Make a PostBin* button.

Grab your PostBin URL and create a Web Hook with it using a test repository.



When you click on the *Test* button a dummy changeset will be sent to the PostBin URL. Refresh your PostBin page and you should see the data sent by FishEye.



This is a demonstration of the data sent by FishEye with a configured WebHook. The next step would be to write a service which captures this data and processes it.

Web Hook Data

Web Hooks will send changeset data in the following JSON representation:


```
{
  "repository": {
    "name" : <repository name>
  },
  "changeset" : {
    "csid" : <changeset id>,
    "displayId" : <changeset id>,
    "author" : <committer>,
    "comment" : <commit message>,
    "date" : <commit date>,
    "branches" : [
      <list of commit branches>
    ],
    "tags" : [
      <list of commit tags>
    ],
    "parents" : [
      <list of parents>
    ]
  }
}
```

Sample Web Hook Scripts

The following script examples show you how to set up Web Hooks in various languages.

Python

This example requires the [simplejson library](#) installed.

```

import simplejson as json
import pprint
from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer

class postRequestHandler(BaseHTTPRequestHandler):
    def do_POST(self):
        content_length = self.headers.getheader('content-length')
        if content_length:
            content_length = int(content_length)
            if self.path.endswith('webhook'):
                self.send_response(200)
                self.end_headers()
                body = self.rfile.read(content_length)
                post_data = json.loads(body)
                print 'Commit added with comment: %s' %
post_data["changeset"]["comment"]

            else:
                self.send_error(404, 'File Not Found %s' % self.path)

        return

    def run():
        server = HTTPServer(('', 8081), postRequestHandler)
        print 'Server started'
        server.serve_forever()

if __name__ == '__main__':
    run()

```

Ruby and Sinatra

The example below would display the changeset information in the logs of the [Sinatra](#) server.

```

require 'rubygems'
require 'sinatra'
require 'json'

post '/' do
  changeset = JSON.parse(request.body.read)
  puts changeset.inspect
end

```

Enabling Smart Commits

-  Because Git and Mercurial allow commits to be attributed to users other than those pushing changes to the repository, if your FishEye instance has any DVCS repositories then Smart Commits will be disabled. This is a security measure to prevent privilege escalation. This only occurs when the feature is first initialised, and will not be disabled if new Git or Mercurial repositories are added afterwards.

Requirements

Smart commits making use of 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](#).

Note that Smart Commits also require the following:

- An Application Link must be configured between FishEye/Crucible and JIRA. See [Adding an Application Link](#).
- A project link must be configured between FishEye/Crucible and JIRA. See [Adding Project Links between Applications](#).

On this page:

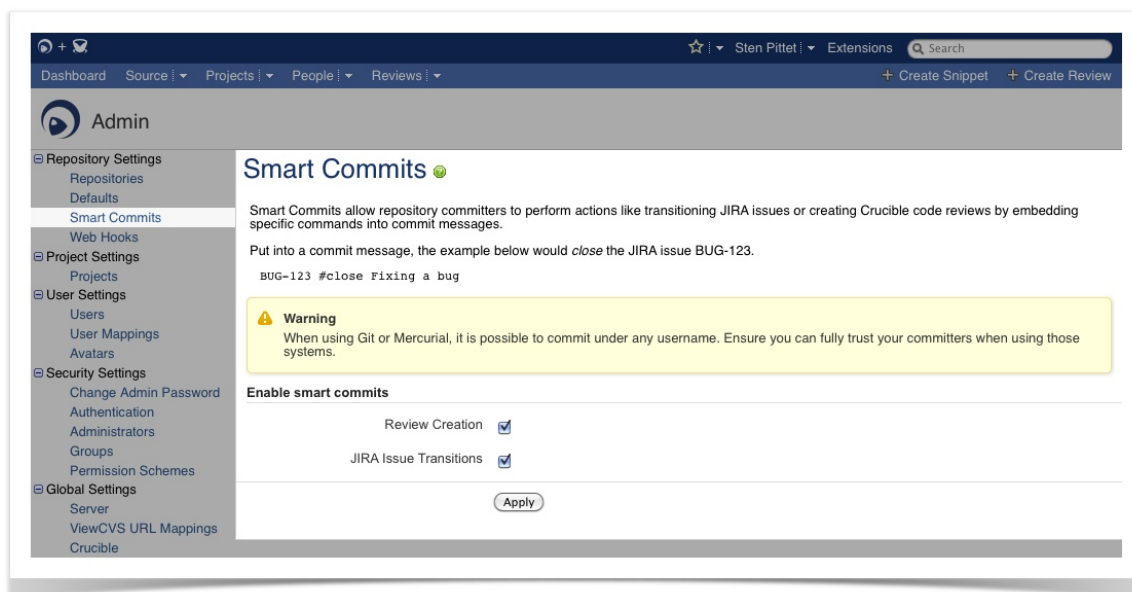
- [Requirements](#)
- [Enabling Smart Commits](#)
- [Disabling Smart Commits](#)
- [FAQ](#)

Related pages:

- [Using smart commits](#)
- [Configuring Web Hooks](#)
- [Transitioning JIRA issues](#)
- [Write your own Smart Commit](#)

Enabling Smart Commits

To enable smart commits, open the FishEye **Administration interface** and go to the **Smart Commits** configuration page.



Select the Smart Commits that you would like to enable.

Disabling Smart Commits

To disable Smart Commits, deselect the checkboxes for all types of Smart Commit.

FAQ

Q. The Smart Commits administration page is reporting an error, saying "No plugins installed"

A. It is possible that the smart commit plugins are disabled in the **Plugins** section of your FishEye instance. Please navigate to **System Settings > Plugins** and open **System Plugins**. Please ensure that the *FishEye/Crucible Review Smart Commit Plugin* and *FishEye/Crucible JIRA Smart Commit Hook Plugin* are enabled (as well as all of their submodules). If you are unable to enable them, please view your FishEye logs for clues and raise a support request at <http://support.atlassian.com/> with your logs attached.

Configuring commit hooks

The [incremental indexing](#) process causes FishEye to poll all repositories at the specified interval to check for new commits, even though there might not be any new information to index. If you have a large number of repositories (> 100), this can lead to:

- A time lag between a commit being made and it appearing in FishEye.
- A high load on the FishEye server, and on the SCM.

Commit hooks allow you to set up your SCM so that indexing of a repository is triggered by a commit to the repository itself. This means that FishEye only runs the indexing process when necessary, and allows automatic [polling](#) to be disabled. Commits will appear sooner in FishEye, and the server load will be reduced.

To set up commit hooks you:

1. [Set the REST API token](#) in FishEye/Crucible.
2. Integrate the commit hook with your SCM.

On this page:

- [Triggering scanning remotely](#)
- [Integrating with your SCM](#)
 - [CVS](#)
 - [SVN](#)
 - [Perforce](#)
 - [Git](#)
 - [Mercurial](#)
- [Decreasing polling frequency](#)

Triggering scanning remotely

Once you've set your REST API token you can use it to trigger FishEye scanning when your repository is updated.

The basic way to do this is set up a shell script similar to:

```
echo Triggering FishEye scan
/usr/bin/curl -X POST -H "X-API-Key: YOUR-API-TOKEN-HERE"
http://SERVER:PORT/CONTEXT/rest-service-fecru/admin/repositories-v1/REPOSITORY-NAME/scan
```

Try running the script; if everything is fine, it will produce no output, and will trigger scanning in FishEye. If there are problems, curl will show an appropriate message.

If you're running on Windows, you'll need curl or a similar program. You can download the Windows version of curl [here](#). You'll need to save the script as a batch file (with the .bat extension).

Note: be sure to specify the full path to the curl binary on your system.

Integrating with your SCM

CVS

1. Checkout the CVSR00T module of your cvs repository:

```
cvs co CVSR00T
```

2. Edit the CVSR00T/logininfo file.
3. Add the following line to the file:

```
ALL /usr/bin/curl -X POST -H "X-API-Key: YOUR-API-KEY-HERE" -m 20  
http://SERVER:PORT/CONTEXT/rest-service-fecru/admin/repositories-v1/CVS-REP  
OSITORY-NAME/scan > /dev/null 2>&1 &
```

4. Commit your changes:

```
cvs commit CVSR00T/logininfo
```

SVN

1. Log into your svn server, go to the repository directory, find the hooks subdirectory there:

```
cd /var/www/svn/hooks
```

2. If it doesn't exist, create a new file called post-commit (or post-commit.bat on Windows), make sure it's executable by the user that the svn process runs as:

```
touch ./post-commit  
chmod 755 ./post-commit
```

3. Make sure the file starts with the following shebang line, pointing to your shell:

```
#!/bin/sh
```

4. Add the following to the post-commit file:

```
/usr/bin/curl -X POST -H "X-API-Key: YOUR-API-KEY-HERE" -m 20  
http://SERVER:PORT/CONTEXT/rest-service-fecru/admin/repositories-v1/SVN-REP  
OSITORY-NAME/scan > /dev/null 2>&1 &
```

You can find more details about svn hooks [here](#).

Perforce

1. As a Perforce administrator execute the following command:

```
p4 triggers
```

2. The trigger table form will be presented.
3. Add a field value for the field 'Triggers', named trigger-X, where X is the next number available for the trigger:

```
trigger-04 change-commit //... "/usr/bin/curl -s -o /dev/null -X POST -H
X-Api-Key:YOUR-API-KEY-HERE -m 20
http://SERVER:PORT/CONTEXT/rest-service-fecru/admin/repositories-v1/PERFORC
E-REPOSITORY-NAME/scan
```

4. You can customize the trigger to run only for a specific depot or directory, by replacing the `//...` above (which causes the trigger to be executed for every file) by a standard Perforce file pattern syntax.

You can find more details about Perforce triggers in the [Perforce System Administrator's guide](#).

Git

1. Choose a repository you want to trigger the scans from, usually this is the repository that all of your developers push to, and that you run CI from. Note that hooks are not propagated when cloning repositories.
2. Go to the hooks subdirectory of your repository:

```
cd /var/www/git/project/hooks
```

3. If it doesn't exist create a new file called `post-receive` make sure it's executable by the git server process.

```
touch ./post-receive
chmod 755 ./post-receive
```

4. Make sure the file starts with the following line, pointing to your shell:

```
#!/bin/sh
```

5. Add the following to the `post-receive` file:

```
/usr/bin/curl -X POST -H "X-Api-Key: YOUR-API-KEY-HERE" -m 20
http://SERVER:PORT/CONTEXT/rest-service-fecru/admin/repositories-v1/GIT-REPOSITO
R-Y-NAME/scan > /dev/null 2>&1 &
```

NOTE: not all methods of serving a git repository support commit hooks - if serving over http, you need to use

smart-http (either using [git-httpd-backend](#) or a dedicated repository manager like gitolite). You can find more information about smart http [here](#). Serving the repository over ssh or git-daemon should allow to run commit hooks as well.

Mercurial

1. Choose a repository you want to trigger the scans from, usually this is the repository that all of your developers push to, and that you run CI from. Note that hooks are not propagated when cloning repositories.
2. Go to .hg subdirectory of your repository:

```
cd /var/www/hg/project/.hg
```

3. If it doesn't exist create a file named hgrc :

```
touch ./hgrc
```

4. Add the following to the hgrc file:

```
[hooks]
changegroup = /usr/bin/curl -X POST -H "X-Api-Key: YOUR-API-KEY-HERE" -m 20
http://SERVER:PORT/CONTEXT/rest-service-fecru/admin/repositories-v1/HG-REPO
SITORY-NAME/scan > /dev/null 2>&1 &
```

NOTE: not all methods of serving a Mercurial repository support commit hooks - if serving over http, you can't use static-http serving.

Decreasing polling frequency

Once your commit hook is set up and successfully notifying FishEye about new commits to your repository, you can decrease the polling frequency on your repository (for example to 1 or 2 hours, instead of the default 1 minute).

The screenshot shows the 'Repository "Ada-BB"' configuration page. On the left is a sidebar with links: Summary, Maintenance, SCM Details, Updates (highlighted), Linkers, Permissions, and Include/Exclude Paths. The main content area is titled 'Updates' and contains the following text: 'Settings which affect how often FishEye and Crucible will check if there are new commits to the repository. need to define a custom updater.' Below this is a checkbox labeled 'Use the system default settings for updates' which is unchecked. Underneath is a 'Disable polling:' checkbox, also unchecked. At the bottom, there is a 'Polling Interval:' label followed by a text input field containing '1 hour' and a help icon.

With commit hooks configured, scheduled polling is only useful if the hook fails, for example because of connectivity issues to the server hosting FishEye. This will decrease the server load, but allow FishEye to still occasionally check for changes, and update the repository if needed.

Setting the REST API token

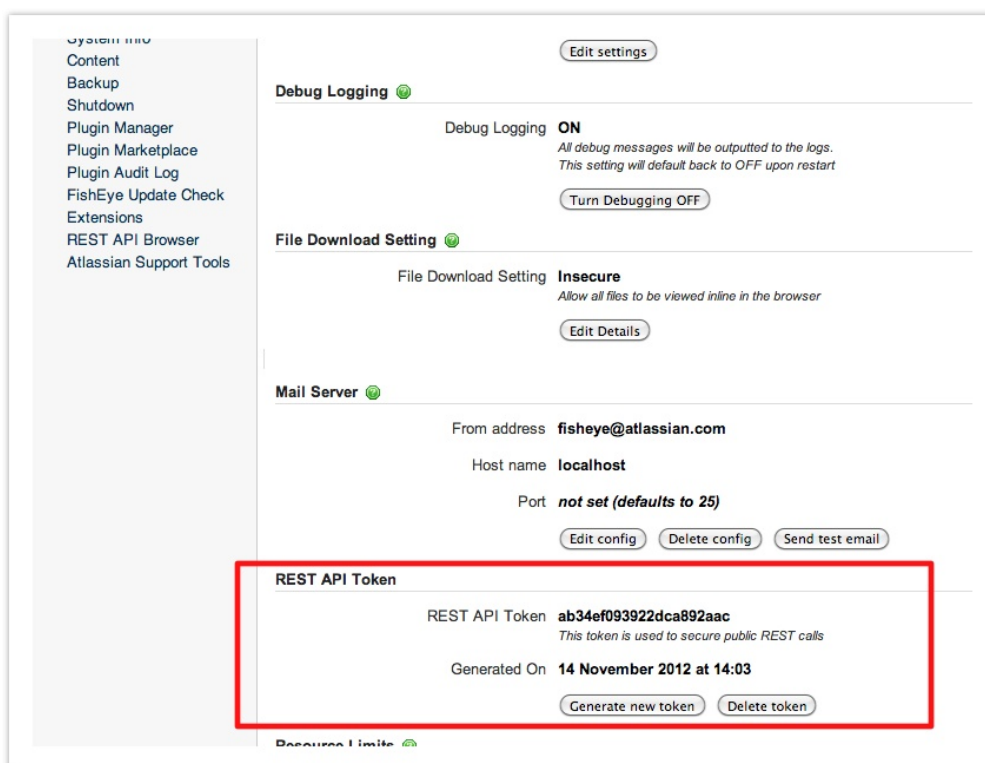
A REST API Token is an alternative way to authenticate to use a REST service in FishEye and Crucible. Instead of specifying a username and password of an authorized user, you can add a request header to the call that includes the REST API token:

X-Api-Key: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

This is currently only supported for a limited subset of REST services. It allows you to use services that normally requires administrator credentials, without exposing those or keeping them (for example in a script or a commit hook). The token cannot be used to log in to FishEye/Crucible or to perform any operations other than the designated REST service calls.

To generate a REST API token log in to the administrator panel of FishEye/Crucible, go to Security Settings/Authentication, and press the 'Generate new token' button in the REST API Token section. A REST API token will be generated and displayed on screen.

You can always prevent access using a previously generated token by deleting it, or generating a new one. Note that this will prevent any existing scripts using the token from working, until you update them with the new token.



Configuring ViewVC Compatibility

For backwards-compatibility and legacy system support, FishEye contains a URL-compatibility mode with the ViewVC (formerly known as ViewCVS) and CVSWeb tools. This allows FishEye to supplant or extend ViewVC, making use of the URLs set up for ViewVC.

To configure ViewVC compatibility, click '**ViewCVS URL Mappings**' on the '**Admin Menu**'.

For example, a ViewVC URL of the form

`http://host/viewcvs.cgi/x/y/z`

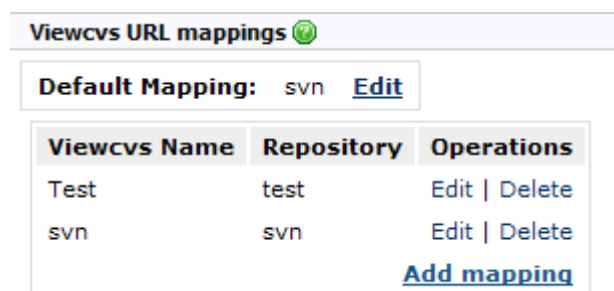
can be viewed in FishEye at


```
http://fisheyehost/viewcvs/x/y/z
```

FishEye can be configured to determine exactly how it provides this compatibility mode. In particular, you can configure how to map ViewVC repository names (cvsroot or root in the query parameter) to FishEye repository names.

The **Default Mapping** can be used to configure which repository to use if no repository is specified in the URL. If a repository name is given in the URL, you can tell FishEye how to translate that to the name of a FishEye repository. Otherwise, FishEye will attempt to use the repository name given in the URL directly.

Screenshot: Configuring ViewVC Compatibility



Viewcvs Name	Repository	Operations
Test	test	Edit Delete
svn	svn	Edit Delete

[Add mapping](#)

Once you have entered the mappings you will need to **restart** Fisheye.

Customising the Welcome Message

 This feature is not fully functional in FishEye 2.2 due to a [bug](#). To read more information about this, please see the [issue](#) in our public issue tracker.

To customise the welcome message, access the administration page, and click '**Customize Welcome & Support Message**' under '**Global Settings**' on the left navigation bar.

The '**Customize Welcome & Support Message**' page opens.

On this page, you can provide your own custom text for the FishEye welcome message that is displayed to users when they first log in. You can also provide custom Support text, providing the contact details of your own support organisation, which also appears on the opening page.

You can enter text into the boxes provided for either message and click the small '**Save Welcome Message**' or '**Save Support Message**' button to save it, or enter text for both messages and click '**Save All**'. The changes are made immediately.

Screenshot: FishEye Customize Welcome and Support Messages

Admin Menu

- Repository Settings**
 - Repository List
 - Repository Defaults
- Global Settings**
 - Server Settings
 - Security
 - Users
 - Groups
 - ViewCVS URL Mappings
 - Change Admin Password
 - Customize Crucible Defect Classifications
 - Projects
 - Permission Schemes
 - Trusted Applications
 - Customize Welcome and Support Message
- System**
 - Sys-Info/Support
 - Content
 - Backup
 - Plugins
 - Shutdown

customize Welcome and Support Messages

Here, you can provide your own custom text for the FishEye Welcome message that is displayed to users when they first log in.

You can also provide custom Support text, providing the contact details of your own support organisation, which also appears on the opening page.

Note: If left blank, the default support or welcome message will be used.

Welcome Message

[Large text input area for Welcome Message]

Save welcome message

Support Message

[Large text input area for Support Message]

Save support message

save all

Restoring the default messages

To revert to the default Welcome or Support messages, simply delete the text shown in the text box and click the respective **'Save'** button.

Manually editing the opening screen

You can also directly edit the XML file that contains the welcome and support messages. This file is called `config.xml`, located in your `/FISHEYE_HOME/` folder.

To do this, simply add the following XML tags to `config.xml`:

```
<content>
  <front-page-message>Example welcome message here</front-page-message>
  <support-message>Example support message here</support-message>
</content>
```

Using HTML

The content in the welcome screen can be arranged using tables, images or HTML tags such as the following:

```
<a href="http://www.atlassian.com">Link to Atlassian Home Page</a>
```

Software Update Notifications

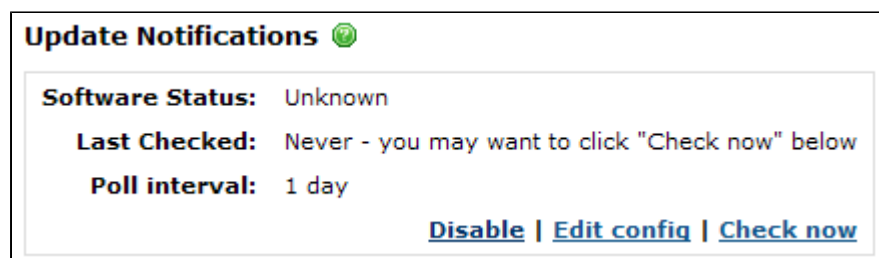
FishEye (and Crucible) can detect and notify you of new versions of the program. In the Admin screen, you can set a background thread to poll Atlassian's servers and report when a new version of FishEye or Crucible has been released.

To use the Update Notification:

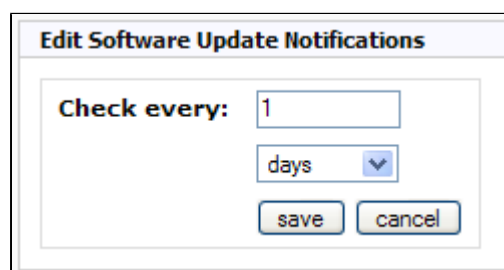
1. From the '**Admin Menu**', click '**Server Settings**' from the left navigation bar.
2. The '**Software Status**' line displays either 'Up to Date', 'New Version Available' or 'Unknown'.
3. The '**Last Checked**' line displays the date when the feature last checked for a new version.
4. Clicking the '**Check Now**' link will immediately check for a new version.
5. Clicking the '**Enable**' link allows you to switch on regular polling of the Atlassian servers, which will do periodic checking for the release of new versions. The default interval between checks is one day.
6. Once enabled, click '**Edit config**' to adjust the settings. You can set an interval of your choosing. The minimum interval allowed is one minute.

 A status of '**Unknown**' is displayed when the feature has not yet been activated.

Screenshot: Settings for Update Notification



Screenshot: Changing the Update Notification Interval



Enabling Access Logging in FishEye

Steps

- Stop FishEye/Crucible,
- and then create the file FISHEYE_HOME/content/WEB-INF/jetty-web.xml with the following content:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN"
"http://www.eclipse.org/jetty/configure.dtd">

<Configure class="org.mortbay.jetty.webapp.WebAppContext">
  <Call name="addHandler">
    <Arg>
      <New class="com.cenqua.fisheye.web.jetty.FishEyeRequestLogHandler">
        <Set name="requestLog">
          <New id="RequestLogImpl" class="org.mortbay.jetty.NCSARequestLog">
            <Arg><SystemProperty name="jetty.logs"
default="..../var/log"/>/fisheye-access.log.yyyy_mm_dd</Arg>
            <Set name="retainDays">90</Set>
            <Set name="append">true</Set>
            <Set name="extended">false</Set>
            <Set name="LogTimeZone">GMT</Set>
          </New>
        </Set>
      </New>
    </Arg>
  </Call>
</Configure>
```

- Restart Fisheye/Crucible.

This will create an access log in FISHEYE_HOME/var/log/fisheye-access.log.yyyy_mm_dd format (e.g. fisheye-access.log.2010_03_17). If you want to change the path to your FISHEYE_INST directory, change the default="..../var/log" to the path to the log folder in FISHEYE_INST.

The log directory must exist

If the path to the log directory given by the default attribute of the SystemProperty tag (defined in the line 10 in the jetty-web.xml above) does not exist, then FishEye will fail to start and will not log any error message.

The path given in the example below is correct when FISHEYE_INST and FISHEYE_HOME are the same directory, otherwise please use the absolute path of your FISHEYE_INST/var/log directory.

Log format

The logs are written in [NCSA format](#):

```

172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=TestRepo&ou
tputtype=image HTTP/1.1" 200 256
172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=npanday&out
puttype=image HTTP/1.1" 200 177
172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=jutils&outp
uttype=image HTTP/1.1" 200 775
172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=P4&outputty
pe=image HTTP/1.1" 200 177
172.20.5.186 - - [17/Mar/2010:22:50:21 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=Rails&outpu
ttype=image HTTP/1.1" 200 1311
172.20.5.186 - - [17/Mar/2010:22:50:22 +0000] "GET
/fe/commitSparkline.do?w=280&h=48&context=repository&repname=FE-2363&out
puttype=image HTTP/1.1" 200 128

```

Please refer to the [Jetty documentation](#) for more configuration options.

Compatibility

If you are using an earlier version of FishEye than FishEye 2.7.8, replace `com.cenqua.fisheye.web.jetty.FishEyeRequestLogHandler` by `org.mortbay.jetty.handler.RequestLogHandler`.

FishEyeRequestLogHandler was added in 2.7.8 to fix an issue where the user credentials would not be added to the NCSA log: [FE-3040](#).

Setting JVM System Properties

FishEye and Crucible use a number of Java Virtual Machine system properties. Most of these are properties set by the virtual machine itself or the FishEye/Crucible launch script. Users can configure the following property to control low level aspects of FishEye's operation:

- `jetty.http.headerbuffersize` - This controls the size of the largest HTTP header value that FishEye will allow (through its embedded servlet engine, Jetty). Some authentication systems require larger header values.

Setting a Larger Header Buffer Size

In FishEye 1.6.5 onwards, you can set a `jetty.http.headerbuffersize` system property (in bytes) to adjust the header size. This can be set by passing the `-D` parameter to the JVM, or by adding the property to `FISHEYE_INST/system.properties`. This file can be created as a plain text file (it follows the Java [properties format](#)) if you need to use this setting.

i The recommended approach is to add the property to `FISHEYE_INST/system.properties`. Add this as a new text file, using the Java [properties format](#).

i Users integrating FishEye with Single Sign On applications may require an adjustment to the JVM properties, specifically the HTTP header size. This may also be useful in other circumstances where FishEye's default HTTP header is too small (at 4096 bytes).

System Properties

This page lists the Java VM system properties which may be set to control certain aspects of FishEye and

Crucible's behaviour. Please refer to [Setting JVM System Properties](#) for how to set these properties in your instance. The version column indicates the version of FishEye/Crucible from which the property is applicable.

Property Name	Values	Default	Purpose	Version
<code>fisheye.branch.recently.active.check.limit</code>	number of branches	250	Used in CommitGraph to control sorting of branches. If more than the limit, sorted alphabetically, otherwise sorted by recent activity.	2.6.0
<code>fisheye.git.http-backend-timeout</code>	Integer milliseconds	3600000	This property sets the amount of time that the FishEye server should allow the underlying Git process to execute when attempting to serve a Git push or pull request.	2.7.3
<code>fisheye.p4.disable.ancestry</code>	true/false	false	If true, stops FishEye from processing P4 ancestry information. If you do not use stable Perforce branchspecs to manage branches, set this to true.	2.7.5
<code>fisheye.search.max.repositories</code>	number of repositories	100	Used by FishEye to limit the search to the top <i>n</i> repositories (by alphabetical order) when a cross-repository search is performed.	2.7.12
<code>crucible.detect.metadata.revision.changes</code>	true/false	true	If true, disables creation of reviews from changesets which contain metadata only changes (e.g svn property changes)	2.6.0

jetty.http.headerbuffersize	bytes	4096	This controls the size of the largest HTTP header value that FishEye will allow (through its embedded servlet engine, Jetty). Some authentication systems require larger header values.	1.6.5
org.mortbay.jetty.Request.maxFormContentSize	bytes	3000000	Max form content size – increase this value if your Git push fails with HTTP error 411 (see this KB article: Git PUSH results in RPC failed. result=22, HTTP code = 413)	

Related Topics

[FishEye Installation Guide](#)

[Setting JAVA_HOME](#)

[Best Practices for FishEye Configuration](#)

FishEye and Crucible Cookies

- [Authentication Cookies](#)
- [The 'Remember Me' Cookie](#)
 - [Cookie Key and Value](#)
 - [Use of Cookie for Authentication](#)
 - [Life of 'Remember Me' Cookies](#)
- [Other cookie usage](#)
 - [crucibleprefs1](#)

Authentication Cookies

FishEye/Crucible uses two cookies:

- The FESESSIONID cookie is created by the Jetty application server and used for session tracking purposes.
- The 'remember me' cookie, `remember`, is generated by FishEye when the user selects the **Remember me** checkbox on the login page. This option is only available in a FishEye only instance. A Crucible instance will always have this option turned on.

 You can read about cookies on the [Wikipedia page](#).

The 'Remember Me' Cookie

The 'remember me' cookie is a long-lived HTTP cookie. This cookie can be used to authenticate an unauthenticated session. FishEye/Crucible generates this cookie when the user logs in and in FishEye only mode also has the **Remember me** checkbox on the login page.

Cookie Key and Value

The cookie key is `remember` and is not configurable.

The cookie contains the user's username in plain text, a unique identifier plus a securely-generated random string.

Use of Cookie for Authentication

When a user requests a web page, if the request is not already authenticated via session-based authentication or otherwise, FishEye/Crucible will match the 'remember me' cookie (if present) against the token stored for the user in the FishEye/Crucible database (if present).

If the random string matches the value stored in the database and the cookie has not expired, the user is authenticated.

Life of 'Remember Me' Cookies

The 'remember me' cookie expires after 365 days of inactivity.

Other cookie usage

crucibleprefs1

crucibleprefs1 is a single cookie, which is used to store different user preferences. These are used to make actions the user has performed in the UI, (e.g setting the syntax used for a snippet) 'sticky'.

This saves the user having to continually set these values each time they visit FishEye/Crucible.

The complete list of values stored in crucibleprefs1 are:

No.	Full Name	Cookie Name	Default Value	Possible Values
1	longTagList	ltl	'M'	'L, M, H'
2	showFileHistImage	sfhi	'Y'	OPEN*
3	showAuthorChart	sac	'Y'	OPEN*
4	showQuickRecent	sqr	'Y'	OPEN*
5	annotateMode	annm	'age'	OPEN*
6	dirListOrder	dlo	'p'	'f, p, l'
7	csShowDiffs	csSD	'false'	'false, true'

8	csTruncateDiffs	csTD	'true'	'true, false'
9	hideHiddenDirectories	hhd	'Y'	'N, Y'
10	hideEmptyDirectories	hed	'N'	'N, Y'
11	includeUserOnHome	iuh	'Y'	'N, Y'
12	hideDeletedFiles	hdf	'N'	'N, Y'
13	alwaysExpandChangesets	aec	'N'	'N, Y'
14	viewMode	hvm	'p'	'p, l'
15	diffColour	dc	'2'	'2, 1'
16	diffMode	dm	'u'	'u, y'
17	diffWrap	dw	'h'	's, h'
18	diffContextBar	dcbt	's'	's, h'
19	selectedPanel	slp	'a'	't, a, o'
20	showPanel	shp	'Y'	'N, Y'
21	kOpts	k	'kv'	'v, kv, o, k'
22	activityStreamView	asv	'all'	'cru, jira, fe, all'
23	ignoreSlurpWarningDialog	iswd	'N'	'N, Y'
24	showEditModeHelp	semh	'Y'	'N, Y'
25	reviewHistoryTab	rht	'details'	'details, timeline'
26	showAuthorColumn	authcol	'Y'	'N, Y'
27	showRevisionColumn	revcol	'Y'	'N, Y'
28	showLineNumberColumn	lnumcol	'Y'	'N, Y'
29	implicitLines	impllines	'all'	'none, all, auth'

30	snippetSyntax	ssyn	'txt'	OPEN* - the syntax highlighting to use for Snippets
31	snippetProject	sprj	"	OPEN* - the name of the project to default to when creating Snippets
32	expCsHeader	xcsH	'N'	' N.Y '
33	lastbr	lsbr	"	** <pre>{ repName: { "type": "Branch" or "Tag", "name": branchName } }</pre>
34	visBranchSet	vbs	"	** <pre>{ "repoName1" : { bl: ["branchName1", "branchName2", "branchName3", "branchName4"] } }</pre>

				<pre> }, "repoName2" : { bl: ["branchName1", "branchName2", "branchName3", "branchName4"] }, "repoName3" : { bl: ["branchName1", "branchName2", "branchName3", "branchName4"] } }</pre>

35	visHighlighter	vhl	"	** <pre>{ "repoName1 ": {"hl":"'Au thor'}, "repoName2 ": {"hl":"'Se arch'}, "repoName3 ": {"hl":"'Li neage'}, ... }</pre>

NOTES

* OPEN means the value is not constrained by code

** JSON Values: examples are given instead of Possible Values.

Configuring the FishEye web server

To configure the server settings, click **Server** (under 'Server Settings') in the Admin area.

For information on configuring SSL for FishEye, see [FishEye SSL configuration](#).

 Some settings on this page require you to restart FishEye for changes to take effect.

Related pages:

- [Integrating with other web servers](#)
- [FishEye SSL configuration](#)
- [Subversion client settings](#)

Screenshot: Configuring server settings

Server Settings

Web Server

HTTP Bind **j2ee_eac-fisheye.atlassian.com.mgmt:38009**

SSL Bind **not set**

SSL Keystore **not set**

SSL Truststore **not set**

Web context **crucible**

If you have set up another web server to redirect requests to FishEye/Crucible, set the scheme, host and port of the URL it redirects from here. You must also set a matching Site URL below.

Proxy scheme **https**

Proxy host **extranet.atlassian.com**

Proxy port **443**

Site URL **https://extranet.atlassian.com/crucible/**

Ajp13 bind **not set**

Server timezone **not set (defaulting to US/Central)**

[Edit settings](#)

Setting	HTTP Bind
Description	The hostname the FishEye web server will bind to. This can take the form of a host name and port number, or you can leave the host name blank. If no host name is specified, then FishEye will bind to all available interfaces.

Examples	You can use bindings like the following: Host name and port number: hostname:8060 Port number only(requires a leading colon): :8060 IP address and port number: 10.0.0.11:60 (At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.)  Do not add http:// to the front i.e. do not add http://hostname:8060 or http://10.0.0.11:60. Simply define hostname:port or IpAddress:port
Restart required	Yes

Setting	Web context
Description	By default, the FishEye application can be accessed via http://HOST:PORT/ , where HOST and PORT are defined as above. You can force the FishEye application to be hosted on a different 'context' or 'path' by specifying a value here.
Example	If you specify a web context of 'fisheye' then FishEye will be accessible from http://HOST:PORT/fisheye/ instead of http://HOST:PORT/ .
Restart required	Yes

Setting	Proxy scheme
Description	Can be set if you are forwarding through to FishEye from another web server. If you set proxy values, you must also add a matching site URL (see below).
Example	Valid settings are http and https.

Restart required	Yes

Setting	Proxy host
Description	Can be set if you are forwarding through to FishEye from another web server. If you set proxy values, you must also add a matching site URL (see below).
Example	A valid setting would be <code>www.example.com</code> , where 'example' is the domain name of your web server.
Restart required	Yes

Setting	Proxy port
Description	Can be set if you are forwarding through to FishEye from another web server. If you set proxy values, you must also add a matching site URL (see below).
Example	The port number of the web server, an integer from 0 through 32,765.
Restart required	Yes

Setting	AJP13 Bind
Description	The bind host name for ajpv13. If no host name is specified, then FishEye will bind to all available interfaces.

Examples	You can use bindings like the following: Host name and port number: hostname:8009 Port number only(requires a leading colon): :8009 IP address and port number: 10.0.0.11:8009 (At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.)  Do not add http:// to the front i.e. do not add http://hostname:8060 or http://10.0.0.11:60. Simply define hostname:port or IpAddress:port
Restart required	Yes

Setting	Server timezone
Description	The time zone to use within FishEye. This time zone is used when displaying dates and parsing EyeQL date expressions. If blank, then the time zone of the server running FishEye is used.
Example	This defaults to the FishEye server's time zone, but you can select another zone from the drop-down list.
Restart required	No
Setting	Site URL

Description	The base URL for this FishEye instance. If not specified, FishEye will attempt to determine this value. You need to set this if you are forwarding through to FishEye from another web server, or if FishEye does not correctly determine the host name you want FishEye users to see FishEye on. Fisheye uses this URL to generate links in notification E-mails. If you have set a proxy scheme/host/port, the values used there must match the site URL.  Do not include a trailing slash ('/') at the end of the URL.
Example	<code>http://www.example.com/apps/fisheye</code>
Restart required	No

FishEye SSL configuration

This page describes how to use SSL with FishEye.

On this page:

- [1. Obtain an SSL certificate](#)
 - [Self-signed certificate](#)
 - [Certificate authority certificate](#)
- [2. Configure FishEye SSL](#)

1. Obtain an SSL certificate

An SSL certificate is required in order for SSL to work in FishEye. There are two ways to obtain one:

- by [signing one yourself](#), or
- by getting one [signed via a Certificate Authority](#).

Self-signed certificate

Self-signed certificates are useful in cases where you require encryption, but do not need to verify the website's identity. They are commonly used for testing and on internal corporate networks (intranets). If a certificate is not signed by a Certification Authority (CA), users may get prompted that the site is untrusted. They may then have to perform several steps to "accept" the certificate before they can access the site. This usually only occurs the first time the site is accessed.

 Please note, the following approach to create the certificate uses Java's [keytool](#), and has been designed for use with Java 1.6. There are other tools for generating certificates such as [openSSL](#), but the examples will be for keytools, unless the functionality required does not exist in keytools.

To obtain a self-signed certificate:

1. Run the following command to create a new keystore file in the FishEye home directory (if one does not already exist):

- Windows:

```
"%JAVA_HOME%\bin\keytool" -keystore \path\to\keystore.kst  
-alias fisheye -genkey -keyalg RSA
```

- Unix/Linux:

```
$JAVA_HOME/bin/keytool -keystore /path/to/keystore.kst -alias  
fisheye -genkey -keyalg RSA
```

2. When the keytool utility prompts you with 'What is your first and last name?', enter the fully qualified hostname of the server running FishEye/Crucible. Do not enter your first name and last name.

What is the fully qualified hostname of my server?

The fully qualified hostname of your server is the name you would type in your web browser, after the `http://` section, to access a conventional website. When the client web browser examines the certificate, it checks this field to make sure it matches the hostname. If it does not, the browser may prevent access to the site and will at least generate messages saying that there is a mismatch. An example of a qualified hostname is: `support.atlassian.com`

3. When the keytool utility prompts you to enter the keystore password and key password, enter your desired passwords. You must also specify these passwords in the FishEye/Crucible web admin (recommended) or specify these passwords directly in your `config.xml` in the corresponding attributes (see [Configure FishEye/Crucible SSL](#) below).

```
<web-server context="/crucible"  
  site-url="http://localhost:6060/crucible/">  
<http bind=":6060"/>  
<ssl bind=":6443" keystore="/path/to/keystore.kst"  
  keystore-password="password123"  
  truststore="/path/to/keystore.kst" truststore-password="password123"/>  
</web-server>
```

4. You now have the minimal requirements to run SSL in FishEye. Next, configure FishEye/Crucible to use SSL as described in the [Configure FishEye/Crucible SSL section](#) below.

Certificate authority certificate

Digital Certificates issued by trusted 3rd party Certification Authorities (CAs) provide verification of the identity of your website. Many CAs simply verify the domain name and issue the certificate. Other CAs such as [VeriSign](#) also verify the existence of your business, the ownership of your domain name and the authority to whom the certificate application was made, thereby providing a higher standard of identification and authenticity.

A list of CA's can be found [here](#). Some of the most well-known CAs are:

- [Verisign](#)
- [Thawte](#)

- [CAcert](#) (relatively new CA, providing free CA certificates)



This list is not an endorsement of the given certificate authorities by Atlassian, and is only provided as an example.

To obtain a certificate signed by a CA:

1. Follow the instructions from the certificate authority you want your certificate signed by. Most CAs have their own instructions for you to follow, e.g. [GoDaddy](#) and [VeriSign](#).
2. If your CA requires a certificate signing request (CSR), use the following command:
 - Windows:

```
"%JAVA_HOME%\bin\keytool" -certreq -alias fisheye -keystore  
\path\to\keystore.kst -file \path\to\fisheye.csr
```

- Unix/Linux:

```
$JAVA_HOME/bin/keytool -certreq -alias fisheye -keystore  
/path/to/keystore.kst -file /path/to/fisheye.csr
```

3. Ensure that the keystore being used to generate the signing request contains an existing key/cert keypair. If you don't already have a key/cert keypair, follow the steps in [Self Signed Certificates](#) to generate a keypair.
4. If the certificate you receive from the CA is not in a format that keytool understands then the `openssl` command can be used to convert formats since keytool does not perform any of these conversions. For example, to convert between DER and PEM formats:

```
openssl x509 -in certificate.der  
-inform DER -outform PEM -out certificate.crt
```

If you have a key and a certificate in separate files you will need to combine them into a PKCS12 format file for loading into a new keystore. See [these instructions](#) for information on how to do so.

5. Load the certificate into the keystore, as follows:

- CA certificate in PEM format:

- Windows:

```
"%JAVA_HOME%\bin\keytool" -keystore \path\to\keystore.kst  
  
-import -alias fisheye -file  
\path\to\certificate.crt  
-trustcacerts
```

- Unix/Linux:

```
$JAVA_HOME/bin/keytool -keystore /path/to/keystore.kst
                        -import -alias fisheye -file
/path/to/certificate.crt
                        -trustcacerts
```

- CA certificate in PKCS12 format:
 - Windows:

```
%JAVA_HOME%\bin\keytool -importkeystore
                        -srckeystore \path\to\certificate.pkcs12
                        -srcstoretype PKCS12 -destkeystore
\path\to\keystore.kst
```

- Unix/Linux:

```
$JAVA_HOME/bin/keytool -importkeystore
                        -srckeystore /path/to/certificate.pkcs12
                        -srcstoretype PKCS12 -destkeystore
/path/to/keystore.kst
```

6. You will now have a keystore file with your certificate in it. Next, configure FishEye/Crucible to use SSL as described in the [Configure FishEye/Crucible SSL section](#) below.

2. Configure FishEye SSL

FishEye requires additional configuration in order to use SSL. This configuration can be done from the web admin as described below (recommended), or added directly into your `config.xml`. Setting the bind address and the keystore and password field, as described below, will enable SSL in FishEye. You will need to restart your server after updating the settings, for the changes to take effect.

To configure FishEye to use SSL:

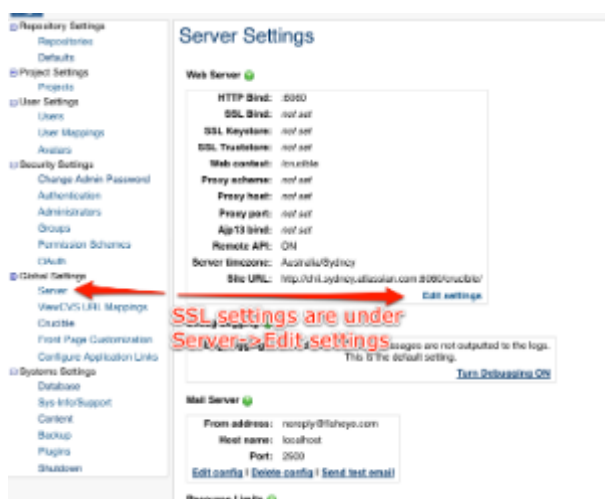
1. Navigate to the FishEye admin area and click **Server**, under 'Global Settings' (see 'Server Settings' screenshot below).
2. Click **Edit settings** in the 'Web Server' section.
3. Update the following fields (see 'Example SSL Settings' screenshot below):

Field	Description	Example value
SSL Bind Address	The SSL port. Most browsers default to 443 as the SSL port for the https protocol.	443

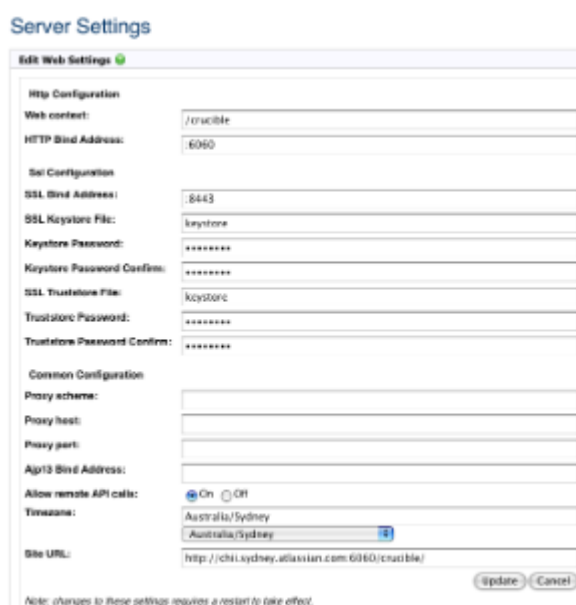
SSL Keystore	Path to the keystore file (as generated in the steps above). This path can be a relative path - e.g., putting in keystore will mean that the file is in FISHEYE_HOME/keystore (or FISHEYE_INST/keystore if it has been set up)	keystore
SSL Keystore Password	The password you used in the above step when creating the keystore file. If you did not set a password, leave this empty. See the passwords section	
SSL Truststore	Optional setting. Path to the truststore file. May be the same as the keystore. Truststore is a list of trusted CAs. Format is the same as the keystore entry.	
SSL Truststore Password	Optional setting. The password you used in the above step when creating the truststore file. If you did not set a password, leave this empty. See the passwords section	

4. Restart your server.

Screenshots: FishEye server settings (click to view larger images)



1. Server Settings



2. Example SSL Settings

Integrating with other web servers

Third Party Web Server Support

Please note that support for third-party web servers is provided on a best-effort basis. Please see [Atlassian Support Offerings](#) for more information.

FishEye has a built-in web server, but commonly runs in an environment that has its own web server. You can easily proxy through to FishEye from this primary web server, so that it appears as if FishEye is part of the primary web server.

In most situations, FishEye can determine the host and port of the primary web server automatically. This is useful when you have multiple virtual hosts proxied through to the one FishEye instance.

If it appears FishEye is having trouble automatically detecting the primary web server's host and port, you will need to set the **Proxy host** and **Proxy port** parameters. If the primary web server is running on WEBHOST:80 and FishEye is running on FEHOST:8060, then you can set FishEye's **Proxy host** and **Proxy port** parameters to WEBHOST and 80.

If the primary web server is using SSL, add the parameter `proxy-scheme="https"` to the connecting server as in the following example:

```
<http bind=":8080" proxy-host="extranet.example.com" proxy-port="443"
proxy-scheme="https"/>
```

You will probably want FishEye to appear in a subdirectory of the primary server. In that case, you need to set FishEye's **web context** parameter. The rest of the page assumes you have set this value to `fisheye`.

You will need to restart FishEye before any of the above parameters take effect.

Then, configure your primary web server as described below.

Apache

The easiest way to proxy through to FishEye is using the `ProxyPass` and `ProxyPassReverse` directives which require the `mod_proxy` module. Add this section to your Apache configuration:

```
ProxyPass /fisheye http://FEHOST:8060/fisheye
ProxyPassReverse /fisheye http://FEHOST:8060/fisheye
```

When you proxy to the FishEye host you should set the web context in the [FishEye web server configuration](#) to the same path as the one you are proxying. In the example above the FishEye web context should be set to `"fisheye"`.

If you want Apache to serve FishEye's static content, then you can do something like this instead:

```
<Directory "/FISHEYE_HOME/content/static" >
  Allow from all
  AllowOverride None
</Directory>
Alias /fisheye/static /FISHEYE_HOME/content/static
ProxyPass /fisheye/static/ !
ProxyPass /fisheye http://FEHOST:8060/fisheye
```

 An alternative to using ProxyPass is to use mod_rewrite with the [P] flag.

AJP

FishEye also supports AJPv13 connectivity. For more information, please see [AJPv13 Authentication](#).

Configuring indexing

Understanding how FishEye indexes repositories

FishEye's administration interface allows you to configure the number of threads used for repository indexing.

FishEye uses two types of indexing operation:

- **Initial indexing** - the complete scan of the repository that is performed when FishEye first accesses the repository, and
- **Incremental indexing** - the quick, background process that regularly scans repositories for changes.

Configuring the number of indexing threads

To configure the number of threads FishEye uses for indexing:

1. Go to the FishEye admin area and click **Server** (under 'Global Settings').
2. Click **Edit Details** (under 'Resource Limits'). You can change the number of threads available for each type of indexing operation.
3. Click **Update**.

Screenshot: FishEye multi-threading configuration

Edit the Resource Limits

Max Initial Index Threads

Max Incremental Index Threads

*This configures the numbers of threads used to index your repositories.
See the [documentation](#) for more details.*

Choosing a thread setting

Changing the number of threads always involves a trade-off in performance; you should generally change these settings in increments of one, then observe the performance impact. A safe range is from 1-3 threads on both settings. Using more than three threads on either setting is not recommended; the more threads, the busier the server will become when indexing. There is certainly no benefit in having more threads than you have repositories. Adding a lot of threads may impact overall system performance.

By default, FishEye uses one thread each for the two indexing types.

Note that decreasing the number of indexing threads does not have an effect until any current scanning ends. To force this effect immediately, perhaps because of a performance issue, a FishEye restart is needed.

For more information on managing FishEye's performance, see the [FishEye Tuning](#) page.

Optimising the Search Indexes on Restart

FishEye/Crucible allows you to optimise the search indexes when you restart your instance.

You can use this feature to:

- Reduce the size of the search indexes,
- Make the search faster if the indexes are significantly under-optimised, and
- Reduce the number of "Too May Files" [errors](#) that you might see in the logs. This is useful on systems where FishEye or Crucible have constraints on the number of files they can open.

In most circumstances, the search indexes should be fairly close to their optimal state. This is usually the case when using Crucible with FishEye, or when using FishEye alone. The search technology we use, [Lucene](#), results in indexes fairly close to optimal when most of the operations on the indexes are [additions](#) (the additions are the changesets that are indexed as they appear in your repositories). For this reason, it should not be necessary to run the optimisation often.

When you are using this feature, the optimisation will run solely on the *next start*. It will not run every time FishEye or Crucible starts, to prevent slowing the start of your instance. The optimisation itself should be fast, and within 10 minutes even for large indexes.

Running the Optimisation

1. Navigate to the FishEye/Crucible administration console.
2. Click the '**Shutdown**' link in the left menu. Note, you will need to confirm the shutdown at a confirmation screen before your instance is shut down.
3. In the window that appears, click the '**Optimize the search indexes the next time that FishEye and Crucible start**' checkbox (or '**Optimize the search indexes the next time that FishEye/Crucible starts**' if you are using either FishEye or Crucible alone).
4. Click '**Shutdown**'. Your instance will be shut down, and the optimisation of the search indexes will be scheduled for the next startup.

Customising FishEye's look and feel

FishEye commercial/academic license users have access to the HTML/JSP source of FishEye and can customise FishEye's look and feel.

FishEye Source Edition

To use custom HTML/JSP content, you must be using a build of FishEye that contains the JSP source. These builds are named `fisheye-1.x.y-jspsource.zip` instead of the normal `fisheye-1.x.y.zip` bundle.

If you have a commercial license assigned to your account, you will see a **source download** option on your download page.



You must install a JSP source build – you cannot copy JSP files from a source build into an existing ordinary installation.

Customising Content

With the JSP source build you can modify any of the files in <FishEye installation

directory>/content/. However we strongly recommend that you use separate installation and FISHEYE_INST directories (as described in [Installing FishEye on Windows](#) and [Installing FishEye on Linux and Mac](#)), and that you store your modified files in FISHEYE_INST/content (If you use FISHEYE_INST/content, you only need to keep your modified HTML/JSP files in that directory. This simplifies your upgrade process).

When you make changes to content, your changes should appear when you next refresh the page in your browser. If they do not, then log in to the FishEye Administration screens, click **Content** under 'System Settings', and follow the instructions on the 'Content' page.

Content

 Note: these functions are intended for people with the JSP-Source Edition.

 When updating JSPs or other content, you may need to **flush** before that content is served by FishEye and Crucible. In some situations, such as deleting an existing file from your custom content, a **flush** may not be sufficient. In such cases:

- Stop FishEye and Crucible.
- Delete the JSP classes directory: /opt/j2ee/domains/atlassian.com/eac-fisheye/data/var/tmp/web
- Restart FishEye and Crucible.

Custom content directory `/opt/j2ee/domains/atlassian.com/eac-fisheye/data/content`

Flush

Environment variables

Environment variables are system-wide settings that are required for certain applications.

Please note, that if you run FishEye as a Windows service, then any environment variables that you require must be set in your <FishEye home directory>/wrapper/conf/wrapper.conf file. See [Running FishEye as a Windows service](#).

The environment variables used by FishEye are listed in the following table:

On this page:

- [Setting environment variables under Windows XP](#)
- [Setting environment variables under Linux or Mac](#)

Environment variable	Notes

JAVA_HOME	Used by FishEye to select the Java Virtual Machine (JVM) to be used to run FishEye. If this environment variable is not set, FishEye will use whatever Java executable is available on the path. In Linux systems, this may sometimes be GCJ-based which causes some problems running FishEye. See either Installing FishEye on Windows or Installing FishEye on Linux and Mac for instructions on setting JAVA_HOME. If you're using the Java Service Wrapper to run FishEye as a service on Windows, the wrapper will use the Java in your PATH instead of using JAVA_HOME. We recommend that JAVA_HOME should point to the Java executable in your PATH.
FISHEYE_OPTS	Used to pass parameters to the Java Virtual Machine (JVM) used to run FishEye. This is typically used to set the Java heap size available to FishEye (see Fix out of Memory Errors). With a Sun JVM, for example, you would use: <pre>FISHEYE_OPTS=-Xmx1024m -XX:MaxPermSize=128m</pre> This would give FishEye a max of 1024 MByte heap, a Max permanent generation size of 128m. See Tuning FishEye performance for more information. You can also use FISHEYE_OPTS to set various System Properties in FishEye. For example: <pre>FISHEYE_OPTS=-Dname.of.property=value</pre> After having set FISHEYE_OPTS and restarting your server, go to Administration > Sys Info/Support > System Info, and check your JVM Input Arguments to ensure that your server is picking up your FISHEYE_OPTS as expected.
FISHEYE_ARGS	FISHEYE_ARGS are the arguments which will be passed to FishEye when it is started. You can set this to --debug, for example, or --debug-perf if you always want to have FishEye debugging put into the FishEye log files.
FISHEYE_LIBRARY_PATH	Used to tell FishEye where it should look to load any additional native libraries.

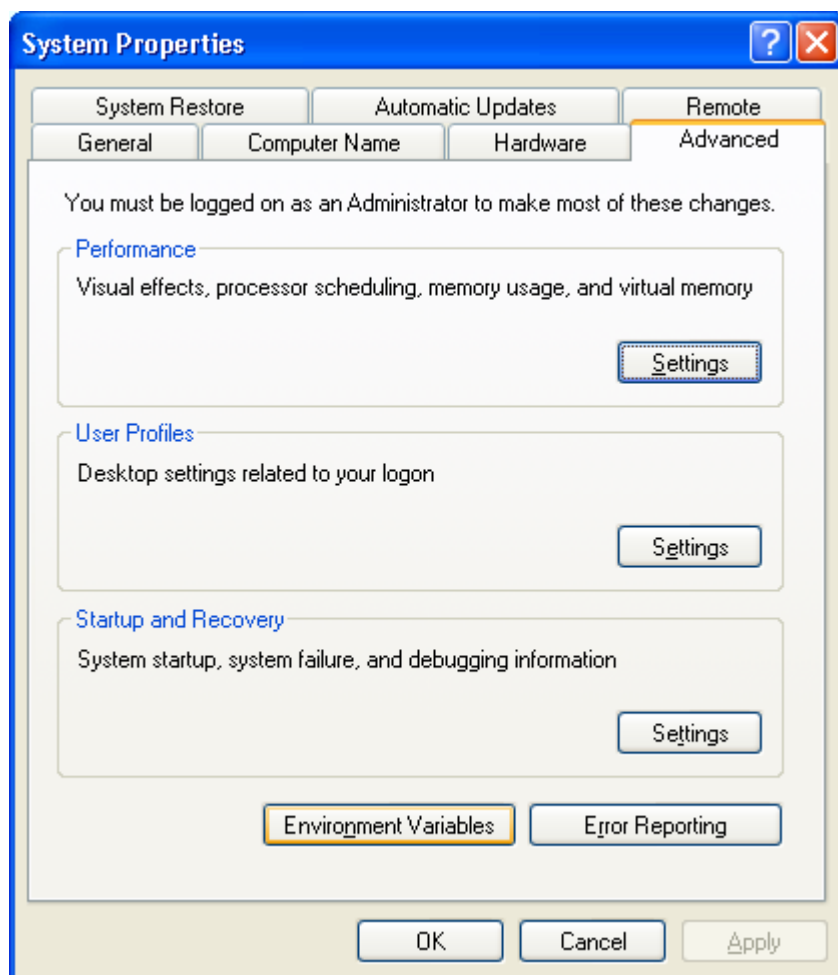
FISHEYE_INST	Used to tell FishEye where to store its data. It is recommended that you separate FishEye's data from its application files in <FishEye home directory> by using this variable. For example: c:\path\to\fisheye_data (no trailing backslash) Read more about using FISHEYE_INST in either <u>Installing FishEye on Windows</u> or <u>Installing FishEye on Linux and Mac</u>.

Setting environment variables under Windows XP

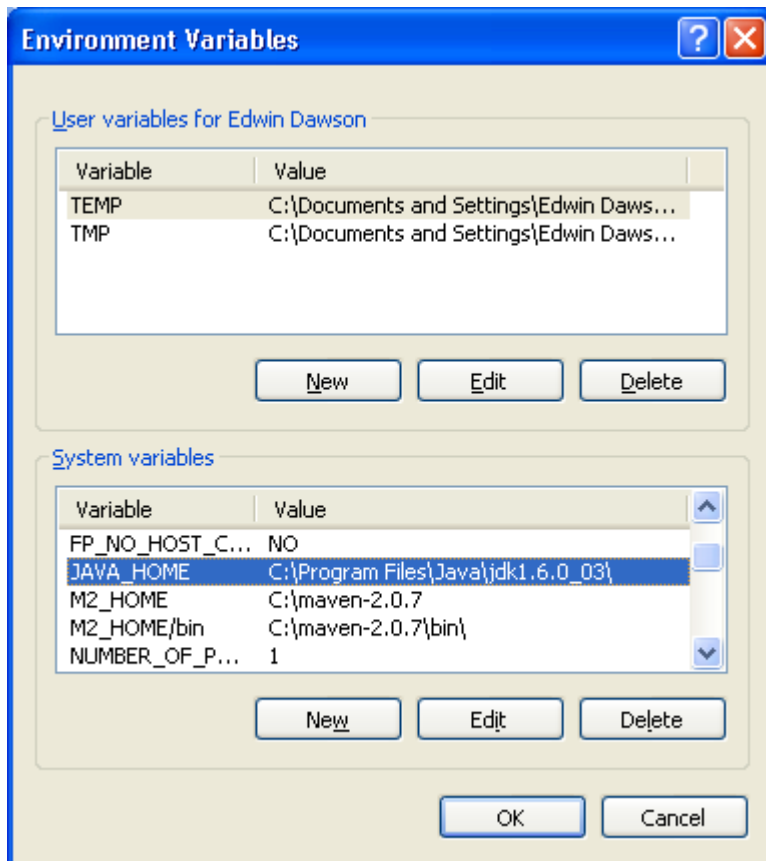
(System environment variables are ignored if FishEye is run as a Windows service. Please refer to [Running FishEye as a Windows service](#).)

(Linux instructions for setting environment variables are [here](#).)

1. Click **Start > Control Panel > System**.



2. Click the **Advanced** tab.
3. Click the **Environment Variables** button.



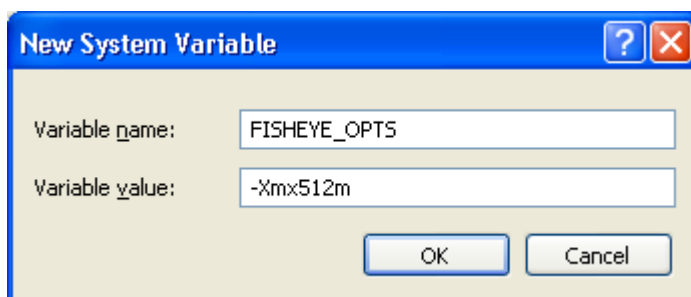
4. Click **New**.

5. In the **Variable name** field, enter the name of the environment variable, for example:

FISHEYE_OPTS

6. In the **Variable value** field, enter the setting as required. This may be quite cryptic, for example the default value for FISHEYE_OPTS is this:

-Xmx256m



7. Restart the computer.

Setting environment variables under Linux or Mac

There are a number of ways to set environment variables on Linux or UNIX based systems (including **Mac OS X**). Here are just two:

For your current user:

1. Open up a shell or terminal window
2. Type this command:

```
vi ~/.profile
```

(vi is a text editor, you can use another if desired)

3. Add this command:

```
export (variable name)=(variable value)
```

Where (variable name) and (variable value) are the environment variable elements. For example, if the environment variable you are setting is FISHEYE_OPTS, and the variable value is -Xmx256m, you would type the following:

```
export FISHEYE_OPTS=-Xmx256m
```

Add this command on its own line at the end of the file.

4. Save, and restart your shell.

For all users in the system:

1. Open up a shell or terminal window
2. vi /etc/profile (replace vi with your favourite text editor)
3. Add export (variable name)=(variable value) on its own line at the end of the file
4. Save, and restart your shell

If you are using a GUI, you may not need to open up the shell. Instead, you might be able to open the file directly in a graphical text editor.

 If you are experiencing memory errors in FishEye, see [Fix Out of Memory errors by increasing available memory](#).

Configuring Outgoing Web Proxy Support for FishEye or Crucible

Accessing External Hosts

FishEye may need to connect to hosts that are outside your network. For example, in Administration > Server Settings > Update Notifications, FishEye needs to access <http://updates.atlassian.com> to check for updates. If the FishEye server can't access the site it will throw the following error:

```
ERROR - Error while checking for newer versions from http://update.atlassian.com
(update.atlassian.com)
```

When trying to check if there are newer versions of FishEye or Crucible

In some environments, access to hosts outside the network are forbidden (in which case, you will need to manually check for updates). In other environments access to hosts outside of the network need to go through a

proxy.

If you do have a proxy server, you need to tell FishEye how to send its requests through the proxy:

Configuring an outbound HTTP proxy in Fisheye

Proxy support is configured by passing certain arguments to the Java Virtual Machine on startup (for Fisheye, you can add them to the FISHEYE_OPTS [Environment Variables](#)). These properties follow the [conventions defined by Sun](#):

- http.proxyHost
- http.proxyPort (default: 80)
- http.nonProxyHosts (default: <none>)

At a minimum, you need to define http.proxyHost to configure an HTTP proxy. System property configuration is described on the [Setting JVM System Properties](#) page.

Properties http.proxyHost and http.proxyPort indicate the proxy server and port that the HTTP protocol handler will use.

```
-Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=8080
```

Property http.nonProxyHosts indicates the hosts which should be connected to directly and not through the proxy server. The value can be a list of hosts, each separated by a |, and in addition a wildcard character (*) can be used for matching. For example:

```
-Dhttp.nonProxyHosts=*.foo.com|localhost
```

Note: You may need to escape the pipe character (|) in some command-line environments.

If the http.nonProxyHosts property is not configured, all web requests will be sent to the proxy.

Configuring HTTP proxy authentication

Proxy authentication is also configured by providing [system properties](#) to Java in your application server's configuration file. Specifically, the following two properties:

- http.proxyUser – username
- http.proxyPassword – secret

How it Looks

So an example of your FISHEYE_OPTS [Environment Variables](#) will be:

```
FISHEYE_OPTS="-Xms128m -Xmx1024m -XX:MaxPermSize=128m  
-Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=8080  
-Dhttp.nonProxyHosts=*.foo.com|localhost -Dhttp.proxyUser=USERNAME  
-Dhttp.proxyPassword=SECRET"
```

After having set the FISHEYE_OPTS and restarting your server, go to Administration > Sys Info/Support > System Info, and check your JVM Input Arguments to ensure that your server is picking up your FISHEYE_OPTS as expected.

Configuring FishEye Security

Best practices for configuring FishEye security

The best way to harden a system is to look at each of the involved systems individually. Contact your company's security officer or department to find out what security policies you should be using. There are many things to consider, such as the configuration of your underlying operating systems, application servers, database servers, network, firewall, routers, etc. It would be impossible to outline all of them here.

This page contains guidelines on good security practices, to the best of our knowledge.

Configuring the web server

Please refer to the following guides for system administrators:

- How to configure Apache to lock down the administration interface to those people who really need it. See [Using Apache to limit access to the Confluence administration interface](#) for guidance.
- How to reduce the risk of brute force attacks: [Configuring CAPTCHA](#).

Configuring the application server

See the following system administrator guide for general hints at the application server level (FishEye uses Jetty, but the principles described here still apply):

- [Tomcat security best practices](#)

Configuring the application

The way you set up FishEye roles, permissions and processes makes a big difference in the security of your FishEye site.

Below are some more FishEye-specific items to consider. None of these provides 100% security. They are measures to reduce impact and to slow down an intruder in case your system does become compromised.

- Restrict the number of users with powerful roles or group memberships. If only one department should have access to particularly sensitive data, then do restrict access to the data to those users. Do not let convenience over-rule security. Do not give all staff access to sensitive data when there is no need.
- Put documented procedures in place for the case of employees leaving the company.
- Perform security audits regularly. Know who can help in case a security breach occurs. Perform 'what if' planning exercises. ('What is the worst thing that could happen if a privileged user's password were stolen while he's on vacation? What can we do to minimise damage?').
- Make sure the FishEye database user (and all datasource database users) only has the amount of database privileges it really needs.
- Monitor your binaries. If an attacker compromises an account on your system, he will usually try to gain access to more accounts. This is sometimes done by adding malicious code, such as by modifying files on the system. Run routine scripts that regularly verify that no malicious change has been made.

Configuring system admin access

Below are some things to consider specifically related to the system admin role:

- Keep the number of FishEye administrators extremely low. For example, 3 system administrator accounts should be the maximum.
- The administrators should have separate FishEye accounts for their administrative roles and for their day to day roles. If John Doe is an administrator, he should have a regular user account without administrator access to do his day to day work. This could be a 'john.doe' account. In addition, he should have an entirely separate account (that cannot be guessed by an outsider and that does not even use his proper name) for administrative work. This account could be 'jane smith' – using a username that is so obscure

or fake that no outsider could guess it. This way, even if an attacker singles out the actual person John Doe and gets hold of his password, the stolen account would most likely be John's regular user account, and the attacker cannot perform administrative actions with that account.

- Lock down administrative actions as much as you can. If there is no need for your administrators to perform administrative actions from outside the office, then lock down access to those actions to known IP addresses, for example. See [Using Apache to limit access to the Confluence administration interface](#) for guidance.

Further precautions

As another precaution:

- Regularly monitor the above requirements. There are many things that could start out well, but deteriorate over time:
 - A system may start out with just 3 administrators, but over the course of a year this could grow to 30 administrators if no one prevents expansion.
 - Apache administration restrictions may be in place at the start of the year, but when the application server is migrated after a few months, people may forget to apply the rules to the new system.

Again, keep in mind that the above steps may only be a fraction of what could apply to you, depending on your security requirements. Also, keep in mind that none of the above rules can guarantee anything. They just make it harder for an intruder to move quickly.

Brute force login protection

This page explains how to configure or disable FishEye's brute force login protection.

FishEye will protect against brute force login attacks by forcing users to solve a CAPTCHA form after a configurable number of consecutive invalid login attempts. By default, this functionality is enabled, and the number of invalid attempts is set to three.

 Once a user logs in successfully, they will no longer be required to solve the CAPTCHA form.

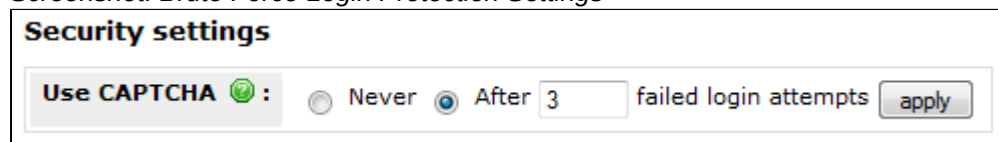
Configuring brute force login protection

To configure brute force login protection:

1. In the Admin area, click **Authentication** under 'Security Settings'.
2. Scroll down to the 'Security Settings' section at the bottom of the screen.
3. The 'Use CAPTCHA for login' options are:
 - Never.
 - After N login attempts (the default number of allowed attempts is three).

Select the desired option (where 'N' is the number of attempts), and click 'Apply'. The changes will be made immediately.

Screenshot: Brute Force Login Protection Settings



The screenshot shows a web interface titled "Security settings". Below the title, there is a section for "Use CAPTCHA" with a green checkmark icon. It features two radio buttons: "Never" and "After". The "After" option is selected, and next to it is a text input field containing the number "3". To the right of the input field is the text "failed login attempts". At the bottom right of this section is an "apply" button.

Brute force protection against remote API calls

Login requests by the FishEye remote API libraries are also covered by the brute force protections. After the number of invalid attempts is exceeded (the default is three), then the remote API for that user will be prevented from making further login attempts (as that user will now be required to solve a CAPTCHA form through the web interface in order to log in).

Configuring CAPTCHA

This page contains instructions on enabling and disabling [CAPTCHA](#) testing on the following login-screen features of FishEye and Crucible:

- Public signup.
- Password retrieval.

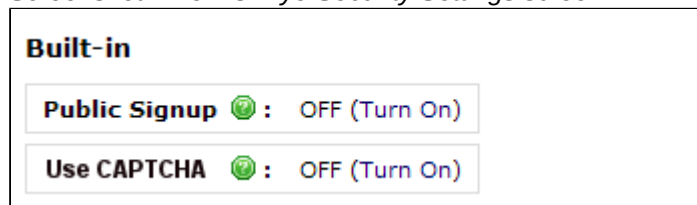
Administrators can enable or disable CAPTCHA testing on the [public signup](#) and [password retrieval](#) screens. CAPTCHA testing is enabled by default.

You may want to turn CAPTCHA off if you are serving FishEye or Crucible behind a firewall.

To do this:

1. Open the FishEye **Admin screen**, then choose '**Security**' from the left navigation bar.
2. The FishEye Security Settings screen opens.
3. Just below '**Public Signup**', the '**Use CAPTCHA**' option can be set to OFF or ON.
4. Select the desired setting by clicking '**Turn (OFF or ON)**'.
5. The setting is immediately changed.

Screenshot: The FishEye Security Settings screen



 To see instructions on configuring CAPTCHA for brute force login protection, see the page on [Brute force login protection](#).

Granting administrator privileges

You can grant Admin privileges to users and groups in FishEye. These privileges exist in addition to the core 'Admin' account (in FishEye/Crucible 1.6 onwards).

 Be sparing in granting Admin privileges, as all Admin users have the 'keys to the kingdom'. They can add or remove other's user or group admin rights, and change the password of the core Admin account.

Setting admin privileges for individual users

Log in as the Admin user and select **Administrators**, under 'Security Settings' in the left navigation panel.

 Take care not to remove admin rights from your own account.

To grant admin rights to a particular user:

1. Select their username in the 'Non-Admin Users' column.
2. Click **Add**.

To remove a user's admin rights:

1. Select their username in the 'Admin Users' column.
2. Click **Remove**.

Setting admin privileges for groups

Log in as the Admin user and select **Administrators**, under 'Security Settings' in the left navigation panel.

⚠ Take care not to remove admin rights from your own account.

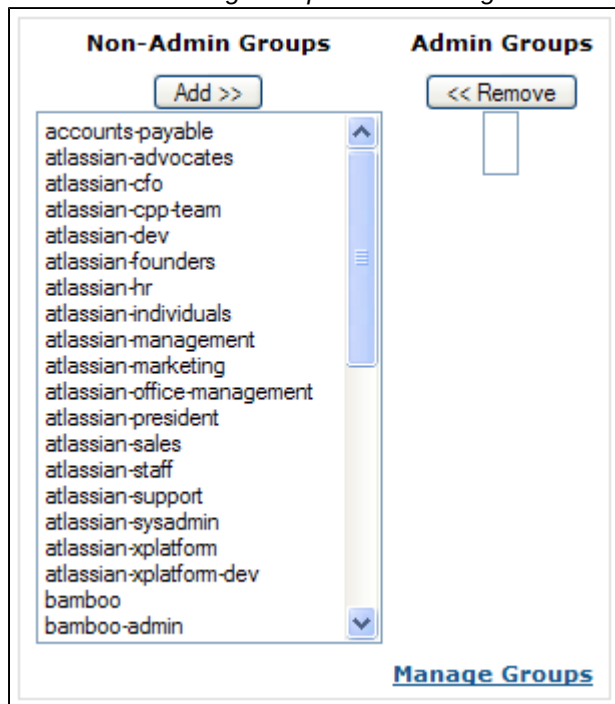
To grant admin rights to a particular group:

1. Select the group name in the 'Non-Admin Groups' column.
2. Click **Add**.

To remove a groups's admin rights:

1. Select the group name in the 'Admin Groupss' column.
2. Click **Remove**.

Screenshot: Setting Group Admin Settings



Configuring public signup

If you enable *public signup* for your FishEye system, visitors can create their own FishEye user accounts using the **Signup** link on the login screen:

Login Required

Username:

Password:

Not a member? [Signup](#) for an account.

Public signup is disabled by default.

To enable public signup:

1. Go to the [FishEye 'Admin Menu'](#).
2. Click **Security** in the left navigation column. The 'Authentication Settings' page will be displayed (see screenshot below).
3. Click **On**, next to 'Public Signup'.
4. Log out of FishEye and verify that the login screen now contains a **Signup** link.

Screenshot: 'Authentication Settings — Public Signup'

Authentication Settings

Permissions Summary

	Allow anon access	Built-in Groups	Action
Global:	YES (No)		
Crucible:	NO (Yes)	GR	
Repository Default:	YES	non set	Edit
Per-repository:			
svn:	YES (default)	default	Edit
website:	YES (default)	default	Edit

Built-in

Public Signup: OFF (On)

Authentication settings

[Setup Crowd authentication](#)

[Setup host authentication](#)

[Setup LDAP](#)

[Setup AJP13 authentication](#)

[Setup Custom authentication](#)

Tuning FishEye performance

This page describes a number of ways in which you can optimise the performance of your FishEye installation.

Configurable indexing threads

FishEye is now multi-threaded, allowing you to control the number of threads dedicated to the repository

indexing process. See the page on [Configuring indexing](#).

On this page:

- [Configurable indexing threads](#)
- [Java heap size](#)
- [Default number of open files](#)
- [Improve browse performance](#)
- [Improving initial scan performance](#)

Related pages:

- [Best practices for FishEye configuration](#)

Java heap size

The heap size of the FishEye Java Virtual Machine is controlled by the [FISHEYE_OPTS](#) environment variable. The best heap size to use is dependent on a number of factors including:

- The source code management (SCM) system being used. Subversion scanning typically uses more memory than CVS, for example.
- The complexity of operations in the repository. Processing changesets which affect many files will use more memory.
- The amount of physical RAM in the system. If the Java heap is too large, it may induce swapping which will impact performance.

FishEye will reserve a portion of the available heap for caching of database data. So in general, the more memory you can supply, the better.

For Subversion repositories, it is also possible to reduce FishEye's memory footprint by reducing the [BlockSize](#) parameter.

If you do run into 'Out of Memory' errors, see [Fix out of Memory Errors](#).

Default number of open files

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

Improve browse performance

For users with very large repositories, you may find the rendering of the Subversion tree slow. You can try disabling checking for empty folders by using the [disable-dirtree-empty-checks](#) option in your [FISHEYE_ARGS](#) environment variable.

Improving initial 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 [svnserve based server](#).

Best practices for FishEye configuration

1. Ensure your FishEye scan performance is as fast as possible.

Use the file:// protocol for fastest indexing performance. See [Improve FishEye scan performance](#) for more information. (If you cannot install FishEye on the server where Subversion is running, use [svnsync](#) to mirror the repository onto the FishEye server).

2. If your repository is really large, consider starting at a sensible revision

This affects Subversion and Perforce. Do this when defining your [Perforce](#) or [SVN](#) repository.

3. Exclude directories if you don't need them.

For example, not everyone may need to access a developer's personal branch on the repository, so you can exclude it from the repository scan. You may also want to exclude large branches/tags that have been deleted

(even though they are deleted in your repository, FishEye will still index them as they once existed). Do this [by using 'Allow' and 'Exclude' Admin settings](#).

4. Consider skipping Perforce Label processing if not important

- Perforce Labels can be slow to process, and thus cause FishEye to index slowly in certain environments.
- Do this by defining the ['Skip Labels' Repository Detail](#).

5. Split your repositories into logical components if you can (For example, by product or project).

- A logical structure will make it simpler to exclude certain branches when they become less relevant to work in progress. This can garner significant performance gains.
- Avoid treating an SCM like a file system — don't alter the structure or move items around without a significant reason for doing so. Make these changes sparingly and as infrequently as possible.
- The more often you make major changes to the structure inside your SCM, the more scanning is required for FishEye to keep track of its status. This especially applies to Subversion, because of its concept of 'cheap copies'. The result is that small changes can be essentially unmeasurable and cause a large amount of re-scanning.

6. Decide on your Subversion tag and branch conventions

Decide what conventions you are going to adopt for your subversion repositories and then stick to them. It's best to stick to one of the standard conventions recommended by Subversion.

7. Exclude tags and/or branches that you delete and recreate often.

You may have a branch or tag that you delete and recreate often, for example a latest tag which holds the latest release. FishEye will take a long time to index this tag/branch as it needs to index its entire history, which can be very large. It is recommended that you exclude this directory from being indexed. See the [documentation on the 'Excludes' option](#).

8. Avoid using the text \$Log\$ in your CVS commit messages.

This is because FishEye does not handle the \$Log RCS expansion keyword correctly. Some diff results (and line numbers in diffs) may appear incorrect in files where \$Log is used.

9. Avoid using symbolic links to refer to your FishEye home directory location.

See this existing [Knowledge Base document](#) for more information.

10. Configure your index threads & memory usage to an appropriate level.

See the page [Tuning FishEye](#) for instructions and the related memory guide, [Fix out of Memory Errors](#).

See Also

- [Improve FishEye scan performance](#)
- [Tuning FishEye](#)

Separating FishEye instance data from application data

By default, FishEye will install and run self-contained within the /<FISHEYE_HOME> directory – the directory where you unzip the package. Alternatively, you can create another "FishEye Instance" directory outside of the /<FISHEYE_HOME> directory and reference it with an environment variable, [FISHEYE_INST](#). This may be necessary for practical or administrative reasons. Here are our recommendations for some of these possible scenarios:

Number of Instances	Installation Type	Upgrade Likelihood	Separate FishEye Instance Directory
Single	Production		Highly Recommended
Single	Development	Likely	Highly Recommended
Single	Development	Not Likely	Not Recommended

In order to run multiple instances of FishEye on the same server, it is **mandatory** to establish a separate [FISHEYE_INST](#) directory.

Number of Instances	Installation Type	Upgrade Likelihood	Separate FishEye Instance Directory
Multiple	Prod. / Dev.	N/A	Mandatory

See the differences between FishEye [folder layouts](#) depending on which installation you choose.

- i** As outlined in [Method 3](#) of our [FishEye upgrade guide](#), you can separate your FishEye instance data (to make future upgrades easier) even if initially you chose not to do so:
1. Shut down the FishEye/Crucible instance if it is running.
 2. Create a directory outside of the /<FISHEYE_HOME> path for the FishEye instance data and then set up the [FISHEYE_INST](#) environment variable.
 3. Copy the following items to /[FISHEYE_INST](#):
 - /<FISHEYE_HOME>/config.xml
 - /<FISHEYE_HOME>/var
 - /<FISHEYE_HOME>/cache
 - /<FISHEYE_HOME>/data (**i** If your data folder is inside /<FISHEYE_HOME>/var, keep it inside folder var so that its final location is /FISHEYE_INST/var/data).
 4. Start the FishEye/Crucible instance.

Can't See Your Repositories?

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 FISHEYE_HOME and FISHEYE_INST. Check your FISHEYE_INST i s pointing to the right directory.

Contacting Support

Raising a Support Request

There are two ways to raise a support request with Atlassian:

- (Recommended) Raise a support ticket directly via our [support site](#) on the internet and then create a [supp](#)

[ort zip](#). The advantage of this method, is that it includes all the relevant files fisheye/crucible support need. You can also be sure that the support case has been created and includes your logs.

- Complete the [support request form](#) via your Fisheye **Administration Console**. The disadvantage of this method, is that your mail may not be forwarded correctly due to an issue or security restriction on your mail server. For example, the zip of your log files are huge, hence your mail server rejects the mail.

Both methods are described below.

Raising a Support Ticket via the Internet

Raise a support ticket via the [Atlassian Support System](#):

1. If you do not already have a free Atlassian support account, [create one here](#).
2. Lodge a detailed description of your problem in the new support ticket.
3. Fill in all applicable information about your system, such as application server, database, etc.
4. If Fisheye is running, go to the '**Support/Sys Info**' screen in your **Administration Console** and copy the text of your system information into the ticket.
5. Go to Administration > Server Settings > Debug Logging, and turn debug logging "ON".
6. Reproduce the problem
7. Create a [support zip](#) to attach to the ticket. If your instance does not start up, attach a zip of your FISHEY E_INST/var/logs directory to the support case.
8. Log in to <https://support.atlassian.com> and select '**Create New Issue**'.
9. Once your ticket is lodged, wait to be notified by email of updates.

Support Utility

i It is recommended that you add a support zip to every interaction with support. The utility will also dump your system information to the logs before zipping them.

You can also use this method to append system information to an existing support ticket.

1. Log in with as a person with admin access
2. Go to Administration > Sys Info/Support. Ensure that everything is checked, then click on "Create Support Zip" button.
3. Attach the created support zip, to the support case [you raised](#).

A screenshot of the Create Support Zip form is shown below:

Admin Menu

- Repository Settings**
 - Repository List (new...)
 - Repository Defaults
- Project Settings**
 - Project List (new...)
- Global Settings**
 - Server Settings
 - Security
 - Users
 - User Mapping
 - Avatar Settings
 - Groups
 - Administrators
 - ViewCVS URL Mappings
 - Change Admin Password
 - Customize Crucible Defect Classifications
 - Permission Schemes
 - Trusted Applications
 - JIRA Servers
 - Customize Front Page
- System**
 - Database Configuration
 - Sys-Info/Support
 - Content
 - Backup

Support

Context sensitive help is available via the *Help* link in the top right hand corner of any FishEye and Crucible screen.

Complete FishEye and Crucible reference documentation is available [here](#).

For any support queries, please log a request in our online support system, or submit the following form.

Create Support Zip

Create a support zip to attach to your support case.
Note: Please be patient. It may take a few minutes depending on how long it takes to zip up your data to a file.

- ☒ Attach FishEye and Crucible Logs
Zip and attach the log files that FishEye and Crucible generates (recommended).
- ☒ Attach Config File
Zip and attach your config file (recommended).
Atlassian will handle your data with strict confidence and will not disclose it to any third parties.
NOTE: Your config file is NOT anonymised.
- ☒ Attach Plugins Configuration
Zip and attach your plugins config files (recommended).
Atlassian will handle your data with strict confidence and will not disclose it to any third parties.
NOTE: Your plugins config file is NOT anonymised.
- ☒ Attach Thread Dump(s)
Zip and attach a thread dump(s) for this instance of FishEye and Crucible (recommended).

[Create Support Zip](#)

Raising a Support Ticket via the Fisheye Administration Console

i Ensure that [SMTP email](#) is set up on your Fisheye instance and your mail server allows zip files.

The advantage of this method is that it is convenient, however the disadvantage is that your mail may not be forwarded correctly due to an issue (e.g. zip file too large) or security restriction on your mail server.

You can also use this method to append system information to an existing support ticket.

On the left navigation bar, click '**Sys-Info/Support**'.

Screenshot: The Sysinfo/Support Menu Option



The **System Info/Support** page loads. On this page, you can fill out a web form which will automatically send an email to Atlassian Support, attaching your FishEye logs and configuration file (if you wish).

i This functionality requires that the FishEye web server is already set up and capable of sending email.

Screenshot: The System Info/Support page

 A screenshot of a web form titled 'Support Request'. The form contains several fields and options:

- *Subject:** A text input field with a placeholder 'Enter a one line summary of the problem.'
- Priority:** A dropdown menu currently set to 'Medium'.
- *Description:** A large text area for a detailed description.
- Existing Support Request:** A text input field with instructions: 'To create a new support request, simply leave this field blank. Or: To attach the information on this page to an existing support request, please enter the relevant issue key (eg. FSP-123).'
- Contact Name:** A text input field.
- Contact Email:** A text input field with instructions: 'Enter a comma-separated list of email addresses'.
- Contact Number:** A text input field.
- ☒ **Attach Fisheye Logs**: A checkbox with instructions: 'Zip and attach the log file that FISHEYE generates (recommended).'
- ☒ **Attach Config File**: A checkbox with instructions: 'Zip and attach your config file(recommended). Atlassian will handle your data with strict confidence and will not disclose it to any third parties. NOTE: Your config file is NOT anonymised.'
- Send Support Request**: A button at the bottom left.

Fields in the Support Request form

On the **Support Request** form there are a number of fields to fill out and options to select.

- **'Subject':** Enter a one line summary of the problem.
- **'Priority':** Choose from Low, Medium, High or Critical.
- **'Description':** Type a detailed description of the problem you are trying to solve.
- **'Existing Support Request':** Leave this field blank to create a new support request. If you have an existing support ticket open at support.atlassian.com, enter the issue key here. This will append this request to the existing ticket.
- **'Contact Name':** Your contact name.

- **'Contact Number'**: Your contact number.
- **'Attach FishEye Logs'**: Tick this box to send Atlassian your FishEye log file.
- **'Attach Config File'**: Tick this box to send Atlassian your FishEye configuration file.
- **'Support Entitlement Number (SEN)'**: Paste your Support Entitlement Number. See [Finding your FishEye or Crucible Support Entitlement Number \(SEN\)](#).

When you have filled out the required fields, click **'Send Support Request'** to finish.

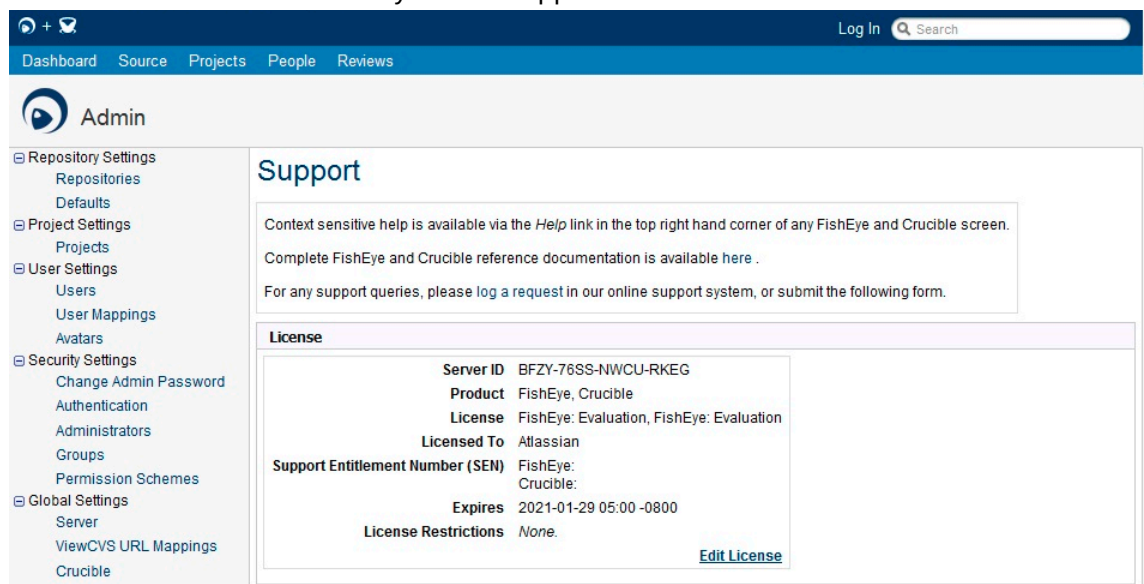
⚠ Note that the form may take several minutes to fully submit, as it takes some time to export and process the log file data.

Finding your FishEye or Crucible Support Entitlement Number (SEN)

There are three ways to find your Support Entitlement Number (SEN):

▼ Method 1: Check in the FishEye Administration Interface

Select Administration >> SysInfo/Support. The SEN is shown:



The screenshot shows the FishEye Administration Interface. The top navigation bar includes 'Dashboard', 'Source', 'Projects', 'People', and 'Reviews'. The left sidebar shows the 'Admin' section with various settings. The main content area is titled 'Support' and contains the following information:

License	
Server ID	BFZY-76SS-NWCU-RKEG
Product	FishEye, Crucible
License	FishEye: Evaluation, FishEye: Evaluation
Licensed To	Atlassian
Support Entitlement Number (SEN)	FishEye: Crucible:
Expires	2021-01-29 05:00 -0800
License Restrictions	None.

An [Edit License](#) link is visible at the bottom right of the license table.

▼ Method 2:
Log into
my.atlas
sian.co
m as the
Account
Holder
or
Technic
al

Contact

Your Support Entitlement Number is available in [<http://my.atlassian.com>]:

Support Entitlement Number

Welcome

Hey [redacted] you've been an Atlassian customer for 3 years now. Thank you for your support! You can manage your 8 licenses below. For questions relating to managing your account, licensing or purchasing, please see the [purchasing and licensing FAQ](#) or [contact a customer service representative](#). To request technical support, please visit our [Support Portal](#).

Licenses [New Evaluation License](#) | [Expand All](#) | [Collapse All](#)

Product	Name	Support Expires	Support
☒ Confluence 2 Users: Personal Server	[redacted]	02 Jun 2009	
☒ Confluence 500 Users: Commercial License	Atlassian	10 Apr 2011	Request Support

Name : Atlassian (edit)

SEN: SEN- [redacted]

Server ID: [redacted] (edit)

☒ Technical Contact: [redacted]

Email:

☒ Billing Contact: [redacted]

Email:

Licensed To: Atlassian

License: [redacted]

This license key is compatible with Confluence 3 or above. If you are using an older version of Confluence, you can [downgrade your license key](#).

Actions: [Renew](#) | [Upgrade](#) | [Download Source](#) | [View Developer License](#)

History:

10 Apr 2008	Initial Purchase
10 Apr 2009	Maintenance Renewal
01 Jun 2009	Converted to Confluence 500 Users: Commercial License
10 Apr 2010	Maintenance Renewal

Method 3: Atlassian Invoice

Your Support Entitlement Number (SEN) appears on the third page of your Atlassian Invoice.

See [Finding Your Support Entitlement Number](#) in the support space for more general information about how Atlassian Support uses this number.

Configuring the database connection pool

This page describes how to change the default database connection pool settings used by FishEye and Crucible.

FishEye and Crucible use the [BoneCP](#) connection pool manager to manage connections to the database. The BoneCP connection pool manager has been preconfigured in FishEye and Crucible to work out-of-the-box for most customers. However, you can override any of the default settings to fine tune performance, if you wish. For example, you may want to try increasing the minimum and/or maximum number of connections, if you are getting significant delays creating connections for clients.

On this page:

- [Setting Minimum and Maximum Database Connections](#)
- [Advanced: Tuning The Connection Pool via Properties](#)

Related pages:

- [Migrating to an external database](#)

Setting Minimum and Maximum Database Connections

To configure the database connection pool:

1. Log in to FishEye or Crucible as an administrator.
2. In the Admin area, click Database (under 'System Settings').
3. Click **Edit**.
4. Enter new values for the **Minimum Pool Connections** and **Maximum Pool Connections** fields.
5. *Advanced users only.* Enter any overrides of BoneCP properties in the **Parameters** field. Enter each parameter-value pair on a new line in the **Parameters** field. See the section below for details of some common properties you may wish to configure.
6. Click **Save**.

Screenshot: Database configuration screen in FishEye / Crucible

Database Configuration

 To switch to a different database, specify the database's configuration settings in the form below and use the *Test Connection* button.
 Changes to connection pool sizes require a full restart before the new settings will take effect.
 FishEye and Crucible are currently connected to the *SQLServer 2005* database at `jdbc:jtds:sqlserver://fecru-mssql2005-vm.private.atlassian.com:1433;databaseName=crucible;`

Configuration

Type SQLServer 2005

Driver Location User Supplied - FISHEYE_INST/lib

URL

User Name

Password

Minimum Pool Connections:

Maximum Pool Connections:

Parameters:

Test Connection

Save Save & Migrate Cancel

Advanced: Tuning The Connection Pool via Properties

The following table shows some properties you can use to tune the database connection pool used by FishEye and Crucible. You can override any of these parameters by entering parameter-value pairs (on separate lines) into the **Parameters** field of the 'Database Configuration' dialog, as described above.

 Refer to the [BoneCP API javadoc](#) for a full list and description of BoneCP properties.

Parameter	Property	Default Value	Description
Connection timeout	<code>bonecp.idleMaxAgeInMinutes</code>	<code>bonecp.idleMaxAgeInMinutes=0</code>	Time to wait before dropping idle connections, in minutes.
Connection test period	<code>bonecp.idleConnectionTestPeriodInMinutes</code>	<code>bonecp.idleConnectionTestPeriodInMinutes=60</code>	Database connections are kept alive by sending ping requests on idle connections. This value sets the time between pings, in minutes.

Connection pool partition count	<code>bonecp.partitionCount</code>	<code>bonecp.partitionCount=3</code>	The database connection pool is split into a number of equal-sized "partitions" to increase concurrent performance. This property sets the number of partitions to use. Note that if you set this property, but not the <code>bonecp.minConnectionsPerPartition</code> or <code>bonecp.maxConnectionsPerPartition</code> properties below, FishEye will use the "Minimum Pool Connections" and "Maximum Pool Connections" field settings to calculate the effective minimum and maximum connections per partition. These values will be calculated to ensure that the field settings are not exceeded.
Max and min number of connections per partition	<code>bonecp.minConnectionsPerPartition</code> <code>bonecp.maxConnectionsPerPartition</code>	Calculated based on value of <code>bonecp.partitionCount</code> and the "Minimum Pool Connections" and "Maximum Pool Connections" field settings	These properties control the minimum and maximum number of connections <i>in each partition</i>. To calculate the <i>total</i> number of connections in the pool: Total minimum connections = <code>bonecp.partitionCount</code> * <code>bonecp.minConnectionsPerPartition</code> Total maximum connections = <code>bonecp.partitionCount</code> * <code>bonecp.maxConnectionsPerPartition</code> Note that setting these properties overrides the "Minimum Pool Connections" and "Maximum Pool Connections" field settings.

Connection acquire increment	<code>bonecp.acquireIncrement</code>	<code>bonecp.acquireIncrement=2</code>	The batch size used when BoneCP will acquire new connections.
Statement cache size	<code>bonecp.statementsCacheSize</code>	<code>bonecp.statementsCacheSize=50</code>	The statement cache size defaults to 50, except when using an Oracle database, for which it defaults to zero to avoid having too many open cursors .
Connection test query	<code>bonecp.connectionTestStatement</code>	By default boneCP will use metadata request	The query to send to the DB to maintain keep-alives and test for dead connections. This is database specific and should be set to a query that consumes the minimal amount of load on the server. Examples: • MySQL: <code>"/ * ping * V SELECT 1",</code> • PostgreSQL: <code>"SELECT NOW()".</code> If you do not set this, then BoneCP will issue a metadata request instead that should work on all databases but is probably slower. (Note: In MySQL, prefixing the statement by <code>/ * ping * V</code> makes the driver issue 1 fast packet instead. See http://blogs.sun.com/SDNChannel/entry/mysql_tips_for_java_developers)

FishEye Release Notes



FishEye 2.10 has now been released. Read the [Release Notes](#).

FishEye release notes and changelogs

- [FishEye upgrade guide](#)
- [FishEye release summary](#)
- [FishEye 2.10 release notes](#)

- [FishEye 2.10 changelog](#)
- [FishEye 2.10 upgrade guide](#)
- [FishEye 2.9 release notes](#)
 - [FishEye 2.9 changelog](#)
 - [FishEye 2.9 upgrade guide](#)
- [FishEye 2.8 release notes](#)
 - [FishEye 2.8 changelog](#)
 - [FishEye 2.8 upgrade guide](#)
 - [Upgrading from FishEye 2.7 with existing ClearCase Repositories](#)
- [FishEye 2.7 Release Notes](#)
 - [FishEye 2.7 Changelog](#)
 - [FishEye 2.7 Upgrade Guide](#)
- [FishEye 2.6 Release Notes](#)
 - [FishEye 2.6 Changelog](#)
 - [FishEye 2.6 Upgrade Guide](#)
- [FishEye 2.5 Release Notes](#)
 - [FishEye 2.5 Changelog](#)
 - [FishEye 2.5 Upgrade Guide](#)
- [FishEye 2.4 Release Notes](#)
 - [FishEye 2.4 Changelog](#)
 - [FishEye 2.4 Upgrade Guide](#)
- [FishEye 2.3 Release Notes](#)
 - [FishEye 2.3 Changelog](#)
 - [FishEye 2.3 Upgrade Guide](#)
- [Older release notes](#)
 - [FishEye 2.2 Release Notes](#)
 - [FishEye 2.2 Changelog](#)
 - [FishEye 2.2 Upgrade Guide](#)
 - [FishEye 2.1 Release Notes](#)
 - [FishEye 2.1 Changelog](#)
 - [FishEye 2.1 Upgrade Guide](#)
 - [FishEye 2.0 Release Notes](#)
 - [FishEye 2.0 Changelog](#)
 - [FishEye 2.0 Upgrade Guide](#)
 - [FishEye 2.0 Beta Release Notes](#)
 - [Upgrading to the FishEye 2.0 Beta](#)
 - [FishEye 2.0 Beta Reviewer's Guide](#)
 - [JIRA Integration in FishEye 2.0 Beta](#)
 - [Git Alpha in FishEye 2.0 Beta](#)
 - [FishEye 1.6 Release Notes](#)
 - [FishEye 1.6 Changelog](#)
 - [FishEye 1.6 Upgrade Guide](#)
 - [FishEye 1.5 Release Notes](#)
 - [FishEye 1.5 Changelog](#)
 - [FishEye 1.5 Upgrade Guide](#)
 - [FishEye 1.4 Release Notes](#)
 - [FishEye 1.4 Changelog](#)
 - [FishEye 1.4 Upgrade Guide](#)
 - [FishEye 1.3 Release Notes](#)
 - [FishEye 1.3 Changelog](#)
 - [FishEye 1.3 Upgrade Guide](#)
- [Security advisories](#)
 - [FishEye and Crucible Security Advisory 2012-08-21](#)
 - [FishEye and Crucible Security Advisory 2012-05-17](#)
 - [FishEye and Crucible Security Advisory 2012-01-31](#)
 - [FishEye and Crucible Security Advisory 2011-11-22](#)
 - [FishEye and Crucible Security Advisory 2011-05-16](#)

- [FishEye and Crucible Security Advisory 2011-01-12](#)
- [FishEye Security Advisory 2010-10-20](#)
- [FishEye Security Advisory 2010-06-16](#)
- [FishEye Security Advisory 2010-05-04](#)
- For changes prior to 1.3, see:
 - [1.2.x Changelog](#)
 - [1.1.x Changelog](#)
 - [1.0.x Changelog](#)

Installation

You can now download FishEye from [here](#). Read about [installing FishEye](#) .

If upgrading from a previous version, please follow the [Upgrade Guide](#).

FishEye upgrade guide

This page describes the recommended method of upgrading to a new version of FishEye.

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.

Before you start

- 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 Changelog](#) and 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](#).

On this page:

- [Before you start](#)
- [Upgrade procedure](#)
 - [Method 1: Using a FISHEYE_INST directory](#)
 - [Method 2: Without a FISHEYE_INST directory](#)
 - [Method 3: Without a FISHEYE_INST directory, but would like to set one up](#)
- [Checking for known issues and troubleshooting the FishEye upgrade](#)

Upgrade procedure

Your upgrade procedure depends on whether you are using a FISHEYE_INST directory (i.e. "FishEye instance" directory).

- The FISHEYE_INST directory is the FishEye data directory (*not* the installation directory) and has a location defined by the [FISHEYE_INST](#) environment variable. It is used to keep the FishEye data completely separate from the FishEye/Crucible application files. We recommend that you configure FishEye/Crucible to use a FISHEYE_INST directory for production instances. Read more about FISHEYE_INST in [Installing FishEye on Windows](#) or [Installing FishEye on Linux and Mac](#).
- The <FishEye home directory> is the location of the FishEye/Crucible application files.

NOTE: For all methods below: if you previously ran FishEye as a Windows service and are installing the new version of FishEye in a new location, you need to reinstall FishEye as a Windows service. Make sure to run ...\\wrapper\\bin\\FishEye-Uninstall-NTService.bat from the old location. Subsequently, prior to starting the new instance of FishEye, follow these instructions to set up [FishEye as a Windows service](#) again. Make sure to copy over the changes from the old ...\\wrapper\\conf\\wrapper.conf file to the new wrapper.conf file.

Method 1: Using a FISHEYE_INST directory

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.
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 Release Notes/Crucible Release Notes](#).

Method 2: Without a FISHEYE_INST directory

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 instance if it is running.
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>/var.
7. Copy (or move) the <FishEye home directory>/cache directory to <New FishEye home directory>/cache.
8. Start FishEye/Crucible from the new installation by running <New FishEye home directory>/bin/run.sh. (Use run.bat on Windows.)
9. Follow any version-specific instructions found in the [FishEye Release Notes/Crucible Release Notes](#).

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 instance if it is running.
3. Set up the FISHEYE_INST [environment variable](#), then create the FISHEYE_INST directory on your file system.
4. Copy <FishEye home directory>/config.xml to the FISHEYE_INST directory.
5. Copy the <FishEye home directory>/var directory to the FISHEYE_INST directory.
6. Copy the <FishEye home directory>/cache directory to the FISHEYE_INST directory.
7. Extract the new FishEye/Crucible archive into a directory such as <New FishEye home directory>.
8. 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.
9. Follow any version-specific instructions found in the [FishEye Release Notes/Crucible Release Notes](#).

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 release summary

FishEye 2.10 — 15 January 2013

- Repository indexing REST API
- Bitbucket and GitHub polling integration
- More in the [release notes](#).

FishEye 2.9 — 14 November 2012

- Simpler JIRA integration
- More JIRA data in FishEye
- Faster JIRA source tab
- More in the [release notes](#).

FishEye 2.8 — 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 2.7 — 7 September 2011

- Managed Git Repositories
- Smart Commits
- Web Hooks
- JIRA FishEye Plugin Improvements
- Small Improvements
- More in [release notes](#).

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 [release notes](#).

FishEye 2.5 — 8 February 2011

- Search Revamp
- Redesigned Activity Stream
- Mercurial and Git Authentication
- RSS Improvements
- Universal Plugin Manager
- More in [release notes](#).

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 [release notes](#).

FishEye 2.3 — 26 May 2010

- Mercurial SCM Alpha
- New 'Aggregate' functions in EyeQL query language
- Revamped Installation Process
- More in [release notes](#).

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 [release notes](#).

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 [release notes](#).

FishEye 2.0 — 30 June 2009

- Activity streams
- People statistics
- Favourites, bookmarks & saved search
- Enhanced JIRA integration
- New user interface
- Git beta
- More in [release notes](#).

FishEye 1.6 — 23 September 08

- FishEye search enhancements
- Multiple admin users
- Remote API improvements
- Changes to charts
- Performance tweaks
- More in [release notes](#).

FishEye 1.5 — 14 April 2008

- Per-author lines of code statistics
- Charting improvements
- Customisable email templates
- More in [release notes](#).

FishEye 1.4 — 5 December 2007

- Enhancements to user management
- Crowd/SSO support
- Crucible integration
- Enhancements to JIRA plugin
- More in [release notes](#).

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)
- More in [release notes](#).

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

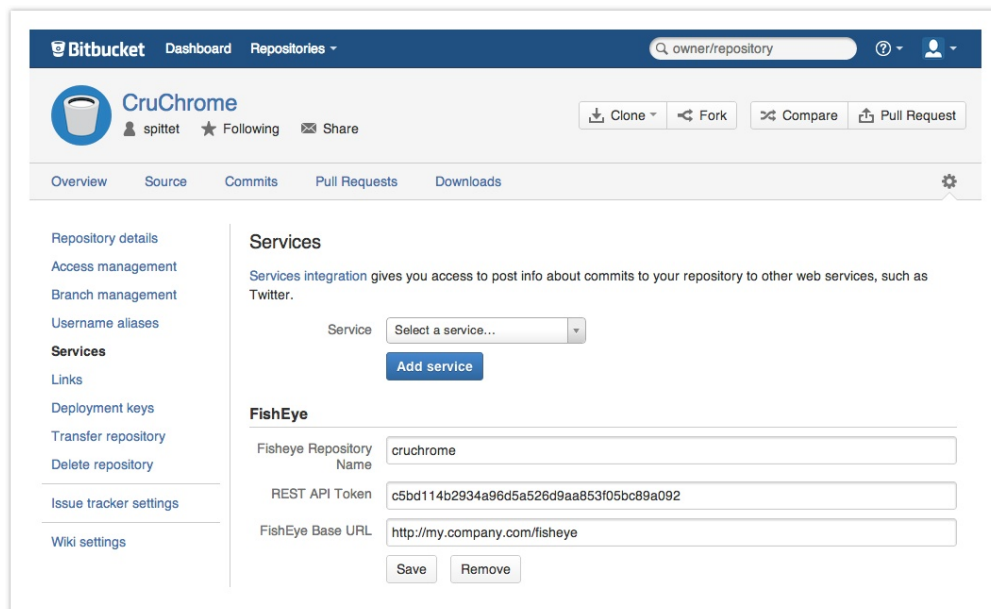
With FishEye 2.10, you can now optimize your instance by using the REST API to trigger indexing when it's necessary. Instead of configuring FishEye to poll each repository every minute, you can add a post-commit or post-receive hook to your repositories which will ping FishEye when new commits need to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



3

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 wicki
 Richard Stephens
 Tom Davies
 Valery Trubnikov

Team lead

Nick Pellow

Product management

Sten Pittet

Project manager

Anton Mazkovoï

Support

Ajay Sridhar
Daniel Rohan
Douglas Fabretti
Felipe Kraemer
Gurleen Anand
Kah Loun Foong
Leonardo Macedo
Malik Mangier
Patrick Hill
Renan Battaglin
Rene Verschoor
Ricardo Martins

Others**Product marketing**

Giancarlo Lionetti

Quality assurance

George Filippoff

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](#).

FishEye 2.10.0

15 January 2013

JIRA Issues (29 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-4191	Add built-in support for use of proxy to contact external plugins site	Unassigned	Dana Cleveland	↓	 Closed	Fixed	Jul 12, 2012	Dec 18, 2012
	FE-4147	Extend REST API to include starting and stopping a repository and to allow for re-indexing	Lukasz Pater [Atlassian]	Tom Towach [Atlassian]	↓	 Closed	Fixed	Jun 18, 2012	Jan 15, 2013
	FE-4398	Checkout errors can terminate svn repo import process	Unassigned	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Dec 10, 2012	Jan 14, 2013
	FE-4361	[CVS] Allow pollPeriod to be configurable	Lukasz Pater [Atlassian]	Brendan Humphreys [Atlassian]	↓	 Closed	Fixed	Nov 11, 2012	Jan 11, 2013
	FE-4227	'Download archive' (aka tarball): name of zip is truncated to 20 characters	Piotr Swiecicki [Atlassian]	Rene Verschuur [Atlassian]	↓	 Closed	Fixed	Aug 10, 2012	Oct 23, 2012
	FE-4083	Review smart commit should	Lukasz Pater [Atlassian]	Andrew Byala	↓	 Closed	Fixed	May 01, 2012	Jan 15, 2013

		populate Crucible Objectives							
	FE-4416	repository refresh status button is broken	Conor MacNeill [Atlassian]	Nick Pellow [Atlassian]	↓	 Closed	Fixed	Dec 19, 2012	Jan 08, 2013
	FE-4404	DiffTextCache files not deleted	Unassigned	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Dec 16, 2012	Jan 10, 2013
	FE-4399	CVS scanner activates repository on every poll, causing high native memory usage	Unassigned	Lukasz Pater [Atlassian]	↑	 Closed	Fixed	Dec 11, 2012	Jan 10, 2013
	FE-4394	The repositoryes admin page is slow with many repositoryes enabled	Unassigned	Lukasz Pater [Atlassian]	↓	 Closed	Fixed	Dec 04, 2012	Dec 21, 2012
	FE-4392	Svnkit fine logging causes very large debug logs	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Dec 03, 2012	Dec 10, 2012
	FE-4391	Validation of start-rev in admin is incorrect	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Dec 03, 2012	Dec 10, 2012
	FE-4371	Fisheye not handling empty git or hg changes	Lukasz Pater [Atlassian]	Gurleen Anand [Atlassian]	↓	 Closed	Fixed	Nov 20, 2012	Jan 07, 2013

		ets properly							
	FE-4368	Changes et Comment re-indexing incorrectly uses a threadlocal session.	Brendan Humphreys [Atlassian]	Brendan Humphreys [Atlassian]		 Close	Fixed	Nov 19, 2012	Dec 10, 2012
	FE-4357	Improve the way permissions are handled, as a lot of time is spent in user/repo permission checks, which should be cached	Brendan Humphreys [Atlassian]	Geoff Crain [Atlassian]		 Close	Fixed	Nov 06, 2012	Jan 14, 2013
	FE-4354	OSGI cache should be excluded from backup	Piotr Swiecicki [Atlassian]	Piotr Swiecicki [Atlassian]		 Close	Fixed	Nov 05, 2012	Nov 12, 2012
	FE-4351	Indexing stalls if parsing Perforce multi print output throws an exception	Lukasz Pater [Atlassian]	Lukasz Pater [Atlassian]		 Close	Fixed	Nov 02, 2012	Nov 13, 2012
	FE-4350	FishEye not fully compatible with Mercurial 2.3	Richard Stephens [Atlassian]	Lukasz Pater [Atlassian]		 Close	Fixed	Nov 02, 2012	Dec 13, 2012
	FE-4349	Disable directory Listing	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]		 Close	Fixed	Nov 01, 2012	Nov 08, 2012

	FE-4332	"The Compressed (zipped) Folder is invalid or corrupted" error in IE8 when downloading tarball from "Download Archive" link	Piotr Swiecicki [Atlassian]	Piotr Swiecicki [Atlassian]	↓	 Closed	Fixed	Oct 18, 2012	Oct 23, 2012
	FE-4317	Provide a mechanism to enable Jetty JMX Beans	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Oct 08, 2012	Dec 18, 2012
	FE-4315	Restore does not set key permissions	Richard Stephens [Atlassian]	Conor MacNeill [Atlassian]	↑	 Closed	Fixed	Oct 07, 2012	Nov 07, 2012
	FE-4207	Fisheye fails to start if the FISHEYE_INST does not contain a config.xml	Val Trubnikov [Atlassian]	Richard Stephens [Atlassian]	↓	 Closed	Fixed	Jul 30, 2012	Nov 23, 2012
	FE-4206	Product update check doesn't support proxy authentication	Richard Stephens [Atlassian]	Tom Davies [Atlassian]	↓	 Closed	Fixed	Jul 30, 2012	Nov 04, 2012
	FE-4195	Non-Crucible users are shown	Maciej Swinarski [Atlassian]	Tom Davies [Atlassian]	↑	 Closed	Fixed	Jul 15, 2012	Dec 04, 2012

		create review links							
	FE-4126	"Select/deselect all users" Checkboxes Not Visible	Piotr Swiecicki [Atlassian]	Daniel Rohan [Atlassian]	↓	 Close	Fixed	Jun 04, 2012	Jan 06, 2013
	FE-4091	Incorrect time indication on comment section when viewing revision details	Piotr Swiecicki [Atlassian]	Alexander Kisliak [Atlassian]	↓	 Close	Fixed	May 10, 2012	Oct 24, 2012
	FE-3757	Work Log Description on Not Populated with Smart Commit	Val Trubnikov [Atlassian]	Greg Carnie	↓	 Close	Fixed	Oct 02, 2011	Nov 19, 2012
	FE-3570	Cannot rename or delete a project if Crucible has been disabled	Val Trubnikov [Atlassian]	Gurleen Anand [Atlassian]	↑	 Close	Fixed	Jul 07, 2011	Dec 02, 2012

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

- 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 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 2.10 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 straight forward. *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!

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.

today

 **Nick Pellow**

 **4d2845d** changed 0 files  bundled-plugins

MERGE: from default

▼  **912baf5** changed 54 files  bundled-plugins

[FECRU-2735](#): version all bundled plugins with the same version number as fisheye

 **FECRU-2735**: Move bundled plugins into the fisheye...

Status: Quality Review

Reporter: Nick Pellow [Atlassian]

Assignee: Nick Pellow [Atlassian]

[report-1.14.22.jar](#)

[report-2.10.0-m1-SNAPSHOT.jar](#)

[system-scm-plugin-1.6.27.jar](#)

[system-scm-plugin-2.10.0-m1-SNAPSHOT.jar](#)

binary  [/lib/bundled-plugins/crucible-git-scm-plugin-1.3.27.jar](#)

binary  [/lib/bundled-plugins/crucible-git-scm-plugin-2.10.0-m1-SNAPSHOT.jar](#)

binary  [/lib/bundled-plugins/crucible-p4-scm-plugin-1.6.27.jar](#)

binary  [/lib/bundled-plugins/crucible-p4-scm-plugin-2.10.0-m1-SNAPSHOT.jar](#)

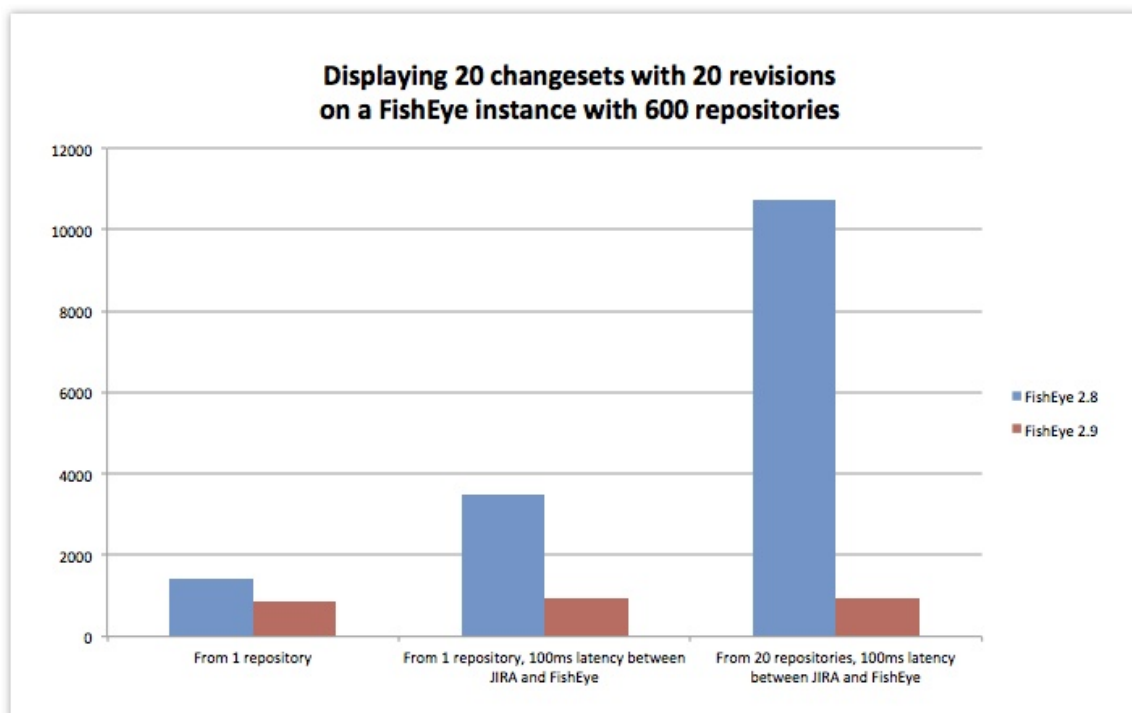
binary  [/lib/bundled-plugins/crucible-svn-scm-plugin-1.7.27.jar](#)

binary  [/lib/bundled-plugins/crucible-svn-scm-plugin-2.10.0-m1-SNAPSHOT.jar](#)

3

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.



4

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 Links 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 wicki
Richard Stephens
Tom Davies
Valery Trubnikov

Team lead

Nick Pellow

Product management

Sten Pittet

Project manager

Anton Mazkovi

Support

Ajay Sridhar
Daniel Rohan
Douglas Fabretti
Felipe Kraemer
Gurleen Anand

Kah Loun Foong
 Leonardo Macedo
 Malik Mangier
 Patrick Hill
 Renan Battaglin
 Rene Verschoor
 Ricardo Martins

Others

Product marketing

Giancarlo Lionetti

Quality assurance

George Filippoff

Technical writing

Paul Watson

Operations

James Fleming

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)

Type	Key	Summar	Assigne	Reporter	Priority	Status	Resoluti	Created	Updated

		y	e			on			
	FE-4392	Svnkit fine logging causes very large debug logs	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Close d	Fixed	Dec 03, 2012	Dec 10, 2012
	FE-4391	Validation of start-rev in admin is incorrect	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Close d	Fixed	Dec 03, 2012	Dec 10, 2012
	FE-4368	Changeset Comment re-indexing incorrectly uses a threadlocal session.	Brendan Humphreys [Atlassian]	Brendan Humphreys [Atlassian]		 Close d	Fixed	Nov 19, 2012	Dec 10, 2012
	FE-4357	Improve the way permissions are handled, as a lot of time is spent in user/repo permission checks, which should be cached	Brendan Humphreys [Atlassian]	Geoff Crain [Atlassian]		 Close d	Fixed	Nov 06, 2012	Jan 14, 2013

From 2.9.0 to 2.9.1

19 November 2012

This is a bug fix release.

JIRA Issues (1 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-4364	REST calls to /rest-service/reposi	Unassigned	Lukasz Pater [Atlassian]	↓	 Closed	Fixed	Nov 14, 2012	Dec 10, 2012

		itories-v1/ and /rest-service-fe/repositories-v1/ fail when there are ClearCase repositories configured							

FishEye 2.9.0**14 November 2012**

JIRA Issues (17 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-4288	Profile viewing source files in FECRU with autolink enabled	Unassigned	Nick Pellow [Atlassian]	↓	 Closed	Fixed	Sep 17, 2012	Sep 24, 2012
	FE-4194	Enable automatic repository->Jira project entity link creation	Unassigned	David Corley	↓	 Closed	Fixed	Jul 13, 2012	Oct 23, 2012
	FE-4151	Every FeCru repo sends a REST call to Crowd with a getUserInfo function when the source tab is clicked in JIRA	Unassigned	Tom Towach [Atlassian]	↓	 Closed	Fixed	Jun 19, 2012	Oct 10, 2012
	FE-4150	Impleme	Geoff	Tom	↓	 Closed	Fixed	Jun 19,	Oct 03,

		nt a REST endpoint that can return the results for all the repos at once. This will be used by the JIRA FishEye Plugin.	Crain [Atlassian]	Towach [Atlassian]		d		2012	2012
	FE-2907	Branch selector should trim whitespace	Jeroen De Raedt [Atlassian]	Dylan Hansen [Atlassian]		 Closed	Fixed	Nov 15, 2010	Oct 22, 2012
	FE-2639	Reduce memory usage of mysql backup process	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Jun 10, 2010	Oct 31, 2012
	FE-2476	Avoid spurious package scanner warnings in the debug log	Brendan Humphreys [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Feb 21, 2010	Oct 04, 2012
	FE-1071	Restructure Fisheye so that svn operations do not occur on the main thread	Conor MacNeill [Atlassian]	Partha Kamal [Atlassian]		 Closed	Fixed	Jan 20, 2009	Sep 24, 2012
	FE-4356	"fisheycetl.sh restore" removes some database connection attributes	Lukasz Pater [Atlassian]	Zhuang Xu [Atlassian]		 Closed	Fixed	Nov 06, 2012	Jan 02, 2013

	FE-4340	SVNKit upgrade 1.7.4v1 to 1.7.6	Piotr Swiecicki [Atlassian]	Piotr Swiecicki [Atlassian]		 Closed	Fixed	Oct 30, 2012	Nov 06, 2012
	FE-4321	Replace comments sourced outside repository scope not processed	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Oct 09, 2012	Oct 31, 2012
	FE-4261	500 error received from FishEye when retrieving changesets for Source tab in JIRA	Geoff Crain [Atlassian]	Ian Grunert [Atlassian]		 Closed	Fixed	Sep 02, 2012	Oct 03, 2012
	FE-4203	Non-ASCII end of line characters in commit messages break git scanning (JVM 1.7 fixes)	Unassigned	Tom Davies [Atlassian]		 Closed	Fixed	Jul 23, 2012	Oct 31, 2012
	FE-4109	Commit graph does not link to JIRA issue if the key is surrounded by []	Richard Stephens [Atlassian]	Lee Netherton		 Closed	Fixed	May 24, 2012	Oct 09, 2012
	FE-3892	P4 JobIds are not being returned with the REST call to rest-service-fe/cha	Geoff Crain [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	Jan 17, 2012	Oct 07, 2012

		ngeset-v1/listChangesets							
	FE-3659	Slow response times in the JIRA FishEye Plugin caused by EyeQL suffering from performance issues	Unassigned	Jason Hinch [Atlassian]		 Closed	Fixed	Aug 21, 2011	Oct 23, 2012
	FE-1965	when showing a deleted image in a changeset diff. the old version / new version is kinda broken	Lukasz Pater [Atlassian]	Brendan Humphreys [Atlassian]		 Closed	Fixed	Jul 27, 2009	Oct 12, 2012

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.

On this page:

- [Upgrade notes](#)
- [Upgrade procedure](#)
- [Checking for known issues and troubleshooting the FishEye upgrade](#)

Related pages:

- [FishEye 2.9 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 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](#)

 Download

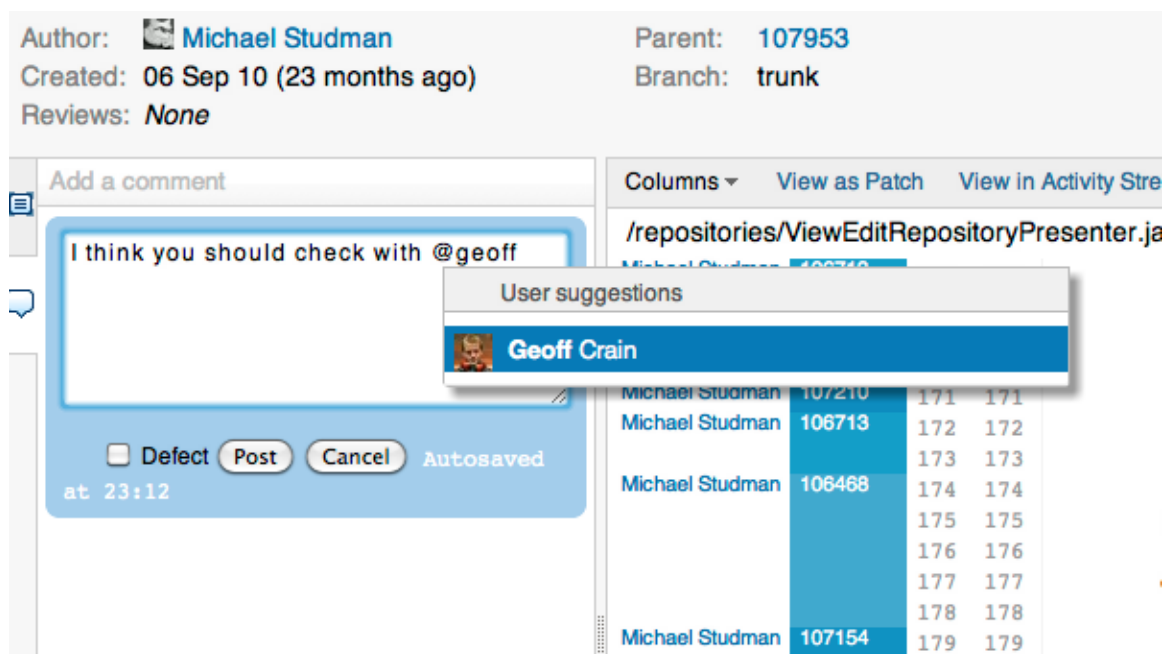
Providing feedback:

Please log your [votes and issues](#). They help us decide what needs doing, and are much appreciated!

1

Mentions

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.



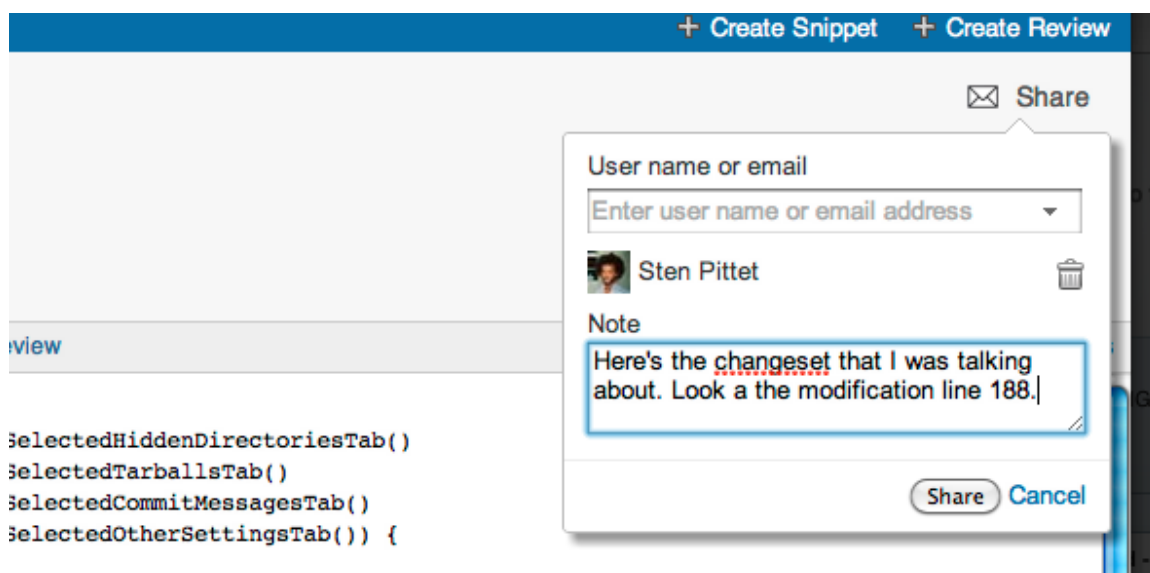
The screenshot shows the FishEye interface for a changeset. At the top, it displays the author as Michael Studman, created on 06 Sep 10 (23 months ago), with no reviews. The parent changeset is 107953 on the trunk branch. Below this, there's a comment box with the text "I think you should check with @geoff". A dropdown menu titled "User suggestions" is open, showing a list of users. The first suggestion is "Geoff Crain" with a small profile picture. To the right of the comment box, there's a table with columns for "Columns", "View as Patch", and "View in Activity Stream". The table lists several changesets by Michael Studman, including 107210, 106713, 106468, and 107154, with corresponding line numbers.

Columns	View as Patch	View in Activity Stream
/repositories/ViewEditRepositoryPresenter.java		
Michael Studman	107210	171 171
Michael Studman	106713	172 172
		173 173
Michael Studman	106468	174 174
		175 175
		176 176
		177 177
		178 178
Michael Studman	107154	179 179

2

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.



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.

5

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 Mazkovo

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

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!



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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-4291	Indexing threadpool size does not shrink	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Sep 18, 2012	Oct 04, 2012

	FE-4286	java.lang.NoClassDefFoundError: Could not initialize class com.sun.jna.Native	Brendan Humphreys [Atlassian]	Malik Mangier [Atlassian]		 Closed	Fixed	Sep 17, 2012	Oct 04, 2012
	FE-4266	Email Notifications fired from plugins fail with: java.lang.ClassCastException: com.cenqua.crucible.notification.Notification cannot be cast to java.lang.Comparable	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	Sep 04, 2012	Oct 03, 2012
	FE-4234	Fix Link for "Maximum Indexable File Size" Help	Brendan Humphreys [Atlassian]	Renan Battaglin [Atlassian]		 Closed	Fixed	Aug 16, 2012	Oct 03, 2012
	FE-4220	Group associations are not preserved for synced users with mixed-case usernames	Tom Davies [Atlassian]	Daniel Rohan [Atlassian]		 Closed	Fixed	Aug 07, 2012	Sep 24, 2012
	FE-3122	Fisheye is indexing changes	Brendan Humphreys	None		 Closed	Fixed	Nov 02, 2010	Oct 03, 2012

		et ids of GIT submodules when it shouldn't be causing	ys [Atlassian]						

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 (2 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-4238	Update database encoding warning message to be more generic	Unassigned	Gurleen Anand [Atlassian]		 Closed	Fixed	Aug 20, 2012	Aug 28, 2012
	FE-4252	Email Review option won't work	Nick Pellow [Atlassian]	Leonardo De Macedo [Atlassian]		 Closed	Fixed	Aug 24, 2012	Sep 09, 2012

FishEye 2.8.0**15 August 2012**

JIRA Issues (28 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-4094	Create 2.8.0-m1 release containing svn 1.7 support	Conor MacNeill [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	May 14, 2012	Jun 18, 2012
	FE-4179	Blitz-test pages changed in admin UI due to radio button issues/changes	Unassigned	Richard Stephens [Atlassian]		 Closed	Fixed	Jul 05, 2012	Jul 23, 2012

	FE-4218	Always have resource monitor start with the FECRU server to log initial data	Unassigned	Daniel Rohan [Atlassian]		 Closed	Fixed	Aug 07, 2012	Aug 10, 2012
	FE-4087	IllegalArgumentEx ception: No local entity with key 'REPOSITORYEntity TypeImpl' exists	Geoff Crain [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	May 07, 2012	Jul 09, 2012
	FE-4017	Add authenticated user in http response headers for logging ability in reverse proxy	Brendan Humphreys [Atlassian]	Issa Gorissen		 Closed	Fixed	Mar 26, 2012	Apr 05, 2012
	FE-4128	Remove deprecated TrustedAppsService	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Jun 05, 2012	Jun 14, 2012
	FE-4095	Upgrade crowd client to 2.3.x. Currently fisheye ships with 2.2.5.	Richard Stephens [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	May 14, 2012	Jul 19, 2012

	FE-3819	Subversion 1.7 Support	Conor MacNeill [Atlassian]	Ajay Sridhar [Atlassian]		 Closed	Fixed	Nov 07, 2011	Nov 21, 2012
	FE-4090	Poor Dashboard page performance for logged in users	Unassigned	Nick Pellow [Atlassian]		 Closed	Fixed	May 08, 2012	Jun 12, 2012
	FE-4224	Subversion Non-Versioned Properties re-scan not working	Unassigned	Kevin Wood		 Closed	Fixed	Aug 09, 2012	Sep 19, 2012
	FE-4222	FishEye privilege escalation vulnerability	Vitaly Osipov [Atlassian]	Paul Watson [Atlassian]		 Closed	Fixed	Aug 09, 2012	Aug 21, 2012
	FE-4198	if non-standard crowd properties are set in config.xml, don't overwrite them if config is changed through UI	Richard Stephens [Atlassian]	Richard Stephens [Atlassian]		 Closed	Fixed	Jul 17, 2012	Jul 19, 2012
	FE-4183	Fix Activity Stream paging (possibly with Ajaxified, infinite scroll or similar)	Brendan Humphreys [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	Jul 09, 2012	Aug 13, 2012
	FE-4182	Hg scanning chokes on invalid	Brendan Humphreys	Brendan Humphreys		 Closed	Fixed	Jul 08, 2012	Jul 10, 2012

		branch name	ys [Atlassian]	ys [Atlassian]					
	FE-4176	Encoding of asian characters + accented characters in email subject doesn't seem to work properly	Unassigned	Richard Stephens [Atlassian]	↓	 Close	Fixed	Jul 05, 2012	Jul 12, 2012
	FE-4173	Polling Interval help text on screen needs to be updated	Roman Tekhov [Atlassian]	Daniel Rohan [Atlassian]	↓	 Close	Fixed	Jul 03, 2012	Jul 16, 2012
	FE-4156	Watch Emails throwing exception in certain circumstances	Unassigned	Nick Pellow [Atlassian]	↑	 Close	Fixed	Jun 24, 2012	Jul 10, 2012
	FE-4139	It takes 5 minutes to index a new branch in git	Conor MacNeill [Atlassian]	Adam Ahmed [Atlassian]	↑	 Close	Fixed	Jun 14, 2012	Jun 26, 2012
	FE-4129	Fix the Subtask integration for Crucible	Unassigned	Douglas Fabretti [Atlassian]	↓	 Close	Fixed	Jun 06, 2012	Nov 20, 2012
	FE-4125	Implicit commit mapping are potentially expensively computed multiple	Brendan Humphreys [Atlassian]	Brendan Humphreys [Atlassian]	↓	 Close	Fixed	Jun 03, 2012	Jun 11, 2012

		times on every request							
	FE-4113	Trying to browse a file name containing a + generates an error	Unassigned	Jean-Yves Avenard		 Closed	Fixed	May 25, 2012	Jul 12, 2012
	FE-4110	Overlapping Label Text in FishEye UI Using Internet Explorer (IE9)	Richard Stephens [Atlassian]	Daniel Rohan [Atlassian]		 Closed	Fixed	May 24, 2012	Jul 19, 2012
	FE-4086	hg indexing falls over on commits with very large commit messages (500k lines)	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	May 03, 2012	May 27, 2012
	FE-4049	Allow specification of which mime-types are to be treated as text	Unassigned	Conor MacNeill [Atlassian]		 Closed	Fixed	Apr 11, 2012	May 14, 2012
	FE-4026	Project Page Performance with many associated repositories	Brendan Humphreys [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Apr 02, 2012	Jul 23, 2012
	FE-4006	Consider removing Crowd Auth and LDAP Auth from	Geoff Crain [Atlassian]	Daniel Rohan [Atlassian]		 Closed	Fixed	Mar 16, 2012	May 03, 2012

		dropdown when creating users in FishEye (was: Authentication Sync (or Re-Sync) Deletes Users Created Only in FE)							
	FE-3930	Exception when upgrading to UPM 2.0 in Fisheye	Nick Pellow [Atlassian]	Douglas Fabretti [Atlassian]		 Close	Fixed	Feb 03, 2012	Jun 26, 2012
	FE-3877	SASL support with bundled svnkit in 2.7.x	Conor MacNeill [Atlassian]	Gurleen Anand [Atlassian]		 Close	Fixed	Jan 05, 2012	Jun 18, 2012

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](#)

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.

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

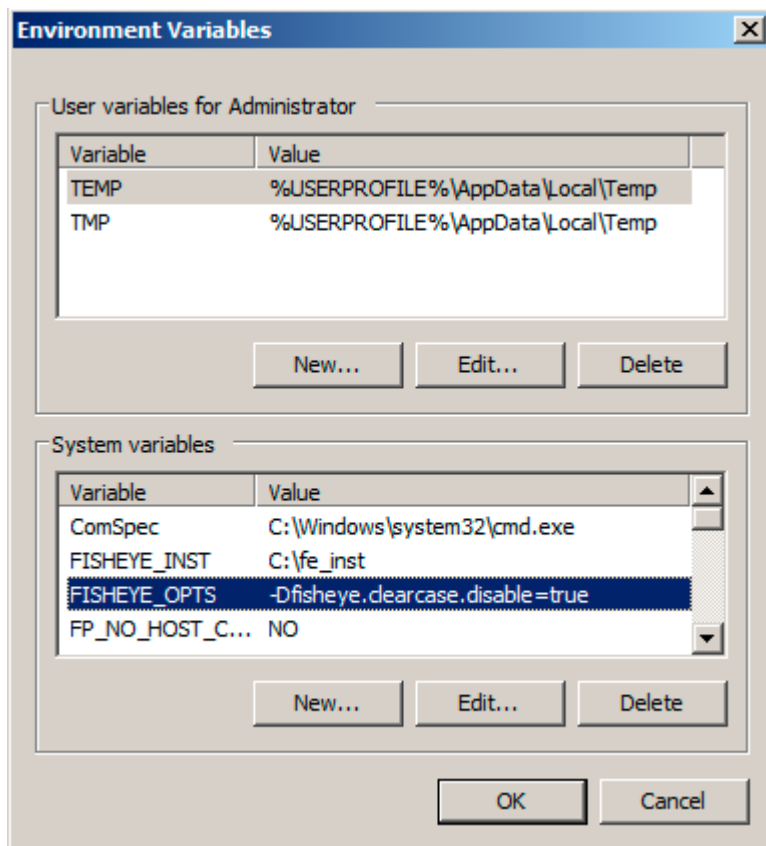
-Dfisheye.clearcase.disable=true

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 2.7 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](#)



- 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.

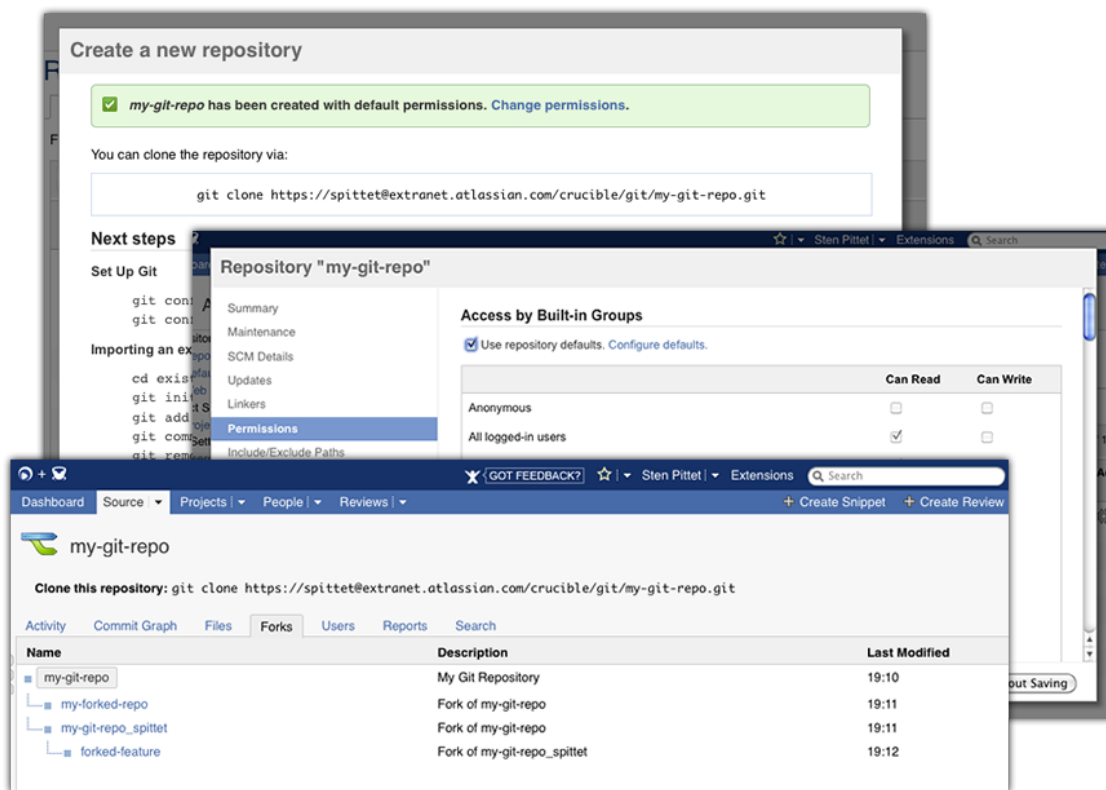
Highlights of FishEye 2.7

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...](#)

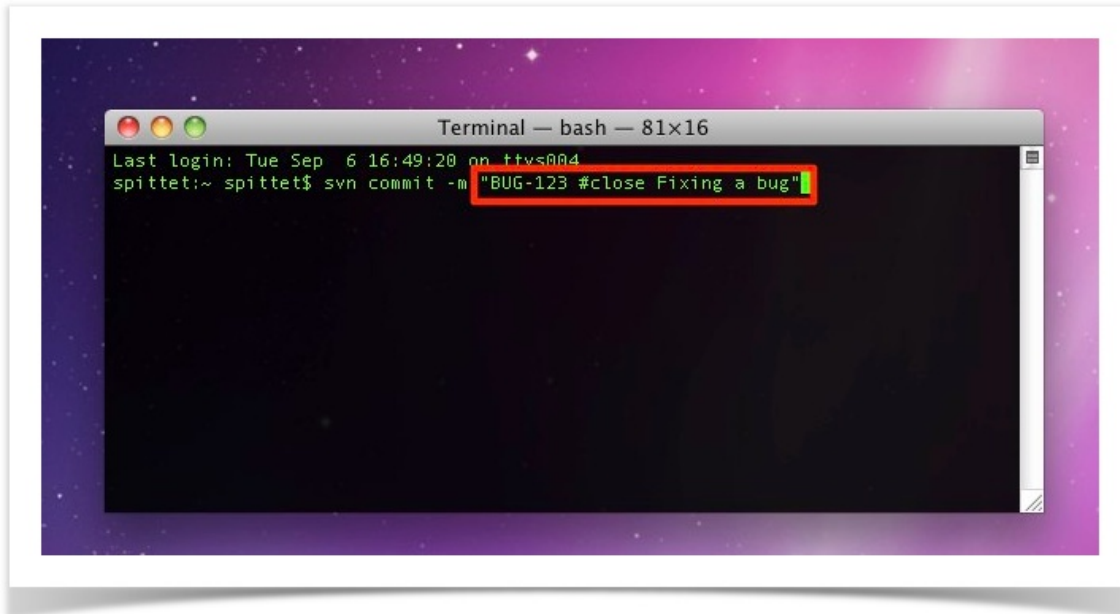
2

Smart Commits

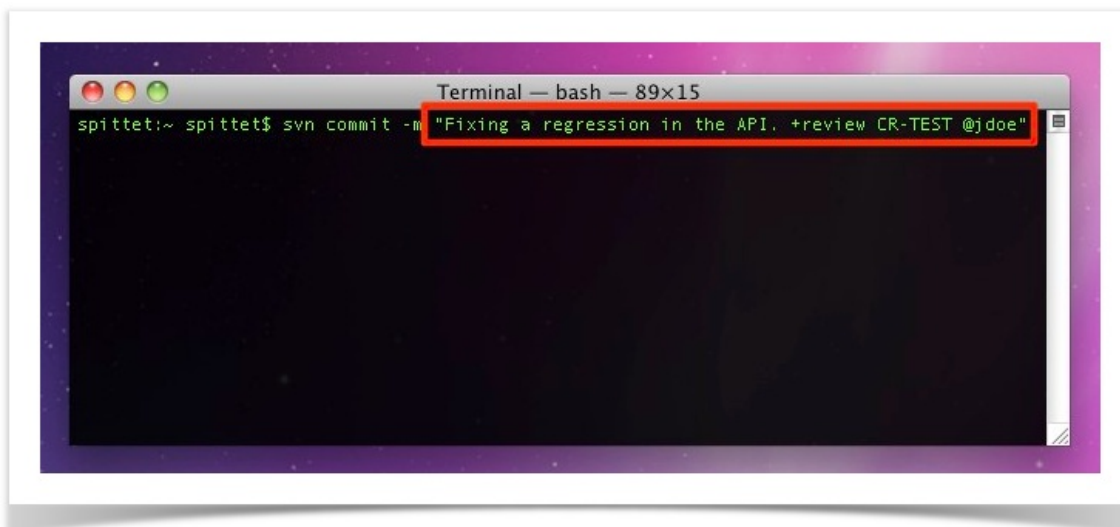
Smart Commits allow repository committers to perform actions like transitioning JIRA issues or creating Crucible code reviews by embedding specific commands into their commit messages.

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.



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:



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.

i 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.

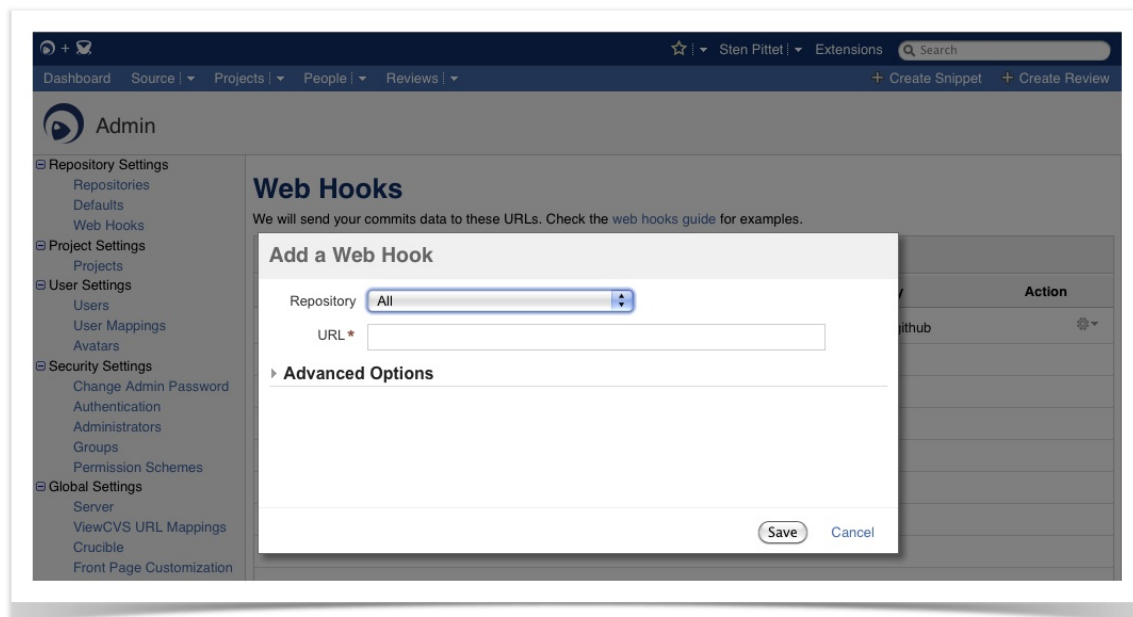
[More...](#)

3

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...](#)

4

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...](#)

5

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 straight forward. *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](#)
- [From 2.7.12 to 2.7.13](#)
- [From 2.7.11 to 2.7.12](#)
- [From 2.7.10 to 2.7.11](#)
- [From 2.7.9 to 2.7.10](#)
- [From 2.7.8 to 2.7.9](#)
- [From 2.7.7 to 2.7.8](#)
- [From 2.7.6 to 2.7.7](#)
- [From 2.7.5 to 2.7.6](#)
- [From 2.7.4 to 2.7.5](#)
- [From 2.7.3 to 2.7.4](#)
- [From 2.7.2 to 2.7.3](#)
- [From 2.7.1 to 2.7.2](#)
- [From 2.7.0 to 2.7.1](#)

From 2.7.13 to 2.7.14

12 June 2012

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

JIRA Issues (5 issues)									
Type	Key	Summar	Assigne	Reporter	Priority	Status	Resoluti	Created	Updated

		y	e			on			
	FE-4119	Missing changesets cause Hg indexing to fail	Unassigned	Tom Davies [Atlassian]	↓	 Closed	Fixed	May 30, 2012	May 30, 2012
	FE-4111	If Anon Access is OFF, smart commits that interact with JIRA will not work	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]	↓	 Closed	Fixed	May 24, 2012	Aug 01, 2012
	FE-4106	Fisheye 2.7.13 startup error when FISHEYE_INST have spaces	Nick Pellow [Atlassian]	Leo Leung	↑	 Closed	Fixed	May 24, 2012	Jun 08, 2012
	FE-4105	100+ requests of type "updateJiraTimeAjax!default" timing out when modifying Time Spent	Geoff Crain [Atlassian]	Daniel Rohan [Atlassian]	↑	 Closed	Fixed	May 23, 2012	Jun 08, 2012
	FE-4103	NPE when indexing git repo with missing ancestor	Tom Davies [Atlassian]	Tom Davies [Atlassian]	↓	 Closed	Fixed	May 22, 2012	Jun 08, 2012

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated

	FE-3961	Emulate pre-2.7 behaviour: when an user is created by the user synchronisation, restore any deleted user with the same username	Brendan Humphreys [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Closed	Fixed	Feb 16, 2012	Aug 20, 2012
	FE-4019	web-server/max-thread reads increases from 20 to 150	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	Mar 13, 2012	Nov 18, 2012
	FE-4018	EyeQL Search Results not displaying author fields	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Mar 27, 2012	Apr 09, 2012
	FE-4017	Add authenticated user in http response headers for logging ability in reverse proxy	Brendan Humphreys [Atlassian]	Issa Gorissen		 Closed	Fixed	Mar 26, 2012	Apr 05, 2012
	FE-3989	Allow disabling of precise contentHash to csid mapping	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Mar 08, 2012	May 01, 2012
	FE-3922	Add custom syntax highlighting for	Brendan Humphreys [Atlassian]	edward webb		 Closed	Fixed	Jan 27, 2012	Nov 07, 2012

		shell scripts							
	FE-3495	com.atlassian.crowd.exception.OperationFailedException if SSO is enabled with JIRA user management	Brendan Humphreys [Atlassian]	Gurleen Anand [Atlassian]		 Closed	Fixed	Jun 13, 2011	Jul 31, 2012
	FE-4084	hidden text behind the left navigation panel when using Firefox 12	Unassigned	Leonardo De Macedo [Atlassian]		 Closed	Fixed	May 01, 2012	Jun 27, 2012
	FE-4070	Fisheye does not allow username returned by AuthToken to be different from username entered by user	Richard Stephens [Atlassian]	Richard Stephens [Atlassian]		 Closed	Fixed	Apr 19, 2012	Apr 30, 2012
	FE-4060	CC Indexing can fetch version 0 and index it even if before the start date.	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Apr 17, 2012	Apr 18, 2012
	FE-4044	Lack of usernames element in security	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Apr 11, 2012	Apr 15, 2012

		config can stop AJPAuth working							
	FE-4043	Handle NPE caused by changes et without a position value in Commit Graph	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Apr 10, 2012	Apr 15, 2012
	FE-4036	P4 GetLates tRevision will fail for paths with no revisions - leads to excessive logging.	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Apr 05, 2012	Apr 10, 2012
	FE-4034	Copy from a file which existed before the start-revision is ignored.	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↑	 Closed	Fixed	Apr 03, 2012	Apr 10, 2012
	FE-4029	Store Perforce Branch Details	Tom Davies [Atlassian]	Tom Davies [Atlassian]	↓	 Closed	Fixed	Apr 02, 2012	Apr 25, 2012
	FE-4025	Modified port not reflected in startup message of uninitialized FishEye	Brendan Humphreys [Atlassian]	Rene Verschuur [Atlassian]	↓	 Closed	Fixed	Mar 31, 2012	Apr 04, 2012
	FE-4024	Clearcase FileHistory can throw an exception if no	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Mar 29, 2012	Apr 10, 2012

		HEAD revisions							
	FE-4020	Deleted git branches show in branch selector	Conor MacNeill [Atlassian]	Matthew Watson		 Close	Fixed	Mar 27, 2012	Apr 18, 2012
	FE-4012	Avatars in user dropdown are too big	Richard Stephens [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Mar 22, 2012	Apr 05, 2012
	FE-4003	JIRA Application Password Field Not Refreshing with Apply Button	Brendan Humphreys [Atlassian]	Daniel Rohan [Atlassian]		 Close	Fixed	Mar 15, 2012	Apr 04, 2012
	FE-3986	An svn copy from an unindexed part of a repo to an indexed part causes an error: com.cenqua.fisheye.rep.DbException: Attempt to load isDeleted from non existent revid = -1	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Close	Fixed	Mar 06, 2012	Jul 13, 2012
	FE-3985	Display the SSID above the license textarea	Richard Stephens [Atlassian]	Stefan Sorin Nicolin		 Close	Fixed	Mar 06, 2012	Mar 27, 2012
	FE-3984	JIRA user manage	Brendan Humphreys	Ajay Sridhar [Atlassian]		 Close	Fixed	Mar 06, 2012	Apr 04, 2012

		ment fails if there are no groups selected to sync	[Atlassian]	n]					
	FE-3979	Missing label for "Allow remote API calls" option in FishEye UI	Richard Stephens [Atlassian]	Sarah Maddox [Atlassian]		 Closed	Fixed	Feb 27, 2012	Apr 23, 2012
	FE-3899	fisheye doesnt handle p4 branch specs in ways that p4 supports	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	Jan 18, 2012	May 01, 2012
	FE-3870	Removed license gets replaced by "Disabled" preventing Update	Conor MacNeill [Atlassian]	Ajay Sridhar [Atlassian]		 Closed	Fixed	Dec 29, 2011	Sep 07, 2012
	FE-3732	Switching from tag to trunk results in a 500 Server Error	Conor MacNeill [Atlassian]	Rene Verschuur [Atlassian]		 Closed	Fixed	Sep 21, 2011	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 (9 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3974	Add the new system property	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Closed	Fixed	Feb 23, 2012	Apr 16, 2012

		to the 2.7 documentation							
	FE-3973	Allow to configure the maximum number of repositories in the Search	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Closed	Fixed	Feb 23, 2012	Aug 01, 2012
	FE-4016	FishEye XML Vulnerability	Vitaly Osipov [Atlassian]	Paul Watson [Atlassian]		 Closed	Fixed	Mar 26, 2012	May 17, 2012
	FE-4001	Perforce branches are unnecessarily repeatedly scanned	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Mar 15, 2012	Apr 02, 2012
	FE-4000	Corruption of CrossRepository Index Properties	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Mar 14, 2012	Mar 21, 2012
	FE-3996	Cleartool since date format need to be English	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Mar 12, 2012	Mar 21, 2012
	FE-3976	ClassCastException on startup on windows	Richard Stephens [Atlassian]	Wannes Geysen		 Closed	Fixed	Feb 27, 2012	Apr 11, 2012
	FE-3970	Commit Calendar incorrect for projects including several paths in the same	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Feb 22, 2012	Feb 28, 2012

		repository							
	FE-3969	The 'Parameters' on the 'Database Configuration' page are ignored.	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Feb 22, 2012	Oct 03, 2012

From 2.7.10 to 2.7.11

27 February 2012

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

JIRA Issues (20 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3941	Add a limit for the number of paths to index to a changeset	Matthew Watson	Matthew Watson		 Closed	Fixed	Feb 09, 2012	Feb 26, 2012
	FE-3937	Add properties to stop branch and tag point ancestry detection for SVN	Matthew Watson	Matthew Watson		 Closed	Fixed	Feb 07, 2012	Feb 26, 2012
	FE-3931	Null Pointer Exception com.cenqua.fisheye.rep.impl.CommonFileRevision.setAuthor(CommonFileRevision.java:149)	Brendan Humphreys [Atlassian]	Cardinal Armand du Piessis de Richelieu, First Minister of Louis XIII		 Closed	Fixed	Feb 03, 2012	Feb 23, 2012

	FE-3958	Content Indexing: Filter the paths to index before processing the list twice.	Matthew Watson	Matthew Watson		 Closed	Fixed	Feb 15, 2012	Feb 21, 2012
	FE-3957	Lucene re-index is only triggered for cross-repo until after first commit	Matthew Watson	Matthew Watson		 Closed	Fixed	Feb 15, 2012	Feb 16, 2012
	FE-3956	Content Indexing: Add an option to only index trunk, not root and trunk	Matthew Watson	Matthew Watson		 Closed	Fixed	Feb 15, 2012	Feb 26, 2012
	FE-3955	ReIndex: Only delete all the content docs if there are >0 docs in the index	Matthew Watson	Matthew Watson		 Closed	Fixed	Feb 15, 2012	Feb 19, 2012
	FE-3953	Per user LoC figures are meaningless when the 'All' button is selected	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Feb 14, 2012	Feb 15, 2012
	FE-3952	Breadcrumb links on the Users tab are incorrect when a branch is selected	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Feb 14, 2012	Feb 14, 2012

	FE-3939	PHP syntax highlighting incorrectly matches keywords at the end of function names	Brendan Humphreys [Atlassian]	Chris Butler		 Close	Fixed	Feb 08, 2012	Feb 08, 2012
	FE-3934	Changesets with large number of revisions can exhaust heap	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Close	Fixed	Feb 06, 2012	Feb 26, 2012
	FE-3932	FishEye appears hung on upgrade. till upgrade finishes.	Conor MacNeill [Atlassian]	Matthew Watson		 Close	Fixed	Feb 06, 2012	Feb 23, 2012
	FE-3921	Fisheye treats files as binary if they used to have a svn:mime-type property. even after the property has been removed	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Close	Fixed	Jan 26, 2012	Feb 23, 2012
	FE-3908	Investigate lib dir containing random JS files and directories. and scmutils.jar is missing.	Unassigned	Nick Pellow [Atlassian]		 Close	Fixed	Jan 22, 2012	Feb 07, 2012
	FE-3905	Upgrade	Matthew	Matthew		 Close	Fixed	Jan 20, 2012	Jan 23, 2012

		from 2.5->2.7 fires events for all commits and replays all smart commits	Watson	Watson		d		2012	2012
	FE-3885	Smart Commits: Using the #time command with a complex time format (i.e. 2d 4h) only saves the 2d part	Nick Pellow [Atlassian]	Felipe Cuozzo [Atlassian]		 Close d	Fixed	Jan 12, 2012	Mar 12, 2012
	FE-3859	Unable to add users when using IE8	Seb Ruiz [Atlassian]	Felipe Kraemer [Atlassian]		 Close d	Fixed	Dec 12, 2011	Nov 07, 2012
	FE-3855	Author column is misaligned in Annotation View	Tom Davies [Atlassian]	Timothy Lusk		 Close d	Fixed	Dec 09, 2011	Feb 19, 2012
	FE-3851	Old broken Support Zip link still present	Brendan Humphreys [Atlassian]	Rene Verschoor [Atlassian]		 Close d	Fixed	Dec 02, 2011	Feb 15, 2012
	FE-3608	Issue keys. wrapped in square brackets [JRA-9] are not linked correctly in quick search results	Brendan Humphreys [Atlassian]	Nick Pellow [Atlassian]		 Close d	Fixed	Jul 27, 2011	Feb 22, 2012

From 2.7.9 to 2.7.10

20 January 2012

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

JIRA Issues (3 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3903	Unable to see FishEye activity stream in JIRA Activity Stream gadget on dashboard when have Repository with no description	Unassigned	Matthew Watson		 Closed	Fixed	Jan 19, 2012	Jan 19, 2012
	FE-3902	Issue Transition controls don't appear when closing a review	Unassigned	None		 Closed	Fixed	Jan 19, 2012	Jan 19, 2012
	FE-3900	fecru-review-issue-transition-plugin fails under fisheye 2.7.9 due to incompatible class change error	Unassigned	Nick Pellow [Atlassian]		 Closed	Fixed	Jan 18, 2012	Jan 19, 2012

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated

		y	e			on			
	FE-3896	FishEye wrongly removes admin privileges for a Crowd group when Crowd is not up when FishEye is starting up	Nick Pellow [Atlassian]	Alex Wei [Atlassian]		 Close	Fixed	Jan 05, 2012	Jan 18, 2012
	FE-3891	Webwork 2 vulnerability	Vitaly Osipov [Atlassian]	Paul Watson [Atlassian]		 Close	Fixed	Jan 16, 2012	Jan 31, 2012
	FE-3888	FishEye does not correctly handle moved tags	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Close	Fixed	Jan 15, 2012	Jan 17, 2012
	FE-3884	DefaultUserManager.getUsersInGroup incorrectly acquires a write lock instead of a read lock	Brendan Humphreys [Atlassian]	Brendan Humphreys [Atlassian]		 Close	Fixed	Jan 12, 2012	Jan 12, 2012
	FE-3880	P4 client doesn't handle job names with a leading -	Brendan Humphreys [Atlassian]	Brendan Humphreys [Atlassian]		 Close	Fixed	Jan 08, 2012	Jan 09, 2012
	FE-3878	When processing perforce file revisions, only branch	Anna Buttfield [Atlassian]	Anna Buttfield [Atlassian]		 Close	Fixed	Jan 08, 2012	Jan 09, 2012

		specs that have been added or modified since the last restart are considered							
	FE-3875	SVN property changes with content lines containing certain strings break diff parsing	Anna Buttfield [Atlassian]	Anna Buttfield [Atlassian]		 Closed	Fixed	Jan 03, 2012	Jan 17, 2012
	FE-3862	JS errors in ie8	Anna Buttfield [Atlassian]	Anna Buttfield [Atlassian]		 Closed	Fixed	Dec 13, 2011	Jan 17, 2012
	FE-3861	lock mismanagement in LuceneConnection	Brendan Humphreys [Atlassian]	Brendan Humphreys [Atlassian]		 Closed	Fixed	Dec 12, 2011	Dec 12, 2011
	FE-3817	Clicking on repository in Fisheye Administration generates error dialog in IE8	Anna Buttfield [Atlassian]	Felipe Kraemer [Atlassian]		 Closed	Fixed	Nov 04, 2011	Jan 17, 2012
	FE-3790	Trying to look at the source of an empty file results in java.lang.ArrayIndexOutOfBoundsException	Brendan Humphreys [Atlassian]	Rene Verschoor [Atlassian]		 Closed	Fixed	Oct 20, 2011	Dec 22, 2011

		exception							
		--1							

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3846	crowd admin groups get removed on startup	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]	↓	 Closed	Fixed	Nov 27, 2011	Nov 29, 2011
	FE-3842	"Internal Server Error" on jira issue hover	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]	↓	 Closed	Fixed	Nov 23, 2011	Nov 23, 2011
	FE-3839	autocomplete dropdowns give inconsistent and sometimes wrong results (results missing)	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]	↓	 Closed	Fixed	Nov 20, 2011	Nov 21, 2011
	FE-3775	Allowed Review Participants field doesn't work when some users are present	Pierre-Etienne Poirot [Atlassian]	Adam Ahmed [Atlassian]	↓	 Closed	Fixed	Oct 11, 2011	Nov 29, 2011
	FE-3745	Changing Block Size triggers a reindex + restart. while only a restart is needed	Pierre-Etienne Poirot [Atlassian]	Rene Verschoor [Atlassian]	↑	 Closed	Fixed	Sep 26, 2011	Nov 24, 2011

	FE-3040	FishEye Access Logging is not writing the username	Pierre-Etienne Poirot [Atlassian]	Felipe Kraemer [Atlassian]	↓	 Closed	Fixed	Apr 27, 2011	Nov 29, 2011
	FE-2473	Cannot find link for downloading latest and greatest RAW revision of a file	Pierre-Etienne Poirot [Atlassian]	David Hergert	↓	 Closed	Fixed	Feb 18, 2010	Nov 29, 2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3836	In SVN, if a tag is created and the identified parent changes et id is not indexed by FishEye the tag won't show up in the branches and tags dropdown or on the commit graph	Anna Buttfield [Atlassian]	Anna Buttfield [Atlassian]	↓	 Closed	Fixed	Nov 16, 2011	Nov 17, 2011
	FE-3833	Perforce jobs are not indexed	Tom Davies [Atlassian]	Tom Davies [Atlassian]	↑	 Closed	Fixed	Nov 15, 2011	Nov 16, 2011

	FE-3832	Links in RSS feed do not use absolute URLs	Seb Ruiz [Atlassian]	Oleg Semyonov		 Closed	Fixed	Nov 15, 2011	Nov 16, 2011
	FE-3826	Jetty log messages are not reported anywhere	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Nov 10, 2011	Nov 16, 2011
	FE-3822	mime type mapping for file names rather than suffixes removes first character of candidate file name before matching with key filename	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Nov 08, 2011	Nov 17, 2011
	FE-3174	When navigating forward in time in activity stream, items are shown earliest to latest down the page	Michael Studman [Atlassian]	None		 Closed	Fixed	Sep 22, 2010	Jan 31, 2012
	FE-3155	Browse/diff menu inactives look active and vice-versa	Jonathan Poh [Atlassian]	None		 Closed	Fixed	Dec 12, 2010	Nov 16, 2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3816	Shutdown the thread pool once the background upgrade tasks have been completed	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]	↓	 Closed	Fixed	Nov 03, 2011	Nov 03, 2011
	FE-3791	Change authentication providers to return CommunicationException when appropriate	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]	↑	 Closed	Fixed	Oct 20, 2011	Nov 01, 2011
	FE-3772	Add option to enable client authentication SSL	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]	↓	 Closed	Fixed	Oct 09, 2011	Oct 17, 2012
	FE-3611	Improve the way avatars are loaded to avoid unnecessary calls	Anna Buttfield [Atlassian]	Renan Battaglin [Atlassian]	↓	 Closed	Fixed	Jul 28, 2011	Nov 03, 2011
	FE-3823	If the total length of the branch names selected for the commit graph is	Unassigned	Tom Davies [Atlassian]	↓	 Closed	Fixed	Nov 09, 2011	Nov 10, 2011

		too long. the spinner never disappears							
	FE-3812	When synchronizing users with LDAP, do not deactivate users if the error is a communication problem	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Closed	Fixed	Nov 01, 2011	Nov 01, 2011
	FE-3809	Upgrade of instances using Oracle fail when upgrade_75.sql is run	Anna Buttfield [Atlassian]	Renan Battaglin [Atlassian]		 Closed	Fixed	Oct 31, 2011	Nov 02, 2011
	FE-3794	Sourcing a Tag from a Tag Can cause FishEye not to display tag contents	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Oct 24, 2011	Nov 03, 2011
	FE-3787	When synchronizing users with Crowd, do not deactivate users if the error is a communication problem	Pierre-Etienne Poirot [Atlassian]	Adam Ahmed [Atlassian]		 Closed	Fixed	Oct 19, 2011	Aug 20, 2012
	FE-3784	Modified tags fail	Adam Ahmed	Adam Ahmed		 Closed	Fixed	Oct 18, 2011	Nov 02, 2011

		in the branch selector	[Atlassian]	[Atlassian]					
	FE-3783	Activity Stream UI sometimes slow in git - it makes calls to git remote show.	Adam Ahmed [Atlassian]	Adam Ahmed [Atlassian]		 Close	Fixed	Oct 17, 2011	Nov 04, 2011
	FE-3767	WARN message 'Repository index does not match the repository configuration' incorrectly logged after upgrading to 2.7	Conor MacNeill [Atlassian]	Michael Heemskeerk [Atlassian]		 Close	Fixed	Oct 05, 2011	Nov 02, 2011
	FE-3645	Old non-http's SimpleLinker configuration passes upgrade but results in error when trying to open repo	Pierre-Etienne Poirot [Atlassian]	Rene Verschuur [Atlassian]		 Close	Fixed	Aug 15, 2011	Nov 04, 2011
	FE-3620	Displayname not being picked up correctly in some cases with JIRA	Pierre-Etienne Poirot [Atlassian]	Gurleen Anand [Atlassian]		 Close	Fixed	Aug 03, 2011	Jan 16, 2012

		user management							

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3799	p4 support: Cleanup and Merge the p4 ancestry suppression flag to 2.6/7/default	Conor MacNeill [Atlassian]	Matthew Watson	↓	 Closed	Fixed	Sep 27, 2011	Jan 16, 2012
	FE-3768	FishEye logs lots of ApplicationPermissionException when synchronizing Crowd users that have no permission on FishEye / Crucible	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]	↓	 Closed	Fixed	Oct 06, 2011	Oct 11, 2011
	FE-3788	Git does not report an error when creating a managed repository over an existing one	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]	↓	 Closed	Fixed	Oct 19, 2011	Oct 19, 2011
	FE-3777	Fix document	Pierre-Etienne	Pierre-Etienne	↓	 Closed	Fixed	Oct 11, 2011	Oct 12, 2011

		tation of the managed repositories' REST API	Poirot [Atlassian]	Poirot [Atlassian]					
	FE-3774	Search results page fails to render for EyeQL queries > 4000 characters on Oracle.	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Oct 10, 2011	Oct 11, 2011
	FE-3755	SVN: Paths under tag directories are displayed as deleted / empty directories (greyed out)	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Sep 30, 2011	Oct 11, 2011
	FE-3752	Repository path. includes. excludes and hidden directories should be copied to newly created forks	Tom Davies [Atlassian]	Tim Pettersen [Atlassian]		 Closed	Fixed	Sep 29, 2011	Oct 04, 2011
	FE-3751	'Store diffs' setting is not respected for initial imports of SVN repositories	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Sep 28, 2011	Oct 09, 2011
	FE-3737	Cannot customiz	Tom Davies	Gurleen Anand		 Closed	Fixed	Sep 22, 2011	Oct 06, 2011

		e front page with global anonymous access turned off	[Atlassian]	[Atlassian]					
	FE-3727	Review coverage report screen messed up when using Firefox 6.0.2 and Chrome 12.0.7	Michael Studman [Atlassian]	Camila Ayres [Atlassian]		 Closed	Fixed	Sep 16, 2011	Oct 26, 2011
	FE-3721	Error when changing content of a review	Tom Davies [Atlassian]	Federico Silva Armas [Atlassian]		 Closed	Fixed	Sep 15, 2011	Jan 16, 2012
	FE-3698	Web Hooks payload received with "application/xml" as the content type	Anna Buttfield [Atlassian]	Sten Pittet [Atlassian]		 Closed	Fixed	Sep 06, 2011	Oct 24, 2011
	FE-3692	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]		 Closed	Fixed	Sep 06, 2011	Oct 09, 2011
	FE-3678	Provide a more friendly message to the users	Anna Buttfield [Atlassian]	Zed Yap [Atlassian]		 Closed	Fixed	Sep 05, 2011	Oct 12, 2011
	FE-2625	code	Pierre-Eti	Tracey		 Closed	Fixed	Jun 03,	Oct 10,

		metrics error for branches directories	enne Poirot [Atlassian]	Lum [Atlassian]	d		2010	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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3748	NullPointerException while loading user preferences from database, causing login failures	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Sep 27, 2011	Sep 27, 2011
	FE-3747	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.	Tim Pettersen [Atlassian]	Tim Pettersen [Atlassian]		 Closed	Fixed	Sep 26, 2011	Sep 27, 2011
	FE-3742	After a user has been deleted and then added again, subsequent	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Sep 26, 2011	Sep 27, 2011

		ent deletions fail							
	FE-3741	Manual Auth Re-Sync button doesn't do anything	Tom Davies [Atlassian]	Phil Davies		 Close	Fixed	Sep 23, 2011	Sep 27, 2011
	FE-3738	Log files not included in support zip	Matthew Watson	Matthew Watson		 Close	Fixed	Sep 23, 2011	Jan 16, 2012
	FE-3735	Commit events fired on Git repos when branches are added/removed	Seb Ruiz [Atlassian]	Matt Ryall [Atlassian]		 Close	Fixed	Sep 22, 2011	Sep 27, 2011
	FE-3734	Error upon exceeding license limit with Crowd user management	Tom Davies [Atlassian]	Gurleen Anand [Atlassian]		 Close	Fixed	Sep 21, 2011	Sep 27, 2011
	FE-3716	Hq authentication fails with special character in password	Michael Studman [Atlassian]	Gurleen Anand [Atlassian]		 Close	Fixed	Sep 13, 2011	Jan 16, 2012
	FE-3168	Restricting Fisheye content in Projects Has No Effect	Michael Studman [Atlassian]	None		 Close	Fixed	Dec 06, 2010	Dec 07, 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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3717	Augment the git-http-backend from 1min to 1hour	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Closed	Fixed	Sep 14, 2011	Sep 19, 2011
	FE-3635	Backup fails if write.lock file still exists in lucene	Michael Studman [Atlassian]	Renan Battaglin [Atlassian]		 Closed	Fixed	Aug 09, 2011	Jan 16, 2012

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3616	Change com.cenqua.fisheye.logging.DailyRollingFileAppender to keep the extension .log	Conor MacNeill [Atlassian]	Felipe Cuozzo [Atlassian]		 Closed	Fixed	Aug 02, 2011	Jan 16, 2012
	FE-3718	Eclipse Git integration (egit) fails to clone from managed Git repository	Unassigned	Tim Pettersen [Atlassian]		 Closed	Fixed	Sep 14, 2011	Sep 19, 2011
	FE-3711	Some Directory links in Quick	Jason Hinch [Atlassian]	Jason Hinch [Atlassian]		 Closed	Fixed	Sep 12, 2011	Jan 16, 2012

		Search have double encoded URLs							
	FE-3696	Web Hooks should be deleted when deleting a repository	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]		 Closed	Fixed	Sep 06, 2011	Sep 13, 2011
	FE-3693	Pagination numbers at bottom of Repositories screen overlap with system info footer	Jonathan Poh [Atlassian]	Rene Verschoor [Atlassian]		 Closed	Fixed	Sep 06, 2011	Sep 14, 2011
	FE-3647	Google collections and Guava both bundled. package exported without version	Conor MacNeill [Atlassian]	Richard Wallace [Atlassian]		 Closed	Fixed	Aug 15, 2011	Sep 14, 2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3701	CommitHookModuleDescriptor#enable can fail if ActiveObj	Jason Hinch [Atlassian]	Jason Hinch [Atlassian]		 Closed	Fixed	Sep 07, 2011	Sep 11, 2011

		ectsConfi guration for Commit Hooks Plugin is not available yet							
	FE-3700	Can't reset user password	Pierre-Eti enne Poirot [Atlassia n]	Conor MacNeill [Atlassia n]		 Close d	Fixed	Sep 07, 2011	Sep 08, 2011
	FE-3695	Deleting a web hook dialog has a broken layout	Jason Hinch [Atlassia n]	Seb Ruiz [Atlassia n]		 Close d	Fixed	Sep 06, 2011	Sep 08, 2011
	FE-3694	Testing a Web Hook doesn't send mock changes et displayId	Seb Ruiz [Atlassia n]	Seb Ruiz [Atlassia n]		 Close d	Fixed	Sep 06, 2011	Sep 06, 2011
	FE-3018	Unneces sary implicit antglobbi ng in Query3	Brendan Humphre ys [Atlassia n]	Brendan Humphre ys [Atlassia n]		 Close d	Fixed	Apr 11, 2011	Oct 05, 2012

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



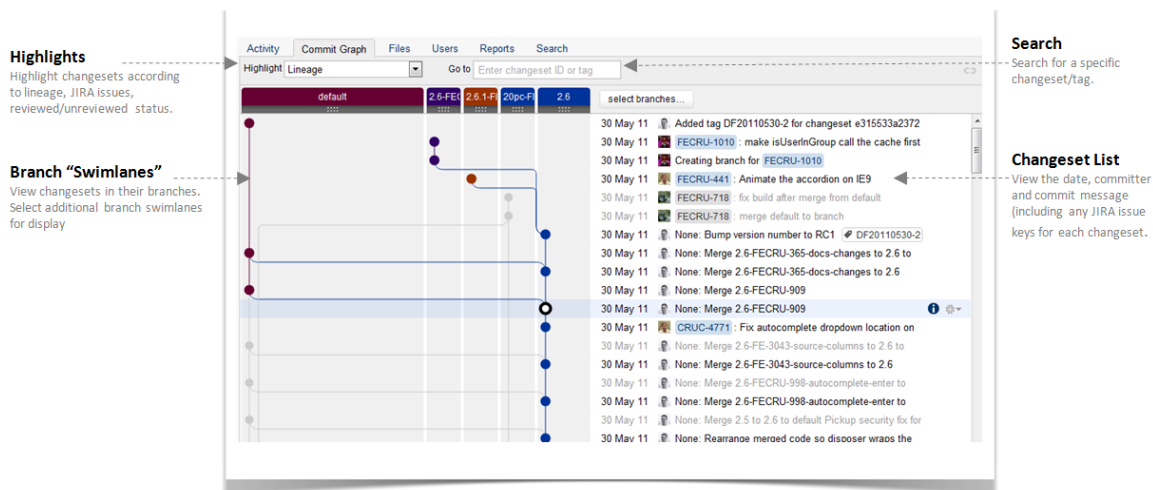
- Thank you for all your issues and votes. [Keep logging issues](#) to help us keep improving!
- Read the [release notices](#) for important information about this release.

Highlights of FishEye 2.6

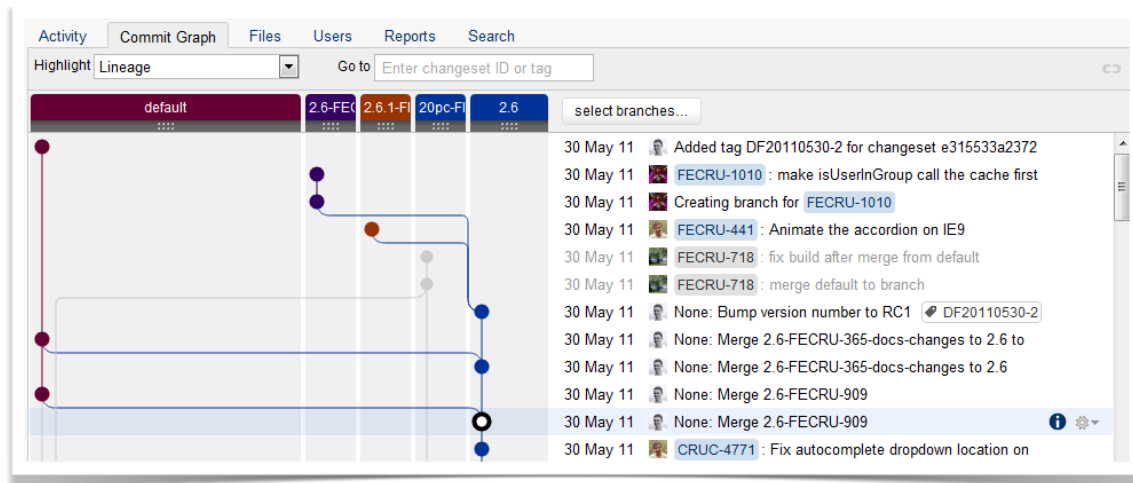


Repository Commit Graph

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.



[More...](#)

2

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.

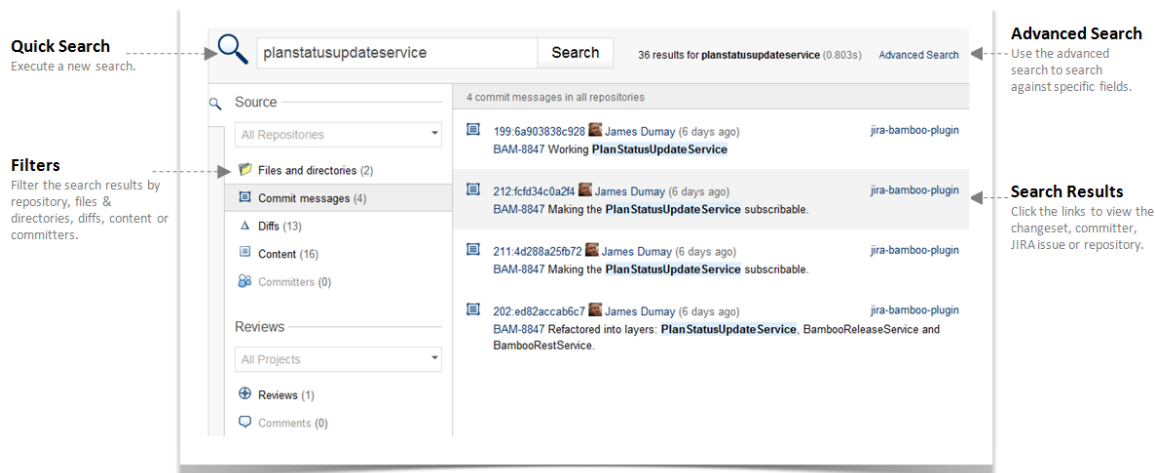
The screenshot shows the 'Connect to JIRA' configuration screen in the FishEye setup wizard. On the left is a sidebar with steps: 1. License, 2. Inclusions, 3. Connect to JIRA (selected), 4. Admin Password, and 5. Done. The main content area has the title 'Connect to JIRA' and a note: 'If you have JIRA 4.3 or later, Crucible can use JIRA for user management. This is not recommended for more than 500 users. Learn more [here](#).' Below this are three input fields: 'JIRA Base URL:', 'Admin Username:', and 'Admin Password:'. The 'Admin Username' field has a note: 'This username must have system administrator rights on your JIRA server.' There is also a link for 'Advanced Options...'. At the bottom right are two buttons: 'Connect to JIRA' and 'Skip'.

[More...](#)

3

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...](#)

4

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.




Seb Ruiz changed 8 files
May 10 2011 00:05

 24826:6cb2f90f9818
 2.6-visualisation on FE-hg

[FECRU-835](#): Add a toolbar dropdown with 'recently active' branches

src/content/.../visualisation/visualisation.jsp	(+31,-0)	History	Source	Diff
src/content/WEB-INF/tags/au/auDropDown.tag	(+4,-1)	History	Source	Diff
src/content/WEB-INF/tags/cru/masthead.tag	(+1,-1)	History	Source	Diff
src/content/static/.../fe/visualisation/branch-set.js	(+8,-8)	History	Source	Diff
src/content/static/.../drawing/branch-header.js	(+22,-3)	History	Source	Diff
src/content/.../visualisation/project-visualiser.js	(+11,-1)	History	Source	Diff
src/content/static/.../style/style-visualisation.css	(+15,-1)	History	Source	Diff
src/java/com/.../vis/ViewRepositoryGraphAction.java	(+45,-21)	History	Source	Diff

[comment on this changeset](#) | [create a review](#)

[show watches](#)

5

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.









[Dashboard](#)
[Source](#)
[Projects](#)
[People](#)
[Reviews](#)

[Create Snippet](#)
[Create Review](#)



Dashboard

[My Reviews](#)

[Inbox \(12\)](#)

[To Review \(4\)](#)

[To Summarize \(1\)](#)

[In Draft \(7\)](#)

[Require My Approval \(0\)](#)

[Outbox \(0\)](#)

[Out For Review \(0\)](#)

[Completed \(0\)](#)

[Archive \(5\)](#)

[Closed \(5\)](#)

[Abandoned \(0\)](#)

[My Snippets](#)

[My Open Snippets \(1\)](#)

[My Snippets \(1\)](#)

[Committer Mappings](#)

[0 committer mappings](#)

[All](#)
[Commits](#)
[Reviews](#)
[Issues](#)
[Expand all](#)
[Show My Activity](#)

Today


Seb Ruiz
1 min ago

 CR-FE-5599 summarized and closed


Nick Pellow
1 min ago

 Re: CR-FE-5593 replied to Tim Pettersen

 done. am going to get this merged to default.

 25776:e29071439734 changed 1 file

 2.6-FECRU-938-review-key-search

 FECRU-938: ensure search for review-key doesn't redirect user. even on the review page with a single result.

 +5 -0

 /test/plugins/fisheye/.../functest/search/ReviewDetailsSearchFuncTest.java


Jason Hinch
1 min ago

 CR-FE-5390 finished reviewing


Seb Ruiz
3 mins ago

 Re: CR-FE-5599 replied to Adam Ahmed

Favourites
Your Favourites are now accessible via the header.

Activity Stream
The Dashboard always shows the activity stream.

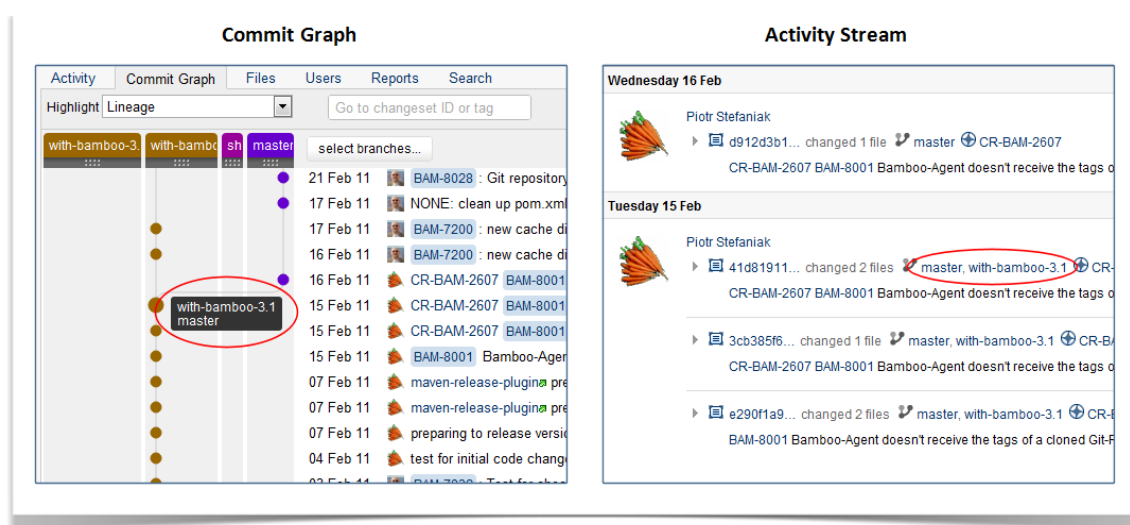
[More...](#)

6

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.



7

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.

8

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.

9

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 for information about any known issues for this release.

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3891	Webwork 2 vulnerability	Vitaly Osipov [Atlassian]	Paul Watson [Atlassian]		 Closed	Fixed	Jan 16, 2012	Jan 31, 2012

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3674	JIRA Activity not showing in ActivityStream when using OAuth	Geoff Crain [Atlassian]	Matthew Watson		 Closed	Fixed	Aug 29, 2011	Sep 01, 2011
	FE-3667	ClearCase: revisions aren't being linked with their parent(s), resulting in incorrect blame	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Aug 24, 2011	Aug 28, 2011
	FE-3664	Quick Search does not respect global or repository's settings for rendering wiki markup	Jason Hinch [Atlassian]	Jason Hinch [Atlassian]		 Closed	Fixed	Aug 23, 2011	Sep 01, 2011
	FE-3662	Cannot create Position that is between 2 identical positions	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Aug 22, 2011	Aug 25, 2011
	FE-3658	ClearCase: no files are indexed for an imported	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Aug 21, 2011	Aug 24, 2011

		repository (using clearfsimport)							
	FE-3653	File files longer than 10000 lines are not being loaded correctly	Jason Hinch [Atlassian]	Jason Hinch [Atlassian]		 Close	Fixed	Aug 17, 2011	Aug 28, 2011
	FE-3652	Cannot Create a review from a ClearCase changes et which contains deletions	Jason Hinch [Atlassian]	Jason Hinch [Atlassian]		 Close	Fixed	Aug 17, 2011	Aug 24, 2011
	FE-3631	Error executing command (cleartool ...) : \"" is not recognized as an internal or external command. On Windows	Unassigned	Rene Verschoor [Atlassian]		 Close	Fixed	Aug 08, 2011	Aug 31, 2011
	FE-3494	Setting minimum font size on Firefox. makes the layout break	Joe Xie [atlassian]	Nicolas Wildner [Atlassian]		 Close	Fixed	Jun 13, 2011	Aug 30, 2011
	FE-3160	ClearCase support for moving and renaming files is lacking	Michael Heemskeerk [Atlassian]	None		 Close	Fixed	Aug 17, 2010	Aug 28, 2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3666	NPE on FishEye setup page coming from JAACS	Unassigned	Geoff Crain [Atlassian]	↓	 Closed	Fixed	Aug 24, 2011	Aug 24, 2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3800	StoreDiff and Loc can't be turned off without a reindex	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Aug 08, 2011	Oct 24, 2011
	FE-3655	IllegalStateExceptions and IllegalArgumentExceptions in activity streams	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]	↑	 Closed	Fixed	Aug 18, 2011	Aug 29, 2011
	FE-3654	JIRA FishEye plugin not working - Error handling trusted applications authentication attempt: BAD_SI	Tim Pettersen [Atlassian]	Tim Pettersen [Atlassian]	↑	 Closed	Fixed	Aug 17, 2011	Jan 24, 2012

		GNATUR E							
	FE-3639	Fishey Crowd integration doesn't work when the group name has a space	Geoff Crain [Atlassian]	Ajay Sridhar [Atlassian]		 Close	Fixed	Aug 10, 2011	Aug 18, 2011
	FE-3636	Empty "tag" definition for Subversion Repos. overrides the trunk RegExp	Geoff Crain [Atlassian]	Nick Pellow [Atlassian]		 Close	Fixed	Aug 09, 2011	Aug 18, 2011
	FE-3634	FishEye implementation of SAL UserProfilePicture.getProfilePictureUri does not return URI relative to base URI	Arun Bhalla [Atlassian]	Arun Bhalla [Atlassian]		 Close	Fixed	Aug 08, 2011	Sep 02, 2011
	FE-3633	FishEye implementation of SAL UserProfilePicture.getPageUri() does not return correct URI	Arun Bhalla [Atlassian]	Arun Bhalla [Atlassian]		 Close	Fixed	Aug 08, 2011	Aug 10, 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)									
Type	Key	Summar	Assigne	Reporter	Priority	Status	Resoluti	Created	Updated

		y	e			on			
	FE-3626	FishEye chokes on a changeset that has an 'empty' branch name	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]		 Close	Fixed	Aug 05, 2011	Aug 09, 2011
	FE-3607	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	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Close	Fixed	Jul 26, 2011	Aug 09, 2011
	FE-3599	Upgrading of CVS repositories leaves file handles open...	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]		 Close	Fixed	Jul 25, 2011	Jul 25, 2011
	FE-3597	Clearcase lshistory parsing can get out of sync with output	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Close	Fixed	Jul 25, 2011	Jul 28, 2011
	FE-3596	HG: Commits with no commit message break indexing	Matthew Watson	Michael Studman [Atlassian]		 Close	Fixed	Jul 22, 2011	Aug 07, 2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3589	Update Commit Graph branch selector to work "live" without the Done button	Michael Studman [Atlassian]	Adam Ahmed [Atlassian]	↓	 Closed	Fixed	Jul 19, 2011	Jul 20, 2011
	FE-3497	Incorrect p4 executable blocks FishEye starting	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]	🛑	 Closed	Fixed	Jun 13, 2011	Jul 04, 2011
	FE-3488	Check Java version compatibility at startup	Unassigned	Rene Verschoor [Atlassian]	↓	 Closed	Fixed	Jun 09, 2011	Jul 19, 2011
	FE-3485	Add repository name next to changesets under Source tab	Jason Hinch [Atlassian]	Gurleen Anand [Atlassian]	↑	 Closed	Fixed	Jun 08, 2011	Jun 23, 2011
	FE-3584	Admin: Empty groups don't have a link to edit them	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]	↑	 Closed	Fixed	Jul 17, 2011	Jul 19, 2011
	FE-3575	Upgrading ancestry for CVS changeset fails on 2.6.1	Michael Heemskeerk [Atlassian]	Gurleen Anand [Atlassian]	↑	 Closed	Fixed	Jul 10, 2011	Jul 18, 2011
	FE-3564	NullPointerException when trying to view the changes	Matthew Watson	Nick Pellow [Atlassian]	↓	 Closed	Fixed	Jul 06, 2011	Jul 11, 2011

		et containing a file revision that has no branches							
	FE-3562	NullPointerException when a changeset has no parents. Causing scanner to restart.	Michael Heemskeerk [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	Jul 06, 2011	Jul 18, 2011
	FE-3559	FishEye can't scan repository. "Cannot create a position that is between two identical positions"	Michael Heemskeerk [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	Jul 05, 2011	Aug 17, 2011
	FE-3558	Mercurial (HG) 1.9 not compatible with FishEye	Matthew Watson	Nick Pellow [Atlassian]		 Closed	Fixed	Jul 05, 2011	Jul 19, 2011
	FE-3540	Files with cyrillics in their names generated warning during the indexing of new commits	Conor MacNeill [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Closed	Fixed	Jun 26, 2011	Jul 19, 2011
	FE-3536	REST documentation is confusing	Seb Ruiz [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Jun 21, 2011	Jun 28, 2011
	FE-3528	NPE with	Joe Xie	Adam		 Closed	Fixed	Jun 16,	Jul 11,

		CVS repo when doing a Lucene index upgrade (when there is some strange head-less state)	[atlassian]	Ahmed [Atlassian]		d		2011	2011
	FE-3518	Legacy Edit Repository default throw exception	Michael Studman [Atlassian]	Zed Yap [Atlassian]		 Closed	Fixed	Jun 15, 2011	Jul 18, 2011
	FE-3507	Polling mechanism skips repositories when there are a lot of them	Tom Davies [Atlassian]	Rene Verschoor [Atlassian]		 Closed	Fixed	Jun 14, 2011	Aug 17, 2011
	FE-3478	Git diff processing fails for diffs with ^M characters in them	Unassigned	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Jun 02, 2011	Jul 19, 2011
	FE-3029	HG: A Commit message ending in a line with only "P:" on it breaks indexing and repo won't index.	Unassigned	Matthew Watson		 Closed	Fixed	Apr 19, 2011	Jul 20, 2011
	FE-2988	ClearCase: Editing a repository config in the admin prompts for a	Michael Studman [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Mar 21, 2011	Sep 26, 2011

		restart of the repository, but triggers a re-index							

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3546	Git - remove branch fails	Anna Buttfield [Atlassian]	Adam Ahmed [Atlassian]	↓	 Closed	Fixed	Jun 29, 2011	Jun 30, 2011
	FE-3543	Nullpointer exception in your showcase / sandbox	Unassigned	Jens Bormüller [avono AG]	↑	 Closed	Fixed	Jun 29, 2011	Jul 06, 2011
	FE-3529	NPE during EyeQL when review has null description	Adam Ahmed [Atlassian]	Adam Ahmed [Atlassian]	↑	 Closed	Fixed	Jun 18, 2011	Jun 19, 2011
	FE-3513	No Horizontal Scroll Bar on Any Non-GWT Admin Pages	Seb Ruiz [Atlassian]	Tom Davies [Atlassian]	↑	 Closed	Fixed	May 26, 2011	Jun 29, 2011
	FE-3510	Login screen after setup has incomplete wording	Seb Ruiz [Atlassian]	Sarah Maddox [Atlassian]	↑	 Closed	Fixed	Jun 02, 2011	Jun 20, 2011
	FE-3505	Alerts boxes show up	Jason Hinch [Atlassian]	None	↓	 Closed	Fixed	Jun 01, 2011	Jun 14, 2011

		on every page for IE 7 saying that the browser isn't supported. Would be incredibly annoying	n]						
	FE-3491	Cannot restore into SQL Server 2008 using built in JAR (without copying the jtds jar into FISHEYE_INST/lib)	Michael Studman [Atlassian]	Adam Ahmed [Atlassian]		 Closed	Fixed	Jun 11, 2011	Dec 19, 2011
	FE-3487	attempt to set vcs executable to non-existent file quietly ignored	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Jun 09, 2011	Jun 15, 2011
	FE-3482	Extra request to admin pages	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Jun 07, 2011	Jun 07, 2011
	FE-3481	QuickNav fails on repo specific pages	Anna Buttfield [Atlassian]	Adam Ahmed [Atlassian]		 Closed	Fixed	May 26, 2011	Jun 21, 2011
	FE-3463	'Save' button on Database Configuration. Edit Admin page doesn't	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	May 26, 2011	Oct 17, 2011

		save the new config							

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 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



- 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

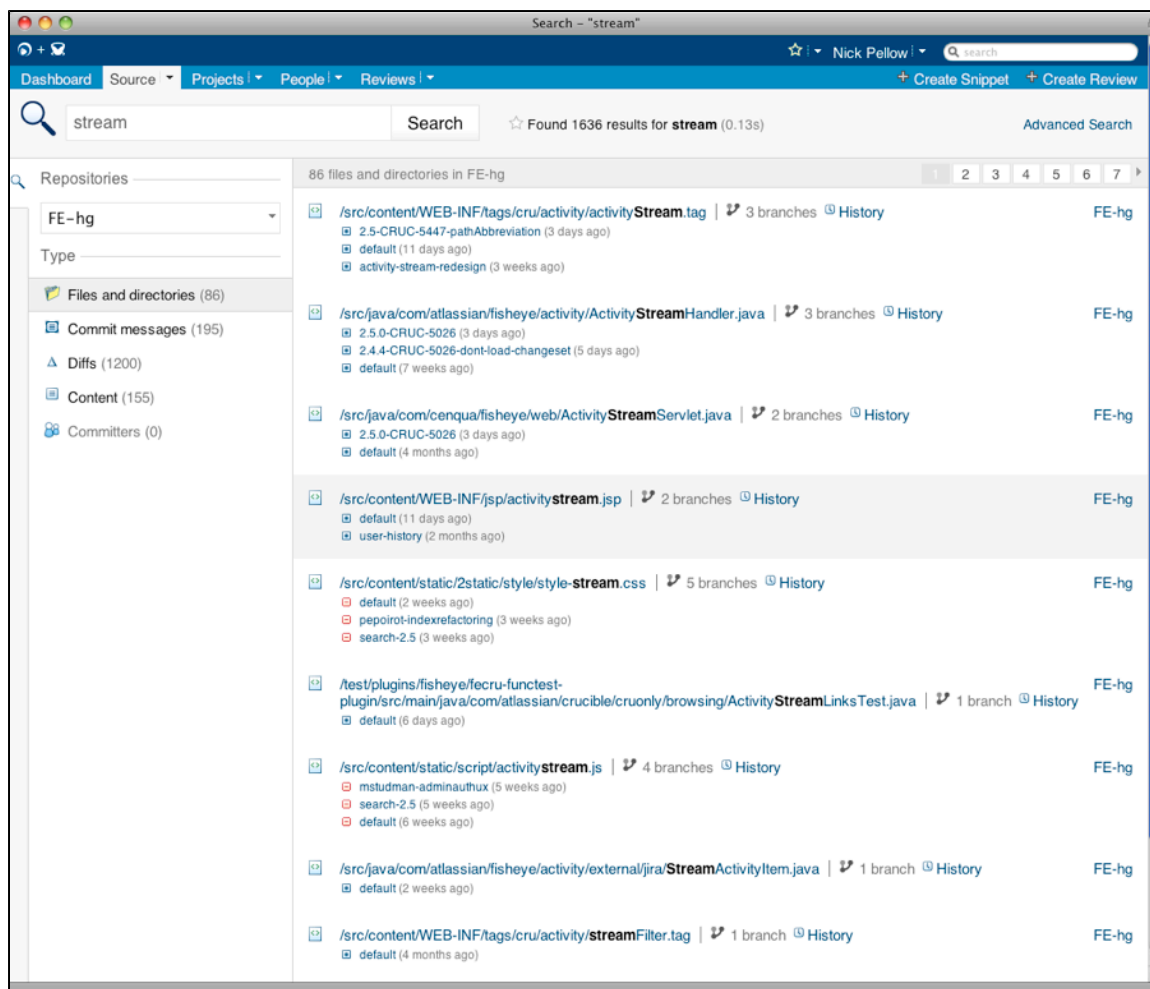


Search Revamp

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...](#)



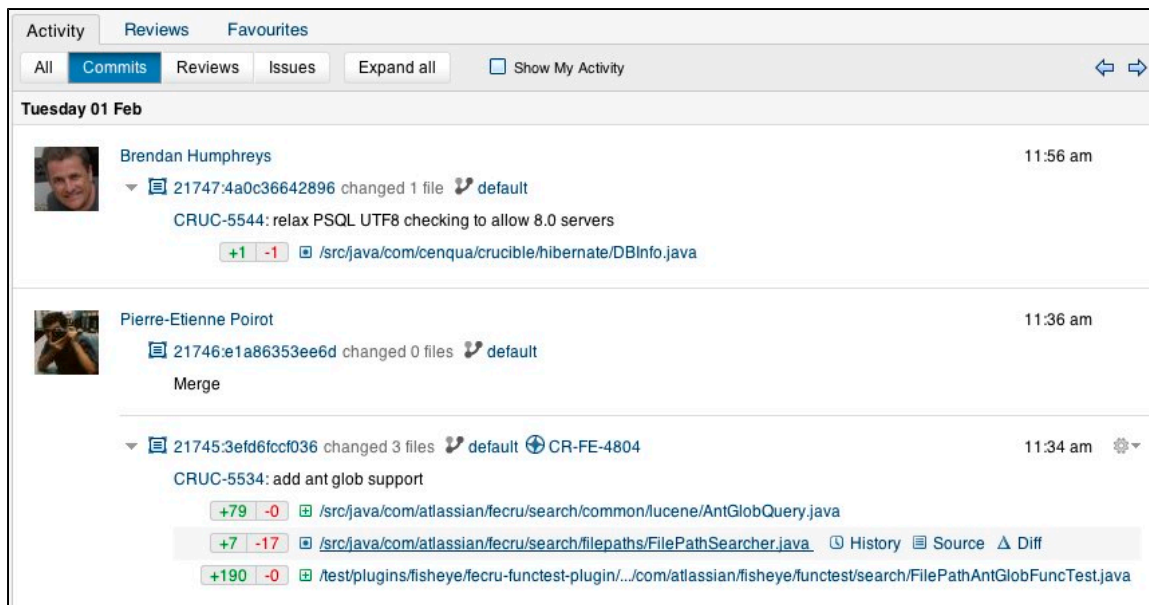
2

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...](#)



Activity | Reviews | Favourites

All | **Commits** | Reviews | Issues | Expand all | ☐ Show My Activity

Tuesday 01 Feb

 **Brendan Humphreys** 11:56 am

21747:4a0c36642896 changed 1 file  default

CRUC-5544: relax PSQL UTF8 checking to allow 8.0 servers

+1 -1 [/src/java/com/cenqua/crucible/hibernate/DBInfo.java](#)

 **Pierre-Etienne Poirot** 11:36 am

21746:e1a86353ee6d changed 0 files  default

Merge

21745:3efd6f6cf036 changed 3 files  default  CR-FE-4804 11:34 am 

CRUC-5534: add ant glob support

+79 -0 [/src/java/com/atlassian/feclu/search/common/lucene/AntGlobQuery.java](#)

+7 -17 [/src/java/com/atlassian/feclu/search/filepaths/FilePathSearcher.java](#) [History](#) [Source](#) [Diff](#)

+190 -0 [/test/plugins/fisheye/feclu-functest-plugin/.../com/atlassian/fisheye/functest/search/FilePathAntGlobFuncTest.java](#)

3

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:

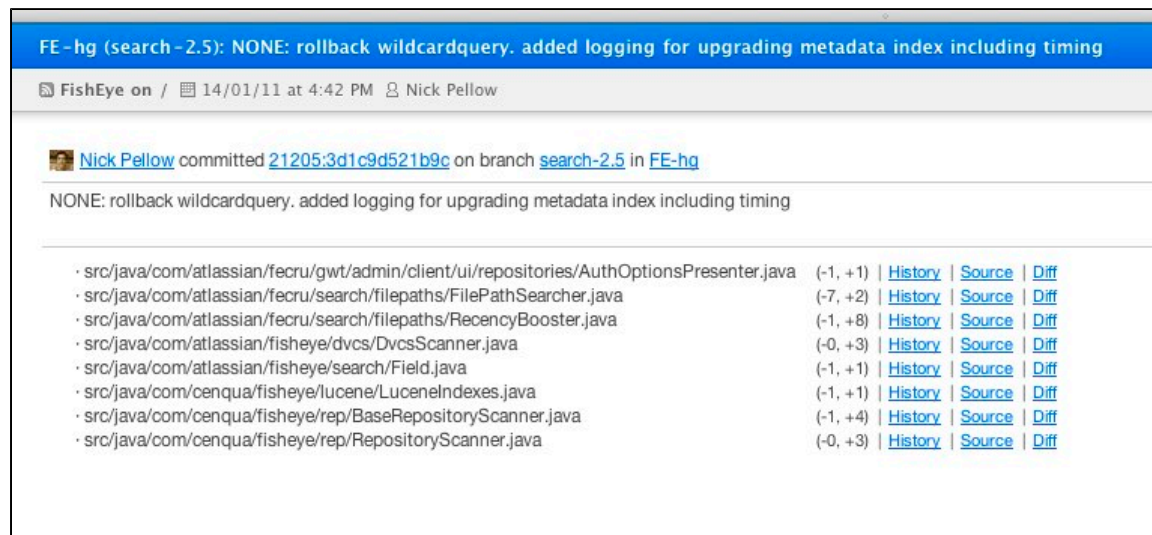
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQgQC0Zf6OPjwI2jPxMt4ZRk6Y1IQDwXMfb7alxCjfs0YYI
Qw9vy08iB5VI/xbLBbqv7dvc00uJ8gLIODzPGXoBa8r6tCiCN4K2eG7tP5YNA8qpR2HyOMf27oD
+jEMP2caYgFBwtNmVpVAJxrlaHn0hMjCFqJYU7+U5VKKKeuThb0hCw== ttt on FishEye at
https://extranet.atlassian.com/crucible

Copy this public key to your repository server
Remove

4

RSS Improvements

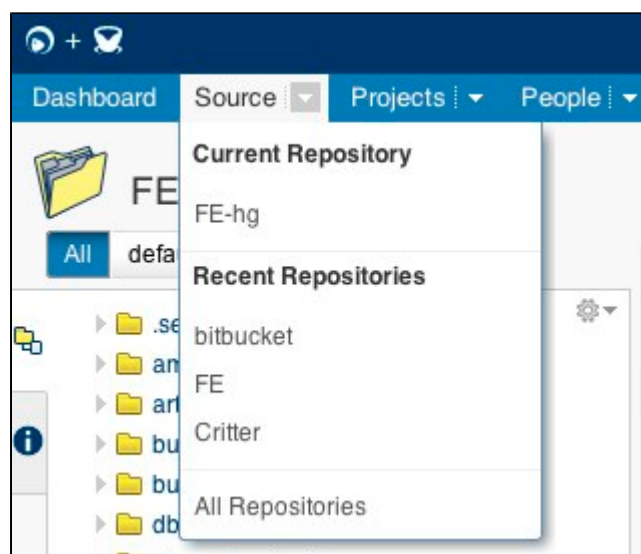
FishEye 2.5 introduces some fine tuning for rendered RSS feeds. RSS content is easier to digest (better titles) and allows you to consume FishEye activity through your favourite RSS reader. We render wiki-markup in RSS which will allow you to click through links and view images inline.



5

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3529	NPE during EyeQL when review has null description	Adam Ahmed [Atlassian]	Adam Ahmed [Atlassian]		 Closed	Fixed	Jun 18, 2011	Jun 19, 2011
	FE-3487	attempt to set vcs executable to non-existent file quietly ignored	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Jun 09, 2011	Jun 15, 2011
	FE-3482	Extra request to admin pages	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Jun 07, 2011	Jun 07, 2011
	FE-3463	'Save'	Tom	Tom		 Closed	Fixed	May 26,	Oct 17,

		button on Database Configuration. Edit Admin page doesn't save the new config	Davies [Atlassian]	Davies [Atlassian]		d		2011	2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3442	Do not throw exception when git content cannot be fetched.	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	May 16, 2011	May 19, 2011
	FE-3439	Prevent SSH from dropping down to password auth when using SSH keys	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	May 12, 2011	May 17, 2011
	FE-3453	Admin: Repository view: Clicking refresh causes same status message to repeat	Pierre-Etienne Poirot [Atlassian]	Florian Dejako	↓	 Closed	Fixed	May 19, 2011	May 20, 2011
	FE-3444	Handle unversioned	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	May 17, 2011	May 17, 2011

		element in Isactivity							
	FE-3436	500 error in the search (java.lang.NullPointerException)	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]	↑	 Closed	Fixed	May 11, 2011	May 26, 2011
	FE-3345	ClearCase: custom element types are treated as binary files.	Conor MacNeill [Atlassian]	Michael Heemskeerk [Atlassian]	↑	 Closed	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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3811	Permission Checking Bug in FishEye Changeset Tooltips	Vitaly Osipov [Atlassian]	Paul Watson [Atlassian]	↑	 Closed	Fixed	Nov 01, 2011	Nov 30, 2011
	FE-3798	XSS vulnerability in a user's comment	Vitaly Osipov [Atlassian]	Paul Watson [Atlassian]	↑	 Closed	Fixed	Oct 24, 2011	Nov 30, 2011
	FE-3797	XSS vulnerability in user's profile display name	Vitaly Osipov [Atlassian]	Paul Watson [Atlassian]	↑	 Closed	Fixed	Oct 24, 2011	Nov 30, 2011
	FE-3432	Perforce escape for * is case sensitive	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	May 10, 2011	May 10, 2011
	FE-3028	Gutter	Tom	Julien	↑	 Closed	Fixed	Apr 18,	May 09,

		renderer breaks the source layout	Davies [Atlassian]	Hoarau [Old]		d		2011	2011
	FE-3026	Search hint: when a hint is specified, clicking on a different search type does not do anything	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Closed	Fixed	Apr 17, 2011	Apr 18, 2011
	FE-3025	FishEye watch notifications can potentially not send changesets due to long running incremental slurps	Jason Hinch [Atlassian]	Jason Hinch [Atlassian]		 Closed	Fixed	Apr 14, 2011	May 05, 2011
	FE-3023	Annotation page - resizing window in FE causes double scrollbars and/or a really long page.	Adam Ahmed [Atlassian]	Adam Ahmed [Atlassian]		 Closed	Fixed	Apr 11, 2011	May 09, 2011
	FE-3021	AJP authentication regression	Jason Hinch [Atlassian]	Jason Hinch [Atlassian]		 Closed	Fixed	Apr 11, 2011	Apr 28, 2011

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)

Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3014	add repository name back to activity items	Adam Ahmed [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	Apr 06, 2011	Apr 06, 2011
	FE-3381	IE8 source view window resize bug	Unassigned	None		 Closed	Fixed	Jan 27, 2011	May 09, 2011
	FE-3032	XSS vulnerability in FishEye/ Crucible Reviews List	Vitaly Osipov [Atlassian]	Andrew Lui [Atlassian]		 Closed	Fixed	Apr 20, 2011	May 30, 2011
	FE-3031	XSS vulnerability in FishEye/ Crucible dashboard - review activity	Vitaly Osipov [Atlassian]	Andrew Lui [Atlassian]		 Closed	Fixed	Apr 20, 2011	May 30, 2011
	FE-3011	Diff View - deeplink to a line using an 'l' hash (e.g., #l98) causes js error	Adam Ahmed [Atlassian]	Adam Ahmed [Atlassian]		 Closed	Fixed	Apr 04, 2011	Apr 04, 2011
	FE-3000	clearcase ucm: we aren't storing the latest possible activity time, so it's possible to get a	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	Mar 28, 2011	Mar 29, 2011

		lot of unnecessary data							
	FE-2997	FeCru TransactionTemplate implementation should close the Hibernate Session opened during the transaction	Tom Davies [Atlassian]	Richard Wallace [Atlassian]	↓	 Closed	Fixed	Mar 25, 2011	Apr 13, 2011
	FE-1752	In diff view, selecting a context clears any ignore settings, and vice versa	Adam Ahmed [Atlassian]	Anna Buttfield [Atlassian]	↑	 Closed	Fixed	Jun 22, 2009	Mar 31, 2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3805	DB migration : fix incorrect message "The database is case insensitive. Crucible requires that the database use a case	Pierre-Etienne Poirot [Atlassian]	Murat Uysal	↓	 Closed	Fixed	Oct 27, 2011	Oct 31, 2011

		sensitive collation" to reflect the current check: "The database is not Unicode. Crucible requires ..."							
	FE-3250	DB migration : fix incorrect message "The database is case insensitive. Crucible requires that the database use a case sensitive collation" to reflect the current check: "The database is not Unicode. Crucible requires ..."	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Closed	Fixed	Mar 24, 2011	Oct 27, 2011
	FE-3075	NullPointerException in GroupEditorAction	Pierre-Etienne Poirot [Atlassian]	None		 Closed	Fixed	Mar 20, 2011	May 09, 2011
	FE-3052	Recent visited tabs spins forever when REST endpoints	Pierre-Etienne Poirot [Atlassian]	None		 Closed	Fixed	Mar 15, 2011	May 09, 2011

		are disabled							
	FE-2998	clearcase base: use the latest repository activity when indexing branches	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	Mar 27, 2011	Mar 27, 2011
	FE-2987	UCM ClearCase: specifying a project inclusion rule and then saving the repository results in an invalid config.xml	Conor MacNeill [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Mar 21, 2011	Mar 24, 2011
	FE-2985	Typo: Scheduled Backups	Tim Pettersen [Atlassian]	Matt Doar (Custom Ware)		 Closed	Fixed	Mar 18, 2011	Mar 20, 2011
	FE-2981	HG indexing fails with "not found in manifest!" error	Matthew Watson	Matthew Watson		 Closed	Fixed	Mar 15, 2011	Mar 23, 2011
	FE-2974	UCM ClearCase: indexing error 'No such file or directory'	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Mar 08, 2011	Mar 11, 2011
	FE-2972	Go-To-Changeset doesn't work	Matthew Watson	Erik van Zijst [Atlassian]		 Closed	Fixed	Mar 06, 2011	Mar 24, 2011
	FE-2966	Deadlock in Jetty	Joe Xie [atlassian]	Gurleen Anand [Atlassian]		 Closed	Fixed	Feb 24, 2011	Mar 21, 2011

		connection timeout		n]					
	FE-2954	LineDecorator.decorateLine(int) never call	Tom Davies [Atlassian]	Julien Hoarau [Old]		 Closed	Fixed	Feb 16, 2011	Mar 28, 2011

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3255	Add option to disable git head detection and tag processing	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Feb 03, 2011	May 09, 2011
	FE-3343	UCM ClearCase: FishEye doesn't index project components that aren't components of the project's PVOB	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Feb 06, 2011	May 09, 2011
	FE-3157	Git isn't detecting binary files at all	Michael Heemskeerk [Atlassian]	None		 Closed	Fixed	Feb 02, 2011	May 09, 2011
	FE-3123	Anchoring Patch to Git Repo Causes NPE	Geoff Crain [Atlassian]	None		 Closed	Fixed	Feb 02, 2011	May 09, 2011
	FE-2976	NPE during Mercurial indexing	Unassigned	Michael Heemskeerk		 Closed	Fixed	Mar 10, 2011	May 18, 2011

				rk [Atlassian]					
	FE-2969	Error trying to connect to HG repo using ssh on windows	Geoff Crain [Atlassian]	Matthew Watson		 Close	Fixed	Mar 03, 2011	Mar 04, 2011
	FE-2964	ClearCase: FishEye doesn't handle versions that are deleted from a ClearCase VOB	Michael Heemske rk [Atlassian]	Michael Heemske rk [Atlassian]		 Close	Fixed	Feb 24, 2011	Mar 03, 2011
	FE-2960	ClearCase: cleartool error message is missed by FishEye and as a result indexing stops in error for known 'error' conditions	Michael Heemske rk [Atlassian]	Michael Heemske rk [Atlassian]		 Close	Fixed	Feb 22, 2011	Mar 03, 2011
	FE-2956	P4 Unicode Handling	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Close	Fixed	Feb 17, 2011	Mar 02, 2011
	FE-2948	java.lang.IllegalStateException - No match found exception thrown when the clearcase .process Branch	Michael Heemske rk [Atlassian]	Gurleen Anand [Atlassian]		 Close	Fixed	Feb 09, 2011	Mar 03, 2011

		command times out after 1 hour							

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)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	CRUC-5612	Make Oracle multodb tests support testing of upgrades	Michael Studman [Atlassian]	Michael Studman [Atlassian]	↓	 Closed	Fixed	Feb 09, 2011	Feb 16, 2011
	CRUC-5609	Review rss feed provides no way to navigate to review in thunderbird	Seb Ruiz [Atlassian]	Brydie McCoy [Atlassian]	↑	 Closed	Fixed	Feb 08, 2011	Feb 09, 2011
	CRUC-5608	Cannot Edit Per-Repository Authentication Settings	Jason Hinch [Atlassian]	Dylan Hansen [Atlassian]	↓	 Closed	Fixed	Feb 08, 2011	Jun 23, 2011
	CRUC-5599	Upgrade scripts do not always create the same schema as scripts to create a new database	Michael Studman [Atlassian]	Michael Studman [Atlassian]	↓	 Closed	Fixed	Feb 07, 2011	Feb 16, 2011
	CRUC-5597	The "Tabs" on the statistics panel of the "Project	Joe Xie [atlassian]	Brenden Bain [Atlassian]	↓	 Closed	Fixed	Feb 06, 2011	Feb 13, 2011

		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	Luis Miranda [Atlassian]	Brenden Bain [Atlassian]		 Close	Fixed	Feb 04, 2011	Feb 13, 2011
	CRUC-5581	UAL webitem link for fisheye configuration links to a page that shows all possible options, even if those options are global and will not apply to the instance being edited	Luis Miranda [Atlassian]	Joe Xie [atlassian]		 Close	Fixed	Feb 02, 2011	Feb 13, 2011
	CRUC-5574	Update "refresh cache" links to synch form App Links as well.	Joe Xie [atlassian]	Mark Lassau [Atlassian]		 Close	Fixed	Feb 02, 2011	Feb 15, 2011
	CRUC-5526	FishEye instance gets re-create	Joe Xie [atlassian]	Luis Miranda [Atlassian]		 Close	Fixed	Jan 30, 2011	Feb 13, 2011

		d on JIRA restart							
	CRUC-5475	Create func tests for all search result pages	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]		 Close	Fixed	Jan 21, 2011	Feb 20, 2011
	CRUC-5426	Add a link in admin UI to UAL admin screen	Joe Xie [atlassian]	Luis Miranda [Atlassian]		 Close	Fixed	Jan 18, 2011	Feb 15, 2011
	CRUC-5400	On Plugin startup, synchronise App Links	Joe Xie [atlassian]	Mark Lassau [Atlassian]		 Close	Fixed	Jan 17, 2011	Feb 13, 2011
	CRUC-5188	Update admin UI	Mark Lassau [Atlassian]	Joe Xie [atlassian]		 Close	Fixed	Dec 13, 2010	Feb 13, 2011
	CRUC-5183	Extend oauth services to REST resources (such as gadgets)	Joe Xie [atlassian]	Joe Xie [atlassian]		 Close	Fixed	Dec 13, 2010	Feb 13, 2011
	CRUC-5156	upgrade tasks to create Application Links for existing legacy links on plugin startup	Joe Xie [atlassian]	Joe Xie [atlassian]		 Close	Fixed	Dec 12, 2010	Feb 13, 2011
	CRUC-5154	Add support for OAuth authentication	Joe Xie [atlassian]	Joe Xie [atlassian]		 Close	Fixed	Dec 12, 2010	Feb 13, 2011
	CRUC-5026	ActivityStreamHandler incorrectly uses	Tom Davies [Atlassian]	Brendan Humphreys [Atlassian]		 Close	Fixed	Nov 23, 2010	Nov 14, 2011

		Source.g etChang eset(csid) to validate a changes et ID						

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](#) 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 2.5 Release Notes](#)

FishEye 2.4 Release Notes

20 October 2010

With great pleasure, Atlassian presents FishEye 2.4, now with even more improvements to make [browsing your source code](#) easier, especially for [Mercurial and Git users](#).

Highlights of this Release:

- [Branch and Tag Selector](#)
- [File History Redesigned](#)
- [Easier Application Linking](#)
- [SSL Support](#)
- [User Interface Improvements](#)
- [Performance Improvements](#)
- [And Even More Improvements](#)

Responding to your Feedback:

★ Over 80 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.

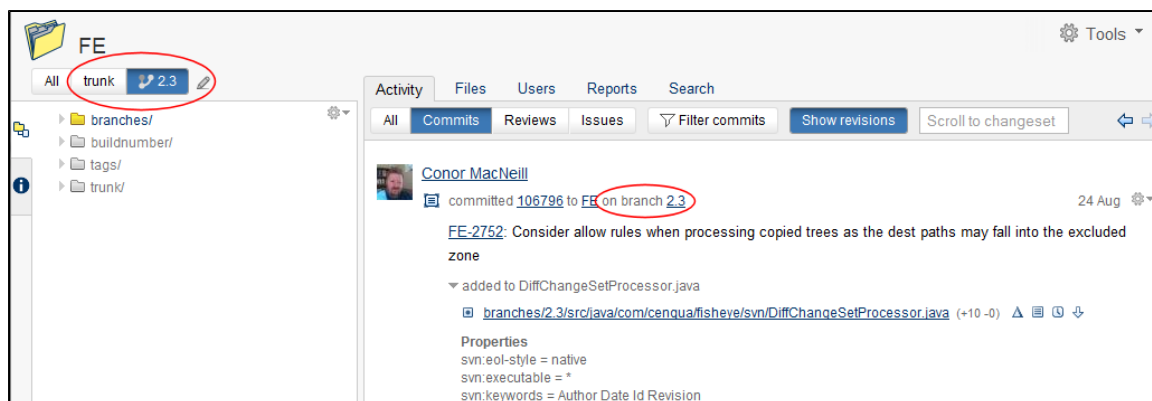
Highlights of FishEye 2.4



Branch and Tag Selector

DVCS users will find it easier to work with their repositories using FishEye's new branch and tag selector. The branch and tag selector allows you to view information only related to a specific branch/tag when browsing your repositories. FishEye will attempt to detect the main branch, e.g. master on Git, default on Mercurial or MAIN on CVS, but you can also select any branch manually. The branch/tag is kept in context when navigating throughout the application, e.g. if you switch from viewing a repository to viewing the activity stream, only information related to the selected branch will be displayed.

Note, this feature currently does not work for ClearCase repositories due to technical constraints.

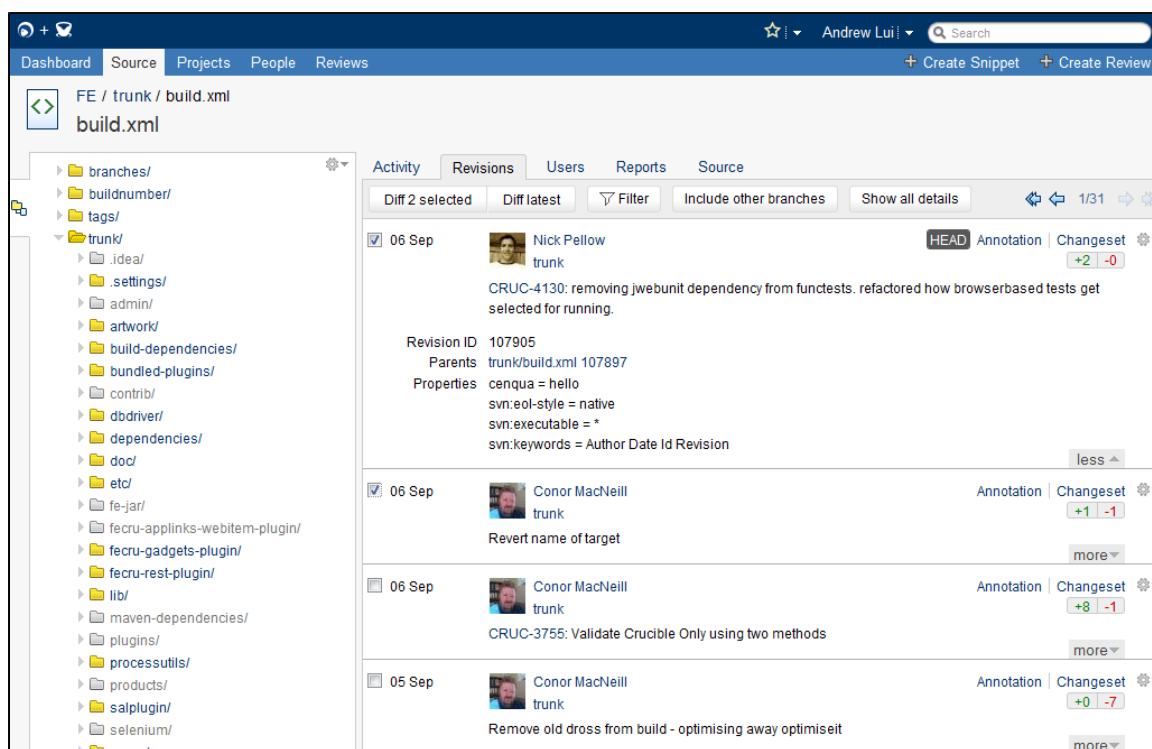


[More...](#)

2

File History Redesigned

The file history display in FishEye has been redesigned to provide a cleaner and more intuitive interface. The neatly summarised list makes it easier for you to find the information you need, at a glance. You can also use the new filter to find the desired file revision, or expand file revisions to display additional information.



[More...](#)

3

Easier Application Linking

FishEye now includes a brand new version of the Application Links plugin. You can use this plugin to easily link your FishEye server to other applications, like a [JIRA](#) server or another FishEye server. You can choose between the Trusted Applications protocol, OAuth or basic HTTP authentication. Linking two applications allows you to share information and access one application's functions from within the other. For example, if you linked your FishEye instance with a JIRA server, you could view JIRA issues in your FishEye activity stream or view the changesets associated with an issue in JIRA.

Configure Application Links 

 Add Application Link

Name	Application	Application URL	Configured Authentication 	Primary	Actions
 STAC	JIRA	https://studio.atlassian.com	Basic Access		Configure Delete Make Primary JIRA Settings
 jac	JIRA	http://jira.atlassian.com	OAuth Basic Access		Configure Delete JIRA Settings

[More....](#)



SSL Support

By [popular request](#), this FishEye release provides full SSL support. You will now be able to configure SSL for your FishEye instance using a self-signed or certificate-authority certificate.

Server Settings

Edit Web Settings

Http Configuration

Web context:

HTTP Bind Address:

Ssl Configuration

SSL Bind Address:

SSL Keystore File:

Keystore Password:

Keystore Password Confirm:

SSL Truststore File:

Truststore Password:

Truststore Password Confirm:

Common Configuration

Proxy scheme:

Proxy host:

Proxy port:

Ajp13 Bind Address:

Allow remote API calls: ☒ On ☐ Off

Timezone:
 

Site URL:

Note: changes to these settings requires a restart to take effect.

[More....](#)

5

User Interface Improvements

We're continuing our ongoing work to improve the FishEye user interface. This release includes a brand new repository administration console and a redesigned header. We've also improved the toolbars on a number of screens including the dashboard, as well as replaced dropdowns throughout the application with autocomplete controls.

Repositories

Native (17) Plugins (5)

FishEye and Crucible natively supports Subversion, CVS, ClearCase, Perforce, Git and Mercurial repositories 🟢

Name	Type	Description	Location	State	Last Update
AGMP	Subversion	a good marketing ploy	https://studio.atlassian.com/svn/AGMP	Disabled	
applinks	Subversion		https://studio.atlassian.com/svn/APL	Disabled	
atlassian	Subversion		https://svn.atlassian.com/svn/public/atlassian	Running	40 seconds ago
BM	Subversion	Build Monitor	https://labs.atlassian.com/svn/BM	Running	47 seconds ago
CLOV	Subversion	Clover	https://studio.atlassian.com/svn/CLOV	Running	19 seconds ago
contrib	Subversion		https://svn.atlassian.com/svn/public/contrib	Disabled	
Critter	Git	Crucible Command Line Client	git://github.com/sebr/critter.git/	Running	16 seconds ago
FE	Subversion	FishEye	https://studio.atlassian.com/svn/FE	Running	13 seconds ago
FE-hg	Mercurial	FE from hgsubversion conversion	/opt/hg-bb-clone/fe/	Running	46 seconds ago
FE-hg-bb	Mercurial	FE indexed from a clone of the bitbucket hosted repo	/home/mwatson/fe/	Disabled	
FEP	Subversion	JIRA FishEye Plugin	https://studio.plugins.atlassian.com/svn/FISH	Running	28 seconds ago

Add...

6

Performance Improvements

This FishEye release includes a number of performance improvements. We've been tinkering under the hood to improve page rendering. Viewing file annotations will be significantly faster, once you have loaded the relevant page, as we have implemented progressive rendering for the annotations. Pagination controls have also been added to the bottom of long pages, like the activity stream.

Activity	Revisions	Users	Reports	Source
All	Commits	Reviews	Issues	Filter commits
Show revisions				Scroll to changeset
 				
	Tim Pettersen			
	summarized and closed CR-FE-3698	27 Sep		
No summary.				
	summarized and closed CR-FE-3683	27 Sep		
No summary.				

7

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

- **Security advisory.** This release fixes a security flaw. Please refer to the [security advisory](#) for details of the security vulnerability, risk assessment and mitigation strategies.
- **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 2.4 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 2.4 Changelog

This page contains information about the FishEye 2.4 minor releases.

 Please read the [FishEye 2.4 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 2.4.5 to 2.4.6](#)
- [From 2.4.4 to 2.4.5](#)
- [From 2.4.3 to 2.4.4](#)
- [From 2.4.2 to 2.4.3](#)
- [From 2.4.1 to 2.4.2](#)
- [From 2.4.0 to 2.4.1](#)

From 2.4.5 to 2.4.6

11 April 2011

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

JIRA Issues (6 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3032	XSS vulnerability in FishEye/ Crucible Reviews List	Vitaly Osipov [Atlassian]	Andrew Lui [Atlassian]		 Closed	Fixed	Apr 20, 2011	May 30, 2011
	FE-3031	XSS vulnerability in FishEye/ Crucible dashboard - review activity	Vitaly Osipov [Atlassian]	Andrew Lui [Atlassian]		 Closed	Fixed	Apr 20, 2011	May 30, 2011
	CRUC-5	XSS	Vitaly	Andrew		 Closed	Fixed	Apr 20,	May 30,

	735	vulnerability in Crucible Author Mapping	Osipov [Atlassian]	Lui [Atlassian]		d		2011	2011
	CRUC-5734	XSS vulnerability in Crucible's Snippets	Vitaly Osipov [Atlassian]	Andrew Lui [Atlassian]		 Closed	Fixed	Apr 20, 2011	May 30, 2011
	CRUC-5616	Empty responses from p4 commands cause NullPointerException	Tom Davies [Atlassian]	Tom Davies [Atlassian]		 Closed	Fixed	Feb 10, 2011	Apr 08, 2011
	CRUC-5429	Error generating FishEye backup on Windows	Adam Ahmed [Atlassian]	Felipe Kraemer [Atlassian]		 Closed	Fixed	Jan 18, 2011	Apr 06, 2011

From 2.4.4 to 2.4.5

3 February 2011

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

JIRA Issues (8 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3172	ClearCase: symbolic links cause errors during indexing	Michael Heemskeerk [Atlassian]	None		 Closed	Fixed	Apr 07, 2010	May 09, 2011
	CRUC-5737	XSS vulnerability in Crucible Comments Search	Vitaly Osipov [Atlassian]	Andrew Lui [Atlassian]		 Closed	Fixed	Apr 20, 2011	May 15, 2011
	CRUC-5736	XSS vulnerability in Crucible	Vitaly Osipov [Atlassian]	Andrew Lui [Atlassian]		 Closed	Fixed	Apr 20, 2011	May 30, 2011

		changes et comment s in search results							
	CRUC-5392	Improve notification DB query performance	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Jan 14, 2011	Feb 01, 2011
	CRUC-5362	Check UCM root dirs against include rules to avoid needless lshistory calls	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↑	 Closed	Fixed	Jan 11, 2011	Feb 01, 2011
	CRUC-5359	REST API CreateReview issues	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]	↓	 Closed	Fixed	Jan 11, 2011	Feb 01, 2011
	CRUC-5317	Can't Create Reviews via REST API using JSON	Conor MacNeill [Atlassian]	Dylan Hansen [Atlassian]	↑	 Closed	Fixed	Jan 05, 2011	Sep 01, 2011
	CRUC-5117	Image files (jpg) not completely visible in source tab	Adam Ahmed [Atlassian]	Gurleen Anand [Atlassian]	↑	 Closed	Fixed	Dec 02, 2010	Apr 03, 2011

From 2.4.3 to 2.4.4

11 January 2011

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

The release includes fixes for a number of security issues. Please see the [Security Advisory](#) for more information.

 This release includes a number fixes of issues related to **git** support. FishEye users with git repositories should consider a full re-index of their repositories if they have previously experienced any problems. Contact Atlassian Support via <http://support.atlassian.com> if any problems persist.

JIRA Issues (36 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3420	Disable HTTP chunking when sending SOAP requests to JIRA	Jason Hinch [Atlassian]	Pierre-Etienne Poirot [Atlassian]	↓	 Closed	Fixed	Dec 01, 2010	May 09, 2011
	FE-3376	GWT Admin: Some styling touchups needed on filterable/pageable repository lists	Unassigned	None	↓	 Closed	Fixed	Oct 28, 2010	May 09, 2011
	FE-3362	Improve the Git documentation	Andrew Lui [Atlassian]	Andrew Lui [Atlassian]	↑	 Closed	Fixed	Sep 22, 2010	May 09, 2011
	FE-3355	ClearCase: indexing stops when a branch is defined but never used in a VOB.	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]	↑	 Closed	Fixed	Jan 03, 2011	May 09, 2011
	FE-3248	ClearCase UCM activity support is brittle (detection of modified activities etc.)	Unassigned	Michael Heemskeerk [Atlassian]	↓	 Closed	Fixed	Aug 27, 2010	May 09, 2011
	FE-3187	Base ClearCase: File revisions from excluded branches are not	Michael Heemskeerk [Atlassian]	None	↑	 Closed	Fixed	Sep 16, 2010	May 09, 2011

		picked up when merged to an included branch							
	FE-3176	Copy and rename detection for Git repositories is broken	Jason Hinch [Atlassian]	None	↓	 Close	Fixed	Nov 30, 2010	May 09, 2011
	FE-3171	Repositories created with . in the name cannot be loaded in Admin Console	Michael Studman [Atlassian]	None	↑	 Close	Fixed	Nov 12, 2010	May 09, 2011
	FE-3170	ClearCase: ancestor relations are not created for CC file revisions	Michael Heemskeerk [Atlassian]	None	↑	 Close	Fixed	Oct 14, 2010	May 09, 2011
	FE-3138	When using git, iterating over the manifest (for tags or branch heads) is too slow	Michael Heemskeerk [Atlassian]	None	↓	 Close	Fixed	Dec 08, 2010	May 09, 2011
	FE-3086	ClearCase: checkins of different users may be grouped into the same changeset	Unassigned	None	↑	 Close	Fixed	Nov 03, 2010	May 09, 2011
	FE-3079	Fisheye	Jason	None	↑	 Close	Fixed	Nov 29,	May 09,

		Incompatible with Git 1.7.3 on Windows	Hinch [Atlassian]			d		2010	2011
	FE-1883	Option to disable the people tab for security reasons	Unassigned	Ryan Chee [Atlassian]		 Closed	Fixed	Jul 03, 2009	Mar 03, 2011
	CRUC-5312	XSS vulnerability in FishEye/Crucible Code Macro	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 05, 2011	Jan 11, 2011
	CRUC-5310	XSS vulnerability in FishEye Repository Configuration	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 04, 2011	Jan 11, 2011
	CRUC-5307	XSS vulnerability in Crucible's Linked JIRA Issue	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 04, 2011	Jan 11, 2011
	CRUC-5255	clearcase performance of lsactivity is bad	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	Dec 20, 2010	Jan 31, 2011
	CRUC-5254	Add repository name to log messages created during indexing.	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Dec 20, 2010	Dec 23, 2010
	CRUC-5252	UserLoggedInCondition does not work for plugins	Michael Studman [Atlassian]	Michael Studman [Atlassian]		 Closed	Fixed	Dec 20, 2010	Jan 11, 2011
	CRUC-5	activity	Geoff	Geoff			Fixed	Dec 19,	Jan 11,

	246	stream shows duplicate commits in cvs when commits are aggregated.	Crain [Atlassian]	Crain [Atlassian]		 Closed		2010	2011
	CRUC-5244	clearcase indexing doesnt clear its changes cache in the right spot	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	Dec 19, 2010	Jan 09, 2011
	CRUC-5216	problems with permissions	Geoff Crain [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	Dec 15, 2010	Jan 06, 2011
	CRUC-5162	Rescan commit comments in Perforce repositories updates no working	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Dec 12, 2010	Jun 30, 2011
	CRUC-5134	JavaScript errors when loading admin repository tables with hundreds of repositories	Michael Studman [Atlassian]	Michael Studman [Atlassian]		 Closed	Fixed	Dec 09, 2010	Dec 14, 2010
	CRUC-5069	Login screen messaging broken	Craig Sharkie [Atlassian]	Brendan Humphreys [Atlassian]		 Closed	Fixed	Nov 29, 2010	Feb 06, 2011
	CRUC-5065	Git indexing errors (parents not	Michael Heemskeerk [Atlassian]	Anna Butfield [Atlassian]		 Closed	Fixed	Nov 29, 2010	Jan 06, 2011

		found. stack overflows)							
	CRUC-5009	Help links in admin are failing	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Nov 24, 2010	Jan 06, 2011
	CRUC-4993	Git - error when switch directories	Jason Hinch [Atlassian]	Adam Ahmed [Atlassian]		 Close	Fixed	Nov 21, 2010	Dec 12, 2010
	CRUC-4951	changes et view: when a file has a really long name/path the options overflow covering content	Craig Sharkie [Atlassian]	Brydie McCoy [Atlassian]		 Close	Fixed	Nov 15, 2010	Dec 13, 2010
	CRUC-4904	Trusted apps jira integration docs are missing paths critical to sub-task creation	Unassigned	Anna Lyons [Atlassian]		 Close	Fixed	Nov 11, 2010	Nov 30, 2010
	CRUC-4900	IE8 deleted files aren't greyed out on the filetree	Craig Sharkie [Atlassian]	Tim Pettersen [Atlassian]		 Close	Fixed	Nov 11, 2010	Dec 13, 2010
	CRUC-4894	font shiftiness on eyeql results	Craig Sharkie [Atlassian]	Tim Pettersen [Atlassian]		 Close	Fixed	Nov 10, 2010	Dec 13, 2010
	CRUC-4891	ie8 - black waiting screen with spinner	Craig Sharkie [Atlassian]	Tim Pettersen [Atlassian]		 Close	Fixed	Nov 10, 2010	Dec 13, 2010

		off to the side before creating a review							
	CRUC-4871	Can't click on stars or permalinks on review page in IE8	Craig Sharkie [Atlassian]	Tim Pettersen [Atlassian]		 Closed	Fixed	Nov 10, 2010	Dec 13, 2010
	CRUC-4865	Loading spinner does not disappear on Binary Files	Jason Hinch [Atlassian]	Nick Pellow [Atlassian]		 Closed	Fixed	Nov 10, 2010	Dec 12, 2010
	CRUC-4462	Authors/moderators should be able to complete reviews	Jason Hinch [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Oct 05, 2010	Dec 12, 2010

From 2.4.2 to 2.4.3**26 November 2010**

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

JIRA Issues (26 issues)

Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3062	The way Fisheye handles newlines when parsing command line outputs is completely wrong	Jason Hinch [Atlassian]	None		 Closed	Fixed	Nov 08, 2010	May 09, 2011
	FE-3048	"Inconsistent state while finding the revision for file"	Michael Heemskeerk [Atlassian]	None		 Closed	Fixed	Nov 17, 2010	May 09, 2011

		with Git repositories							
	CRUC-5309	XSS vulnerability in Crucible Review Reload	Unassigned	Andrew Lui [Atlassian]		 Close	Fixed	Jan 04, 2011	Jan 11, 2011
	CRUC-5008	File history review links are misstyled	Seb Ruiz [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Nov 24, 2010	Nov 24, 2010
	CRUC-5003	Crucible and FishEye dashboard pages have missing or partial ghost features	Craig Sharkie [Atlassian]	Craig Sharkie [Atlassian]		 Close	Fixed	Nov 23, 2010	Nov 25, 2010
	CRUC-4998	ChangesetTooltip Action.findChangeSet() eagerly creates cruRevisions	Seb Ruiz [Atlassian]	Brendan Humphreys [Atlassian]		 Close	Fixed	Nov 22, 2010	Nov 29, 2010
	CRUC-4996	Map .json files to javascript syntax highlighting	Seb Ruiz [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Nov 22, 2010	Nov 22, 2010
	CRUC-4991	Manifest implementation has large memory footprint	Anna Buttfield [Atlassian]	Anna Buttfield [Atlassian]		 Close	Fixed	Nov 21, 2010	Nov 25, 2010
	CRUC-4987	Git Indexing Issues	Conor MacNeill [Atlassian]	Conor MacNeill [Atlassian]		 Close	Fixed	Nov 18, 2010	Nov 29, 2010
	CRUC-4985	Cannot Change Password in User Settings	Tom Davies [Atlassian]	Dylan Hansen [Atlassian]		 Close	Fixed	Nov 18, 2010	Dec 02, 2010

	CRUC-4983	"You do not have permission to see all the search results." for custom search filter in Crucible	Unassigned	Gurleen Anand [Atlassian]		 Close	Fixed	Nov 18, 2010	Nov 18, 2010
	CRUC-4982	Review splitter dragging highlights text	Craig Sharkie [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Nov 18, 2010	Nov 25, 2010
	CRUC-4928	ClassCastException in ParameterSetInsertion.checkAuthors due to concurrent map modification	Anna Buttfield [Atlassian]	Anna Buttfield [Atlassian]		 Close	Fixed	Nov 14, 2010	Nov 25, 2010
	CRUC-4912	NullPointerException when Checking the Author for the BaseLine Count	Anna Buttfield [Atlassian]	Chai Ying Chan [Atlassian]		 Close	Fixed	Nov 12, 2010	Jan 03, 2011
	CRUC-4908	Log full response body of failed subtask creation response to help troubleshooting	Tim Pettersen [Atlassian]	Tim Pettersen [Atlassian]		 Close	Fixed	Nov 11, 2010	Dec 04, 2011
	CRUC-4906	Diffs link gone from watch emails	Seb Ruiz [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Nov 11, 2010	Nov 14, 2010
	CRUC-4	Diff links	Unassigned	Seb Ruiz		 Close	Fixed	Nov 11,	Sep 08,

	905	gone from watch emails	ed	[Atlassian]		d		2010	2011
	CRUC-4879	you are not prompted for a repo rescan after removing an excludes pattern	Joe Xie [atlassian]	Tim Pettersen [Atlassian]		 Close d	Fixed	Nov 10, 2010	Nov 17, 2010
	CRUC-4872	Rescan repo after adding an exclude fails	Tom Davies [Atlassian]	Tim Pettersen [Atlassian]		 Close d	Fixed	Nov 10, 2010	Dec 07, 2010
	CRUC-4842	Return the ability to add "favourites"	Tom Davies [Atlassian]	Tim Buntel [Atlassian]		 Close d	Fixed	Nov 08, 2010	Nov 25, 2010
	CRUC-4680	On Review close, don't show time if its less than one minute	Unassigned	Craig Sharkie [Atlassian]		 Close d	Fixed	Oct 25, 2010	Sep 08, 2011
	CRUC-4576	Review titles not truncated properly	Unassigned	Seb Ruiz [Atlassian]		 Close d	Fixed	Oct 14, 2010	Jan 05, 2011
	CRUC-4575	Review titles not truncated properly	Seb Ruiz [Atlassian]	Seb Ruiz [Atlassian]		 Close d	Fixed	Oct 14, 2010	Nov 24, 2010
	CRUC-4488	When the QuickSearch fails, the spinner remains active and there is no failure	Pierre-Etienne Poirot [Atlassian]	Pierre-Etienne Poirot [Atlassian]		 Close d	Fixed	Oct 11, 2010	Jan 05, 2011

		indication to the user							
	CRUC-4231	Not setting a review due date results in an unfriendly message	Tom Davies [Atlassian]	Conor MacNeill [Atlassian]		 Closed	Fixed	Sep 15, 2010	Nov 18, 2010
	CRUC-4175	JspTagException is thrown when navigating annotated view	Unassigned	Ming Giet Chong [Atlassian]		 Closed	Fixed	Sep 01, 2010	Nov 25, 2010

From 2.4.1 to 2.4.2

11 November 2010

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

JIRA Issues (64 issues)

Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3417	Show the branch changeset was committed to in watch emails	Seb Ruiz [Atlassian]	Seb Ruiz [Atlassian]		 Closed	Fixed	Oct 25, 2010	May 09, 2011
	FE-3393	Sections of the code are unreadable in side-by-side diff in IE7	Unassigned	None		 Closed	Fixed	Jun 28, 2010	May 09, 2011
	FE-3392	JavaScript error occurs when clicking a folder when you have never	Unassigned	None		 Closed	Fixed	Nov 05, 2010	May 09, 2011

		selected a branch in the branch filter							
	FE-3391	Watch email contains broken link when there is no context path	Unassigned	None		 Closed	Fixed	Jul 14, 2010	May 09, 2011
	FE-3390	Search results data sorting incorrect	Unassigned	None		 Closed	Fixed	Oct 15, 2010	May 09, 2011
	FE-3388	"IllegalStateException: Indexer is stopped" on 2.4 Upgrade	Unassigned	None		 Closed	Fixed	Oct 21, 2010	May 09, 2011
	FE-3387	ClearCase: 'Add' and 'Test Connection' buttons in admin don't do anything when both 'VOB To Include' and 'Include Pattern' and/or 'Exclude Pattern' is specified	Unassigned	None		 Closed	Fixed	Oct 25, 2010	May 09, 2011
	FE-3385	Committee name holds when you mouseover	Unassigned	None		 Closed	Fixed	Oct 13, 2010	May 09, 2011

		er it somewhere else							
	FE-3379	svn does not fully support the branch selector	Unassigned	None	↓	 Closed	Fixed	Oct 26, 2010	May 09, 2011
	FE-3378	plain text rendering of wiki text in the activity stream for changes et comment s is wrong	Unassigned	None	↓	 Closed	Fixed	Nov 02, 2010	May 09, 2011
	FE-3377	Quick file links in activity stream disappear if browser isn't wide enough	Unassigned	None	↓	 Closed	Fixed	Oct 20, 2010	May 09, 2011
	FE-3375	ClearCaseRepositoryCache.getChangeSet throws IllegalStateExceptionException when the changes et does not exist. The cache for other SCM types don't	Unassigned	None	↓	 Closed	Fixed	Oct 21, 2010	May 09, 2011
	FE-3374	[IE] More ie issues	Unassigned	None	↓	 Closed	Fixed	Oct 12, 2010	May 09, 2011
	FE-3369	Path segment	Unassigned	None	↑	 Closed	Fixed	Oct 24, 2010	May 09, 2011

		s in Include/Exclude cannot be parsed in 2.4.0							
	FE-3365	Too many objects are kept in memory during changes et indexing, resulting in an OutOfMemoryException	Unassigned	None		 Closed	Fixed	Nov 04, 2010	May 09, 2011
	FE-3364	Activity tab for all repositories is gone in 2.4	Unassigned	None		 Closed	Fixed	Oct 27, 2010	May 09, 2011
	FE-3360	GWT admin for ClearCase is slow because cleartool commands are being executed too often and too early	Michael Studman [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Sep 27, 2010	May 09, 2011
	FE-3348	ClearCase: performance of incremental indexing has decreased in 2.4.0 (Base CC only)	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Closed	Fixed	Oct 21, 2010	May 09, 2011
	FE-3346	Reload of admin	Unassigned	Michael Heemske		 Closed	Fixed	Sep 27, 2010	May 09, 2011

		page causes CPU max out		rk [Atlassian]					
	FE-3188	Can't view projects in FishEye only mode	Michael Heemske rk [Atlassian]	None		 Close	Fixed	Aug 09, 2010	May 09, 2011
	FE-3179	ClearCase: CPU usage has increased significantly between 2.3.6 and 2.4-M6	Michael Heemske rk [Atlassian]	None		 Close	Fixed	Oct 04, 2010	May 09, 2011
	FE-3173	ClearCase indexing misses revisions or indexes revisions multiple times if the server clocks of the ClearCase server and the FishEye servers are out of synch or in different timezones	Michael Heemske rk [Atlassian]	None		 Close	Fixed	Oct 26, 2010	May 09, 2011
	FE-3163	ClearCase: project exclude rules aren't being stored	Michael Heemske rk [Atlassian]	None		 Close	Fixed	Oct 24, 2010	May 09, 2011
	FE-3148	Change	Michael	None		 Close	Fixed	Jun 02,	May 09,

		the logging level of valid 'error' messages	Heemske rk [Atlassian]			d		2010	2011
	FE-3145	Excludes cause indexing to fail for Git repositories	Unassigned	None		 Closed	Fixed	Nov 01, 2010	May 09, 2011
	FE-3124	Git: Use of file content-hash to map to file revision is insufficient	Michael Heemske rk [Atlassian]	None		 Closed	Fixed	Oct 18, 2010	May 09, 2011
	FE-3119	ClearCase: specifying include or exclude rules for projects and vobs results in an invalid config.xml file, causing errors on restart	Michael Heemske rk [Atlassian]	None		 Closed	Fixed	Oct 24, 2010	May 09, 2011
	FE-3106	ClearCase: FE maxes out CPU when the connection to the ClearCase server is lost	Michael Heemske rk [Atlassian]	None		 Closed	Fixed	Sep 21, 2010	May 09, 2011
	FE-3095	ClearCase interactive process	Michael Heemske rk [Atlassian]	None		 Closed	Fixed	Aug 30, 2010	May 09, 2011

		logs timeout and forgets it has failed							
	FE-3050	ClearCase: nothing is indexed when branch excludes is non-empty and branch includes is empty	Michael Heemskeerk [Atlassian]	None		 Closed	Fixed	Oct 25, 2010	May 09, 2011
	FE-3049	ClearCase: error in storing repository configuration after creating new repo	Michael Heemskeerk [Atlassian]	None		 Closed	Fixed	Oct 04, 2010	May 09, 2011
	CRUC-5313	XSS vulnerability in FishEye/ Crucible Changeset Page Heading	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 05, 2011	Jan 11, 2011
	CRUC-5311	XSS vulnerability in FishEye Charts	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 04, 2011	Jan 11, 2011
	CRUC-5308	XSS vulnerability in Crucible's Email Reviews	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 04, 2011	Jan 11, 2011
	CRUC-4858	FishEye dialogue sends user to URL escaped URL	Michael Heemskeerk [Atlassian]	James Roper [Atlassian]		 Closed	Fixed	Sep 12, 2010	Nov 10, 2010
	CRUC-4	Upgrade	Tim	Tim		 Closed	Fixed	Nov 09,	Nov 09,

	850	to JIRA Blockers 1.6.9	Pettersen [Atlassian]	Pettersen [Atlassian]		d		2010	2010
	CRUC-4 845	Map xslt files for xml syntax highlighting	Seb Ruiz [Atlassian]	Seb Ruiz [Atlassian]	↓	 Close d	Fixed	Nov 08, 2010	Nov 08, 2010
	CRUC-4 716	PermalD field is empty in REST JSON responses	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]	↓	 Close d	Fixed	Oct 29, 2010	Nov 09, 2010
	CRUC-4 710	addGeneralComment fails when called outside a HTTP request. e.g. a plugin commit listener	Joe Xie [atlassian]	Nick Pellow [Atlassian]	↓	 Close d	Fixed	Oct 27, 2010	Nov 14, 2010
	CRUC-4 693	Snippet Page: Join Review issues	Adam Ahmed [Atlassian]	Adam Ahmed [Atlassian]	↓	 Close d	Fixed	Oct 26, 2010	Nov 10, 2010
	CRUC-4 692	Reviewers dropdown in Crucible Reviews spins forever in IE8 - cannot add reviewers	Adam Ahmed [Atlassian]	David Hergert [Windstream]	⬇	 Close d	Fixed	Oct 26, 2010	Nov 03, 2010
	CRUC-4 691	Property NotFoundException when Printing Snippet	Michael Heemskeerk [Atlassian]	Chai Ying Chan [Atlassian]	↑	 Close d	Fixed	Oct 26, 2010	Nov 10, 2010
	CRUC-4 690	No Description for	Seb Ruiz [Atlassian]	Chai Ying Chan	↓	 Close d	Fixed	Oct 26, 2010	Nov 02, 2010

		Repository LDAP Settings		[Atlassian]					
	CRUC-4656	"Edit Details", "Summarize" and "Tools" menu missing under "Reviews" tab in Opera in 2.4	Craig Sharkie [Atlassian]	Gurleen Anand [Atlassian]	↓	 Close	Fixed	Oct 21, 2010	Nov 09, 2010
	CRUC-4647	Bring the "Deleted" lozenge in line with product/corporate styles	Unassigned	Craig Sharkie [Atlassian]	↓	 Close	Fixed	Oct 20, 2010	Jun 19, 2011
	CRUC-4636	Make more of GWT admin deep linkable	Michael Studman [Atlassian]	Michael Studman [Atlassian]	↓	 Close	Fixed	Oct 19, 2010	Nov 10, 2010
	CRUC-4635	Better linking in GWT Admin	Michael Studman [Atlassian]	Michael Studman [Atlassian]	↓	 Close	Fixed	Oct 19, 2010	Nov 10, 2010
	CRUC-4614	Annotation links from file history don't preserve wayback spec	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]	↑	 Close	Fixed	Oct 18, 2010	Nov 15, 2010
	CRUC-4613	Links to comments in the same review don't work	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]	↓	 Close	Fixed	Oct 18, 2010	Nov 10, 2010
	CRUC-4612	infinite redirects to login via dialog iframes	Tom Davies [Atlassian]	Geoff Crain [Atlassian]	↑	 Close	Fixed	Jun 20, 2010	Nov 11, 2010
	CRUC-4	Clear	Tom	Chris			Fixed	Oct 15,	Nov 25,

	597	filter button goes to changeset	Davies [Atlassian]	Lam [Atlassian]		 Close		2010	2010
	CRUC-4585	Crucible Inbox: Summarize state review with incomplete reviewers shows in no filter	Tom Davies [Atlassian]	Matthew Watson		 Close	Fixed	Oct 14, 2010	Nov 25, 2010
	CRUC-4547	Join Review is duplicated	Craig Sharkie [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Oct 12, 2010	Nov 09, 2010
	CRUC-4541	Sometimes can't edit objectives	Adam Ahmed [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Oct 12, 2010	Nov 25, 2010
	CRUC-4529	Wiki links are broken	Pierre-Etienne Poirot [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Oct 12, 2010	Nov 09, 2010
	CRUC-4527	[Admin] No 'test connection' button when creating repo	Michael Studman [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Oct 12, 2010	Nov 25, 2010
	CRUC-4491	[Admin] Repo list needs pagination and filtering	Michael Studman [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Oct 11, 2010	Nov 10, 2010
	CRUC-4363	Changeset header UI annoyances	Jason Hinch [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Sep 22, 2010	Jan 05, 2011
	CRUC-4347	Set min height on body of snippet page	Adam Ahmed [Atlassian]	Jason Hinch [Atlassian]		 Close	Fixed	Sep 21, 2010	Nov 09, 2010

	CRUC-4241	selecting a tag on reports page won't work	Tom Davies [Atlassian]	Geoff Crain [Atlassian]		 Closed	Fixed	Sep 16, 2010	Nov 10, 2010
	CRUC-4131	Links in new admin ui are not real links	Michael Studman [Atlassian]	Anna Buttfield [Atlassian]		 Closed	Fixed	Aug 30, 2010	Nov 10, 2010
	CRUC-4081	[Stars] Stars in dropdowns are broken	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]		 Closed	Fixed	Aug 26, 2010	Nov 17, 2010
	CRUC-4047	[IE] Problems on Snippet Page	Adam Ahmed [Atlassian]	Anna Lyons [Atlassian]		 Closed	Fixed	Aug 22, 2010	Nov 09, 2010
	CRUC-3673	Make the rename user functionality more visible	Pierre-Etienne Poirot [Atlassian]	Andrew Myers [Atlassian]		 Closed	Fixed	Jun 23, 2010	Nov 09, 2010

From 2.4.0 to 2.4.1**27 October 2010**

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

JIRA Issues (20 issues)									
Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
	FE-3386	NullPointerException when indexing repositories on upgrade to 2.4	Unassigned	None		 Closed	Fixed	Oct 21, 2010	May 09, 2011
	FE-3367	NPE when scanning git repo trying to read commit date:	Unassigned	None		 Closed	Fixed	Aug 31, 2010	May 09, 2011

	FE-3194	Perforce case-sensitivity issues on FE 2.4 Upgrade	Conor MacNeill [Atlassian]	None		 Closed	Fixed	Oct 21, 2010	May 09, 2011
	FE-3112	Nasty error message when trying to start repo	Andrew Myers [Atlassian]	None		 Closed	Fixed	Oct 20, 2010	May 09, 2011
	FE-3109	NullPointerException on p4 describe	Conor MacNeill [Atlassian]	None		 Closed	Fixed	Oct 26, 2010	May 09, 2011
	FE-3096	Invocation of git blame fails for files that contain spaces	Conor MacNeill [Atlassian]	None		 Closed	Fixed	Oct 20, 2010	May 09, 2011
	FE-3063	Perforce branch specification doesn't work	Matthew Watson	None		 Closed	Fixed	Nov 19, 2008	May 09, 2011
	CRUC-5312	XSS vulnerability in FishEye/ Crucible Code Macro	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 05, 2011	Jan 11, 2011
	CRUC-5310	XSS vulnerability in FishEye Repository Configuration	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 04, 2011	Jan 11, 2011
	CRUC-5307	XSS vulnerability in Crucible's Linked JIRA Issue	Unassigned	Andrew Lui [Atlassian]		 Closed	Fixed	Jan 04, 2011	Jan 11, 2011
	CRUC-4695	Perforce patch reviews	Conor MacNeill [Atlassian]	Dylan Hansen [Atlassian]		 Closed	Fixed	Oct 26, 2010	Oct 26, 2010

		throw errors in 2.4.0	n]	n]					
	CRUC-4674	Permlid element can not be serialized to JSON for REST requests	Nick Pellow [Atlassian]	Nick Pellow [Atlassian]		 Close	Fixed	Oct 25, 2010	Nov 15, 2010
	CRUC-4673	fix api breakage from 2.3.x to 2.4.0	Joe Xie [atlassian]	Joe Xie [atlassian]		 Close	Fixed	Oct 24, 2010	Jan 05, 2011
	CRUC-4651	NullPointerException in SvnCache.getFileRevision	Michael Heemskeerk [Atlassian]	Michael Heemskeerk [Atlassian]		 Close	Fixed	Oct 21, 2010	Nov 15, 2010
	CRUC-4649	Whenever viewing the users tab, if the user has an email address for their login name. Crucible errors out	Jason Hinch [Atlassian]	Dylan Hansen [Atlassian]		 Close	Fixed	Oct 20, 2010	Nov 15, 2010
	CRUC-4616	SVN branch selector enabled with ajax loaded directories	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]		 Close	Fixed	Oct 18, 2010	Nov 15, 2010
	CRUC-4598	Testing a Connection in the Edit Repo Dialog. doesn't use the values from the form	Pierre-Etienne Poirot [Atlassian]	Nick Pellow [Atlassian]		 Close	Fixed	Oct 15, 2010	Nov 15, 2010
	CRUC-4	[IE7] -	Adam	Adam		 Close	Fixed	Oct 14,	Jan 05,

	591	Horizontal Scrollbar on File History page	Ahmed [Atlassian]	Ahmed [Atlassian]		d		2010	2011
	CRUC-4538	Changes et link changes server url	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]	↓	 Close d	Fixed	Oct 12, 2010	Jan 05, 2011
	CRUC-4534	Changes et hover of CVS broken	Tom Davies [Atlassian]	Seb Ruiz [Atlassian]	↓	 Close d	Fixed	Oct 12, 2010	Jan 05, 2011

FishEye 2.4 Upgrade Guide

Below are some important notes on upgrading to **FishEye 2.4**. For details of the new features and improvements in this release, please read the [FishEye 2.4 Release Notes](#).

On this page:

- [Upgrade Notes](#)
- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Notes

- **FishEye 2.4:**
 - If you are using a **Mercurial** or **Git** repository with FishEye, it will be **automatically re-indexed** when you upgrade to FishEye 2.4. This is because we have upgraded the database schema for these SCMs.
 - Lines of code data will not display for Git and Mercurial repositories in this release. This affects features like [charts](#). Please see this FAQ for more information: [Cannot View Lines of Code Information in FishEye](#).
- **FishEye 2.4.1:**
 - If you use FishEye/Crucible with **Perforce** and previously upgraded to FishEye/Crucible 2.4.0, you must reindex all of your Perforce repositories after upgrading to FishEye/Crucible 2.4.1. Note, FishEye/Crucible will not automatically reindex your Perforce repositories, you will need to start the reindex manually.
- **FishEye 2.4.2:**
 - If you are using a **ClearCase** repository with FishEye, it will be **automatically re-indexed** when you upgrade to FishEye 2.4.2. This is because we have upgraded the database schema for this SCM.

Upgrade Procedure



Upgrade a test environment first

As always, please test your upgrades in your test environment before rolling into production.

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 2.4 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 2.4 Release Notes](#)

FishEye 2.3 Release Notes

26 May 2010

 For details on minor releases since FishEye 2.3, see the [FishEye Changelog](#).

Atlassian presents FishEye 2.3

FishEye 2.3 brings support for Mercurial, the distributed version control system. It also delivers a brand-new installation wizard, for seamless initial setup.

Highlights of this release:

- [Mercurial SCM Alpha](#)
- [New 'Aggregate' functions in EyeQL query language](#)
- [Revamped Installation Process](#)
- [Numerous improvements and bug fixes](#)

Thank you for your interest in FishEye 2.3.



See the documentation on [Upgrading to this version](#).

Installing FishEye 2.3

[Download FishEye 2.3](#) now. See the documentation on [Upgrading to this version](#).

Highlights of FishEye 2.3



Mercurial SCM Alpha

FishEye 2.3 adds alpha support for Mercurial repositories. Atlassian is providing early access to this functionality.

for our customers. There are still a few kinks to be worked out, but it provides full access to FishEye.

Screenshot: A Mercurial Repository in Action

Diff	Revision	Date	Message	Lines	Author
	10835:6e876718d0cf	05 Apr	commands: leftover from d7b601f1e02c	-1	
	10834:4ab459a6c25c	05 Apr	commands: small refactoring in summary	+5 -8	
	10833:d7b601f1e02c	05 Apr	commands: retrieve tags from context object	+1 -2	
	10832:420bc8124904	05 Apr	summary: make use of output labeling The individual	+22 -9	
	10830:824310023e4a	05 Apr	commands: fix typo	+1 -1	
	10818:d14d45fae927	03 Apr	diff: make use of output labeling	+7 -5	
	10817:2096496b40ec	03 Apr	status: make use of output labeling	+7 -3	
	10816:635d601e8f21	03 Apr	qgrep: make use of output labeling	+10 -2	
	10654:153dd9139b0e	12 Mar	any is not available for python2.4	+1 -1	
	10651:5f091fc1bab7	12 Mar	style: use consistent variable names (*mod) with	+3 -3	
	10650:9ea7238ad935	12 Mar	archive: autodetect archive type by extension	+22 -4	
	10649:e13797685ee6	12 Mar	merge with stable	+16 -16	
	10644:63948e7d37f7	11 Mar	server: initialize wsqi app in command, then wrap	+10 -6	
	10635:27027bee318e	11 Mar	serve: fix port config	+6 -1	
	10632:54fb6e1fafe6	11 Mar	drop (default: 8000), non-zero default is automatically	+1 -1	

See the [documentation](#) for more details.

2

New 'Aggregate' functions in EyeQL query language

EyeQL, FishEye's sophisticated query language has become much more powerful with the addition of new aggregating functions. These allow you to quickly gather statistics such as counts, maximums, minimums and averages across the following:

- Lines added or removed,
- Authors,
- Changesets,
- Tags,
- Reviews (and more).

Screenshot: New EyeQL Functions

Search Criteria

Enter your EyeQL query: (EyeQL Help)

```
select revisions
where date >=
2010-01-31T13:00:00.00Z
and on branch default

group by changeset

return csid,
count(revisions),
sum(linesAdded),
sum(linesRemoved)
```

mercurial

Files Activity Users Reports Query

Link to this search Download results as CSV

Row	Changeset ID	count(Revision)	sum(Lines Added)	sum(Lines Removed)
1	10299:e768941f14c1	1	5	5
2	10302:00099580ca35	1	1	1
3	10313:31141fd7c9cc	1	1	1
4	10316:d117089386e2	1	3	2
5	10320:808c187fc556	1	1	1
6	10322:d9a2bc2f776b	2	4	3
7	10324:55d134ef8ab7	1	1	1
8	10325:bc72e21f9dc8	1	12	7

These data points can be calculated across your whole query result, or grouped by changeset, file or directory. When you're browsing the results, you can click in the results area to see the relevant group contents on an item.

Screenshot: FishEye Group Contents Dialog

Files Activity Users Reports Query

Link to this search Download results as CSV Displaying changesets 1 - 30 of 152: 1 2 3 4 5 6 ▶

Row	Changeset ID	Date	count(Revision)	sum(Lines Added)	sum(Lines Removed)	max(Total Lines)
1	10299:e768941f14c1	01 Feb	1	5	5	360
2	changeset e768941f14c1b0fdb428023c58ca06bd9c9da30c			1	1	253
3				1	1	253
4				3	2	3728
5	10320:808c187fc556	06 Feb	1	1	1	2169
6	10322:d9a2bc2f776b	06 Feb	2	4	3	2170

Go to: Group Contents

See the [EyeQL documentation](#) for more information.

3

Revamped Installation Process

FishEye's installation process has been given a thorough revision and a visual facelift. It's now smoother, faster and provides a better experience.

Screenshot: FishEye's New Installation Screen



See the [documentation](#) for more details.

4

Numerous improvements and bug fixes

[Visit our issue tracker](#) to see the full list of improvements and bug fixes.

FishEye 2.3 Changelog

This page contains information about the FishEye 2.3 minor releases.

 Please read the [FishEye 2.3 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 2.3.7 to 2.3.8](#)
- [From 2.3.6 to 2.3.7](#)
- [From 2.3.5 to 2.3.6](#)
- [From 2.3.4 to 2.3.5](#)
- [From 2.3.3 to 2.3.4](#)
- [From 2.3.2 to 2.3.3](#)
- [From 2.3.1 to 2.3.2](#)
- [From 2.3.0 to 2.3.1](#)

From 2.3.7 to 2.3.8

12 January 2011

This is a bug fix release that only addresses security issues. Please see the [Security Advisory](#) for more information.

The complete list of issues follows below.

<u>JIRA Issues</u> (3 issues)			
Key	Summary	Priority	Status
CRUC-5313	XSS vulnerability in FishEye/Crucible Changeset Page Heading		 Closed
CRUC-5312	XSS vulnerability in FishEye/Crucible Code Macro		 Closed
CRUC-5311	XSS vulnerability in FishEye Charts		 Closed

From 2.3.6 to 2.3.7

20 October 2010

This is a bug fix release that only addresses security issues. Please see the [Security Advisory](#) for more information.

From 2.3.5 to 2.3.6

25 August 2010

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

<u>JIRA Issues</u> (10 issues)			
Key	Summary	Priority	Status
FE-3120	ClearCase InteractiveProcess ignores error stream		 Closed
FE-3356	ClearCase autocreate view broken if view already exists (in a different location)		 Closed
FE-3251	ClearCase repository configuration in FishEye administration is overly complicated and incomplete		 Closed
FE-2719	Clearcase: use -stgloc -auto for creating views.		 Closed
FE-3384	restore will happily "restore" any zip file, regardless of content		 Closed
FE-2733	Error while indexing a perforce repository		 Closed
FE-3418	Check P4TICKETS environment variable when testing p4 connection, and on p4 login failure		 Closed

FE-3196	Test Repository option for ClearCase repositories needs to do more thorough checks	↓	 Closed
FE-2543	FishEye fails to process changes for a dynamic view	↓	 Closed
FE-2530	Provide ability for users to specify the config spec to be used for the auto created views	↓	 Closed

From 2.3.4 to 2.3.5**26 July 2010**

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

JIRA Issues (5 issues)			
Key	Summary	Priority	Status
FE-2708	XML/HTML/PDF Docs are not available for FishEye/Crucible 2.3	↑	 Closed
FE-2691	Git scanning fails with "Repository index failed due to error - class java.lang.IllegalStateException: Failed to parse specs correctly"	↑	 Closed
FE-3352	ClearCase FishEye indexer is re-indexing changes it has already indexed.	↑	 Closed
FE-3344	FishEye - ClearCase incorrectly detects a dynamic view as snapshot view when FE is not started with current view set to dynamic view	↑	 Closed
FE-3301	HG: Error getting content for a file stops entire indexing process.	↓	 Closed

From 2.3.3 to 2.3.4**13 July 2010**

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

JIRA Issues (10 issues)			
Key	Summary	Priority	Status
FE-2673	Clearcase Indexing can fail to progress in incremental indexing after initial index completed	↑	👤 Closed
FE-2640	SAL API packages not properly exported	↑	👤 Closed
FE-2676	Search results preview dropdown not showing up	↑	👤 Closed
FE-2649	Merge studio crowd classloader fix from 1.6.x	↑	👤 Closed
FE-3419	Support a url parameter to specify the REST return format	↑	👤 Closed
FE-3368	Lucene indexer fails on null values	↓	👤 Closed
FE-3366	Atlaseye returning empty RSS feeds	↓	👤 Closed
FE-3363	Error when sending changeset notifications to users with a batch-email preference	↓	👤 Closed
FE-3373	incorrect line count changes	↓	👤 Closed
FE-2317	Update changeset-mail-*.ftl files	↓	👤 Closed

From 2.3.2 to 2.3.3**16th June 2010**

This is a bug fix release that addresses security issues. Please see the [Security Advisory](#) for more information.

The complete list of issues follows below.

JIRA Issues (1 issues)			
Key	Summary	Priority	Status
FE-2633	use safe parameter interceptor	↓	👤 Closed

From 2.3.1 to 2.3.2**3rd June 2010**

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

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

From 2.3.0 to 2.3.1

27th May 2010

This is a bug fix release, addressing an issue that occurs when upgrading FishEye with an expired license.

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

FishEye 2.3 Upgrade Guide

Below are some important notes on upgrading to **FishEye 2.3**. For details of the new features and improvements in this release, please read the [FishEye 2.3 Release Notes](#) and [FishEye 2.3 Changelog](#).

On this page:

- [Upgrade Notes](#)
- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Notes

- If you use FishEye with ClearCase and you upgrade to FishEye/Crucible 2.3.4 or later from a version prior to 2.3.4, FishEye/Crucible will reindex all ClearCase repositories.

Upgrade Procedure



Upgrade a test environment first

As always, please test your upgrades in your test environment before rolling into production.

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 2.3 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 2.3 Release Notes](#)

[FishEye 2.3 Changelog](#)

Older release notes

Release notes for older versions of FishEye are kept as children of this page:

- [FishEye 2.2 Release Notes](#)
- [FishEye 2.1 Release Notes](#)
- [FishEye 2.0 Release Notes](#)
- [FishEye 2.0 Beta Release Notes](#)
- [FishEye 1.6 Release Notes](#)
- [FishEye 1.5 Release Notes](#)
- [FishEye 1.4 Release Notes](#)
- [FishEye 1.3 Release Notes](#)

FishEye 2.2 Release Notes

 FishEye 2.10 has now been released. Read the [Release Notes](#).

18 February 2010

 For details on minor releases since FishEye 2.2, see the [FishEye Changelog](#).

Atlassian presents FishEye 2.2

FishEye 2.2 is focused on improved user experience, adding a powerful side-by-side diff visualisation, along with improved quick-navigation, quick-search and performance enhancements.

Highlights of this release:

- [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](#)

Thank you for your interest in FishEye 2.2.



See the documentation on [Upgrading to this version](#).

Installing FishEye 2.2

[Download FishEye 2.2](#) now. See the documentation on [Upgrading to this version](#).

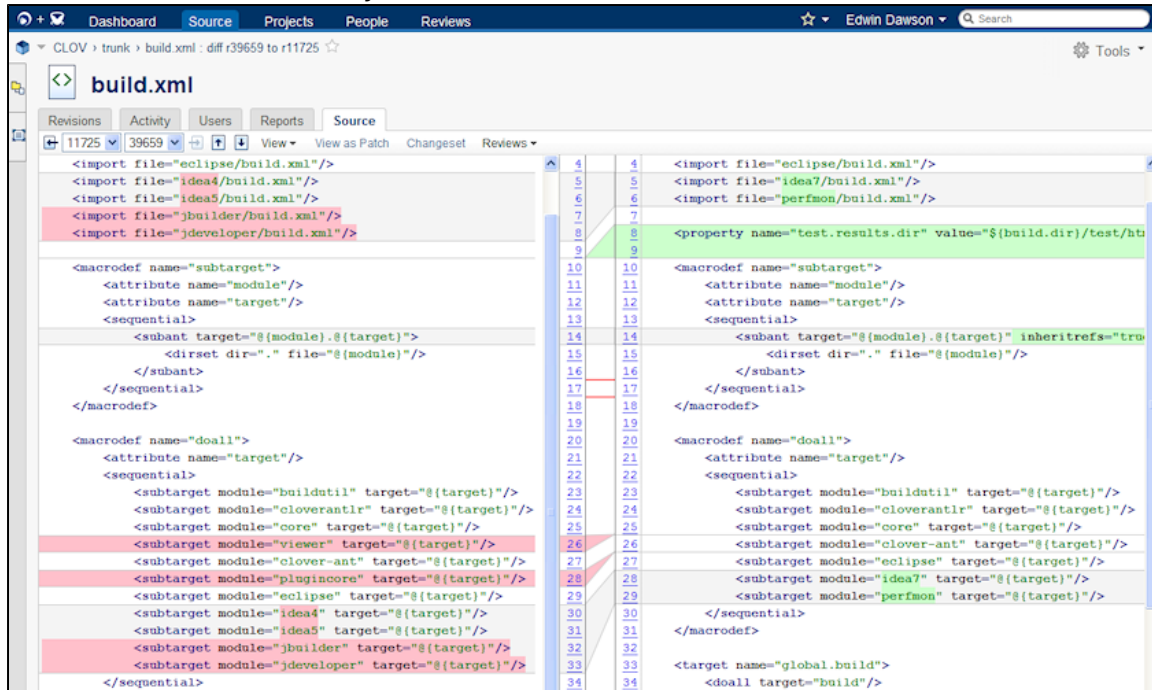
Highlights of FishEye 2.2

1

Enhanced Side-by-Side Diff View Mode

FishEye 2.1 adds improvements for the side by side diff view mode, which displays a powerful visualisation of how code has changed between revisions. Two independently scrolling windows allow you to see new and old content clearly delineated, with explicit dynamic markings that anchor each change to its exact location in the left and right windows.

Screenshot: Enhanced 'Side-by-Side Diff' Mode



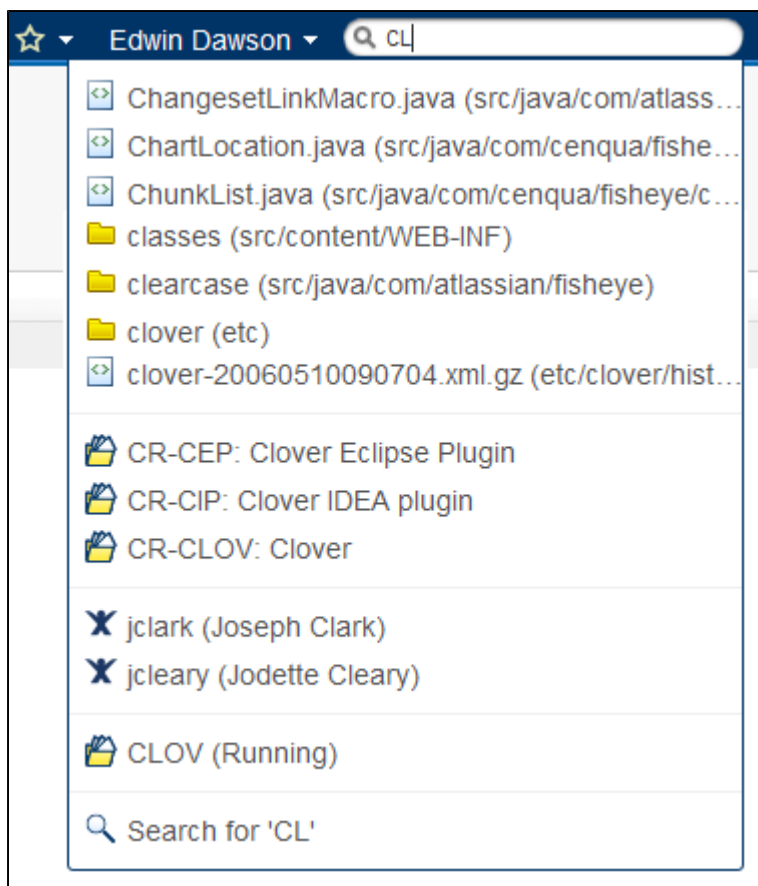
See the [documentation](#) for more details.

2

Improved Quick Navigation

FishEye's Quick Nav has been re-written to display results faster, and to return more relevant results. It still supports powerful IDE-like functions such as camelcase detection, multiple directory selection and cross-repository results. All these items appear in a handy preview window, when you type text into the search box.

Screenshot: Improved Quick Nav



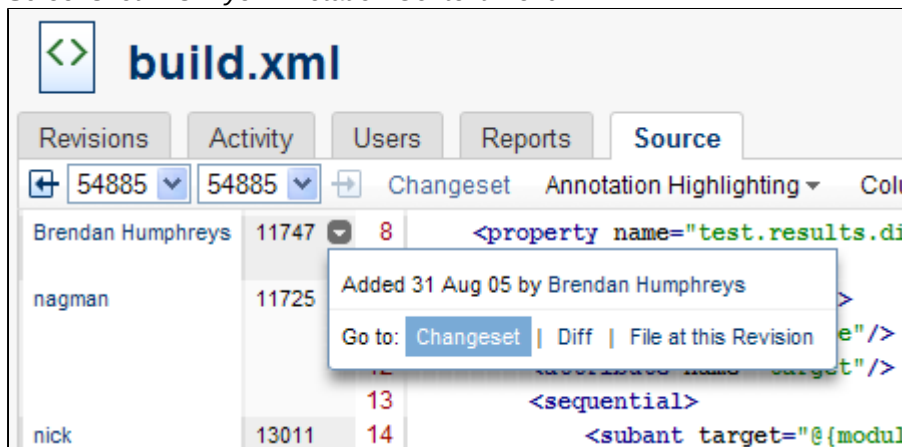
See the [documentation](#) for more details.

3

Annotation Context Menu

In FishEye, you can now drill down into the history of any line of code in a file. By hovering your mouse cursor next to the revision number in the left column, you can open the annotation context menu. From there, you can view the changeset where this file was changed, the diff of this line with the last change, or the full file from the time when this line changed. This speeds investigation of how certain lines have changed, when, and by whom.

Screenshot: FishEye Annotation Context Menu



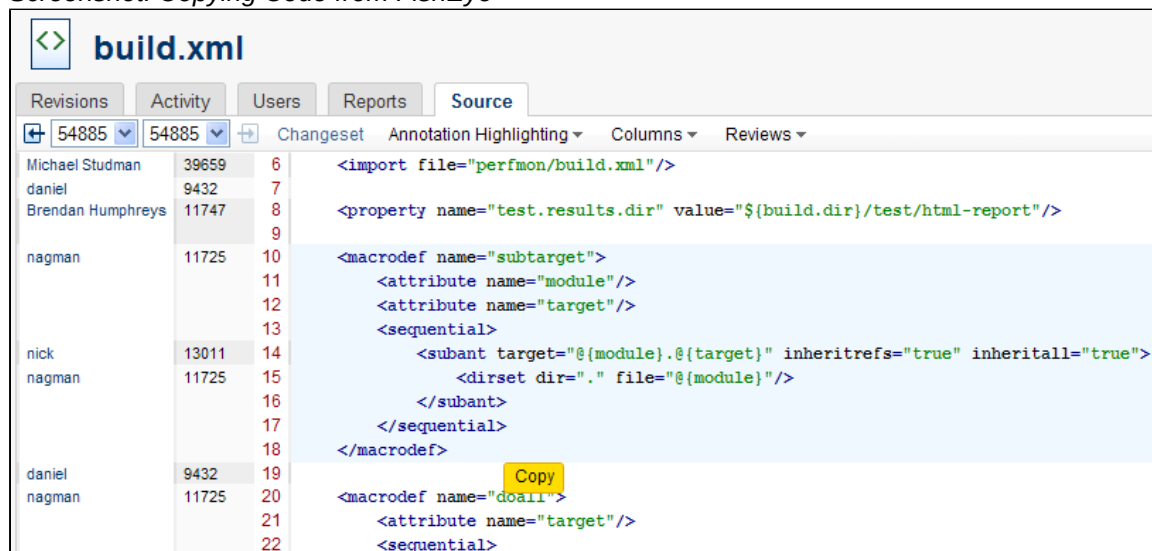
See the [documentation](#) for more details.

4

Code Copying

FishEye now lets you neatly copy and paste multiple lines of code directly from FishEye to the system clipboard by dragging in the FishEye window to select. Previously, some other information would also be copied from the browser display. Now, only the code itself is copied for your convenience.

Screenshot: Copying Code from FishEye



See the [documentation](#) for more details.

5

ClearCase and Git Support Now Final

FishEye's support for IBM ClearCase and Git has been in public beta for a number of months, but with the release of FishEye 2.2 it's now officially final. Having benefited from a good dose of spit and polish by our developers, these features are now fully supported and ready for production use.

See the [FishEye ClearCase documentation](#) and the [FishEye Git documentation](#) for more details.

6

Numerous improvements and bug fixes

[Visit our issue tracker](#) to see the full list of improvements and bug fixes.

FishEye 2.2 Changelog

This page contains information about the FishEye 2.2 minor releases.



Please read the [FishEye 2.2 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 2.2.1 to 2.2.3](#)
- [From 2.2.0 to 2.2.1](#)

From 2.2.1 to 2.2.3**4th May 2010**

This release addresses critical security issues. Atlassian strongly recommends that you upgrade to the latest version. Please read the [Security Advisory](#) for details.

This is a security and bug fix release, addressing the issues listed below. See the [Security Advisory](#) for specific information about the fixes and patches which are available.

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

 FishEye version 2.2.2 was an internal release.

From 2.2.0 to 2.2.1**9th March 2010**

This is a bugfix release, addressing the following issues:

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

FishEye 2.2 Upgrade Guide

Below are some important notes on upgrading to **FishEye 2.2**. For details of the new features and improvements in this release, please read the [FishEye 2.2 Release Notes](#) and [FishEye 2.2 Changelog](#).

On this page:

- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Procedure**Upgrade a test environment first**

As always, please test your upgrades in your test environment before rolling into production.

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 2.2 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 2.2 Release Notes](#)

[FishEye 2.2 Changelog](#)

FishEye 2.1 Release Notes

 FishEye 2.10 has now been released. Read the [Release Notes](#).

12 November 2009

 For details on minor releases since FishEye 2.1, see the [FishEye Changelog](#).

Atlassian presents FishEye 2.1

FishEye 2.1 adds new search options, streamlined JIRA integration and runs significantly faster.

Highlights of this release:

- [Wiki Markup in Commit Messages](#)
- [Streamlined JIRA Integration](#)
- [FishEye Admin API](#)
- [History Page Performance Increases](#)
- [ClearCase Support Now in Beta](#)
- [Numerous improvements and bug fixes](#)

Thank you for your interest in FishEye 2.1.



See the documentation on [Upgrading to this version](#).

Installing FishEye 2.1

You can now download FishEye 2.1 from [here](#). See the documentation on [Upgrading to this version](#).

Highlights of FishEye 2.1

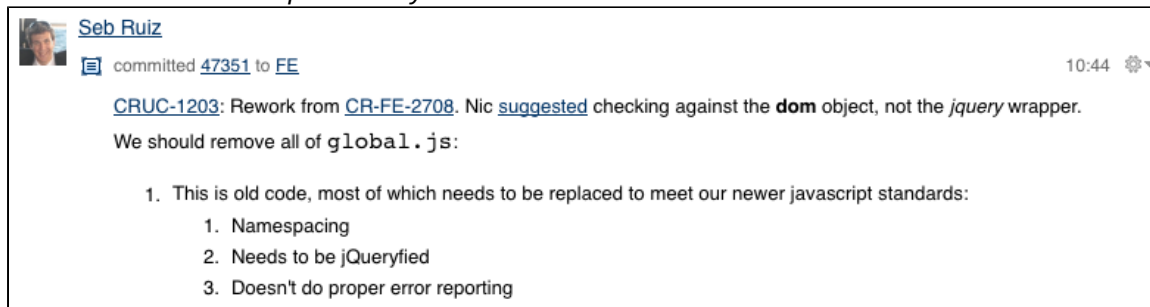


Wiki Markup in Commit Messages

You can now use Atlassian Wiki Markup in your [commit messages](#) — **bold text**, *italics*, code, links, images and

much more — FishEye will render it. [Crucible](#) now has Wiki Markup Rendering as well.

Screenshot: Wiki Markup in Activity Stream

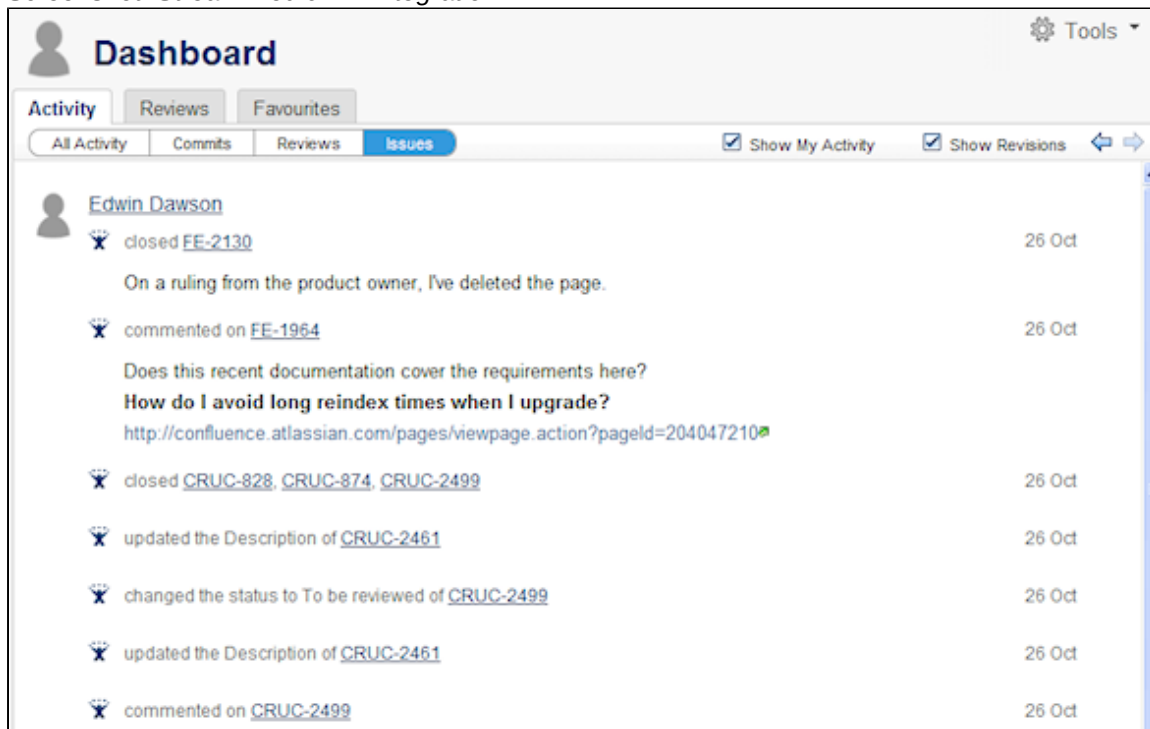


2

Streamlined JIRA Integration

The JIRA hover feature has been improved, now resolving JIRA user names as they appear in FishEye. The integration has also been updated to support JIRA 4.

Screenshot: Streamlined JIRA Integration



3

FishEye Admin API

The FishEye administration API has been improved. It is now possible to let plugins control repositories, for example adding new repositories, starting, stopping and so on. This work will be followed by a REST layer to also make this available to remote clients and scripts. We also added [Atlassian Plugin SDK](#) support to FishEye and Crucible, simplifying build management for plugin developers. The developer documentation for FishEye and Crucible has been co-located into a [new documentation space](#) as well.

Screenshot: New Development Documentation Space



The History page's now uses "lazy loading" - getting only those features, such as revisions and commit log messages, that can initially be seen by the viewer and then building out the rest of the page contents as they're required. New pagination options were added giving access to the first and last pages of information, in addition to the previous and next. This makes things work a lot faster.

The screenshot shows the IntelliJ IDEA interface with the 'build.xml' file open. The 'Revisions' tab is active, showing a list of revisions. The selected revision is 47064, dated 03 Nov, with the message 'NONE: checkstyle fixes, setup-idea overwrites existing files in .idea'. The 'Diff' view shows the changes for this revision, highlighting the 'build.xml' file. The 'Properties' tab is also visible, showing details for revision 47064, including the author 'Tom Dawes' and the date '03 November 2009 22:54 (25 hours)'.

Revision	Date	Message	Delta
47064	03 Nov	NONE: checkstyle fixes, setup-idea overwrites existing files in .idea	+2 -2
47055	03 Nov	NONE: force the ant restore to use the specified jdbc jar file by putting it as the first entry in the classpath	+4 -4
46973	02 Nov	NONE: minor cleanup	+10 -9
46807	29 Oct	NONE: double ensure that we don't deploy snapshots on test failure	+2 -1
46540	27 Oct	Revert 46538 'Add clover-tasks back to run tests'	+1 -1
46538	27 Oct	Add clover-tasks back to run tests	+1 -1
46052	19 Oct	CRUC-2379: improve AMPS sample data, take 2	-1
46036	19 Oct	CRUC-2379: improve AMPS sample data	+4 -2
46026	19 Oct	progress towards amps.dist & dist2amps ant tasks	+49 -16
46025	19 Oct	NONE: add ability to deploy snapshots on CI builds	+12 -6
45973	19 Oct	NONE: including crucible db in amps dist	+1
45945	18 Oct	NONE: Added amps-fecru-data zip deploy.	+5
45808	15 Oct	NONE: Fix doc.dac build target that broke after the api directory got moved.	+5 -5
45786	14 Oct	NONE: definepoms can't depend on copypoms for prod builds	+2 -2
45612	13 Oct	FE-2180: Update to modz 0.8	+1 -1

Properties:

- Author: Tom Dawes
- Revision: 47064
- Changeset: 47064
- Date: 03 November 2009 22:54 (25 hours)
- Properties: cenqua = hello, svn:executable = *, svn:keywords = Author Date Id Revision

5

Created by Atlassian in 2012. Licensed under a [Creative Commons Attribution 2.5 Australia License](#).

FishEye's support for IBM ClearCase is now in beta, following extensive feedback from our brave alpha testers



Numerous improvements and bug fixes

[Visit our issue tracker](#) to see the full list of improvements and bug fixes.

FishEye 2.1 Changelog

This page contains information about the FishEye 2.1 minor releases.

 Please read the [FishEye 2.1 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 2.1.3 to 2.1.4](#)
- [From 2.1.2 to 2.1.3](#)
- [From 2.1.1 to 2.1.2](#)
- [From 2.1.0 to 2.1.1](#)

From 2.1.3 to 2.1.4

27th January 2010

This is a bugfix release, addressing the following issues:

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

From 2.1.2 to 2.1.3

13th January 2010

This is a bugfix release, addressing a number of ClearCase bugs and performance problems.

The following issues are addressed by this release:

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

From 2.1.1 to 2.1.2

19th November 2009

This is a bugfix release.

The following issues are addressed by this release:

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

From 2.1.0 to 2.1.1

17th November 2009

This is a bugfix release.

The following issues are addressed by this release:

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

FishEye 2.1 Upgrade Guide

Below are some important notes on upgrading to **FishEye 2.1**. For details of the new features and improvements in this release, please read the [FishEye 2.1 Release Notes](#) and [FishEye 2.1 Changelog](#).

On this page:

- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Procedure

Upgrade a test environment first

As always, please test your upgrades in your test environment before rolling into production.

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 2.1 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 2.1 Release Notes](#)

[FishEye 2.1 Changelog](#)

FishEye 2.0 Release Notes

 FishEye 2.10 has now been released. Read the [Release Notes](#).

30 June 2009

 For details on minor releases since FishEye 2.0, see the [FishEye Changelog](#).

Atlassian presents FishEye 2.0

FishEye 2.0 adds enhanced JIRA integration and a brand new user interface.

Highlights of this release:

- [Activity Streams](#)
- [People](#)
- [Favourites, bookmarks & saved search](#)
- [Enhanced JIRA Integration](#)
- [Projects](#)
- [New User Interface](#)
- [Quick Navigation & Improved Quick Search](#)
- [Git Beta](#)
- [Numerous improvements and bug fixes](#)

Thank you for your interest in FishEye 2.0.



See the documentation on [Upgrading to this version](#).

Installing FishEye 2.0

You can now download the FishEye 2.0 from [here](#). See the documentation on [Upgrading to this version](#).

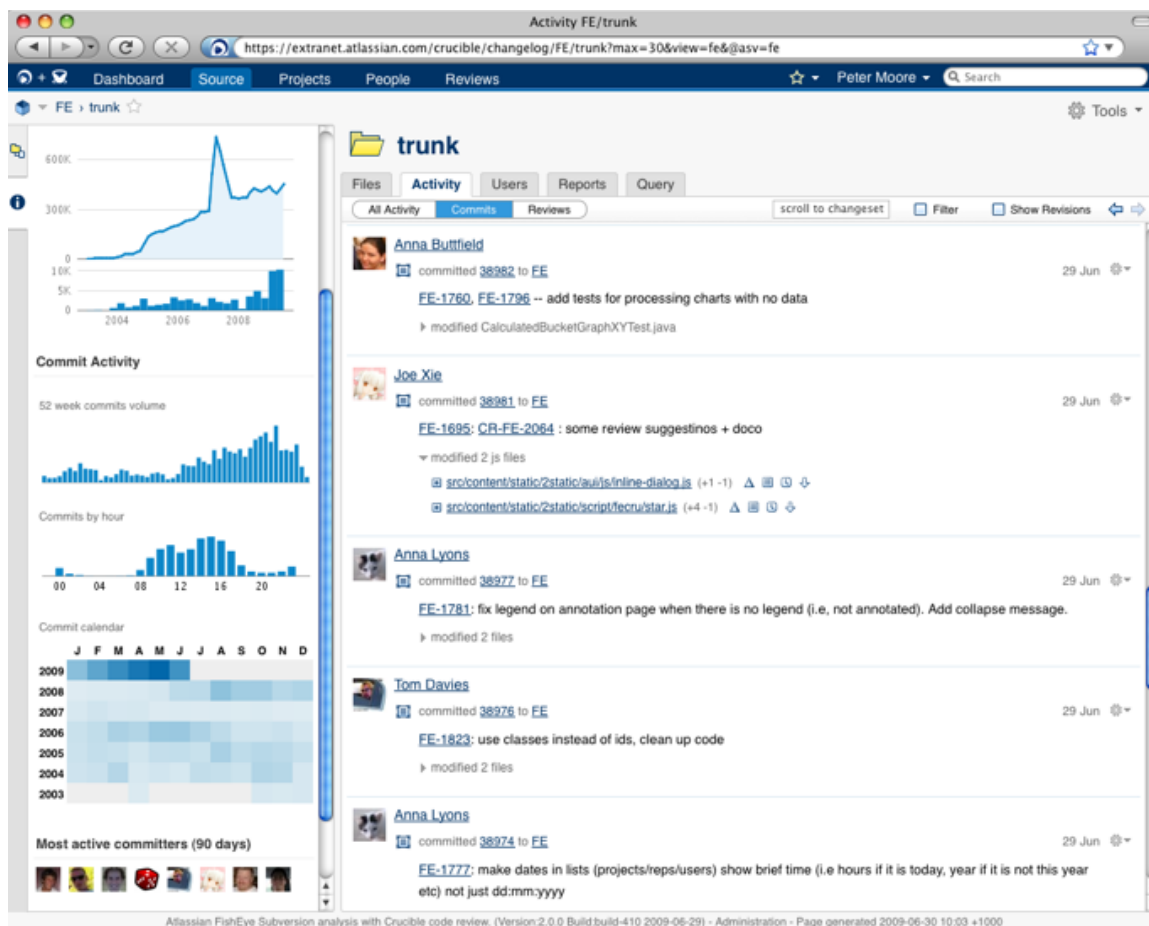
Highlights of FishEye 2.0



Activity Streams

FishEye 2 provides an activity stream showing commits, JIRA issues, and Crucible review activities. Activity streams are available across all repositories, from any directory, person, project and even individual files. Your home page provides a custom feed generated from the people, projects, and source you have selected as favorites.

Screenshot: Activity for a directory

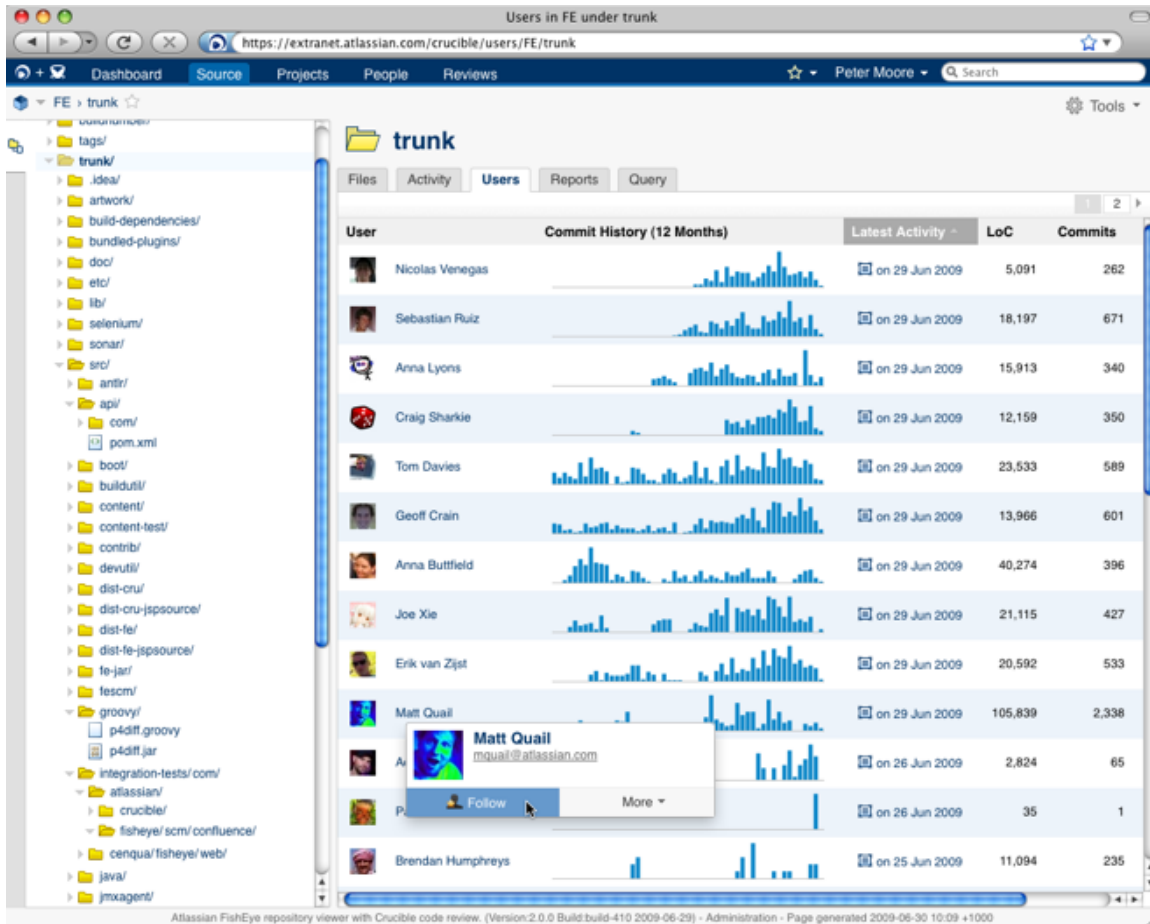


2

People

Each user and committer in FishEye 2 has a page showing their statistics and activity. You can also see activity grouped by user from all repositories down to an individual file. You can sort by latest activity, commits, line count, even reviews using Crucible.

Screenshot: Users on trunk sorted by activity

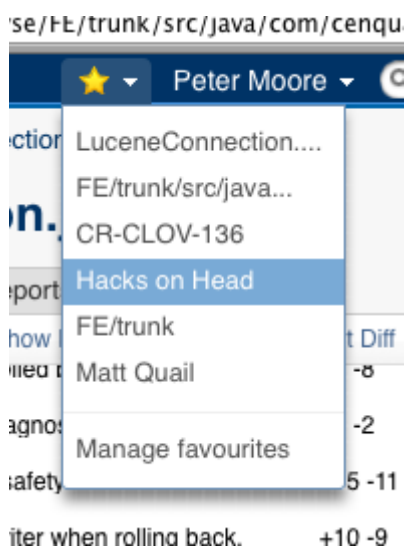


3

Favourites, bookmarks & saved search

Clicking on the star icon on almost any artifact in FishEye adds that artifact to the your bookmark menu, and adds any downstream activity for that artifact to your personal dashboard. It also makes them searchable in quicknav. Charts and searches can be saved and given custom names for simpler reuse.

Screenshot: New Favourite Menu

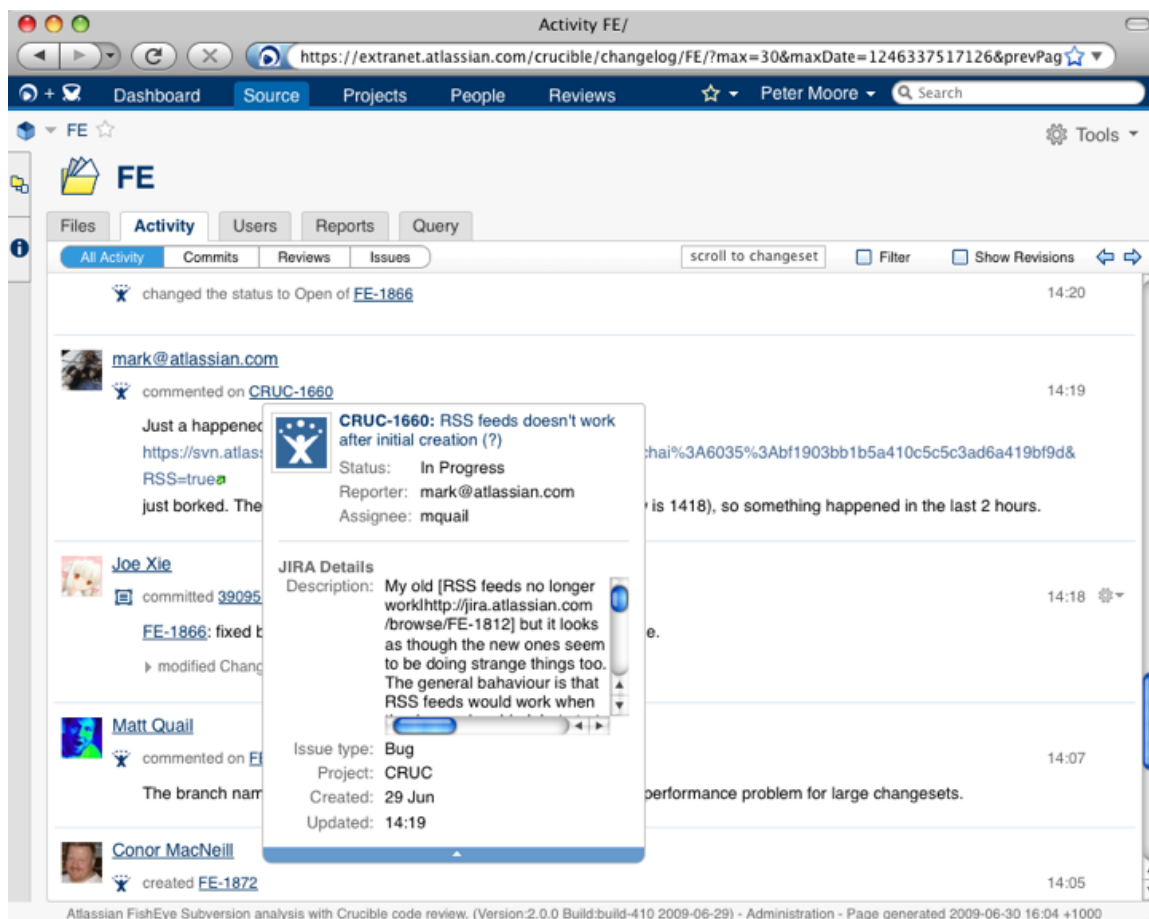


4

Enhanced JIRA Integration

Multiple JIRA servers can be configured for your FishEye instance. Projects and repositories can be mapped to one, several, or all JIRA projects. FishEye retrieves summary issue information when you mouseover a JIRA key anywhere in FishEye. Optionally you can include issue activity in your FishEye activity streams as well.

Screenshot: Enhanced JIRA Integration

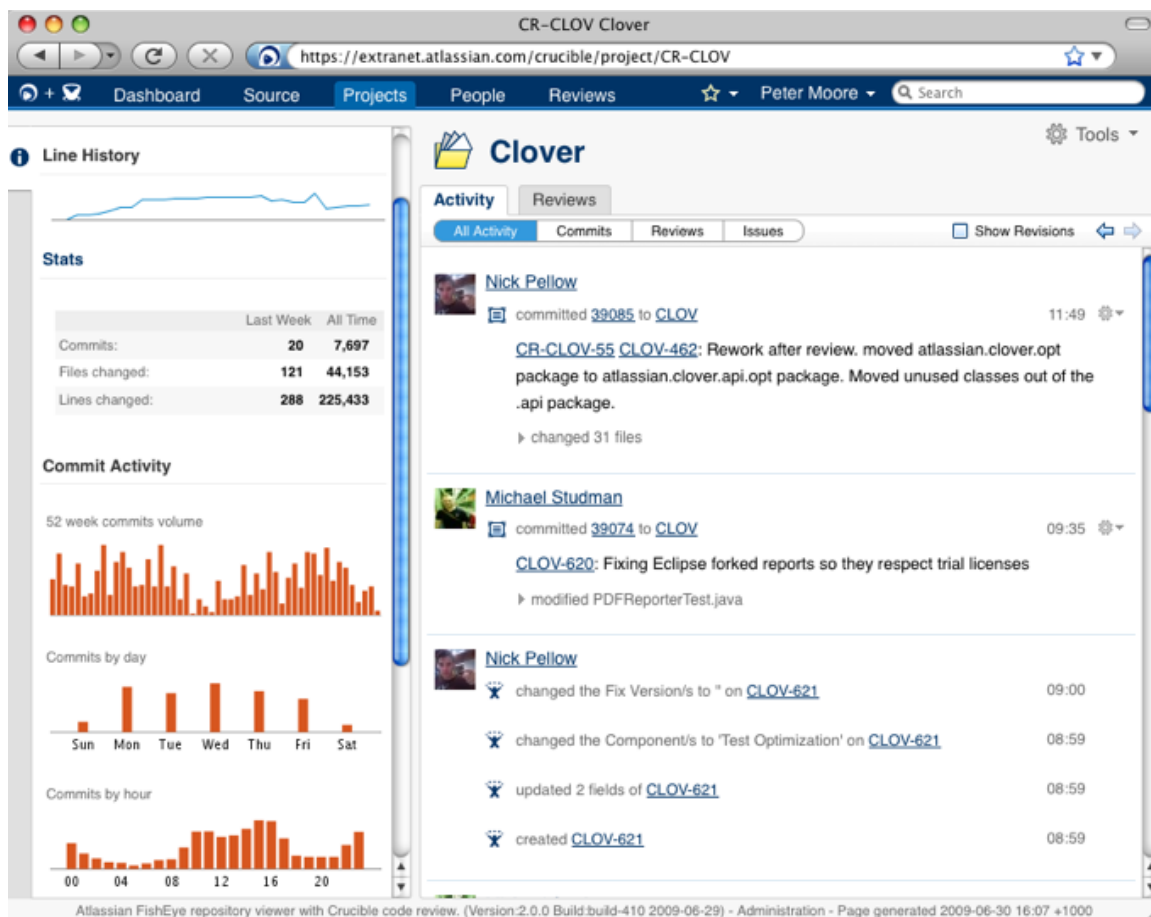


5

Projects

Multiple repositories or subsets of repositories can be combined into a project to provide a more focussed reporting and activity streams for teams that have source spread across repositories, or even different source control systems.

Screenshot: A Project page

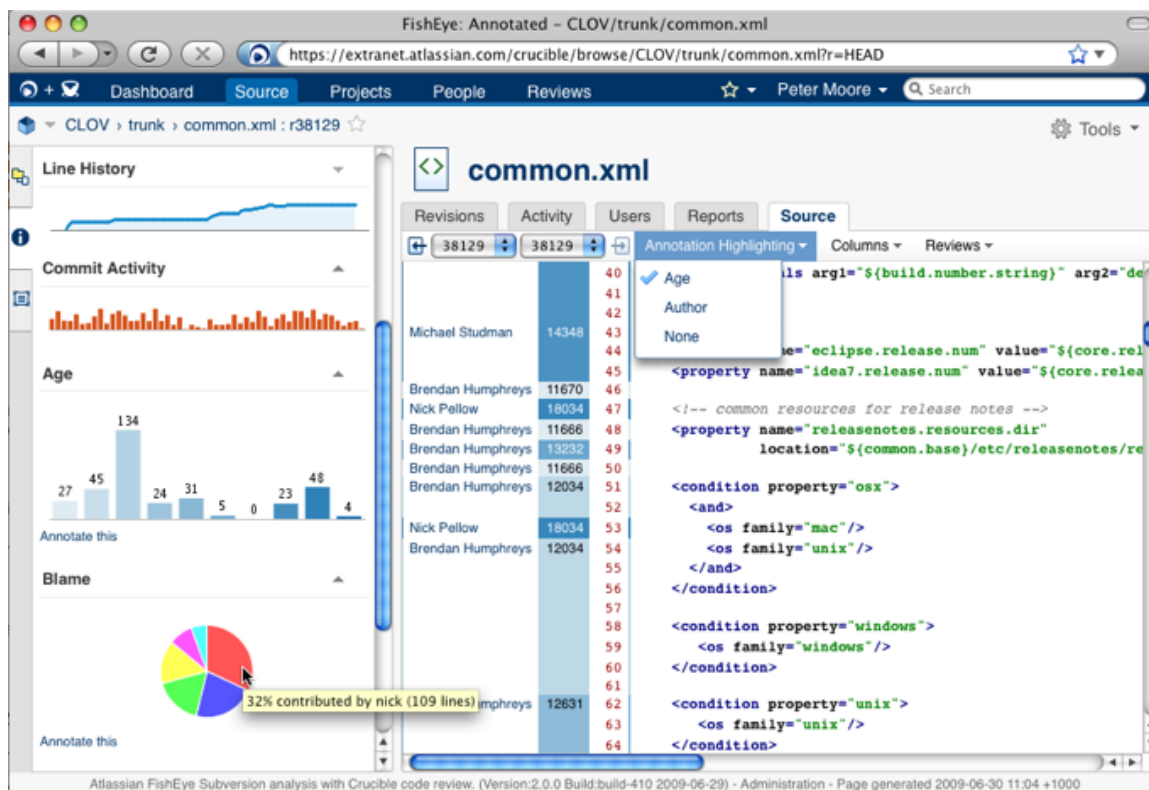


6

New User Interface

The FishEye team has completely revamped the user interface. FishEye now uses an intuitive three pane view with a rewritten file explorer and collapsible chart and information panes. There is better access to menus, user hovers, and lots more.

Screenshot: New annotation

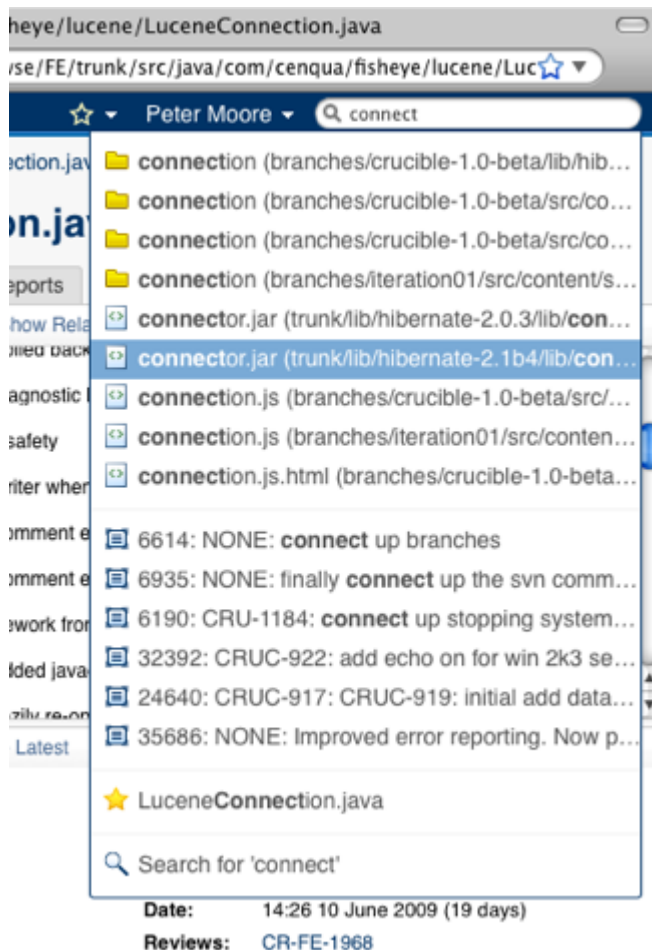


7

Quick Navigation & Improved Quick Search

Typing into the search box on the top right of every page now gives you quick access to directories, files, changesets, even your personal favourites. Quick search now works across repositories and is significantly faster.

Screenshot: Quicknav

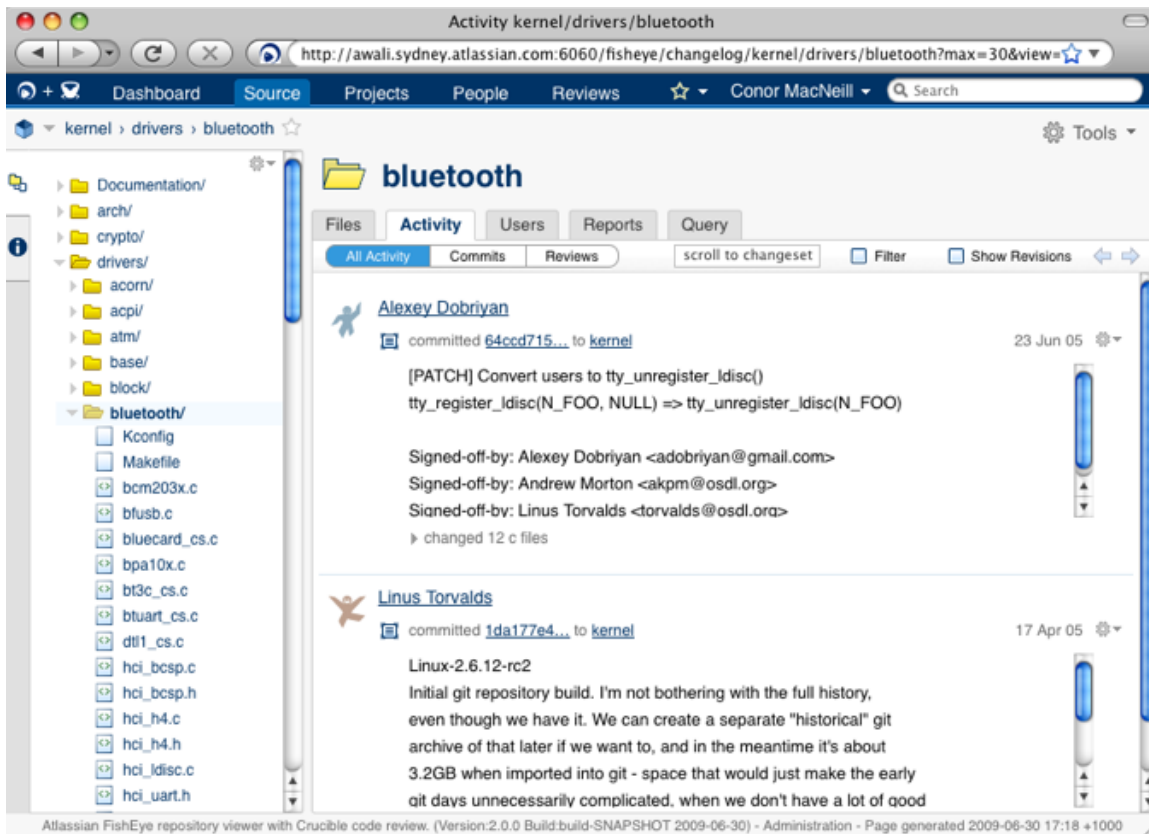


8

Git Beta

FishEye 2 adds beta support for Git repositories. There are still a few kinks to be worked out, but it provides full access to FishEye's functionality.

Screenshot: The linux kernel's Git repository in FishEye



9

Numerous improvements and bug fixes

[Visit our issue tracker](#) to see the full list of improvements and bug fixes.

FishEye 2.0 Changelog

This page contains information about the FishEye 2.0 minor releases.

 Please read the [FishEye 2.0 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 2.0.5 to 2.0.6](#)
- [From 2.0.4 to 2.0.5](#)
- [From 2.0.3 to 2.0.4](#)
- [From 2.0.2 to 2.0.3](#)
- [From 2.0.1 to 2.0.2](#)
- [From 2.0 to 2.0.1](#)
- [From 2.0 Beta3 to 2.0](#)
- [From 2.0 Beta2 to 2.0 Beta3](#)
- [From 1.6.6 to 2.0 Beta](#)

From 2.0.5 to 2.0.6

8th October 2009

This is a bugfix release.

This release fixes a bug that affected Crucible-only installations.

*From 2.0.4 to 2.0.5***6th October 2009**

This is a bugfix and improvement release.

- **Support for new licenses:** [Starter licenses](#) are special offer, low-cost licenses that allow small teams to make use of Atlassian products.

The following issues are addressed by this release:

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

*From 2.0.3 to 2.0.4***8th September 2009**

This is a bugfix and improvement release. This release addresses the following issues:

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

*From 2.0.2 to 2.0.3***18th August 2009**

This is a bugfix release which includes the following issues:

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

*From 2.0.1 to 2.0.2***24th July 2009**

This is a bugfix release which includes the following issues:

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

*From 2.0 to 2.0.1***14th July 2009**

This is a bugfix release which includes the following issues:

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

*From 2.0 Beta3 to 2.0***30th June 2009**

Full list of issues in this release:

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

From 2.0 Beta2 to 2.0 Beta3

5th June 2009

Full list of issues in this release:

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

From 1.6.6 to 2.0 Beta

Full list of issues in this release:

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

FishEye 2.0 Upgrade Guide

Below are some important notes on upgrading to **FishEye 2.0**. For details of the new features and improvements in this release, please read the [FishEye 2.0 Release Notes](#) and [FishEye 2.0 Changelog](#).

On this page:

- [Upgrade Notes](#)
 - [Supported Browsers](#)
 - [MySQL Enterprise Server Database Issues](#)
 - [Problems with FishEye Freezing Unexpectedly](#)
 - [Other Notes](#)
- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Notes

Supported Browsers

FishEye 2.0 now supports the following browsers:

- Safari 3 (or later)
- FireFox 3 (or later)
- Internet Explorer 7 (or later)

 Internet Explorer 6 is no longer supported.

MySQL Enterprise Server Database Issues

When migrating your database to MySQL Enterprise Server, you may encounter [problems with very long comments in MySQL](#).

Problems with FishEye Freezing Unexpectedly

A known issue may cause FishEye 2.0 to [freeze unexpectedly](#).

Other Notes

- Please note that upgrading from 1.6.x will require an (automatic) full reindex.
- Please note that, unless "[Store Diff Info](#)" is disabled, FishEye's disk space requirements are much greater than in previous versions. On an existing FishEye instance, this will first become apparent when the instance is reindexed. Please also see the disk space recommendations on the [System Requirements](#) page.

Upgrade Procedure

Upgrade a test environment first

As always, please test your upgrades in your test environment before rolling into production.

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 2.0 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 2.0 Release Notes](#)

[FishEye 2.0 Changelog](#)

FishEye 2.0 Beta Release Notes

FishEye 2.0 Beta is a public development release leading up to **FishEye 2.0**. For all production use of FishEye, please use [the latest complete release](#).

 This page refers to an updated version of the Beta (Beta 3). We strongly recommend all beta users [upgrade to this release](#).

 Do not use in production.
Beta releases should not be used in production environments.

 Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing FishEye development process. As such:
 - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
 - Features in development releases may be incomplete, may change or be removed before the next full release.
 - Firefox 3 and Safari are the only browsers supported.

5 June 2009

Atlassian presents FishEye 2.0 Beta

FishEye 2.0 adds enhanced JIRA integration and a brand new user interface.

Highlights of this release:

- [Enhanced JIRA Integration](#)
- [New User Interface](#)
- [People View](#)
- [Plus numerous improvements and bug fixes](#)

Thank you for your interest in FishEye 2.0 Beta.



See the documentation on [Upgrading to this version](#).

Installing FishEye 2.0 Beta

You can now download the FishEye 2.0 Beta from [here](#). See the documentation on [Upgrading to this version](#).

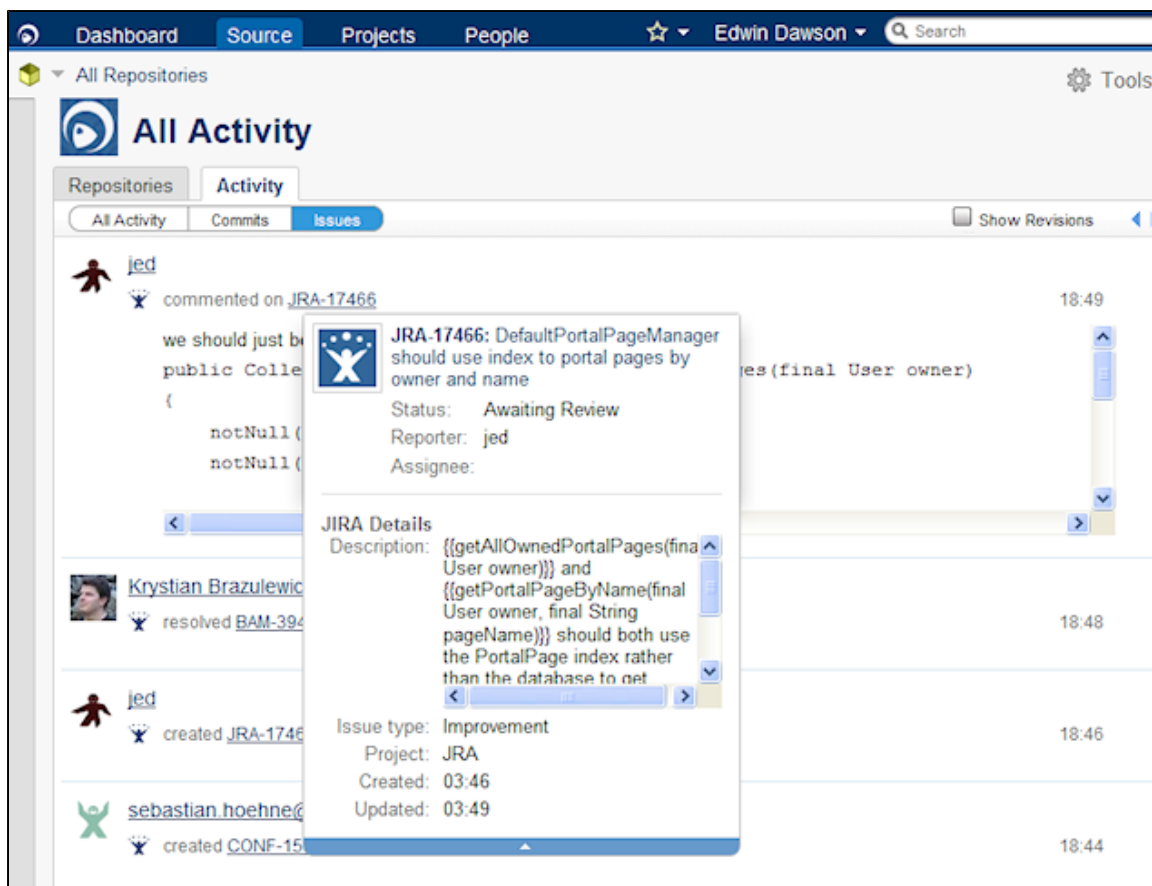
Highlights of FishEye 2.0 Beta



Enhanced JIRA Integration

FishEye now has better JIRA integration, allowing you to see regular JIRA updates on your FishEye dashboard as well as click on issue names to visit the JIRA instance they belong to. See [instructions for JIRA configuration](#)

Screenshot: Enhanced JIRA Integration



2

New User Interface

Taking on board wide-ranging feedback from customers, the FishEye team has completely revamped the user interface of the product, adding more views on your work and allowing you to access controls from multiple locations, allowing for different work styles.

Screenshot: New User Interface

The screenshot shows the FishEye web interface for a project named 'CLOV'. The 'Source' tab is selected, displaying the file 'build.xml'. The left sidebar shows a file tree with folders like 'testlib/' and 'tutorial/', and various files including 'build.xml'. The main area shows a table of revisions for 'build.xml'.

Diff	Revision	Date	Message	Lines	Author	Branch
☐	23874	26 Nov 08	NONE: show junit output in the tutorial	+1 -1	Nick Pellow	trunk
☐	22988	04 Nov 08	NONE: minor changes to tutorial build files.	+33 -29	Nick Pellow	trunk
☐	13228	29 Aug 07	CCD-479: make tutorial work on jdk1.4	+2 -2	Brendan Humphreys	trunk
☐	12717	25 Feb 07	CCD-379: Creation of Maven Tutorials, and amendments to structure of tutorial	+2 -2	andrea	trunk
☐	12685	08 Feb 07	CCD-379: Reworked tutorial to acknowledge changes made to build and build_completed.xml. Also made change per review recommendation to naming conventions.	+3 -3	andrea	trunk
☐	12683	07 Feb 07	CCD-379: Reworked more changes to the build and build_completed.xml after review	+8 -9	andrea	trunk
☐	12680	06 Feb 07	CCD-379: re-org of tutorial	+29 -16	andrea	trunk
☐	12631	14 Jan 07	NONE: merging Clover2 back to trunk	+9 -5	Brendan Humphreys	trunk
☐	11666	17 Aug 05	upgrade to proguard 3.3.2, obfuscate clover for Ant, more re-arranging	+31	Brendan Humphreys	trunk
☐	8472	20 May 03	7338: Removed env and env.ant_home refs	+2 -3	kim	trunk

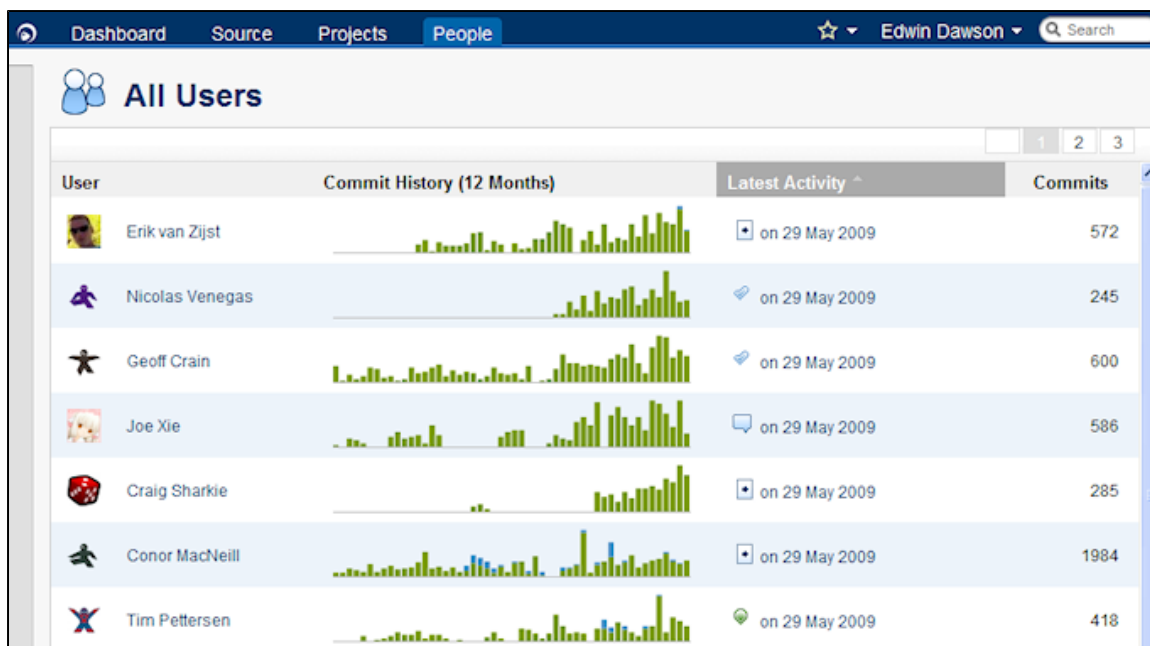
Below the table, the 'Annotated' view is selected, showing the content of revision 23874: 'NONE: show junit output in the tutorial'. The author is Nick Pellow. The revision number is 23874, and the changeset is also 23874. The date is 17:42 26 November 2008 (6 months). The branch is 'clover-2.4.x clover-2.5.x clover-distributed fedex11-mstudman tag:clover-ant-2.4.2'. The copied to path is 'branches/clover-2.4.x/trunk/tutorial/build.xml 31720'. The tags are 'clover-ant-2.4.2 idea7-2.4.2-v20081224113000 eclipse-2.4.2.v20090112202100 clover-ant-2.5.0 idea7-2.5.0.v20090511090000 eclipse-2.5.0.v20090511090000'. The properties are 'svn:eol-style = native' and 'svn:keywords = Author Date Id Revision'.

3

People View

You can now view detailed charts and activity statistics people who use your FishEye instance. You can compare number of commits charted over time and other activity in detail.

Screenshot: People View



4

Plus numerous improvements and bug fixes

Alpha support for [Git](#) is activated in the beta but not complete. For more information, see the [Git alpha documentation](#).

[Visit our issue tracker](#) to see the full list of improvements and bug fixes between Beta 2 and Beta 3. We strongly recommend all beta users [upgrade to the latest beta release](#).

See the [Beta Reviewer's Guide](#) for a list of known issues and guidance on the beta experience.

Upgrading to the FishEye 2.0 Beta

FishEye 2.0 Beta is a public development release leading up to **FishEye 2.0**. For all production use and testing of FishEye, please use [the latest official release](#).

Do not use in production.
Beta releases should not be used in production environments.

Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing FishEye development process. As such:
 - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
 - Features in development releases may be incomplete, or may change or be removed before the next full release.
 - There will be an upgrade path from the 2.0 Beta to the final release.

This page contains instructions on how to upgrade your FishEye instance to the FishEye 2.0 Beta.

Before you Start

- Before upgrading you should always read the [Release Notes](#) for the version you are upgrading to, as well as any versions you are skipping.
- We strongly recommend you make a backup of your data before upgrading FishEye.** Simply make

a copy of your FishEye_install_dir/var/data/ directory.

- [Download](#) the FishEye zip file.

Upgrade Procedure

Your upgrade procedure depends on whether you are using a separate FISHEYE_INST directory. Read more about FISHEYE_INST in the [Installation Guide](#).

Method 1: Using a Separate FISHEYE_INST Directory

1. Shutdown your existing fisheye server.
2. Make a backup of your FISHEYE_INST directory.
3. Extract the new FishEye version to a directory, leaving your [FISHEYE_INST](#) environment variable set to its existing location.
4. Start FishEye from the new installation.
5. Follow any version-specific instructions found in the [Release Notes](#).

Method 2: No Separate FISHEYE_INST Directory

You will need to copy some files from your old FishEye installation to your new one.

1. Extract the new FishEye instance into a directory such as /NEW_FISHEYE/.
2. Delete the /NEW_FISHEYE/var directory.
3. Shut down the old FishEye instance if it is running.
4. Copy /OLD_FISHEYE/config.xml to /NEW_FISHEYE/.
5. Copy (or move) the /OLD_FISHEYE/var directory to /NEW_FISHEYE/var.
6. If you have a Cenqua-issued FishEye license, copy your fisheye.license to /NEW_FISHEYE/. (Atlassian-issued licenses are included within config.xml.)
7. Start FishEye from the new installation.
8. Follow any version-specific instructions found in the [Release Notes](#).

Method 3 - Without a FISHEYE_INST Directory, but would like to set one up

1. Shut down the old FishEye instance if it is running.
2. Set up the FISHEYE_INST [environment variable](#), then create the FISHEYE_INST directory on your filesystem.
3. Copy the /OLD_FISHEYE/config.xml to /FISHEYE_INST.
4. Copy the /OLD_FISHEYE/var directory to /FISHEYE_INST.
5. Extract the new Fisheye archive into a directory such as /NEW_FISHEYE/.
6. Start Fisheye from the new installation by running NEW_FISHEYE/bin/run.sh. (Use run.bat on Windows).
7. 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 FISHEYE_HOME and FISHEYE_INST. Check your FISHEYE_INST is pointing to the right directory.

FishEye 2.0 Beta Reviewer's Guide

FishEye 2.0 Beta is a public development release leading up to **FishEye 2.0**. For all production use and testing of FishEye, please use [the latest official release](#).



Do not use in production.
Beta releases should not be used in production environments.

 Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing FishEye development process. As such:
 - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
 - Features in development releases may be incomplete, or may change or be removed before the next full release.
 - FireFox 3 and Safari are the only browsers supported.

Thank you for your interest in the FishEye 2.0 Beta. This page contains some direction on what is ready for testing in the beta, what the known issues are and how you can submit feedback.

Known Issues

This is a list of known issues with the beta; please do not raise requests related to these as solutions for them are already under way.

- Timestamp information will be destroyed; if you install the beta release as an upgrade on your existing FishEye instance data, then every timestamp in the database will be reset to midnight and that information is irreversibly lost. The date information remains intact.
- Clicking on the calendar date picker from the Changelog filter makes the filter disappear, without applying the constraints.

Features Ready For Testing

The following features in the FishEye 2.0 Beta are relatively hardened and using these thoroughly will help contribute to the final product.

- External Database Support; You can now store Crucible's internal data (reviews and associated data) in a MySQL Enterprise Server or PostgreSQL database, as an alternative to the built-in HSQLDB. Note: Fisheye's repository cache's and the Fisheye user data is still stored on disk using infinity db.
- Stars; add colleagues, reviews and files to your favourites list, then view updates related to them as a feed.
- Charlietars; the automatically generated Crucible avatars should work smoothly. Also, you can sign up to Globally Recognised Avatars (<http://www.gravatar.com>) to upload a profile image and use that instead of the Charlie image.

Submitting feedback

To submit feedback on the FishEye 2.0 Beta, please use the [FishEye Forums](#).

JIRA Integration in FishEye 2.0 Beta

FishEye 2.0 Beta is a public development release leading up to **FishEye 2.0**. For all production use and testing of FishEye, please use [the latest official release](#).

 Do not use in production.
Beta releases should not be used in production environments.

 Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing FishEye development process. As such:
 - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
 - Features in development releases may be incomplete, or may change or be removed before the next full release.

This page contains instructions for setting up JIRA integration in FishEye.

 [JIRA](#) is Atlassian's issue tracking product, which can be used to manage projects and associated work.

 Before you begin: Ensure that you configure your JIRA instance to [enable sub-tasks](#), [enable unassigned issues](#) and [allow Remote API access](#). The instructions on this page have been tested with JIRA 3.13.4.

On this page:

- [Opening the Administration Screen for JIRA Integration](#)
- [Adding a New JIRA Server](#)
- [Editing Default JIRA Server Mappings](#)
- [Operations on Existing Servers](#)
 - [Edit settings for an existing JIRA server](#)
 - [Edit mappings for an existing JIRA server](#)
 - [Delete an existing JIRA server](#)

JIRA issues can be viewed in the main Dashboard view in FishEye. This requires you to enter details on the required JIRA server(s) via the FishEye administration screens.

Opening the Administration Screen for JIRA Integration

To set up JIRA integration, open the Administration screen and then click '**JIRA Servers**' under the '**Global Settings**' sub-menu on the left navigation bar. The '**View JIRA Servers**' administration page opens.

Screenshot: The View JIRA Servers Page

View JIRA Servers 								
Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete
Add a JIRA Server Edit default JIRA Server mappings								

On the View JIRA Servers page, you can carry out a number of operations as listed on this page.

Adding a New JIRA Server

To add a new JIRA server from the View JIRA Servers page, click '**Add JIRA Server**'.

The '**Add JIRA Server**' page opens.

Screenshot: The Add JIRA Server Page

Add JIRA Server

Server Details

Name:

URL:

Default Authentication

Username:

Password:

Options

☐ Include in Activity Streams

☐ Authenticate as Trusted Application

Test
 Save
 Cancel

A number of fields and options must be filled out or selected on this page. See the table below for information on each field.

Option	Type	Description	Required
Name	Text Field	A descriptive name for the JIRA server.	Yes
URL	Text Field	The Internet address of the JIRA server.	Yes
Allow Unassigned	True/False Button	Allow unassigned sub-tasks.	No
Username	Text Field	The username of an account on the JIRA instance (All activity that takes place will be attributed to this user, unless using the Trusted Application setting).	Yes
Password	Text Field	The password for the account on the JIRA instance.	Yes
Include in Activity Streams	Check Box	Allows JIRA information to appear on the Dashboard.	No

Authenticate as Trusted Application	Check Box	Allows the system to interface with JIRA and let users log on with their own accounts (and use their own accounts on the JIRA server. See complete FishEye documentation and complete JIRA documentation .	No

Once you've filled out the necessary fields, click **'Test'** to ensure that your details are correct. If you have a positive message return from the test, click **'Save'**.

Editing Default JIRA Server Mappings

i This setting enables the FishEye feature that shows JIRA information in a dynamic window when you hover the mouse over a JIRA issue key in FishEye. It will also turn every issue key into a hyperlink to that issue in FishEye.

To enable this feature, click **'Edit Default JIRA Server Mappings'** from the View JIRA Servers page. The **'Map JIRA Project Default'** page opens.

Screenshot: The Default JIRA Server Mappings Page

On this page, select the FishEye repositories or Crucible Projects that you wish to associate with all the JIRA servers you have configured for use in FishEye. You can click **'add all'** to quickly include them all in this category. You can remove individual items by clicking the small 'X' marks.

Once you've finished, click **'Save'**.

! You should disable any existing FishEye [linkers](#) you have set up for JIRA, as they will override this feature and prevent the dynamic dialog box from appearing when you mouse over an issue.

Operations on Existing Servers

Once you have configured an existing JIRA server, there are three main operations you can carry out on it: **'Edit'**, **'Mappings'** and **'Delete'**. These options appear on the far right of the screen.

Screenshot: Operations in the JIRA Servers Page

View JIRA Servers 								
Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete
Add a JIRA Server Edit default JIRA Server mappings								

Edit settings for an existing JIRA server

When you click '**Edit**', you can adjust any of the general settings you configured when you first added the server.

Edit mappings for an existing JIRA server

When you click '**Mappings**', a page is loaded that is almost identical to the '**Default Mapping**' screen, but allows you to choose mappings only for that specific JIRA server.

Delete an existing JIRA server

Clicking '**Delete**' will remove the server from the list.

Git Alpha in FishEye 2.0 Beta

FishEye 2.0 Beta is a public development release leading up to **FishEye 2.0**. For all production use and testing of FishEye, please use [the latest official release](#).



Do not use in production.
Beta releases should not be used in production environments.



Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing FishEye development process. As such:
 - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
 - Features in development releases may be incomplete, or may change or be removed before the next full release.

This page contains information about the alpha pre-release support for [Git 1.6](#), which is activated (but not complete) in the FishEye 2.0 Beta.

Introduction and Disclaimer

Git Alpha support is included in this beta release for Git 1.6. Atlassian stresses that this is alpha level support which means the following:

- There is no admin support yet (no UI controls for the feature).
- There are missing features - notably author line count support.
- Subsequent betas are likely to require re-indexing (although Git indexing is reasonably fast).
- Problems are to be expected. Do not use this on production instances.

However, feedback (see below) is appreciated and your comments will contribute to the quality of the final product.

Configuration

To configure a Git 1.6 repository in FishEye, you will need to stop FishEye and edit your `config.xml` file directly. An example repository configuration follows.

```
<repository name="test" enabled="true">
  <git location="git://git.example.com/repo.git"/>
    <security>
      <required-groups/>
    </security>
</repository>
```

The <repository> tag must directly follow the </repository-defaults> ending tag, or another ending tag for a repository (</repository>). If you do not place the repository tag correctly your application will not start up and is likely to throw an error like the following:

```
ERROR - Errors parsing /Path/To/FISHEYE_INST/config.xml:
ERROR - at line 373: Expected elements 'backup@
http://www.cenqua.com/fisheye/config-1 check-for-updates@
http://www.cenqua.com/fisheye/config-1 quicksearch-weights@
http://www.cenqua.com/fisheye/config-1 database@
http://www.cenqua.com/fisheye/config-1' instead of 'repository@
http://www.cenqua.com/fisheye/config-1' here in element config@
http://www.cenqua.com/fisheye/config-1
```

Submitting Feedback

We're very interested in your feedback. The best place for submitting feedback is the [FishEye forums](#).

FishEye 1.6 Release Notes

 FishEye 2.10 has now been released. Read the [Release Notes](#).

23 September 2008

 For details on minor releases since FishEye 1.6, see the [FishEye Changelog](#).

Atlassian presents FishEye 1.6

FishEye release 1.6 is a major release that adds functional and performance improvements. FishEye 1.6 has a faster, more powerful Quick Search which includes change indexing. It also now supports assigning administration privileges to user accounts or groups. This new FishEye is faster, containing both tune-ups of the core code as well as new features enhancing use in teams. Finally, there's also been additions made to the technology powering FishEye extensions, for third-party developers.

Highlights of this release:

- [FishEye Search Enhancements](#)
- [Multiple Admin Users](#)
- [Remote API Improvements](#)
- [Changes to Charts](#)
- [Performance Tweaks](#)
- [Numerous improvements and bug-fixes](#)



Upgrading to FishEye 1.6

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

Highlights of FishEye 1.6

1

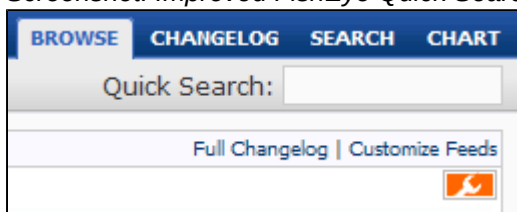
FishEye Search Enhancements

FishEye now indexes the content of every commit. This enables searching on all content in your Subversion, Perforce or CVS repository. Furthermore, you can search for added or deleted content. [Read more](#).

Quick Search

FishEye's Quick Search has been completely rewritten for better accuracy and performance; Quick Search results are now returned instantaneously regardless of the repository size. Result types are better weighted to increase their relevance. Results have content preview with hit highlighting and are properly weighted, taking time into account. Changeset results are returned based on content modifications, additions and deletions. FishEye now indexes the full content of every commit and will return changesets for content hits. Path and filename search is now an order of magnitude faster on large repositories. Finally, support for keywords in search enables you to quickly get the result you are looking for. For example, entering `cs:1902` will take you straight to changeset 1902.

Screenshot: Improved FishEye Quick Search



General Search

You can now search specifically for added and deleted content. This enables you to quickly find when code was deleted or modified. For instance, you can search for a method name to find out what it was previously called.

Advanced Search and EyeQL

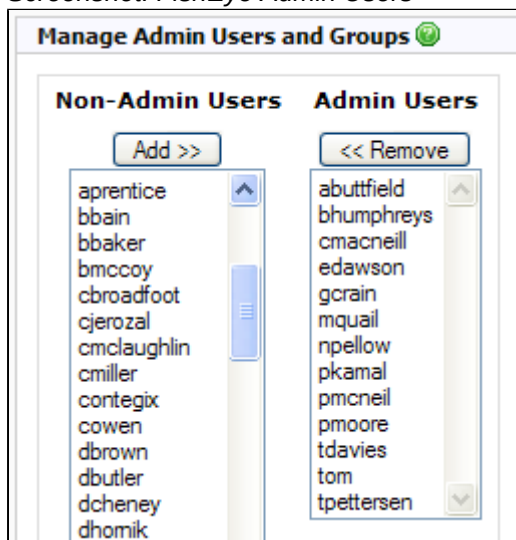
All of the improvements made to Quick Search and General Search are available when using Advanced Search and EyeQL.

2

Multiple Admin Users

FishEye now allows the Administrator to grant other FishEye users Admin status. These Admin Users can also carry out any of the tasks that may have required the Administrator password. Admin privileges can be conferred using built-in or external directory group membership. [Read more](#).

Screenshot: FishEye Admin Users



3

Remote API Improvements

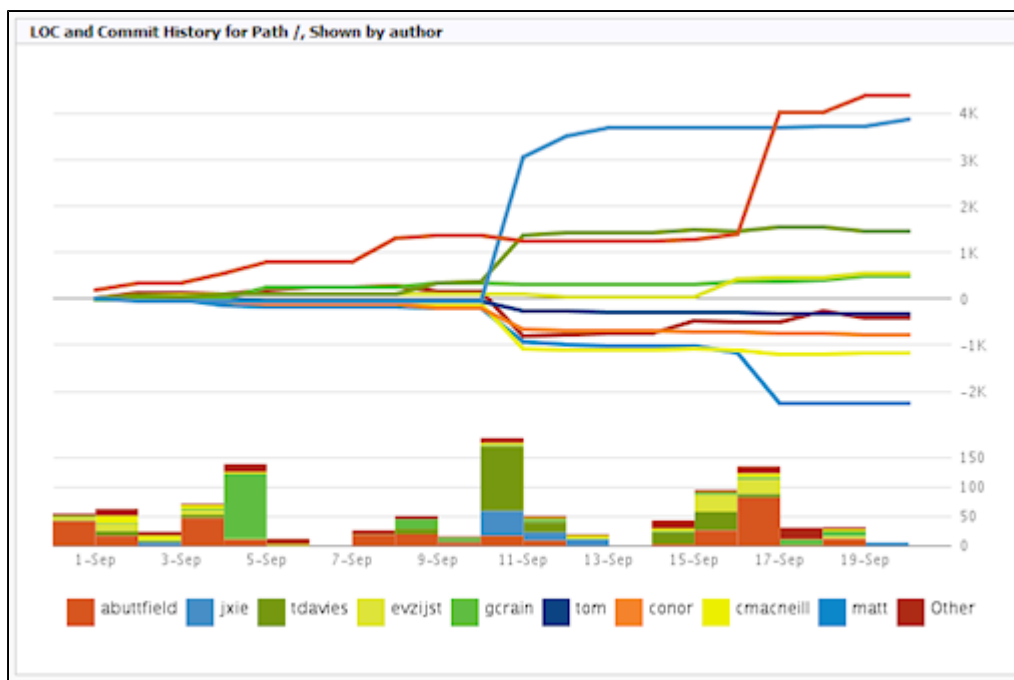
Enhancing the Remote API, the FishEye team have incorporated a **maxreturns** option, which lets you control the quantity of returned results. Also the range of functions is extended with the new **ancestor return** clause and **history search** items. Finally, Perforce data is now exposed in the remote API, allowing Perforce jobs to be accessed via remote API calls. [Read more.](#)

4

Changes to Charts

FishEye 1.6 has a new **change** chart type. The change chart shows relative net line activity for a period. It is intended to give a quick "zoomed in" view of activity by extension, author, or subdirectory for a short period. Change charts start from a linecount of 0 at the start date for comparative purposes. [Read more.](#)

Screenshot: FishEye's Change Chart



5

Performance Tweaks

Performance users can now specify a changelist to start scanning from. For some users, this will dramatically improve indexing time and runtime performance by ignoring irrelevant historical data. This is achieved with one simple configuration option called '[skip labels](#)'.

6

Numerous improvements and bug-fixes

JIRA Issues (68 issues)			
Key	Summary	Priority	Status
FE-638	webwork 2.2.6 is not setting svnsymbolic in editrepository		Closed
FE-674	RSS Feed Title is missing space		Closed
FE-675	RSS Feed Entries have almost no information in title		Closed
FE-668	"Edit repository details" throws an NPE when a p4 repo has invalid info		Closed
FE-658	Chart constraint dropped on second level subdir		Closed
FE-650	Documentation: New 'Advanced' mode hides attributes in Add Repository screen		Closed

FE-648	Disabling the check box next to the config.xml file still sends the config.xml file via Admin > SysInfo > Raise a support request		 Closed
FE-630	Bundle SAL 1.1 in FishEye		 Closed
FE-617	for new svn repositories, default for t/b/t should be None		 Closed
FE-605	added files appear as empty diffs		 Closed
FE-600	TODO appearing in UI for comments		 Closed
FE-288	Repositories still occasionally get stuck in Stopping state		 Closed
FE-82	Re-index request: show message "Could not stop repository within 20 seconds. Re-index aborted."		 Closed
FE-437	improve performance of filename searches in quicksearch		 Closed
FE-438	Do content searches in quicksearch		 Closed
FE-436	Improved Quick Search		 Closed
FE-393	Use a single regular expression to catch all tag/branch/trunk patterns		 Closed
FE-378	review multithreading of RevCacheReader		 Closed
FE-467	Calculating the correct version for diffs doesn't work for perforce		 Closed
FE-435	Quick-search redo UI requirements		 Closed
FE-434	show match-in-context in quicksearch (hit highlighting)		 Closed
FE-389	only ask group to do group-membership tests for crowd users		 Closed
FE-338	Please add ability to specify initial revision from which to begin initial scan		 Closed
FE-301	IndexOutOfBoundsException on when opening annotated view		 Closed

FE-296	Get Id button does not work in Trusted Application screen under IE		 Closed
FE-584	Create the ability for customers to create support cases via Fisheye		 Closed
FE-578	"Search just <repo:parh>" breadcrumb links are borked - escapes parameter separators		 Closed
FE-566	diff-to-previous on annotate page 404s		 Closed
FE-562	resolve springsource/log4j versioning problem		 Closed
FE-549	Search tokenizes on underscores		 Closed
FE-532	upgrade to webwork 2.2.7 (fixes security problem)		 Closed
FE-521	CLONE -StackOverflow		 Closed
FE-559	Add 'ancestor' return clause to EyeQL		 Closed
FE-555	Documentation: Add maxReturn parameter to remote API		 Closed
FE-554	Add LIMIT clause to EyeQL documentation		 Closed
FE-552	Have an error page rather than 403 page when SVN permission denied		 Closed
FE-550	improve catch-all svn symbolic regex		 Closed
FE-541	Allow limiting of number of results returned by remote API		 Closed
FE-540	"Data Types and Structures" information in the Remote API page		 Closed
FE-536	"List Repositories" method in the remote API documentation		 Closed
FE-524	upgrade to trusted apps 1.0, remove seraph dep		 Closed
FE-483	Ensure all user preferences are in user profile		 Closed
FE-515	Allow P4 label scanning to be skipped		 Closed
FE-511	NPE when configuration file was not found		 Closed

FE-505	Retrieve Password		 Closed
FE-504	command line reindex doesn't work when loopback is not 127.0.0.1		 Closed
FE-503	com.cenqua.fisheye.svn.SvnCache is throwing NPEs		 Closed
FE-500	Expose Changeset "Fixes Perforce Jobs" data in EyeQL (and REST api)		 Closed
FE-494	Add a link to the changeset on the annotation page		 Closed
FE-479	next and previous links on diff and annotation pages		 Closed
FE-687	RSS Feed entry items truncate differently to 1.5.x		 Closed
FE-685	Automatic user management from Crowd		 Closed
FE-651	Weight quicksearch results by date		 Closed
FE-649	Cannot specify starting revision when creating perforce repo. The option only appears in the edit screen		 Closed
FE-643	"Request Garbage Collection" link on SysInfo/Support page redirects badly		 Closed
FE-636	Make update polling configurable in admin section		 Closed
FE-607	Create documentation for new feature, create support issues via fisheye Admin > Sysinfo screen		 Closed
FE-596	CSS syntax highlighting omissions		 Closed
FE-464	upgrade to latest spring (at least 2.5.4)		 Closed
FE-428	Check for updates option		 Closed
FE-319	Can't access /admin/ when logged in via Crowd		 Closed
FE-482	SVN repositories default to UTF-8		 Closed
FE-513	Redirected to javax.servlet.ServletException 500 error page when		 Closed

[accessing to a disabled/stopped repository](#)

FE-495	Self Signup layout borked	↓	 Closed
FE-491	Hide optional fields in repo setup	↓	 Closed
FE-678	Link to file view in quicksearch results	↓	 Closed
FE-529	Add an "Add Repository" link to the top of the repository list page in admin screen	↓	 Closed
FE-487	Remove the colon trailing repository names on the Fisheye home page	↓	 Closed

FishEye 1.6 Changelog

This page contains information about the FishEye 1.6 minor releases.

 Please read the [FishEye 1.6 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 1.6.5.a to 1.6.6](#)
- [From 1.6.4 to 1.6.5.a](#)
- [From 1.6.3 to 1.6.4](#)
- [From 1.6.2 to 1.6.3](#)
- [From 1.6.1 to 1.6.2](#)
- [From 1.6.0 to 1.6.1](#)

From 1.6.5.a to 1.6.6

10th February 2009

This release is a bugfix release which address the following issues:

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

From 1.6.4 to 1.6.5.a

22 December 2008

This release contains a number of improvements and bug fixes.

 If you are using Perforce with a case-insensitive file system (such as in Windows), then you will need to re-index your repository after upgrading to this release.

Full list of issues fixed in this release:

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

From 1.6.3 to 1.6.4**20 November 2008**

This release contains bug fixes and minor improvements, and includes the new plugin points developed for [Atlas Camp 2008](#).

Full list of issues fixed in this release:

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

From 1.6.2 to 1.6.3**5 November 2008**

This release rolls together several improvements and bug fixes.

- FishEye now indexes diff text for CVS repositories.
- Quicksearch improvements.
- Improve support for local SVN 1.5 repositories.

 To take advantage of FishEye's diff text search or historical file content quick search functions, you will need to re-index your repository.

Full list of issues fixed in this release:

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

From 1.6.1 to 1.6.2

This version number was skipped, in order to keep FishEye and Crucible version numbers in parallel. There is no FishEye release with a version of 1.6.2.

From 1.6.0 to 1.6.1**24 September 2008**

This is a bug fix release.

- [FE-700](#) An initial import of a subversion repository will generate a NPE in some circumstances.

FishEye 1.6 Upgrade Guide

Below are some important notes on upgrading to **FishEye 1.6**. For details of the new features and improvements in this release, please read the [FishEye 1.6 Release Notes](#) and [FishEye 1.6 Changelog](#).

On this page:

- [Upgrade Notes](#)
- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Notes

- To take advantage of the new diff contents search in Quick Search, you will need to re-index your repository.
- Crowd 1.3 users will need to upgrade to Crowd 1.4.4 or later due to an incompatibility with this version of FishEye.

Upgrade Procedure

Upgrade a test environment first

As always, please test your upgrades in your test environment before rolling into production.

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 Knowledge Base](#) for known issues 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 1.6 Release Notes](#)

[FishEye 1.6 Changelog](#)

FishEye 1.5 Release Notes

 FishEye 2.10 has now been released. Read the [Release Notes](#).

15 April 2008

 For details on minor releases since FishEye 1.5, see the [FishEye Changelog](#).

Atlassian presents FishEye 1.5

FishEye release 1.5 is a major release that adds the ability to present historical, per-author line count information. This new suite of graphs show how much each user has contributed to the code base, over time.

Highlights of this release:

- [Per-author lines of code statistics](#)
- [Charting improvements](#)
- [Customisable email templates](#)
- [Numerous improvements and bug-fixes](#)



Upgrading to FishEye 1.5

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

Highlights of FishEye 1.5

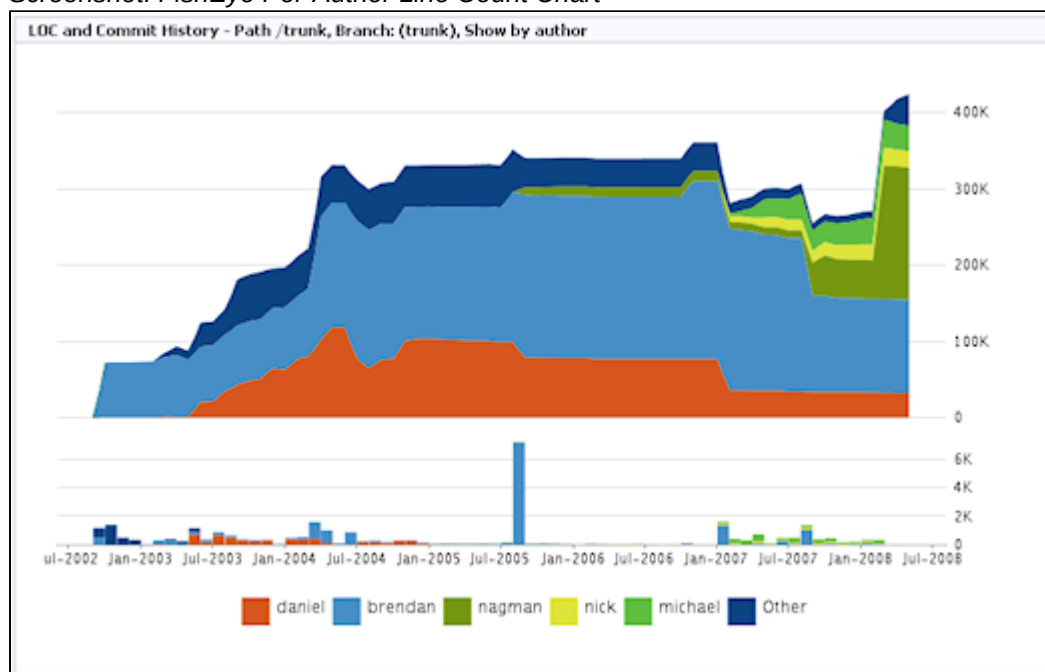
1

Per-author lines of code statistics

Statistics for lines of code is now broken down per-author, providing an all-new level of detail. This allows you to see how many lines of code were contributed to your project by each author, over time.

⚠ This requires changing a setting and re-scanning existing repositories. See the [FishEye upgrade guide](#) for more information.

Screenshot: FishEye Per-Author Line Count Chart

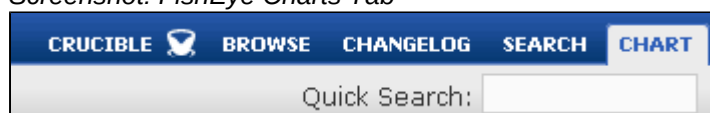


2

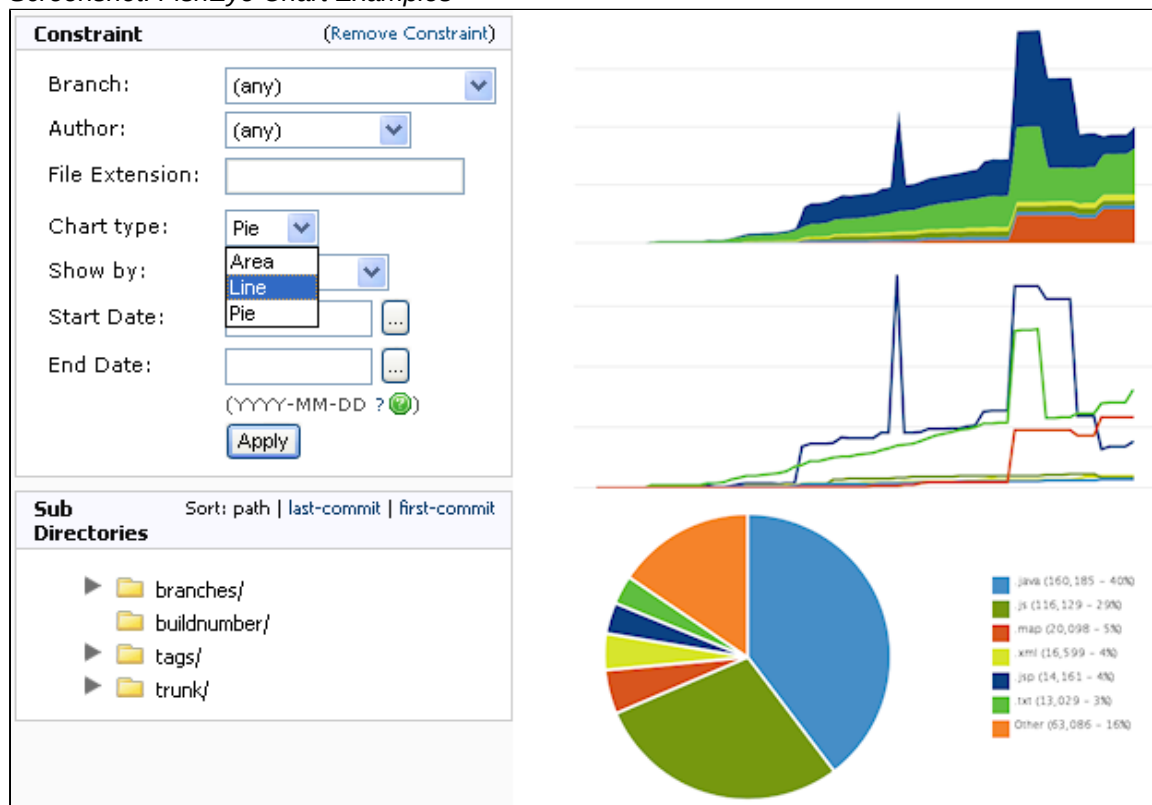
Charting improvements

The line graphs in FishEye have been improved, providing a better view of lines of code statistics from your project, as well as showing how this has grown.

Screenshot: FishEye Charts Tab



Screenshot: FishEye Chart Examples



3

Customisable email templates

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

4

Numerous improvements and bug-fixes

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

FishEye 1.5 Changelog

This page contains information about the FishEye 1.5 minor releases.



Please read the [FishEye 1.5 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 1.5.3 to 1.5.4](#)
- [From 1.5.2 to 1.5.3](#)
- [From 1.5.1 to 1.5.2](#)
- [From 1.5.0 to 1.5.1](#)

From 1.5.3 to 1.5.4

1 August 2008

This release contains minor improvements and bug fixes.

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

From 1.5.2 to 1.5.3

23 June 2008

This release contains bug fixes.

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

From 1.5.1 to 1.5.2

27 May 2008

This release contains bug fixes.

 Note: This release of FishEye corrects to some Perforce line counts when storing diffs. If you have been having this problem, you will need to ensure that the store-diffs setting is set to "true" and do a full re-index of your repository.

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

From 1.5.0 to 1.5.1

24 April 2008

This release contains bug fixes.

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

FishEye 1.5 Upgrade Guide

Below are some important notes on upgrading to **FishEye 1.5**. For details of the new features and improvements in this release, please read the [FishEye 1.5 Release Notes](#) and [FishEye 1.5 Changelog](#).

On this page:

- [Upgrade Notes](#)
- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Notes

- Per-Author Line Counts require the '**Store Diff**' setting to be true, but this will be false for existing repositories. Existing repositories will require this setting to be changed to 'true', then a full re-scan of that repository must be done. This is not essential to continue using your FishEye instance — it is only required for per-author graphs on the FishEye chart page.
- Upgrading from 1.2.5 (or earlier) or 1.3beta8 (or earlier) will force a complete **re-scan of CVS repositories**.
- Upgrading to this version will force a complete **re-scan of P4 repositories**.
- Upgrading from 1.1.x (or earlier) or 1.2beta2 (or earlier) will force a complete **re-scan of SVN repositories**.

Upgrade Procedure

Upgrade a test environment first

As always, please test your upgrades in your test environment before rolling into production.

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 Knowledge Base](#) for known issues 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 1.5 Release Notes](#)

[FishEye 1.5 Changelog](#)

FishEye 1.4 Release Notes

 FishEye 2.10 has now been released. Read the [Release Notes](#).

5 December 2007

 For details on minor releases since FishEye 1.4, see the [FishEye Changelog](#).

The Atlassian FishEye team is delighted to present FishEye 1.4.

FishEye 1.4 is a major release which focuses on integration, user management and performance.

The updated FishEye Administration interface provides support for groups and improved user management screens. The new built-in integration with Atlassian [Crowd](#) extends your authentication and authorisation capabilities. You can now include users and groups from one or more Crowd directories, and provide single

sign-on (SSO) across Atlassian products plus any other applications that support SSO.

The latest releases of FishEye and Crucible work together even more tightly than before, allowing you to see at a glance which files/changesets have been reviewed. You can also search within FishEye for files that have not yet been reviewed.

Highlights of this release:

- [Enhancements to user management](#)
- [Crowd/SSO support](#)
- [Crucible integration](#)
- [Enhancements to JIRA plugin](#)
- [Plus over 30 improvements and bug-fixes](#)

Responding to your feedback:

★ 19 new feature requests/improvements implemented

Your [\[votes and issues\]](#)<http://jira.atlassian.com/browse/FE> help us keep improving our products, and are much appreciated.

Upgrading to FishEye 1.4

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

Highlights of FishEye 1.4

1

Enhancements to user management

In FishEye 1.3.7, we introduced support for [public signup](#) (self-registration). Now in release 1.4:

- Improved user interface makes user administration easier.
- [Groups](#) are supported.
- [Renaming of users](#) is supported.
- Read the [documentation](#).

Edit User Groups: matt

Available Groups	Groups
Join >>	<< Leave
team-2	team-1
	team-3

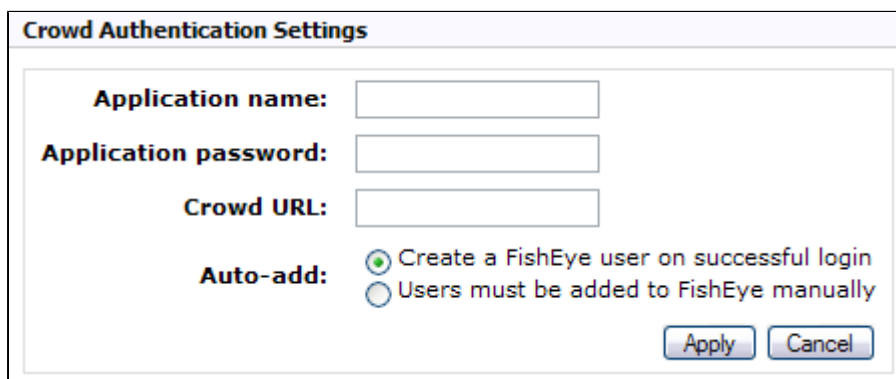
[Back to matt](#)

2

Crowd/SSO support

- Inbuilt integration with Atlassian Crowd for authentication and authorisation.

- Users and groups in your Crowd directories now supported in FishEye.
- Single sign on (SSO) support via Crowd e.g. you can now sign in just once to access Atlassian JIRA, FishEye, Crucible, Confluence and Bamboo, and any other applications which support SSO.
- Read the [documentation](#).



Crowd Authentication Settings

Application name:

Application password:

Crowd URL:

Auto-add:

☒ Create a FishEye user on successful login

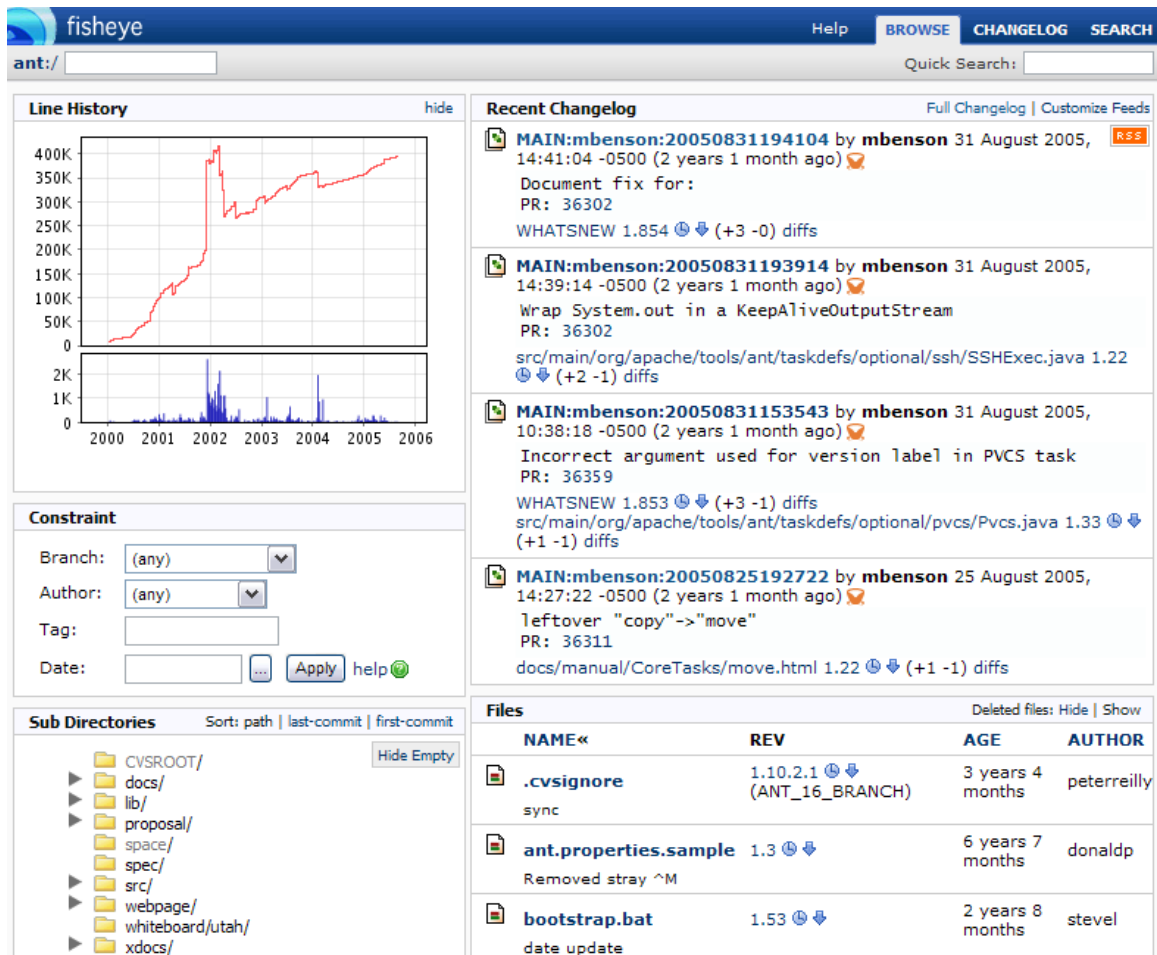
☐ Users must be added to FishEye manually

3

Crucible integration

Closer integration between FishEye 1.4 and [Crucible 1.2](#):

- Links to existing Crucible reviews on the FishEye screens. So you can see which files/changesets have been reviewed.
- Search for Crucible data via EyeQL. For example, you can search for files that have not yet been reviewed.



4

Enhancements to JIRA plugin

The new version 1.2 of the [FishEye-for-JIRA plugin](#) includes some useful improvements:

- New 'FishEye' tab for JIRA issues and projects.
- Improved ability to create a Crucible review from the 'FishEye' tab within a JIRA issue.
- The 'FishEye' tab now shows review status (if applicable).
- Ability to connect your JIRA instance to *multiple* FishEye instances.
- Ability to configure the FishEye plugin via the AppLinks plugin.
- The FishEye plugin is now fully internationalisable.

5

Plus over 30 improvements and bug-fixes

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

FishEye 1.4 Changelog

This page contains information about the FishEye 1.4 minor releases.

⚠ Please read the [FishEye 1.4 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 1.4.2 to 1.4.3](#)
- [From 1.4.1 to 1.4.2](#)
- [From 1.4 to 1.4.1](#)

From 1.4.2 to 1.4.3

7 February 2008

This release contains bug fixes.

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

From 1.4.1 to 1.4.2

This release contains some minor improvements and bug fixes.

- **Trusted Application Support**
FishEye now allows you to set up trusted communications with other Atlassian applications. At this point, the JIRA FishEye plugin supports Trusted Applications. The JIRA FishEye plugin can request information from FishEye on behalf of the currently logged-in user, and FishEye will not ask the user to log in again or to supply a password. Previously FishEye would have used a single 'system' account to determine permissions. Now, FishEye/Crucible can apply the correct permission settings for the logged-in user.
- FishEye now bundles the SVNkit Client as the default library for interfacing with Subversion. This streamlines FishEye configuration for Subversion users.
- FishEye now pre-calculates line-graph data. This should improve performance in the rendering of line graphs.
- Hyphens are now allowed in project key names.

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

From 1.4 to 1.4.1

This is a small bug-fix release.

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

FishEye 1.4 Upgrade Guide

Below are some important notes on upgrading to **FishEye 1.4**. For details of the new features and improvements in this release, please read the [FishEye 1.4 Release Notes](#) and [FishEye 1.4 Changelog](#).

On this page:

- [Upgrade Notes](#)
- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Notes

- If you were using the old Crowd Authenticator plugin in FishEye 1.3.x, you need to switch to the in-built Crowd Authenticator in FishEye 1.4+. Please see the [Crowd documentation](#) for details.
- As of version 1.3, FishEye requires a **JVM version 1.5** or later. Previously, 1.4+ was required.

- Upgrading from 1.2.5 (or earlier) or 1.3beta8 (or earlier) will force a complete **re-index of CVS repositories**.
- Upgrading to this version will force a complete **re-index of P4 repositories**.
- Upgrading from 1.1.x (or earlier) or 1.2beta2 (or earlier) will force a complete **re-index of SVN repositories**.

Upgrade Procedure

Upgrade a test environment first

As always, please test your upgrades in your test environment before rolling into production.

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 Knowledge Base](#) for known issues 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 1.4 Release Notes](#)

[FishEye 1.4 Changelog](#)

FishEye 1.3 Release Notes

 FishEye 2.10 has now been released. Read the [Release Notes](#).

1 August 2007

 For details on minor releases since FishEye 1.3, see the [FishEye Changelog](#).

FishEye 1.3 contains many bug fixes and improvements, and adds support for [Perforce](#).

Upgrading FishEye

You can now download FishEye from [here](#). Information on installing FishEye can be found [here](#). If upgrading from a previous version, please follow the [Upgrade Guide](#).

Highlights of FishEye 1.3

- 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)
- Refer to the [changelog](#) for more details.

FishEye 1.3 Changelog

This page contains information about the FishEye 1.3 minor releases.

 Please read the [FishEye 1.3 Upgrade Guide](#) before upgrading to any of the minor releases below.

On this page:

- [From 1.3.7 to 1.3.8](#)
- [From 1.3.6 to 1.3.7](#)
- [From 1.3.5 to 1.3.6](#)
- [From 1.3.4 to 1.3.5](#)
- [From 1.3.3 to 1.3.4](#)
- [From 1.3.2 to 1.3.3](#)
- [From 1.3.1 to 1.3.2](#)
- [From 1.3 to 1.3.1](#)
- [From 1.3beta9 to 1.3](#)
- [From 1.3beta8 to 1.3beta9](#)
- [From 1.2.5 to 1.3beta8](#)

From 1.3.7 to 1.3.8

This release provides performance and bug fix improvements for Subversion and Perforce indexing including:

- Perforce determination of line counts is much more efficient
- Handle cases where the content of a file has been removed from a Perforce depot
- Change Subversion indexing strategy for repositories whose initial check-in is a large commit or copy
- Fix error which broke line count graph caching
- correct issues dealing with changes to tagged files
- correct issue with property changes in German locale when using SvnKit

From 1.3.6 to 1.3.7

This is a small bug-fix release ([list of issues](#)).

From 1.3.5 to 1.3.6

This is a small bug-fix release. It addresses a stack-overflow problem for some configurations.

From 1.3.4 to 1.3.5

Note: Upgrading to this version will force a complete re-index of P4 repositories.

Improvements

- User-friendly UI-based license entry and maintenance.
- Syntax highlight files when displaying a diff.
- Add Ruby and Coldfusion syntax highlighting. Look for custom syntax files in FISHEYE_INST/syntax.
- EyeQL enhancement: a new clause to match on changeset ids `csid = "1234"`.
- EyeQL enhancement: more return types such as `isAdded` and `isBinary`.

Fixes

- Many performance improvements when scanning Perforce repositories.
- Changelog missing some changesets in CVS when there is a path constraint.
- Handle svn diff output in languages other than English.
- Fix problem with long directory names wrapping in the directory tree.

From 1.3.3 to 1.3.4

- Fix compatibility between Cenqua and Atlassian licenses.
- [SVN] Convert slash / characters to a hyphen - in tag/branch names.
- [SVN] Better handling of precedence of tag/branch/trunk symbolic rules.
- [SVN] Add a manual test field in Admin screens to test symbolic rule setup.
- [P4] Add ability to configure Perforce updater in Admin screens.
- [P4] Some improvements when scanning Perforce repositories.

From 1.3.2 to 1.3.3

This build allows FishEye to be used with Atlassian licenses.

From 1.3.1 to 1.3.2

- Fix potential XSS vulnerability in quick-search page.
- Fix problem sending watch emails where the commit message contains a tab character.
- [SVN] Add support for requesting a rescan between given revisions.
- [SVN] Improve scan performance, and better handle add operations from outside FishEye's view of the repository.
- [SVN] Improve scan performance by not fetchings diffs for binary files.
- [SVN] Timeout settings now configurable via Admin screens.
- [SVN] Display SVN properties at the directory level.
- [P4] Address problem fetching labels for files.
- Fix Javascript problem in IE when logging into the Admin screens.

From 1.3 to 1.3.1

- The truncate diff setting should now work in Internet Explorer.
- Fix issue with duplicate paths in tarball generation.
- [P4] An empty <p4-config> element would cause an error.
- Unknown repos now return a 404 status rather than 500.
- [SVN] Handle empty content files when using SvnKit.
- [CVS] Allow \$ in author names.
- FishEye now uses the tabwidth setting in each user's profile.
- [SVN] Fix issue where FishEye incorrectly states that no username was supplied.
- Fix IE7 directory spacing problem.
- Implement side-by-side diffs.

From 1.3beta9 to 1.3

- Various improvements when scanning Perforce repositories.
- [SVN] Fix for problem with diff hyperlinks to re-added files.
- Fix problem where some paths were not correctly html-escaped.
- Fix 'NoSuchFieldError deferredExpression' problem on some platforms (due to a 3rd-party library being included twice).
- Ensure LDAP connections are closed in all situations.

From 1.3beta8 to 1.3beta9

Upgrading to 1.3beta9 will force a complete re-index of CVS repositories.

- Upgrade JVM requirement to 1.5+.
- Upgrade embedded HTTP engine (Jetty). This fixes some bugs and improves performance under load.
- Fix a performance problem (esp. under load). "Recent Changes" pages should return much faster now.
- Fix a very slow memory leak when FishEye is under load (for example, when it is being crawled by a web spider).
- Fix a problem where daily-emails would break after a backup was performed.
- [CVS] Fix an error introduced when FishEye builds its repository cache. This requires a full re-scan of CVS repositories.

- [CVS] Fix a problem where FishEye could not parse in RCS files author names that were only numerical digits.
- [CVS] Fix bug when creating tar/zip files from a branch constraint.
- [SVN] FishEye will now timeout long running SVN connections that have blocked.
- [SVN] Fix problem where FishEye was not storing SVN properties correctly.
- [SVN] Fix a bug when entering a revision beyond the current last revision in quick search.

From 1.2.5 to 1.3beta8

- [SVN] When importing a repository from a given start revision, you can now nominate if it should import the state of the repository at that revision, or just import changes made after that revision.
- [CVS] Fix a bug where FishEye would send out watch emails for historical changesets after a re-index.
- Performance improvements to changeset page when one of the files in the changeset has a very large history.
- [SVN] Some changes that improve the speed of the initial-scan for some SVN repositories.
- Fix a bug when FishEye generates RSS feed urls constrained by author, when the author has an "@" in their name.
- [SVN] Fix a bug when a tag is deleted (as part of a move).

FishEye 1.3 Upgrade Guide

Below are some important notes on upgrading to **FishEye 1.3**. For details of the new features and improvements in this release, please read the [FishEye 1.3 Release Notes](#) and [FishEye 1.3 Changelog](#).

On this page:

- [Upgrade Notes](#)
- [Upgrade Procedure](#)
- [Checking for Known Issues and Troubleshooting the FishEye Upgrade](#)

Upgrade Notes

- As of version 1.3, FishEye requires a **JVM version 1.5** or later. Previously, 1.4+ was required.
- Upgrading from 1.2.5 (or earlier) or 1.3beta8 (or earlier) will force a complete **re-index of CVS repositories**.
- Upgrading to this version will force a complete **re-index of P4 repositories**.
- Upgrading from 1.1.x (or earlier) or 1.2beta2 (or earlier) will force a complete **re-index of SVN repositories**.

Upgrade Procedure

Upgrade a test environment first

As always, please test your upgrades in your test environment before rolling into production.

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 Knowledge Base](#) for known issues 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](#)

[g 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 1.3 Release Notes](#)

[FishEye 1.3 Changelog](#)

Security advisories

Finding and reporting a security vulnerability

Atlassian's approach to releasing patches is detailed in [How to Report a Security Issue](#).

Publication of Bamboo security advisories

Atlassian's approach to publishing security advisories is detailed in [Security Advisory Publishing Policy](#).

Severity levels

Atlassian's scale for measuring security issues is detailed in [Severity Levels for Security Issues](#).

Our patch policy

Atlassian's approach to releasing patches is detailed in our [Security Patch Policy](#).

Security advisories

- [FishEye and Crucible Security Advisory 2012-08-21](#)
- [FishEye and Crucible Security Advisory 2012-05-17](#)
- [FishEye and Crucible Security Advisory 2012-01-31](#)
- [FishEye and Crucible Security Advisory 2011-11-22](#)
- [FishEye and Crucible Security Advisory 2011-05-16](#)
- [FishEye and Crucible Security Advisory 2011-01-12](#)
- [FishEye Security Advisory 2010-10-20](#)
- [FishEye Security Advisory 2010-06-16](#)
- [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.atlassian.com/>.

In this advisory:

- [Elevation of privileges vulnerability](#)

Elevation of privileges vulnerability**Severity**

Atlassian rates the severity level of this vulnerability as **Medium**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

This is an independent assessment and you should evaluate its applicability to your own IT environment.

Description

We have identified and fixed a vulnerability in FishEye and Crucible that results from behaviour of certain third-party frameworks used in FishEye and Crucible. This vulnerability allows any attacker to:

- Set the FishEye and Crucible instance to allow anonymous access
- Set the FishEye and Crucible instance to allow anonymous signup

All versions of FishEye and Crucible up to and including 2.7.14 are affected by this vulnerability. The vulnerability is fixed in FishEye and/or Crucible 2.8.0 and later. This issue can be tracked at

[FE-4222 - Authenticate](#) to see issue details and [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. jira

FishEye and/or Crucible Vulnerability	Affected versions	Fixed Version	Issue Tracking
Elevation of privileges	2.5.x or earlier	2.5.9	FE-4222 - Authenticate to see issue details
	2.6.x	2.6.9	
	2.7.x	2.7.15, 2.8.0	CRUC-6188 - Authenticate to see issue details

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.

FishEye and Crucible Component	Affected FishEye and Crucible Versions	Fixed Version	Issue Tracking
WebWork 2	2.0 - 2.7.8	2.6.7 2.7.9	FE-3891

Risk Mitigation

We recommend that you upgrade your FishEye and Crucible installations to fix this vulnerability.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can restrict access to your instance of FishEye/Crucible by using a firewall.

Fix

FishEye and Crucible 2.6.7, 2.7.9 and later versions fix this issue. View the issue linked above for information on fix versions. For a full description of the latest version of FishEye and Crucible, see the [FishEye](#) and [Crucible](#) release notes. You can download the latest versions from the [FishEye](#) and [Crucible](#) download centres. There are no patches available for these issues.

FishEye and Crucible Security Advisory 2011-11-22

This advisory discloses a number of security vulnerabilities that we have found and fixed in versions of FishEye/Crucible earlier than 2.5.7. You need to upgrade your existing FishEye and Crucible installations to fix these vulnerabilities. Enterprise Hosted customers should request an upgrade by raising a support request at <http://support.atlassian.com> in the "Enterprise Hosting Support" project. Neither JIRA Studio nor Atlassian OnDemand are vulnerable to any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerabilities listed in this advisory have been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [XSS Vulnerabilities](#)
 - [Severity](#)

- [Risk Assessment](#)
- [Vulnerabilities](#)
- [Risk Mitigation](#)
- [Fix](#)
- [Permission Verification Vulnerabilities](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)

XSS Vulnerabilities

Severity

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, medium or low. These vulnerabilities are **not** critical.

Risk Assessment

We have identified and fixed a number of stored cross-site scripting (XSS) vulnerabilities which affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). XSS vulnerabilities allow an attacker to embed their own JavaScript into a FishEye/Crucible page. You can read more about XSS attacks on [Wikipedia](#), at [The Web Application Security Consortium](#) and other places on the web.

Vulnerabilities

The table below describes the FishEye/Crucible versions and the specific functionality affected by the XSS vulnerabilities.

FishEye/Crucible Feature	Affected FishEye/Crucible Versions	Fixed Version	Issue Tracking
FishEye user profile - display name	2.5.4 and earlier	2.5.5	FE-3797
FishEye user profile - snippets in a user's comment	2.5.4 and earlier	2.5.5	FE-3798

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable [public signup](#) to your instance as a temporary mitigation until you have applied the upgrade. For tighter access control, you can restrict access to trusted [groups](#).

Fix

FishEye and Crucible 2.5.5 and later versions fix two these issues. View the issues linked above for information on fix versions. You can download the latest versions of FishEye and Crucible from the download centres ([FishE](#)

[ye download centre](#), [Crucible download centre](#)).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

Permission Verification Vulnerabilities

Severity

Atlassian rates the severity level of these vulnerabilities as **medium**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, medium or low. These vulnerabilities are **not** critical.

Risk Assessment

We have identified and fixed two permission verification bugs which affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). These vulnerabilities allow users to view metadata for changesets and reviews, from repositories/projects that they do not have access to, via tooltips.

Vulnerability

The table below describes the FishEye/Crucible versions and the specific functionality affected by the vulnerabilities.

FishEye/Crucible Feature	Affected FishEye/Crucible Versions	Fixed Version	Issue Tracking
FishEye changeset tooltips	FishEye 2.4.6 to 2.5.6	FishEye 2.5.7	FE-3811
Crucible review tooltips	Crucible 2.4.6 to 2.5.6	Crucible 2.5.7	CRUC-5811

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can [disable anonymous access](#) to your instance. Logged-in users will still be able to view metadata that they do not have permission to view, but anonymous users will be prevented from accessing this information.

Fix

FishEye and Crucible 2.5.7 and later versions fix these two issues. View the issues linked above for information on earlier fix versions for each issue. For a full description of this release, see the [FishEye 2.5 changelog](#) and [Crucible 2.5 Changelog](#). You can download the latest version of FishEye and Crucible from the download centres ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

FishEye and Crucible Security Advisory 2011-05-16

This advisory announces a number of security vulnerabilities that we have found and fixed in recent versions of FishEye/Crucible. You need to upgrade your existing FishEye/Crucible installations to fix these vulnerabilities. Enterprise Hosted customers should request an upgrade by filing a ticket at <http://support.atlassian.com>. JIRA Studio is not vulnerable to any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerabilities listed in this advisory have been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit

them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [XSS Vulnerabilities](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)

XSS Vulnerabilities

Severity

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a number of cross-site scripting (XSS) vulnerabilities which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). XSS vulnerabilities potentially allow an attacker to embed their own JavaScript into a FishEye/Crucible page. You can read more about XSS attacks at cgisecurity.com, [The Web Application Security Consortium](http://TheWebApplicationSecurityConsortium.com) and other places on the web.

Vulnerability

The table below describes the FishEye/Crucible versions and the specific functionality affected by the XSS vulnerabilities.

FishEye/Crucible Feature	Affected FishEye/Crucible Versions	Issue Tracking
Crucible snippets	Crucible 2.4.5 to 2.5.0	CRUC-5734
Crucible author mapping	Crucible 2.4.5 to 2.5.0	CRUC-5735
Crucible changeset comments in search results	Crucible 2.3.0 to 2.5.0	CRUC-5736
Crucible comments search	Crucible 2.2.6 to 2.5.0	CRUC-5737
FishEye/Crucible dashboard - review activity	FishEye/Crucible 2.2.8 to 2.5.2	FE-3031
FishEye/Crucible reviews list	FishEye/Crucible 2.2.8 to 2.5.2	FE-3032

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable [public signup](#) to your instance until you have applied the upgrade. For even tighter control, you could restrict access to trusted [groups](#).

Fix

FishEye/Crucible 2.5.4 fixes all of these issues. View the issues linked above for information on earlier fix versions for each issue. For a full description of this release, see the [FishEye 2.5 changelog](#) and [Crucible 2.5 Changelog](#). You can download the latest version of FishEye/Crucible from the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

Our thanks to **Marian Ventuneac** of <http://www.ventuneac.net>, who reported [FE-3031](#) and [FE-3032](#). We [fully support the reporting of vulnerabilities](#) and we appreciate it when people work with us to identify and solve the problem.

FishEye and Crucible Security Advisory 2011-01-12

This advisory announces a number of security vulnerabilities that we have found and fixed in recent versions of FishEye/Crucible. You need to upgrade your existing FishEye/Crucible installations to fix these vulnerabilities. Enterprise Hosted customers should request an upgrade by filing a ticket at <http://support.atlassian.com>. JIRA Studio is not vulnerable to any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerabilities listed in this advisory have been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at <http://support.atlassian.com/>.

In this advisory:

- [XSS Vulnerabilities](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Administration password logged in debug log](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Review comment search returns comments that a user has no permission to view](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)
 - [Fix](#)
- [Anonymous global access exposes entire user list](#)
 - [Severity](#)
 - [Risk Assessment](#)
 - [Vulnerability](#)
 - [Risk Mitigation](#)

- [Fix](#)

XSS Vulnerabilities

Severity

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a number of cross-site scripting (XSS) vulnerabilities which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). XSS vulnerabilities potentially allow an attacker to embed their own JavaScript into a FishEye/Crucible page. You can read more about XSS attacks at [cgisecurity.com](#), [The Web Application Security Consortium](#) and other places on the web.

Vulnerability

The table below describes the FishEye/Crucible versions and the specific functionality affected by the XSS vulnerabilities.

FishEye/Crucible Feature	Affected FishEye/Crucible Versions	Issue Tracking
Crucible review – linked JIRA issue	Crucible 2.0.1 – 2.3.7	CRUC-5307
Crucible email reviews	Crucible 2.2.0 – 2.4.0	CRUC-5308
Crucible review reload	Crucible 2.2.0 – 2.4.2	CRUC-5309
Crucible edit review details screen	Crucible 2.2.0 – 2.3.7	CRUC-5345
FishEye repository configuration	FishEye 2.4.0	CRUC-5310
FishEye charts	FishEye 2.2.0 - 2.4.0	CRUC-5311
FishEye/Crucible code macro	FishEye/Crucible 2.2.0 – 2.4.0	CRUC-5312
FishEye/Crucible changeset page heading	FishEye/Crucible 2.3.2 – 2.4.0	CRUC-5313

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable [public signup](#) to your instance until you have applied the upgrade. For even tighter control, you could restrict access to trusted [groups](#).

Fix

FishEye/Crucible 2.4.4 fixes all of these issues. View the issues linked above for information on earlier fix versions for each issue. For a full description of this release, see the [FishEye 2.4 changelog](#) and [Crucible 2.4 Changelog](#). You can download the latest version of FishEye/Crucible from the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

Administration password logged in debug log

Severity

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a vulnerability in the FishEye/Crucible logging which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). This vulnerability allows administrator passwords to be logged in clear text when debug logging is enabled.

Vulnerability

This vulnerability affects FishEye and Crucible 2.2.0 to 2.4.0.

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable access logging. See [Enabling Access Logging in FishEye](#) and [Enabling Access Logging in Crucible](#). You can also apply file restrictions to your log files. Note, this issue only occurs when DEBUG logging is turned on (off by default) when an administrator logs in.

Fix

FishEye/Crucible 2.4.2 and later fix this issue. For a full description of this release, see the [FishEye 2.4 changelog](#) and [Crucible 2.4 Changelog](#). You can download the latest version of FishEye/Crucible from the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix this vulnerability. You must upgrade your FishEye/Crucible installation.

Review comment search returns comments that a user has no permission to view

Severity

Atlassian rates the severity level of these vulnerabilities as **medium**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a vulnerability in the Crucible review comment search which may affect Crucible instances, including publicly available instances (that is, internet-facing servers). This vulnerability allows review comments to be displayed for projects that are not publicly viewable.

Vulnerability

This vulnerability affects Crucible 2.2.0 to 2.4.3.

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Fix

FishEye/Crucible 2.2.5, 2.3.8 and 2.4.4 fix this issue. For a full description of this release, see the [FishEye 2.4 changelog](#) and [Crucible 2.4 Changelog](#). You can download these versions of FishEye/Crucible via the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix this vulnerability. You must upgrade your FishEye/Crucible installation.

Anonymous global access exposes entire user list

Severity

Atlassian rates the severity level of these vulnerabilities as **medium**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank the severity as critical, high, moderate or low.

Risk Assessment

We have identified and fixed a vulnerability in the FishEye/Crucible anonymous global access which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). This vulnerability exposes the user list (usernames and emails) of a FishEye/Crucible instance for access when anonymous global access is enabled.

Vulnerability

This vulnerability affects FishEye and Crucible 2.2.0 to 2.4.3.

Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable global anonymous access. See [Configuring Anonymous Access](#).

Fix

FishEye/Crucible 2.2.6, 2.3.8 and 2.4.4 fix this issue. For a full description of this release, see the [FishEye 2.4 changelog](#) and [Crucible 2.4 Changelog](#). You can download these versions of FishEye/Crucible via the download centre ([FishEye download centre](#), [Crucible download centre](#)).

There are no patches available to fix this vulnerability. You must upgrade your FishEye/Crucible installation.

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.

FishEye Feature	Affected FishEye Versions	Issue Tracking
Code Metrics Plugin	2.0.x to 2.3.6 inclusive	CRUC-4572
FishEye Revision ID Parameters on Annotated Views	2.3.0 to 2.3.6 inclusive	CRUC-4641

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.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.3.2.	2.3.3 update, also available as patches for 2.3.2 and 2.2.3.	This vulnerability allows a motivated attacker to call remote code on the host server.	Critical

This vulnerability has been discovered in XWork by OpenSymphony, a command pattern framework which is used by FishEye and Crucible.

About the XWork Framework:

- See the [OpenSymphony XWork page](#) for more information about XWork.

Risk Mitigation

We strongly recommend either upgrading or patching your FishEye/Crucible installation to fix this vulnerability. Please see the 'Fix' section below.

Fix

These issues have been fixed in FishEye 2.3.3 (see the [changelog](#)), which you can download from the [download centre](#).

It has also been fixed in Crucible 2.3.3 (see the [changelog](#)), which you can download from the [download centre](#).

Later versions will include protection from this vulnerability.

This fix is also provided as a patch for FishEye/Crucible 2.3.2 and 2.2.3, which you can download from [links on this page](#). Customers on earlier point versions of FishEye/Crucible will have to upgrade to version 2.3.2 or 2.2.3 before applying the patch. Atlassian recommends you upgrade to FishEye/Crucible 2.3.3.

Download Patches for Earlier FishEye / Crucible Versions

i These patch releases contain security fixes, which apply to the shared FishEye architecture that is the basis of both FishEye and Crucible.

Please note that these patches are for specific point versions of FishEye (2.3.2 and 2.2.3). If you are running an earlier version than these, you will need to upgrade to a version specifically addressed by one of these patches. Atlassian strongly recommends that you upgrade to [FishEye 2.3.3](#) / [Crucible 2.3.3](#) or later.

[MD5 checksums](#) are provided to allow verification of the downloaded files.

Patch for FishEye / Crucible 2.3.2

File	FishEye / Crucible Version	Release Date	MD5 Checksum

fisheye-crucible-2.3.2-patch1.zip	2.3.2	16th June, 2010	6fe98db821a6d26f26907688af2ccd84

Patch for FishEye / Crucible 2.2.3

File	FishEye / Crucible Version	Release Date	MD5 Checksum
fisheye-crucible-2.2.3-patch1.zip	2.2.3	16th June, 2010	6fe98db821a6d26f26907688af2ccd84

Our thanks to **Meder Kydraliev** 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.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.2.1	2.2.3 update, also available as patches for certain versions, listed on this page .	This vulnerability allows a motivated attacker to perform admin actions.	Critical

Risk Mitigation

We strongly recommend either upgrading or patching your FishEye installation to fix this vulnerability. Please see the 'Fix' section below.

i Note: If you are an Atlassian JIRA Studio customer, we have assessed that your system is secure and implemented additional protections for it.

Fix

These issues have been fixed in FishEye 2.2.3 (see the [changelog](#)), which you can download from the [download centre](#). Later versions will include protection from this vulnerability.

This fix is also provided as a patch for FishEye 2.1.4, 2.0.6 and 1.6.6, which you can download from [this page](#). Customers on earlier point versions of FishEye will have to upgrade to version 2.1.4, 2.0.6 or 1.6.6 before applying the patch. We recommend you upgrade to FishEye 2.2.3.

XSS Vulnerabilities in FishEye

Severity

Atlassian rates these vulnerabilities as **critical**, according to the scale published in [Severity Levels for Security Issues](#). The scale allows us to rank a vulnerability as critical, high, moderate or low.

Risk Assessment

We have identified and fixed several cross-site scripting (XSS) vulnerabilities in FishEye, which may affect

FishEye instances. These vulnerabilities have security implications and are especially important for anyone running publicly accessible instances of FishEye.

- The attacker might take advantage of the vulnerability to steal other users' session cookies or other credentials, by sending the credentials back to the attacker's own web server.
- The attacker's text and script might be displayed to other people viewing a FishEye page. This is potentially damaging to your company's reputation.

You can read more about XSS attacks at [cgisecurity](#), [CERT](#) and other places on the web.

Vulnerability

All versions of FishEye are affected by these XSS vulnerabilities.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.2.1	2.2.3 only	An attacker could take advantage of this vulnerability to steal other users' session cookies or other credentials, or the attacker's text and script might be displayed to other people viewing a FishEye page.	Critical

Risk Mitigation

We strongly recommend upgrading your FishEye installation to fix these vulnerabilities. Please see the 'Fix' section below.

Fix

These issues have been fixed in FishEye 2.2.3 (see the [changelog](#)), which you can download from the [download centre](#).

Prevention of Brute Force Attacks

Severity

Atlassian rates this vulnerability as **moderate**, according to the scale published in [Severity Levels for Security Issues](#).

Risk Assessment

We have improved the security of the following areas in FishEye:

- Prevention of brute force attacks by requiring users to solve a [CAPTCHA](#) test after a maximum number of repeated login attempts.

Vulnerability

We have identified and fixed a problem where FishEye allows an unlimited number of repeated login attempts, potentially opening FishEye to a brute force attack. Details of this improvement are summarised below.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.2.1	2.2.3 only	FishEye allows an unlimited number of login attempts. This makes FishEye vulnerable to a brute force attack.	Moderate

Risk Mitigation

We recommend that you upgrade your FishEye installation to fix these vulnerabilities. Please see the 'fix' section [below](#).

You can also prevent brute force attacks by following our guidelines on [using Fail2Ban to limit login attempts](#).

Fix

This issue has been fixed in FishEye 2.2.3 (see the [changelog](#)). Later versions will include protection from this vulnerability. You can download FishEye 2.2.3 from the [download centre](#).

Changed Behaviour in FishEye

In order to fix these issues, we have changed FishEye's behaviour as follows:

- After three consecutive failed login attempts, FishEye will display a [CAPTCHA](#) form asking the user to enter a given word when attempting to log in again. This will prevent brute force attacks via the login screen. The number of failed attempts needed to trigger the [CAPTCHA](#) testing is configurable. For more information, see the documentation for [Brute force login protection](#).

 In addition, after three consecutive failed login attempts via the FishEye remote API, an error message will be returned. Human intervention will then be required to reset that login account, i.e. solve the [CAPTCHA](#) test via the login screen.

Download Patches for Earlier FishEye / Crucible Versions

 These patch releases contain security fixes, which apply to the shared FishEye architecture that is the basis of both FishEye and Crucible.

These patches fix the Admin Escalation vulnerability only. Please note that these patches are for specific older point versions of FishEye (2.1.4, 2.0.6 or 1.6.6). If you are running an earlier version than these, you will need to upgrade to a version specifically addressed by one of these patches. To update a more recent version of the product (2.1.5 through 2.2.1), please upgrade to FishEye 2.2.3 or later. Atlassian strongly recommends that you [upgrade to FishEye 2.2.3](#) or later.

[MD5 checksums](#) are provided to allow verification of the downloaded files.

Patch for FishEye / Crucible 2.1.4

File	FishEye / Crucible Version	Release Date	MD5 Checksum
fisheye-crucible-2.1.4-patch1.zip	2.1.4	4th May, 2010	6062fa2e1ad93729527357fb97b0d2ea

Patch for FishEye / Crucible 2.0.6

File	FishEye / Crucible Version	Release Date	MD5 Checksum
fisheye-crucible-2.0.6-patch1.zip	2.0.6	4th May, 2010	6aae75e2a5308121887bf9532473cf75

Patch for FishEye 1.6.6

File	FishEye Version	Release Date	MD5 Checksum
fisheye-1.6.6-patch1.zip	1.6.6	4th May, 2010	210ef3358aff83861733f8f22d331d7e

Patch for Crucible 1.6.6

File	Crucible Version	Release Date	MD5 Checksum
crucible-1.6.6-patch1.zip	1.6.6	4th May, 2010	48e8e8ada0ddb3fc8671459051df1120

 To acquire all of the fixes on this page, upgrade to [FishEye 2.2.3](#), which you can download from the [download centre](#).

FishEye FAQ

[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?](#)

- [Example EyeQL Queries](#)
 - [How do I 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](#)

 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:

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.

Example EyeQL Queries

EyeQL

- [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?](#)

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:

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

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)

```
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
```

How do I show all changesets which do not have reviews?

The following query will return any changesets that have not been reviewed.

```
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?
 - ▼ Click here to expand...

A: There is a toggle to enable the API under "Server Settings" in the web admin interface. See [Configuring the FishEye web server](#) for more details.
- **Q:** Is there any way to return unique results from an EyeQL query?
 - ▼ Click here to expand...

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.asp>

[x](#) 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 ["tricked" 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](#).

 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.



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.
 -  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 [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](#).

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**.

 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.

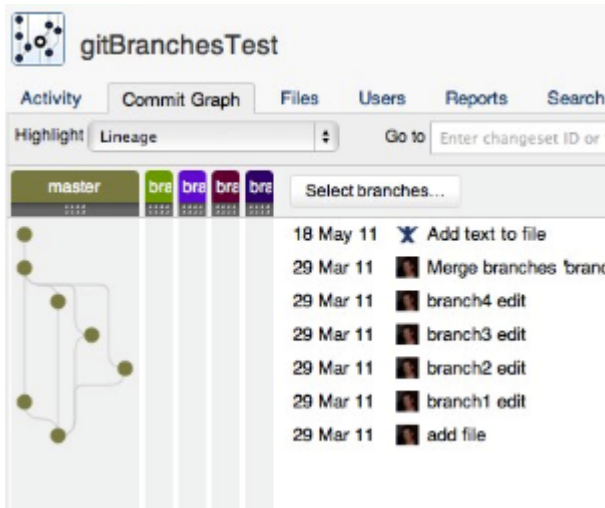
When you view the commit graph for a Git repository, FishEye works from the leftmost swimlane to the right doing the following:

- 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.

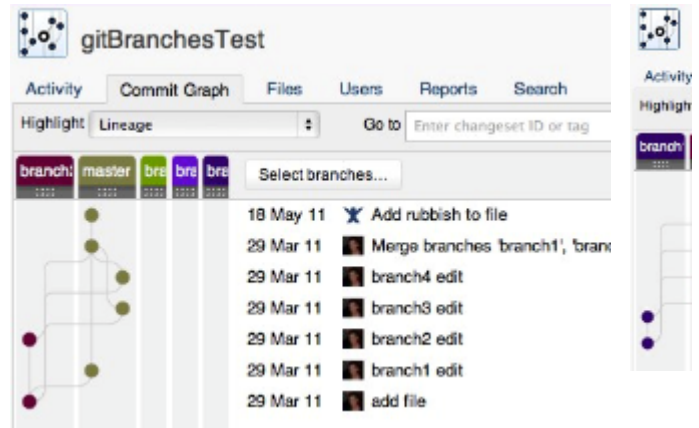
For example, if the 'master' swimlane is to the left of another swimlane, e.g. 'fisheye-2.6' branch, there will be no changesets shown in the 'fisheye-2.6' swimlane, as all the commits will be picked up in the 'master' swimlane. However, if you move the 'fisheye-2.6' swimlane to the left of the 'master' 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)



1



2

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.

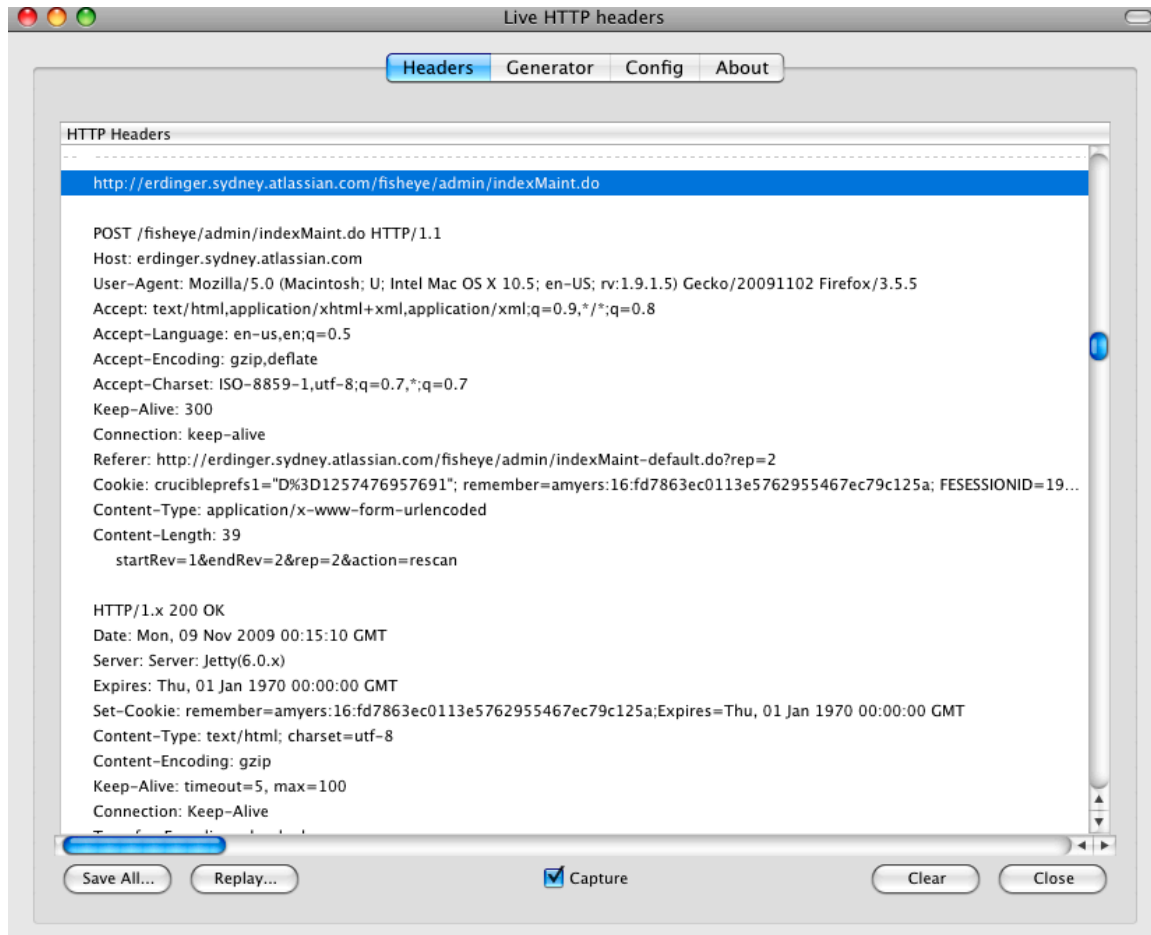
What SCM systems are supported by FishEye?

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&action=rescan)
4. Now we can construct a script with wget to automate this:

```

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
--post-data="startRev=0&endRev=58&rep=2&action=rescan"
http://erdinger.sydney.atlassian.com/fisheye/admin/indexMaint.do
  
```

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](#) — This page contains information about improving the performance of FishEye repository scans.
- [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 [svnsync based server](#).

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`
 - `<FishEye home directory>/cache`
- 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:

```
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.

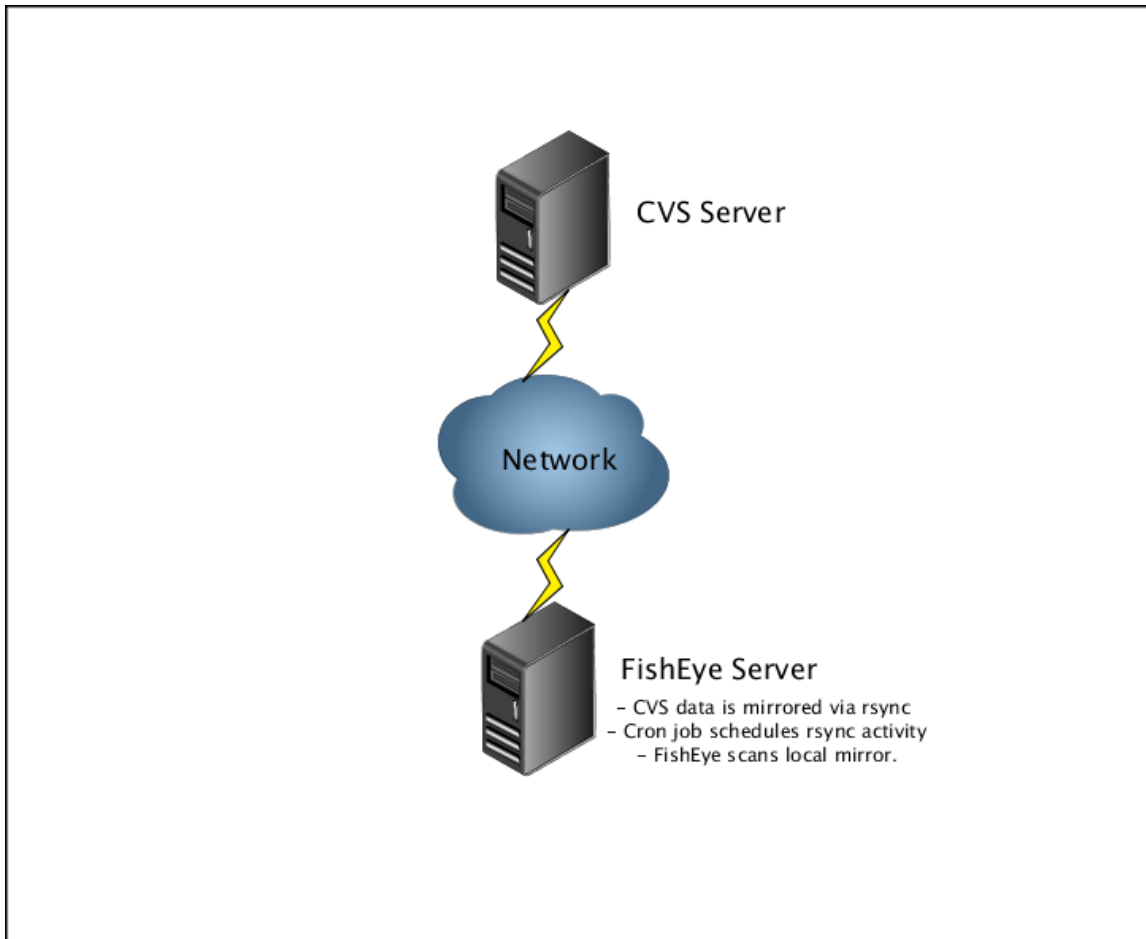
Setting up a CVS mirror with rsync

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 your CVS instance as CVS_HOME and your new 'mirror directory' on the FishEye server as MIRROR_HOME.
3. Type the following rsync command into the console of the FishEye server:

```
rsync --backup CVS_HOME MIRROR_HOME
```

A real-world example would look something like this:

```
rsync --backup \\julius\webapps\cvs\data \\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 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 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:

```
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:

Support

Information

New licenses can be acquired from my.atlassian.com.

Your Server ID is **BT5B-TJIC-GVD6-69C7**

Licenses

FishEye License

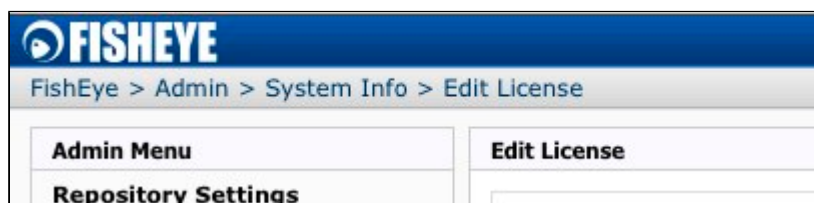
Crucible License

Click Update to save

Delete the Crucible license here

Update Cancel

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

i 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.jirafisheyepugin&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.jirafisheyepugin = DEBUG, console, filelog
log4j.additivity.com.atlassian.jirafisheyepugin = 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/atlassian

lassian/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*.

i If you associate reviews with JIRA issues from the Crucible user interface, they will not necessarily show up in the issue tab panel. We are working on this issue (see [FISH-316](#)), but in the meantime, a workaround is to include the related issue key in the title of the review.

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

FishEye Licensing FAQ

- [Are anonymous users counted towards FishEye's licence limits?](#) — Users accessing FishEye anonymously are, for all intents and purposes, unlimited users.
- [Restrictions on FishEye Starter Licenses](#)

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

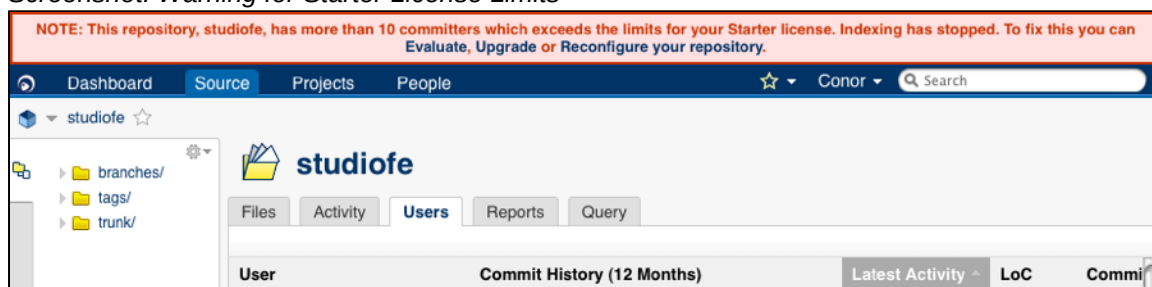
What happens if I exceed the Starter License limits?

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

Question	Answer
What happens when the 11th unique committer is encountered during indexing?	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.
What happens when a FishEye instance with a Starter license is started, using existing indexes with more than five repositories?	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.
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?	FishEye will display all information currently indexed but for each repository that has reached the ten committer limit, no further revisions will be indexed.
What happens on upgrade from a Starter license, if indexing has been paused due to the committer limit being reached?	On restart of FishEye, indexing will resume for all repositories. Each repository can be 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.
What happens when upgrading from a Starter license, when repositories have not started due to the repository limit being reached?	On restart of FishEye, all enabled repositories will start. Each repository can be restarted individually to avoid restarting FishEye.
What happens if my evaluation license has expired and I upgrade to a Starter license, however I have exceeded the Starter license limitations?	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.

What happens when downgrading to a Starter license, where the repository limit has been exceeded?	A maximum of five of your configured repositories will start running. The remainder will not start but will continue to be available.
What happens when downgrading to a Starter license, where the committer limit has been exceeded for one or more repositories?	No further indexing will occur for the repositories where the committer limit has been exceeded.

Repository Management FAQ

FishEye 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)

Setting the root location for internal repositories

i If you have already created internal repositories, you must **migrate them when you update the repositories root attribute** - see [Migrating repositories to a changed root location](#).

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.
2. Make a **backup copy of your `config.xml` file**.
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

i This describes how to migrate existing internal repositories when following the instructions at [Setting the root location for internal repositories](#).

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.

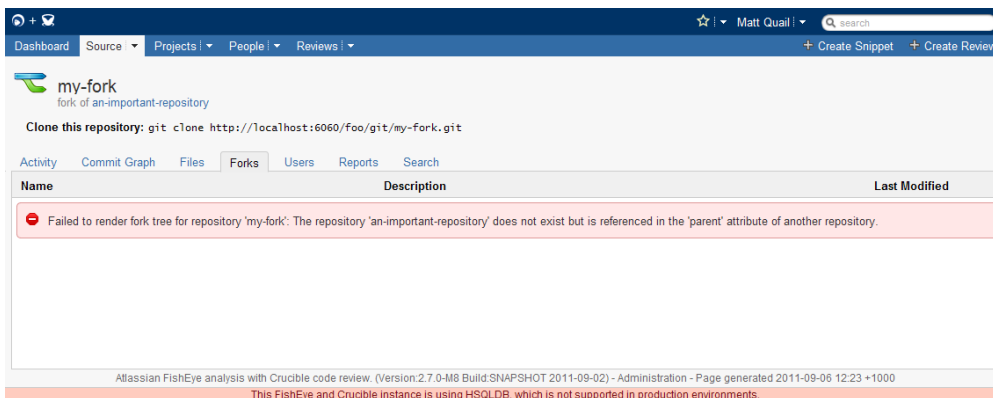
2. Copy all of the directories in your current repositories-root (by default, this is in the %FISHEYE_IN ST%/manged_repos directory) to the new location that you have specified.
3. Open your config.xml file for editing.
4. Update the location attribute of each <git> element that is the child of a managed <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...



...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.
2. Make a **backup copy of your config.xml file**.
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 authenticated using these credentials. Also, this method of authentication is potentially unsuitable if you are accessing your FishEye server via a proxy, as all curl requests that target a path on that proxy server will be authenticated using your `.netrc` credentials.

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/...`

Linux or OSX

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

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:

```
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/dir1` is copied to `/tags/tag1/dir1`.

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.

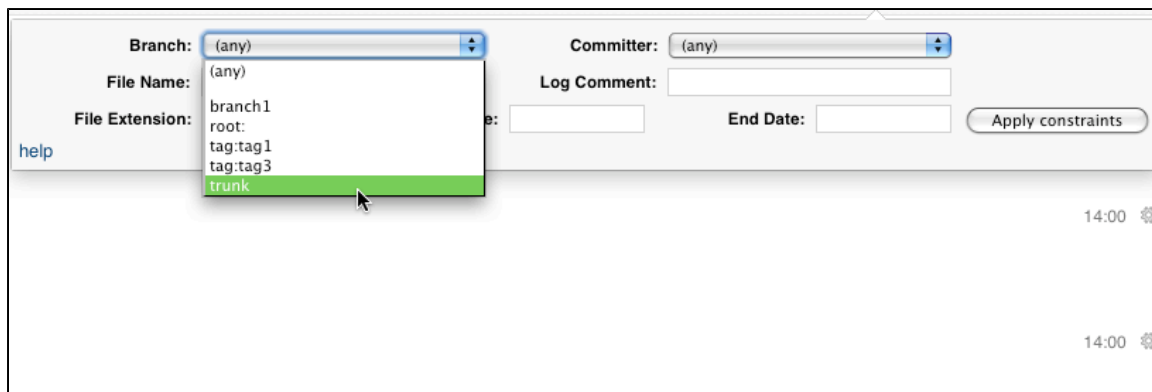
On this page:

- [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

i 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.

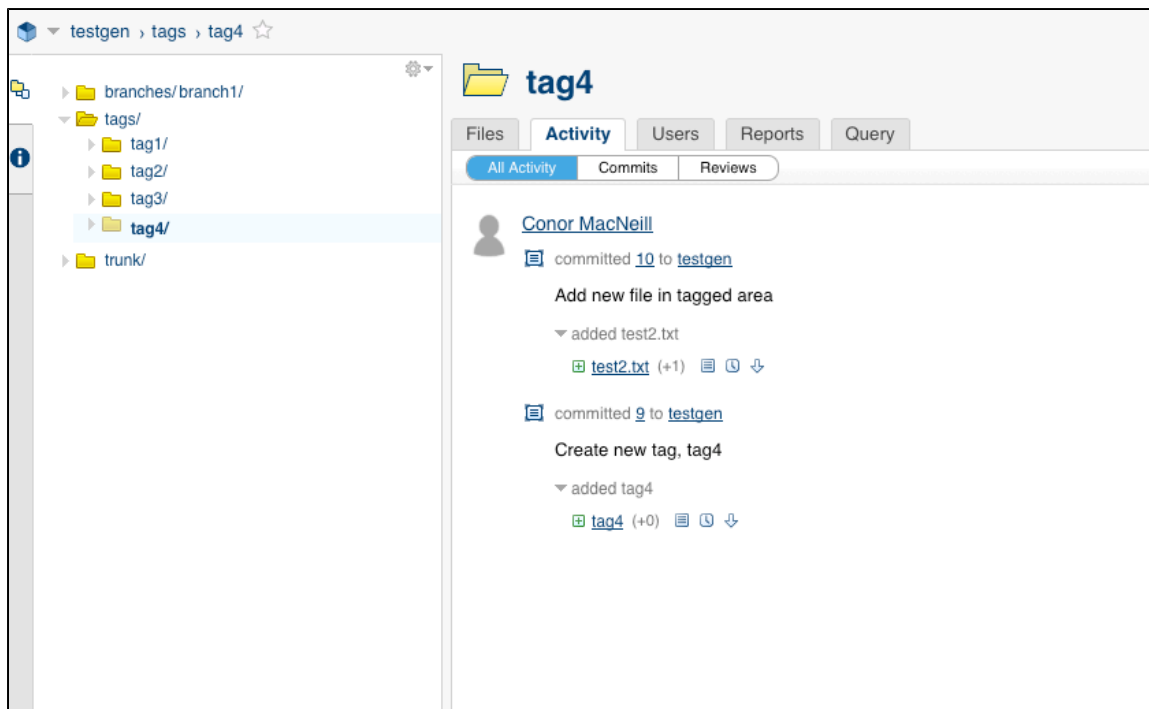
For regular changes in your Subversion repository, FishEye records each change against a branch where the change took place. If, however, after tagging, you make a change to a file in the tagged area, you are making a change outside trunk or a recognized branch. FishEye records such changes by creating 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

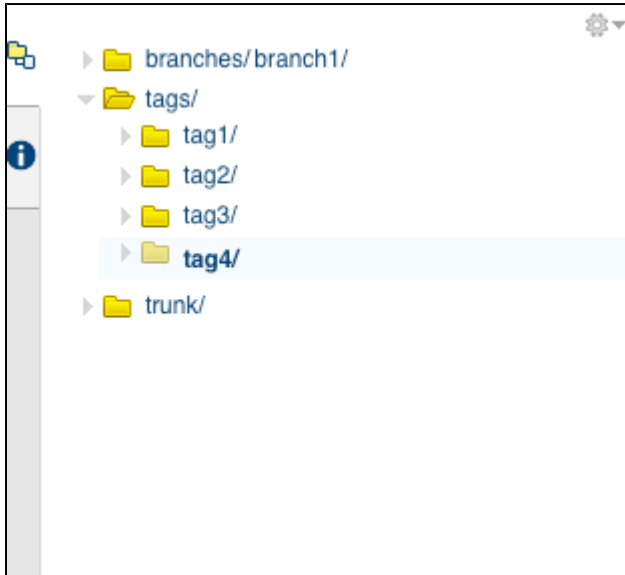
In general, it's not good practice to make changes in the tag areas of a Subversion repository. Such changes can easily get lost if they are not applied to trunk or a current branch. It is preferable to make the change in trunk or a branch and then create a new tag to capture the update. Nevertheless, 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



i If you want to constrain to a tag, enter the tag name in the tag field of the constraint filter.

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.

i 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

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)
- OR
- 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/CONE> 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 or RSS.

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 JIRA Studio, since JIRA Studio 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](#).

Visit our general [Atlassian 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:

CVSS score range	Severity in advisory
0 – 2.9	Low
3 – 5.9	Medium
6.0 – 7.9	High
8.0 – 10.0	Critical

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

FishEye Troubleshooting

- [After I commit a change to my CVS repository, it takes a long time before it appears in FishEye.](#) — 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](#) — 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.
- [Generating a Thread Dump Externally](#) — If FishEye stops responding or is showing poor performance, providing thread dumps to support can help diagnose the problem.
- [I have installed FishEye, and the initial scan is taking a long time. Is this normal?](#) — 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.
- [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.
- [Initial scan and page loads are slow on Subversion](#) — 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.
- [It seems that FishEye's HTTP Header is Too Small](#)
- [JIRA Integration Issues](#)
- [Message '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.
- [Problems with very long comments and MySQL migration](#) — There is a known issue with FishEye 2.0.x and very long comments when migrating your database to MySQL.
- [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.

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
2009-07-15 15:34:45,557 ERROR [btpool0-519] fisheye.app
HibernateUtil-commitTransaction - Commit failed rolling back.
...
...
Caused by: java.sql.BatchUpdateException: Lock wait timeout exceeded;
try restarting transaction
    at
com.mysql.jdbc.ServerPreparedStatement.executeBatch(ServerPreparedStatement.
java:647)
    at
com.mchange.v2.c3p0.impl.NewProxyPreparedStatement.executeBatch(NewProxy
PreparedStatement.java:1723)
    at
org.hibernate.jdbc.BatchingBatcher.doExecuteBatch(BatchingBatcher.java:4
8)
    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.

 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 Unixes

Find the process ID of the JVM and use the **ps** command to get list of all processes:

```
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:

```
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 1.0 Final can be obtained from the `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-deb`

ug.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:

```
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:

```
2009-07-16 16:56:12,390 ERROR [ThreadPool1] fisheye.app
com.cenqua.crucible.actions.admin.database.DBEditHelper-doGet -
Database migration failed:
java.sql.BatchUpdateException: Data truncation: Data too long for column
'cru_message' at row 1
java.sql.BatchUpdateException: Data truncation: Data too long for column
'cru_message' at row 1
```

 You may not see the message if you are running MySQL with default settings.

For more information, see the [JIRA issue](#).

Workaround

If your data contains very long comments or review descriptions (longer than 21,845 multibyte unicode characters), consider avoiding use of MySQL until 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](#) 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:

```
AllowEncodedSlashes On
```

This directive is documented [here](#).

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

- [Atlassian Support](#)
- [Support Policies](#)

Forums

- [FishEye Forum](#)
- [FishEye Developers Forum](#)

Mailing Lists

- Visit <http://my.atlassian.com> to sign up for mailing lists relating to Atlassian products, such as technical alerts, product announcements and developer updates.

Feature Requests

- [Issue Tracker and Feature Requests for FishEye](#)

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.

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.

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](#)

None

No guides yet

We do not yet have any guides to link here. Be the first to suggest one!

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](#)