# FISHEYE DOCUMENTATION

# .bookmarks

**FishEye 1.4 Bookmarks**
This page is a container for all the bookmarks in this space. Do not delete or move it or you will lose all your bookmarks.
Bookmarks in FishEye 1.4 | Links for FishEye 1.4

## The 15 most recent bookmarks in FishEye 2.6

There are no bookmarks to display.

# FishEye Documentation Home

## FishEye 2.6.x

### User's Guide

The FishEye User's Guide is for developers, project managers, testers – anyone who uses FishEye. New to FishEye? Start by exploring the FishEye screens and configuring repositories, viewing the changelog and creating charts.

### Administrator's Guide

The FishEye Administrator's Guide is for people with FishEye administration rights. It will help you set up users and groups, email notifications, and JIRA integration. You may want to customise the look and feel of FishEye, and tune it for best performance. Admin tasks such as backup are also covered. You may also find the Knowledge Base, FAQ and FishEye Forum useful.

### Installation Guide

The FishEye Installation Guide is for people who are installing FishEye for the first time. Check the supported platforms, then download and install FishEye. Where to next? The FishEye 101 will help you get started. If you are using other Atlassian products, take a look at the Integration Guide.

### Upgrade Guide

The FishEye Upgrade Guide is for people who are upgrading their instance of FishEye. Start by reading the latest Release Notes and version-specific Upgrade Guide for the version to which you are upgrading, then download FishEye and follow the main Upgrade Guide.

### Developer Resources

These resources are for software developers who want to create their own plugins for FishEye. Take a look at the Development Hub and the API Documentation. You may also find the FishEye Developers Forum useful.

## FishEye 101

Welcome to FishEye 101, an introductory guide to FishEye and a tour of the most interesting FishEye features. Use this page to guide your evaluation process or quickly get up to speed with FishEye.

| FishEye 101 |
| --- |
| Thanks for taking the time to try FishEye. To help you make the most of your time, we've taken the liberty of compiling some easy instructions for configuring and using FishEye. |

Software developers are the intended audience for this document.

## Getting FishEye up and running

Want to get up and running quickly? Follow these simple steps and you should have FishEye ready to go in no time. **Setting up FishEye can take less than half an hour.**

### Install

First things first, if you haven't already got FishEye up and running carry out the relevant steps below:

For Windows: (click to expand)

Check Supported Platforms before you begin.

1. Download FishEye from the Atlassian Download Center
2. Unzip the downloaded package to the desired installation directory: `c:\fisheye` (see the documentation if you need help with this).
3. Launch FishEye from the command line, typing `bin\start.bat`. A FishEye console window will appear, listing the startup progress. Give it a few minutes to complete its process. When the message "server started" is displayed, you can continue.

   You will also receive a warning in the console output about a 'missing license'. This is normal.
4. On the same machine go to `http://localhost:8060/` (or type the hostname or IP address in place of `localhost` from another machine) in your web browser. You will be prompted to enter a license key (obtained at http://my.atlassian.com) and set the admin password. (Tip: you can just use one Crucible evaluation license key to try both FishEye & Crucible.)

It should only take you a few minutes to have a running instance of FishEye. If you have any difficulty during the installation process (i.e. it takes longer than twenty minutes) please contact our support team for assistance. They're ready to help at any time.

For Linux and Mac OS X: (click to expand)

Check Supported Platforms before you begin.

1. Download FishEye from the Atlassian Download Center
2. Unzip the downloaded package to the desired installation directory: `/opt/fecru-x.x.x` (see the documentation if you need help with this).
3. Launch FishEye from the command line, typing `./bin/run.sh` . Console output will appear, listing the startup progress. Give it a few moments to complete its process. When the message "server started" is displayed, you can continue.

   You will also receive a warning in the console output about a 'missing license'. This is normal.
4. On the same machine go to `http://localhost:8060/` (or type the hostname or IP address in place of `localhost` from another machine) in your web browser. You will be prompted to enter a license key (obtained at http://my.atlassian.com) and set the admin password. (Tip: you can just use one Crucible evaluation license key to try both FishEye & Crucible.)

It should only take you a few minutes to have a running instance of FishEye. If you have any difficulty during the installation process (i.e. it takes longer than twenty minutes) please contact our support team for assistance. They're ready to help at any time.

### Scan

FishEye's fast performance is built on the way it scans repositories. (click to expand)

Once installed, FishEye needs to do an initial index, where it accesses, indexes and organizes a view of your repository (including all historical items) back to the earliest commit.

FishEye operates in a completely read-only mode, so your repository data is safe. Also, always run FishEye on an SCM account that is limited to read-only access.

Click on the '**Administration**' link in the footer bar to access the administration interface, then click '**Add Repository**'.

There are lots of options when adding a repository, but in most situations you should be able to run with the defaults. For evaluation purposes, we recommend that you just index one project, which is usually a subset of your whole repository (rather than everything). If you choose to index your entire repository, just 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). Indexing just one project, you can have FishEye up and running quickly for evaluation. The basic process is slightly different for each SCM type. Carry out the respective process for your SCM below:

- Subversion set the *Path* field to a subdirectory that includes the directories trunk, branches, tags for your project
- Perforce & Git set the *Path* to a subdirectory that contains your project's source.
- CVS set the *CVS dir* to point to your project.

The default indexing settings can have challenges digesting non-standard SCM configurations. If the initial index appears to hang, please call support as this usually indicates you have a configuration problem, that can be quickly resolved by our friendly support engineers.

> Check out the best practices guide for your FishEye configuration if you have a large repository or are having difficulty with setup.

### Add Users

You can automatically import users from an LDAP directory or similar. (click to expand)

FishEye was designed to improve team communication through sharing SCM artifacts, so you will probably want to add accounts for all your team members.  There are a few approaches that you can take to adding users:

1. Leave FishEye in anonymous mode. There are two downsides here, anyone who can see FishEye, can see all the source and you won't be able to use email watches or save preferences.
2. Built-in users with "public sign up". Anyone can create an account, but not especially secure.
3. Built-in users. This is simple to set up for a small user base, however adding each user manually is not recommended for large-scale systems with many users.
4. External authentication. This includes sources such as LDAP, Active Directory or similar.
    a. Atlassian Crowd. If you already have a Crowd server, this is a snap to set up.
    b. LDAP/Active Directory. You can point FishEye at your existing corporate directory. If you are familiar with LDAP, this should be easy.
    c. Host/PAM Authentication. You can use the local accounts from the physical machine FishEye is running on.

If you opt for accounts, be sure to make yourself and other relevant users FishEye administrators. You may also be interested in pre-loading users from LDAP/Crowd and setting repository level access control based on groups.

### Configure Mail (optional)

Email notifications are a great way to keep up with FishEye activity. (click to expand)

FishEye can deliver email feeds based on user configured parameters. For example, you can watch the contributions of a specific colleague or changes to a branch of code you're working on. To take advantage of this feature, you will need to tell FishEye about your SMTP server.

### Configure JIRA (optional)

FishEye integrates with JIRA, Atlassian's enterprise issue tracker. (click to expand)

1. If you already have a JIRA server set up, you can go ahead and set up integration between the two. Once established, activity on relevant JIRA issues will appear in the FishEye activity stream. JIRA issue keys will also be hyperlinked, also a small information window will load when you mouse-over JIRA issue keys.
2. Visit the FishEye documentation for instructions on how to set up JIRA integration in FishEye.

### Setup Complete!

Your FishEye instance is now established. (click to expand)

Congratulations! Our support records indicate that over 80% of support calls happen during this installation phase. Once you have made it this far, the rest of the evaluation ought to flow smoothly.

> If you do need assistance, our support team is ready to help. Create an issue at our support system under the FishEye project.

There are heaps of tweaks and configuration options that you can experiment with. Check out the FishEye Administrator's Guide for more information.

## Learning the Basics

FishEye is a tool that lets you view the contents of your Source Code Management (SCM) repository as a web page. In this section you will learn more about understanding the FishEye UI, search options, and notifications.

### Browsing your Code Repository

#### Browsing & History

- Power-browse your repository. (click to expand)

Fisheye's Windows Explorer-style interface allows you to efficiently navigate your source tree and view the depths of your repository in a structured way. Select a file to view its entire revision history. You also have one click access to statistics, line history, commit volume, and much more. See the documentation for more.

> Everything in FishEye can be bookmarked. See the documentation for more.

#### Full Source View

- View the full source of any file revision. (click to expand)

The 'Source' tab shows the blame information associated with every line (allowing you to pinpoint who is responsible for every change). Author and age can be displayed as an aid to annotation. See the documentation for more.

#### Changesets & Diffs

- See every change and collaborate anywhere on the web. (click to expand)

Every commit and any diff can be viewed and easily linked to. This removes ambiguity from discussions in other mediums (such as discussions held via instant messaging, wiki pages, email, issue trackers and so on). Try the side by side diff view to which gives you two panes that are synchronised horizontally and vertically, colour markers clearly show the relationship, and you can step through changes at your chosen amount of context. See the documentation for more.

#### Filtering the changes made to the source code

- Quickly find what you're looking for. (click to expand)

The 'Activity' tab displays the changes made to the repository/branch/directory/file. Find the changes that you're seeking by filtering commits based on log message, path, author, date, branch (and other fields). This control is a snappy filter button under the Source 'Activity' tab. See the documentation for more.

### Activity and People

#### Activity Streams

- You can see commits and updates from the users in FishEye rolling by. (click to expand)

This information appears as a stream on the Dashboard and other index pages, sorted chronologically showing you the latest changes. Updates can be viewed as an inline stream or RSS feed. See the documentation for more.

### Your Personal Dashboard

- See your own work at a glance and a stream of work items that are relevant to you. (click to expand)

  Click the 'Dashboard' tab to see a stream of all your own activity; your personal code commits; your reviews (if you are using Crucible) and your tracked issue updates (if you are using JIRA). See the documentation for more.

### People Lists

- In FishEye, you can view useful updates and statistics from your team. (click to expand)

  On the People index page, you can see the commit history in global lines of code (LOC) that each person has contributed (expressed as a line graph) and their total number of commits. Also, the most recent piece of activity is shown as a clickable item. See the documentation for more.

### People Pages

- Each person who makes code changes has a page. (click to expand)

  You can click on a person's name to see detailed information about their additions to the repository, showing details of their work and summaries of their activity. Additionally, you can see their work on tracked issues and code reviews if using FishEye with Crucible and JIRA integration is set up. See the documentation for more.

### Subscribe to SCM Updates

- Keep track of FishEye activity when your FishEye session is closed. (click to expand)

  Every user can keep an eye on changes from RSS Feeds and Email Watches to the source directories that interest them or even individual files. The idea is to encourage people to subscribe to the level of notifications that suits them so the signal isn't lost in the noise. Here are two ways to test this out:

  1. Set up a customised feed by going navigating to the 'Tools' menu in the upper right hand corner. Fill in the feed attributes and use the constraints to get data about a specific users contributions to a specific branch of code. Select to subscribe either as an RSS feed or an email watch. For fun, you can comment on their code the moment they commit until they can't take it anymore.

  2. Simulate a spontaneous watch request by surfing through the directories. Once you find an interesting contribution, hit the RSS button on in the 'Tools' menu to get updated on that particular file.

  See the documentation for more.

## Sharing your source

### Send Links to Your Code

- Everything in FishEye can be linked. (click to expand)

  A changeset, a diff, even a specific line in a specific revision can be linked. You'll find a lot of stuff is underlined. Keep this in mind as you evaluate FishEye, there are plenty of links that are crying out to be copied to clipboard and sent on to your teammates.

  If you look at the URL structure, you'll see that FishEye URLs are very predictable and hence can be easily generated by hand. There are also some extra wrinkles that are great for bookmarks.

  Useful Hacks:

  - Every line in an annotation is a permalink to that line, clicking the revision number in the gutter takes you to that line in the revision it was last changed (even when the line number is different). Add a post commit hook in your SCM that gives developers the FishEye link as soon as they commit.
  - Using the keyword HEAD will give you the latest version of file.
  - Use branch or tag names in place of revision numbers in diffs.
    See the documentation for more.

## Search

### Quick Nav

- When you type in the search box, matches are instantly shown below. (click to expand)

You can quickly find what you are looking for by typing one word or part of the name of what you are looking for. FishEye's Quick Nav feature will immediately show matches and suggestions below, before you've even pressed Enter or activated a proper search. Try typing a CSID to go directly to that page. See the documentation for more.

### Quick Search

- FishEye supports some powerful search commands. (click to expand)

The box in the top right corner of every page can be used to quickly search the repository. Start typing and a list of suggestions will pop up to help point you in the right direction. You can also use syntax like `author:anna` to immediately return results which have "anna" in the author field.

See the documentation for more.

### Running Queries & EyeQL

- You can create complex queries with FishEye's own query language. (click to expand)

Clicking 'Query' on the navigation bar opens the Simple Search screen. Here, you have access to a wide range of powerful searching functionality including file content searching, and grouping results by changeset, revision, file or directory.

Sometimes, you need data from your repository retrieved and sorted in ways that are unique to your own situation, beyond what the pre-fab GUI can give you. From the Simple Search screen, click 'Switch to EyeQL Search'. Here, you can build searches using FishEye's powerful embedded query language, called EyeQL.

See the search documentation and EyeQL Reference Guide for more.

## Advanced Features

FishEye will allow you to learn more about your code beyond just reading the source. Produce useful reports and charts, integrate with other systems, and much more.

## Reports and Charts

### Embedded Charts

- FishEye now shows charts in various locations. (click to expand)

Throughout the user interface, there are visualisations to help you view the activity in this area. The ubiquitous chart in FishEye plots net lines of code (that's just lines not NLOC for those who are interested) on the top and commit volume (the number of files committed) on the bottom. Note that this chart is generated on the fly based on what you are looking at, from the root down to individual files. On the changelog page the green bar shows you the period that is covered by the changesets on the page you are viewing. Clicking on the chart takes you to that point in time in your changelog.

See the documentation for more.

### Custom Charts

- You can define specific metrics to be charted within FishEye. (click to expand)

To start using the more complex charts, click on the Chart tab on the right of the header, or "jump to detailed chart" on any of the baby charts. There is a lot of really cool stuff in these charts.

Some things to try:

- Find out who your biggest contributors are - select "Show by Author" as a Pie chart.
- See what file types you have committed - select "Show by Extension" then select your user ID as in the Author field.
- See who has been changing the most lines in the last month - clear your selections and select "Show by Author" and a date a month prior in the "Start Date" field. If you have a lot of source the chart will probably be flat. If so, change the chart type to "Change", this will zero the line count at the start date.
- Compare your changes against another person - pick yourself and another author (hold the 'control' key to select multiple) in the author list. This isn't very useful but can be a bit of fun!

Note that the committer gets a line for each line they add or change. The previous owner of a modified or deleted line loses one. Currently any change, even whitespace, counts.

See the documentation for more.

### Plugin Reports

- FishEye is extensible, allowing you to create your own reports. (click to expand)

  You can create your own report as a plugin and have it appear in the menu, under the '**Source**' tab, and the '**Reports**' sub-tab. For example, the '**Code Metrics**' option that appears in the user interface is actually a plugin, added in to FishEye. You can use this plugin as a basis to create your own custom reports and add them in to FishEye. See the FishEye developer documentation for more.

### Leveraging CSV and the REST API

- You can export data and extend FishEye programmatically. (click to expand)

  When you need to extract repository data for a report, you can export the results of FishEye searches to a CSV (comma-separated values) file which imports directly into spreadsheets and other office applications. Simply select the radio buttons at the bottom of the Simple Search dialog and the CSV file will be generated automatically, prompting you to save the file.

  With its own API (Application Programming Interface), FishEye is extensible. If you need FishEye to do more than it ships with and you've got programming chops of your own, the API allows you to build extensions to the application to suit your needs. See the API documentation for more.

## Integrate with other systems

### Integrate with JIRA issue tracker

- Link your source code to your JIRA issues

  JIRA is Atlassian's issue tracking product - issues are part of software development for every team and can be tracked and managed in JIRA.

  Linking your JIRA instance to FishEye adds a source tab to every JIRA issue and browse project screen. Conversely, FishEye activity streams will show JIRA activity for linked projects and JIRA information is visible and linkable in FishEye source details.

  View the documentation here.

  > Not using JIRA and still want to link FishEye to your issue tracker - it is possible using JIRA linkers. Learn more.

### Integrate with your favorite IDE

- Acess FishEye information from Eclipse, IntelliJ, and Visual Studio

  The Atlassian Connector for Eclipse, IntelliJ, and Visual Studio links the FishEye web interface directly to your editor. One click and you can open a file in FishEye to get detailed information about the source you are looking at. Also, use the handy 'copy to clipboard' option to quickly share references to your file with others.

  Instructions on the IDE Connector can be found here.

## Advanced Features to Try

### Advanced Configuration Options

- FishEye has bucketloads of configuration options. (click to expand)

  The FishEye developers try and set sensible defaults so that everything "just works" in most situations. However, if you are a power user, there are lots of knobs to twiddle. Check out the documentation below to learn about the nitty gritty.

  - Customize the welcome page
  - Enable tarballs
  - Proxy FishEye through a separate web server
  - Add more indexing threads
  - Throttle repository access
  - Software update notifications
  - ViewVC compatibility

Thanks for taking the time to evaluate FishEye using this guide. To help continue your journey, our support staff are always ready to answer your questions in the FishEye Forum, or solve specific problems at our support portal http://support.atlassian.com.

# About FishEye

Your source code repository contains so much useful information, but it is not always easy to extract, interpret or keep up to date.

## How FishEye can Help

FishEye opens up your repository, helping you to understand your changing source code:

- Track changes to your own, your team's, or everyone's source code.
- Choose to be notified by email and/or RSS feeds.
- View the configurable changelog.
- Use the powerful search functionality
- Construct your own sophisticated queries with EyeQL and integrate the results with other tools via the FishEye API.
- Link to any artifact in your repository: commits, diffs, directories, file histories, revisions, source lines, and search results.
- Analyse your repository via:
    - Line graphs at every node from root to revision.
    - Showing 'Related Revisions', a list of modifications from all branches, sorted by revision number.
    - File annotations for age and ownership.

## Starting Points

For an overview of FishEye's features, take the Feature Tour.

If you are installing FishEye for the first time, read the Quick Start Guide.

For FishEye troubleshooting information, see the FAQ.

## What's New in FishEye?

See the FishEye Release Notes.

## Known Limitations

- 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.
- When indexing the content of files, FishEye has an internal limit on the number of tokens/words in the file it can index. Any text past the one-millionth token/word in a file is ignored.

## System Requirements

See the Supported Platforms.

## Why is it called FishEye?

A fish eye has a wide viewing angle, allowing it to see many things at once. This is a metaphor for how FishEye allows you to easily view the complexity of your source control repositories.

# FishEye User's Guide

# FishEye Quick Start Guide

This guide will explain how to get FishEye installed and running as easily as possible. For more advanced installation topics, see the Installation Guide.

### Step 1. Install FishEye

1. Download the FishEye zip file and extract it. This document assumes you have extracted FishEye to `/FISHEYE_HOME/`.
2. Ensure you have installed an appropriate Java runtime - see Supported Platforms. Ensure that `java` is in the PATH, or that the `JAVA_HOME` environment variable is set.
3. If you intend to use FishEye with Subversion, please be sure to read about the Supported Platforms, Subversion client setup, and granting permission to FishEye to scan your repository.
4. If you intend to use FishEye with Perforce, please ensure you read about the Supported Platforms and Perforce client setup.

### Step 2. Run FishEye

1. You can start FishEye immediately with the following:
    - For Unix-based systems:

    ```
    $ cd /FISHEYE_HOME/bin
    $ ./run.sh
    ```

    - For Windows-based systems:

    ```
    C:\> cd FISHEYE_HOME\bin
    C:\FISHEYE_HOME\bin> run.bat
    ```

2. Once started, FishEye will run its own HTTP web server on port 8060. You can access FishEye immediately by going to `http://HOSTNAME:8060/` in a browser.

> **Default ports**
> By default, FishEye will listen on port 8060 for HTTP requests. It also listens on 127.0.0.1:8059 as a control port. You can configure both of these in the FishEye Administration pages or by editing `/FISHEYE_HOME/config.xml` and restarting FishEye.

> **Read-only access for FishEye**
> We recommend you run FishEye as a user that has only **read** access to your repository.
>
> An exception to this rule is users running the JIRA FishEye plugin with Perforce Job Integration. In that scenario, you must give FishEye write access.

### Step 3. Set up FishEye

1. The first time you access FishEye from a browser, you will be asked to enter an administrator password. This password will give you access to the FishEye Administration pages.
2. You will also be prompted for a trial license, which you can find here.
3. Once you have set up an administrator password, you can access the Administration pages at `http://HOSTNAME:8060/admin/`.
4. One of the first steps will be to add a repository.

### Step 4. Use FishEye

1. Once you have added a repository, you can view it in FishEye at `http://HOSTNAME:8060/`.
2. FishEye needs to build an index and cache of the contents of your repository, so some information will not appear in FishEye until this is complete.

### Stopping FishEye

To stop the FishEye server:

- For Unix-based systems:

```
$ cd /FISHEYE_HOME/bin
$ ./stop.sh
```

- For Windows-based systems:

```
C:\> cd FISHEYE_HOME\bin
C:\FISHEYE_HOME\bin> stop.bat
```

Want a hands-on tour of the best FishEye features? See the FishEye 101 page.

# 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 commiters 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 ).
- 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*

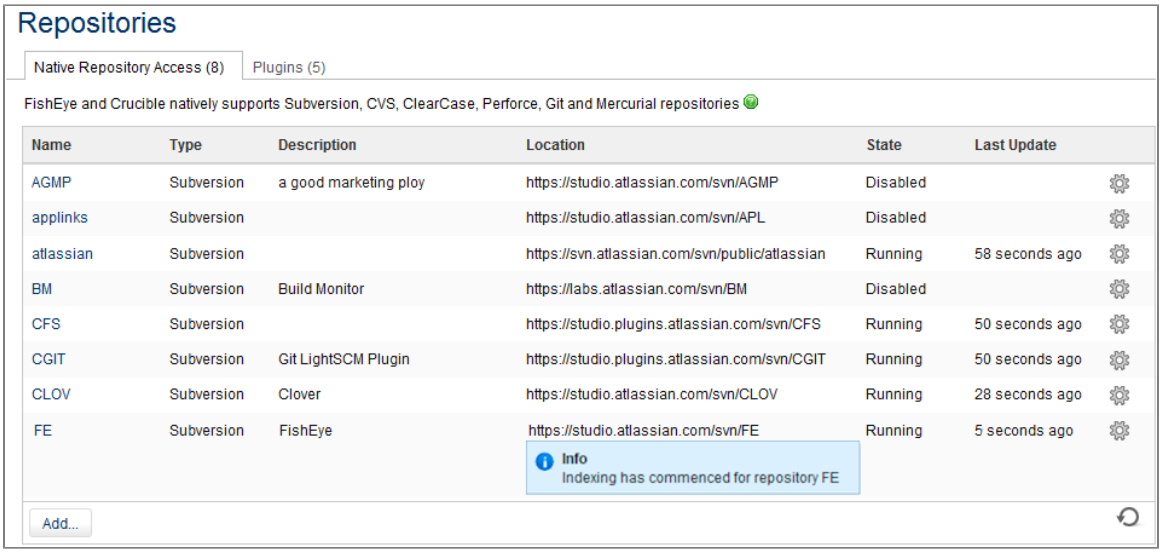
When you reach a source file, a summary page is shown, displaying recent revisions to the file.

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

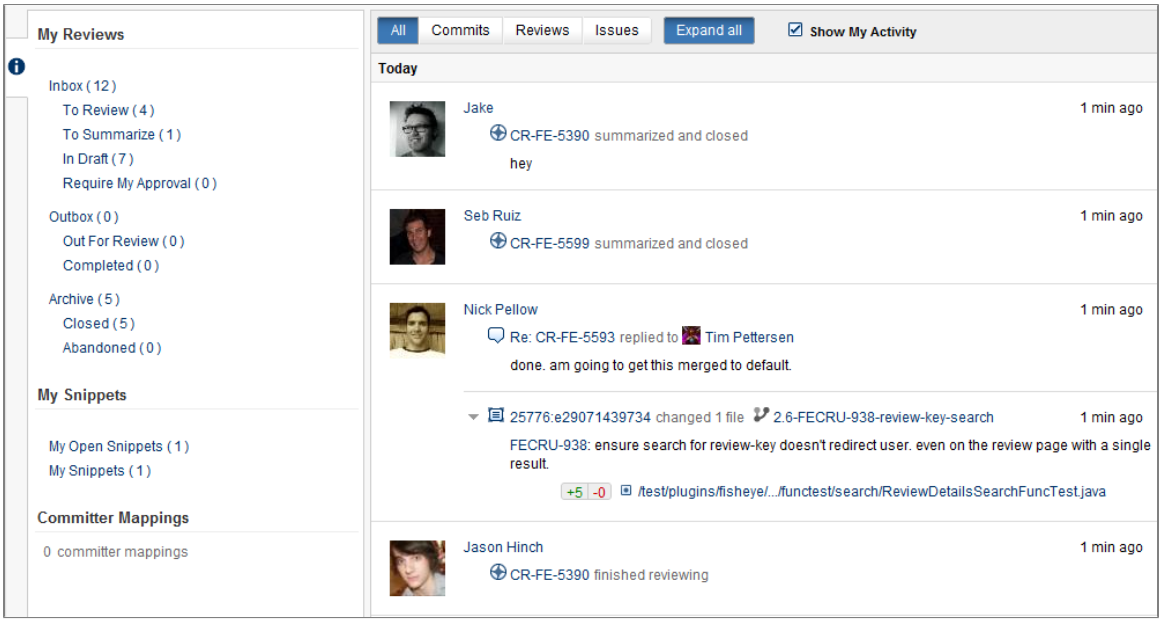
| Sub-Tab Name | Description |
|---|---|

| | |
|---|---|
| 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):<br>• All Activity — The default view, showing commits, reviews[1] and JIRA issues[2].<br>• Commits — Shows commits in the activity stream.<br>• Reviews[1] — Shows review activity in the activity stream.<br>• Scroll to Changeset — Opens the changeset ID specified in the text field (press Enter to carry out the action).<br>• Filter — Applies constraints to the current activity stream.<br>• Show Revisions — If this is selected, then changeset items are automatically expanded to show modified files.<br>• Earlier Activity (Left Arrow icon) — Loads a page of earlier activity.<br>• Later Activity (Right Arrow icon) — Loads a page of later activity.<br>[1] If you are using Crucible<br>[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. |

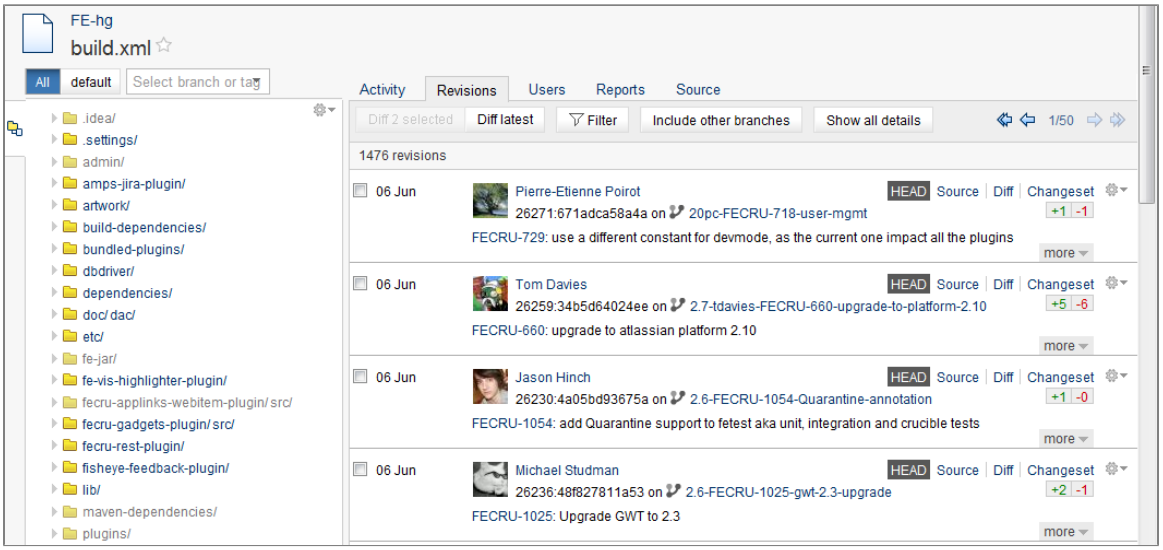
*Screenshot: The Repositories View*




*Screenshot: The Activity View*




*Screenshot: Repository Sub-Tabs — Viewing a File*




# Browsing through a Repository

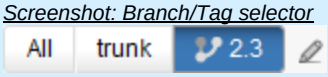
You can use FishEye to select a repository and browse through it. The repository view provides you with information about the files in the repository, activity occurring related to the repository and users committing to the repository. You can also generate charts and search for specific file revisions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

**On this page:**

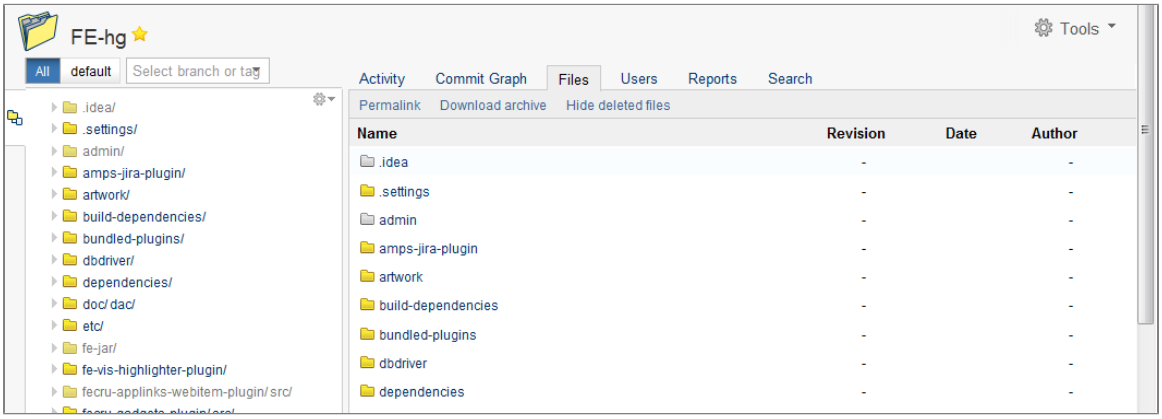


## Browsing through a Repository

**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 (non-ClearCase repositories only). 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.
     *Screenshot: Branch/Tag selector*
     
     
4. You can view various information about the repository, as outlined below. If you navigate to a folder, the context of the information below will change. For example, if you navigate to a particular folder in the left navigation tree, the activity, files and users information, reports generated and search results will all relate to that folder.
   *A greyed out item is either a deleted file or an empty/deleted (folder).*
   - '**Activity**' tab — View the commit, review and issue (requires JIRA) activity related to the repository/folder. The activity stream is similar to the changelog activity stream, see Viewing the Changelog for more information.
   - '**Commit Graph**' tab — Visualise the repository, using the commit graph. See Viewing the Commit Graph for a Repository for more information.
   - '**Files**' tab — View the contents of the repository/folder being viewed.
   - '**Users**' tab — View the commit history of the users that have committed changes to files in the repository/folder. See Viewing People's Statistics for more information.
   - '**Reports**' — View activity charts for the repository/folder. See FishEye Charts for more information.
   - '**Search**' — Search the repository/folder. See Searching FishEye for more information.
5. Click any file, when browsing the repository, to view information about the file. See Viewing a File for more information.

*Screenshot: Browsing a repository*




## Hiding Empty Directories and Deleted Files

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

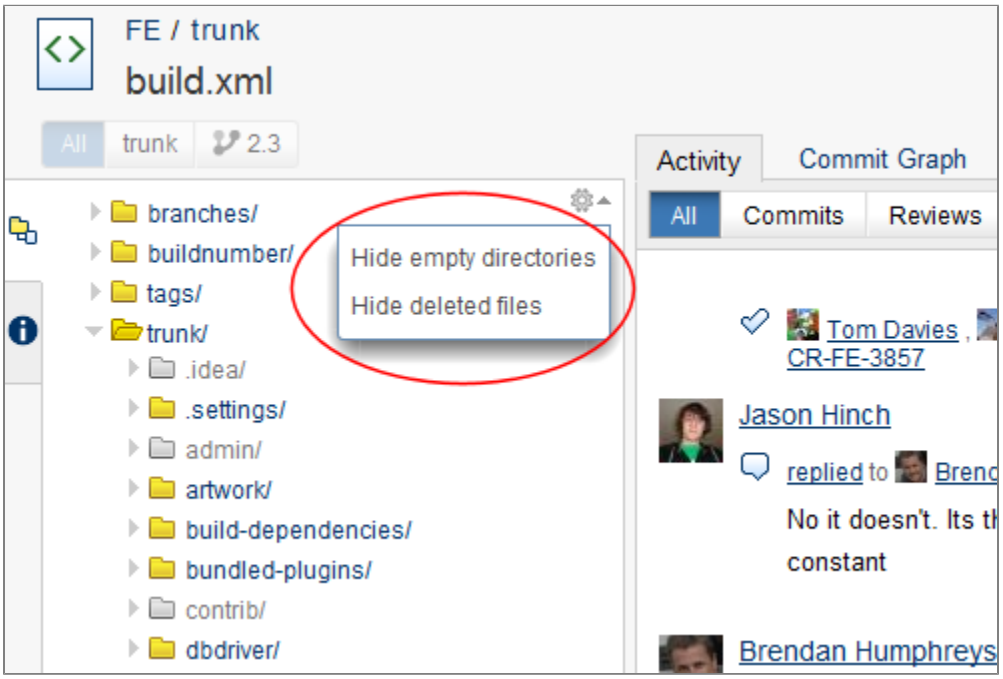
You can choose to hide deleted files and empty directories in the left navigation tree when browsing through a repository, as described below.

**To hide deleted directories/files in your navigation tree,**

1. Click the '**Source**' tab and browse a repository.
2. In the left hand navigation panel, click the to show the dropdown menu:
   - 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:**



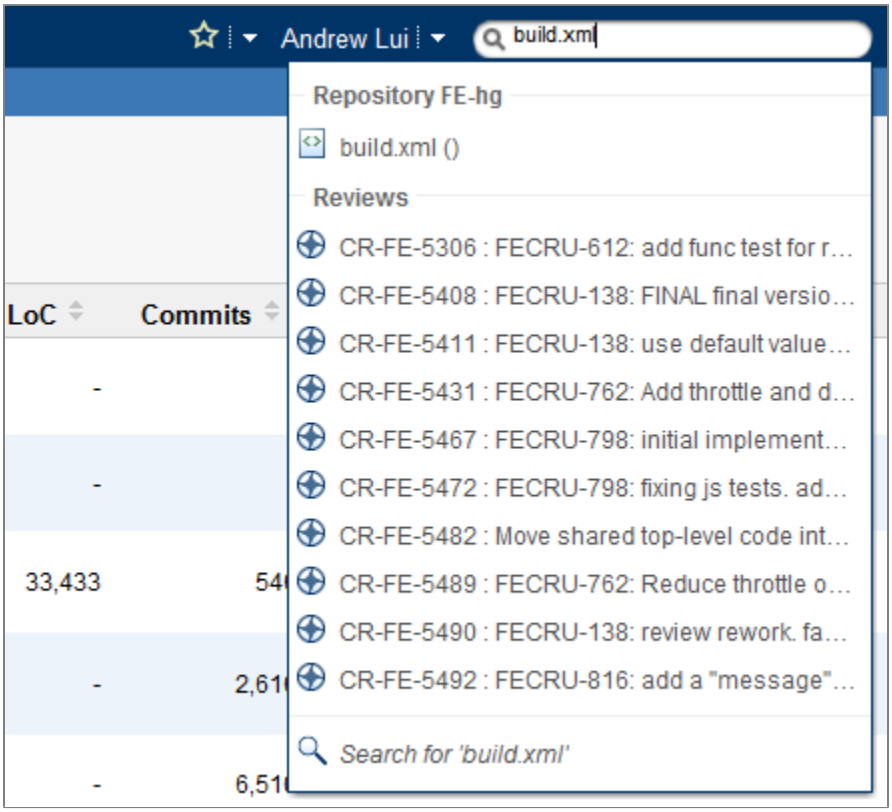
## Using the Quick Search

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.<br>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.<br>• `file` — Search against a file/directory name only.<br>• `commit` — Search against a commit message only.<br>• `diff` — Search against a diff (lines added/removed) only.<br>• `content` — Search against contents of a file only.<br>• `committer` — Search against a committer only. | Search for `file:build.xml` and FishEye will return files that have a name matching `build.xml`. |
| **Searching within a Directory** ( AntGlobs) | AntGlobs can be used in the Quick Search to help search within a specific directory. | Search for `/src/**/gwt/*.xml` and FishEye will return all files with a `.xml` suffix that are below both a src and a gwt directory. e.g. **`src`** `/java/com/atlassian/fecru/`**`gwt`**`/FecruCore.gwt`**`.xml`** but not **`src`** `/java/com/atlassian/fecru/ApplicationContext.xml` |
| **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 `update build version` or `update version in file`. |

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

### Using the Advanced Search

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

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

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

#### Specifying Search Criteria using the Standard Search Interface

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

Unable to render embedded object: File (FishEye-2.0-simplesearch.PNG) not found.
*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.

Unable to render embedded object: File (FishEye-2.0-eyeql-search.PNG) not found.
*Screenshot above: Advanced Search Criteria — EyeQL Interface*

### Notes

***Related Topics***

Searching Crucible *(Crucible documentation)*

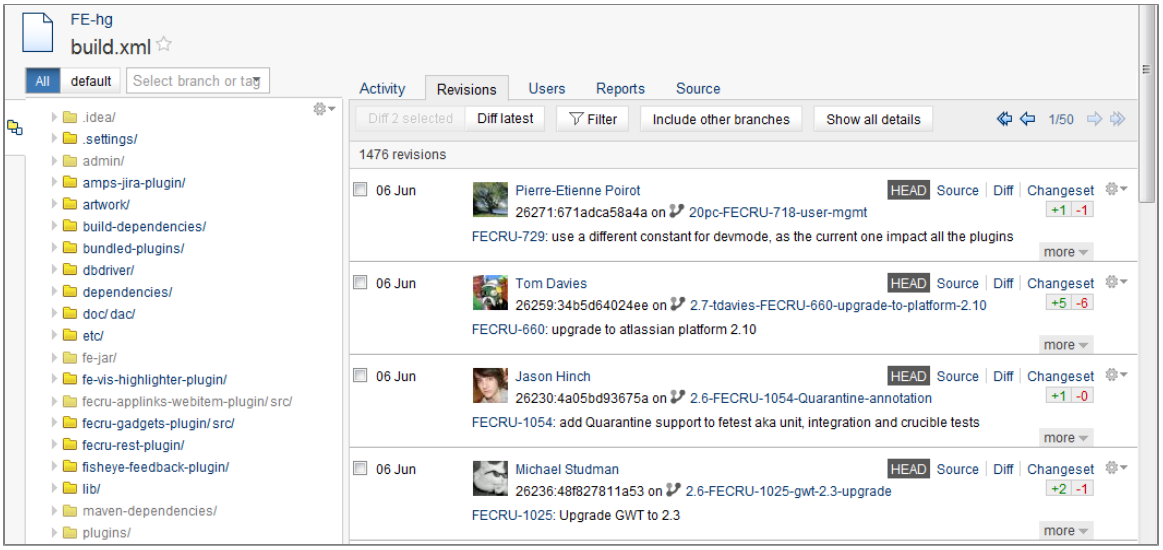
## Viewing a File

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
- Copying and Pasting Code from FishEye

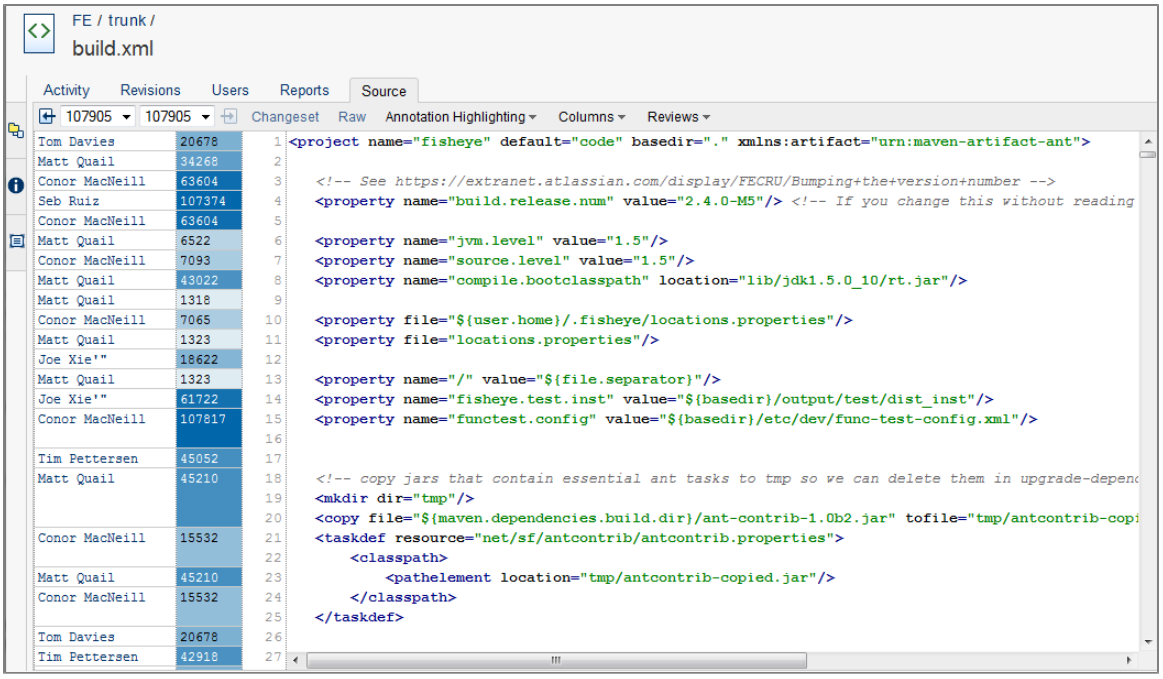
## Viewing File Content

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*:

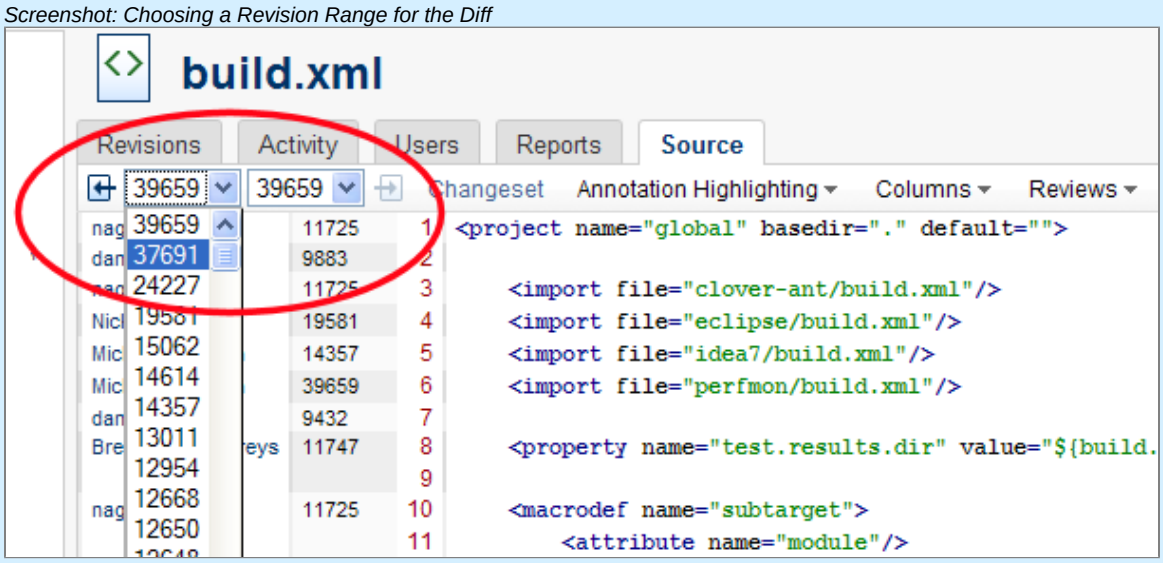


### *Opening side by side diff view*

**To open FishEye's side by side diff view,**

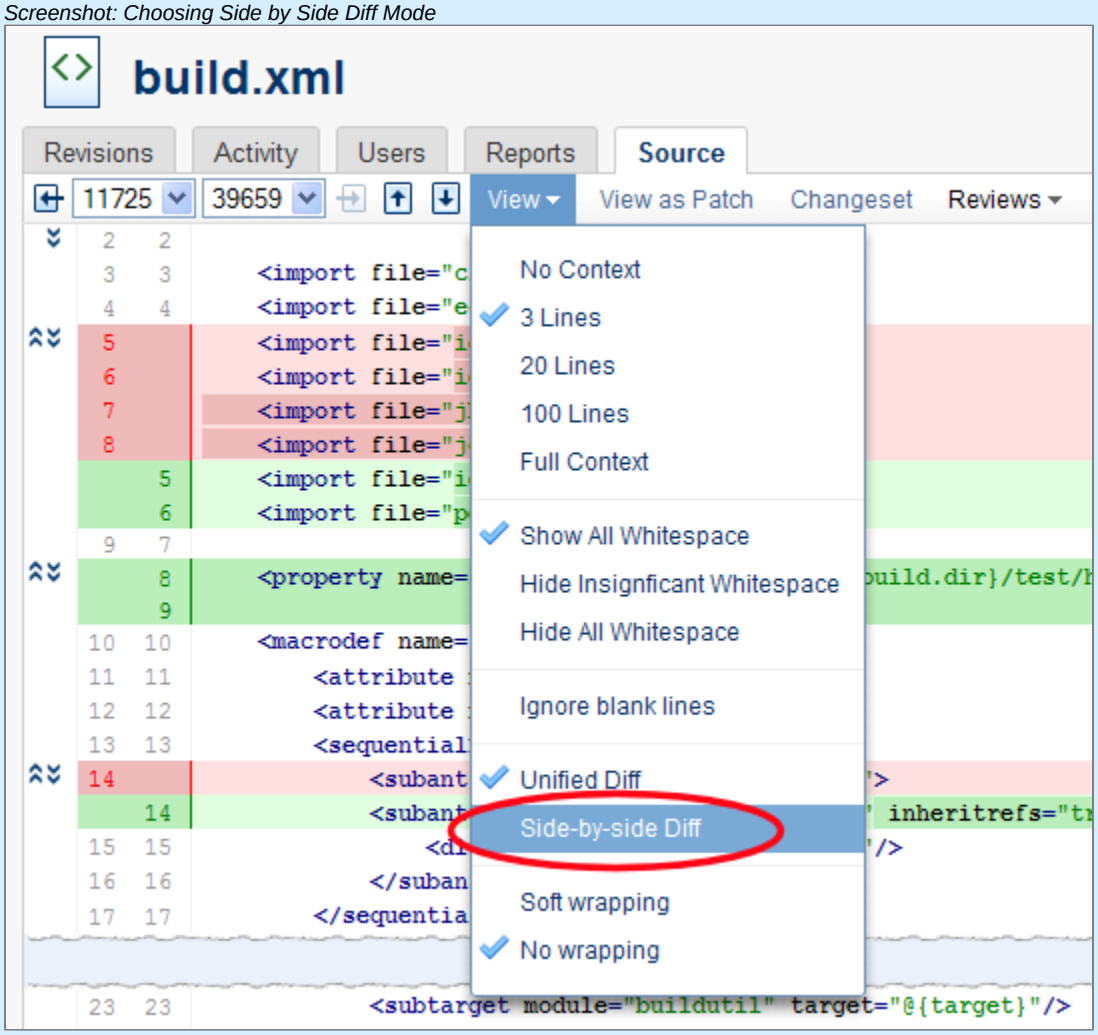
1. Open the source code view for the file in question.
2. Select a range of revisions to compare between.

*Screenshot: Choosing a Revision Range for the Diff*



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

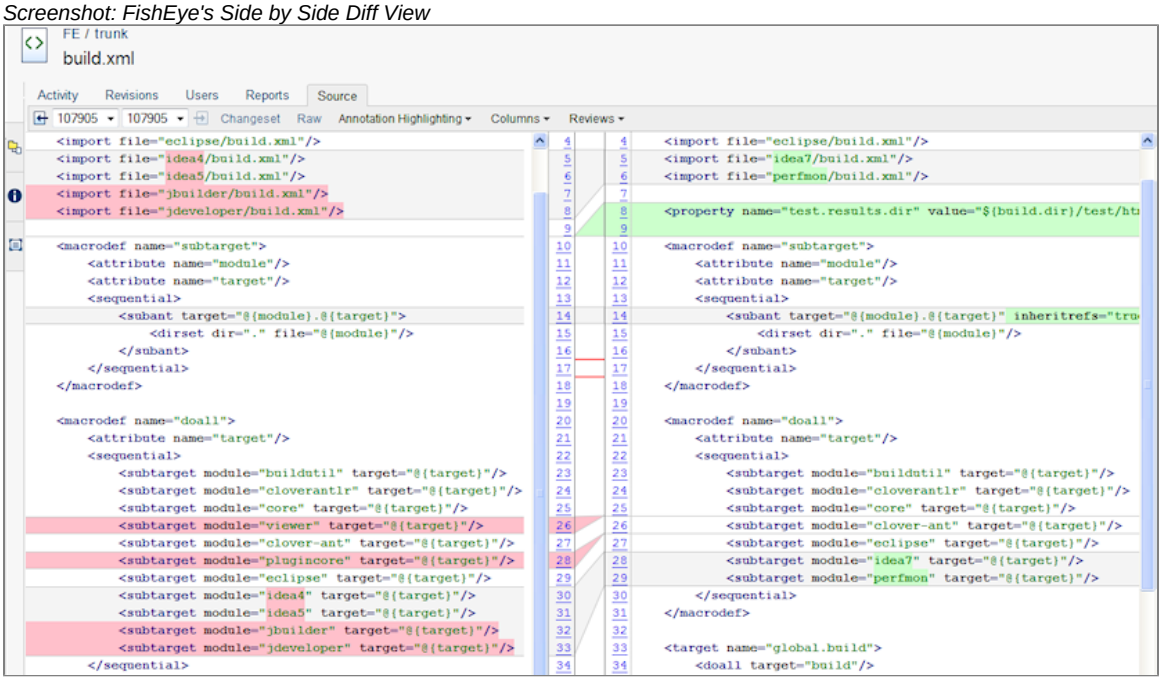
*Screenshot: Choosing Side by Side Diff Mode*




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 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 properties.
    - See the 'Overview of a File Revision' diagram below for information about the individual revisions.

*Screenshot: File Revisions*

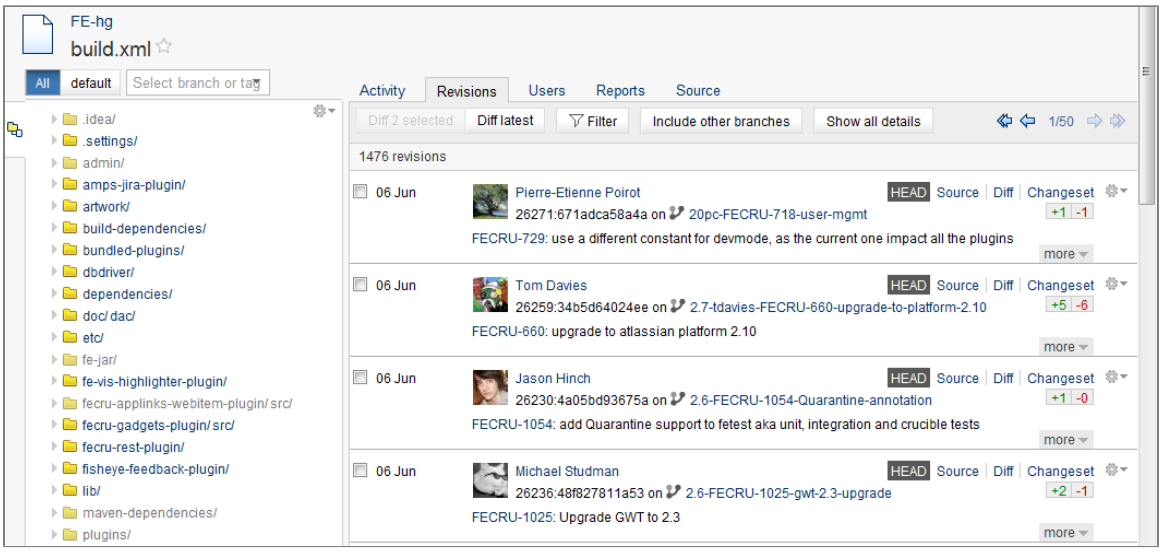
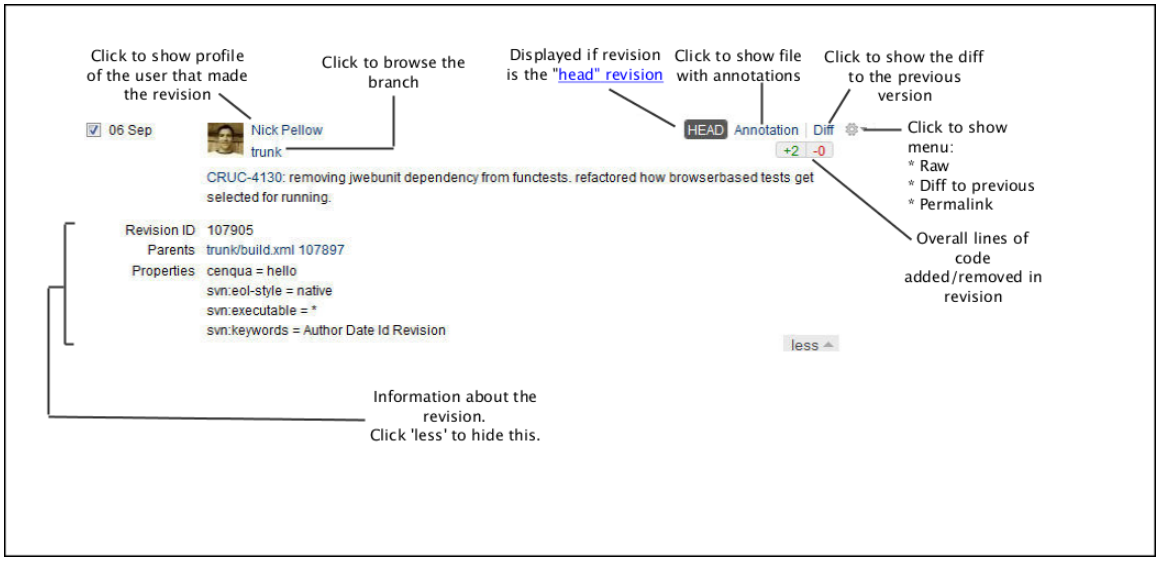



*Diagram: Overview of a File Revision*




## Viewing the Changelog

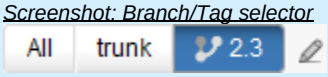
The changelog is a record of the commits, reviews and issues for a repository, branch, directory or file. You can view the recent activity in the changelog when browsing a repository/branch/directory or viewing a file. This information is displayed in an "activity stream" on the relevant screen.

**On this page:**

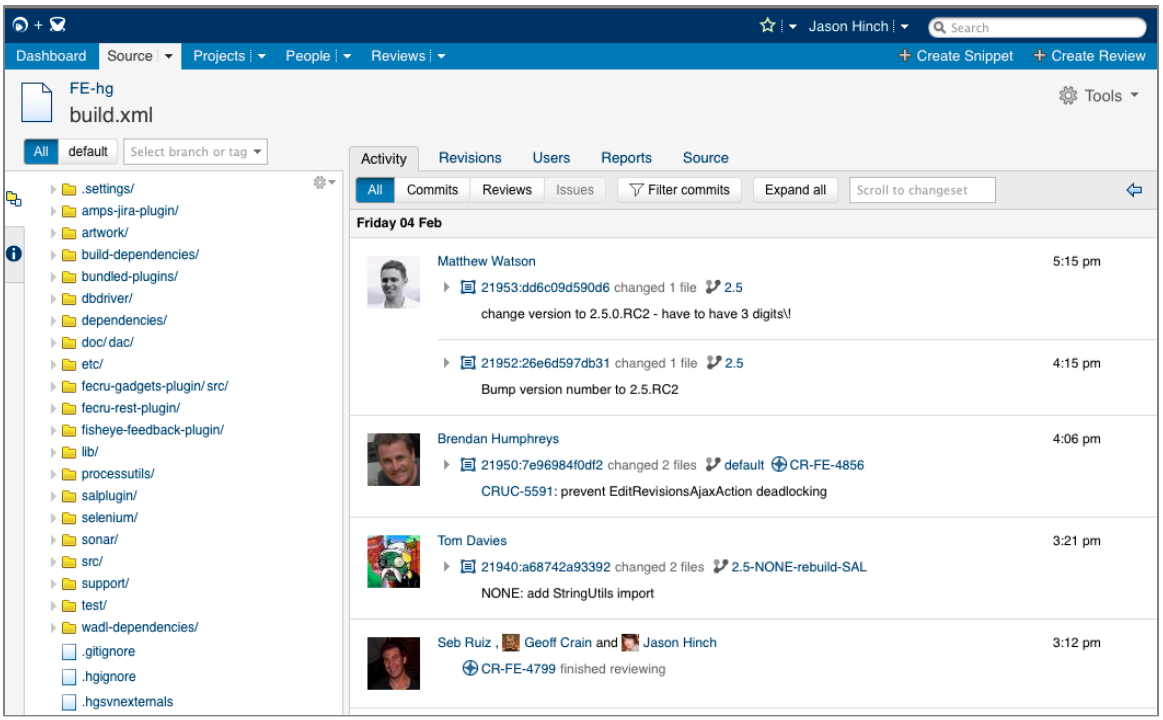


### *Viewing the Changelog Activity*

**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. *(optional)* Select the branch or tag of the repository that you want to view information for, using the branch/tag selector (non-ClearCase repositories only). 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.
     *Screenshot: Branch/Tag selector*
     
     
3. Click the '**Activity**' tab. The recent changelog activity of your repository/branch/directory or file will be displayed.
4. Use the following functions in the '**Activity**' tab to filter/show and navigate through the changelog activity:
   - **All** tab — Click to show commits, reviews[1] and JIRA issues[2] activity in the activity stream. This is the default view.
   - **Commits** tab — Click to show only commits in the activity stream.
   - **Reviews**[1] tab — Click to show only review activity in the activity stream. This tab will not be displayed unless you have integrated FishEye with Crucible[1].
   - **Issues**[2] tab — Click to show only issue activity in the activity stream. This tab will not be displayed unless you have integrated FishEye with JIRA[2].
     *This tab is currently only enabled when viewing a repository, due to technical constraints.*
   - **Filter commits** *(see below)* — See Filtering Commit Activity for the Changelog for more information.
   - **Show Revisions** — Click to expand every changeset in the activity stream to show all modified files related to the changesets. Click this button again to collapse the changesets.
   - **Scroll to Changeset** — Enter a changeset ID (e.g. `107856`) and press Enter on your keyboard to scroll to the relevant changeset in the activity stream.
     [1] Requires Crucible
     [2] Requires JIRA

*Screenshot: Viewing the changelog activity for a file*




### *Filtering Commit Activity for the Changelog*

You can filter the commits that are displayed in the activity stream, i.e. Commits in the '**All**' and '**Commits**' sub-tabs under the '**Activity**' tab.

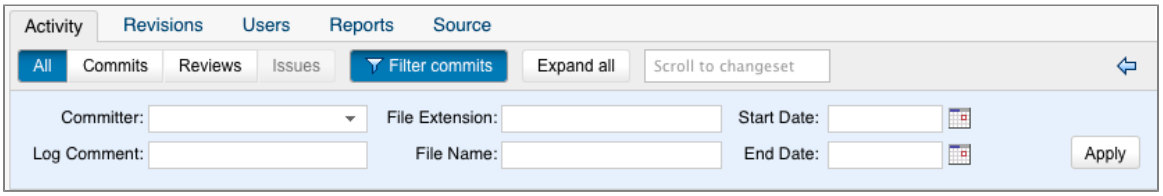
Note, you cannot use the commits filter to filter reviews or issues displayed in the activity stream. If you apply a commits filter to your activity stream, the following will occur:

- The 'Issues' tab will be disabled.
- The 'Reviews' tab will be enabled, but the commits filter will be disabled if you view the reviews activity (i.e. reviews will not be filtered).

**To filter commit activity,**

1. View the '**Activity**' tab for the desired repository/branch/directory or file.
2. Click either the '**All**' or '**Commits**' sub-tab. The relevant activity will be displayed.
   *The '**Filter commits**' button will be disabled on the 'Reviews' and 'Issues' tabs.*
3. Click the '**Filter commits**' button. The filter fields will be displayed.
4. Enter your criteria for the commits to be displayed:
   - '**Committer**' — Shows only changesets checked in by the given committer/author.
   - '**Log Comment**' — Shows only changesets where the commit comment matches the given text.
   - '**File Extension**' — Shows only changesets that contain files with the specified file extension.
   - '**File Name**' — Shows only changesets that contain a given file.
   - '**Start Date**' — Shows only 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**' — Shows only 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 '-').
5. Click the '**Apply**' button to apply the filter to the activity displayed.
   - If you want to clear the filter, click the '**Clear**' link. This link will only display when the filter is active.
     You can also click the '**Filter commits**' button again. The filter fields will be hidden and the filter removed.

*Screenshot: Using the Filter*



### *Watching the Changelog Activity*

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

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

**To watch an activity stream,**

1. Navigate to the activity stream that you want to watch.
2. Click the '**Tools**' menu and click '**Watch**'. The page will reload and a watch will be set up for the activity stream (the watch icon will be coloured, not grey).
    - If you want to remove the watch, navigate the activity stream, click 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.*

## Copying and Pasting Code from FishEye

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.

#### Copying multiple lines of code

**To copy multiple lines of code from FishEye:**

1. Click and drag in the browser window to select the desired lines.
2. A small temporary button labelled '**copy**' appears.
3. Click the '**copy**' button.
4. The selected code is added to the system clipboard. You can now use your operating system's regular paste function to make use of the code elsewhere.

#### Copying a single line of code

**To copy part of a line of code from FishEye:**

1. Click and drag in the browser window to select the desired characters inside the line of code.
2. Use your operating system's '**copy**' function (Ctrl-C, Command-C or Edit > Copy).
3. The selected code is added to the system clipboard. You can now use your operating system's regular paste function to make use of the code elsewhere.

*Screenshot: Copying Code from FishEye*

# 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

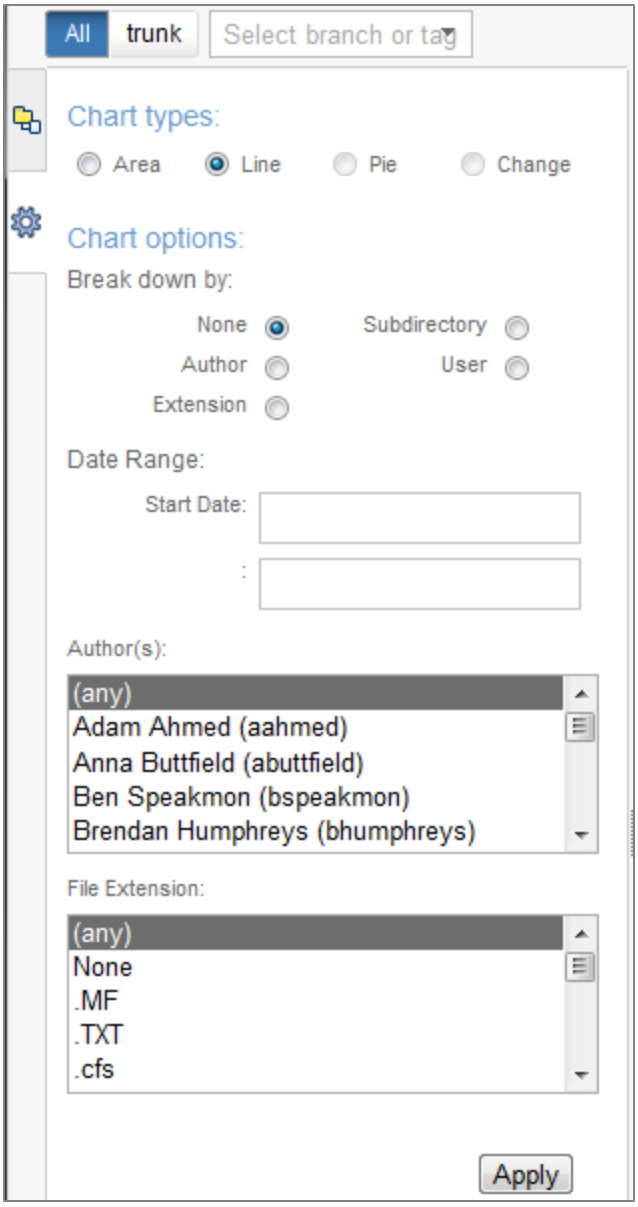
You can view chart information controlled by various criteria. Simply select the desired constraints and click the '**Apply**' button.

| 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) |

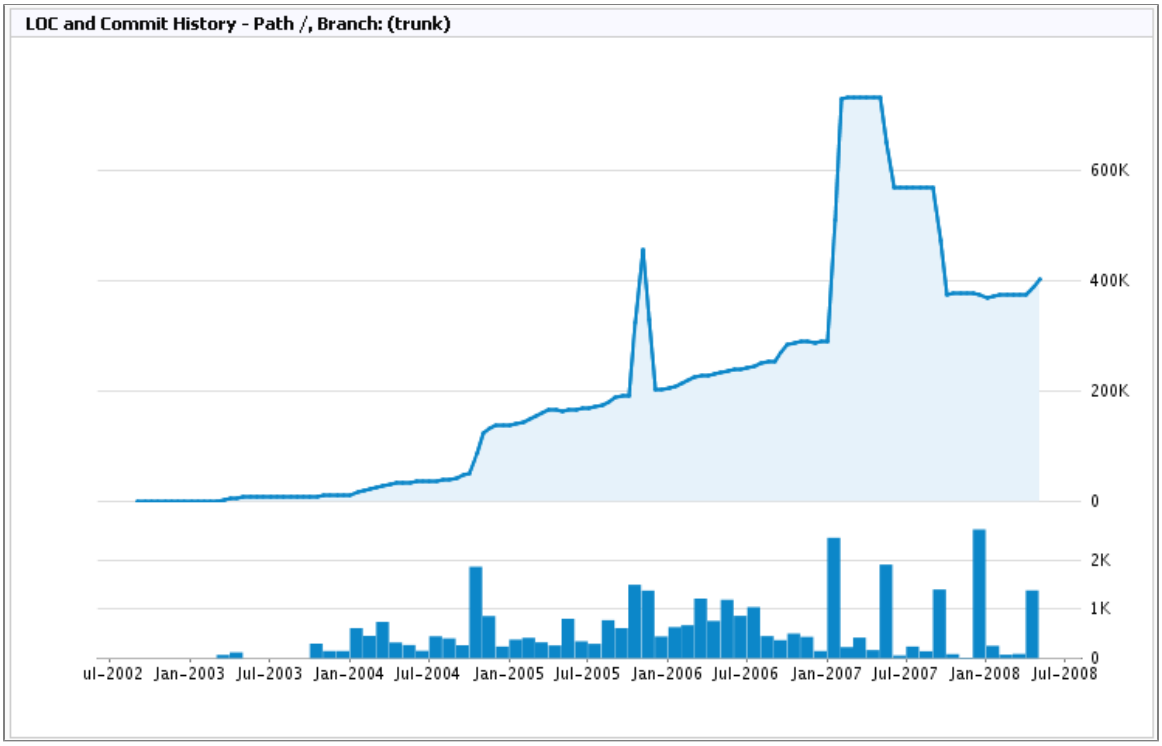
*The '**change**' chart displays the change in lines of code, for a specific date range, expressed as a line graph. For example, if the lines of code at the start date is 100, the start point will be zero and the rest of the graph shifted by 100 lines.

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

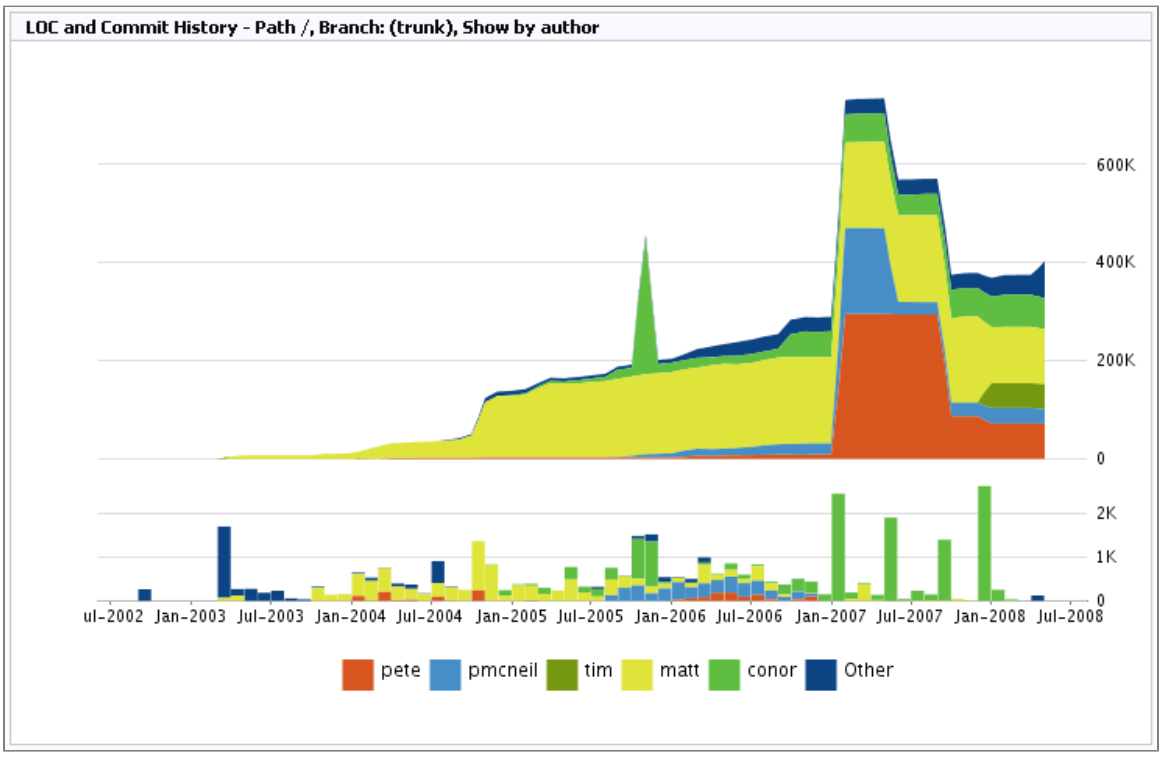
*Screenshot: FishEye custom chart settings*



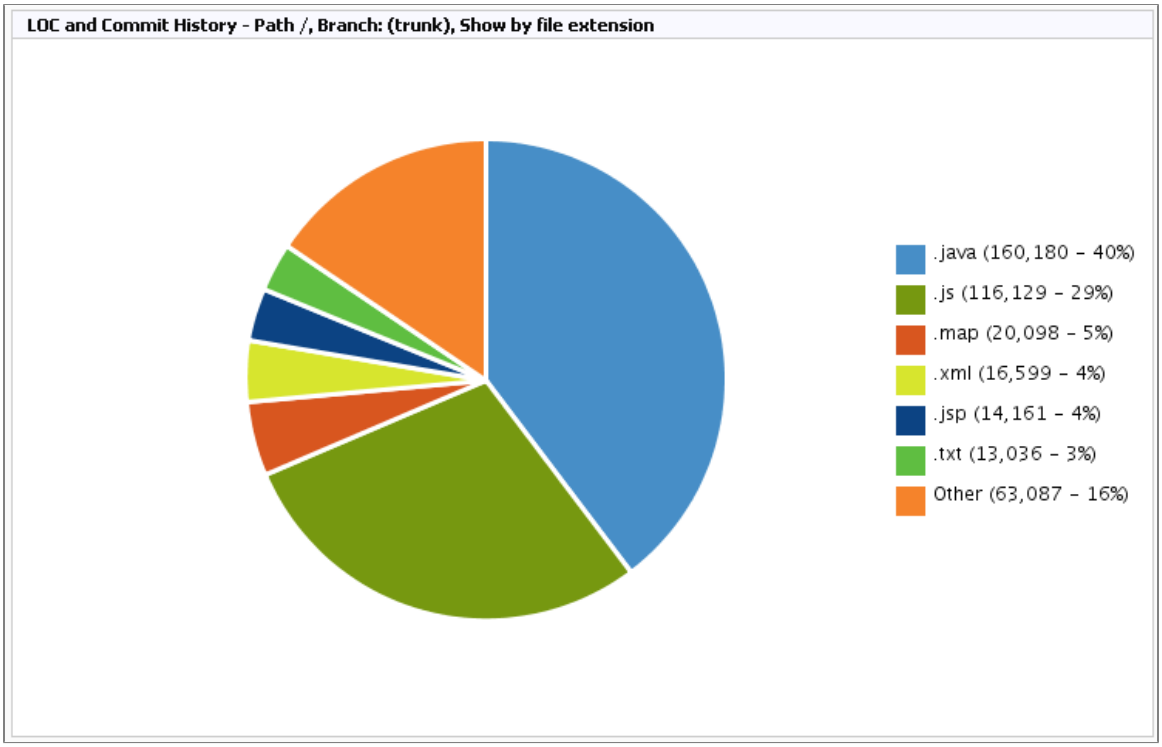

*Screenshot: FishEye per-author LOC chart*




*Screenshot: FishEye per-author LOC chart showing multiple authors*




*Screenshot: FishEye LOC chart by file extension*



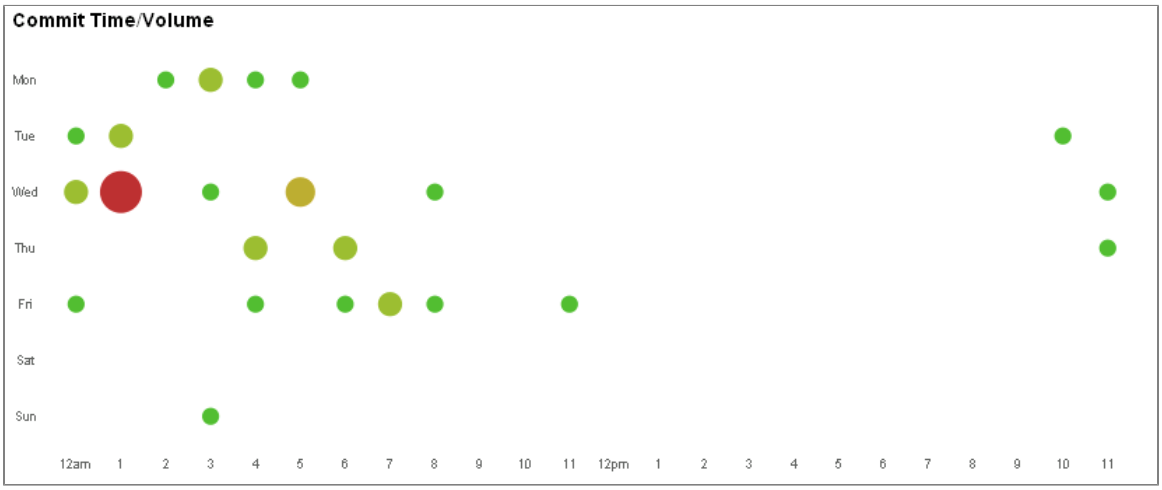

**Per-Author Lines of Code Statistics**

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.

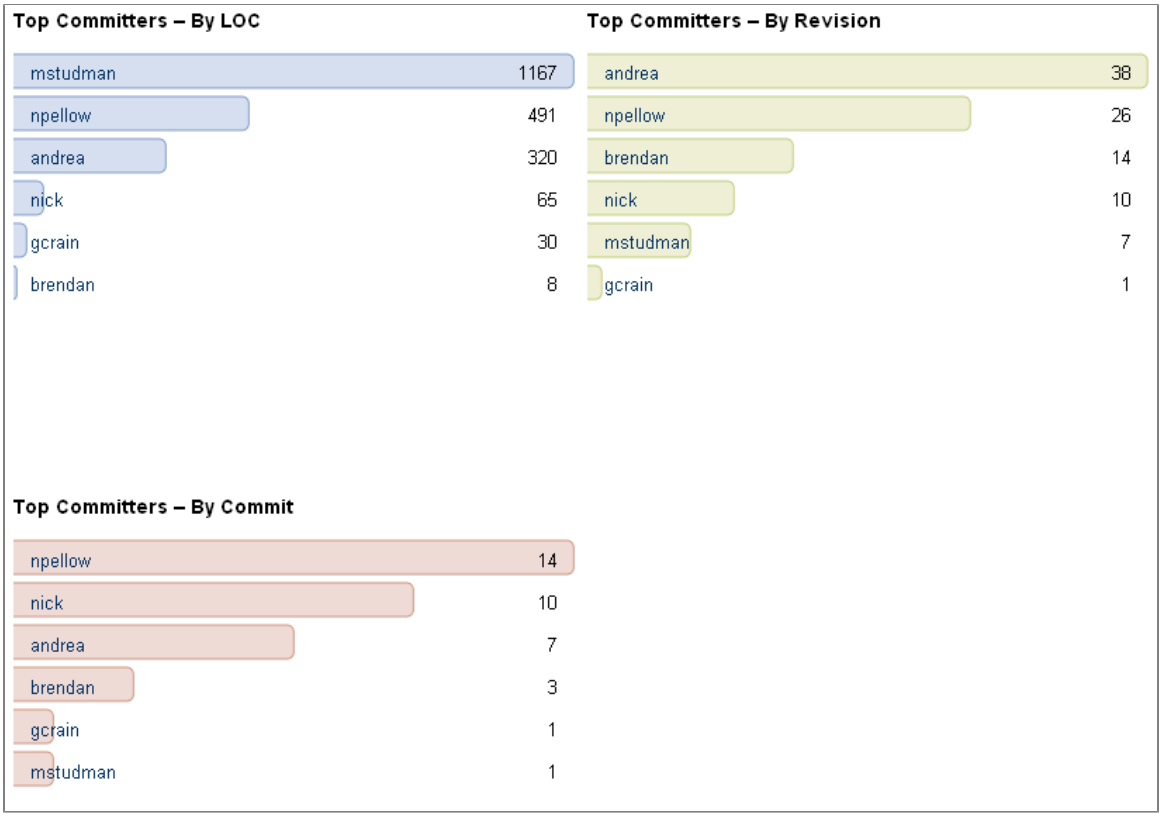
## Code Metrics

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

This page contains instructions on using the '**Favourites**' feature in FishEye to select, view and manage items of interest.

*On this page*:



### *Favourites Overview*

FishEye allows you to tag certain items as your favourites. You can select changesets, files, people and repositories to be added to your favourites. Once your favourites list is created, you can view it or see a stream of all activity relating to your favourite items. We suggest you select items that you are currently working on as favourites, to create a more relevant personalised view.

> If you are using Crucible, you can also add code reviews to your favourites.

### *Adding Items to Your Favourites*

To add an item to your favourites, follow one of the steps below.

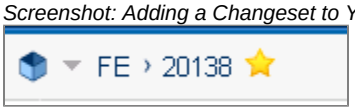
#### Adding a Person to Your Favourites

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

#### Adding a Changeset to Your Favourites

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

*Screenshot: Adding a Changeset to Your Favourites*




#### Adding a File or Folder to Your Favourites

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

#### Adding a Repository to Your Favourites

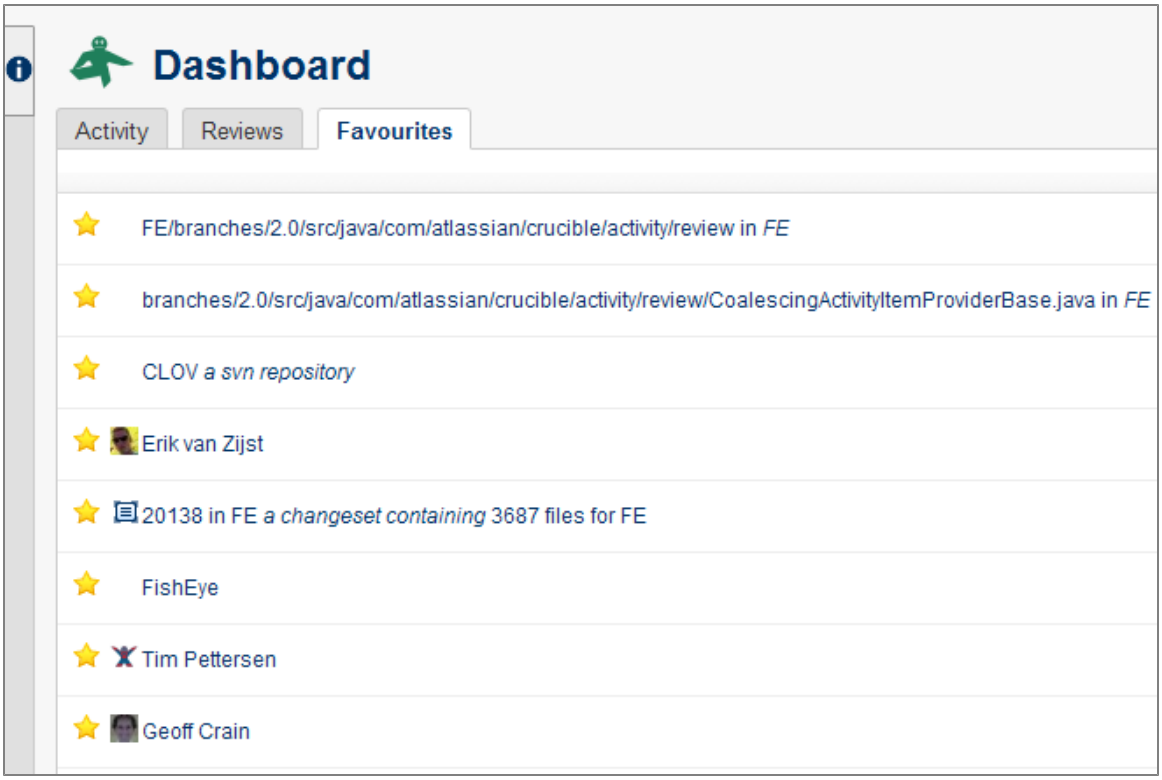
To add a repository to your favourites, click the '**Source**' tab. The '**Source**' view appears. Here, simply click the grey star icon that appears next to the name of the desired repository. The star icon will turn yellow, showing that it is selected.

*Screenshot: Adding a Repository to Your Favourites*


### *Viewing Your Favourite Items*

To view your favourite items, click '**Dashboard**' tab at the top left of the page and then the '**Favourites**' second level tab, beneath that.

*Screenshot: Viewing Your Favourites*

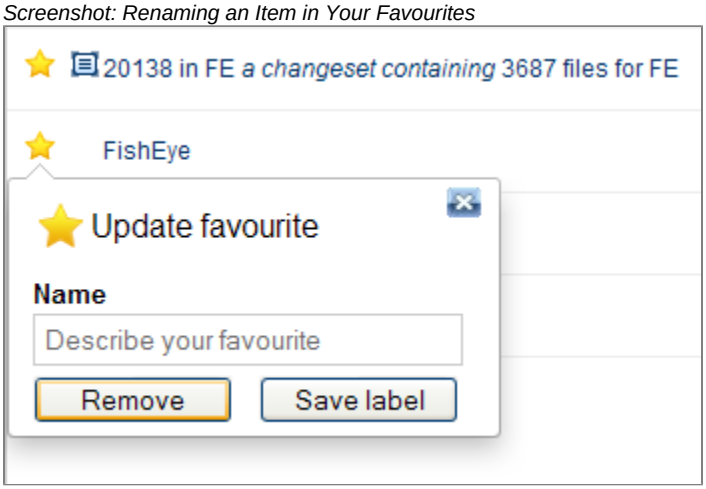



### *Renaming an Item In Your Favourites*

To rename an item in your favourites, open the Favourites drop-down menu (the gold star icon located at the top centre of the FishEye screen, next to your user menu). Select the option called '**Manage favourites**'. The Dashboard favourites page opens, showing all of your favourites in the system. To rename any item (changing its favourite display name only — not the name of item itself), simply click the yellow star to the left of its name. A small pop-up menu will appear, the '**Update Favourites**' menu. Type the desired name into the '**Name**' field and click the '**Save label**' button. The label will be updated for the favourites view.

> Due to FE-2348 you cannot currently rename favourites on Directories, Users or Committers

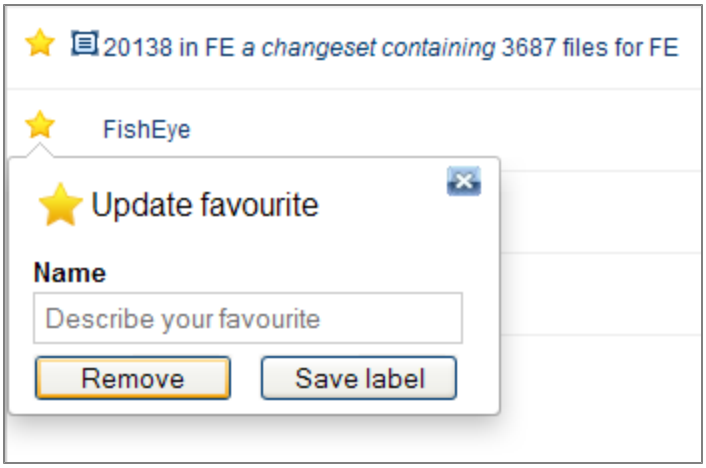
*Screenshot: Renaming an Item in Your Favourites*




### *Removing an Item From Your Favourites*

To remove an item from your favourites, open the Favourites drop-down menu (the gold star icon located at the top centre of the FishEye screen, next to your user menu). Select the option called '**Manage favourites**'. The Dashboard favourites page opens, showing all of your favourites in the system. To remove any item, simply click the yellow star to the left of its name. A small pop-up menu will appear, the '**Update Favourites**' menu. Click the '**Remove**' button. The star will turn grey, showing that it has been removed from your favourites.

*Screenshot: Removing an Item From Your Favourites*




## 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 higlights 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.

**On this page:**



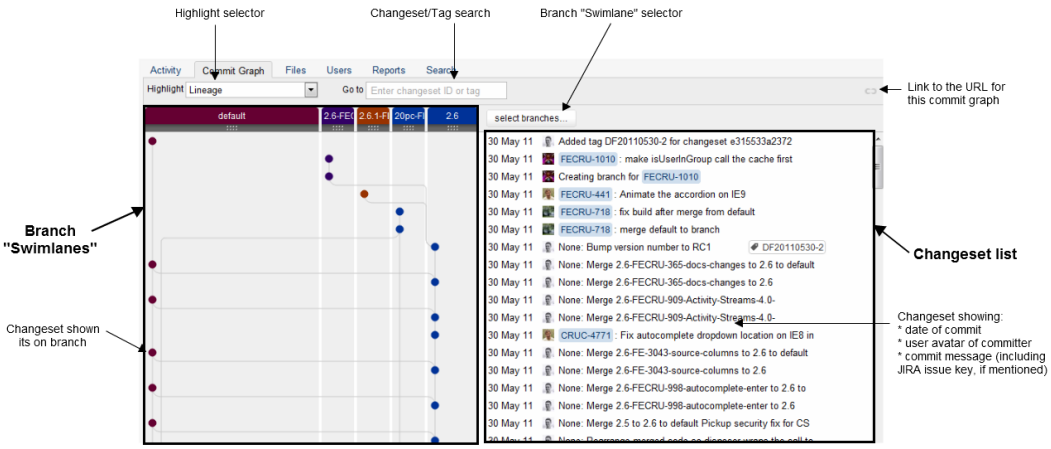
### Before You Begin

- The commit graph is not supported for ClearCase repositories.
- Subversion repositories currently do not show lines between branch swimlanes (i.e. merging).
- 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.

### Browsing to the Commit Graph for a Repository

**To view the commit graph for a repository:**

1. Navigate to the desired repository, as described on Browsing through a Repository.
2. Click the '**Commit Graph**' tab.
3. 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.

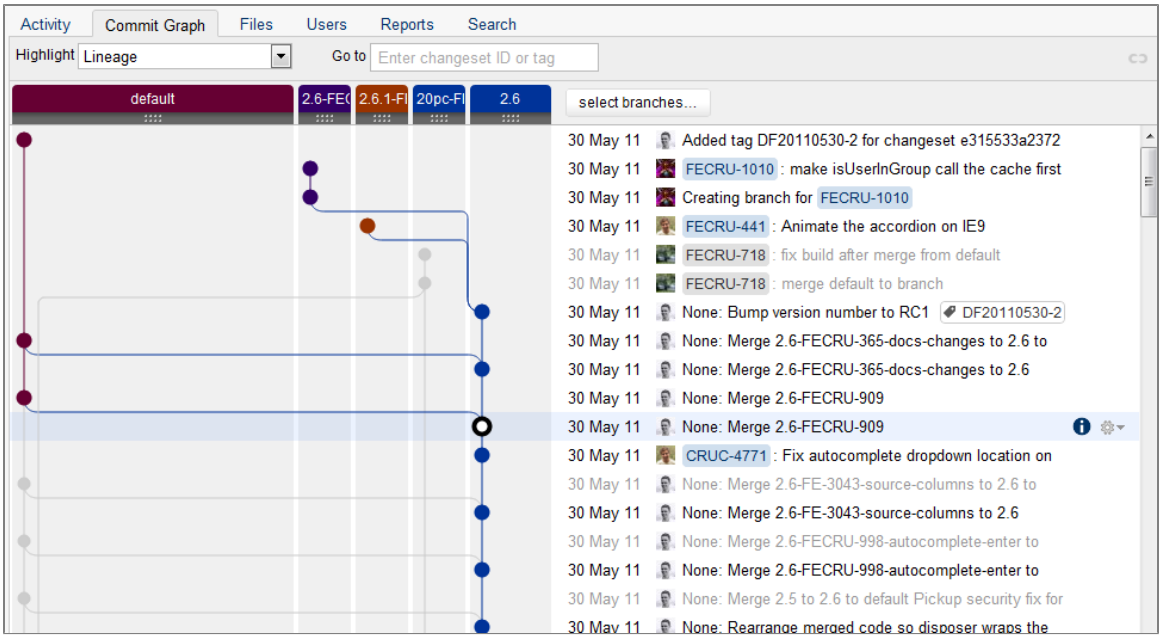
**In this section:**



### Highlighting the Lineage of a Changeset

| 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)*

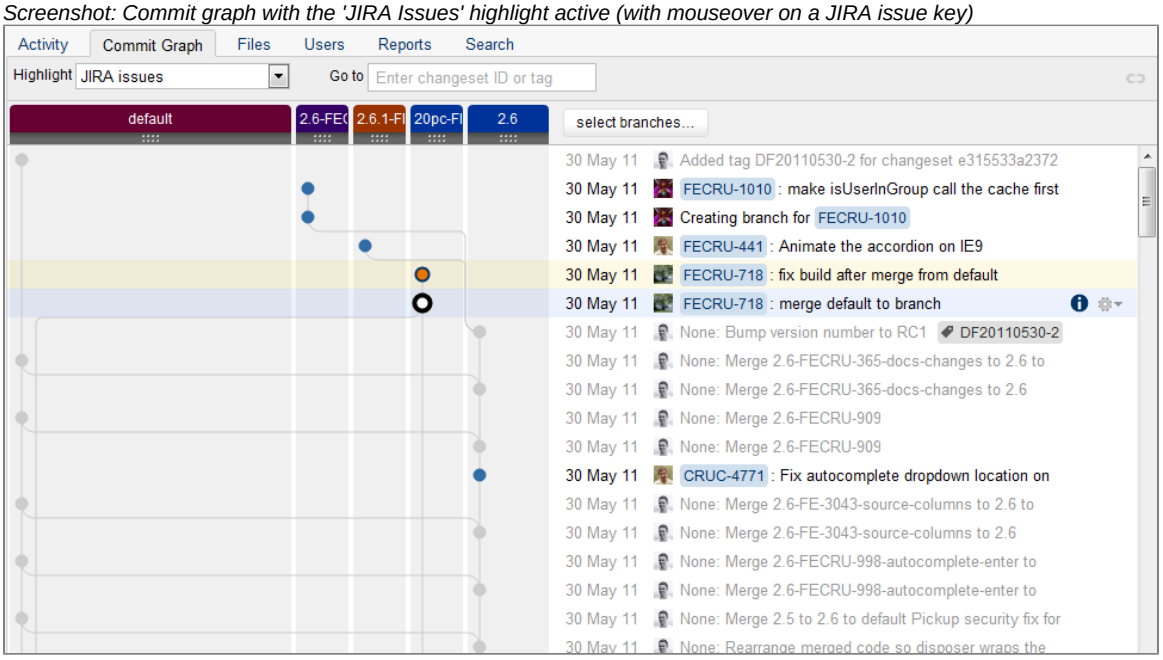



### Highlighting JIRA Issues

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)*

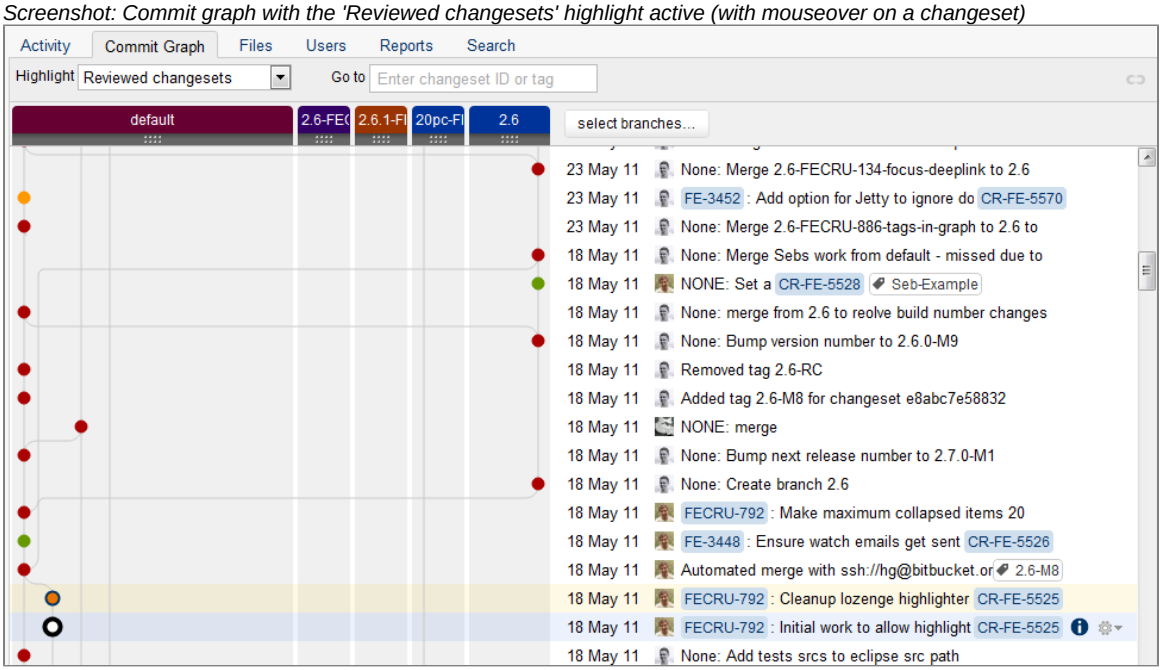



### Highlighting Reviewed Changesets

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:<br>* Red dot: Unreviewed changeset, i.e. associated with review in 'Dead' or 'Rejected' state or no review associated.<br>* Yellow dot: Changeset under review, i.e. associated with review not in 'Dead', 'Rejected' or 'Closed' state.<br>* 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)*



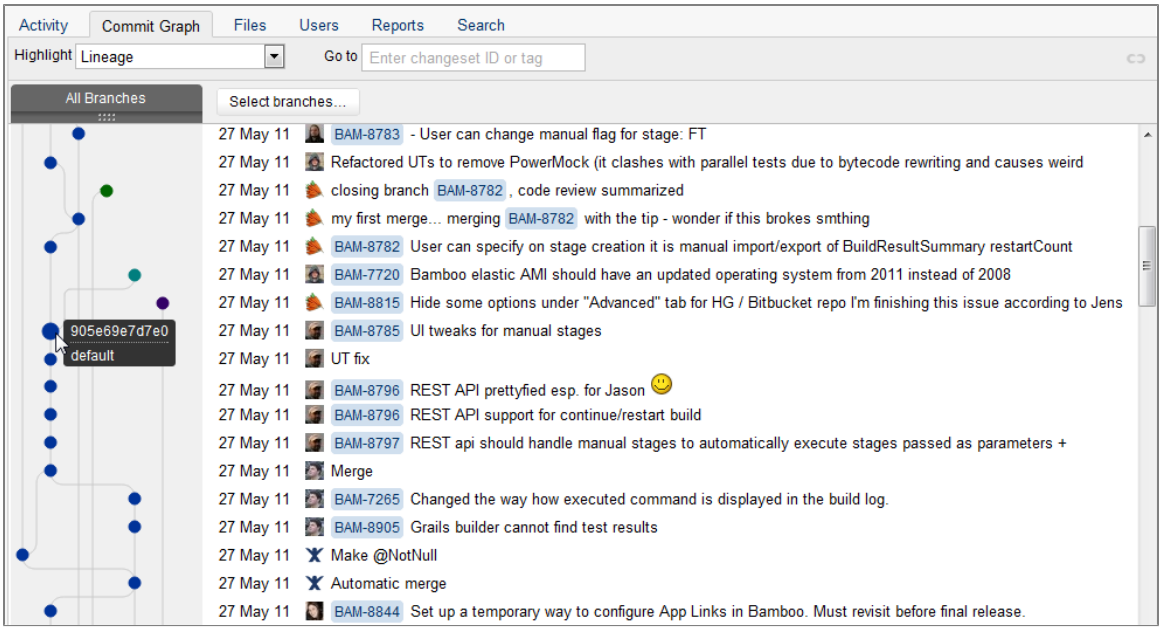

### Viewing Changesets Across All Branches

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 Changesets

### Notes

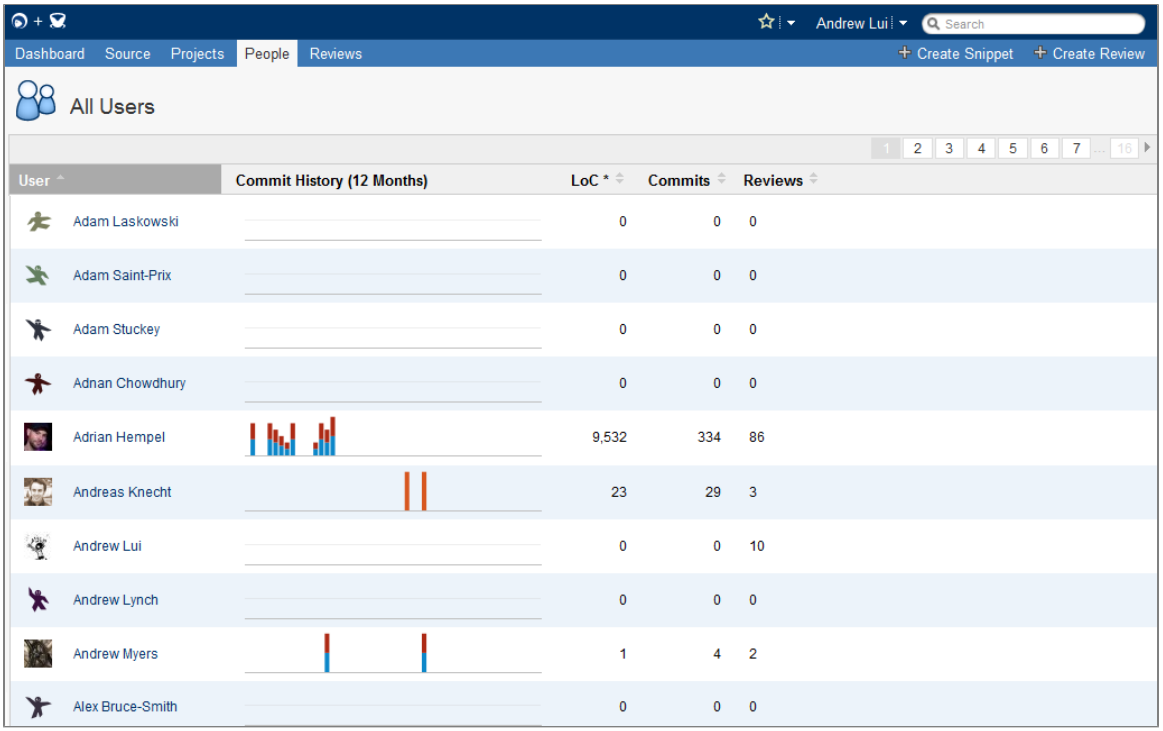
***Related Topics***

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

## Viewing 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 page. The list of all People appears.

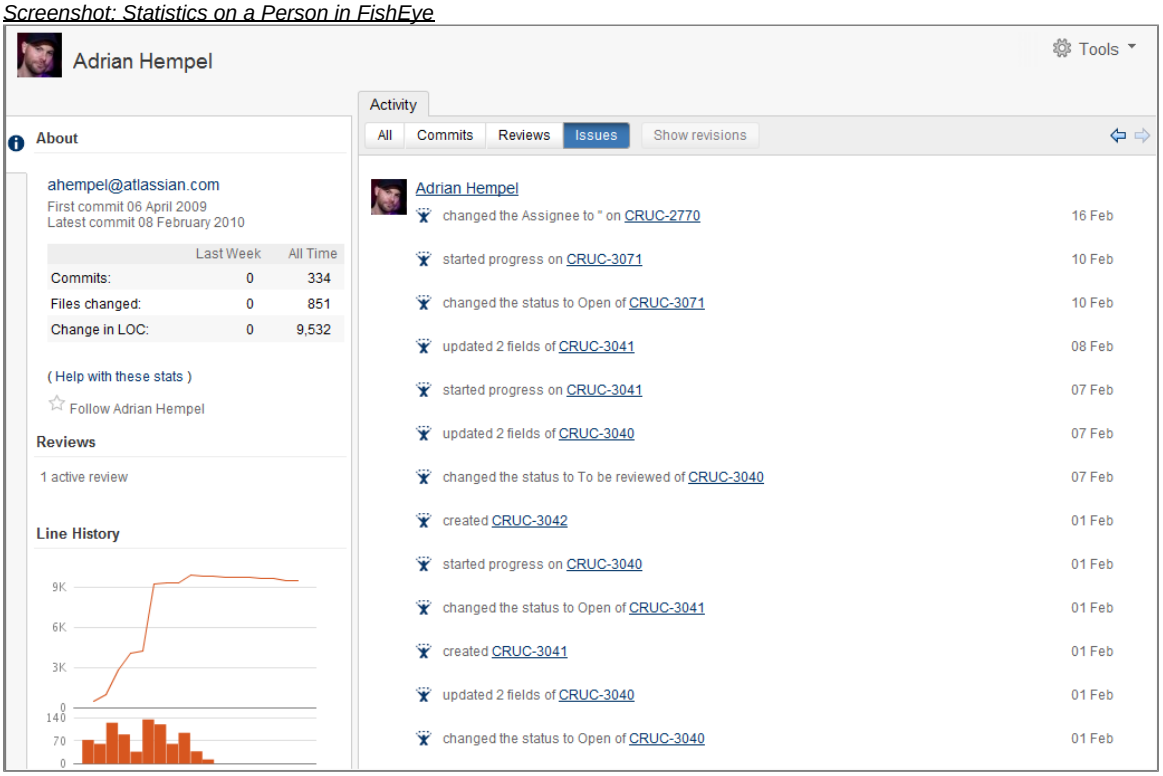
*Screenshot: List of all People in FishEye*




The list shown is comprised of all users that have accounts on the system. On the People index page, you can see the commit history in global lines of code (LOC) that each person has contributed (expressed as a line graph) and their total number of commits. Also, the most recent piece of activity is shown as a clickable item.

You can click on a person's name to see detailed information about their additions to the repository, and issue updates and reviews if using FishEye with Crucible and JIRA integration is set up.

*Screenshot: Statistics on a Person in FishEye*



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

**Avatars**

By default, each user has a unique avatar that is randomly formed from the text in their email address. Users can choose to upload their own avatar image by uploading an image to an external service such as Gravatar, which FishEye supports. See the page on Changing your User Profile.

> If you are using Crucible, statistics on each person's code reviews are also available.

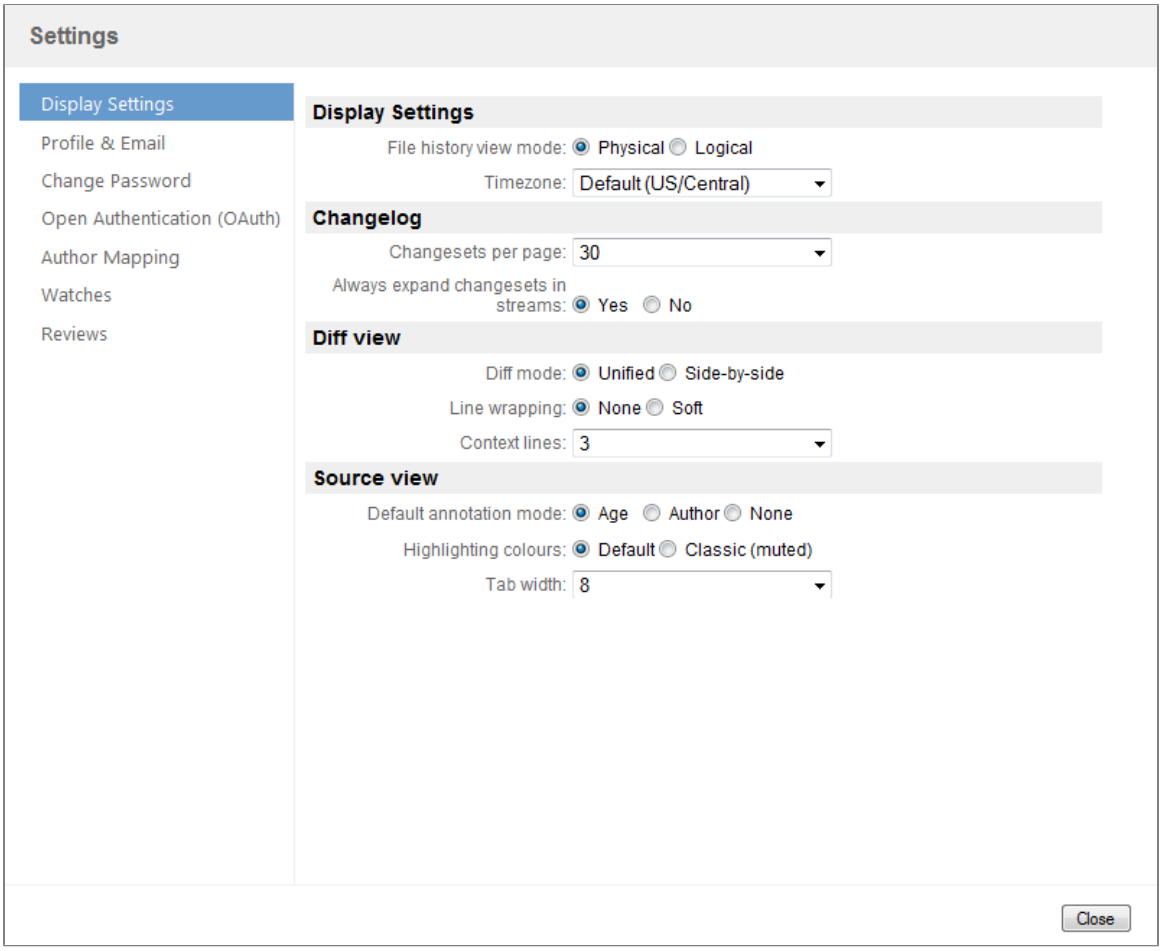
## Changing your User Profile

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. Click the User Menu (labelled with your username) at the top of the screen, then select '**Settings**'. Your user settings will be displayed.
3. Update your user settings as desired. Each tab is described in more detail below:
    - Display Settings Tab
    - Profile and Email Tab
    - Change Password Tab
    - Open Authentication (OAuth)
    - Author Mapping Tab
    - Watches Tab
    - Reviews tab
4. Click the '**Close**' button when you have finished updating your settings.

*Screenshot: User Profile Settings*




## Display Settings Tab

The options in this tab allow you to amend the display settings.

### *Display Settings*

| Field | Value |
|---|---|
| **File history view mode** | Default is 'Logical'.<br>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*

| Field | Value |
|---|---|
| **Changesets per page** | The default is 30 per page. |
| **Always expand changesets in stream** | Default is Yes. |

### *Diff View*

| Field | Value |
|---|---|
| **Diff mode** | Default is 'Unified'. Can be changed to 'Side-by-side' diffs. |
| **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*

| Field | Value |
|---|---|
| **Default annotation mode** | Default is 'Age'. It can be changed to 'Author' or 'None'. |
| **Highlighting Colours** | The default colour 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 Tab

| Field | Value |
|---|---|
| **Display Name** | Name displayed for the user currently logged in. |
| **Email Address** | The address all email notifications will be sent to. |
| **Email Format** | Default is text. Can be changed to be sent as HTML. |
| **Send Watch Emails** | The frequency at which emails will be sent for watch notifications:<br>/* 'Immediately' — The email is sent every time a change is made. This is the default value.<br>/* 'Daily' — You will receive a daily email summarising changes. |
| **Profile Picture** | Upload an avatar image of your choice. This image will be displayed next to your username throughout FishEye/Crucible.<br>Accepted formats are JPG, GIF and PNG. Image file size limit is 2Mb. Images will be automatically be cropped on upload. |

## Change Password Tab

Change your password via this tab, if required. Please note, the passwords are **case-sensitive**.

This tab will not display if your FishEye instance is connected to an external LDAP authentication source, like 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 Tab

This functionality is used by Crucible.

The 'Author mapping' tab allows you to make an association between you (as a logged-in user in Crucible) and an author in each repository.

This is only necessary if the name of the user within Crucible is different to the name within the repository. Crucible will by default check to see whether the usernames match.

## Watches Tab

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

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

## Reviews tab

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'*

| | |
|---|---|
| **Auto-mark files as read** | Default is 'Yes'. Can also be set to 'No'. |

### *Review Notifications Events*

The following options can be set to 'Immediate', 'Batch', or 'No':

| | |
|---|---|
| **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. |
| **Participant finished** | Default is 'Immediate'. An email is sent (to the Moderator only) when any reviewer has completed their review. |
| **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'. |

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

The following options can be set to 'Yes' or 'No':

| | |
|---|---|
| **Set review to incomplete if defect is raised** | Default is 'Yes'. This allows reviews to be resurrected automatically to deal with new code or defects. Can also be set to 'No'. |
| **Set review to incomplete if revision is added** | Default is 'Yes'. This allows reviews to be resurrected automatically to deal with new code or defects. Can also be set to 'No'. |
| **My actions** | Default is 'No'. If set to 'Yes', an email is sent every time you perform an action on a review. |

## Re-setting your Password

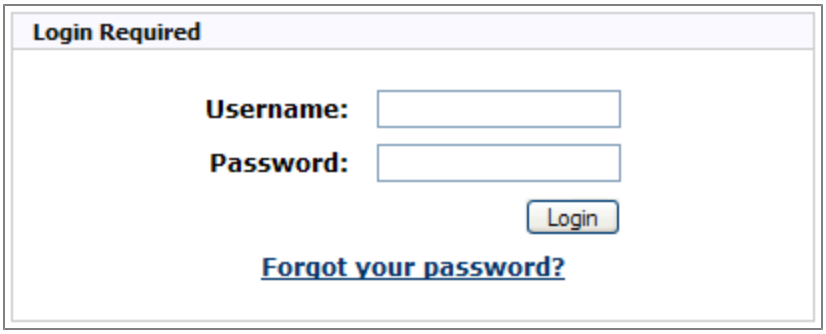
If you need to reset your password, FishEye has an integrated mechanism to generate a new password and send it to the email address in your profile.

**To reset your password:**

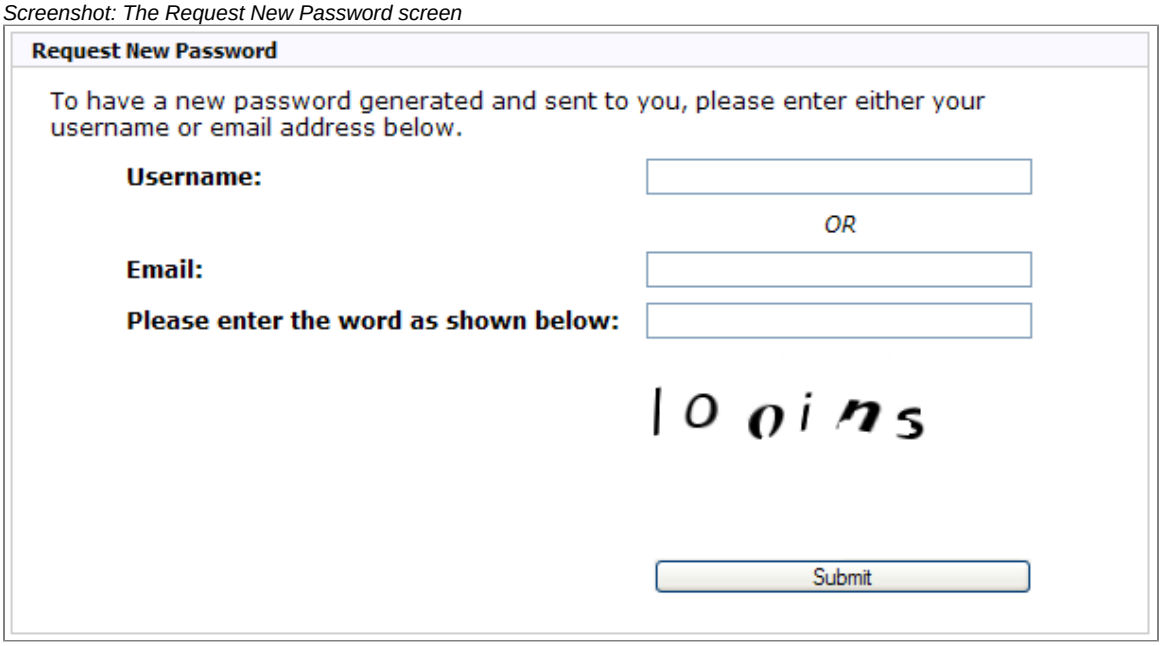
1. On the log in screen, click the '**Forgot your password?**' link. The '**Request New Password**' screen opens.
2. Fill out your username or email address and the Captcha step. That is, click in the form field labelled '**Please enter the word as shown below**' and type the graphical letters shown 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.

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*




*Screenshot: The Request New Password screen*




## Antglob Reference 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 *path segments*. |

Remember that Antglobs 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`<br>but not `/foo.txty` or `/bar/foo.txty/` |
| /*.txt | Matches `/foo.txt`<br>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`<br>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`<br>`todaygmt` | The instant at 00:00:00 today. (server-time* or GMT) |
| `thisweek`<br>`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`<br>`thismonthgmt` | The instant at 00:00:00 on the first day of this month. (server-time* or GMT) |
| `thisyear`<br>`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-field*s are **isBinary**, **isAdded**, **isDeleted**, **isCopied**, **isMoved**. e.g. **count(isAdded)** will return the number of added files.

*numeric-field*s are **totalLines**, **linesAdded**, **linesRemoved**.

*other-field* can be **path**, **dir**, **author**, **date**, **csid**, **tags** or **reviews**. e.g. **count(distinct path)** will return the number of unique paths.
**count(distinct tags)** will return the number of unique tags.

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

With no `group by` clause, you can have:

- `return` *normal columns*
- `return` *aggregates*

With a `group by changeset|csid` clause:

- `return` *normal columns*
- `return csid, comment, date, author`, *aggregates*

With a `group by file|path` clause:

- `return` *normal columns*
- `return path`, *aggregates*

With a `group by dir|directory` clause:

- `return` *normal columns*
- `return dir`, *aggregates*

i.e. The EyeQL can contain a **returns** clause that contains all non-aggregate columns, or all aggregate columns.
Non-aggregate and aggregate columns can only be mixed if the columns are unique for the grouping.

*limit-clause:*

( *length* | *offset*, *length* | *length* **offset** *offset* )

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

### Examples

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

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

# FishEye Administrator's Guide

Once you have installed and configured 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.

### Topics



# 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

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

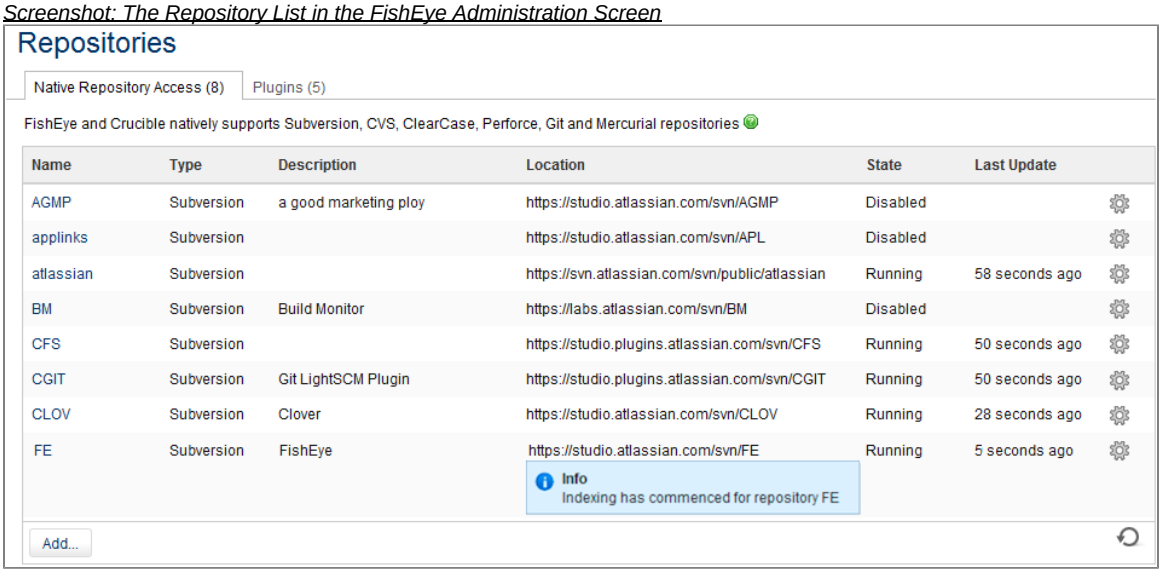
1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

   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)* — Delete the FishEye profile for the disabled repository. See Operations.
     - '**Application Links**' — Configure application links for the repository.
   - Click the '**Add**' link 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 the '**Defaults**' link in the left menu. See Repository Options for more information.

*Screenshot: The Repository List in the FishEye Administration Screen*




## Adding a New Repository

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

- Adding a Repository
  - ClearCase
  - CVS
  - Git
  - Mercurial
  - Perforce
  - Subversion
    - SVN fisheye.access
    - SVN Tag and Branch Structure

- How Tags Work in Subversion

### Accessing FishEye Repository Options

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

- Repository Options
  - Allow (Process)
  - Authentication
  - Commit Message Syntax
  - Configuring Repository Details
  - Hidden Directories
  - Indexer
  - Linkers
  - Operations
  - Permissions
  - Properties
  - Store Diff Info
  - Tarball Settings
  - Updater
  - Watches

## Adding a Repository

The instructions on this page describe how to add a new repository to FishEye. Once a repository has been added, further configuration options are available, depending upon the type of repository.

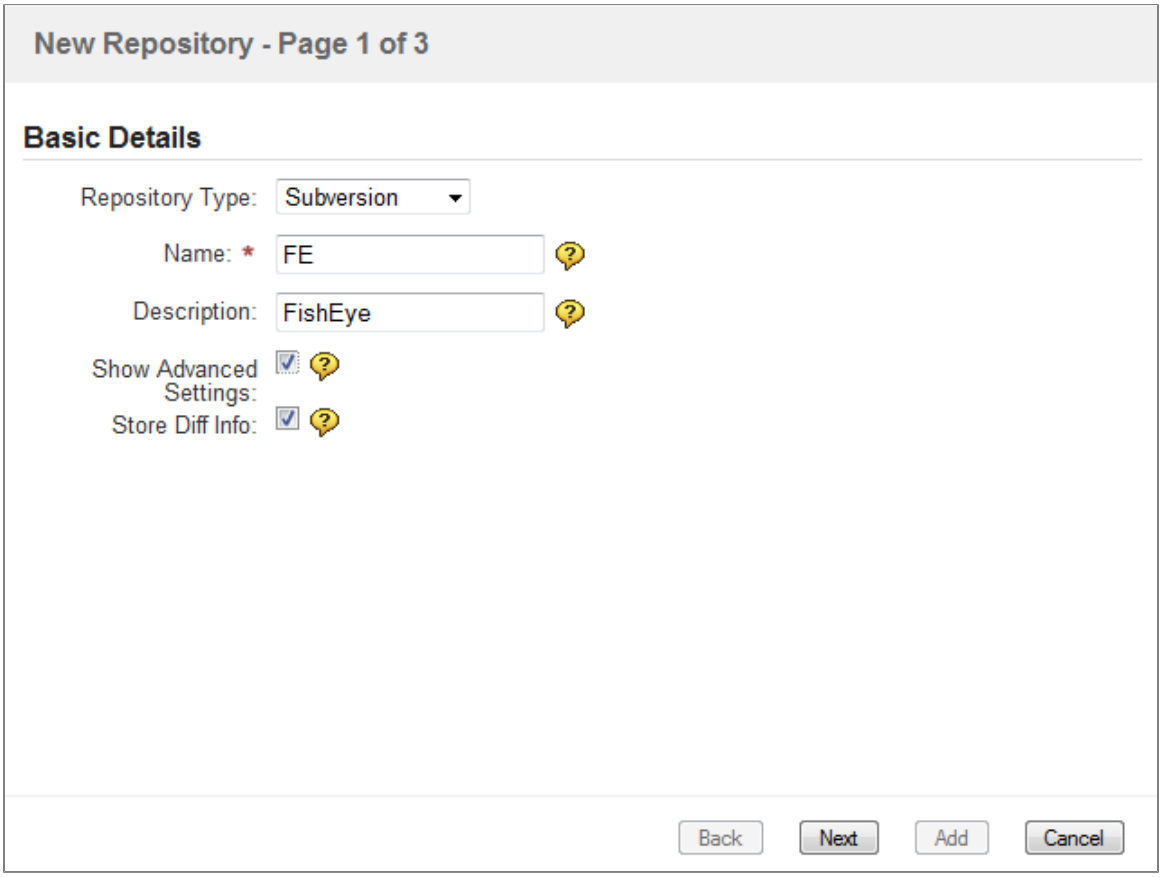
Please note, FishEye needs to build an index and cache of a repository when the repository is first added. This begins when you first **enable** a repository, and may take some time to complete.

**To add a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

2. Click the '**Repositories**' link. The list of repositories set up in your FishEye instance will be displayed.
3. Click the '**Add...**' link at the bottom of your list of repositories. The 'Basic Details' screen of the 'New Repository' dialogue will be displayed.
4. Complete the 'Basic Details' for your new repository, as follows:
   - Select a '**Repository type**' from the dropdown list.
   - Enter a '**Name**' for your repository.
   - Enter a '**Description** for your repository. This is an optional field.
   - Tick the '**Show Advanced Settings**' checkbox to show advanced settings for your repository. The following fields that display will vary depending on the chosen repository type:
     - '**Store Diff Info**' — *(non-CVS repositories only)* Tick this checkbox to allow FishEye to store information about file diffs (not the diffs themselves). Read more about this setting in Configuring Repository Details.
5. Click the '**Next**' button to configure the repository settings specific to the repository type. See the following topics for more information:
   - ClearCase
   - CVS
   - Git
   - Mercurial
   - Perforce
   - Subversion
6. Click the '**Add**' to add your new repository when you have entered all of the repository settings.

*Screenshot: Adding a Repository*




## ClearCase

> **End of Support for ClearCase**
> On **April 4th 2012**, we are ending FishEye support for IBM ClearCase. For more information, see End of Support Announcement for IBM ClearCase.

This page contains instructions for how to set up a ClearCase repository in FishEye, a configuration reference and a list of known issues.

If you also have Crucible, you will also be able to run Crucible reviews on code from your ClearCase repository, once configured.

**On this page:**



### *Requirements*

The instructions on this page require the following:

**Applications:**

- IBM ClearCase 2003.06.10 or later

- The `cleartool` command must be installed on the same server as FishEye and must be available in the `PATH` of the user that is running FishEye.

**Permissions:**

- The FishEye process must be run as a user that is part of the required groups with access to the ClearCase VOBs. This is because FishEye uses the Cleartool client to access VOBs, and ClearCase uses the user and group information from the operating system to grant or limit access to VOB content.

### *Setting up a ClearCase Repository*

**To add a ClearCase repository,**

1. Open the 'New Repository' dialogue by following the instructions on Adding a Repository.
2. Set your ClearCase repository details, as described below. Also see the screenshots below.

***1. Enter the Basic Repository Details***

| Field | Description | Allowed values |
|---|---|---|
| **Repository Type** | Select 'ClearCase'. | ClearCase |
| **Name** | Enter a display name. This name will be displayed in the list of FishEye repositories. | Free text |
| **Description** | Optionally enter a description for this repository. | Free text |

***2. Enter the ClearCase Settings***

| Field | Description | Allowed values |
|---|---|---|
| **View Location** | If '**Auto Create View**' is ticked, enter the location of a directory accessible to the FishEye instance where snapshot views can be created.<br>If '**Auto Create View**' is not ticked, enter the location of an existing ClearCase view. This can be either a dynamic or a snapshot view. (Note, this is **not** the view storage location typically .vws directory). | A system path |
| **Auto Create View** | Tick the checkbox, if you want FishEye to create views for each VOB or UCM Project being indexed.<br>Do not tick the checkbox, if you want FishEye to use an existing view (specified in the 'View Location' field below). | Yes/No |
| **UCM** | Choose whether the underlying ClearCase repository uses UCM or Base ClearCase.<br>The rest of the fields on this screen will change depending on which option you choose (see screenshots below) | Yes / No |

If you have selected **'UCM' to be 'No'** (i.e. you are configuring a Base ClearCase repository) complete the following fields:

| Field | Description | Allowed values |
|---|---|---|
| **Main Branches Only** | Tick this checkbox, if you only want changes made on the main branch or delivered to the main branch to be indexed. Do not tick this checkbox, if you want all changes to be indexed. | Yes / No |
| **Index Start Date** | Enter an index start date, if you only want changes that were made after this date to be indexed. Note, files that have not changed since the index start date will not be displayed at all. | Date (YYYY-MM-DD) |
| **Block Size** | Enter the block size, i.e. how many change sets FishEye and Crucible will process in one batch.<br>*(This setting only displays if you tick the '**Show advanced settings**' checkbox at the bottom of the dialogue.)* | Number |
| **Command Timeout** | Enter how long you want FishEye and Crucible to wait for ClearTool commands to complete, e.g. "100000". If not set, this will default to "3600000".<br>*(This setting only displays if you tick the '**Show advanced settings**' checkbox at the bottom of the dialogue.)* | Number (in milliseconds) |
| **VOB to Include** | If you only need FishEye to index a *single VOB*, select the VOB to index. The dropdown list will contain all non-UCM VOBs found in the ClearCase installation. | Auto-populated |
| **Include Pattern** | If you need FishEye to index *multiple VOBs*, use a pattern to specify which VOBs to include. Multiple inclusion patterns can be separated with a comma.<br>See the Inclusion/Exclusion Settings section for examples of inclusion patterns. | Free text |

| | | |
|---|---|---|
| **Exclude Pattern** | If you need FishEye to index *multiple VOBs*, use a pattern to specify which VOBs to exclude. Multiple exclusion patterns can be separated with a comma.<br>See the Inclusion/Exclusion Settings section for examples of exclusion patterns. | Free text |
| **Branches to Include** | Enter the list of the branches to include in indexing. An empty list will cause all branches to be indexed. | Table |
| **Branches to Exclude** | Enter the list of the branches to exclude from indexing. | Free text |

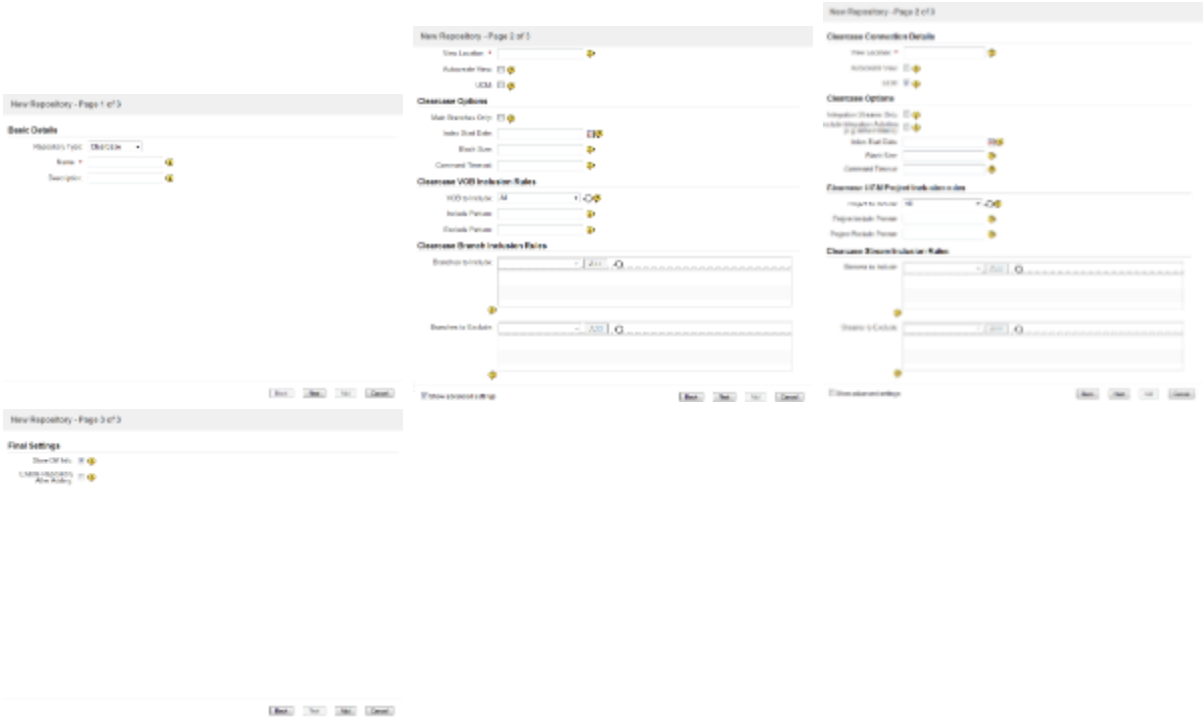
If you have selected **'UCM' to be 'Yes'** (i.e. you are configuring a UCM ClearCase repository) complete the following fields:

| Field | Description | Allowed values |
|---|---|---|
| **Integration Streams Only** | Choose whether FishEye and Crucible should index changes made on development and integration streams or only integration streams. We recommend that users choose 'Yes' for this option. | Yes / No |
| **Include Integration Activities** | Choose whether to include integration activities, i.e. when enabled, rebase and deliver activities will also be indexed. | Yes / No |
| **Index Start Date** | Enter an index start date, if you only want changes that were made after this date to be indexed. Note, files that have not changed since the index start date will not be displayed at all. | Date (YYYY-MM-DD) |
| **Block Size** | Enter the block size, i.e. how many change sets FishEye and Crucible will process in one batch.<br>*(This setting only displays if you tick the '**Show advanced settings**' checkbox at the bottom of the dialogue.)* | Number |
| **Command Timeout** | Enter how long you want FishEye and Crucible to wait for ClearTool commands to complete, e.g. "100000". If not set, this will default to "3600000".<br>*(This setting only displays if you tick the '**Show advanced settings**' checkbox at the bottom of the dialogue.)* | Number (in milliseconds) |
| **Project To Include** | A drop down list displaying all the UCM Projects found in the ClearCase installation. If users only require that FishEye index a single UCM Project, they should select the Project to index from this drop down list. If '**Auto Create View**' is set to 'False' (i.e. using an existing view), you must select a single project not 'All', as a view can only be used for a single UCM project. | Auto populated |
| **Project Include Pattern** | If you need FishEye to index *multiple UCM Projects*, use a pattern to specify which UCM Projects to include. Multiple inclusion patterns can be separated with a comma. See the Inclusion/Exclusion Settings section for examples of inclusion patterns. | Free text |
| **Project Exclude Pattern** | If you need FishEye to index *multiple UCM Projects*, use a pattern to specify which UCM Projects to exclude. Multiple exclusion patterns can be separated with a comma. See the Inclusion/Exclusion Settings section for examples of exclusion patterns. | Free text |
| **Streams to Include** | Enter the list of the UCM streams to include in indexing. An empty list will cause all streams to be indexed. | Free text |
| **Streams to Exclude** | Enter the list of the UCM streams to exclude from indexing. | Free text |

***3. Enter the Final Settings***

| Field | Description | Allowed values |
|---|---|---|
| **Store Diff Info** | Choose whether to enable the Store Diff Info setting. See Store Diff Info for more information on this setting. | Yes / No |
| **Enable Repository After Adding** | Choose whether the repository will be accessible in FishEye immediately. | Yes / No |

*Screenshots: Adding a ClearCase Repository (click to view gallery)*



Step 1
Step 2a (Base)
Step 2b (UCM)
Step 3

### *Notes*

#### *Inclusion/Exclusion Settings*

The following points provide guidelines for the settings which may need to be applied in order to restrict the number of ClearCase Projects/VOBs indexed by FishEye:

- If you want all the VOBs/UCM Projects within your environment to be indexed, then you don't need to add any additional information on the Edit Repository screen. This is the default behaviour.
- If you want several VOBs/UCM Projects to be included (but not all), then you should include appropriate details in the VOB Includes/Excludes fields
- If you only wish for a single VOB/UCM Project to be indexed, then you should select the specific VOB/UCM Project from the 'VOB to Include' or 'UCM Project to Include' drop down list. This will force FishEye to **only** index the selected VOB/UCM Project.

When using the Includes/Excludes fields, you can enter one or more patterns, separated by a comma (note, you cannot use spaces). A pattern can be either a regular expression or a plain string. Matching is done as follows:

- A plain string pattern is considered to match a VOB or UCM Project if the string is part of the VOB/UCM Project (e.g. myvob matches vob:/vobs/myvob and vob:/vobs/myvob2)
- A regular expression is considered to match a VOB or UCM Project if it is the regular expression matches the VOB/UCM Project string (e.g. project:.*_Release.* matches project:Product1_Release2.1@/my_pvob)

For Base ClearCase configurations, the patterns are matched against the VOB name as listed in the select box (e.g. vob:/vobs/yourvob):

| VOB Include/Exclude Pattern | Description |
|---|---|
| /vobs/myvob1,/vobs/myvob2 | matches /vobs/myvob1 and /vobs/myvob2, but also /vobs/myvob12 |
| vob:/vobs/.*vob | regexp pattern that matches /vobs/myvob, /vobs/yourvob, but not /vobs/myvob2 |
| vob:/vobs/.*vob,/vobs/myvob1 | combination of a regexp pattern and a simple VOB name. |

For UCM ClearCase configurations, the patterns are matched against the full UCM Project as listed in the select box (e.g. project:MyUCMProject@/my_pvob):

| UCM Project Include/Exclude Pattern | Description |
|---|---|

| | |
|---|---|
| MyProduct1,MyProduct2 | matches all UCM projects whose name contains MyProduct1 or MyProduct2 (e.g. MyProduct12_Rel3.1) |
| project:MyProduct_Rel\d+.* | regexp pattern that matches MyProduct_Rel1, MyProduct_Rel2, but not MyProduct_CoolNewFeature |
| .@**/my_pvob,.**@/my_otherpvob | regexp patterns that match all UCM projects in the /my_pvob and /my_otherpvob PVOBs. |

#### *View Creation*

As part of the repository scanning logic, FishEye will create a view for each Project (for ClearCase UCM environments) or VOB (for Base ClearCase) using the locations defined in the 'View Location' and 'View Storage Location' fields. This is required in order for the underlying 'cleartool' commands to be executed in the correct context. Please note that FishEye will not perform updates on these views - it is intended that these views will remain unpopulated.

#### *Indexing Logic*

It may be helpful to understand how FishEye's ClearCase support carries out indexing. Please see the following sections:

**UCM ClearCase Indexing:**
The ClearCase support will attempt to index all the available content within a ClearCase environment. The logic works as follows (ClearCase specific terms are underlined see definitions):

- All **PVOBs** that are available are identified.
- For each **PVOB**, find all the **Projects** contained within the **PVOB**.
- For each **Project**, find all the **Streams** associated with the **Project**.
- For each **Stream**, find all the **Activities** that have been delivered to the project.
- Find the **Versions** that were included in each **Activity** and index the **Version** information.
- Any **Labels** attached to **Versions** are also indexed.

PVOB stands for *Project Versioned Object Base*.

**Base ClearCase Indexing:**

The logic for the Base ClearCase support is,

All non-UCM VOBs that are available are identified.

- For each **VOB**, find all the **Branches** that contain versions.
- For each **Branch**, find the check-ins and index the version information.
- Any **Labels** attached to **Versions** are also indexed.

#### *Allocating Time for Repository Scanning*

The initial scan of a repository is a time and resource intensive operation, more so if the ClearCase repository being indexed is large (both in terms of the number of ClearCase projects and the number of change sets included in each project). In the Atlassian test environment (running in a virtual machine), each commit included in a change set would take approximately one second to complete (the time taken in a non-VM environment seems to be slightly faster at approx 700ms). You can use these numbers to estimate the time it will take to scan your repository; it could take many hours or possible days to complete.

### *Changelog*

Changes included in 2.1.3

**Config.xml schema changes:**
The structure of the underlying schema for the ClearCase configuration config.xml file has changed. The effect of this is that for repositories created prior to version 2.1.3, the VOB/UCM Project Inclusion rules won't appear in the Administration UI. However, the previously entered values for these fields will still be used as part of the repository scanning logic.

In order for these fields to be displayed in the Administration UI, the values for these fields should be re-entered.

**Interactive invocation of cleartool commands:**
As a performance improvement measure, a number of the cleartool commands executed by FishEye as part of the repository scanning logic are now executed in 'interactive' mode. That is, a cleartool process (one per repository) is kept open for the duration of the indexing process.

The execution of commands in interactive mode can be disabled by adding a 'disableInteractiveProcess' attribute to the specific ClearCase repository defined in the config.xml file.

**Performance Improvements:**
Subsequent indexing operations for Base ClearCase repositories will take the last indexed date into account, so the 'cleartool lshistory' output will only include those changes that have not already been indexed.

Changes included in 2.1.2

In the first release, the include/exclude rules for VOBs and Projects were handled by the 'Include/Exclude' rules item on the administration page. Based on feedback received during initial version testing, this has been updated to provide additional flexibility:

- The VOBs which are indexed can be controlled via the 'VOB to Include' and 'VOB Include/Exclude Patterns' fields.
- Similarly, the UCM Projects which are indexed can be controlled via the 'UCM Project to Include' and 'UCM Project Include/Exclude Patterns' fields.
- The Include/Exclude rules on the Administration page now apply to files/directories that are indexed within a ClearCase VOB/Project. The values entered into these fields perform the matching logic as defined on the Allow (Process) page

### *Text Types*

From version 2.5.6 FishEye allows you to specify a file which lists the types which are to be considered as text types. If a file 'types.txt' is found in the FISHEYE_INST root directory, FishEye will load this file. Each line in this file is considered to be the name of a Clearcase type which is to be considered a text type. When FishEye detects a file of a type which matches a line in this file, it will remap the type to 'text_file' internally. FishEye will not treat such files as binary.

There is only a single file for all Clearcase repositories in a FishEye instance and all Clearcase repositories will use this file. To pick up changes in this file, a repository must be restarted. The mapping must be in place before FishEye indexes a revision with that type otherwise FishEye will indicate the file is binary. Addition of a type to the list in 'types.txt' after FishEye has indexed revisions of that type will require the repository to be reindexed to pick up that change.

### *Known Issues*

There are a number of known issues with ClearCase support in FishEye. These are listed below.

XML files cannot be viewed as 'Annotated' source

Currently XML files cannot be viewed as 'Annotated' source. By default, ClearCase using a specific type manager to store XML files. This type manager does not support the 'cleartool annotate' command, which is used by the logic in FishEye that displays the Annotated source.

Further to this, by default ClearCase treats any files not defined in the 'default.magic' file as 'compressed' (for instance, property files are not included in the default.magic file). Only text-based type managers can be annotated (and hence, can be displayed via the 'Annotated Source' link). The type manager can be updated by performing the following steps:

1. Update the default.magic file to include appropriate rules that specify the type manager to use for files of a given naming convention (this will take effect for newly created elements)
2. Modify the type manager for existing elements through the 'cleartool chtype' command.

Further information on the ClearCase type manager is available on the following pages:

- Type Manager white paper
- cleartool chtype command reference
- cc.magic reference

Cleartool output limited to 64K of data

There is a known bug with earlier versions of ClearCase that limit the Cleartool output to 64K of data. This may affect projects that contain a large amount of changes included in a changeset. This bug can be fixed by upgrading ClearCase — see this page for more information.

### *Feedback and Support*

Please raise a support ticket to seek assistance with FishEye ClearCase support.

## 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.

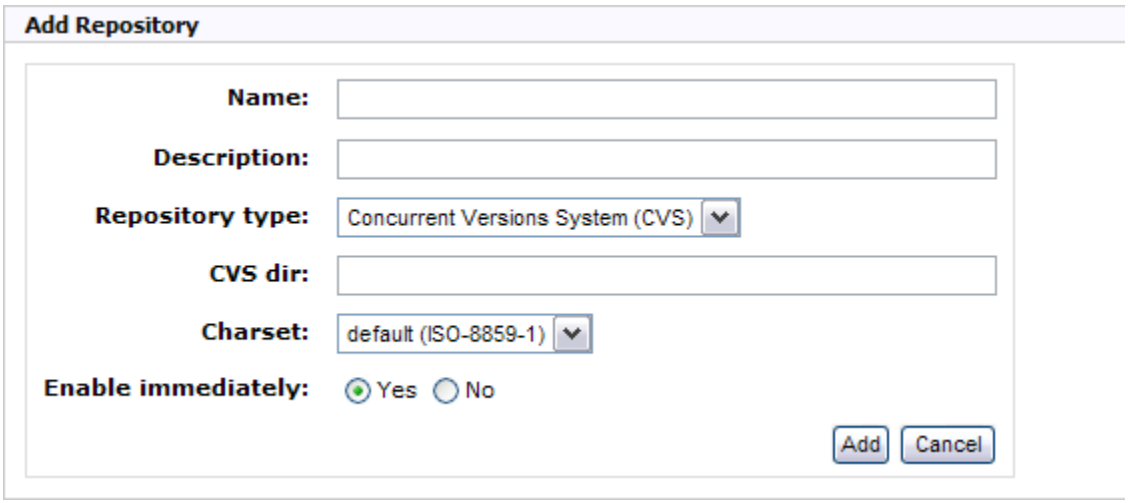
Known Limitations

- 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*




## 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.

**On this page:**



### *Requirements*

- The server running FishEye must have Git (version 1.6 or later) installed.
  *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.*

### *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. |

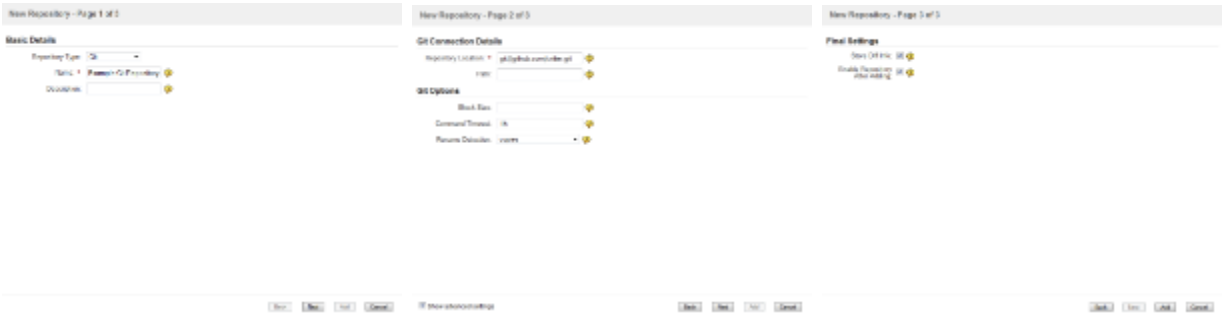
***Step 2***

| 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. See Git URLs in the Git documentation for information on Git URLs. Do not use spaces in your URL. *FishEye does not directly support the ssh protocol for Git repositories. To use ssh, launch FishEye within the context of ssh-agent, to which the appropriate ssh keys have been added.* |
| 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 (see the *--find-copies-harder* option) |

***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*



Step 1
Step 2
Step 3

### *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.

Related Links

- Git Client Configuration
- Crucible Repository Configuration
- General Crucible Configuration

## Mercurial

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

- Define repository details, as described below.
- Set FishEye's repository options.

**On this page:**



### *Requirements*

- Before you add your Mercurial repositories, you will need to set your executable location for hg (Mercurial).

### *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*

Step 1
Step 2
Step 3

### *Notes*

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

Related Links

- Mercurial Client Configuration
- Crucible Repository Configuration
- General Crucible Configuration

## Perforce

When adding or managing a Perforce repository, you must:

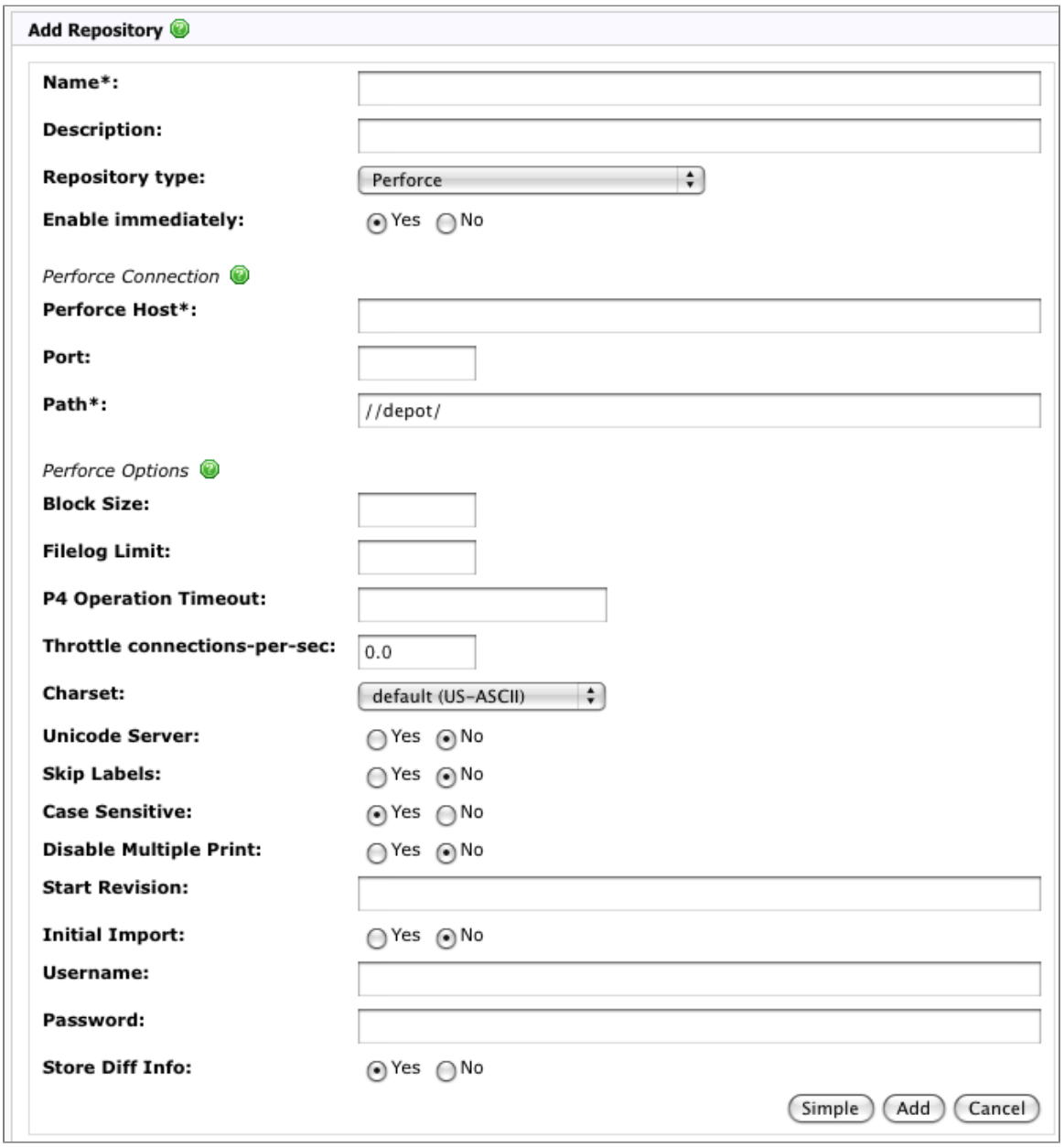
- Define repository details, as described below.
- Set FishEye's repository options.
- Set up a Perforce client.

#### 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. |
| 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.<br>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`. |
| 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.<br>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'.<br>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*




## 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. We recommend using the 'file://' protocol if possible.

**Knowledge Base**
You may find some useful information in the Knowledge Base too.

**SVN Repository Details**

| Detail | Description |
|---|---|
| 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. |
| Repository type | The type of repository - select "Subversion" |
| 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 |
| 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".<br>(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. |
| 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 `inetd/xinetd` to service the `svnserve` protocol. `xinetd` has, by default, an incoming connection limit which can cause FishEye to disrupt other `svnserve`-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 `fisheye.access` 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 `fisheye.access` 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.<br>'**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.<br>'**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. |

| | |
|---|---|
| Username/Password | The credentials to use if your repository requires authentication. |
| 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.

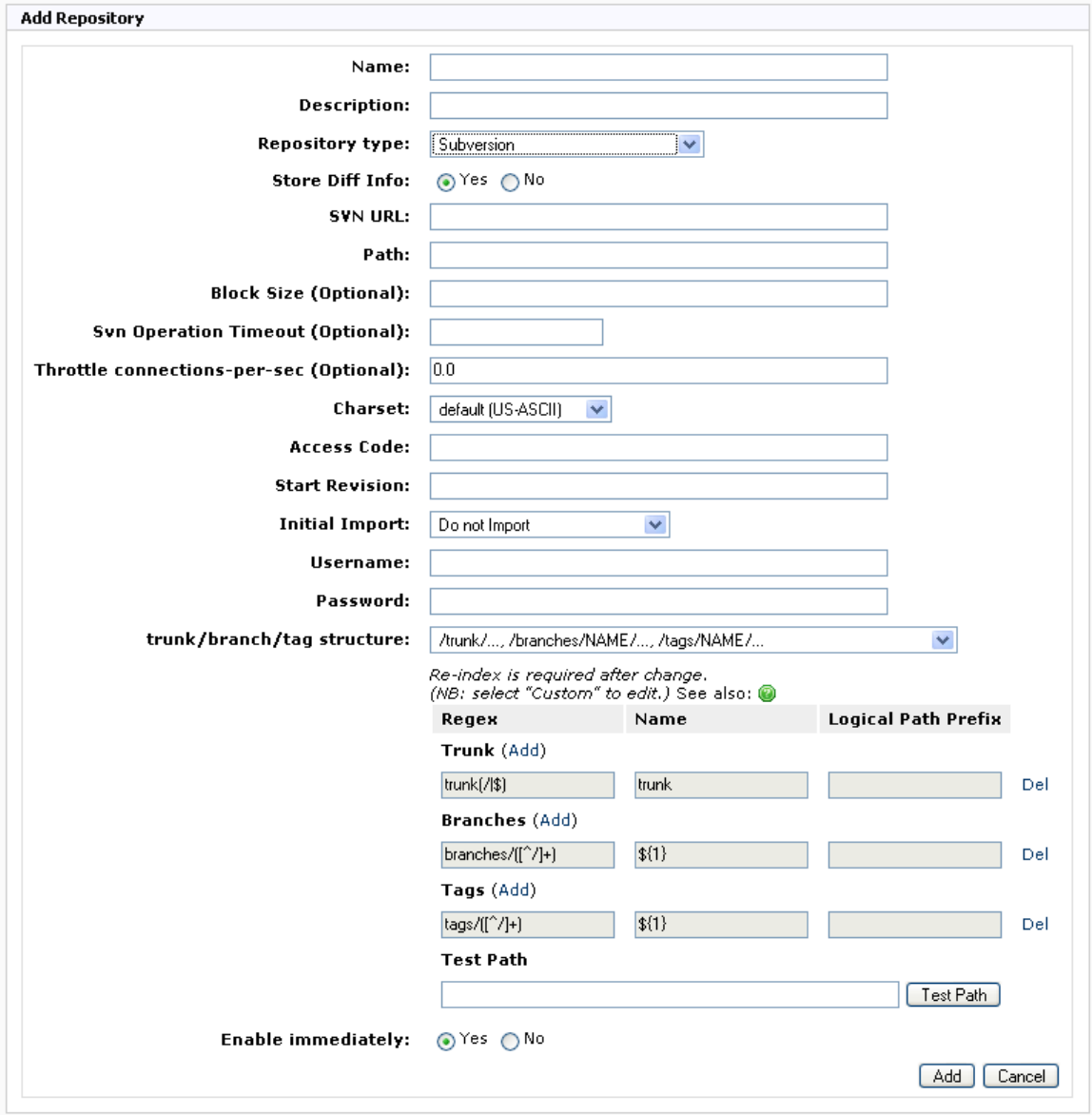
You will get something like the following:

```
>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.

*Screenshot: Adding a SVN Repository*




## 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: `fisheye.access` |
|---|---|---|
| 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.

*On this page:*



### *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.

***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.

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 flowchart below for answers:

**Flowchart: Choosing the correct configuration method**

**1.** If you point FishEye at your project root:

```
/PROJECT1
```

-and you have simply the following:

```
trunk
branches
tags
```

-at this level of the repository, 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:

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

**3.** Otherwise, you have a custom layout.

***Custom layouts***

> 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 How Subversion Works 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(/\|$)` | `trunk` | N/A |
| `branches/([^/]+)` | `${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/([^/]+)/branches/([^/]+)` | `${1}-${2}` | `plugin_${1}` |
| tags | `core/tags/([^/]+)` | `core-${1}` | `core` |
| tags | `plugins/([^/]+)/tags/([^/]+)` | `${1}-${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/([^/]+)` | `${1}` | N/A |
| `branches/([^/]+)/([^/]+)` | `${1}-${2}` | N/A |
| `tags/([^/]+)/([^/]+)` | `${1}-${2}` | 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.

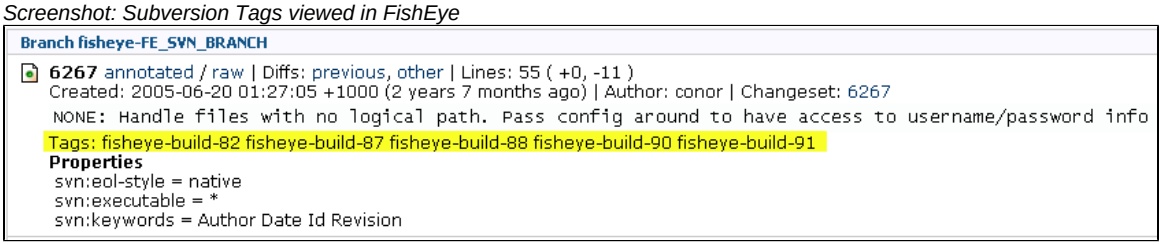
Discovering your Subversion Tag & Branch Structure

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.

## 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

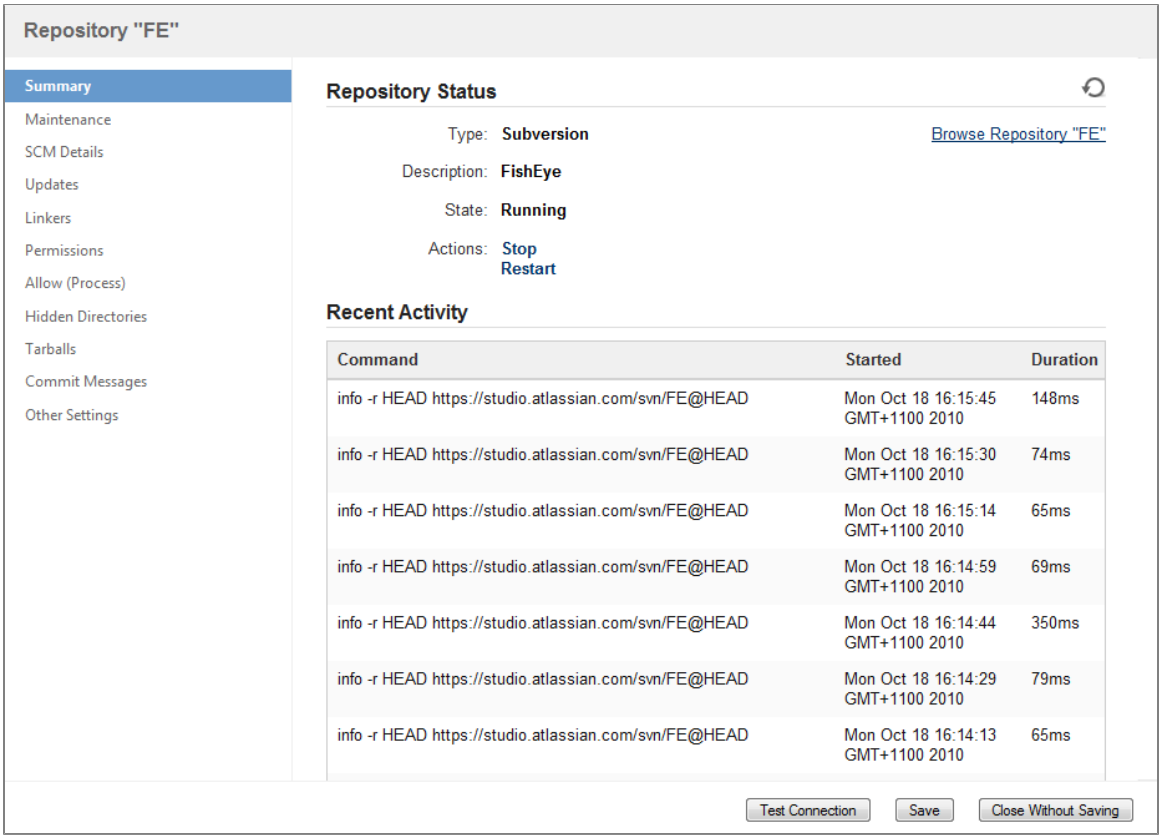
**To configure options for a specific repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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 Allow (Process).
        - 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*



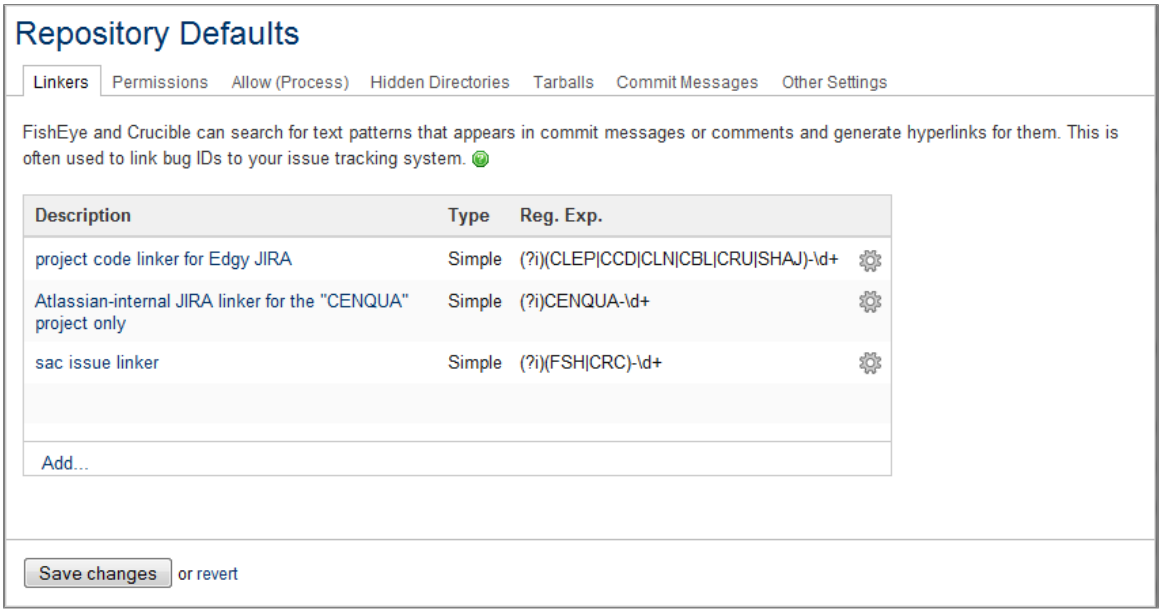

## Configuring Options Affecting All Repositories

**To configure options affecting all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

2. Click the '**Defaults**' link under the '**Repository Settings**' section in the '**Admin**' menu. The 'Repository Defaults' screen will display with the 'Linkers' tab shown.
3. Click the relevant tab to configure the desired repository default setting:
    - Linkers — See Linkers.
    - Permissions — See Permissions.
    - Allow (Process) — See Allow (Process).
    - Hidden Directories — See Hidden Directories.
    - Tarballs — See Tarball Settings.
    - Commit Messages — See Commit Message Syntax.
    - Other Settings — See Watches and Properties.
4. 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*




***Related Topics***

- Allow (Process)
- Authentication
- Commit Message Syntax
- Configuring Repository Details
- Hidden Directories
- Indexer
- Linkers
- Operations
- Permissions
- Properties
- Store Diff Info
- Tarball Settings
- Updater
- Watches

## Allow (Process)

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 'Allow (Process)' repository option.

**On this page:**



### *Configuring Allow (Process)*

You can configure the allow (process) option for a specific repository or configure linkers for all repositories.

#### *Configuring Allow (Process) for a Specific Repository*

**To configure the allow (process) option for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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 '**Allow (Process)**' tab. The 'Allow (Process)' 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.

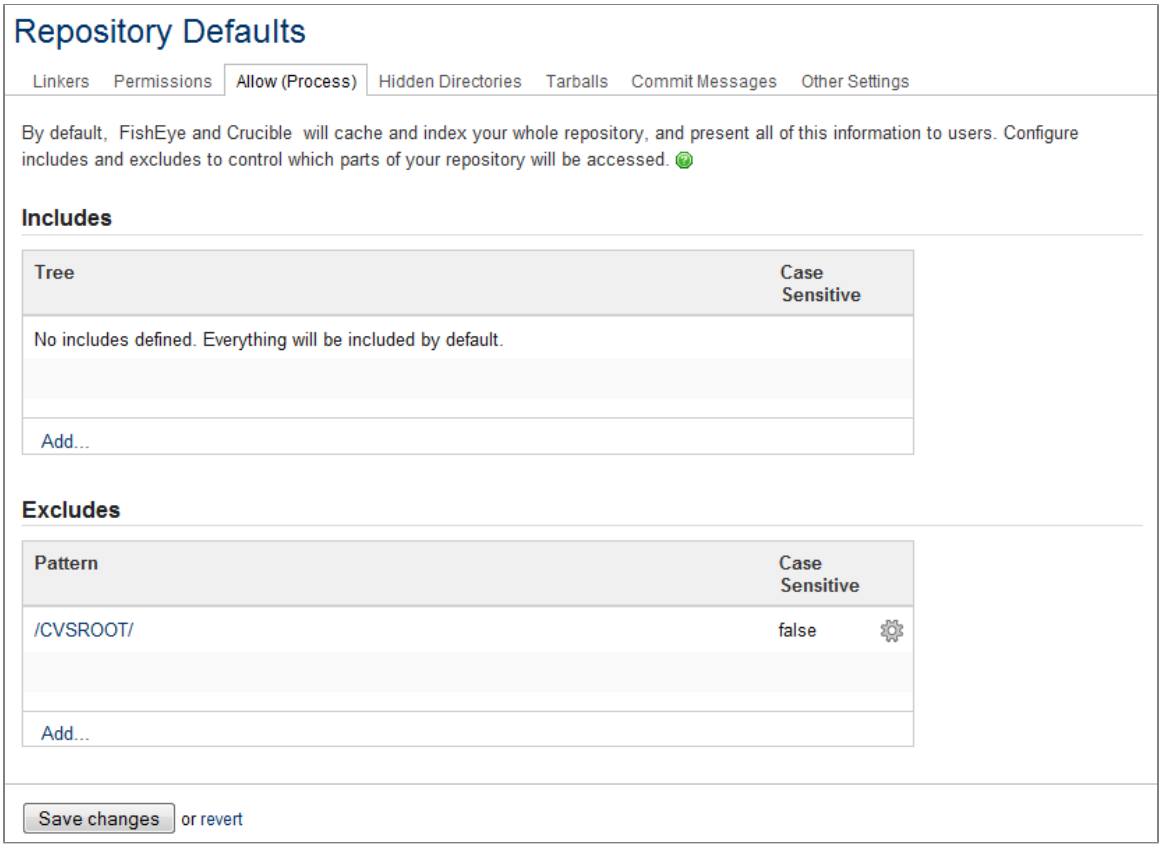
***Configure Allow (Process) for All Repositories***

**To configure the allow (process) for all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

2. Click the '**Defaults**' link. The 'Repository Defaults' will be displayed.
3. Click the '**Allow (Process)**' tab. The 'Allow (Process)' 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 allow (process) for all repositories*




### *Adding an Include*

The '**Includes**' subsection of the '**Allow (Process)**' 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 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:
   - '**Tree**' — 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 '**Allow (Process)**' 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 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:
   - '**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.

## 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.

*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*

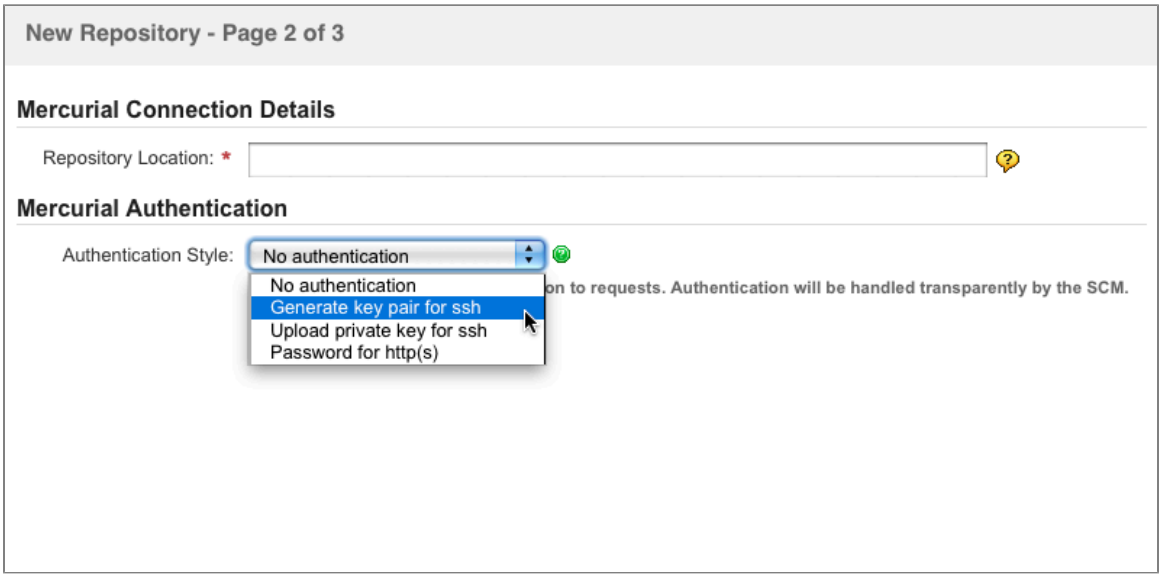
The authentication styles are configured as part of a particular repository definition.

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 "Selecting an Authentication Style" 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.

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.

*Screenshot: No Authentication selected for a local repository*

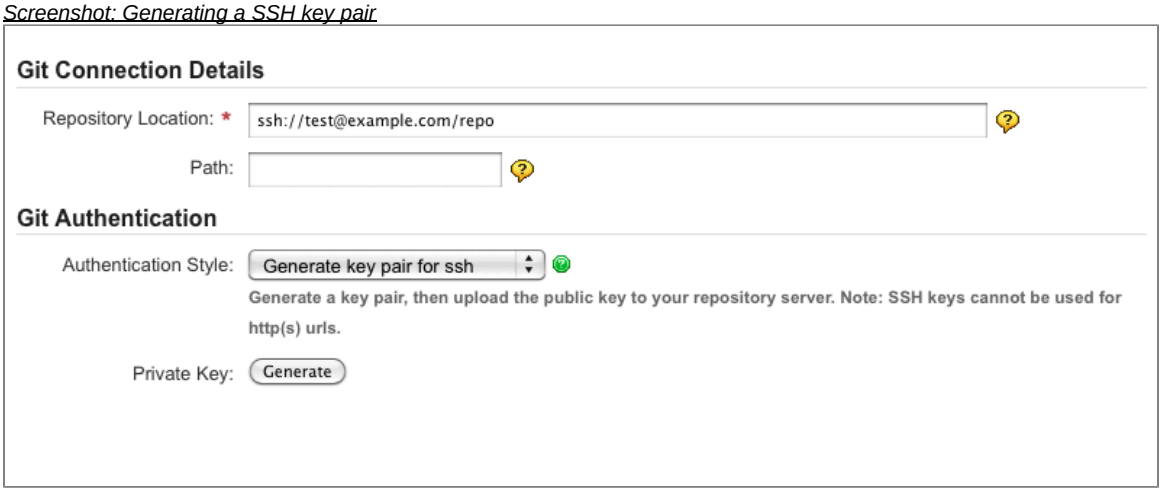

- If you specify "No Authentication" but enter a repository URL which includes a username and password, FishEye will remove the password from the URL and set the authentication style to "Password", storing the password specified in the password field.

#### *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.

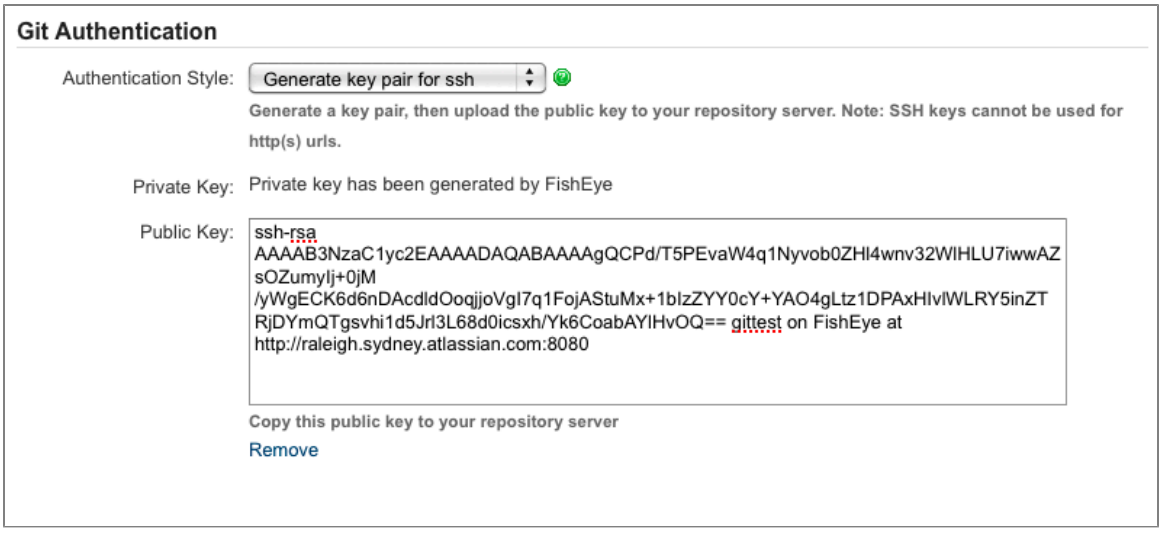
Click the '**Generate**' button to generate the key pair (see "Generating a SSH key pair" screenshot below). FishEye will generate and store the public and private key pair. The key is specific to the repository being indexed.

*Screenshot: Generating a SSH key pair*




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 admininstrators.

*Screenshot: Public Key of Generated Key displayed*

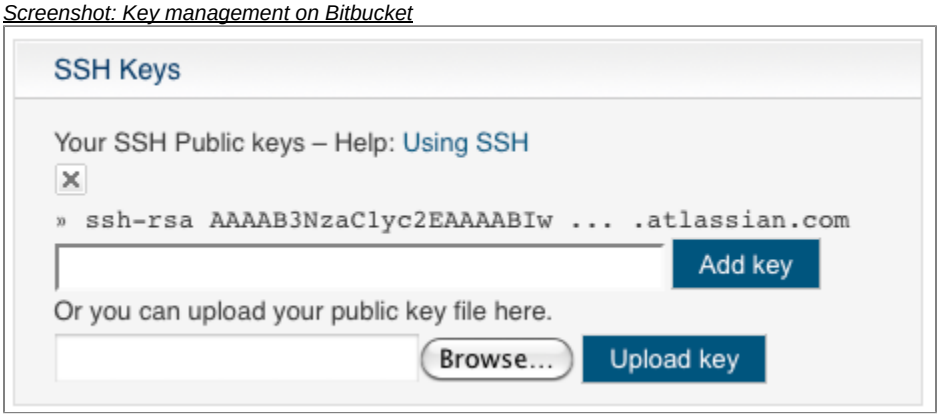



If you wish to change the key, you can remove the existing key by clicking the '**Remove**' link and then generate a new key.

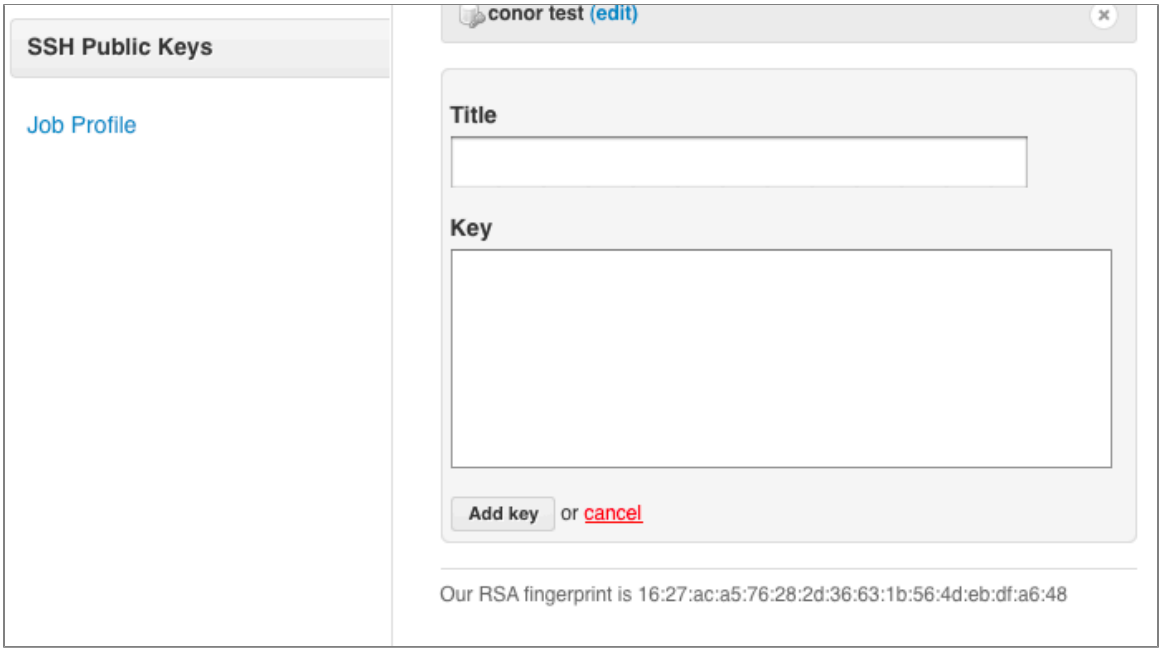
When using SSH keys, you will typically specify a username as part of the URL you use to access the repository.

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*




*Screenshot: Key management on GitHub*



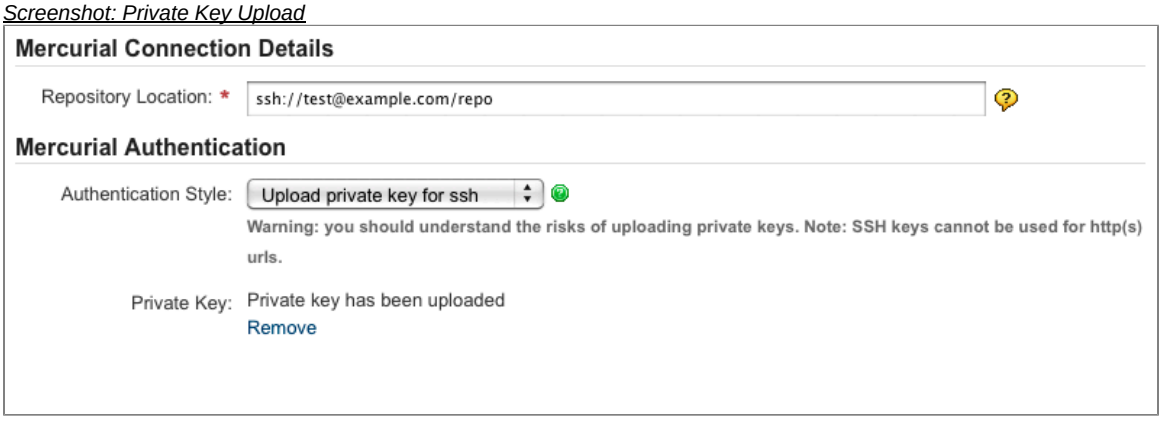

***Upload Private Key for SSH***

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*

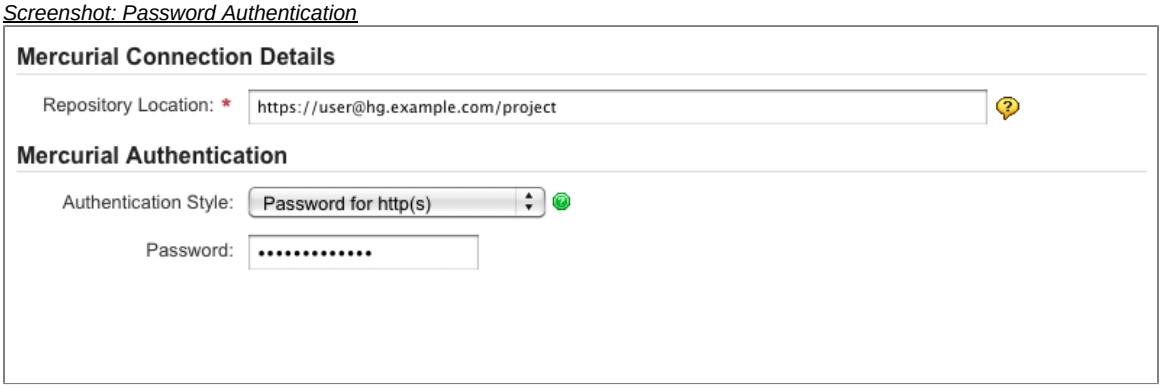



***Tip:*** *If you are uploading under OSX and using 10.6 or later, you can press command-shift-period to display hidden files and directories. This makes it easier to access ssh key files stored in their default .ssh directory*

***Password***

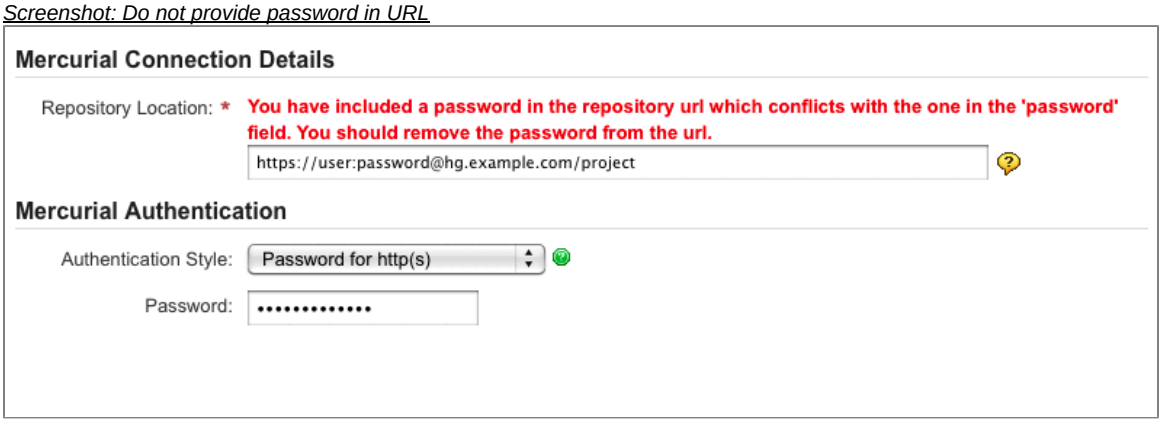
This authentication style is used when using http/https to access your repository with a username and password. For these repositories, the repository URL includes the username for the account and the password is configured in the '**Password**' field (see "Password Authentication" screenshot below).

*Screenshot: Password Authentication*




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*




- 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.

## 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:**

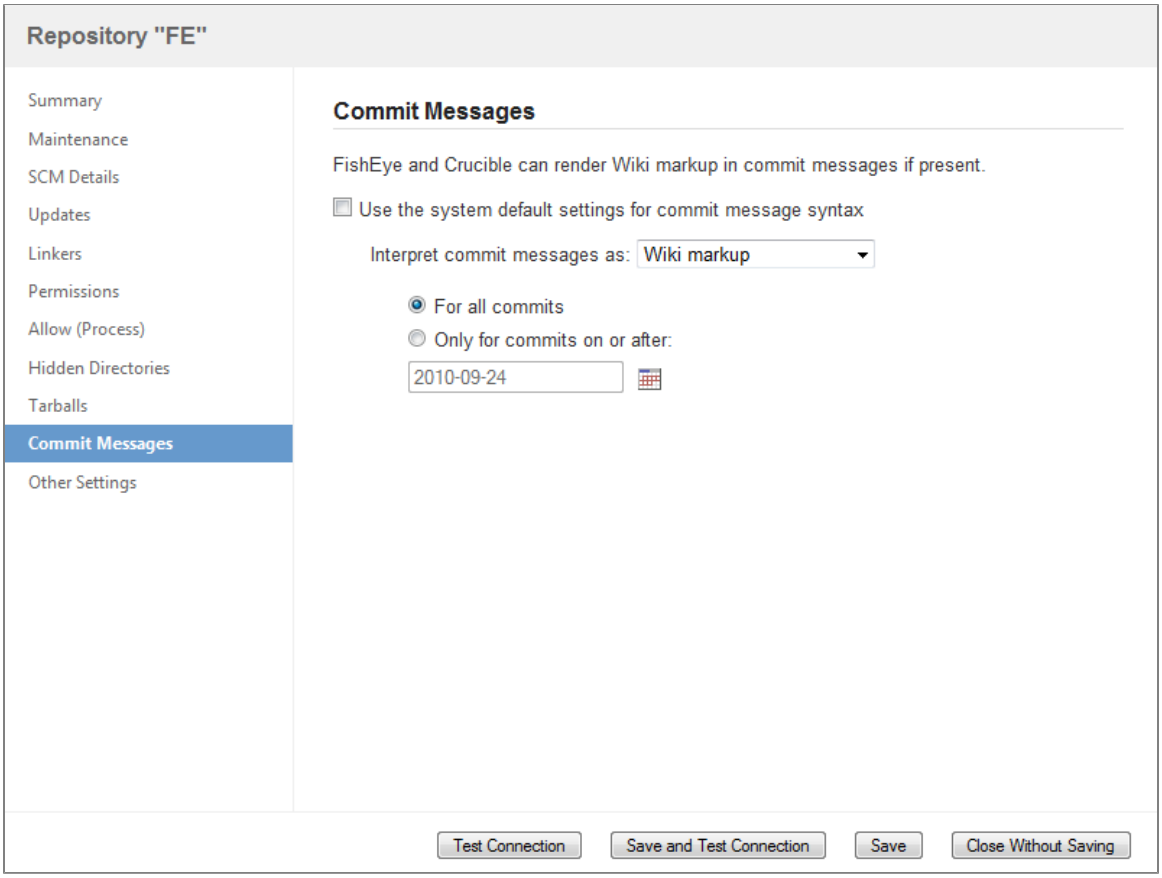


### *Configuring Commit Message Syntax for a Specific Repository*

**To configure commit message syntax for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.
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*



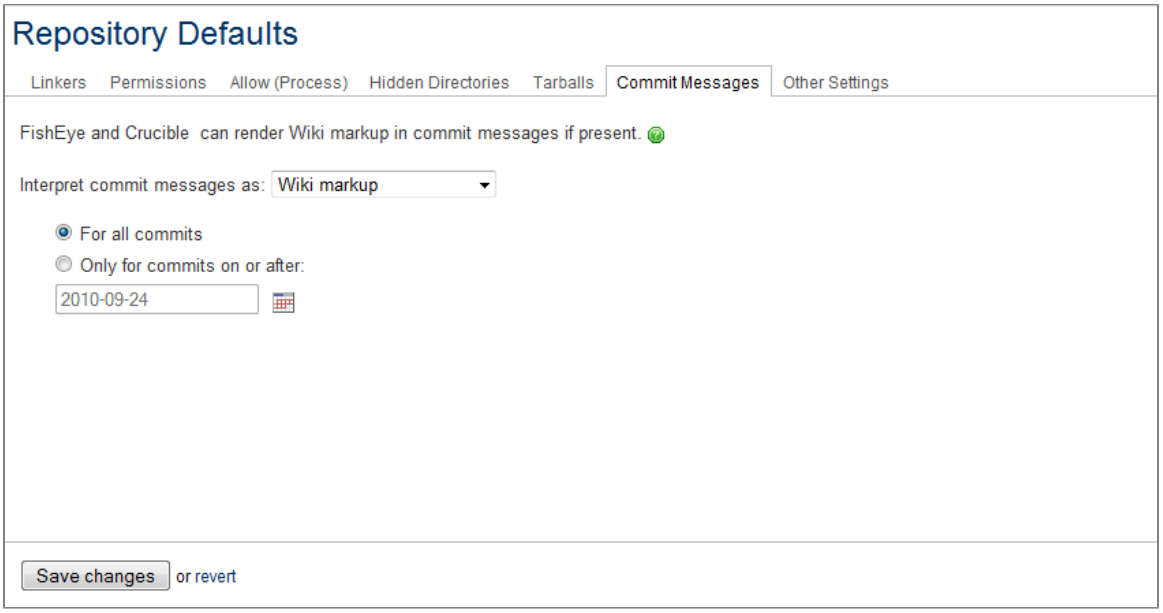

### *Configure Commit Message Syntax for All Repositories*

**To configure commit message syntax for all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

2. Click the '**Defaults**' link. The 'Repository Defaults' will be displayed.
3. Click the '**Commit Messages**' tab. The 'Commit Messages' screen will be displayed (see screenshot below).
4. Update the commit message syntax for this repository as follows:
    - Select '**Interpret commit messages as**' to be either 'Plain text' or 'Wiki markup'. his 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.
5. Save your changes.

*Screenshot: Configuring commit message syntax for all repositories*




## 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 ClearCase repository.

**On this page:**



### *Configuring the SCM Details for a Repository*

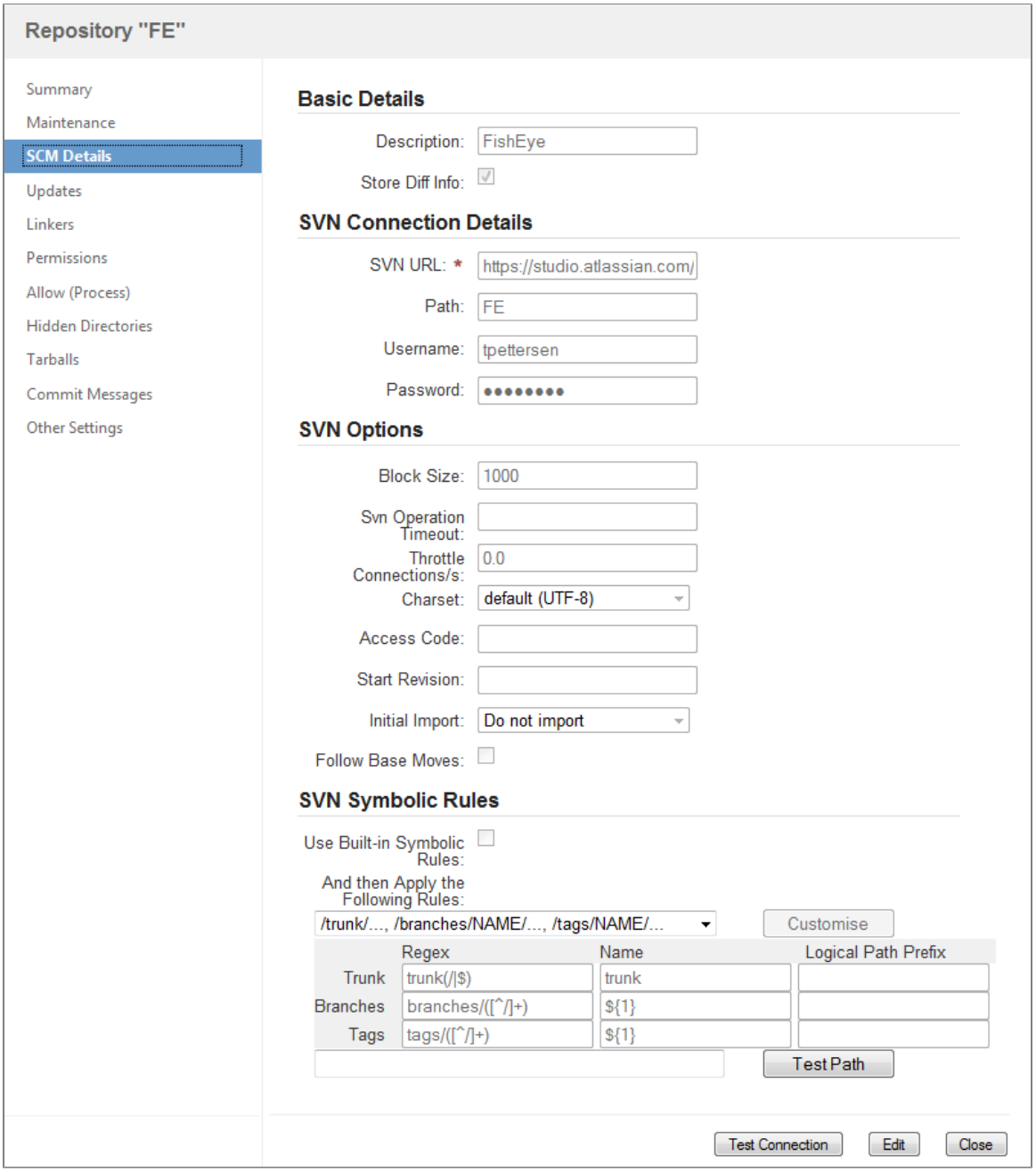
**To configure the SCM details for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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.*
     - ClearCase
     - 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*




### *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**.
- 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.

Related Topics

Adding a Repository
Repository Options

## 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*

**To configure hidden directories for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.
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 the '**Add...**' link 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 click '**Edit**' from the dropdown menu 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 click '**Delete**' from the dropdown menu to delete the hidden directory pattern.
6. Save your changes.

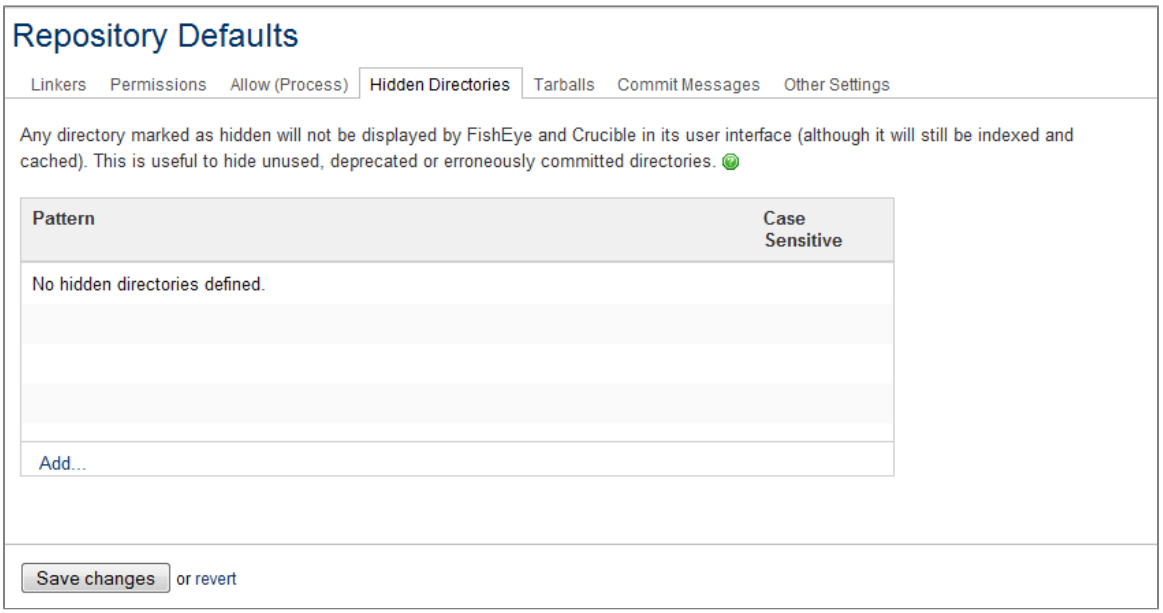
#### *Configure Hidden Directories for All Repositories*

**To configure hidden directories for all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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 click '**Edit**' from the dropdown menu 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 click '**Delete**' from the dropdown menu to delete the hidden directory pattern.
5. Save your changes.

*Screenshot: Configuring hidden directories for all repositories*

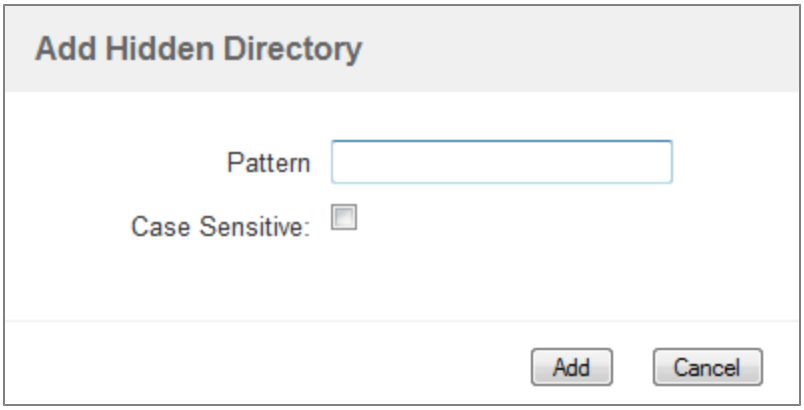



### *Adding a Hidden Directory*

**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 the '**Add...**' link 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 the '**Add**' button to add the hidden directory pattern.

*Screenshot: Adding a new hidden directory pattern*




## Indexer

You can manually perform a number of indexing functions on your repositories in FishEye, such as re-indexing a repository or re-scanning (indexing) a repository.

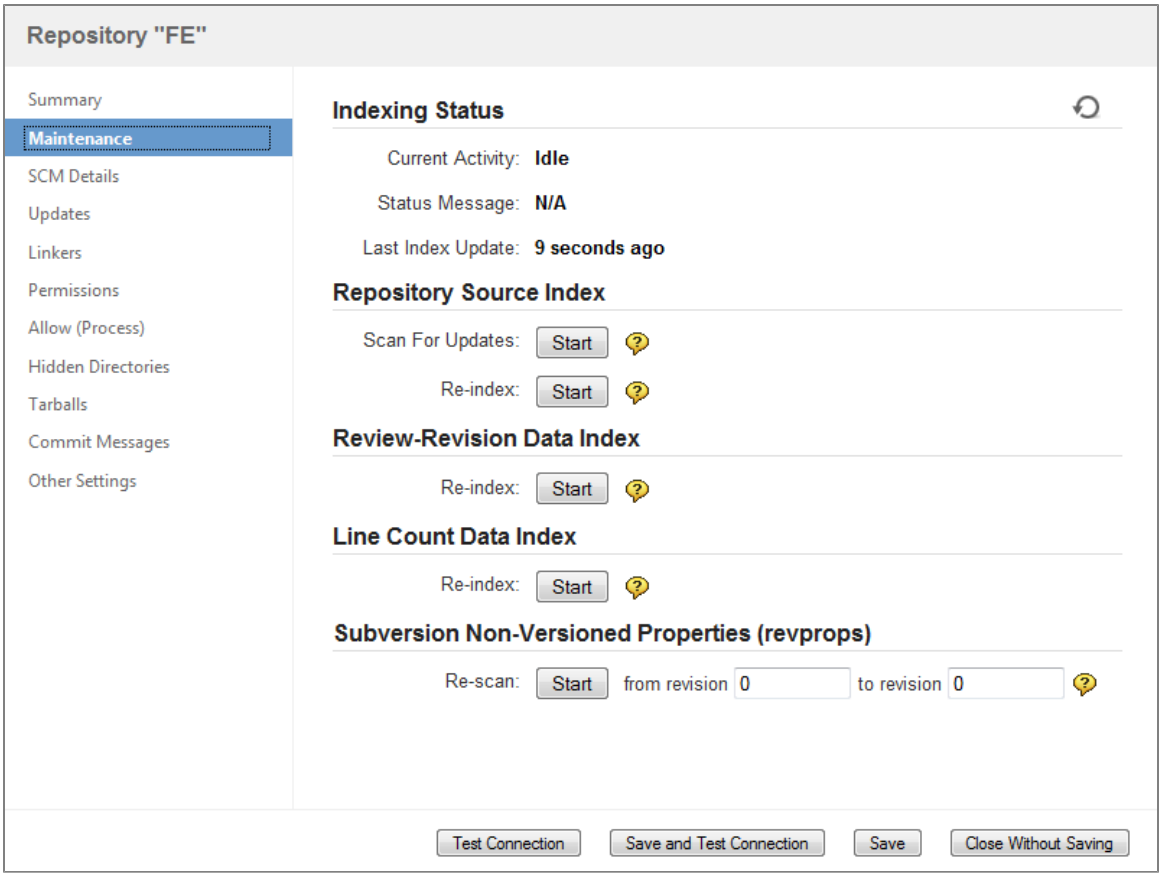
Note, an indexing scan for a repository can take hours or possibly days, depending on the size of the repository, network speed, machine performance and other factors.

**To access indexing functions for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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 '**Maintenance**' tab. Indexing information for the repository will be displayed as well as indexing functions (see screenshot below).
    - '**Repository Source Index**':
        - Click the '**Start**' button next to '**Scan for Updates**' to run a repository scan (also referred to as indexing) now. If the repository has already been indexed, it will be 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.
        - Click the '**Start**' button next to '**Re-index**' to delete the current cache and re-index the repository from the beginning. This action will also restart the repository.
    - '**Review-Revision Data Index**' — *(applies only when using Crucible with FishEye)* Click the '**Start**' button next to '**Re-index**' to re-index all the Crucible review data associated with the current repository.
    - '**Line Count Data Index**' — Click the '**Start**' button next to '**Re-index**' to 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)* Set the revision numbers to scan from and to, and click the '**Start**' button next to '**Re-index**' to rescan non versioned properties (revprops). 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*




## 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*

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 menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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.

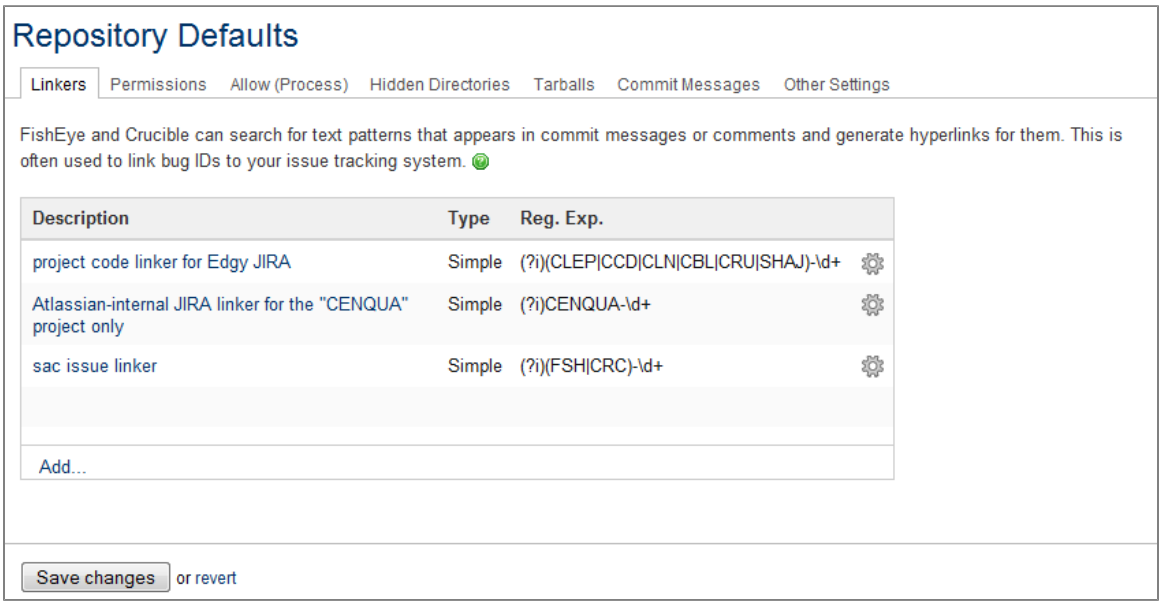
***Configure Linkers for All Repositories***

**To configure the linkers for all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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*




### *Adding a Linker*

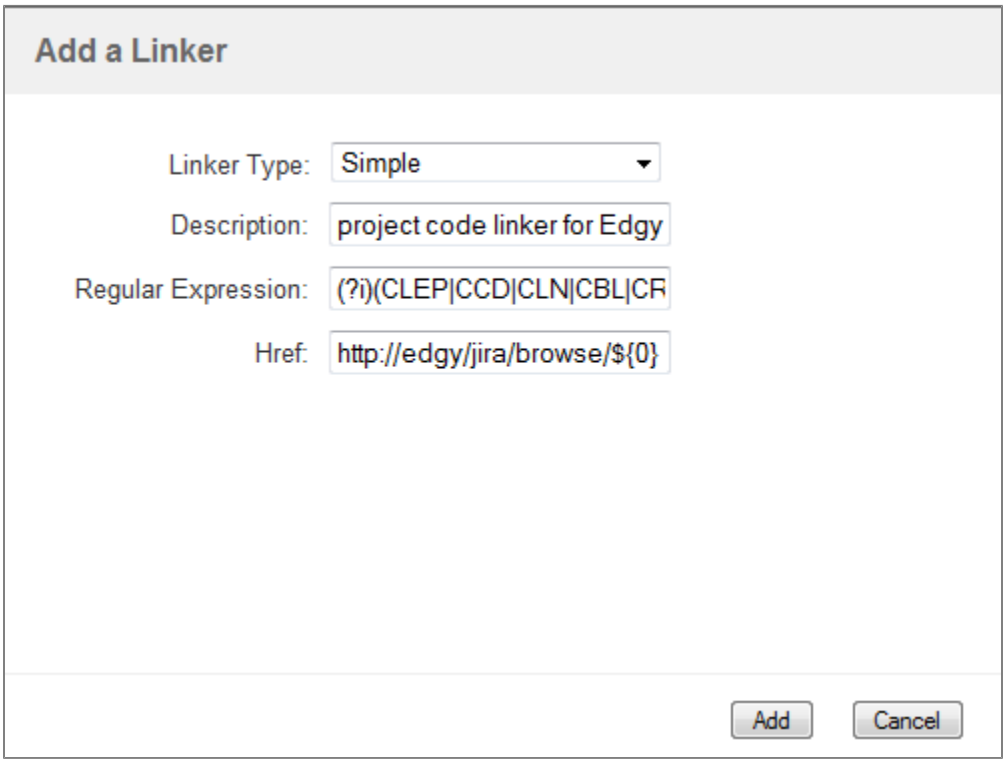
**To add a new linker,**

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*




### *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 isssue 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 seperated 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 menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.
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 choose to allow anonymous access, i.e. anyone can browse your repositories.

**On this page:**



### *Configuring Permissions*

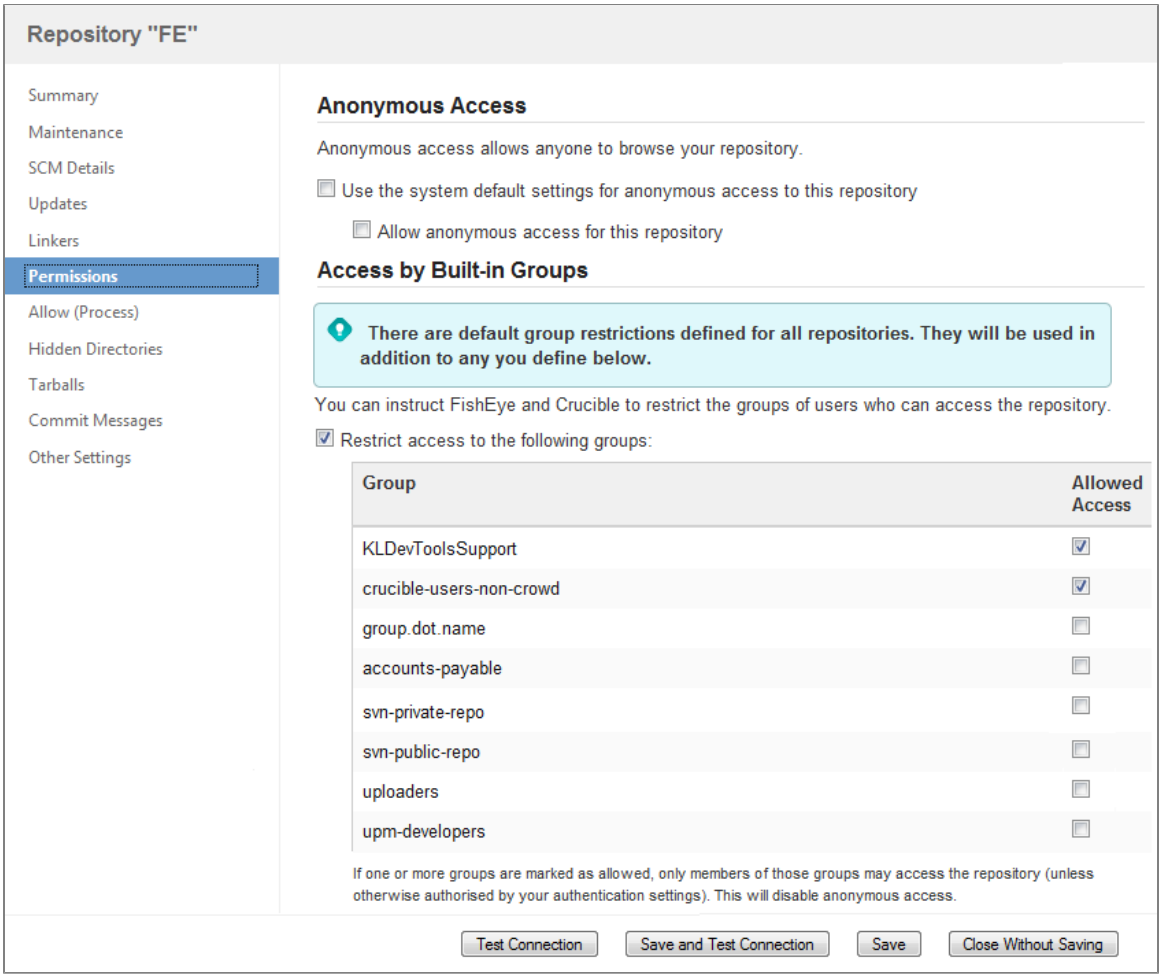
You can configure permissions for a specific repository or configure linkers for all repositories.

#### *Configuring Permissions for a Specific Repository*

**To configure the permissions for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.
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 '**Permissions**' tab. The 'Permissions' screen will be displayed (see screenshot below).
5. Configure the permissions for the repository as desired:
    - Configure anonymous access — See Configuring Anonymous Access below.
    - Configuring user group access — See Configuring User Group Access below.
6. Save your changes.

*Screenshot: Configuring permissions for a specific repository*



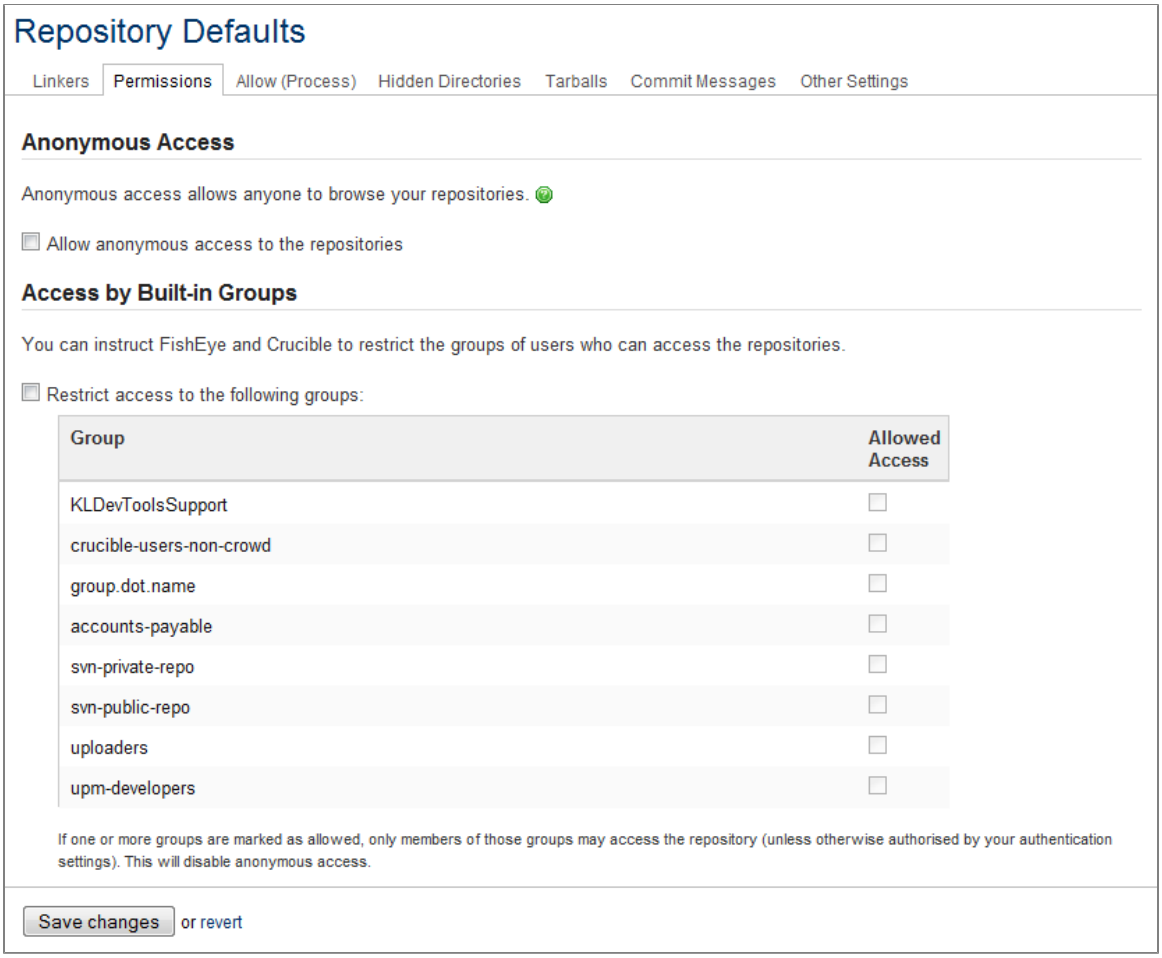

***Configure Permissions for All Repositories***

**To configure the permissions for all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

2. Click the '**Defaults**' link. The 'Repository Defaults' will be displayed.
3. Click the '**Permissions**' tab. The 'Permissions' screen will be displayed (see screenshot below).
4. Configure the permissions for all repositories as desired:
   - Configure anonymous access — See Configuring Anonymous Access below.
   - Configuring user group access — See Configuring User Group Access below.
5. Save your changes.

*Screenshot: Configuring permissions for all repositories*




### *Configuring Anonymous Access*

**To configure the permissions for all repositories,**

1. Navigate to the permissions for a specific repository or the repository defaults, as described above, and update the following fields:
    - '**Use the system default settings for anonymous access to this repository**' *(specific repository only)*--- Tick this checkbox if you want to use the system default settings for anonymous access, i.e. if you allow/disallow anonymous access in your repository defaults, this repository will also allow/disallow anonymous access respectively.
    - '**Allow anonymous access for this repository**' *(specific repository only)*--- Tick this checkbox to allow anonymous access to the specific repository, i.e. anyone can browse your repository.
    - '**Allow anonymous access to the repositories**' *(repository defaults only)* — Tick this checkbox to allow anonymous access to all repositories that been configured to use the system default settings for anonymous access.
2. Save your changes.

### *Configuring User Group Access*

**To configure the permissions for all repositories,**

1. Navigate to the permissions for a specific repository or the repository defaults, as described above, and update the following fields:
2. Tick the 'Restrict access to the following groups' checkbox, if you want to specify access for particular user groups only.
3. Tick the checkboxes next to the groups that you want to give access to the repository(s). User group access for specific repositories is in addition to user group access granted for all repositories (via your repository defaults).
4. Save your changes.

For more information, see Associating a Group with a Repository.

## 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:**

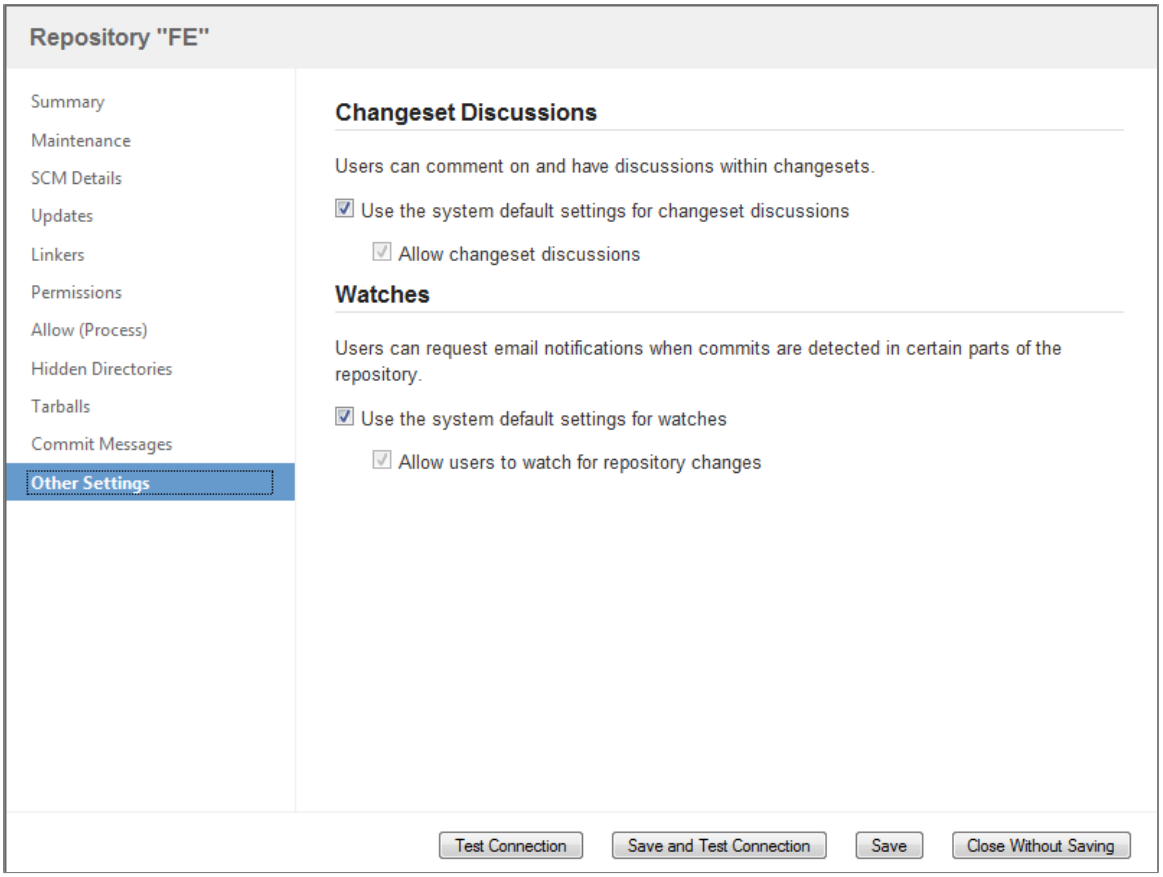


### *Configuring Properties for a Specific Repository*

**To configure properties for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.
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*



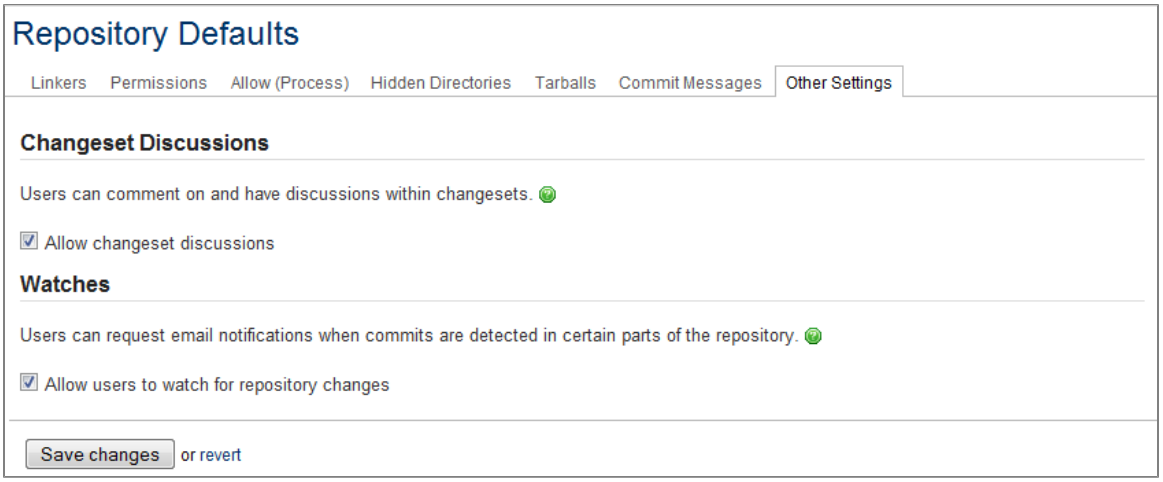

### *Configure Properties for All Repositories*

**To configure properties for all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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*




## 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**.
- 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:**



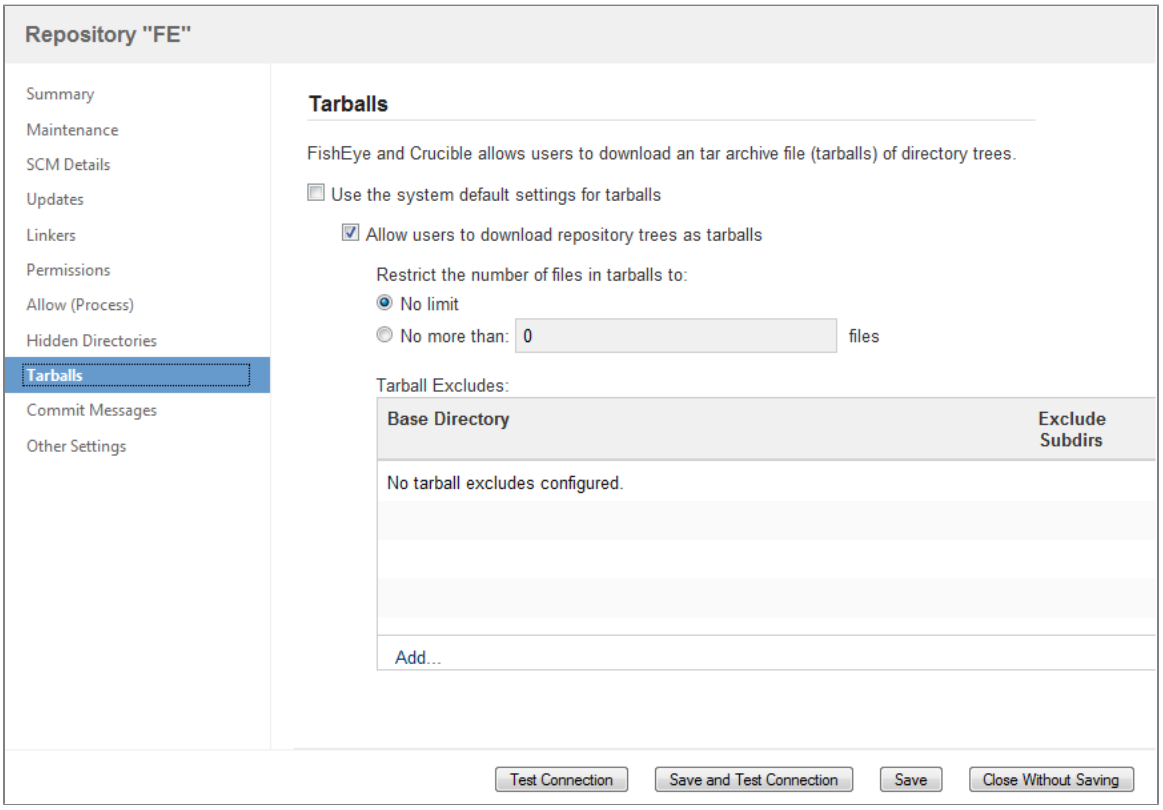
### *Configuring Tarball Settings for a Specific Repository*

**To configure tarball settings for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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*



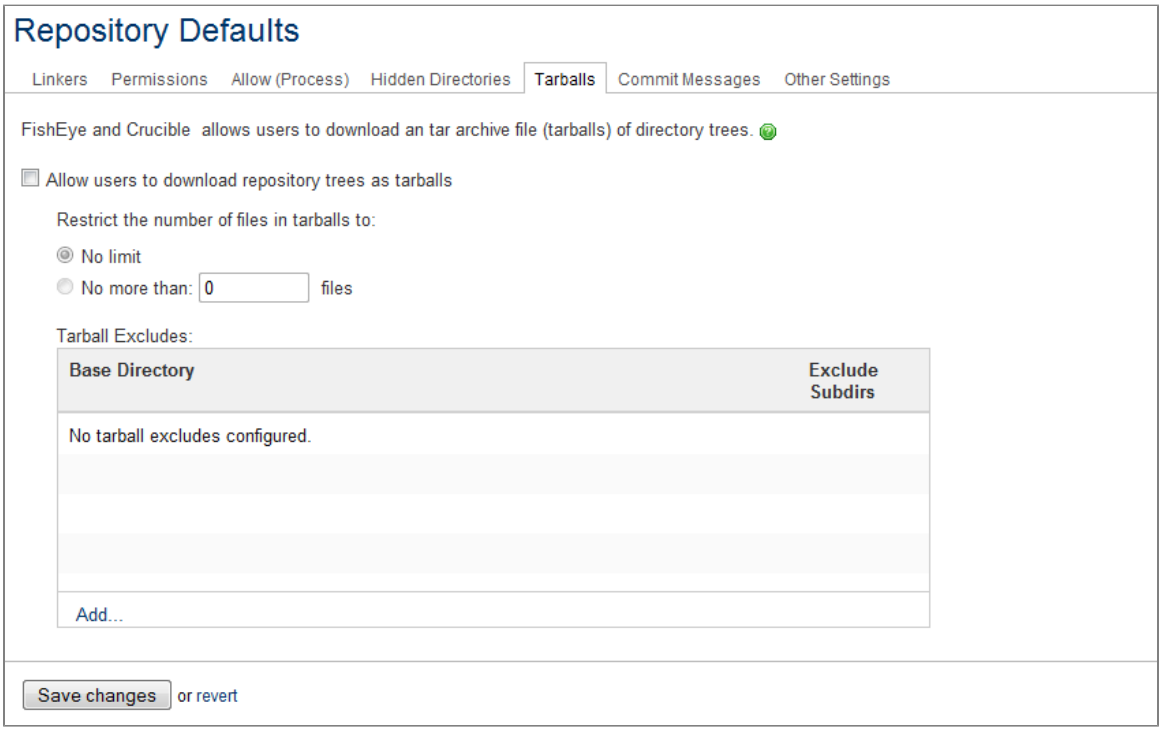

### *Configure Tarball Settings for All Repositories*

**To configure watches for all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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*




## 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,**

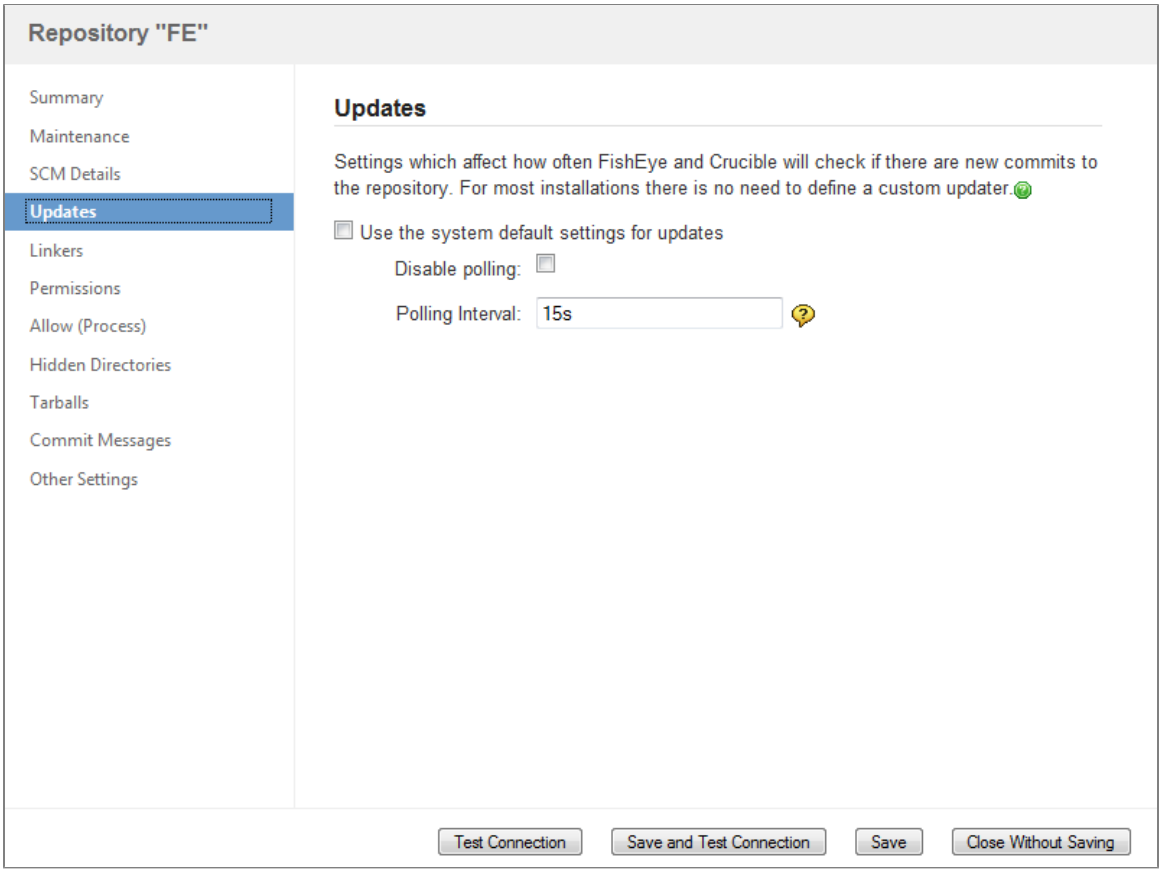
1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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 '**Updates**' tab. The 'Updates' screen will be displayed (see screenshot below).
5. Tick the '**Use the system default settings for updates**' checkbox to use the default updater settings. In most cases, you can leave this ticked. If you want to customise the updater settings, untick the checkbox and complete the following fields, as desired:
    - '**Disable polling**' — Tick this checkbox if you want to disable repository polling altogether. 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 will check to see if there have been any 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:

    > FishEye will monitor your CVS history file `CVSROOT/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.

    - '**History file**' — Specify 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 `./CVSROOT/history`.
    - '**Full scan period**' — Specify 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, `10second`. If you do not specify a unit of time, the default unit is days. For example, `15min`. 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**' — Specify the prefix to strip off files found in the history file. This will make the files relative to this repository's CVS directory.
      You **must** configure a strip prefix if the CVS directory specified is not the root of the CVS repository. For example, your CVS is located at `/usr/local/cvsroot`, but you have specified `/usr/local/cvsroot/foo/bar` as the CVS directory of this repository. In this case, you will need to set the history file as `../../CVSROOT/history` and specify a strip prefix of `foo/bar`.
7. Click the '**Save**' button to save your changes.
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 you polling interval to one hour, your next scan will begin one hour after your initial scan is complete.

*Screenshot: Repository Updates Settings*




## Watches

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.

**On this page:**

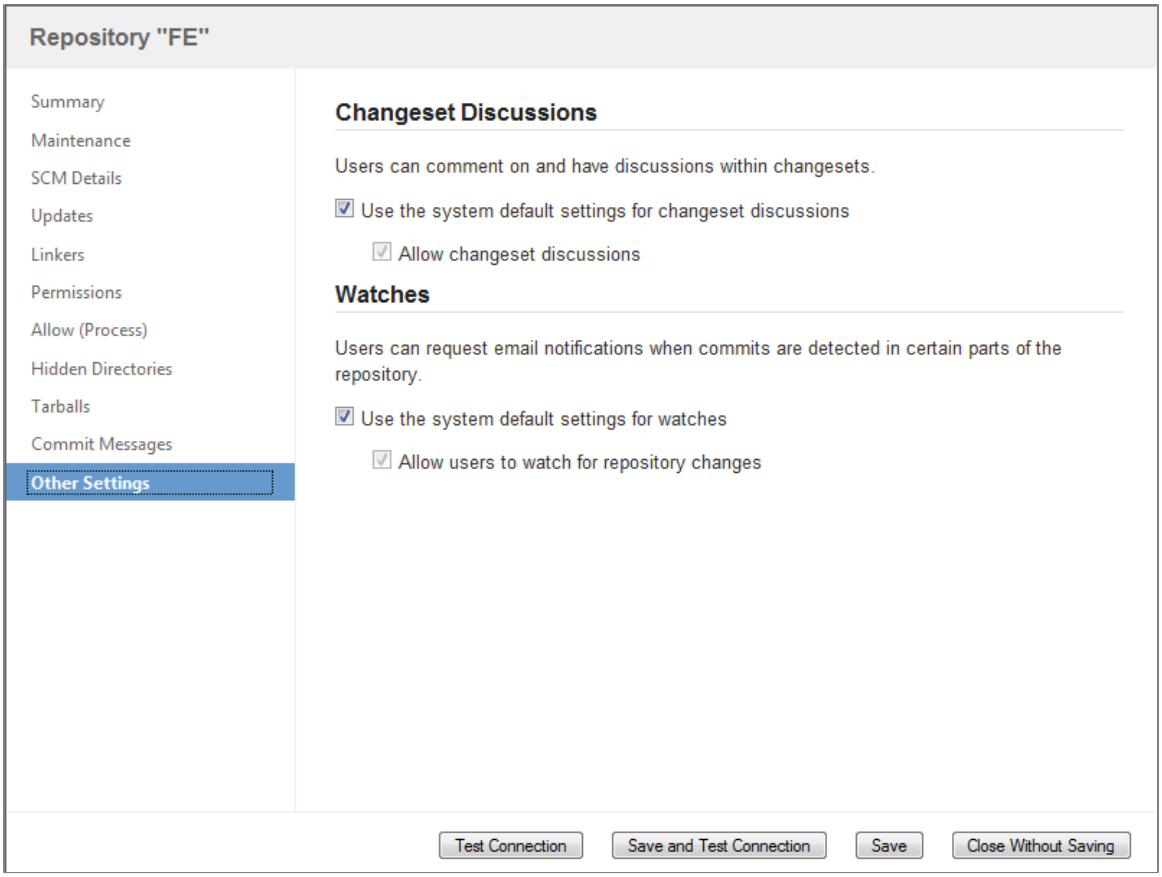


### *Configuring Watches for a Specific Repository*

**To configure watches for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.
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 '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**' — Tick this checkbox 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**' — Tick/Untick this checkbox to enable/disable watches for the repository. This checkbox will only be enabled if the '**Use the system default settings for watches**' checkbox is not ticked.
6. Save your changes.

*Screenshot: Configuring watches for a specific repository*

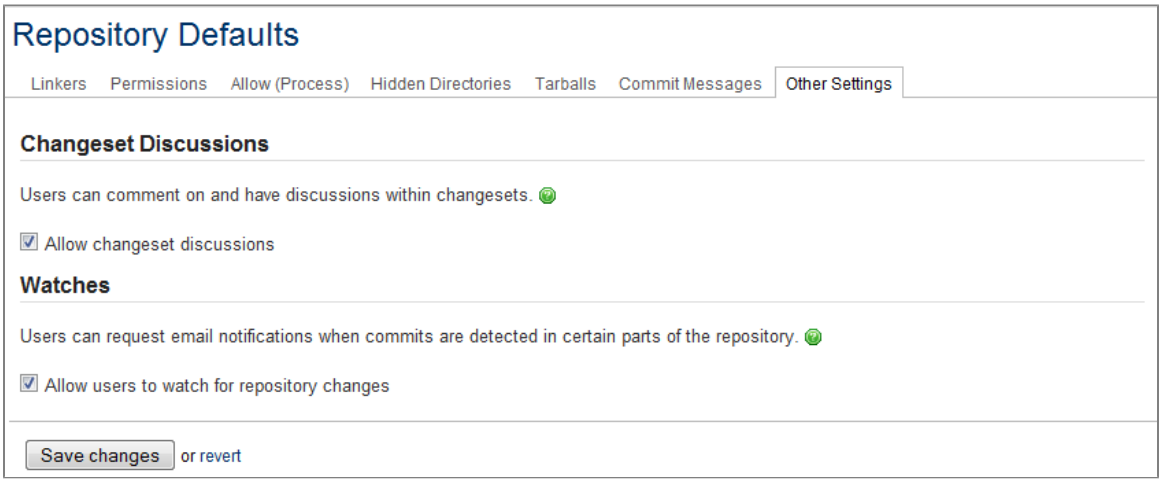



### *Configure Watches for All Repositories*

**To configure watches for all repositories,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.
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**' — Tick/Untick this checkbox to enable/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*



# 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
```

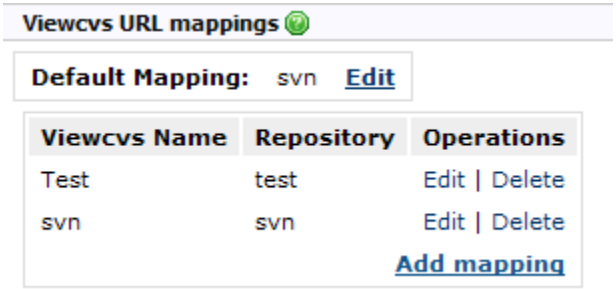
can be viewed in FishEye at

```
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*




Once you have entered the mappings you will need to **restart** Fisheye.

# Configuring SMTP

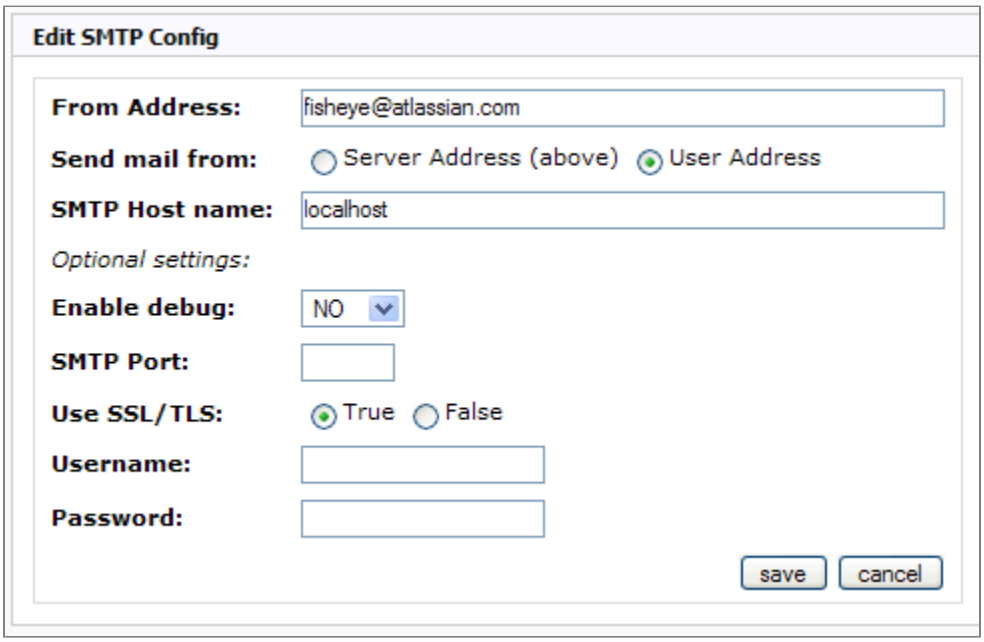
To configure SMTP settings, click '**Server Settings**' on the '**Admin Menu**'.

You can enter the following parameters:

| | |
|---|---|
| **From Address** | The from email address used when FishEye sends an email, e.g. 'fisheye-noreply@example.com' |
| **Send mail from** | Selects either the 'Server Address' (default, as above) or the 'User Address', which selects the email address in the user's profile.<br>(**Note: this only applies to Crucible notifications. FishEye will always use the Server Address.**) |
| **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, defaults to 25. 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. |

Once you have configured SMTP, you can use the '**Send test email**' link on the '**Server Settings**' page to confirm the SMTP connectivity.

*Screenshot: Configuring SMTP*




# 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

Git 1.6 is supported at this stage.

#### Configuration

FishEye requires Git 1.6.0 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

FishEye supports Mercurial version 1.5.1 or later (Python version 2.4.3 or later).

#### Configuration

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

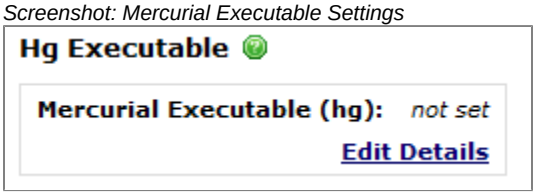
1. Open the FishEye Admin screen.

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.

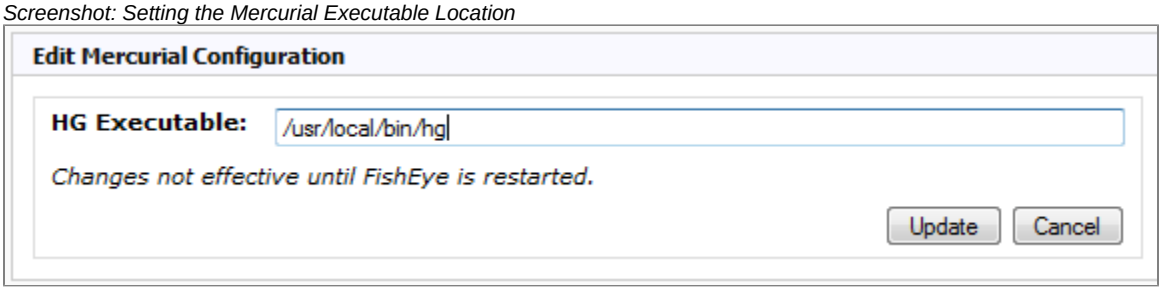
Following this process, you will be able to add your Mercurial repository to FishEye.

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 Settings**' in the FishEye '**Admin Menu**'.

> **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.
>
> 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 `libsvnjavah-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, under **Admin** > **Server Settings** > **Subversion Client**.

| | |
|---|---|
| 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.<br>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"/>
```

### 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

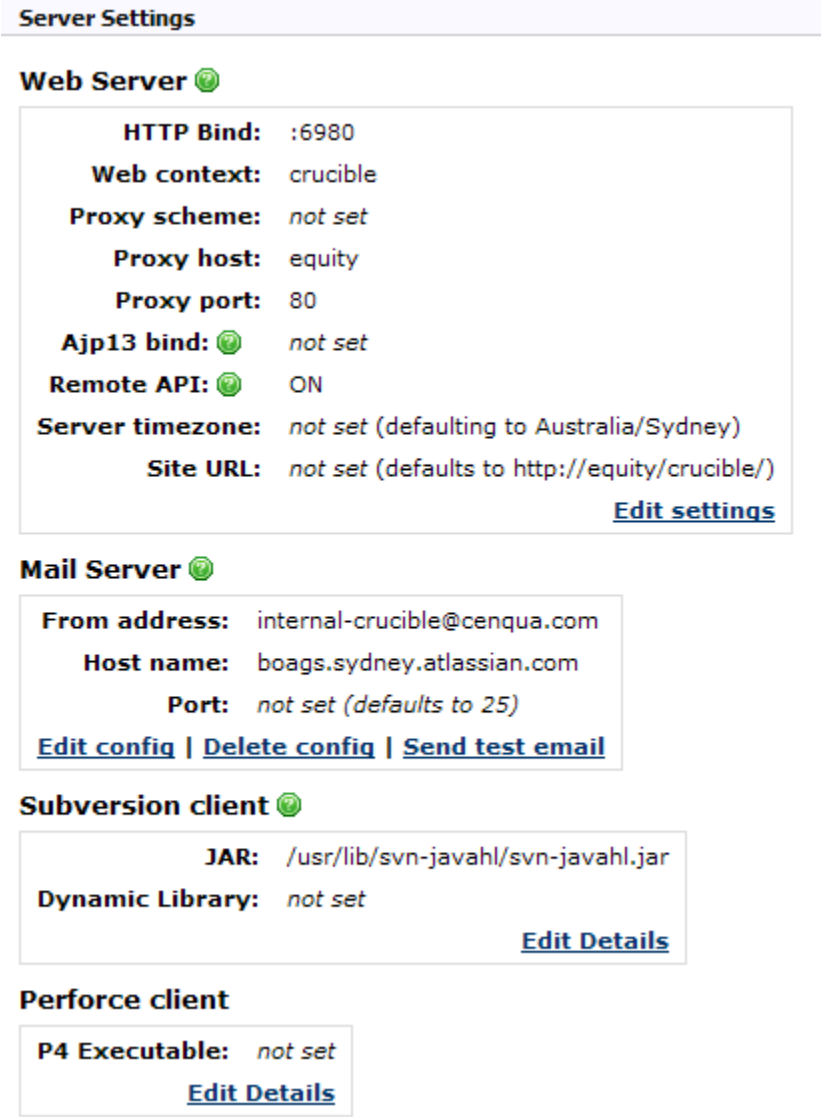
Native Subversion Clients are also supported by FishEye as an alternative to SVNkit. This process requires additional configuration. Read the Native Subversion Client instructions for more information.

## Setting up your Web Server

To configure the server settings, click '**Server Settings**' on the '**Admin Menu**'.

- Configuring the FishEye Web Server
- Integrating with Other Web Servers

*Screenshot: Configuring Server Settings*




## Configuring the FishEye Web Server

To configure the server settings, click '**Server Settings**' on the '**Admin Menu**'.

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.

| 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:<br>• **Host name and port number**:<br>`hostname:8060`<br>• **port number only** (requires a leading colon):<br>`:8060`<br>• **IP address and port number**:<br>`10.0.0.11:60`<br>(*At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.*)<br>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. |
| **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. |
| **Example** | A valid setting would be `www.example.com`, 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. |
| **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:<br>• **Host name and port number**:<br>`hostname:8009`<br>• **port number only** (requires a leading colon):<br>`:8009`<br>• **IP address and port number**:<br>`10.0.0.11:8009`<br>(*At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.*)<br>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 | Remote API |
|---|---|
| **Description** | Enables/disables FishEye's Remote API. Clicking on the help link will take you to the API doc. |
| **Example** | '**On**' enables the Remote API. |
| **Restart required** | No |

| 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. |
| **Example** | This is used whenever you have set FishEye up in a sub-folder on the web server, for example `www.example.com/apps/fisheye`. |
| **Restart required** | No |

See also Subversion Client settings.

## FishEye SSL Configuration

This page contains instructions on using SSL with FishEye.

**On this page:**

### *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 %FISHEYE_HOME%/keystore -alias fisheye
     -genkey -keyalg RSA
     ```

   - **Unix/Linux:**

     ```
     $JAVA_HOME/bin/keytool -keystore $FISHEYE_HOME/keystore -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="keystore" keystore-password="password123"
         truststore="keystore" 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
     %FISHEYE_HOME%\keystore -file %FISHEYE_HOME%\fisheye.csr
     ```

   - **Unix/Linux:**

     ```
     $JAVA_HOME/bin/keytool -certreq -alias fisheye -keystore
     $FISHEYE_HOME/keystore -file $FISHEYE_HOME/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 you have a key and a certificate in separate files, you need to combine them into a PKCS12 format file for loading into a new keystore. `Keytool` does not perform any of these conversions. Therefore, the easiest way to do this is via `openssl`, by running the following command:

   ```
   openssl x509 -in certificate.der
           -inform DER -outform PEM -out certificate.crt
   ```

   to convert between DER and PEM. See these instructions for information on how to convert these separate files into PKCS12 format.

5. Load the certificate into the keystore, as follows:
   - CA certificate in PEM format:
     - **Windows:**

       ```
       "%JAVA_HOME%\bin\keytool" -keystore %FISHEYE_HOME%\keystore
               -import -alias fisheye -file %FISHEYE_HOME%\certificate.crt
               -trustcacerts
       ```

     - **Unix/Linux**:

       ```
       $JAVA_HOME/bin/keytool -keystore $FISHEYE_HOME\keystore
               -import -alias fisheye -file $FISHEYE_HOME/certificate.crt
               -trustcacerts
       ```

   - CA certificate in PKCS12 format:
     - **Windows:**

       ```
       %JAVA_HOME%\bin\keytool -importkeystore
               -srckeystore %FISHEYE_HOME%\certificate.pkcs12
               -srcstoretype PKCS12 -destkeystore %FISHEYE_HOME\keystore
       ```

     - **Unix/Linux:**

       ```
       $JAVA_HOME/bin/keytool -importkeystore
               -srckeystore $FISHEYE_HOME/certificate.pkcs12
               -srcstoretype PKCS12 -destkeystore $FISHEYE_HOME/keystore
       ```

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 web admin and click '**Server**' under the '**Global Settings**' section. The 'Server Settings' page will be displayed (see 'Server Settings' screenshot below).
2. Click the '**Edit Settings**' link in the '**Web Server**' section. The settings will refresh as editable.
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` | 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 the How to Get Legendary Support from Atlassian 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` directive, which requires the `mod_proxy` module. Add this section to your Apache configuration:

```
ProxyPass /fisheye http://FEHOST:8060/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.

# 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 support 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 `FISHEYE_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.

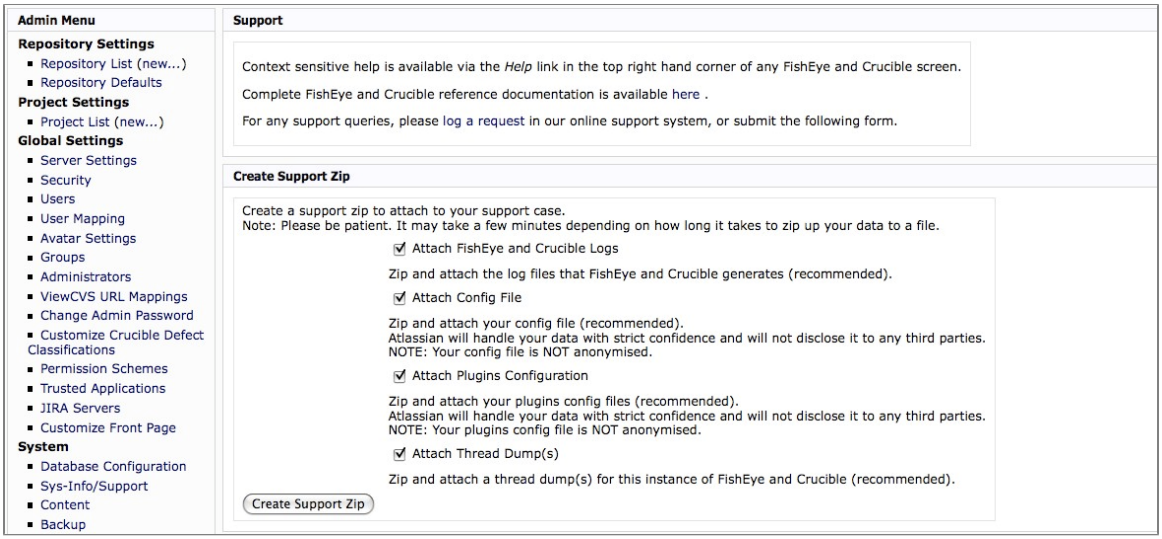
### *Support Utility*

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:




### *Raising a Support Ticket via the Fisheye Administration Console*

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.

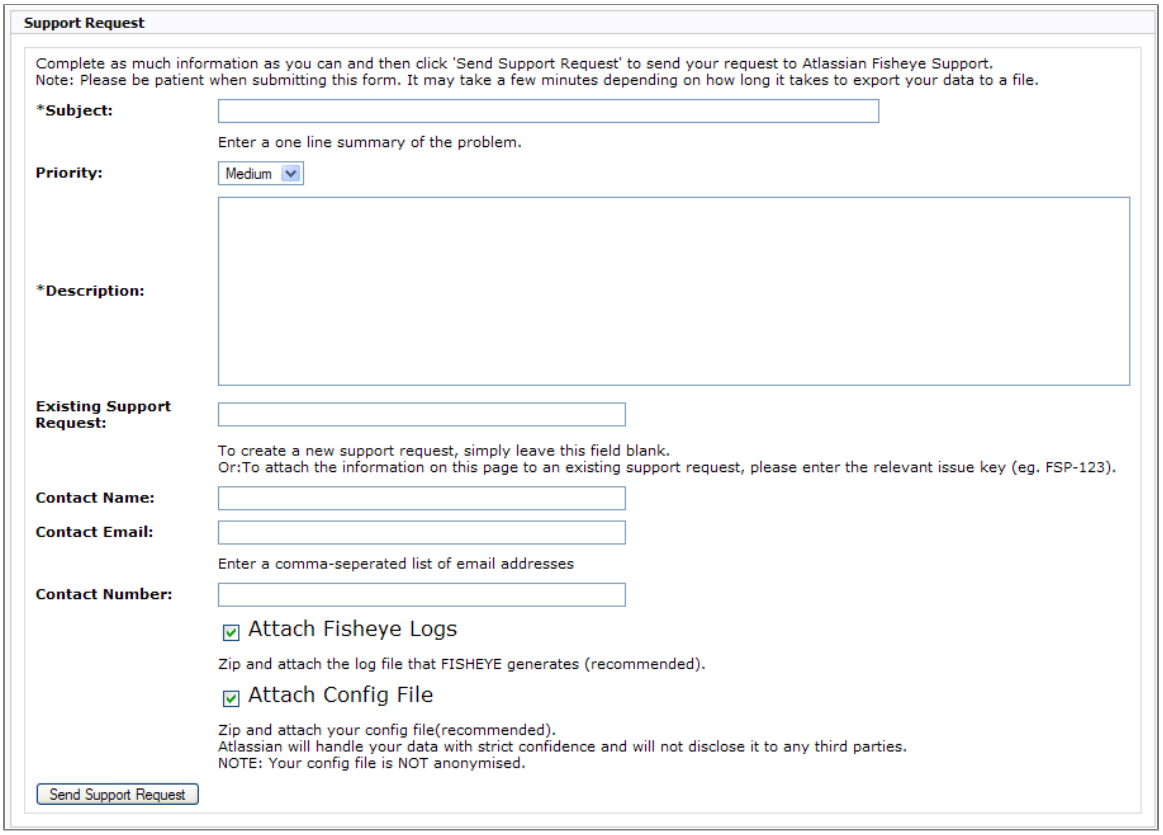
On the left navigation bar, click '**Sys-Info/Support**'.

*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).

This functionality requires that the FishEye web server is already set up and capable of sending email.

*Screenshot: The System Info/Support page*




### 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.

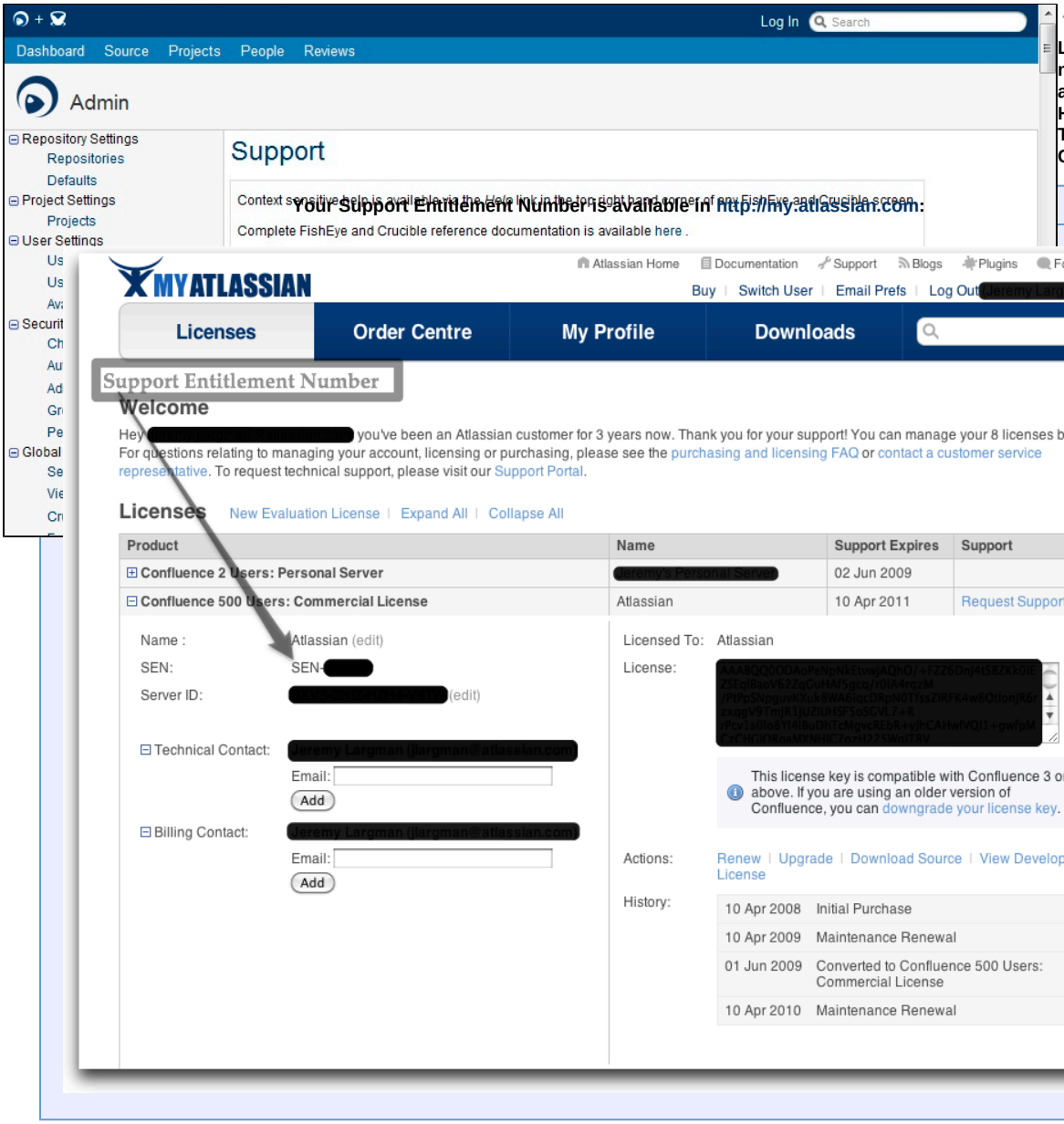
Note that the form may take several minutes to fully submit, as it takes some time to export and process the log file data.

## Finding Your FishEye or Crucible Support Entitlement Number (SEN)

There are three ways to find you Support Entitlement Number (SEN):

**Method 1: Check in the FishEye Administration Interface**

Select `Administration >> SysInfo/Support`. The SEN is shown:




**Method 2: Log into my.atlassian.com as the Account Holder or Technical Contact**

**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.

# Migrating to an External Database

This page contains instructions for migrating from your default embedded FishEye database to an external database. You may want to migrate to an external database 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

You can use a number of alternatives to the built-in HSQLDB database for storing FishEye and Crucible's relational data. The supported alternative databases are listed on the Supported Platforms page. Please note, that only the database versions listed on that page are supported.

The pages linked below outline the steps required to switch to an external database:

- Migrating to MySQL Enterprise Server
- 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.

# Migrating to MySQL Enterprise Server

To switch to a MySQL Enterprise Server, install MySQL Enterprise Server 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 MySQL Database

Before you begin:

- Ensure you install a version of MySQL that is supported. Check the Supported Platform Matrix for exact versions.

1. The JDBC drivers for MySQL Enterprise Server 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 MySQL Enterprise Server JDBC driver (a `.jar` file) from the download 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.

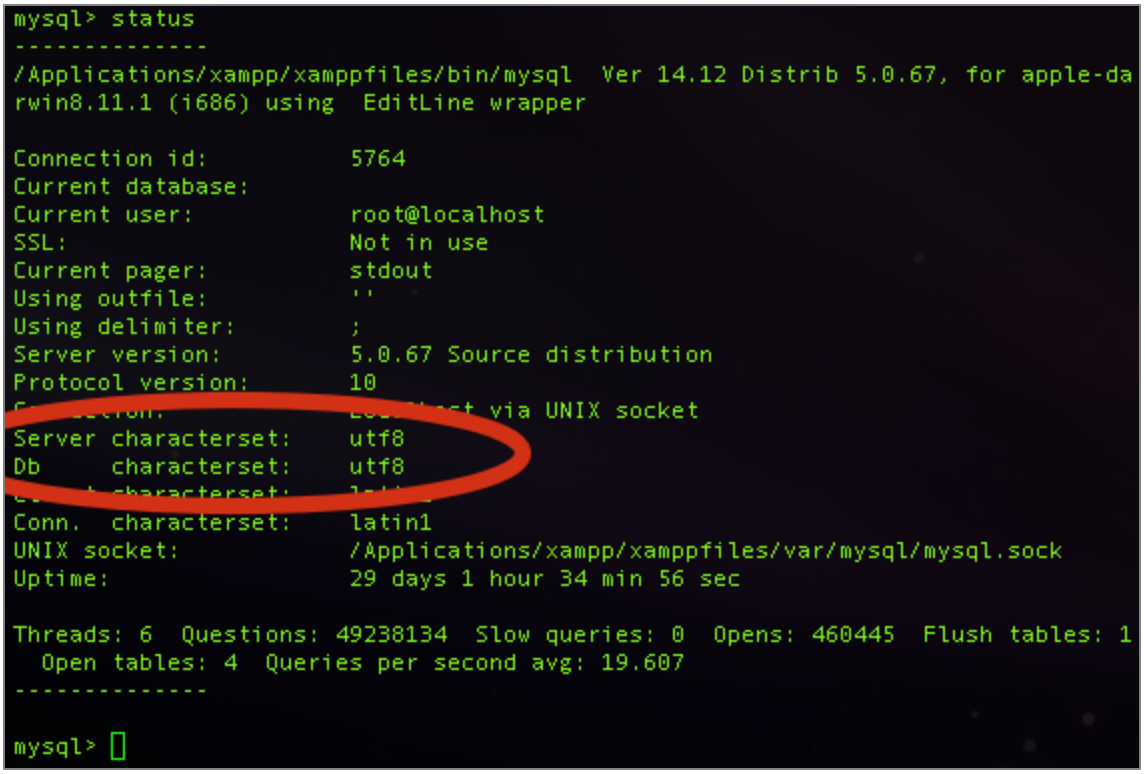
2. Create a UTF-8 Database:

```
CREATE DATABASE fisheye CHARACTER SET utf8 COLLATE utf8_bin;
```

3. 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. It has to be declared in the Server section, which is the section after `[mysqld]`:

```
[mysqld]
character-set-server=utf8
```

4. Use the **`status`** command to verify database character encoding information:
   *Screenshot: Using the MySQL Enterprise Server Status Command*




5. 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 Enterprise Server 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)
```

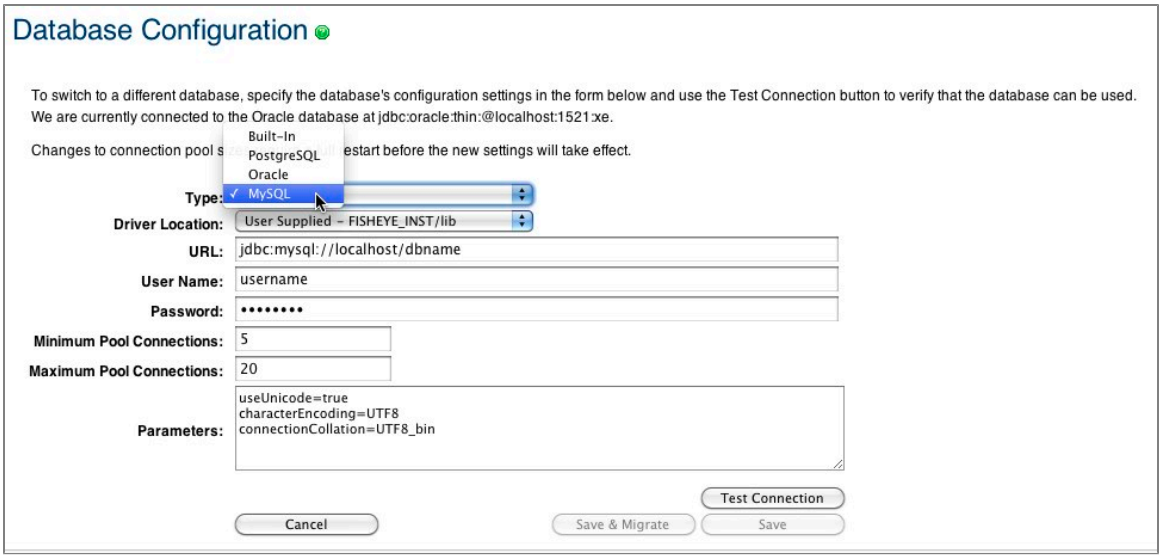
### Step 2. Configure FishEye/Crucible to use MySQL, and Migrate Data

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*

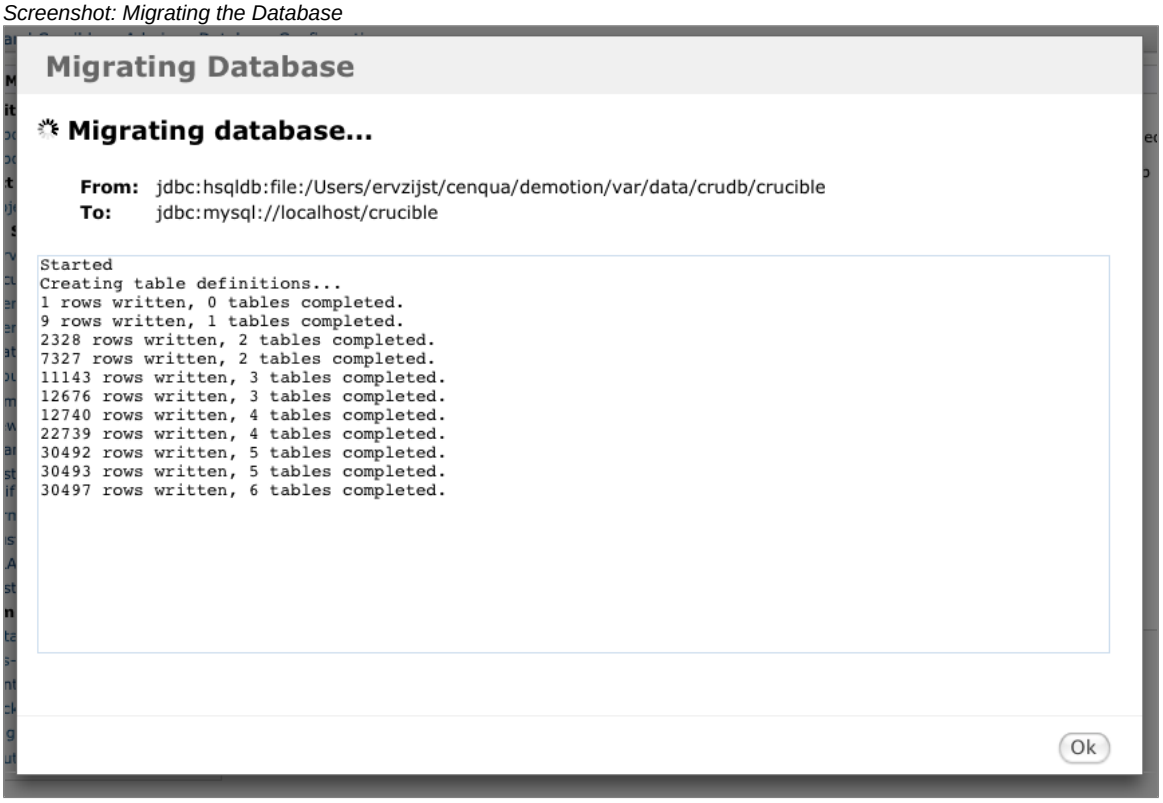



If this fails, verify that you have the MySQL Enterprise Server JDBC driver `.jar` file in the classpath (by placing the `.jar` file in `FISHEYE_INST/lib`). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges 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 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

### Notes

***Related Topics***

Troubleshooting Databases

## 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.

> 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

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 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

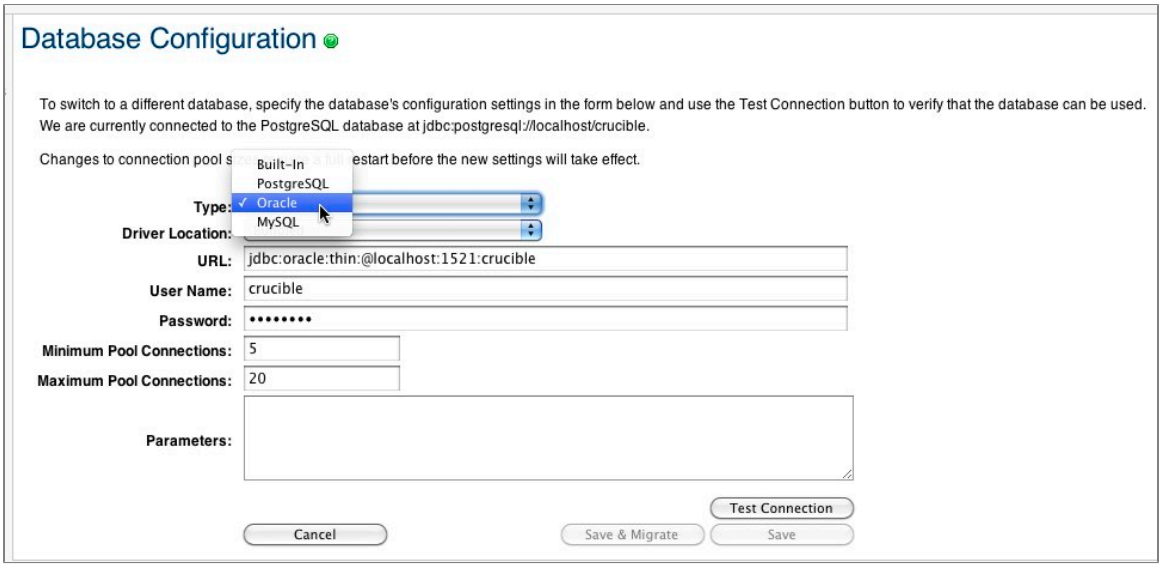
2. Then click '**Test Connection**' to verify that FishEye/Crucible can log in to the database:

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*

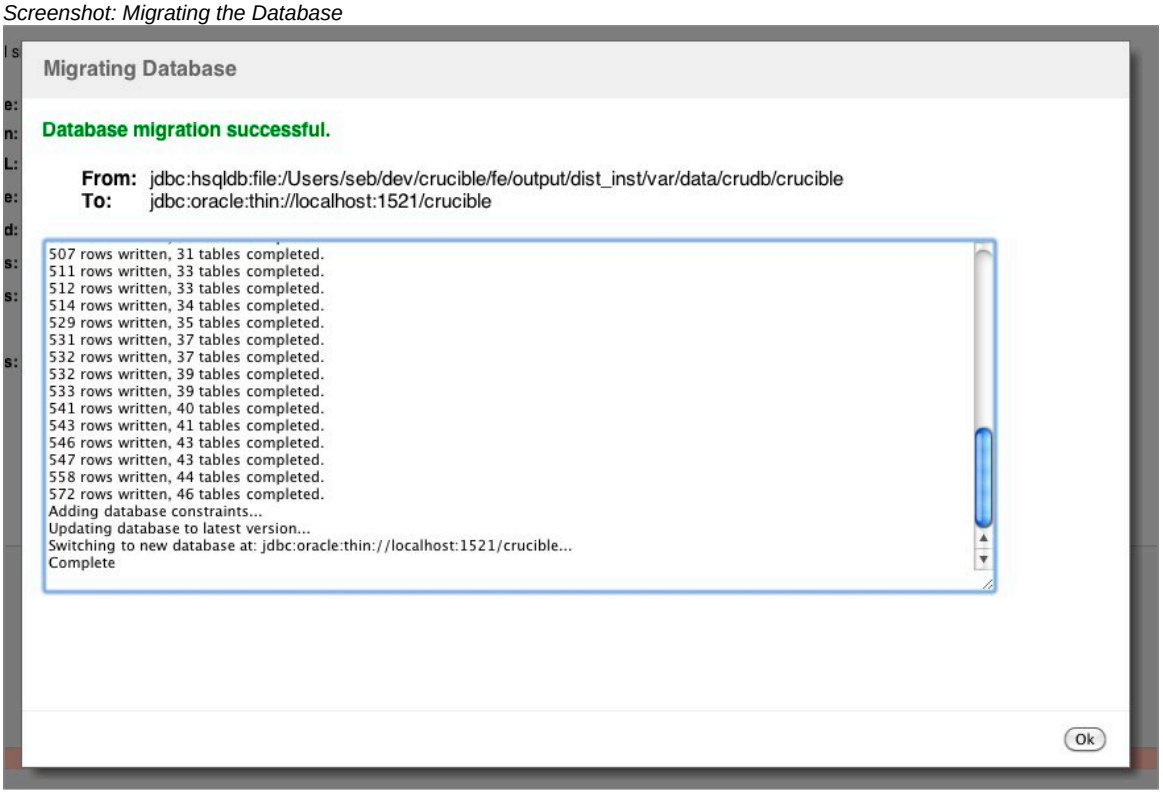



If this fails, verify that you have the Oracle JDBC driver `.jar` file in the classpath (by placing the `.jar` file in `FISHEYE_INST/lib`). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

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.

### Notes

***Related Topics***

Troubleshooting Databases

## 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.



### 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 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;
```

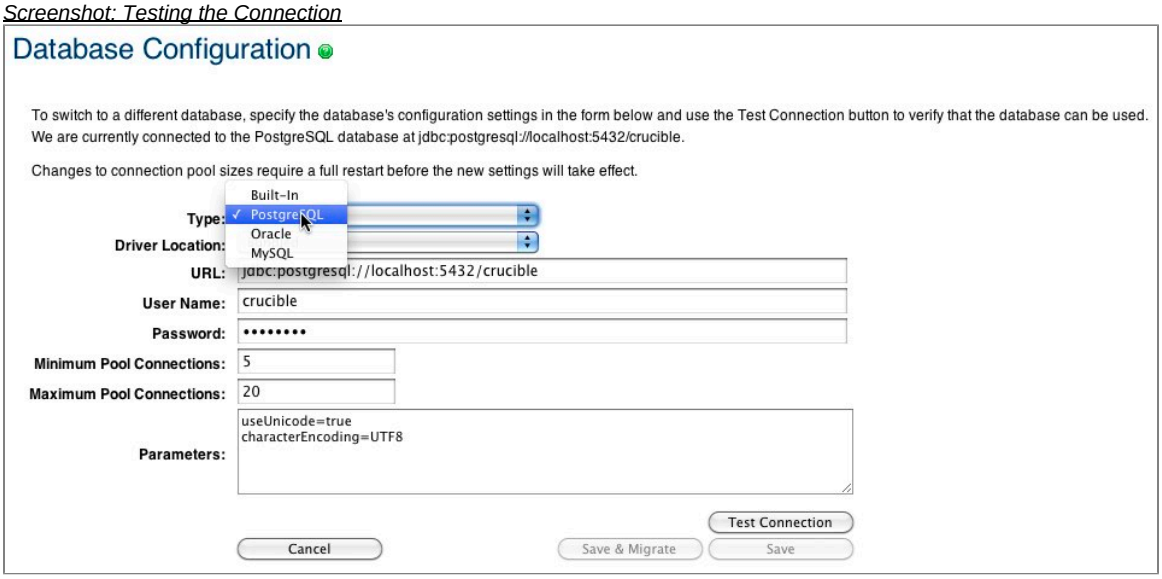
### Step 2. Configure FishEye/Crucible to use PostgreSQL, and Migrate Data

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. Then click '**Test Connection**' to verify that FishEye/Crucible can log in to the database:
3. Select `PostgreSQL` 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*

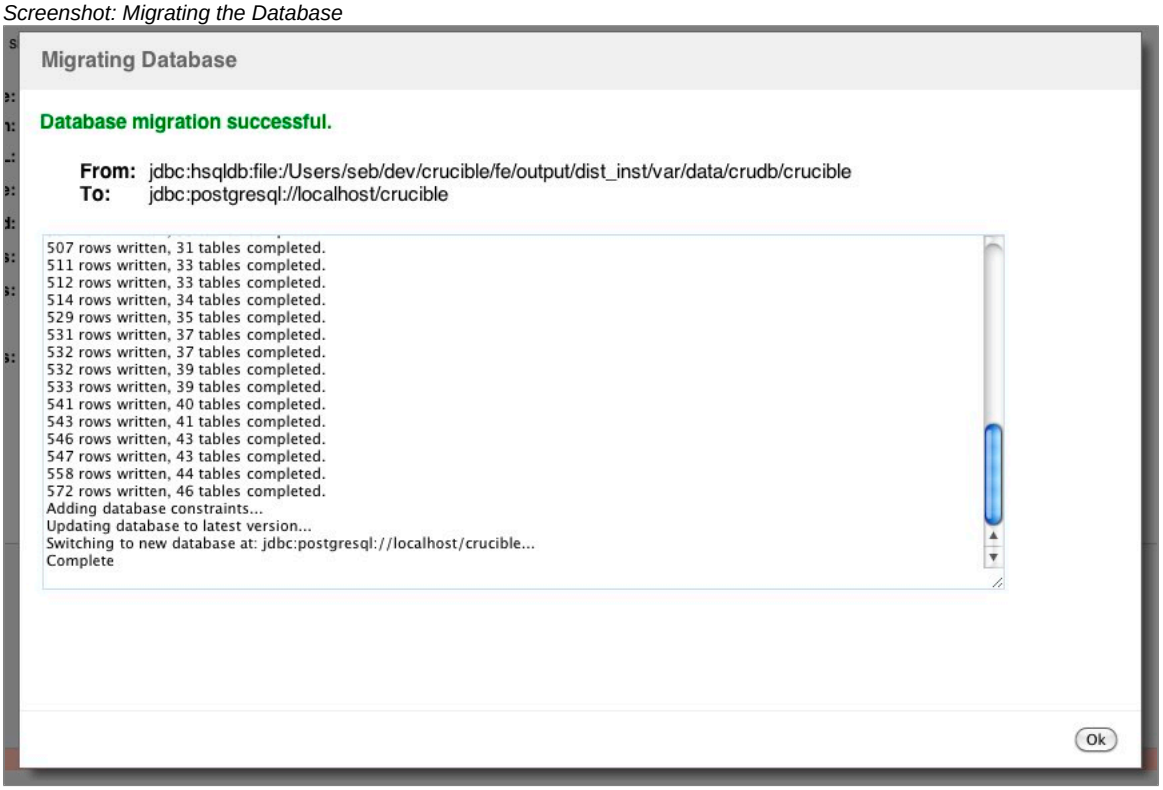
   
   

   If this fails, verify that you have the PostgreSQL JDBC driver `.jar` file in the classpath (by placing the `.jar` file in `FISHEYE_INST/lib`). Also, ensure that the database user can log in to the database from the machine that FishEye/Crucible is running on and that all the required privileges are present.

6. Click '**Save & Migrate 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 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

### Notes

#### *Related Topics*

Troubleshooting Databases

## Migrating to SQL Server

To migrate FishEye/Crucible to an SQL Server database, install SQL Server and follow the steps below.

**On this page:**

### Before You Begin

- Check that you are using version of SQL Server that is supported for use with FishEye. See Supported Platforms.

### 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.

### 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:

[ 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.
2. 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.
3. Click '**Test Connection**' to verify that FishEye/Crucible can log in to the database (see 'Testing the Connection' screenshot below).
4. Click '**Save & Migrate Data**' to start the migration process (see 'Migrating the Database' screenshot below). If the migration fails, FishEye/Crucible will not switch to the new database and will report the problems encountered.


1. Configuring the Database
2. Migrating Data

*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:**

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 sqlserver2008 \
        --jdbcurl jdbc:jtds:sqlserver://hostname:port;databaseName=dbName; \
        --username crucible \
        --password password
```

3. When the import is complete, FishEye/Crucible can be started and will use SQLServer.

[ Configuring and Migrating via FishEye/Crucible's Administration Console ] [ Configuring and Migrating via the Command Line ]

### Notes

***Related Topics***

Backing Up and Restoring FishEye Data
Backing Up and Restoring Crucible Data
Troubleshooting Databases
SQL Server Online resources (MSDN)

## Software Update Notifications

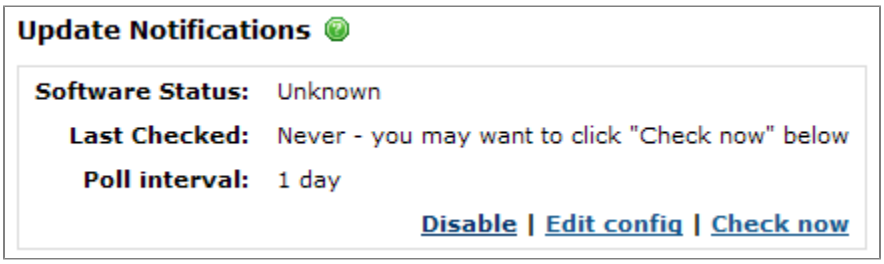
FishEye (and Crucible) can detect and notify you of new versions of the program. In the Admin screen, you can set a background thread to poll Atlassian's servers and report when a new version of FishEye or Crucible has been released.

**To use the Update Notification,**

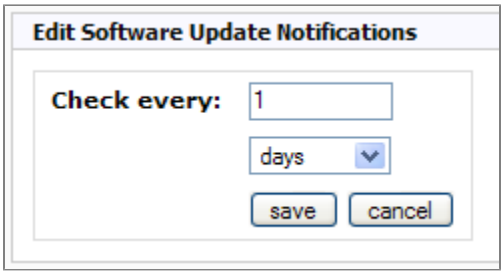
1. From the '**Admin Menu**', click '**Server Settings**' from the left navigation bar.
2. The '**Software Status**' line displays either 'Up to Date', 'New Version Available' or 'Unknown'.
3. The '**Last Checked**' line displays the date when the feature last checked for a new version.
4. Clicking the '**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*



# Running Scheduled Events

This page is deprecated. Please see Backing Up and Restoring FishEye Data.

# 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.

They templates are only for changing the appearance and order of certain content inside the messages.

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.

#### *Editing the Subject Line*

1. Open the '**fisheye-mail-subject.ftl**' template file from `FISHEYE_HOME/templates/` in a text editor.
2. Type in your new text for the email subject, ensuring that all of the content is contained within line 1 of the template. '**fisheye-mail-subject.ftl**' is used as the subject template for all FishEye email notifications.
3. Save and close the file.
4. Restarting FishEye will activate the new templates.

#### *Editing the Header*

Header information will be included at the beginning of the email body text.

1. Open the '**fisheye-mail-header.ftl**' template file from `FISHEYE_HOME/templates/` in a text editor.
2. Add your new header content. '**fisheye-mail-header.ftl**' is used as the header template for all FishEye email notifications.
3. Save and close the file.
4. Restarting FishEye will activate the new templates.

#### *Editing the Footer*

Footer information will be included at the end of the email body text.

1. Open the '**fisheye-mail-footer.ftl**' template file from `FISHEYE_HOME/templates/` in a text editor. '**fisheye-mail-footer.ftl**' is used as the footer template for all FishEye email notifications.
2. Add your new footer content.
3. Save and close the file.
4. Restarting FishEye will activate the new templates.

After an edit, the change to the email template will take place immediately. No restart is required.

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/templates/` folder.

| Template filename | Purpose |
|---|---|
| **fisheye-mail-subject.ftl** | Subject template |
| **fisheye-mail-header.ftl** | Header template |
| **fisheye-mail-footer.ftl** | Footer template |
| **changeset-mail-html.ftl** | HTML email template |
| **changeset-mail-plain.ftl** | Plain text email template |

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 *changeset* |

| | | |
|---|---|---|
| **author** | Author of the commit | Belongs to a *changeset* |
| **dateValue** | the date of the commit | Belongs to a *changeset* |
| **revisionInfos** | A list of revisions for the changeset | Belongs to a *changeset* |

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 *revisionInfo*, under a *changeset* |
| **revision** | The revision number | Belongs to a *revisionInfo*, under a *changeset* |
| **binary** | Boolean indicating whether file is binary | Belongs to a *revisionInfo*, under a *changeset* |
| **dead** | Boolean indicating whether file is deleted | Belongs to a *revisionInfo*, under a *changeset* |
| **move** | Boolean indicating whether file is moved | Belongs to a *revisionInfo*, under a *changeset* |
| **copy** | Boolean indicating whether file is copied | Belongs to a *revisionInfo*, under a *changeset* |
| **added** | Boolean indicating whether file is added | Belongs to a *revisionInfo*, under a *changeset* |
| **linesAdded** | Number of lines added | Belongs to a *revisionInfo*, under a *changeset* |
| **linesRemoved** | Number of lines removed | Belongs to a *revisionInfo*, under a *changeset* |

## Setting up your Users and Security

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.

Anonymous access to FishEye is allowed by default. You can disable anonymous access at a global level and per repository.

For an overview of FishEye security, please see Understanding security.

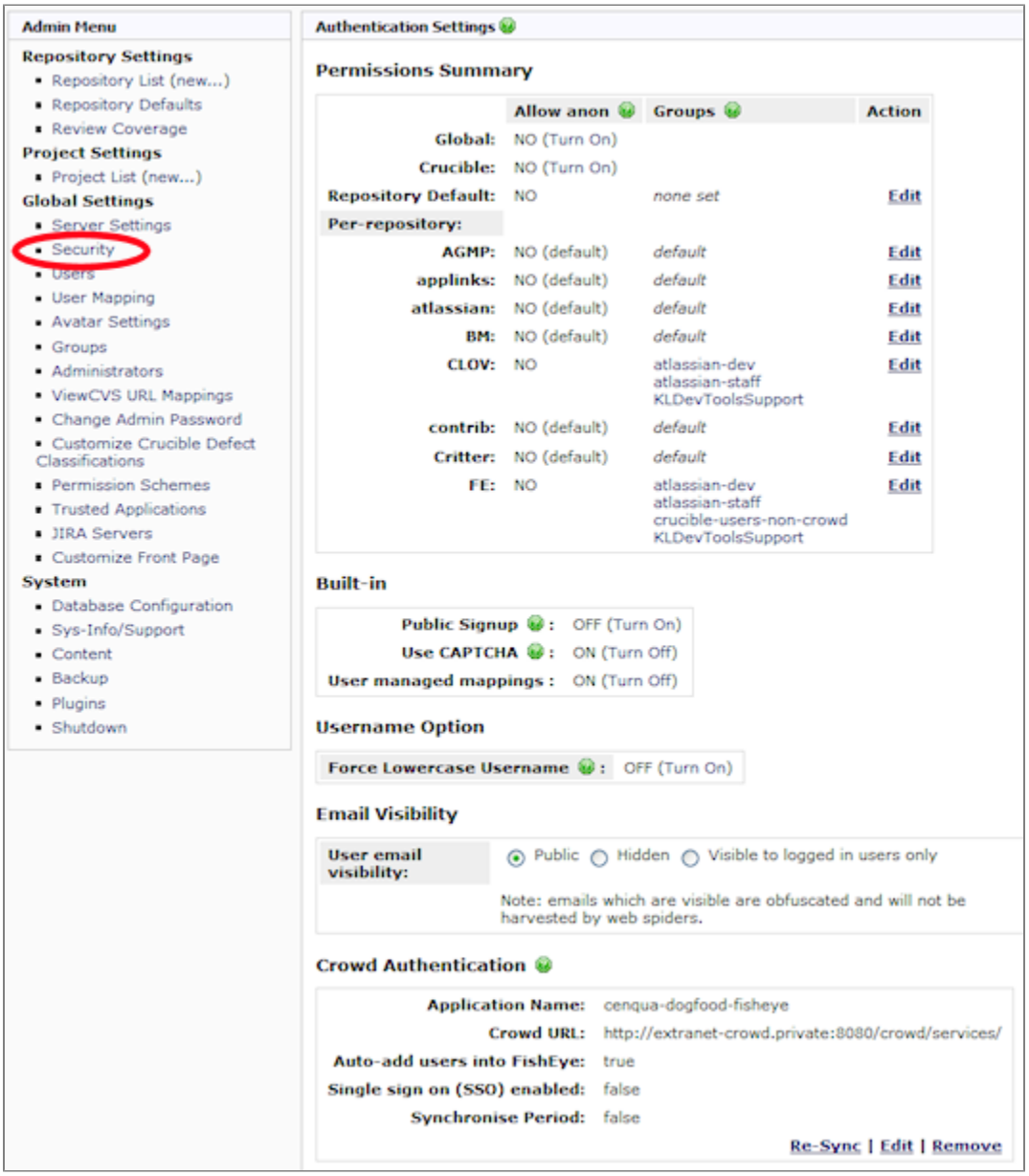
To configure your authentication settings, click '**Users/Security**' on the '**Admin Menu**'.

Select one of the options below for specific information on settings:

- Adding a User to a Group
- Associating a Group with a Repository
- Brute Force Login Protection
- Configuring Anonymous Access
- Configuring Avatar Settings
- Configuring CAPTCHA
- Configuring External Authentication Sources
- Configuring Public Signup
- Creating a Group
- Creating a User
- Deleting or deactivating a User
- Editing a User's Details
- Granting Administrator Privileges
- Loading Users from Active Directory, LDAP, Crowd or JIRA
- Understanding security

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

*Screenshot: Authentication Settings*




### *Other Related Security Resources*

No content found for label(s) security-resources.

---

## Adding a User to a Group

There are two types of FishEye user groups:

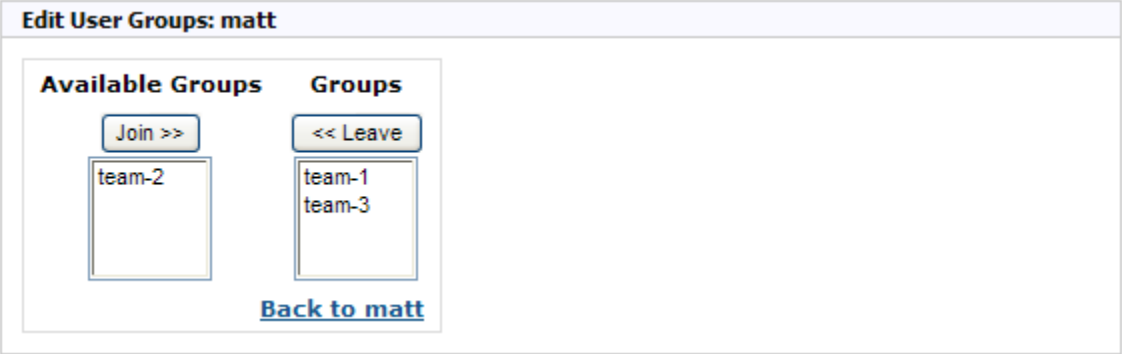
- 'Built-in' groups — these are stored in the FishEye database.
- 'External' groups — these are stored in an external directory (e.g. Crowd), if any are configured. See Configuring External Authentication Sources.

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. 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 '**Username**' link under the '**User**' column.
   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.
4. The '**Edit User**' screen will be displayed (for more information please see Editing a User's Details). Click the '**Edit Groups**' link.
5. The '**Edit User Groups**' screen will be displayed as shown in the screenshot below.
   - To add the user to a group, select the group in the '**Available Groups**' column at left and click the '**Join**' button.
   - To remove the user from a group, select the group in the '**Groups**' column at right and click the '**Leave**' button.
6. Click the '**Back to user**' button.

*Screenshot: Edit User Groups*



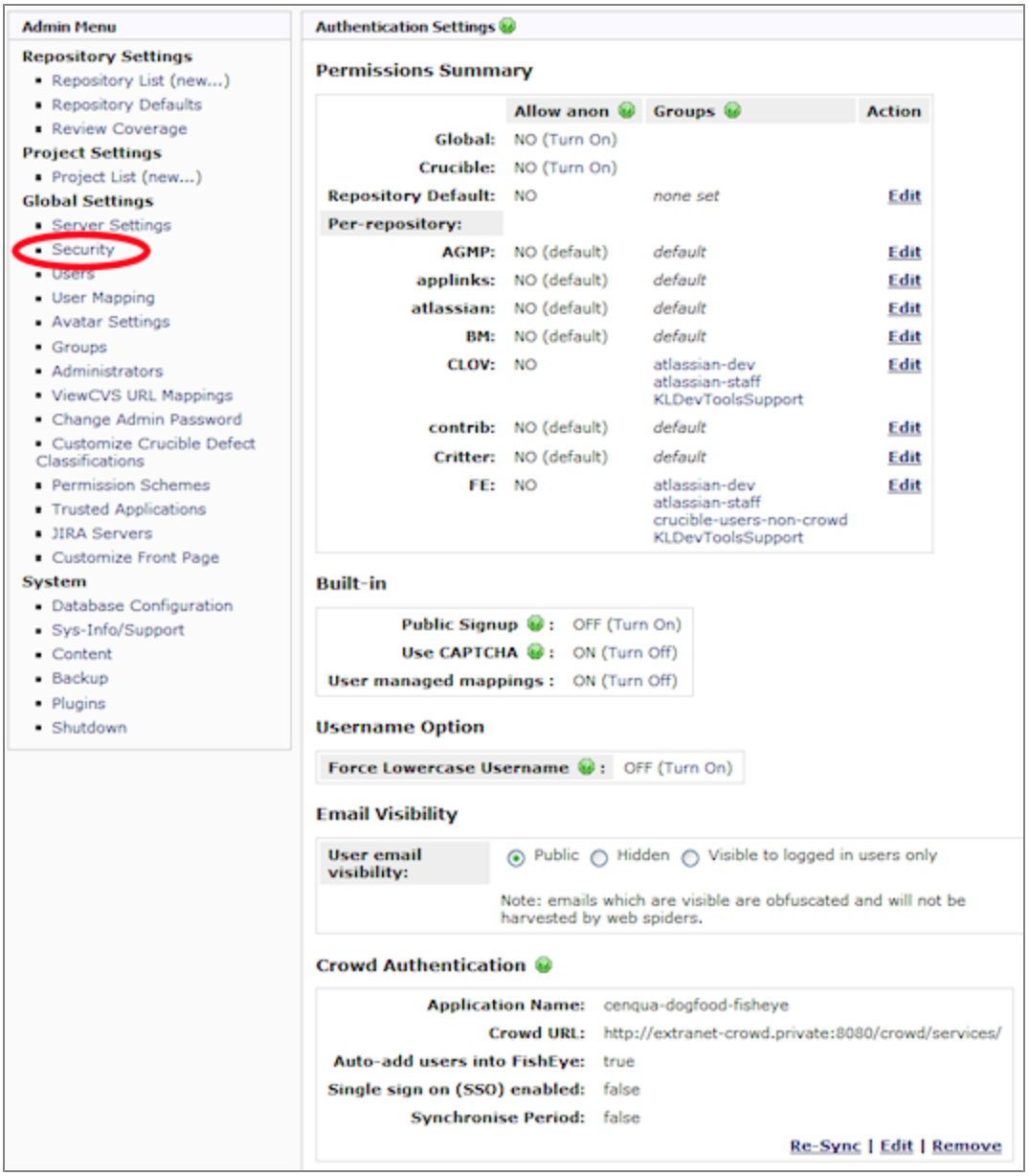

## Associating a Group with a Repository

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 security.

**To associate a group with a repository,**

1. Click '**Security**' on the '**Admin Menu**'.
2. The '**Authentication Settings**' screen will be displayed (see screenshot below), showing a list of existing groups.
3. In the '**Permissions Summary**' section, under '**Per-repository**', click the '**Edit**' link corresponding to the repository with which you wish to associate a group.
   Or, to associate a group with *all* repositories by default, click the '**Edit**' link in the '**Repository Default**' row.
4. The '**Edit Security**' screen will be displayed. Select the relevant group and click the '**Join**' button.
5. Click the '**Update**' button to return to the '**Authentication Settings**' screen.
6. Your new group will now appear in the '**Groups**' column for your chosen repository.

*Screenshot: Authentication Settings*




## 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

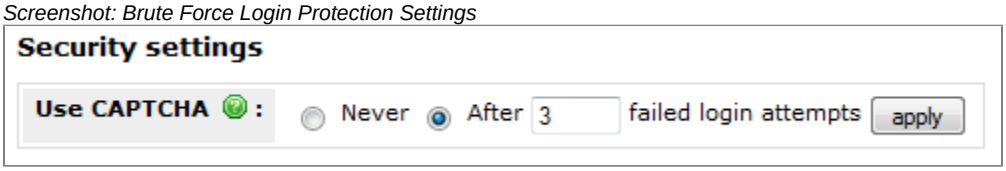
To configure brute force login protection:

1. Open the FishEye Admin screen click Security on the left-hand navigation bar. The 'Authentication Settings' screen opens.
2. Scroll down to the 'Security Settings' section at the bottom of the screen.
3. The option 'Use CAPTCHA' is displayed. You can select the following options:

- 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*




### 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 Anonymous Access

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

- Global.
- Repository default.
- Per repository.

Note: in Crucible, anonymous access is also subject to individual project permissions (see Creating a Permission Scheme).

**To configure anonymous access,**

From the '**Admin Menu**', choose one of the following options:

- '**Security**' - Allows you to change anonymous access at all levels.
- '**Repository Defaults**' - Allows you to change the default setting for repositories.
- A repository name - Allows you to change the setting for the specific repository.

## 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.

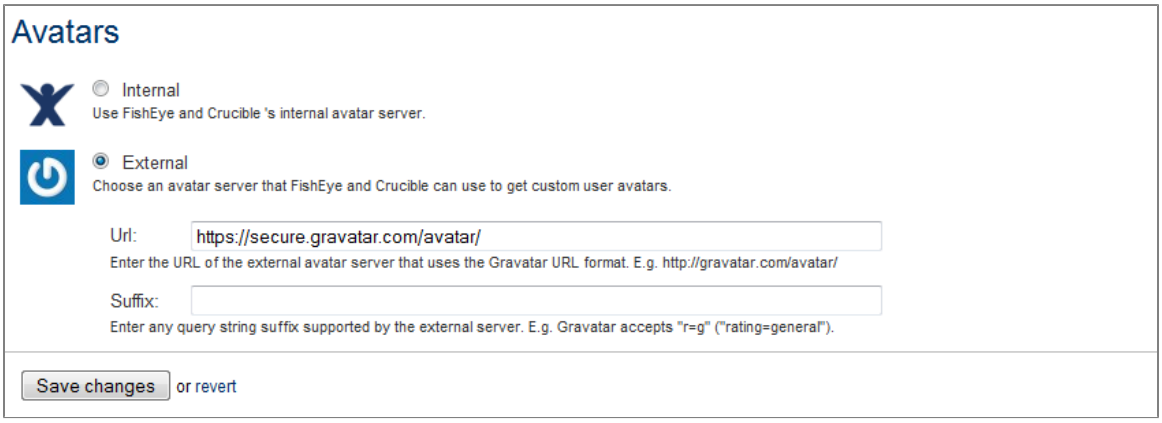
**To configure avatar settings,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

2. Click '**Avatars**' under the '**User Settings**' section in the left menu. The 'Avatars' page will be displayed (see screenshot).
3. Choose the desired avatar configuration for your FishEye/Crucible instance:
   - '**Internal**' — Choose this option if you want to use FishEye/Crucible's default avatars, "Charlitars" (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**' — Choose this option if you want to 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. Specify your avatar server, as follows:
     - '**Url**' — Type the address of the external server, e.g. `http://gravatar.com/avatar/`. Use the same protocol that is used to access FishEye, e.g. 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: `https://secure.gravatar.com/avatar/`
     - '**Suffix**' — *(optional)* 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"`
4. Click '**Save changes**' to save the avatar configuration.

*Screenshot: 'Configure Avatar Settings'*




### Notes

- 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.

## 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.

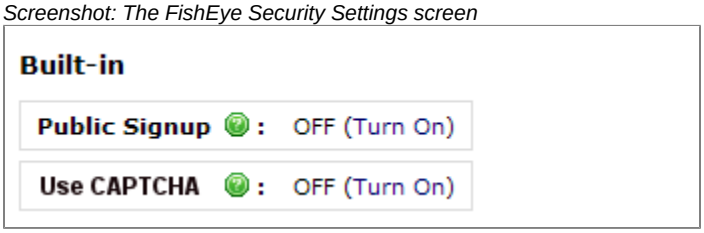
Administrators can enable or disable CAPTCHA testing on the public signup and password retrieval screens. CAPTCHA testing is enabled by default.

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.

## 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

## 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

Here is one example of how 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
```

## 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*


## LDAP Authentication

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

*On this page:*



#### Global Settings

Global LDAP settings are:

| | |
|---|---|
| URL | The URL of the LDAP server, e.g. `ldap://localhost:389`. |
| Base DN | The base search space for users, e.g. `dc=example,dc=com` |
| User Filter | The LDAP search for locating users, e.g. `uid=${USERNAME}`. The `${USERNAME}` 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: `(&(uid=${USERNAME})(group=fisheye))`. |
| 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. You can specify this per repository, or just specify it in the Repository Defaults to have it apply to all repositories  where not specified for the individual repository:

| | |
|---|---|
| LDAP restriction | An LDAP filter used to check if a given user can access a given repository, e.g. `(&(uid=${USERNAME})(group=${REP}))`. The `${REP}` variable is replaced with the name of the repository in question. |
| Match Type | One of 'user' (default) or 'any'. This setting modifies the meaning of LDAP restriction.<br>If set to 'user', then FishEye expects the filter to match the exact DN of the current user. If it does match, then the user has access to the repository. Commonly, if your user object contains the list of groups the user has access to, then you would use a 'user' match.<br>If set to 'any', then the filter just needs to match one result for the user to have access to the repository. Commonly, if your group object contains the list of UID members, then you would use an 'any' match. In such a case, your LDAP restriction filter may look like this: `(&(uniquemember=${USERNAME})(cn=${REP},ou=groups,ou=com)(objectClass=groupofuniquenames))`. That is, return the group of which the current user is a member. |

#### Active Directory

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

| | |
|---|---|
| URL | `ldap://HOSTNAME:389` |

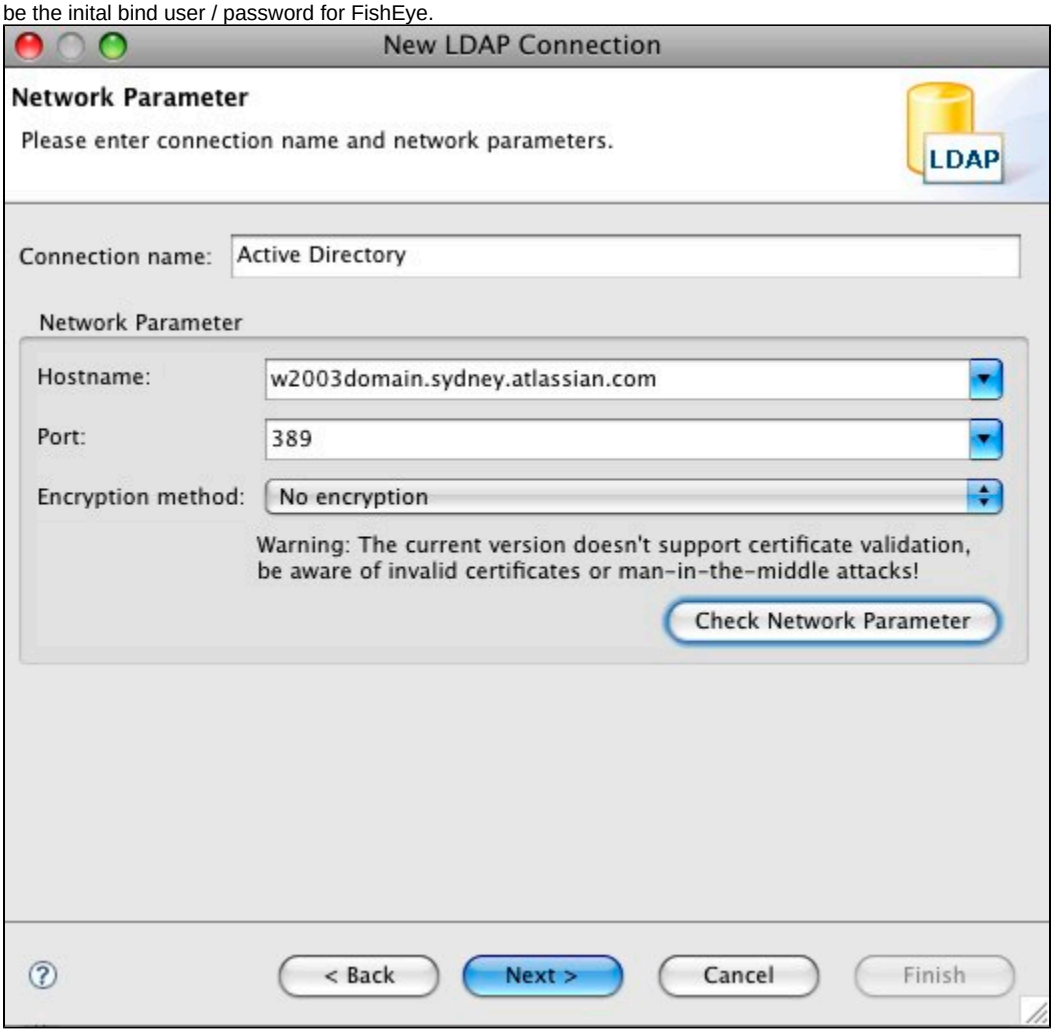
| | |
|---|---|
| Base DN | `DC=corp,DC=example,DC=com` |
| User Filter | `(sAMAccountName=${USERNAME})` |
| UID Attribute | `sAMAccountName` |
| Email attribute | `mail` |
| Initial bind DN | `corp.example.com/Users/SomeUser` |

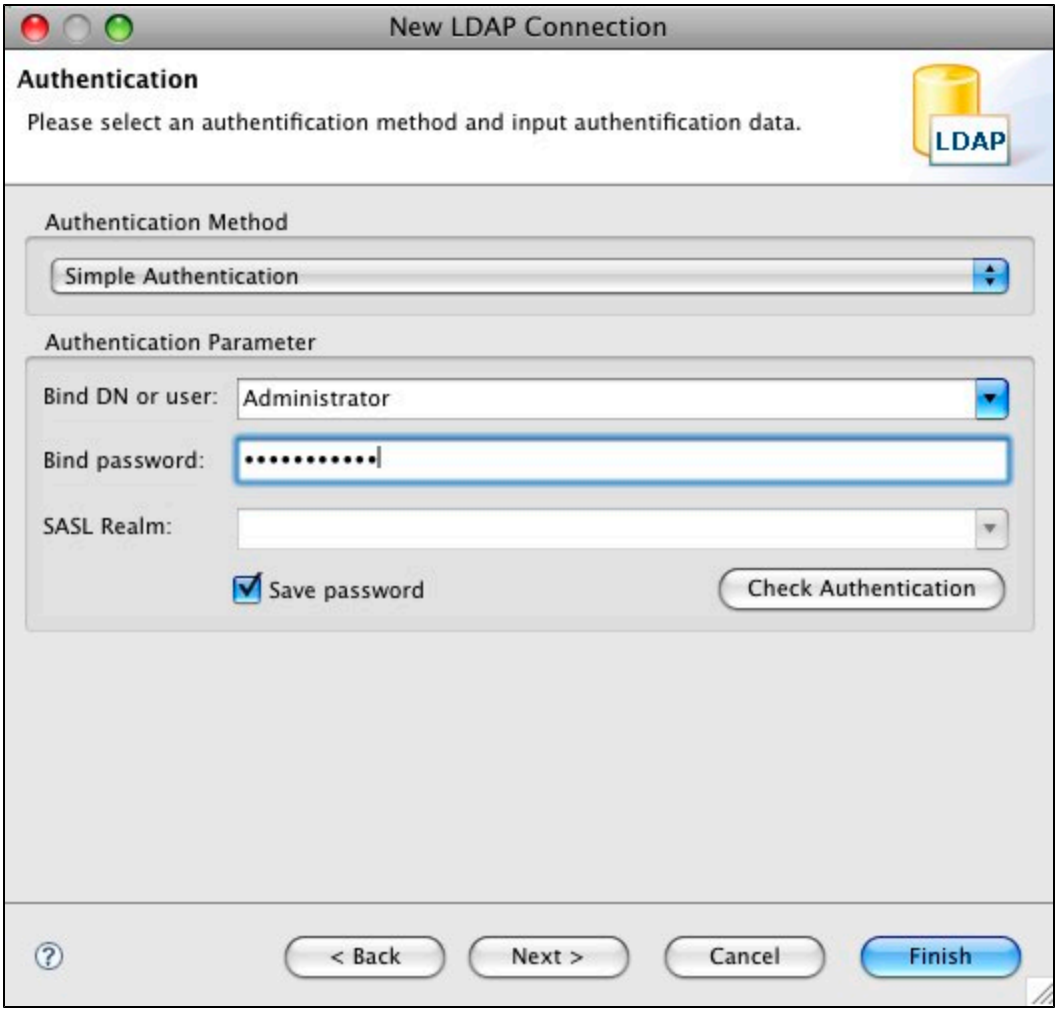
### How to connect FishEye to Active Directory or another LDAP server

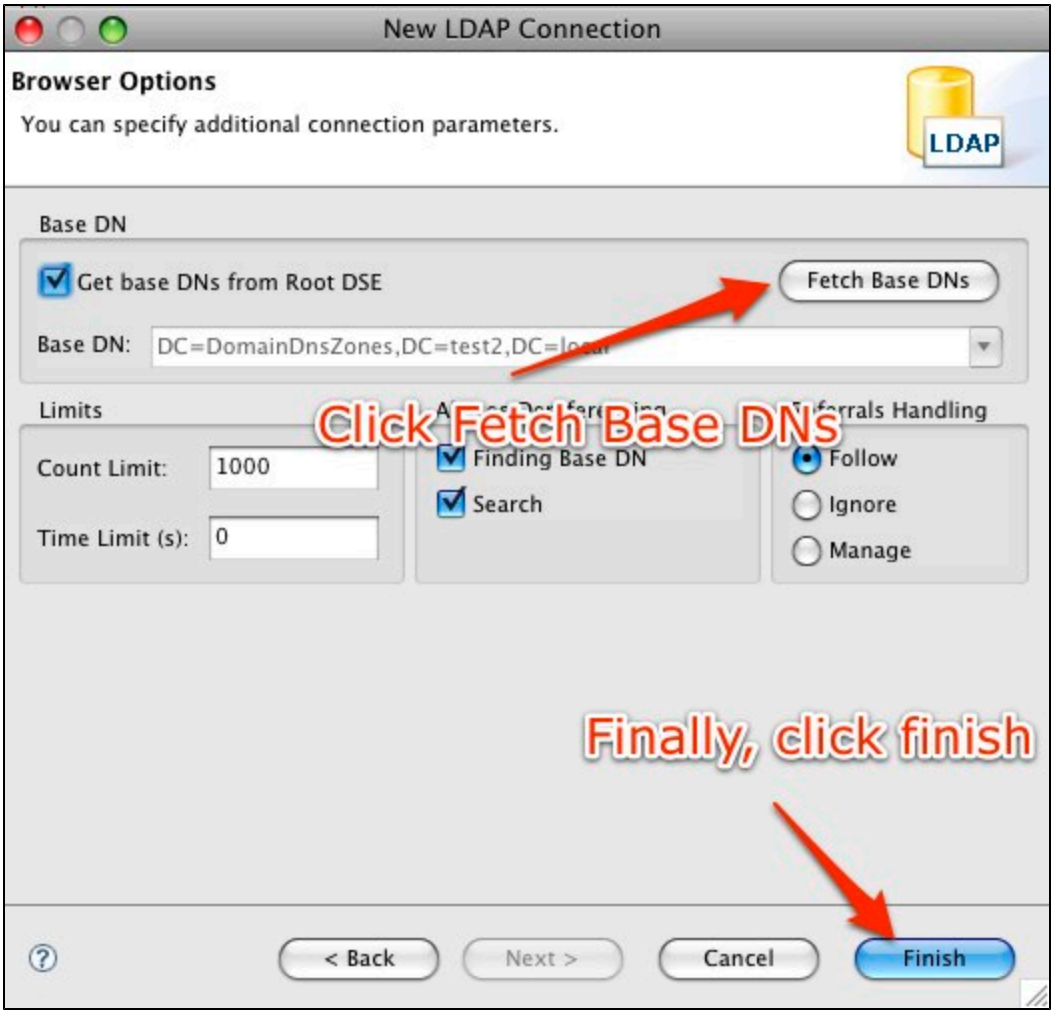
These instructions target Active Directory, but you should be able to use the same process for other LDAP servers.

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.

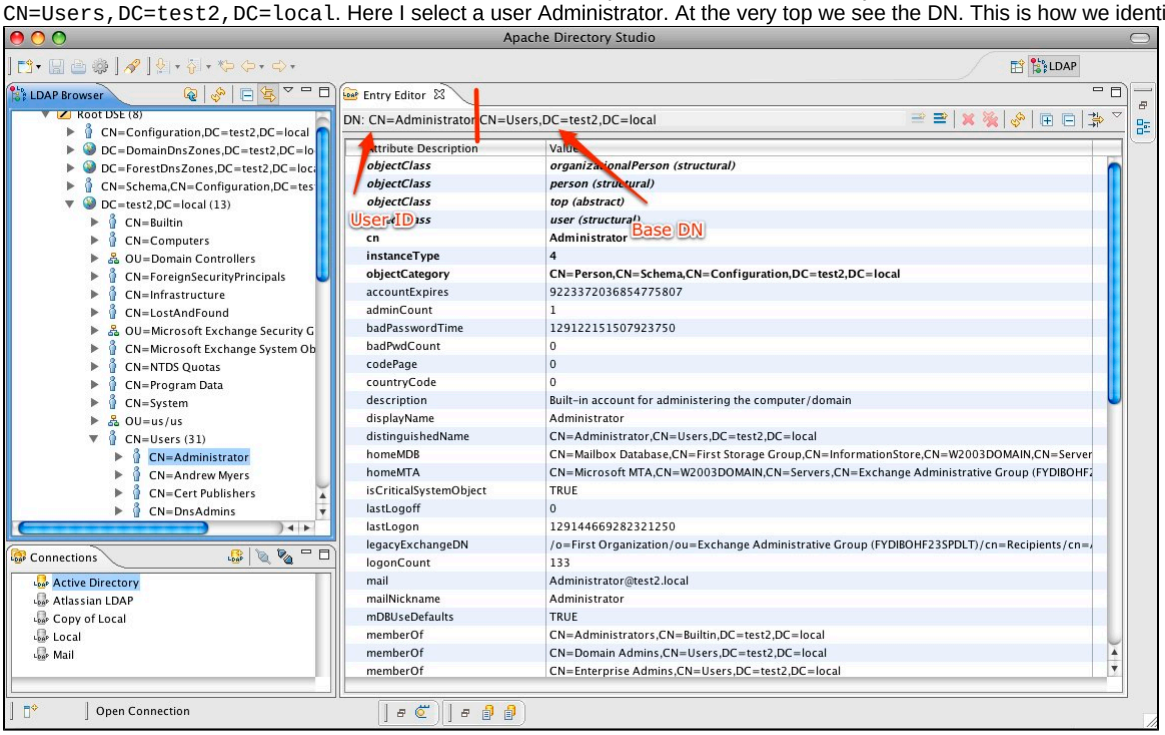
1. Download and install Apache Directory Studio.
2. Once installed Apache Directory Studio go into File -> New -> LDAP Connection and follow the wizard. The details you enter in step 2 will be the inital bind user / password for FishEye.




New LDAP Connection
Authentication
Please select an authentification method and input authentification data.
LDAP
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.



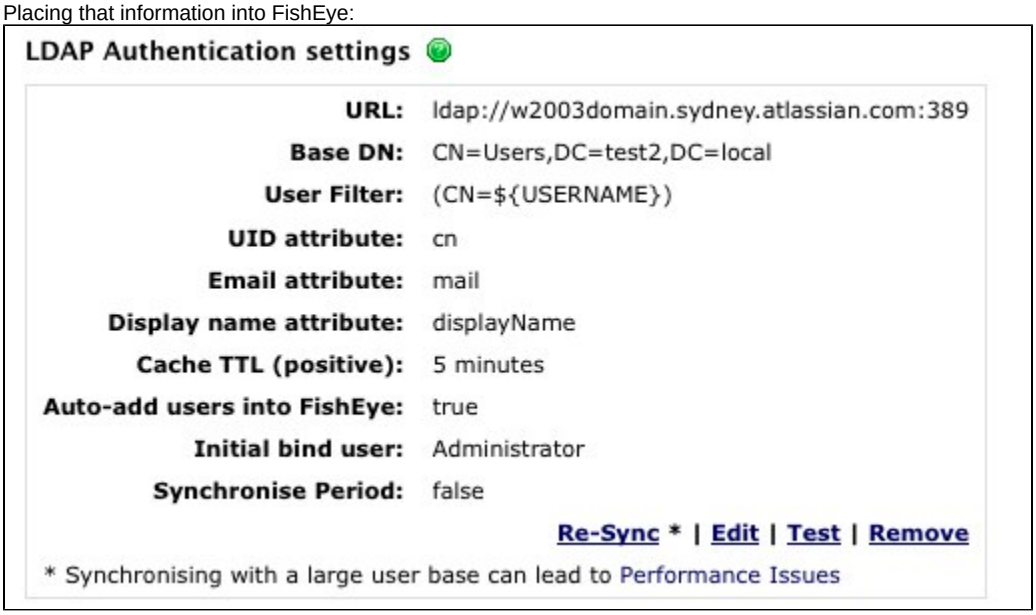

In this instance the DN is `CN=Administrator,CN=Users,DC=test2,DC=local`, so the Base DN for FishEye would be `CN=Users,DC=test2,DC=local`. `CN=Administator` identifies the user, so the user filter would be `(CN=${USERNAME})`.

> 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:




***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*

If you are using JIRA 4.3 or later, you can manage your FishEye/Crucible users in JIRA. See JIRA Integration in FishEye.

### *Integrating FishEye with Crowd*

Atlassian's Crowd user management application can be integrated with FishEye. Please see the document Integrating Crowd with FishEye in the *Crowd Administrator's Guide*.

### *Notes*

- FishEye is bundled with the Crowd 2.2.5 client library, and therefore is intended to operate with Crowd 2.2 or later.

## 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 the '**Setup Custom authentication**' link on the FishEye '**Admin**' -> '**Security**' page.

You will be presented with a form containing the following fields to be set:

| | |
|---|---|
| Classname | The fully qualified class name of your AbstractFishEyeAuthenticator, e.g. `com.cenqua.fisheye.user.plugin.ExampleFishEyeAuthenticator`. |
| 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 `init()` method. This field should comply with the java.util.Properties format. Example: <br> `# comments`<br>`name1=value1`<br>`name2=value2` |

#### 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.<br>**PAM:** The service name in your PAM configuration to use. If blank, `fisheye` 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. |

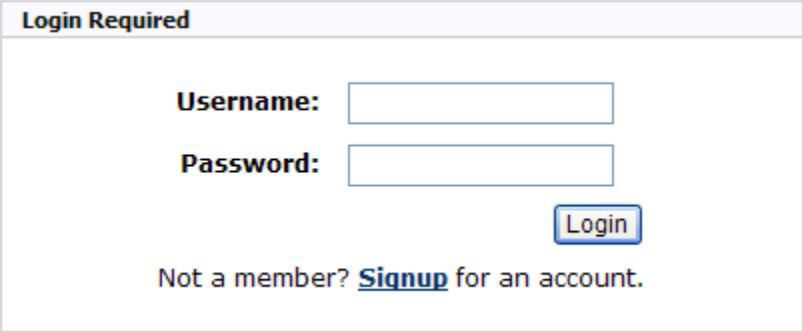
**Per-Repository Settings**

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: `cvsusers & cvs${REP}` The `${REP}` variable is replaced with the name of the repository in question. |

## Configuring Public Signup

If you enable *public signup* for your FishEye system, visitors can create their own FishEye user accounts via the '**Signup**' link on the login screen:

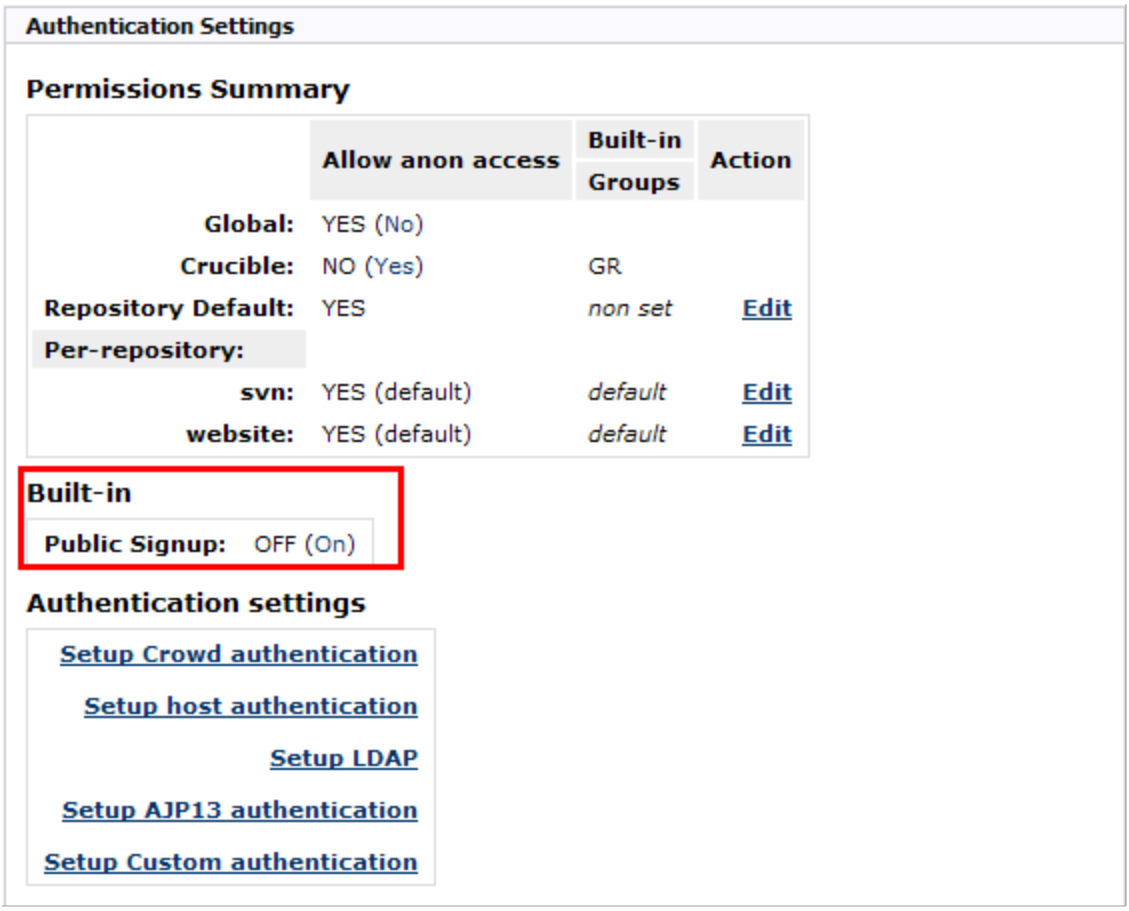



Public signup is disabled by default.

**To enable public signup,**

1. Go to the FishEye '**Admin Menu**'.
2. Click the '**Security**' link in the left navigation column.
3. The '**Authentication Settings**' page will be displayed (see screenshot below).
4. Next to '**Public Signup**', click the '**On**' link.
5. Log out of FishEye and verify that the login screen now contains a '**Signup**' link.

*Screenshot: 'Authentication Settings — Public Signup'*




## 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.

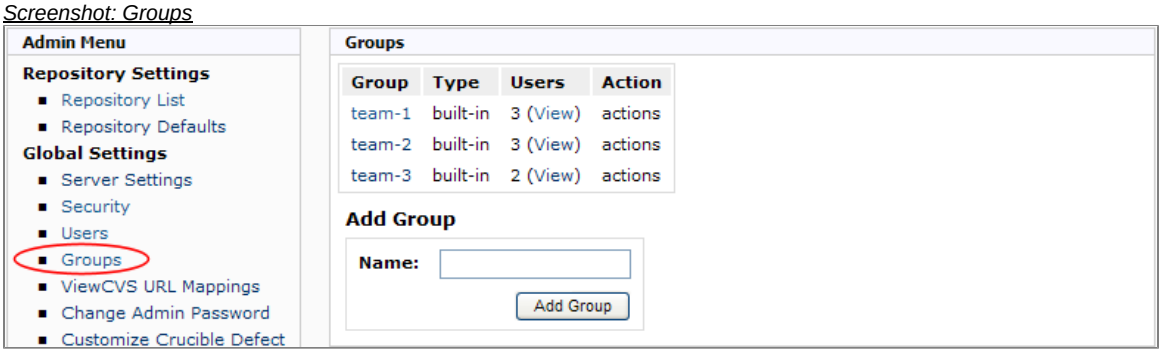
  Note that 'external' groups can only be created in your external directory.

**To add a 'built-in' group,**

1. Click '**Groups**' on the '**Admin Menu**'.
2. The '**Groups**' screen will be displayed (see screenshot below), showing a list of existing groups.
3. Type the name of your new group into the '**Name**' field and click the '**Add Group**' button at the bottom of the screen.
4. 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.

*Screenshot: Groups*



## Creating a User

There are two types of user accounts:

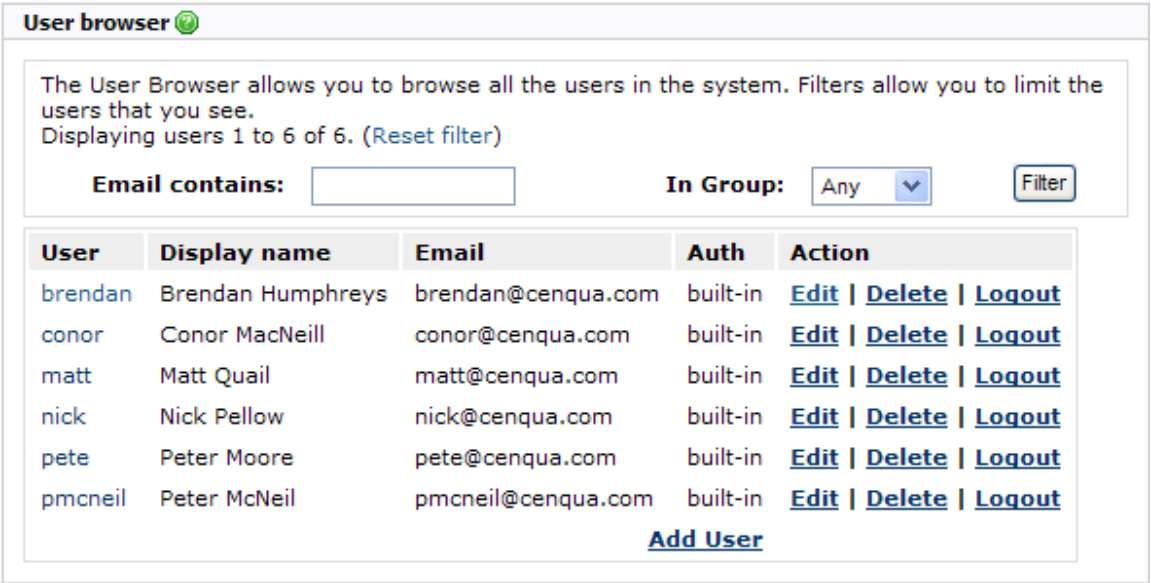
- 'Built-in' user accounts — these are stored in the application's local database.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if any are configured. See Configuring External Authentication Sources.

> **Note re 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 will be automatically added the first time they log in.

**To add a new user,**

1. Click '**Users**' on the '**Admin Menu**'.
2. The '**User Browser**' screen will be displayed (see screenshot below). Click the '**Add User**' button at the bottom of the screen.
3. The '**Add new user**' screen will be displayed.
4. In the '**Username**' field, type the user's login name. You can use the following characters:
   - letters and numbers
   - hyphen ('-')
   - underscore ('_')
   - 'at' sign ('@')
5. In the '**Display name**' field, type the user's display-name.
6. *(Optional)* In the '**Email**' field, type the user's email address. This address is where the user will receive notifications.
7. In the '**Auth Type**' field, select either '**Built-in**' or the name of the appropriate external directory where the user will be stored.
8. *(For built-in users only)* In the '**Password**' and '**Confirm Password**' fields, type the user's password.
   The user can easily change their own password later.
9. Click the '**Add**' button.

*Screenshot: User Browser*




## 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.

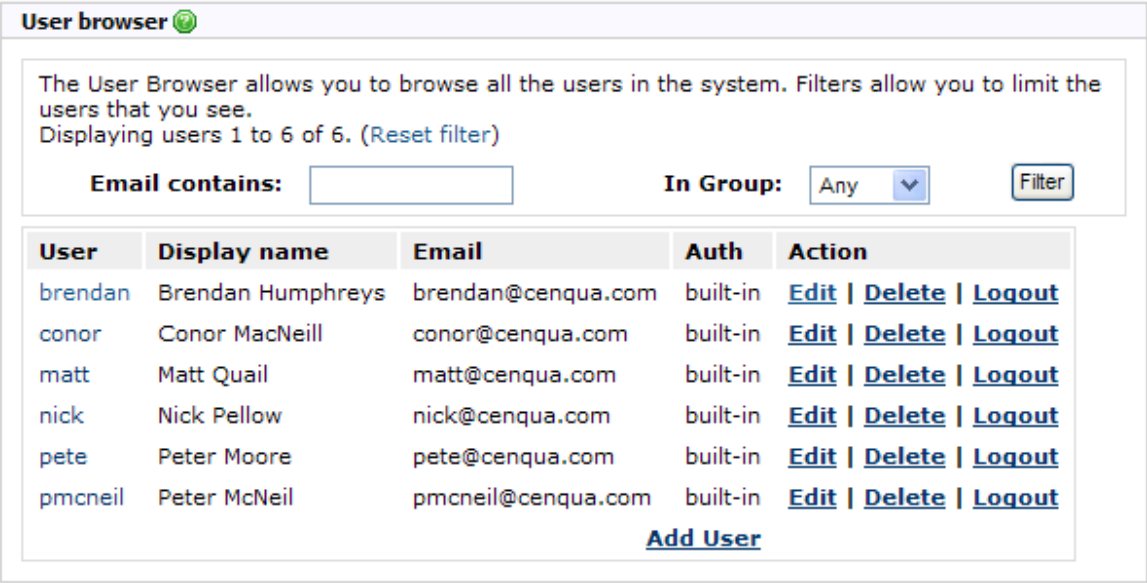
**To deactivate a user's Crucible account,**

1. Click '**Users**' on the '**Admin Menu**'.
2. The '**User Browser**' screen will be displayed (see screenshot below), showing a list of users.
3. 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.
4. The '**Edit User**' screen will be displayed. Deselect the '**Active Crucible user**' check-box.
5. Click the '**Apply**' button.

**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 the '**Filter**' button.

*Screenshot: Authentication Settings*




## Editing a User's Details

There are two types of user accounts:

- 'Built-in' user accounts — these are stored in the application's local database.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if any are configured. See Configuring External Authentication Sources.

**To edit a user's details,**

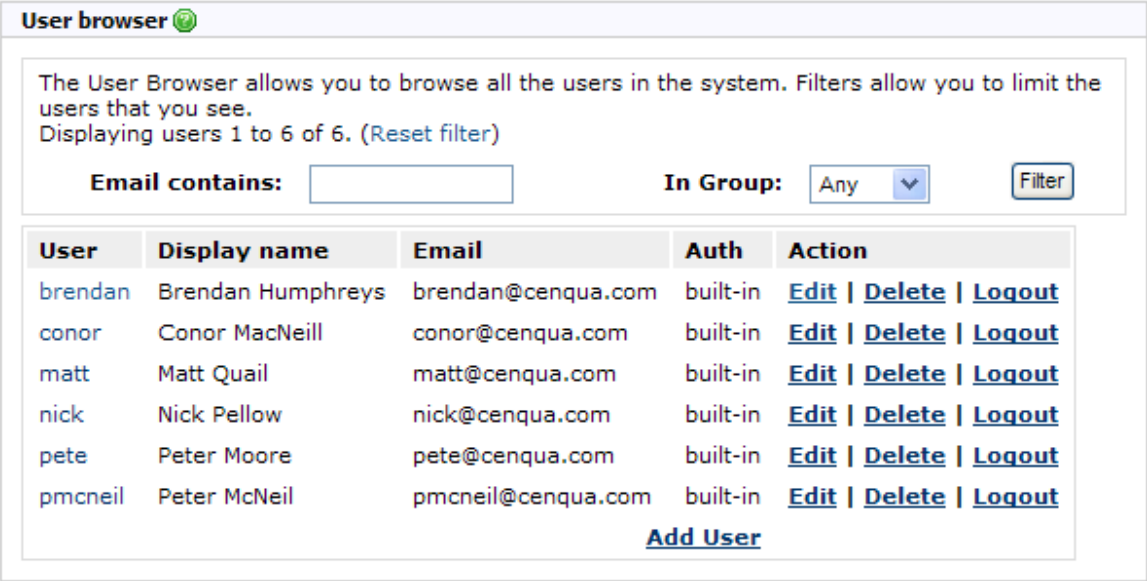
1. Click '**Users**' on the '**Admin Menu**'.
2. The '**User Browser**' screen will be displayed (see screenshot below), showing a list of users.
3. Locate the user and click the corresponding '**Edit**' link.

   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.
4. The '**Edit User**' screen will be displayed. You can update the following fields:
   - '**Display Name**' — type the user's display-name.

     To change the user's login name, see Renaming a User.
   - '**Email**' — type the user's email address. This address is where the user will receive notifications.
   - '**Auth Type**' — select either '**Built-in**' or the name of the appropriate external directory where the user will be stored.

     By changing the '**Auth Type**', you are moving the user to a different directory.
5. Click the '**Apply**' button.

*Screenshot: User Browser*




## Changing a User's Password

There are two types of user accounts:

- 'Built-in' user accounts — these are stored in the application's local database.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if any are configured. See Configuring External Authentication Sources.

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

**To change a user's password,**

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 '**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.
4. The '**Edit User**' screen will be displayed. Click the '**Change Password**' link.
5. The '**Change Password**' screen will be displayed. Type the new password and click the '**Apply**' button.

### Renaming a User

There are two types of user accounts:

- 'Built-in' user accounts — these are stored in the application's local database.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if any are 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. 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 '**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.
4. The '**Edit User**' screen will be displayed. Click the '**Rename**' link.
5. The '**Rename user**' screen will be displayed. Type the new username and click the '**Rename**' button.

## Granting Administrator Privileges

FishEye can now grant Admin privileges to users and groups. These 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.

Once logged in as admin, you have the following option in the Admin screen:

*Screenshot: The Administrators Menu Option*


### *Setting Admin privileges for individual users*

To set user's Admin rights, log in as the Admin user and select '**Global Settings**' > '**Administrators**' from the left navigation bar.

To grant a user admin rights:

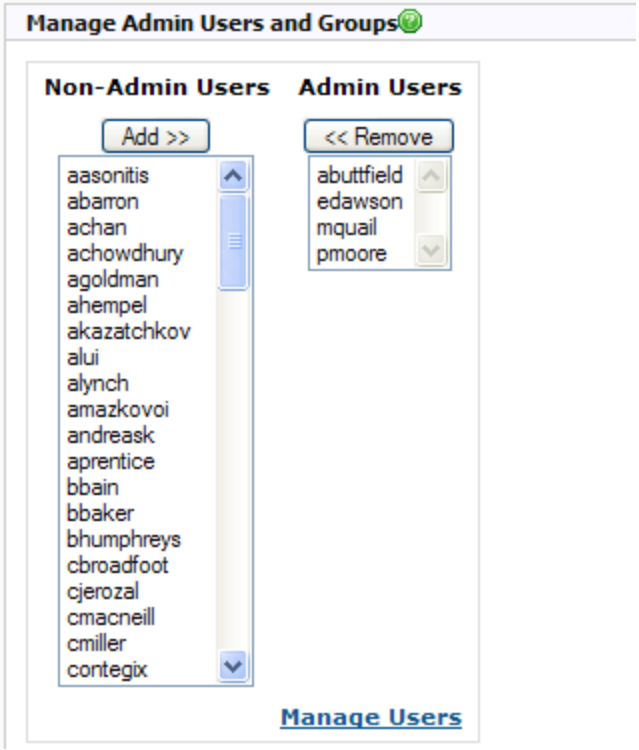
- Select the username in the '**Non-Admin Users**' column.
- Click the '**Add**' button.
- The username appears in the '**Admin Users**' column.

To remove a user's admin rights:

- Select the username in the '**Admin Users**' column.
- Click the '**Remove**' button.
- The username appears in the '**Non-Admin Users**' column.

Take care not to remove admin rights from your own account.

*Screenshot: Setting User's Admin Settings*




### *Setting Admin privileges for groups*

To set a group's Admin rights, log in as the Admin user and select '**Global Settings**' > '**Administrators**' from the left navigation bar.

To grant a group admin rights:

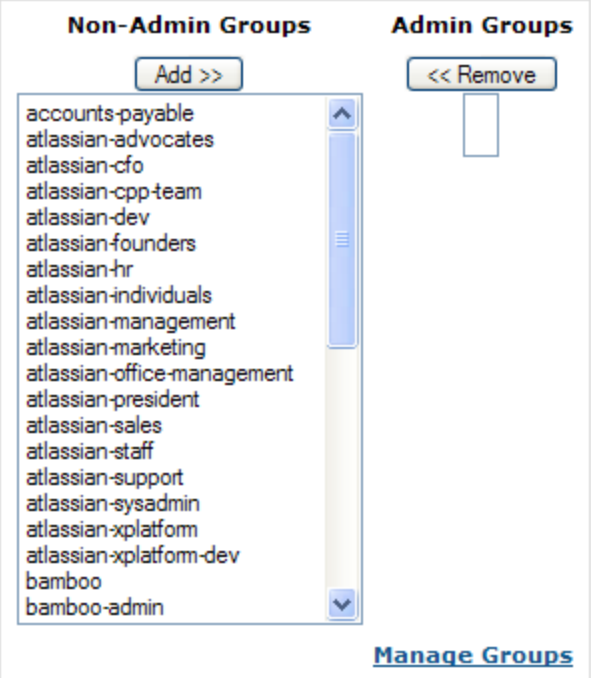
- Select the group name in the '**Non-Admin Groups**' column.
- Click the '**Add**' button.
- The group name appears in the '**Admin Groups**' column.

To remove a group's admin rights:

- Select the group name in the '**Admin Groups**' column.
- Click the '**Remove**' button.
- The groups name appears in the '**Non-Admin Groups**' column.

Take care not to remove admin rights from your own account.

*Screenshot: Setting Group Admin Settings*




## 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 Repositories.

## Synchronising With Large Repositories

When FishEye synchronises with external 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 you 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.
```

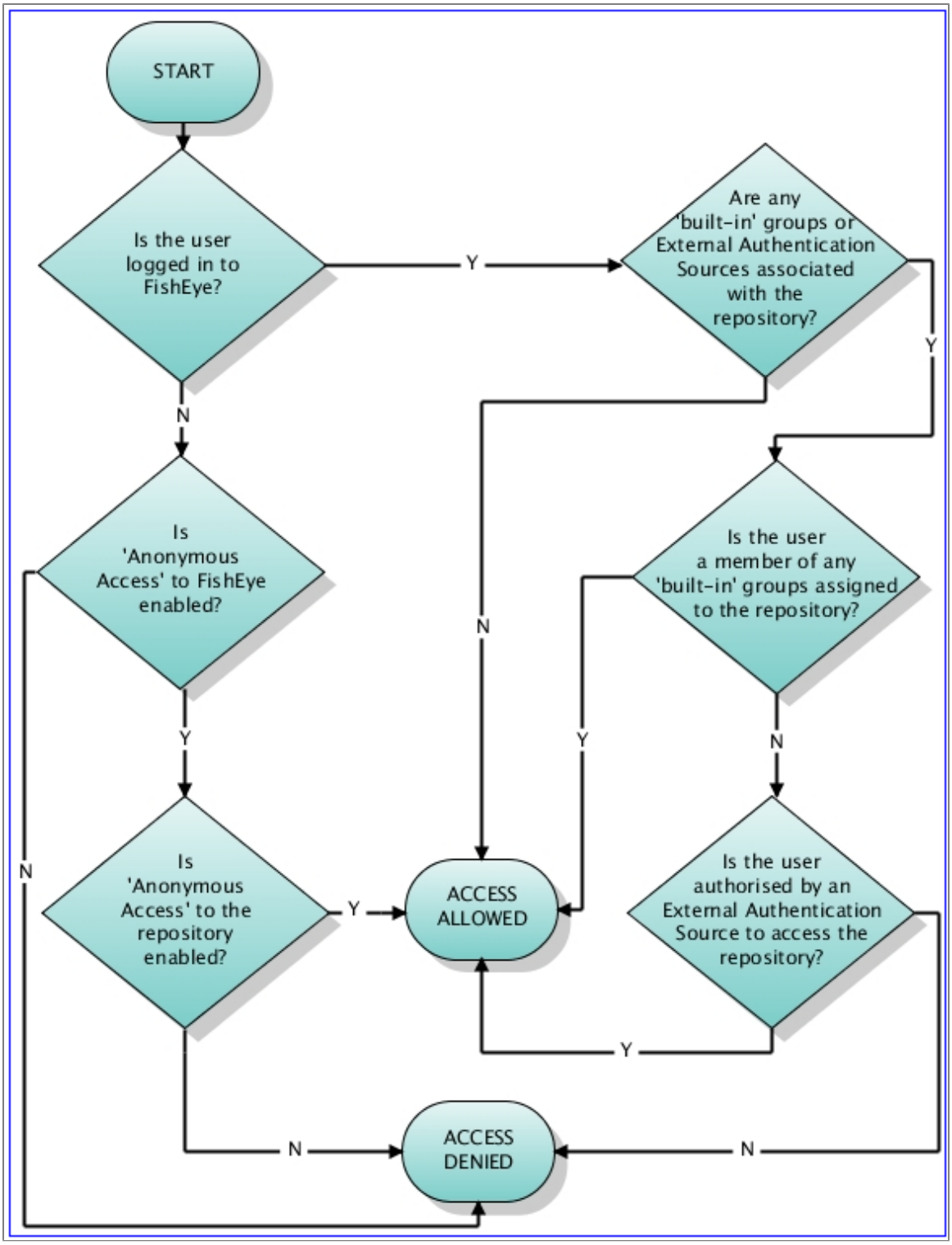
Additionally, when setting up external authentication for large repositories: set "Synchronise users with LDAP" to no. Next turn on "Auto-add" users to on, so that any External user that matches your defined user filter who logs into FishEye, will be added as a user (NOTE: Auto-add does not mean any user will have access to FishEye, only users that can successful 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.

## Understanding security

The following flowchart shows how to determine whether a user is allowed to access a FishEye repository:

*Screenshot: FishEye Security Flowchart*




## Advanced Administration Options

- Command-Line Options
- Configuring Indexing
- Customising FishEye's Look & Feel
- Environment Variables
- Re-indexing your Repository
- Tuning FishEye

## Command-Line Options

A FishEye instance can be managed using the `fisheyectl` 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/fisheyectl.sh command [options]
```

Windows usage:

```
\FISHEYE_HOME\bin\fisheyectl.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):

| | |
|---|---|
| `--config path` | Load configuration from the file at `path`. Default is `$FISHEYE_INST/config.xml`. |
| `--quiet` | Do not print anything to the console. |
| `--debug` | Print extra information to the debug log. |
| `--debug-perf` | Print performance-related information to the debug log. |

The following options are currently available, but will be removed at a later date:

| | |
|---|---|
| `--Xtab-width nchars` | Specifies the number of spaces to use to represent a tab character. The default is 8. |
| `--Xdisable-dirtree-empty-checks` | 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.<br>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 |
| `--Xdisable-content-indexing` | Disable the generation of a full-text index for file content. This prevents urther 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. |

### *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:

```
fisheyectl 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:

```
fisheyectl 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:

```
fisheyectl reindex [options] [repname ...]
```

Requests a full reindex 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`. |

### scannow

The `scannow` command requests an incremental scan from the command line.

Usage:

```
fisheyectl scannow -s [repname ...]
```

Options:

| | |
|---|---|
| `-s` | 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:

  ```
  fisheyectl.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:

```
fisheyectl backup -f [filename]
```

The `backup` command creates a zip archive containing important FishEye configuration files.

Options:

| | |
|---|---|
| `-f filename` | Store the backup.zip to `filename`. Default is `$FISHEYE_INST/backup/backup_yyyyddMMHHmmss.zip`. |

## Configuring Indexing

### *Understanding How FishEye Indexes Repositories*

FishEye's administration interface now 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 FishEye does when FishEye first accesses the repository, and
- **Incremental indexing** - the quick, background process that regularly scans repositories for changes.

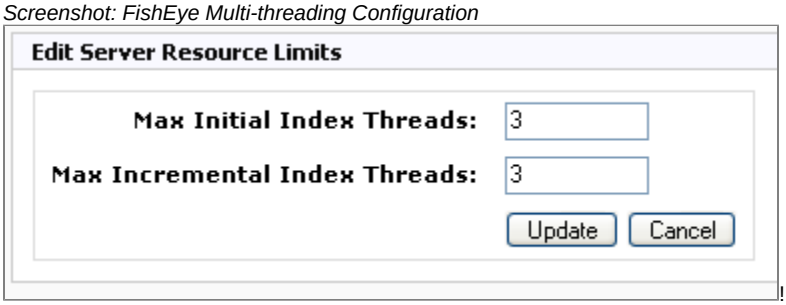
The number of threads available for these indexing operations is now configurable.

Prior to FishEye 1.5, a transient error during processing the initial indexing operation would cause a repository to move to the incremental indexing thread, even when a large amount of indexing remained. This would impact the incremental indexing of other repositories in the FishEye instance. In FishEye 1.5 this has been corrected. A repository will remain on the initial indexing thread until it has successfully completed the initial indexing operation.

### *Setting FishEye's Total Threads*

To configure the number of threads FishEye uses for indexing, open the FishEye '**Admin Screen**', then click '**Server Settings**'. Now, click '**Edit Details**' under '**Resource Limits**'. You can increase the number of threads available for each indexing phase independently. Enter the desired number of threads for each process and Click '**Update**'.

*Screenshot: FishEye Multi-threading Configuration*




### *Choosing a Thread Setting*

As 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 modes. This is equivalent to the functionality in FishEye prior to version 1.5.

For more information on managing FishEye's performance, see the FishEye Tuning page.

*This feature was introduced in FishEye 1.5.*

## Customising FishEye's Look & Feel

FishEye enterprise 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.

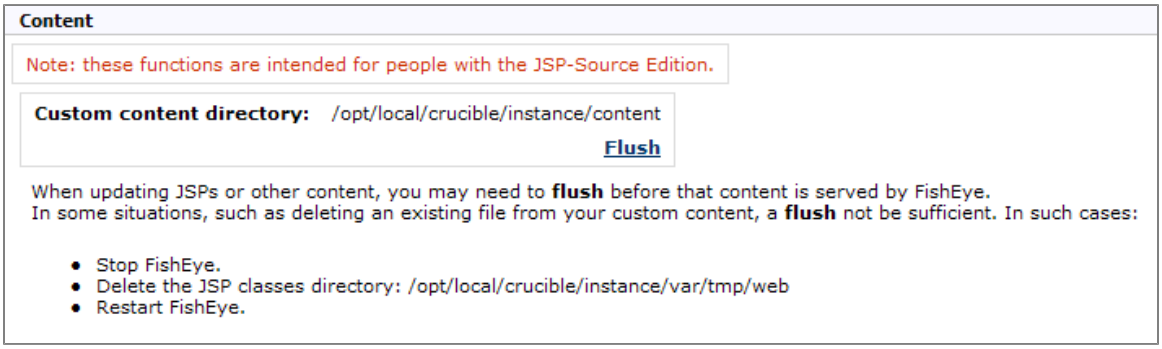
> You must install a JSP source build – you cannot copy JSP files from a source build into an existing ordinary installation.

### *Customising Content*

You can modify any of the files in `FISHEYE_HOME/content/`. However we strongly recommend that you use separate `FISHEYE_HOME` and `FISHEYE_INST` directories (as described in the Installation Guide), and that you store your modified files in `FISHEYE_INST/content` (If you use `FISHEYE_INST/content`, you only need to keep your modified `JSP/HTML` 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 into the FishEye Administration screens, click '**Content**' on the '**Admin Menu**' and follow the instructions on the 'Content' page.

*Screenshot: Content Page*




## Environment Variables

Environment variables are system-wide settings that are required for certain applications. Instructions on Setting Environment Variables are here. The following is a list of the environment variables used by FishEye.

### *JAVA_HOME*

The JAVA_HOME environment variable is 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 the instructions on setting JAVA_HOME.

### *FISHEYE_OPTS*

FishEye uses the FISHEYE_OPTS environment variable 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:

```
FISHEYE_OPTS=-Xmx1024m -XX:MaxPermSize=128m
```

This would give FishEye a max of 1024 MByte heap, a Max permanent generation size of 128m. See Tuning FishEye for more information.

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.

### *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*

The FISHEYE_LIBRARY_PATH environment variable tells FishEye where it should look to load any additional native libraries.

### *FISHEYE_INST*

The FISHEYE_INST variable tells FishEye where to store its data. If you wish to separate FishEye's data from its application files in FISHEYE_HOME, you should use this variable.

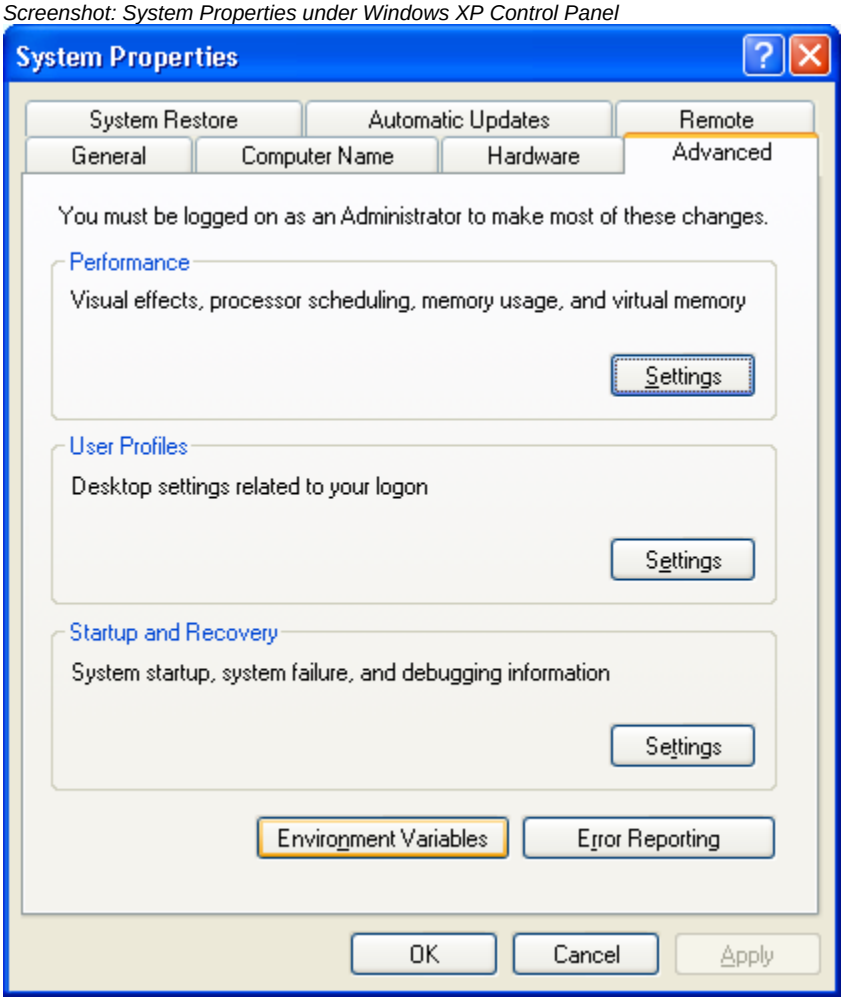
Read more about using FISHEYE_INST in the Installation Guide.

### Setting Environment Variables under Windows XP

(*If you are running Fisheye as a windows service you need to refer to the instructions here*)
(*Linux instructions are here*)

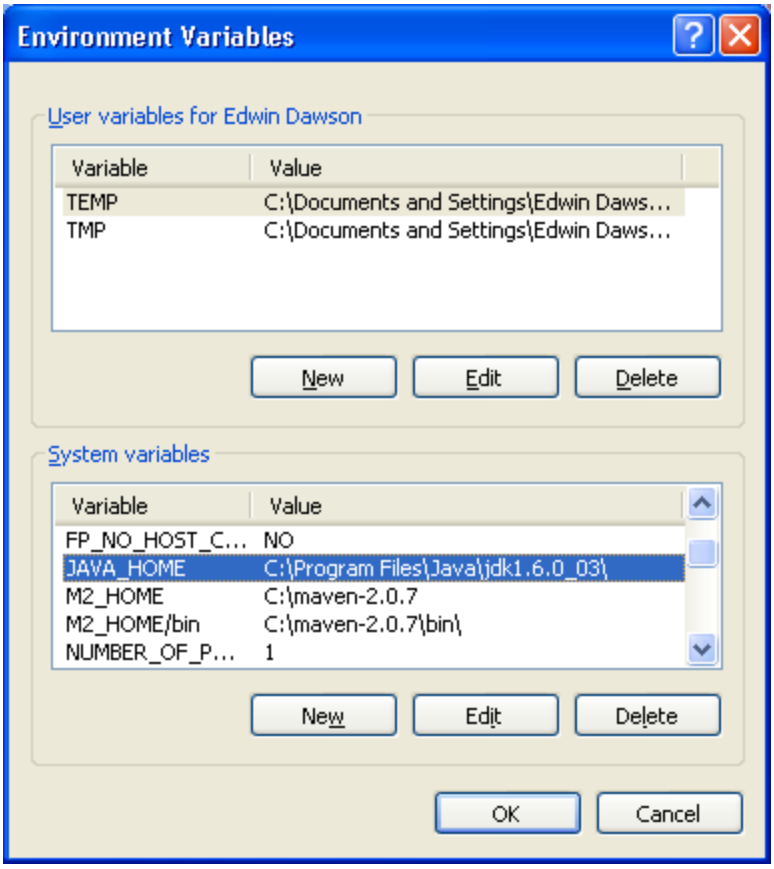
1. Click 'Start' > 'Control Panel' > 'System'.

*Screenshot: System Properties under Windows XP Control Panel*




2. Click the 'Advanced' tab.
3. Click the 'Environment Variables' button.

*Screenshot: Environment Variables under Windows XP Control Panel*



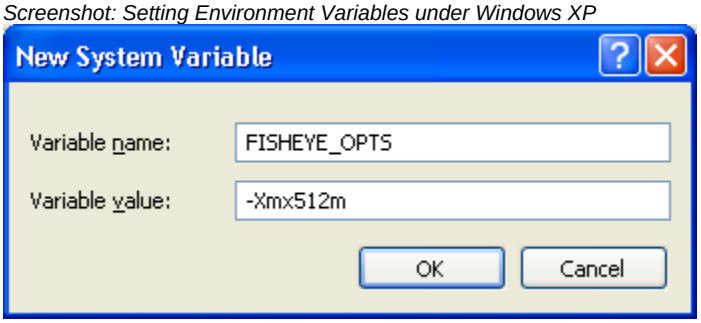

4. Click 'New'.
5. In the 'Variable name' field, enter the name of the environment variable, for example

```
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
```

*Screenshot: Setting Environment Variables under Windows XP*




7. Restart the computer.

### Setting Environment Variables for Windows Services

Please note, that if you do run as a service, then any Environment Variables that you want to set, need to be set in your `FISHEYE_HOME/wrapper/conf/wrapper.conf` file.

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

```
# JDK 1.5 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"
```

For example if you wish to add a FISHEYE_INST environment variable or add the java parameter "MaxPermSize", or the -Xrs options (should be used if running FishEye as a service under Windows, to prevent the JVM closing when an interactive user logs out) then it would be something like:

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

Your memory settings can also be found in this file:

```
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=32

# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=256
```

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

### *Setting Environment Variables under Linux or UNIX based platforms*

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.

# Re-indexing your Repository

This page contains instructions on how to re-index your repository in FishEye.

Re-indexing involves FishEye doing an intensive scan of the repository contents, so it can quickly show the repository data via FishEye's web interface. Re-indexing may be required in a number of situations such as server maintenance, changes in your repository, major setting changes and upgrades to FishEye.

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 by able to view the repository being indexed. As a result, Crucible users will not be able to carry out reviews unless you have Crucible set to store all revisions offline.

### *Performing a Manual Re-index*

**To re-index your repository,**

1. Open the '**Admin Menu**', by clicking the '**Administration**' link in the footer of any FishEye page.
2. The '**Repository List**' screen opens by default. In the main window, the list of configured repositories is shown. Click '**View**' next to the desired repository, then click on maintenance.
3. Select one of the re-indexing options discussed below.
4. The repository will be re-scanned, a process that may take hours or possibly days, depending on the size of the repository, network speed, machine performance and other factors.

How do I mitigate long re-index times when upgrading? See the FAQ page on this topic.

### *Indexing Settings*

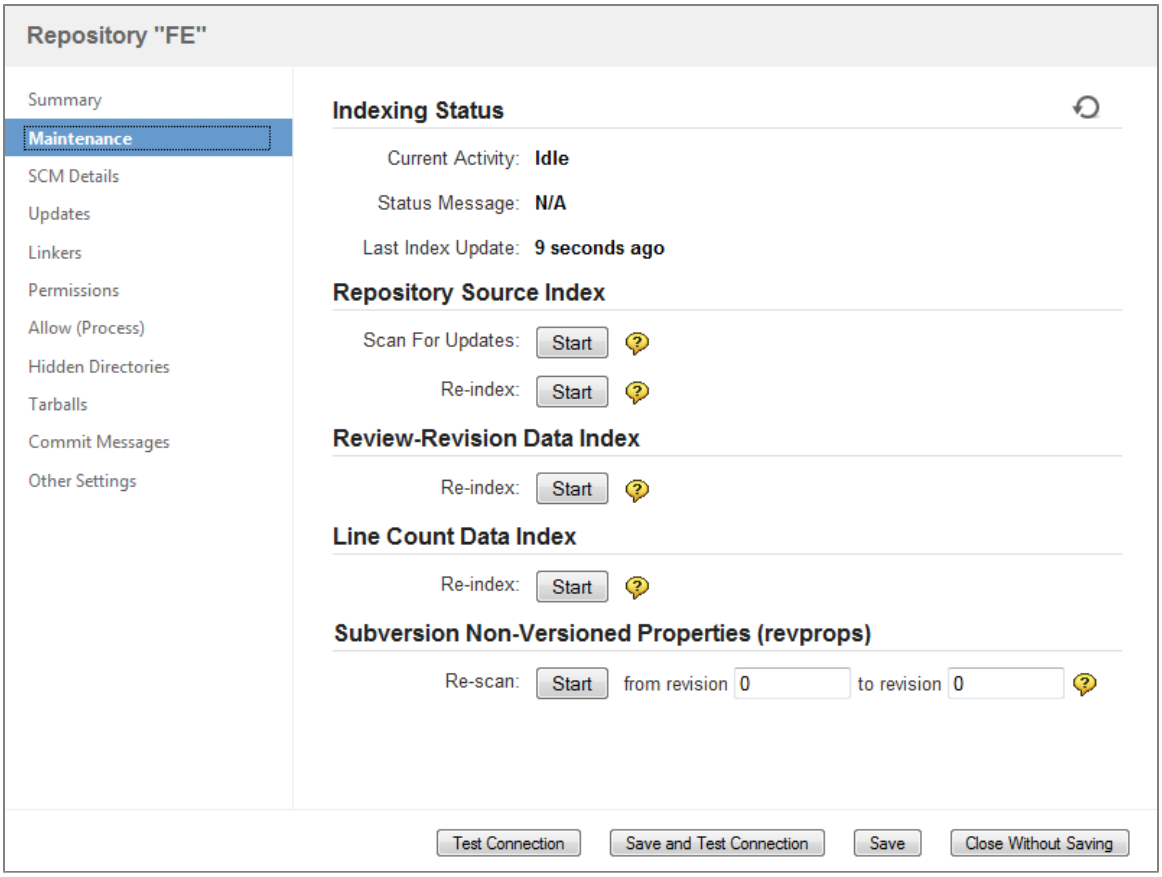
Settings and administrative operations for repository indexing are located on the Indexer page.

**To access indexing functions for a repository,**

1. Click the menu labelled with your user name in the the FishEye/Crucible header, and click the '**Administration**' option. You will need to be logged in as an administrator to see this link. The FishEye/Crucible administration console will be displayed.

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 '**Maintenance**' tab. Indexing information for the repository will be displayed as well as indexing functions (see screenshot below).
    - '**Repository Source Index**':
        - Click the '**Start**' button next to '**Scan for Updates**' to run a repository scan (also referred to as indexing) now. If the repository has already been indexed, it will be 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.
        - Click the '**Start**' button next to '**Re-index**' to delete the current cache and re-index the repository from the beginning. This action will also restart the repository.
    - '**Review-Revision Data Index**' — *(applies only when using Crucible with FishEye)* Click the '**Start**' button next to '**Re-index**' to re-index all the Crucible review data associated with the current repository.
    - '**Line Count Data Index**' — Click the '**Start**' button next to '**Re-index**' to 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)* Set the revision numbers to scan from and to, and click the '**Start**' button next to '**Re-index**' to rescan non versioned properties (revprops). 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*




# Tuning FishEye

This page contains information about improving FishEye's performance.
On this page:



### *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.

### *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.

### *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 (in some cases weeks), 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.

### *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 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.

# Managing Plugins

A **plugin** is an add-on to the core FishEye code, used to extend the FishEye functionality. Some plugins are shipped with FishEye, others are available for you to install yourself.

You can develop additional plugins (see the FishEye developer's guide) or download existing plugins from the Atlassian Plugin Exchange, and install them into your FishEye system.

**On this page:**



## Managing your Plugins

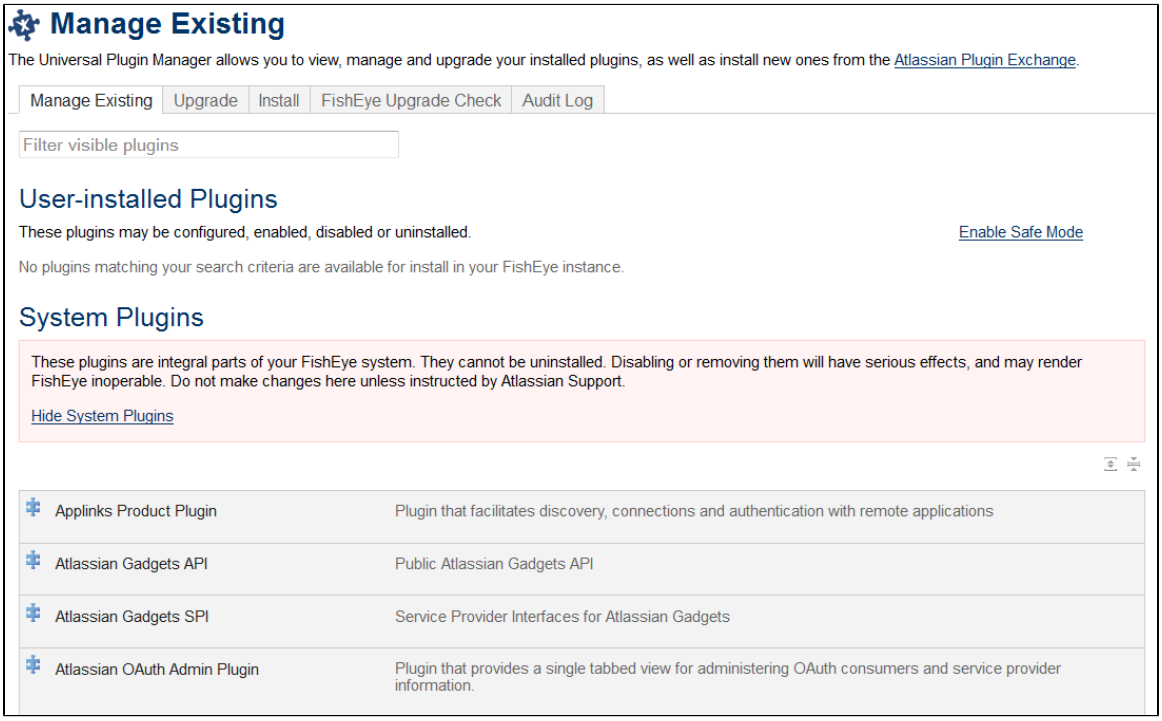


## About the Universal Plugin Manager (UPM)

The **Universal Plugin Manager (UPM)** provides you with a powerful and user-friendly interface to manage your plugins. The Universal Plugin Manager itself is a plugin, which contains a number of modules that are implementations of the Atlassian REST plugin module type. It allows you to perform common plugin tasks, such as:

- Enabling/disabling plugins and their plugin modules.
- Installing new plugins.
- Configuring advanced plugin options.
- Finding out-of-date plugins and updating them.
- Checking the compatibility of your installed plugins against newer versions of the application.

The Universal Plugin Manager also interfaces with the Atlassian Plugin Exchange, so you can browse the wide range of plugins available for your application from within your application. You can install any of these plugins with a single click, or upload your own plugins using the Universal Plugin Manager as well.




*Screenshot above: The Universal Plugin Manager in FishEye*

## Notes

- **Plugin Safety.** Plugins are very powerful. They can change the behaviour of almost any part of the FishEye server. This makes it very important that you trust a plugin before you install it. Always be aware of where and who a plugin comes from.

- **Troubleshooting.** Having problems with the Universal Plugin Manager? Try the Universal Plugin Manager FAQ.

# Viewing your Installed Plugins

Using the Universal Plugin Manager (UPM) you can see a list of plugins installed on your FishEye site. These include plugins that are bundled with FishEye as well as any plugins that you have installed. Both enabled and disabled plugins are displayed.

**On this page:**



## Viewing your Installed Plugins

**To view your installed plugins:**

1. Click the '**Manage Existing**' tab. You will see a list of the plugins installed in your application.
    - The plugins are grouped into 'User-installed Plugins' and 'System Plugins'.
    - You can filter your list by entering keywords in the 'Filter visible plugins' text box.
    - The list of 'System Plugins' will be hidden by default. Click the '**Show System Plugins**' link to see them.
    - Enabled plugins will have this icon: Disabled plugins will have this icon:
    - Click the name of a plugin to view the plugin's details.
    - Click '**Enable Safe Mode**' to run your application in safe mode. This mode disables all user installed plugins.

## Viewing a Plugin's Details

You can view the details for a plugin by clicking the name of a plugin in the list of installed plugins. The summary contains a short description of the plugin as well as links to plugin operations and related information.

- **Plugin key** – A unique key that identifies each plugin in the system.
- **Developer** — The name of the plugin developer and a link to the developer's home page, if provided by the plugin developer.
- **Plugin version** – The version of the plugin currently installed.
- **Manage plugin modules** — Click this link to display the plugin modules below the plugin summary. A module is a component of the plugin. This link will only appear if the plugin has modules. If you want to enable or disable a plugin module, hover your mouse over the module and click the '**Enable**'/'**Disable**' button for that module.
- **Configure** – Click this link to display the configuration settings for the plugin. This link is disabled if the plugin is disabled. Please note that not all plugins have settings that can be configured through the Universal Plugin Manager.
- **Disable** – Click this button to disable the plugin in your application. This button will only appear if the plugin is enabled.
- **Enable** – Click this button to enable the plugin in your application. This button will only appear if the plugin is disabled.
- **Uninstall** – Click this button to uninstall the plugin from your application. This button will only appear for user installed plugins.

## Notes

- **What is the difference between a 'system plugin' and a 'user installed plugin'?**
    - System plugins are shipped with the application. These plugins are essential for the functioning of the system. Although you can disable some of them, you should not do so unless instructed by an Atlassian support engineer. *Note:* Not every system plugin can be disabled. You cannot uninstall any system plugins.
    - User installed plugins are those which have been installed in the application after it was set up. You can install a plugin either by uploading a JAR file or via the Universal Plugin Manager. You can uninstall these plugins.

***Related Topics***

Managing Plugins

# Installing a Plugin

This page describes how to install a plugin in FishEye. You can use plugins to customise and extend the functionality of your application.

A number of plugins are available from the Atlassian Plugin Exchange. You can also create your own as described in the FishEye developer's guide.

**On this page:**



## Adding a Plugin from the Atlassian Plugin Exchange

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To find a plugin in the Atlassian Plugin Exchange and add it to FishEye:**

1. Click the '**Install**' tab in the UPM. You will see a list of featured plugins.
2. Search for your plugin as follows:
   - Enter some keywords that describe the plugin in the '**Search the Plugin Exchange**' search box and press 'Enter'.
   - Alternatively, browse to the desired plugin in the list. You can choose '**Featured**', '**Popular**', '**Supported**' *(by Atlassian)* or '**All available**' from the '**Plugins to show**' dropdown to see a different list of plugins.
3. Click the '**Install**' button for the desired plugin to add it to your application. A confirmation message and the plugin details will appear when the plugin is installed successfully.
   ***Note:*** You may need to restart your application for your change to take effect. The Universal Plugin Manager will inform you if this is the case.
   ***Note:*** Not all plugins can be automatically installed. Some required manual installation. These plugins will have a 'Download' button instead of an 'Install' button. In these cases, you should read and follow the plugin's installation instructions.

## Uploading your own plugin

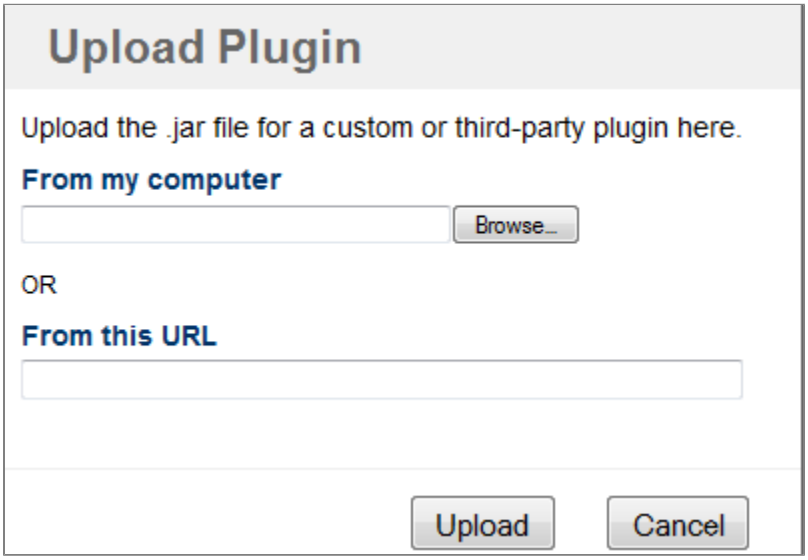
**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To upload your own plugin to FishEye:**

1. Click the '**Install**' tab in the UPM. You will see a list of featured plugins.
2. Click the '**Upload Plugin**' link. The 'Upload Plugin' window will appear.
3. Specify the location of your plugin:
   - If the plugin you want to install is on your computer, use the '**Browse**' dialogue to choose the plugin JAR file.
   - If you want to install a plugin from a remote location, enter the URL of the plugin JAR file in the '**From this URL**' text box.
4. Click the '**Upload**' button to upload and enable your plugin. A confirmation message will appear when the plugin is successfully installed.
   ***Note:*** You may need to restart your application for your change to take effect. The Universal Plugin Manager will inform you if this is the case.

*Screenshot: Uploading a new plugin*



### Notes

- In FishEye, you can install and uninstall both version 1 and version 2 plugins using the Universal Plugin Manager. You will see an 'Install' or an 'Uninstall' button.
- Some entries that you find listed in the Universal Plugin Manager **are not actually plugins**. These entries will show a 'Download' button which allows you to download the application to your desktop and run it.

***Related Topics***

Managing Plugins

## Configuring a Plugin

A number of FishEye plugins offer advanced configuration options. If you have one of these plugins installed in your application, you can view and update these configuration options via the Universal Plugin Manager (UPM).

If you would like to disable or enable a plugin, please refer to Disabling or Enabling a Plugin.

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To configure a plugin in FishEye:**

1. Click the '**Manage Existing**' tab.
2. Locate the plugin that you want to configure and click its title. The plugin details section will expand.
3. Click the '**Configure**' link for that plugin.
    - The link will be disabled if the plugin is disabled.
    - If there is no 'Configure' link, then there are no advanced configuration options available for that plugin.
4. The advanced configuration options for the plugin will appear. Update the configuration settings as desired and save your changes.
   **Note:** The advanced configuration screens are provided by the *plugin*. If you encounter any problems after you click the '**Configure**' link, the plugin is responsible for the issue, not the Universal Plugin Manager.

***Related Topics***

Managing Plugins

## Uninstalling a Plugin

If you wish to completely remove a plugin from FishEye, you can uninstall it via the Universal Plugin Manager (UPM). If you only want to temporarily remove it, you may wish to disable the plugin instead.

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To uninstall a plugin from FishEye:**

1. Click the '**Manage Existing**' tab. You will see a list of the plugins installed in your application.
2. Click the name of the plugin that you wish to uninstall. The plugin details will appear.
3. Click the '**Uninstall**' button. The information summary will display an 'Uninstalling' message and the plugin will be uninstalled from your application.

***Related Topics***

Managing Plugins

## Disabling or Enabling a Plugin

The Universal Plugin Manager (UPM) allows you to disable a plugin in FishEye without permanently removing it. You can also enable any plugins that have been previously disabled. If you want to add or remove a plugin from your FishEye site, please refer to Installing a Plugin or Uninstalling a Plugin respectively.

You can also disable all user installed plugins in FishEye, by enabling **safe mode**. This may help you to diagnose a plugin-related problem more easily.

**On this page:**



## Disabling a Plugin

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To disable a plugin in FishEye:**

1. Click the '**Manage Existing**' tab. You will see a list of the plugins installed in your application. Enabled plugins will have this icon: 
2. Locate the plugin that you want to disable and click the title to expand the plugin details section.
3. Click the '**Disable**' button.
4. Once a plugin has been disabled, you *may* need to restart your application for your change to take effect. If so, you will see a message for the plugin, 'Disabled, requires restart'.
   Once the plugin is fully disabled, you will see an '**Enable**' link for the plugin.

## Enabling a Plugin

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To enable a plugin in FishEye:**

1. Click the '**Manage Existing**' tab. You will see a list of the plugins installed in your application. Disabled plugins will have this icon: 
2. Locate the plugin that you want to enable and click the title to expand the plugin details section.
3. Click the '**Enable**' button.
4. Once a plugin has been enabled, you *may* need to restart your application for your change to take effect. If so, you will see a message for the plugin, 'Enabled, requires restart'.
   Once the plugin is fully disabled, you will see a '**Disable**' link for the plugin.

## Disabling/Enabling all User Installed Plugins (Safe Mode)

Running your application in safe mode disables all user installed plugins at once. By "user installed plugins", we mean plugins that were not shipped with FishEye but were installed later via the UPM.

All plugins that were disabled when you entered safe mode will be re-enabled when you exit safe mode.

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To enable safe mode in FishEye:**

1. Click the '**Manage Existing**' tab. You will see a list of the plugins installed in your application.
2. Click the '**Enable Safe Mode**' button.
3. Click the '**Continue**' button in the confirmation window. All user installed plugins will be disabled and your application will now be running in 'Safe Mode'.
4. You can now make changes to your installed plugins as desired. For example, you may want to enable/disable specific plugins or plugin modules.
5. Exit safe mode by clicking one of the links in the Safe Mode banner:
   - Click '**Exit Safe Mode and restore the previous configuration**' to restore your plugin configuration to its state before you entered Safe Mode.
   - Click '**Exit Safe Mode and keep the current configuration**' to keep all changes made to your plugin configuration during Safe Mode.

***Related Topics***

Managing Plugins

# Checking Plugin Compatibility for FishEye Upgrades

You can use the plugin manager's '**Upgrade Check**' to verify that your plugins will still work after a FishEye upgrade.

For example, if you are thinking of upgrading from FishEye 2.5 to FishEye 2.6, the upgrade check can tell you the following:

- Installed plugins that are compatible with FishEye 2.5 and FishEye 2.6.
- Installed plugins that are not compatible with FishEye 2.6 but will be compatible with FishEye 2.6 if you upgrade them.
- Installed plugins that are not compatible with FishEye 2.6, even if you upgrade them to their latest version.

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To check the compatibility of your plugins against different FishEye versions:**

1. Click the '**Upgrade Check**' tab.
2. In the '**Check compatibility for**' dropdown menu, select the version of your application to check the plugins against.
3. Click the '**Check**' button.
4. The page display any of your installed plugins that are not compatible with the selected application version. The compatibility checker will also check the compatibility of the latest available version of each plugin (if not already upgraded). You can click on the name of any of the plugins to view more information about the plugin.

   The plugins are grouped into sections under the following headings:
   - **Incompatible** – The installed versions of these plugins are not compatible with the selected application version. There are currently no plugin upgrades available that are compatible with the selected application version.
   - **Compatible, if upgraded** – The installed versions of these plugins are not compatible with the selected application version. However, the plugins will be compatible if you upgrade them. There are buttons allowing you to upgrade these plugins.
   - **Compatible if both <the application> and the plugin are upgraded** – The installed versions of these plugins are not compatible with the selected application version. There is a plugin compatible with the newer application version, but it is not compatible with the application version you are currently running. You must upgrade the application and *then* upgrade the plugin. There are buttons allowing you to disable these plugins before proceeding with the upgrade.
   - **Compatible** – The currently installed versions of these plugins are compatible with the selected application version.
   - **Unknown** – These plugins may or may not be compatible with the selected application version. If a plugin is not registered with the Atlassian Plugin Exchange, the Universal Plugin Manager cannot check its compatibility with different application versions.

***Related Topics***

Managing Plugins

# Upgrading your Existing Plugins

Plugins are often developed separately from FishEye. You may wish to upgrade your plugins to more recent versions to allow them to work with your FishEye version or simply to take advantage of new features in a plugin version. The Universal Plugin Manager (UPM) provides you with a list of plugins that have available upgrades and allows you to upgrade each plugin individually or in bulk.

**On this page:**



## Upgrading a Plugin

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To upgrade a plugin in FishEye:**

1. Click the '**Upgrade**' tab. The plugin upgrades page will appear.
   - If there is a later version of a plugin that you have already installed, this page will show the latest **compatible** version of the plugin.
   - You can click the plugin name to expand the row and see more information about the plugin.
   - You can filter your list by entering keywords in the 'Filter plugins' text box.

2. Click the '**Upgrade Now**' button next to the relevant plugin to update it to the plugin version shown.

### Upgrading All your Plugins

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To upgrade all available plugins in FishEye:**

1. Click the '**Upgrade**' tab. The plugin upgrades page will appear.
    - If there is a later version of a plugin that you have already installed, this page will show the latest **compatible** version of the plugin.
    - You can click the plugin name to expand the row and see more information about the plugin.
    - You can filter your list by entering keywords in the 'Filter plugins' text box.
2. Click the '**Upgrade All**' button to update every plugin to the plugin versions shown.
   ***Note:*** Some plugins cannot be installed via the Universal Plugin Manager. You must install these plugins manually. These plugins will not be upgraded automatically.

### Notes

- If you are considering upgrading FishEye, you can use the Universal Plugin Manager to check the compatibility of your plugins with your desired FishEye version. See Checking Plugin Compatibility for FishEye Upgrades.

***Related Topics***

Managing Plugins

## Viewing the Plugin Audit Log

The Universal Plugin Manager (UPM) keeps a log of all plugin activity for your FishEye site. Such activities may be adding plugins, enabling plugins, and so on. You can adjust the period of time for which log entries are kept.

**On this page:**



### Viewing the Plugin Audit Log

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To view the plugin audit log:**

1. Click the '**Audit Log**' tab. The plugin audit log will appear, showing the 25 most recent entries.
2. Use the arrows if you want to view older entries.
3. Click the orange RSS icon if you want to receive the audit log activity in an RSS feed.

### Configuring the Plugin Audit Log

**To access the Universal Plugin Manager in FishEye:**

1. Log in as an administrator and go to the FishEye administration screen.
2. Click '**Plugins**' in the left-hand menu.

**To configure the length of time for which log entries are kept:**

1. Click the '**Audit Log**' tab. The plugin audit log will appear.
2. Click the '**Configure purge policy**' link.
3. In the '**Purge audit log after**' field, specify the number of days for which you wish to keep the logs.
4. Click the '**Confirm**' button.

***Related Topics***

Managing Plugins

# 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.

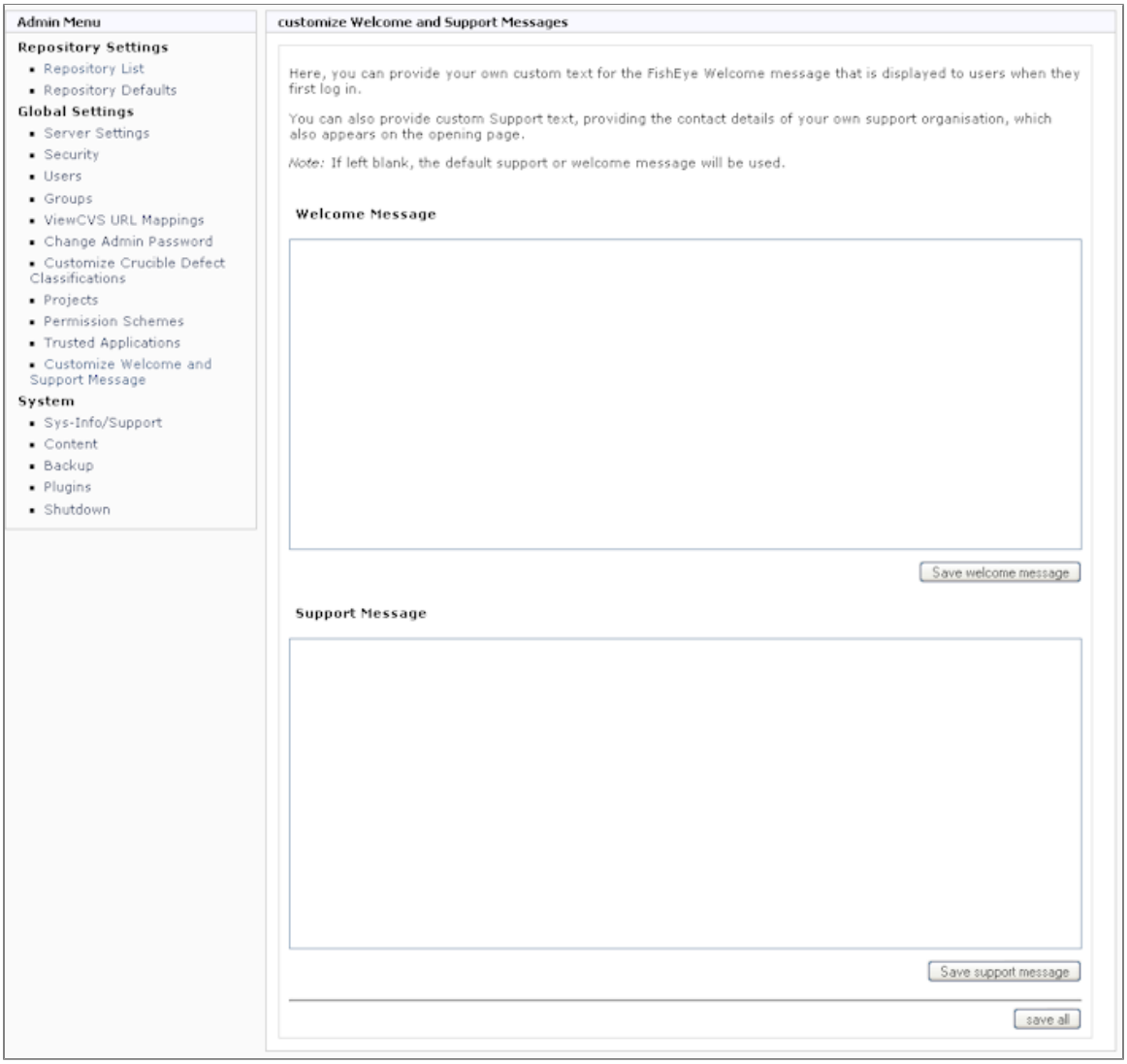
To customise the welcome message, access the administration page, and click '**Customize Welcome & Support Message**' under '**Global Settings**' on the left navigation bar.

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*




### 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>
```

# 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.

*On this page:*



## Backing Up FishEye Data

### The FishEye Admin Interface Backup Process

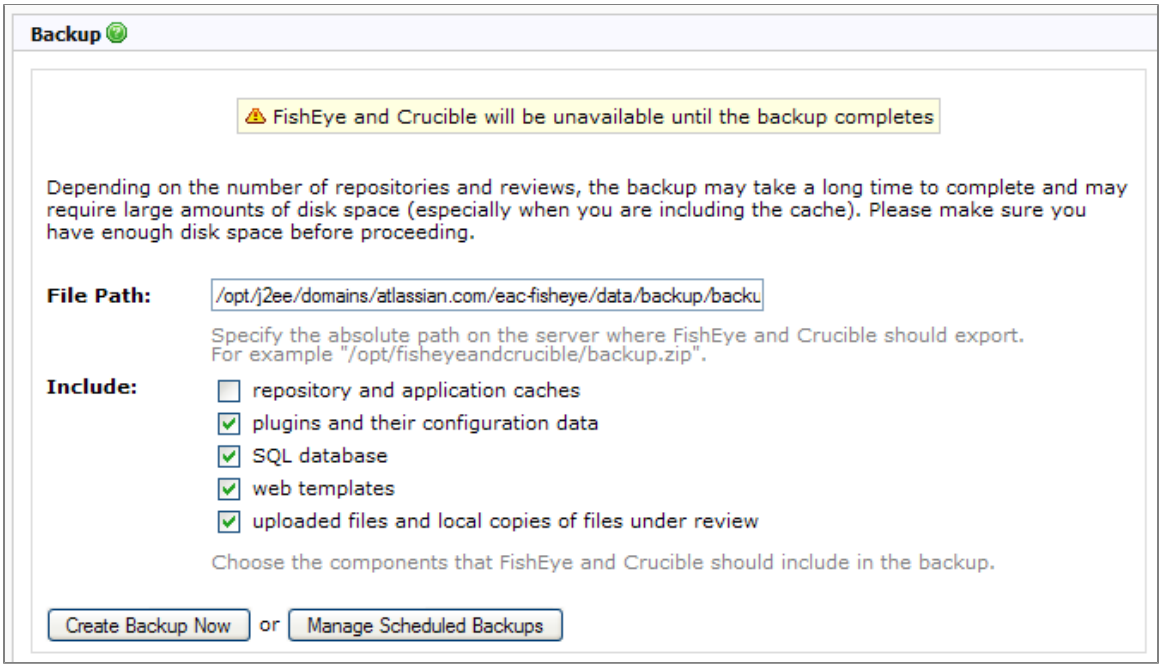
1. Navigate to the FishEye '**Admin**' screen (click the '**Administration**' link in the footer of any FishEye page).
2. On the Admin screen, click '**Backup**' under the '**System**' heading in the left navigation bar. The Backup screen opens.
3. On the Backup screen, the '**File Path**' field indicates where the backup file (in .zip format) will be stored. You can manually edit this path to change it. Under the heading '**Include'**, a list of checkboxes is shown, with the following items:

- Plugins and their configuration data
- SQL database
- Web templates
- Uploaded files and local copies of files under review.
- Repository and application caches.

  Repository and application caches contain temporary data stored from repository scans and library caches that improve startup time. Both will be recreated automatically by re-scanning the source repositories, so the backup files can be reduced by a significant amount by excluding these (if the cost of re-scanning is acceptable).

4. Once you have chosen your options, click '**Create Backup Now**'.

*Screenshot: The FishEye Backup Screen*




## The FishEye Command Line Backup Process

Your FishEye instance must be running during the backup.

1. Open a command line interface on the FishEye server computer.
2. Navigate to the `FISHEYE_HOME/bin/` directory.
3. Run the backup command on the command line with the desired options.
4. The backup is created as a new Zip archive file and placed in the `FISHEYE_INST/backup/` directory.

   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.

## Components of a FishEye Backup

The FishEye backup is highly configurable and allows for many different configurations. This table shows the various components of the backup, what they are for and how they can be used.

| Component | Purpose | Defaults |
|---|---|---|
| SQL Database | Refers to the SQL content database (used by both FishEye and Crucible and containing all user profile data, reviews and their comments). | Backed up by default. |
| Cache | The cache contains data that reflects the state of FishEye's repositories. Without it, FishEye must re-scan its repositories after a backup is restored. The cache also contains OSGI library data that increases startup time. These too can be excluded and will be generated automatically when the application is started. | The cache is not backed up by default as it tends to be large (running a risk of pushing the maximum file size for Java backups), whilst also representing replaceable data. |
| Plugins | Plugins are 3rd-party extensions that you may have installed, and configuration for all plugins (this includes configuration for Crucible's set of standard plugins). | Configuration data for all plugins are backed up by default, as well as all plugins installed in `FISHEYE_INST/var/plugins/user`. |
| Templates | In this context, these are custom freemarker templates that you or your users have created. They live in `FISHEYE_INST/template`. | Templates are backed up by default. You can choose to exclude them from the backup if your templates directory is covered by some other backup mechanism. |
| Uploads | In this context, uploads refers to files which are added to Crucible via the web interface (such as patch file reviews). It also includes each repository-backed file that went under review, when Crucible is configured to make a local copy of every reviewed file. | Uploads are backed up by default. You can choose not to back them up for example when the `FISHEYE_INST/var/data/uploads` directory is already covered by some other backup mechanism. This item only applies when using Crucible with FishEye. |

Note that the backup will always include the configuration data (`config.xml`), your license file and the FishEye user data.

## Backup Command Line Options

These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyectl.bat` and use the correct slashes. Run the command from the `FISHEYE_HOME/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/bin/` directory:

```
$ ./fisheyectl.sh backup --help
```

| Option | Switch | Description | Default setting |
|---|---|---|---|
| **Quiet mode** | -q OR --quiet | Suppresses output | No |
| **Output filename** | -f OR --file | Specify a different path and filename to the `FISHEYE_INST/backup/backup_YYYY-DD-MM_HHmm.zip` file. When filename is omitted, the backup filename contains the date and time. | `FISHEYE_INST/backup/` is the default directory. |
| **Compression level** | --compression OR -c | Sets the Zip compression level, from 1-9. Runs at level 6 if no argument is passed. | Yes (6) |
| **Anonymise** | -a OR --anonymise | Anonymises the SQL database by replacing all text with 'x'. This is only useful when sending a backup to Atlassian as part of a support case. **Please do not anonymise data unless the Support Engineer handling your support case has specifically requested the data anonymised (as often anonymised data will not help reproduce the issue).** | No |
| **Cache Backup** | --cache | Include the repository caching files in the backup. These hold information gained from scanning the repositories and can be quite large (many gigabytes). However, it can shorten the time needed to re-scan the repositories after data is restored. | No. By default, the cache data is excluded from backups. |

### 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/bin/` directory.

**Backing up with compression of 9, quiet mode and setting an output location**

```
$ ./fisheyectl.sh backup --compression 9 -q -f /application_backups/fisheye/20090215.zip
```

**Backup including cache data (also includes all default components)**

```
$ ./fisheyectl.sh backup --cache
```

**Restoring a backup with cache data (also restores all default components)**

```
$ ./fisheyectl.sh restore --cache
```

### Advanced Backup Command Line Settings

In some cases it might be preferable to only backup a limited set of items. This could be useful when your instance uses an external database such as MySQL Enterprise Server 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. |
| **Show help** | --help OR -h | Shows inline help on the command line. | No |

### Known Limitations

Please note that the below limitations are common for any Java based backup tool.

**Archives Containing Over 65535 Files**
Versions of Java earlier than v1.6 (b25) are incapable of handling zip files that contain more than 65,535 files. The solution for this problem is to either upgrade to a version of Java later than v1.6 (b25), or ensure that the archive does not exceed the threshold (contains less than 65,535 files). The FishEye cache (not included in backups by default) can be a contributor of many small files. Hence, exclude the cache from backups if this is likely to be a concern.

**Archives Larger Than 4GB**
Java has trouble reading and writing zip files that are larger than 4GB. As of release 1.5 Java appears capable of reliably creating archives that are over 4GB, but remains unable to extract them. For details see Sun's bug report. Also be aware of the fact that some file systems (including FAT32) have trouble with files larger than 4GB.

As a workaround, make sure you do not create archives that are larger than 4GB. The FishEye cache (not included in backups by default) can be a contributor of a lot of small files (although these tend to compress very well). If you still want to archive everything and end up with an achive that is too large, consider creating separate backups for the FishEye cache and uploaded files respectively.

## Scheduling FishEye Backups

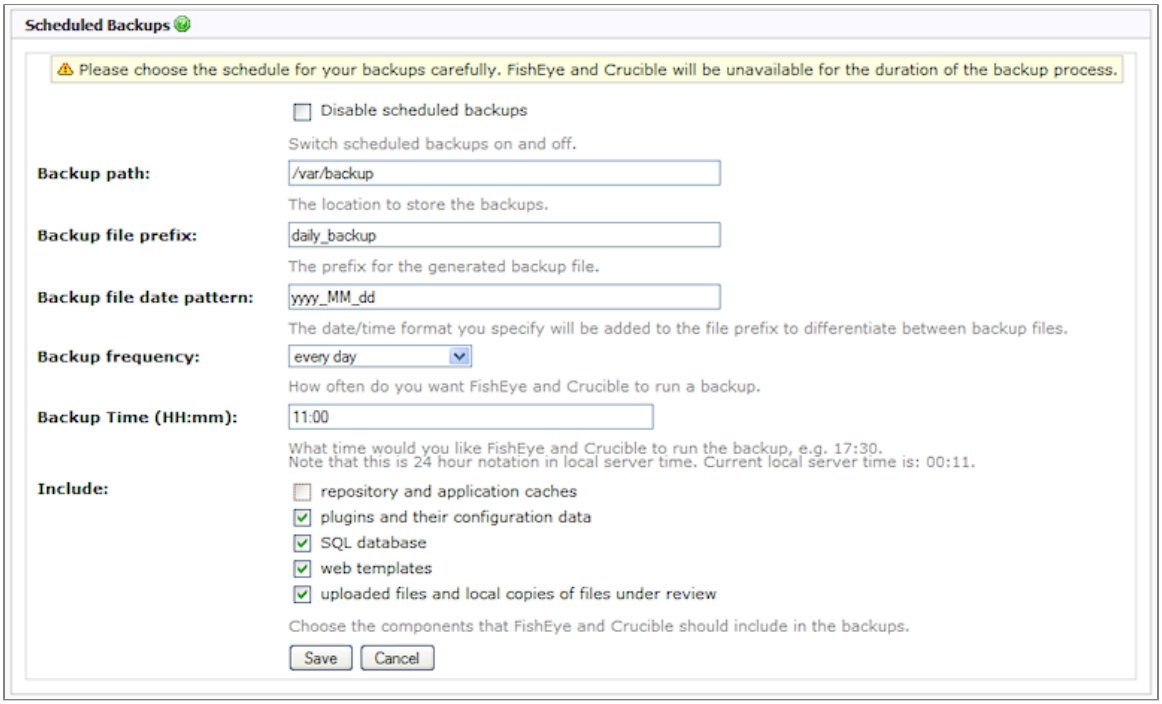
To set a schedule for automatic backups, open the administration screen and click '**Backup**' under '**System**' on the left navigation bar. The 'Backup' page opens. Now, click the link '**Manage Scheduled Backups**' at the bottom of the page. The 'Scheduled Backups' page opens.

On the 'Scheduled Backups' page, click '**Edit**' to adjust the backup schedule. Set the desired options and click '**Save**'.

The options for scheduled backups are detailed in the table below.

| Option name | Description | Allowed Values |
|---|---|---|
| **Disable Scheduled Backups** | Stops regular backups from taking place. | On (disabled) or Off (enabled) |
| **Backup path** | The path where the backup .zip file will be stored. | Any system or network path that FishEye or Crucible can access. |
| **Backup file prefix** | Characters that will be added to the beginning of the backup file name. | Any string of characters that can be used as part of a filename on the local operating system. |
| **Backup file date pattern** | Sets a date for the next (or initial) backup to take place. | Any valid date in the format `yyyy_MM_dd` (year, month, day of the month). |
| **Backup frequency** | Sets how often the backup will take place. | Can be set to '**every day**', '**every Sunday**', '**Monday to Friday**' and '**first day of the month**'. |
| **Backup time (HH:mm)** | The time when the backup will take place. | Any valid 24-hour time in the format `HH:mm` (hours, minutes). |
| **Include** | Specifies which items must be included in the backups (these components are explained at the top of this page). | As per the options for regular on-demand backup (These components are explained at the top of this page). |

*Screenshot: Scheduling Backups in FishEye*




Be aware that scheduled backups can fill up disks unless you regularly move or delete old archives.

# Restoring FishEye Data

## The FishEye Data Command Line Restoration Process

There is currently no way to restore a backup from the web 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.

1. Install FishEye into a new, empty directory (this must be the same version that the backup was created from, or later).
   Note that you cannot restore data into versions of FishEye which are older than the version that created the backup.
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 the 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

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/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/bin/` directory:

```
$ ./fisheyectl.sh restore --help
```

Restores a FishEye/Crucible backup instance.
If you are using an external database (as opposed to the default built-in database), make sure the JDBC driver file is present in the `FISHEYE_INST/lib` directory when running restore.

| Option | Switch | Description | Default |
|---|---|---|---|
| Suppress output | --quiet OR -q | Suppress the output messages from the restore program on the command line. | No |
| Choose file to restore from | --file PATH/FILENAME OR -f PATH/FILENAME | Restore the backup from PATH/FILENAME. | Yes (required) |
| Show inline help | --help OR -h | Displays help for options on the command line. | No |

## Advanced Command Line Restore Settings

By default, the restore program will restore all items found in the backup archive (so if you included the caches using the `--cache` option, these will automatically be restored). However, it is possible to only restore a subset of items from the backup, by explicitly specifying the item names on the command line and only those will be restored.

| Option | Switch | Description |
|---|---|---|
| Restore FishEye cache | --cache | Restore the repository cache backup. |
| Restore plugins | --plugins | Restore 3rd-party plugins and their configuration data. |
| Restore templates | --templates | Restore freemarker templates from the backup (the restored instance will use the built-in templates). |
| Restore uploads | --uploads | Restore uploads (e.g. patch files uploaded into Crucible and contents of files under review). This item only applies when using Crucible with FishEye. |
| Restore Crucible reviews | --sql | Restore the SQL database containing user profiles, reviews and review comments. |
| Set database type | --dbtype OR -t | SQL database type ('mysql', 'postgresql', 'oracle', 'sqlserver2005', 'sqlserver2008', or 'built-in'). Only required when restoring to a database location different to that used at used at backup time. |
| Set JDBC URL | --jdbcurl OR -j | JDBC URL of the SQL database. Only required when restoring to a database location different to that used at used at backup time (not applicable for 'built-in'). |
| Set JDBC username | --username OR -u | JDBC username of the SQL database. Only required when restoring to a database location different to that used at used at backup time (not applicable for 'built-in'). |
| JDBC password | --password OR -p | JDBC password of the SQL database. Only required when restoring to a database location different to that used at used at backup time (not applicable for 'built-in'). |
| JDBC class | --driver OR -d | Specifies the JDBC driver class name needed to access the SQL database. Only required when restoring to a database location different to that used at used at backup time and when using a different JDBC driver than the standard driver associated with the database specified though `--dbtype`. (Not applicable for 'built-in'.) |

### Notes on Migrating Backup Data

When the process restores a SQL database, it looks at the configuration data (`config.xml`) included in the backup archive to learn which database product was used and how to connect to it. When 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 Enterprise Server), use the command line options to point the process to the new database.

### Command Line Example: Migrating Backup Data to MySQL Enterprise Server

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/bin/` directory.

**Restoring to a FishEye instance that uses a different database (ensure the MySQL Enterprise Server 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
```

## Enabling Access Logging in FishEye

Stop Fisheye/Crucible 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="org.mortbay.jetty.handler.RequestLogHandler">
   <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 and that 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 you `FISHEYE_INST` directory, change the `default="./var/log"` to the path to the log folder in `FISHEYE_INST`.

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&outputtype=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&outputtype=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&outputtype=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&outputtype=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&outputtype=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&outputtype=image HTTP/1.1" 200
128
```

Please refer to the Jetty documentation for more configuration options.

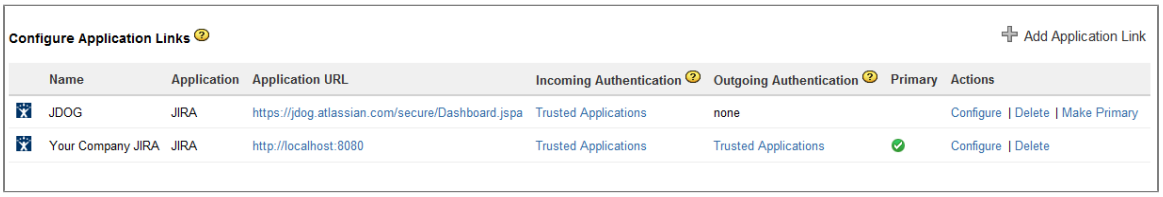
## Integrating FishEye with Other Applications

- Configuring Application Links
- JIRA Integration in FishEye
- Trusted Applications

# 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.




*Screenshot above: Application links to two JIRA servers*

***Related Topics***

- Adding an Application Link
- Configuring Authentication for an Application Link
- Configuring Project Links across Applications
- Deleting an Application Link
- Editing an Application Link
- Making an Application Link the Primary Link
- Relocating an Application Link
- Upgrading an Application Link

## 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*

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 applicaiton, 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.


Step 1
Step 2
Step 3

*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).

Step 1
Step 2

*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*

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?

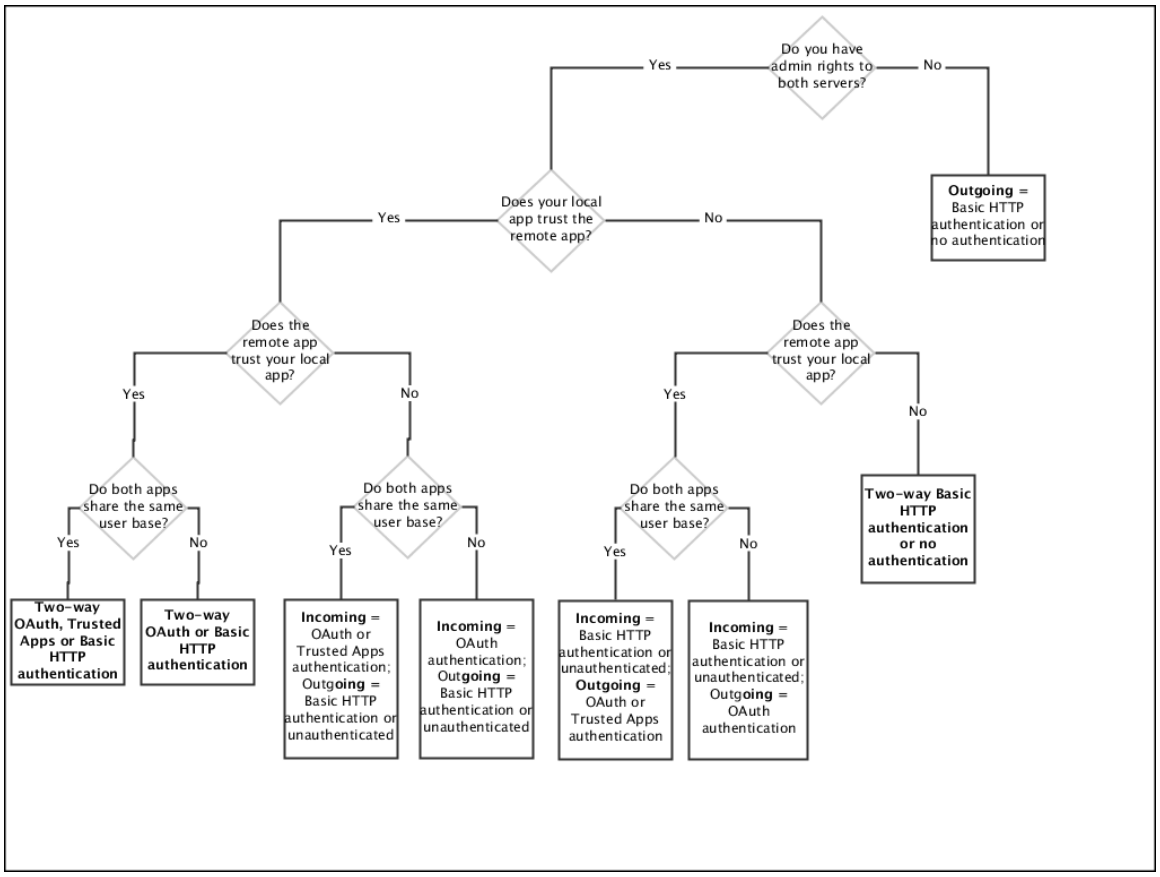
**Common scenarios include:**

- 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
- Incoming and Outgoing Authentication




*Flowchart above: Determining what authentication to configure for an Application Link*

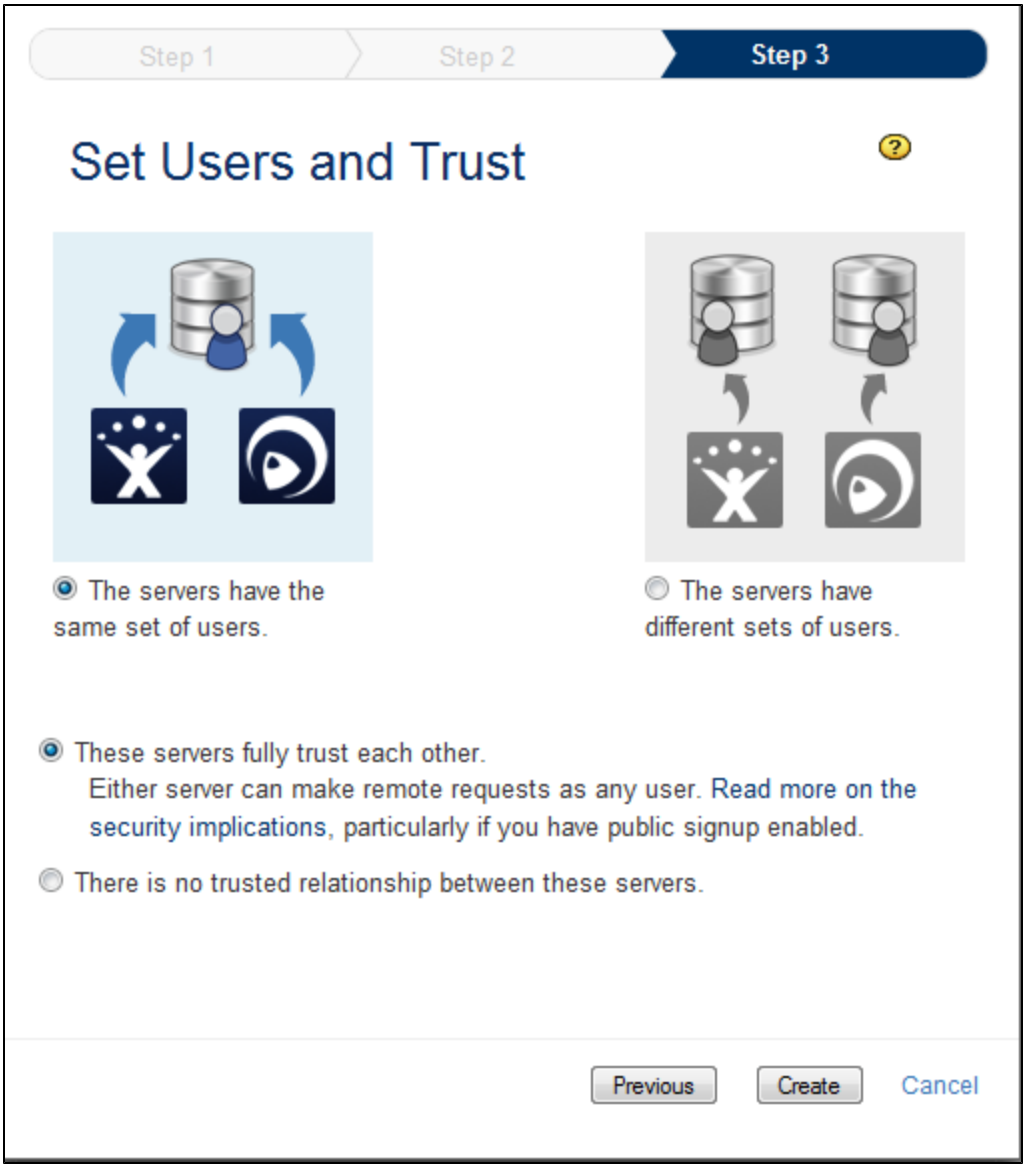
### *Security Implications for each Authentication Type*

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.




*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.

### Configuring Basic HTTP Authentication for an Application Link

Tthis 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

- 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

- 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 Confluence Gadgets for Use in Other Applications]

# 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

- 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 imconing 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**' 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

### 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).

See Configuring Authentication for an Application Link.

## Configuring Project Links across Applications

Let's assume that you are managing a project or team. You would like to connect your project's Confluence space with your JIRA project, and link up your team's source repository too.

When you have connected your applications via Application Links, you can also connect the areas of those applications that contain information relating to your project or team. Using **project links** (also called **entity links**) you can associate one or more projects, spaces and repositories across the linked applications.

To connect all the information relating to the project or team that you are managing, you can link one or more of the following:

- JIRA projects.
- Confluence spaces.
- FishEye repositories.
- FishEye projects. A FishEye 'project' is the Crucible project if you have installed FishEye and Crucible, otherwise it is the paths associated via the 'FishEye Project Content' function in FishEye.
- Crucible projects.
- Bamboo projects.

**On this page:**



### *Uses for Project Links*

The following integration features use project links:

- Activity streams. For example, the project links determine the activity retrieved from JIRA to display in the activity stream of a FishEye repository or a Crucible project.
- The JIRA FishEye plugin. For example:
  - The link between a JIRA project and a FishEye repository determines the repository searched for a particular issue key when displaying the FishEye source tab in JIRA.
  - The link between a JIRA project and a Crucible project determines the Crucible project scanned for review activity when displaying the Crucible reviews tab in JIRA.
  - When you create a defect in Crucible, Crucible will know which JIRA project to put it in.
- Third-party plugins may make use of project links to enrich their functionality too.

### *Managing Project Links*

- Adding Project Links between Applications
- Deleting a Project Link
- Making a Project Link the Primary Link

RELATED TOPICS

Adding an Application Link

## Adding Project Links between Applications

Let's assume that you are managing a project or team. You would like to connect your project's Confluence space with your JIRA project, and link up your team's source repository too.

When you have connected your applications via Application Links, you can also connect the areas of those applications that contain information relating to your project or team. Using **project links** (also called **entity links**) you can associate one or more projects, spaces and repositories across the linked applications.

To connect all the information relating to the project or team that you are managing, you can link one or more of the following:

- JIRA projects.
- Confluence spaces.
- FishEye repositories.
- FishEye projects. A FishEye 'project' is the Crucible project if you have installed FishEye and Crucible, otherwise it is the paths associated via the 'FishEye Project Content' function in FishEye.
- Crucible projects.
- Bamboo projects.

**On this page:**



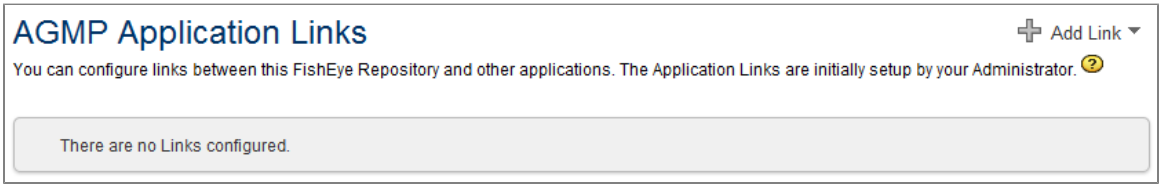
#### Adding a Project Link

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.

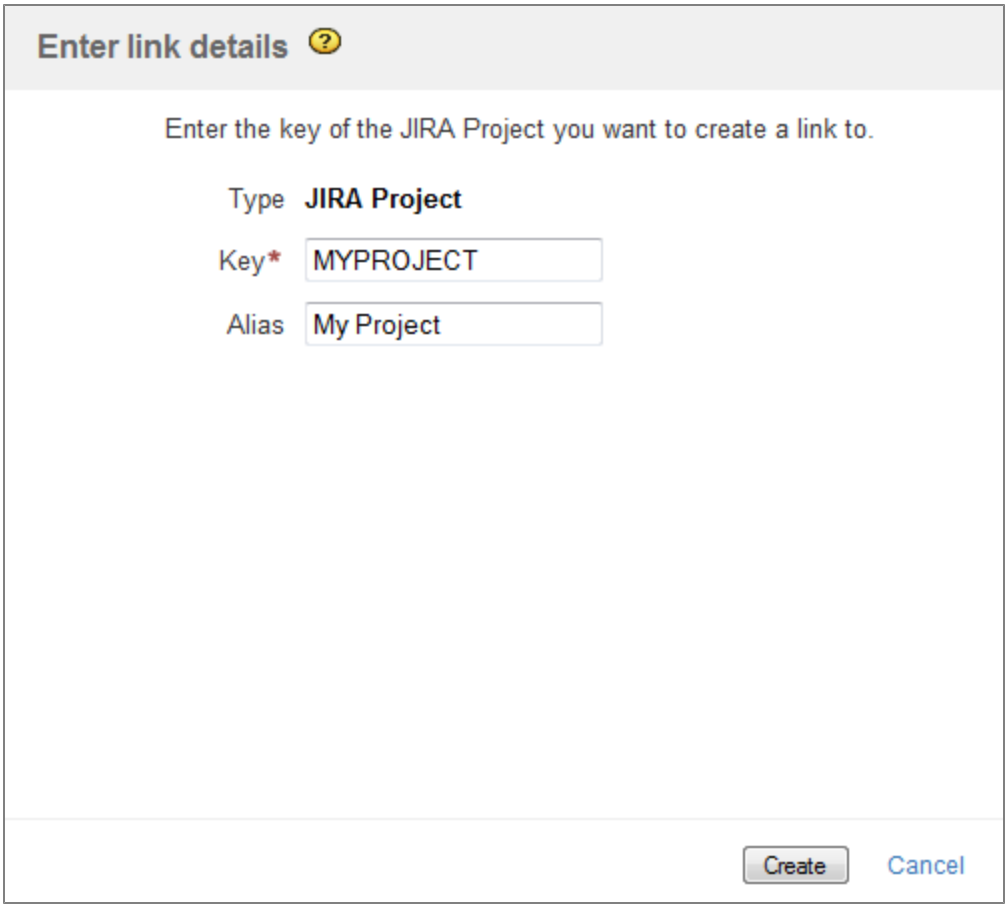
**To link a FishEye project to a project in another application:**

1. Go to the FishEye administration console and find the repository that you want to link from.

2. Click the icon and select '**Application Links**' from the dropdown menu. The project links screen will appear.
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.




*Screenshot above : Project links for a FishEye repository*




*Screenshot above: Adding a project link*

RELATED TOPICS

Making a Project Link the Primary Link
Deleting a Project Link

### Deleting a Project Link

Deleting a project link stops the two projects from sharing information.

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 select '**Application Links**' from the dropdown menu. The project links screen will appear.
3. Click '**Delete**' next to the link that you want to delete.
4. A confirmation screen will appear. Click the '**Confirm**' button to delete the link.

Related Topics

Adding Project Links between Applications
Making a Project Link the Primary 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 select '**Application Links**' from the dropdown menu. The project links screen will appear.
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.

RELATED TOPICS

Adding Project Links between Applications
Deleting a Project 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

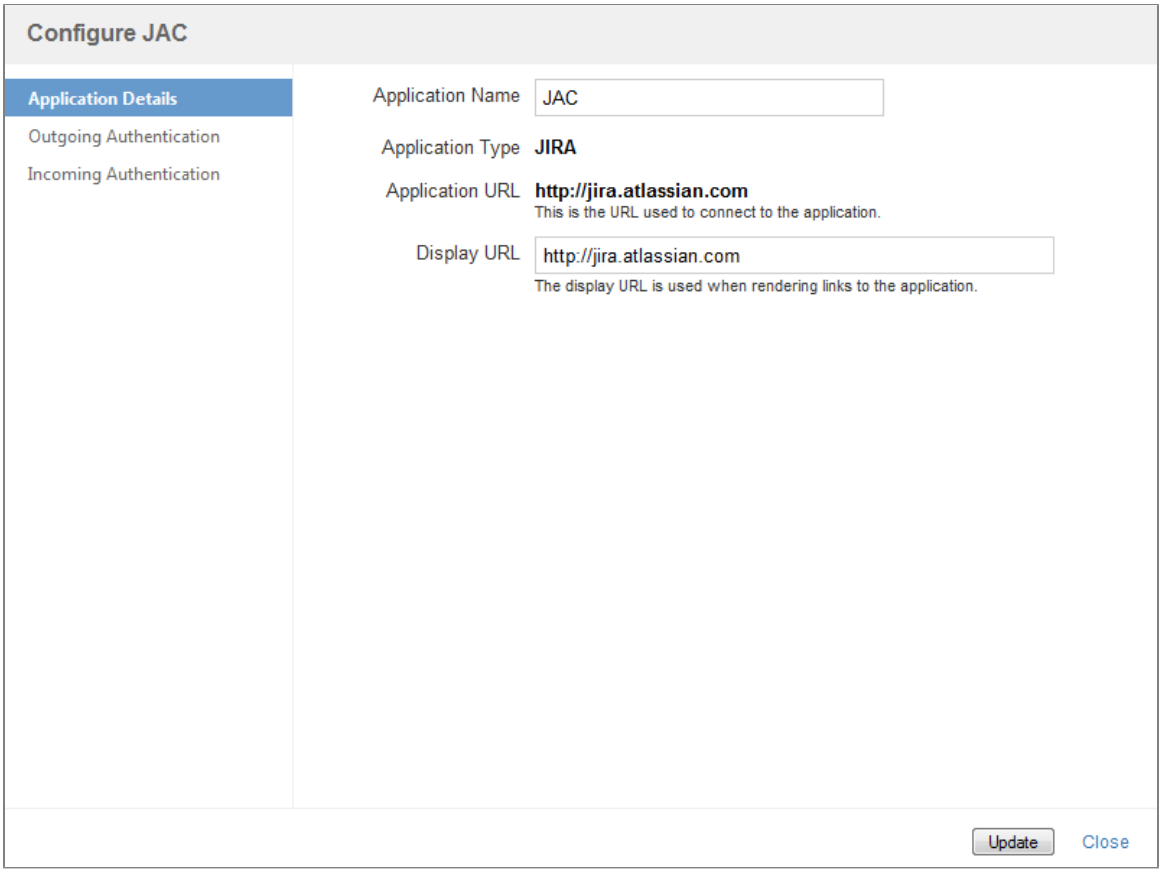
Editing an Application Link
Relocating an Application Link

## 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.




*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.

**On this page:**



### *Making an Application Link the Primary Link*

**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.*

### *Notes*

Please read Making a Project Link the Primary Link for information on how primary project links also influence the information shared between servers.

Related Topics

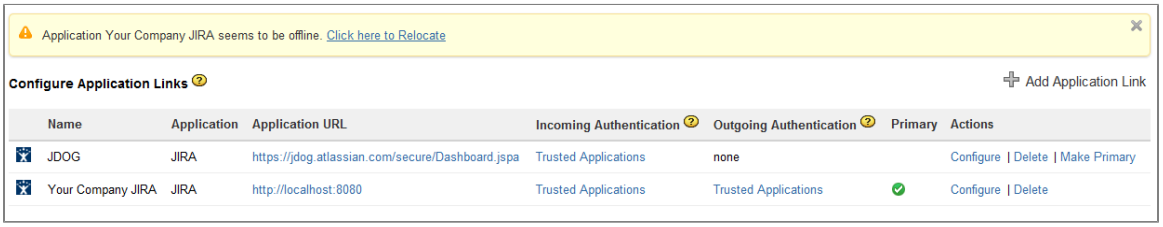
Making a Project Link the Primary Link

## Relocating an Application Link

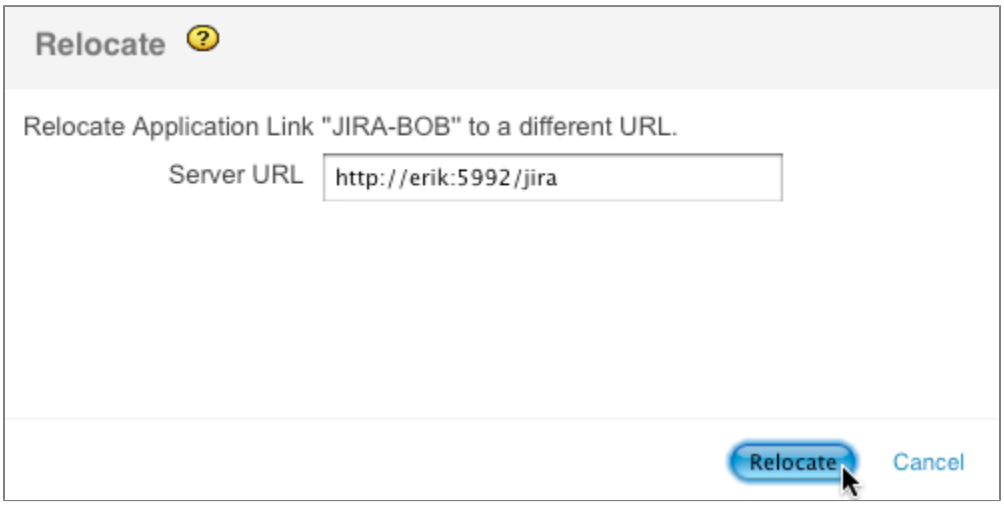
This page describes how to change the location of an application link. You will need to relocate an application link if the target application has moved to a new address.

**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.




*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:**

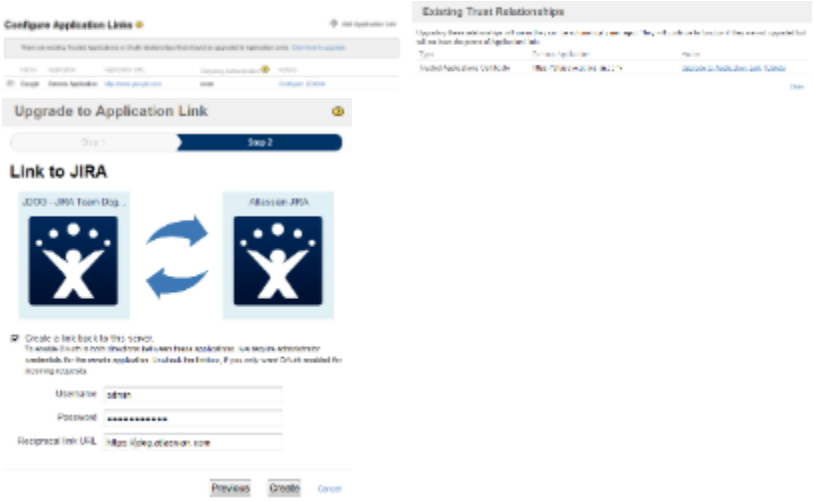


### *Upgrading an Application Link (Local App Upgraded to Include Application Links)*

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
Step 3
Step 4

*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:

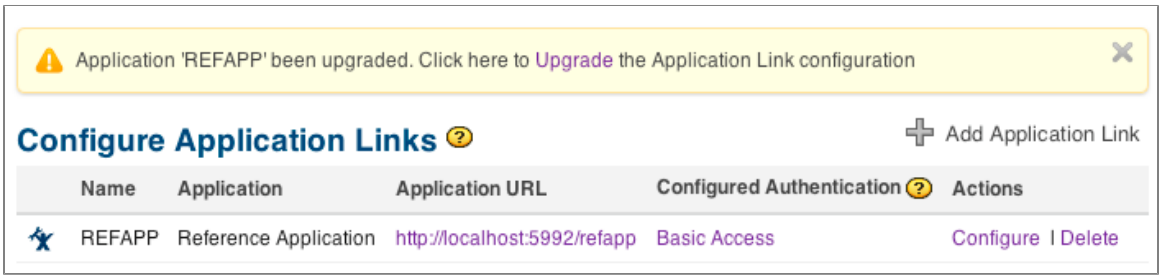
- Setting up OAuth requires manual configuration by the administrator. If your applications support Application Links, exchange of the

consumer keys and public keys is done automatically.
- The Trusted Applications protocol (Atlassian-specific) will not be available for authentication.

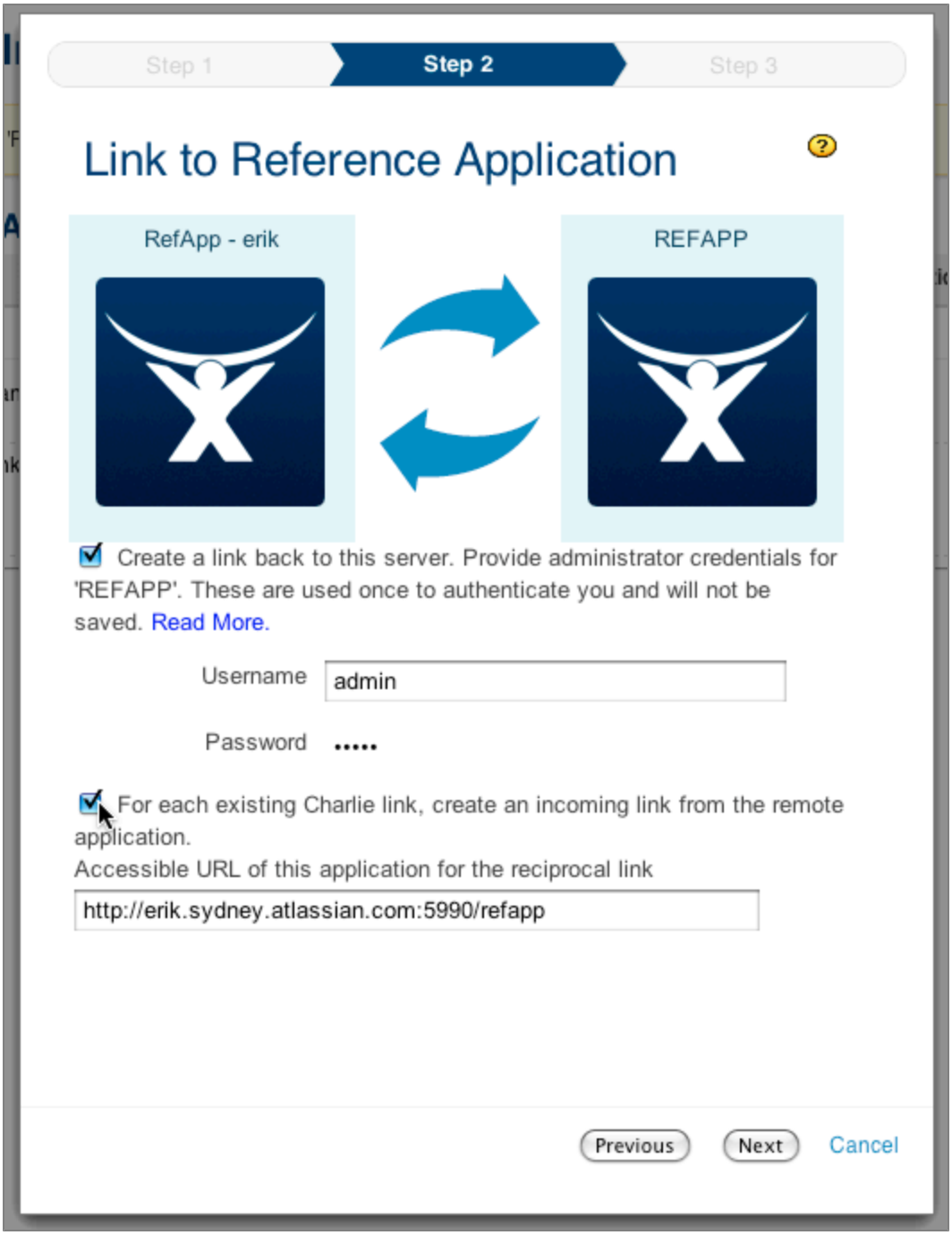
If you upgrade your remote application to a version that does support Application Links, the application link will continue to work. Upgrading your link may simplify link configuration and make additional authentication protocols available (as mentioned above).

**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*




*Screenshot above: Upgrading an application link wizard*

Related Topics

Adding an Application Link
Configuring Authentication for an Application Link

## JIRA Integration in FishEye

JIRA is Atlassian's issue tracking and project management application. This page tells you how to integrate JIRA with FishEye.

### Initial Configuration in JIRA

Configure the following setting in JIRA:

- Allow remote API access.

Consider also the following settings, to make full of the integration between FishEye and JIRA:

- Enable sub-tasks.
- Allow unassigned issues.

**On this page:**



## Quick Setup via the FishEye Setup Wizard

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. 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 via the FishEye Administration Screens

You can further configure the JIRA connections via the FishEye administration screens. 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.

### *1. Configuring the Application Links with JIRA*

Below are step by step instructions to 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:**

1. Go to your FishEye administration screen and add the application link:
   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.
2. If you want to show issues in your FishEye activity streams:
   - Go to the list of application links and click '**JIRA settings**' next to the JIRA application link.
   - Tick the '**Include in Activity Streams**' check box.
   - Click '**Save**'.
3. Connect your projects across the applications:
   - Connect your FishEye repositories with your JIRA projects. See the FishEye guide to adding project links.
   - If you are using Crucible, you may want to connect your Crucible projects with your JIRA projects. See the Crucible guide to adding project links.

### *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 via 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 the '**Administration**' link on the JIRA top navigation bar.
    a. Select '**Other Applications**' from the '**Users, Groups & Roles**' section of the JIRA administration menu.
    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**' on the FishEye/Crucible '**Admin Menu**'.
    b. Click '**Edit**' in the section titled '**JIRA or Crowd Authentication**'.
    c. Enter the following information:
        - **Application name** and **Application password** – Enter the values that you defined for your application in the settings on JIRA.
        - **JIRA/Crowd URL** – The web address of your JIRA server. Examples:

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

        - **Auto-add** – Select '**Create a FishEye/Crucible user on successful login**' (default) to ensure that your JIRA users will be automatically enrolled into FishEye/Crucible when they first log in via JIRA.
        - **Synchronize users with JIRA/Crowd** – Select '**Yes**' (default) to ensure that JIRA will synchronize all changes in the user information on a regular basis. The **synchronization interval** is set to 60 minutes (1 hour) by default.
        - **Single sign on (SSO)** – This option is not available when using JIRA for user management.
        - **Groups of users to synchronize** – Select at least one group to synchronize. The default is '**jira-users**'.
    d. Click '**Apply changes**'.

***Related Topics***

User Management Limitations and Recommendations
Configuring Application Links

## 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

# FishEye Installation and Upgrade Guide

- FishEye Release Notes
- FishEye Installation Guide
- FishEye Upgrade Guide
- Supported Platforms

## FishEye Release Notes

FishEye 2.5 has now been released. Read the Release Notes.

### FishEye Release Notes and Changelogs



### Installation

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.

## 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

DOWNLOAD latest version

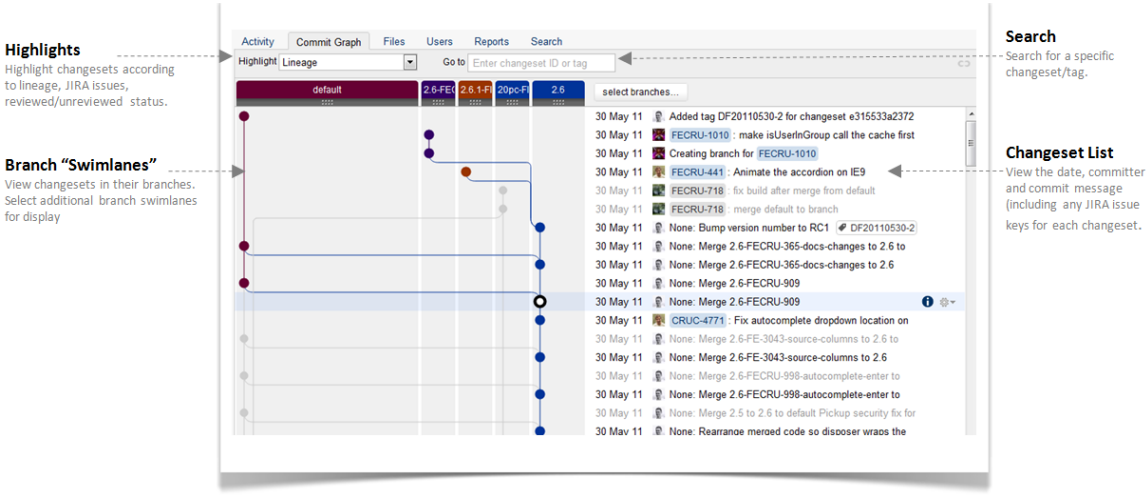
- Thank you for all your issues and votes. Keep logging issues to help us keep improving!
- Read the release notices for important information about this release.

## Highlights of FishEye 2.6

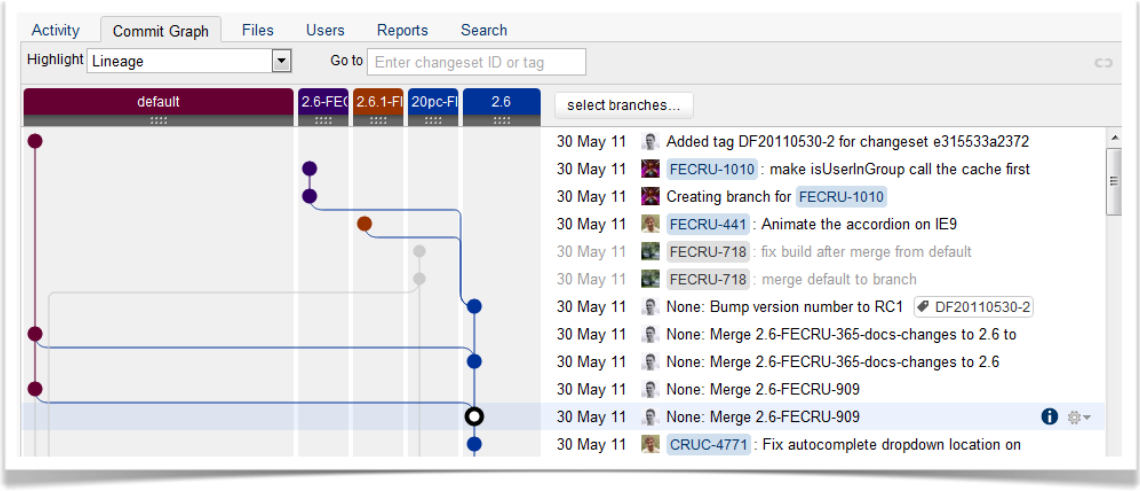
1

### 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.



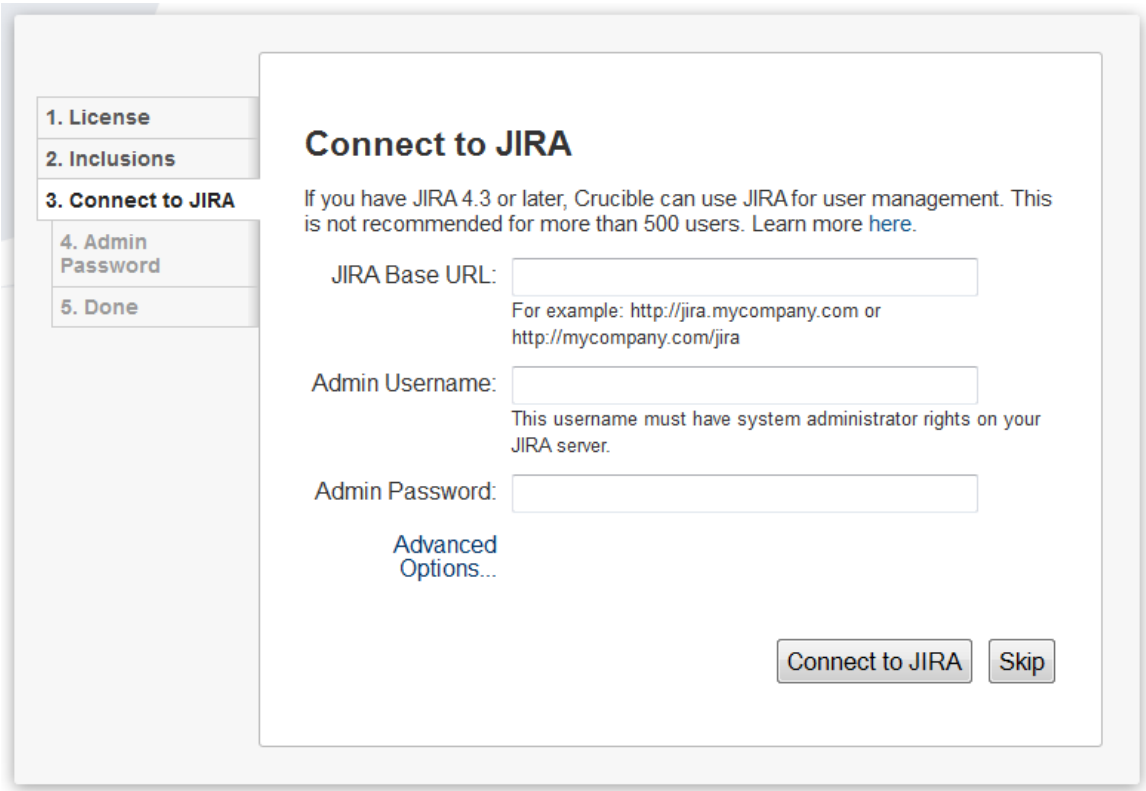

Commit graph highlights provide further context, allowing you to highlight changesets in the same branch, commits with JIRA issues, and reviewed/unreviewed changesets. Clicking a changeset with the appropriate highlight active (selected in the 'Highlight' dropdown) shows you related changesets, such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




More...

### 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.

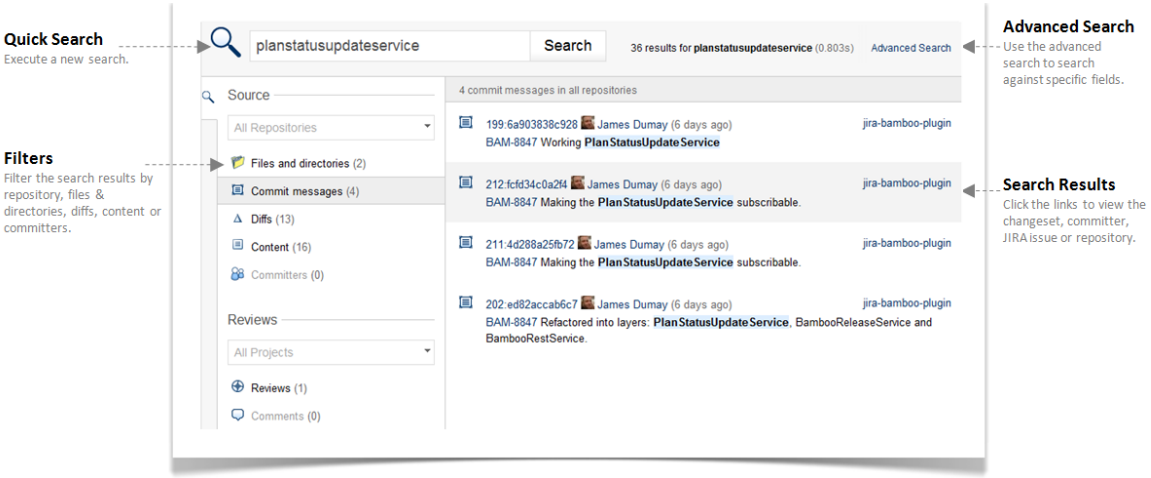



More...

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.

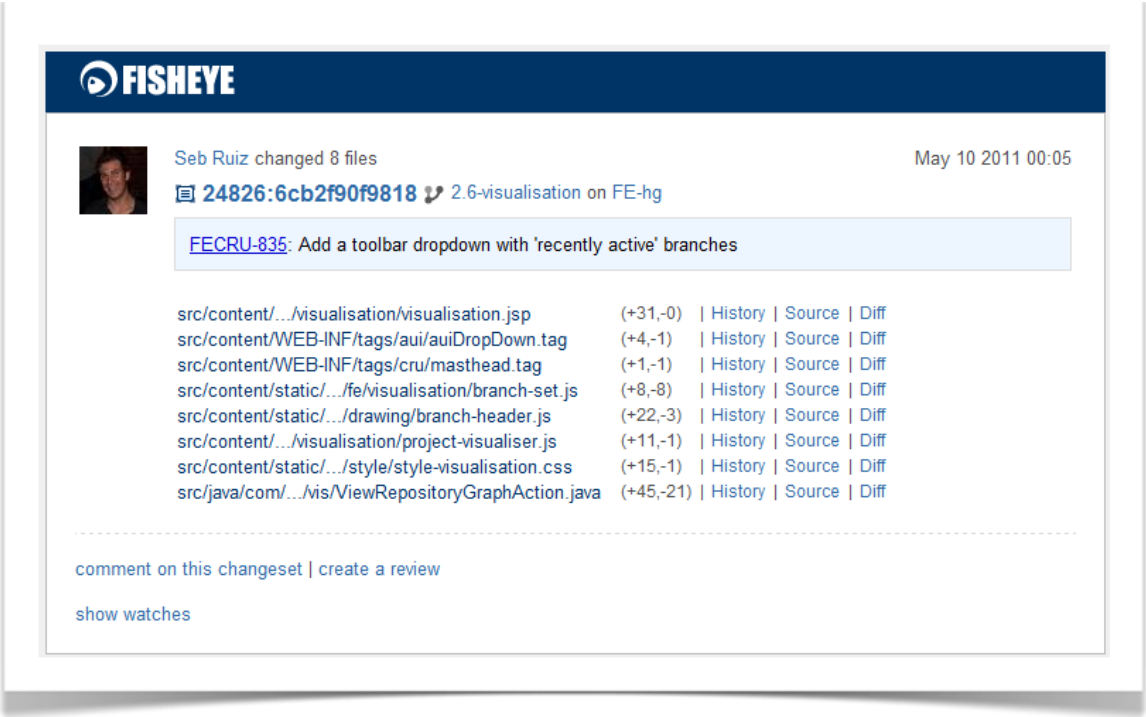



More...

4

### Redesigned HTML Emails

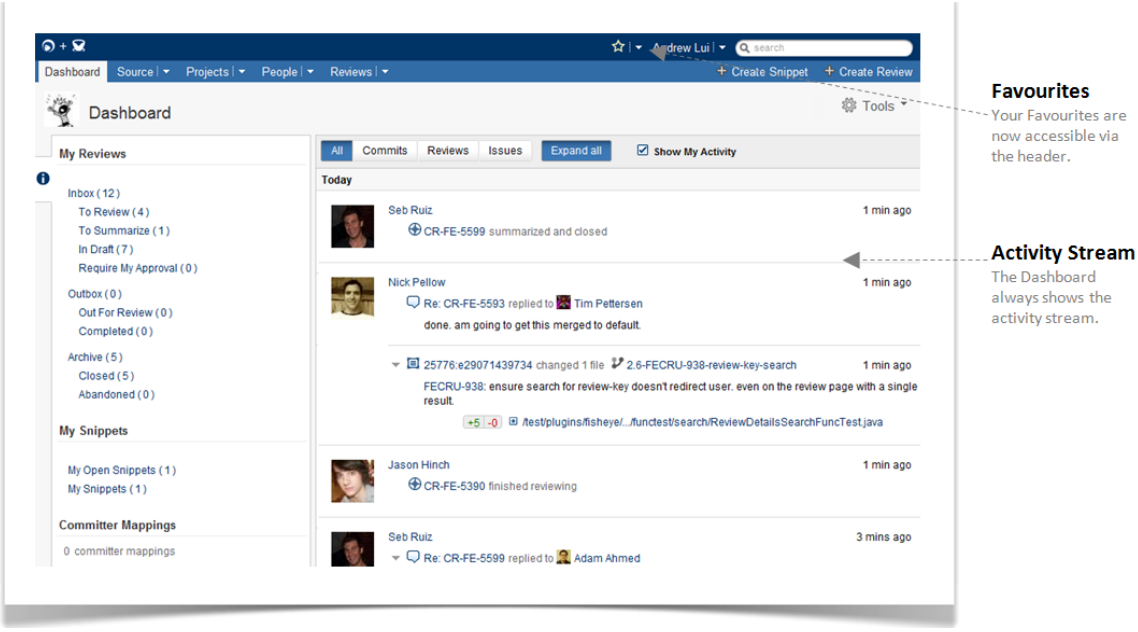
A 20% time project has resulted in email notifications getting a dramatic facelift. Gone are the dreary old HTML emails, replaced by much better-looking ones. You'll see the new emails in action for changeset notifications.




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.



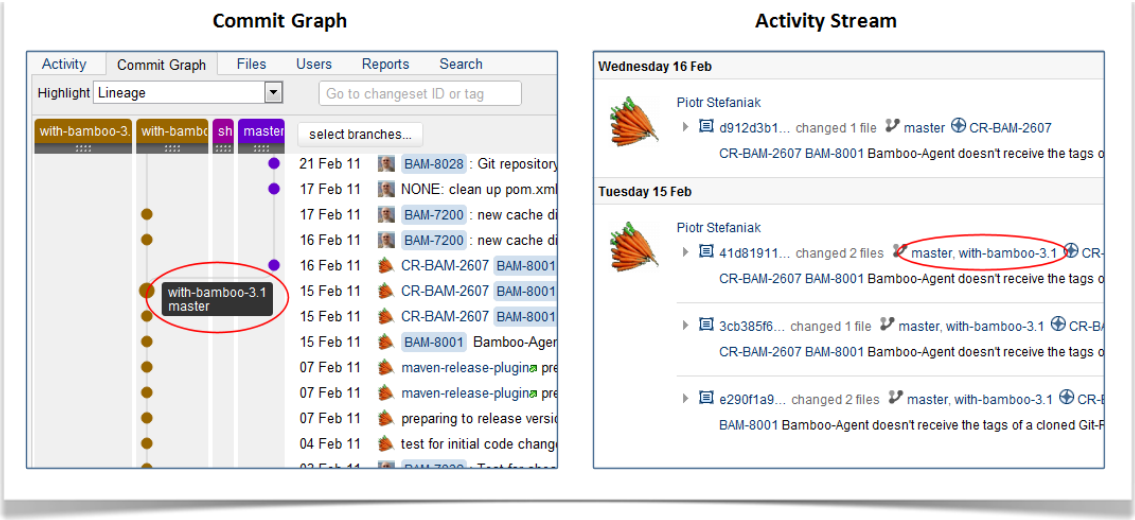

More...

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 2.6.0 to 2.6.1

#### 22 June 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-3546 | Git - remove branch fails | Anna Buttfield [Atlassian] | Adam Ahmed [Atlassian] | | Closed | Fixed | Jun 29, 2011 | Jun 30, 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 on every page for IE 7 saying that the browser isn't supported. Would be incredibly annoying | Jason Hinch [Atlassian] | None | | Closed | Fixed | Jun 01, 2011 | Jun 14, 2011 |
| | 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 | Jun 18, 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 save the new config | Tom Davies [Atlassian] | Tom Davies [Atlassian] | | Closed | Fixed | May 26, 2011 | Jun 19, 2011 |

# 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*

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

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

- **Set aside additional time for the upgrade** — When you upgrade to FishEye 2.6, FishEye will run a number of upgrade tasks and trigger an automatic upgrade of the metadata index. While this is happening, FishEye (and Crucible) can be be accessed, however no new changes in the repositories will be detected until the upgrade is complete. Hence, you should set aside time for these activities to complete. The time taken will depend on the complexity of your code base, as well as the type of repository that you are using. See the following repository-specific notes:

| | |
|---|---|
| **Subversion** | The upgrade tasks for Subversion repositories will take significantly longer in this release, than in previous releases. In previous releases (e.g. FishEye 2.5), the upgrade tasks only required a short amount of time to complete, relative to the time taken to upgrade the metadata index. In this release, the upgrade tasks and metadata index upgrade take approximately equal amounts of time. Hence, the overall upgrade may take up to *twice as long*, compared to prior releases. Our Subversion repository for JIRA development, which is quite complex, took about three hours to complete both the upgrade tasks and metadata index upgrade. |
| **CVS, Perforce, ClearCase** | As per Subversion, the upgrade tasks for CVS and Perforce will take proportionally longer to complete in this release, hence the overall upgrade will take longer. However, the tasks are not as complex as the Subversion upgrade tasks, so you shouldn't notice a dramatic increase in time taken. The upgrade task for ClearCase is not complex at all, so you shouldn't notice a significant increase in time taken for the overall upgrade. |
| **Git, Mercurial** | The upgrade tasks for Git & Mercurial repositories will not take long to complete, perhaps a few minutes. You will still need to set aside time for the metadata index upgrade though. |

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

## *Upgrade Procedure*

**Before you begin**

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

If you are already running a version of FishEye, please follow these instructions on FishEye Upgrade Guide.

### *Checking for Known Issues and Troubleshooting the FishEye Upgrade*

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

- **Check for known issues**. Sometimes we find out about a problem with the latest version of FishEye after we have released the software. In such cases we publish information about the known issues in the FishEye Knowledge Base. Please check the FishEye 2.6 Known Issues in the FishEye Knowledge Base and follow the instructions to apply any necessary patches if necessary.
- **Did you encounter a problem during the FishEye upgrade?** Please refer to the guide to troubleshooting upgrades in the FishEye Knowledge Base.
- If you encounter a problem during the upgrade and cannot solve it, please create a support ticket and one of our support engineers will help you.

RELATED TOPICS

FishEye 2.6 Release Notes

## FishEye 2.5 Release Notes

**8 February 2011**

With great pleasure, Atlassian presents **FishEye 2.5**, providing vastly improved source code searching and DVCS authentication.

**Highlights of this Release:**

- Search Revamp
- Redesigned Activity Stream
- Mercurial and Git Authentication
- RSS Improvements
- Improved Header
- And Even More Improvements

**Responding to your Feedback:**

Over **150** votes satisfied

DOWNLOAD latest version

- Thank you for all your issues and votes. Keep logging issues to help us keep improving!
- Read the release notices for important information about this release.

## Highlights of FishEye 2.5

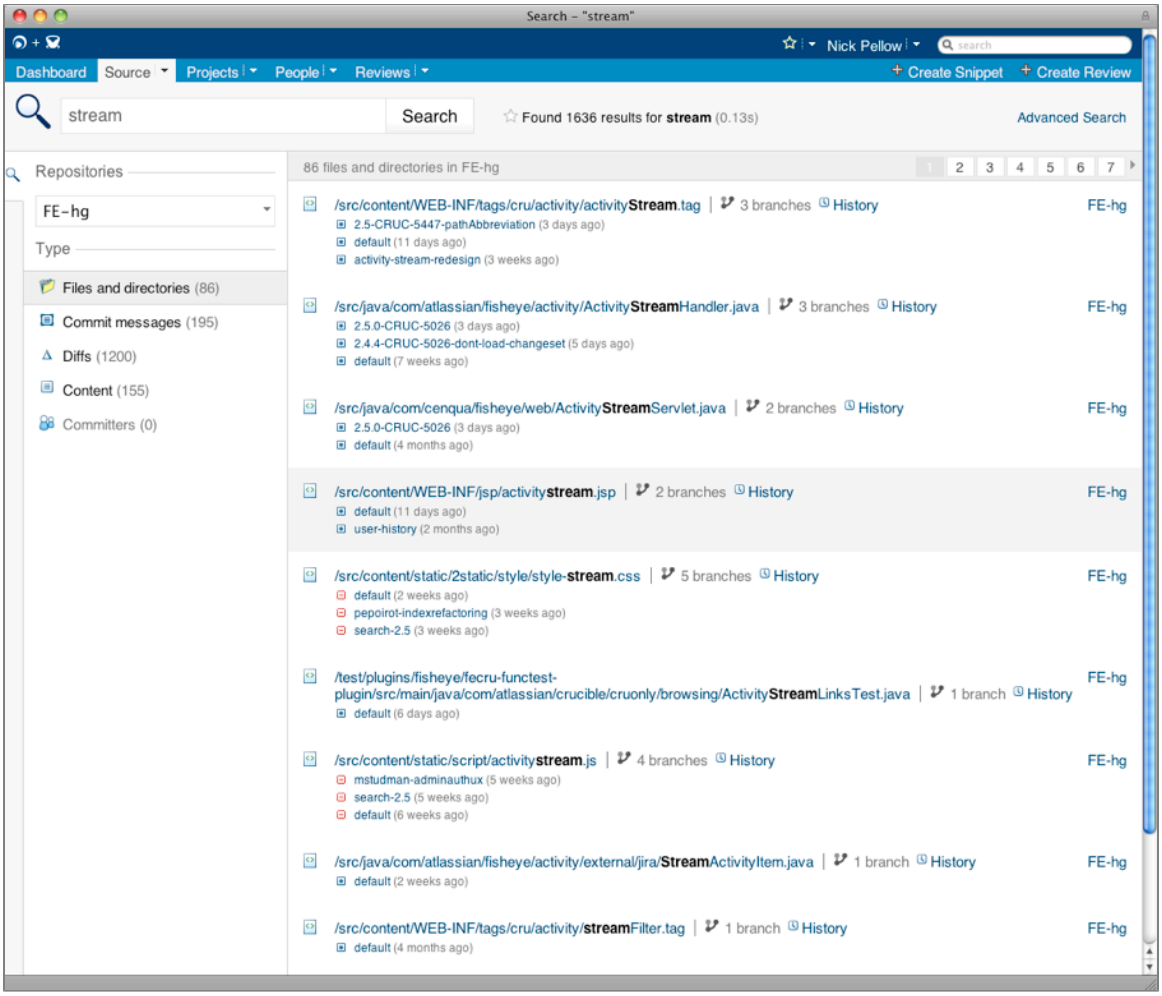
1

### 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...



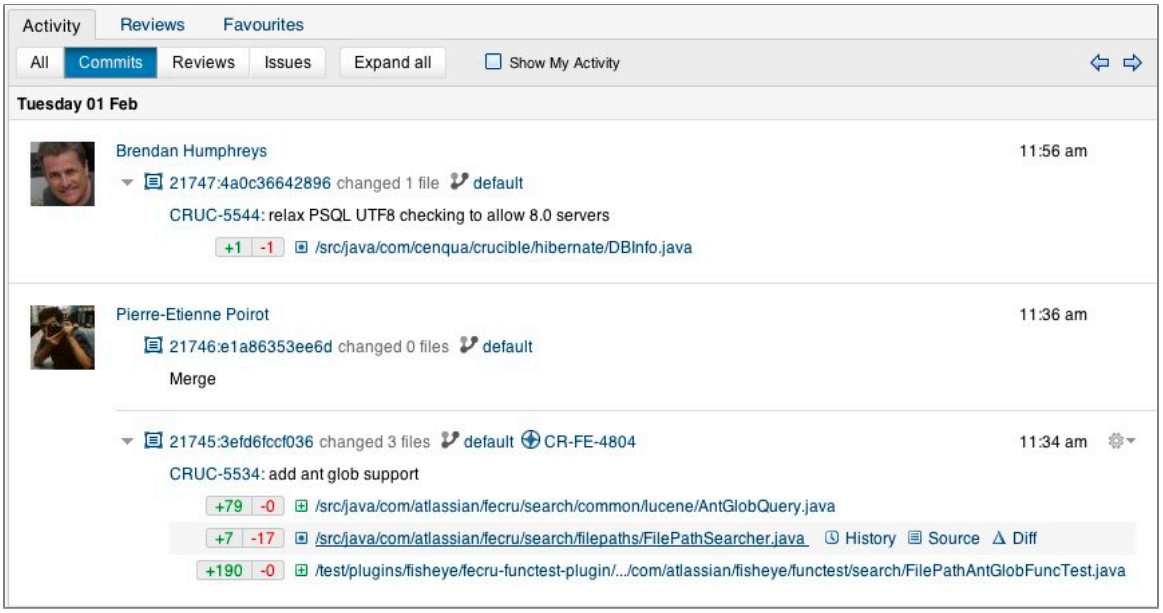

2

### 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...

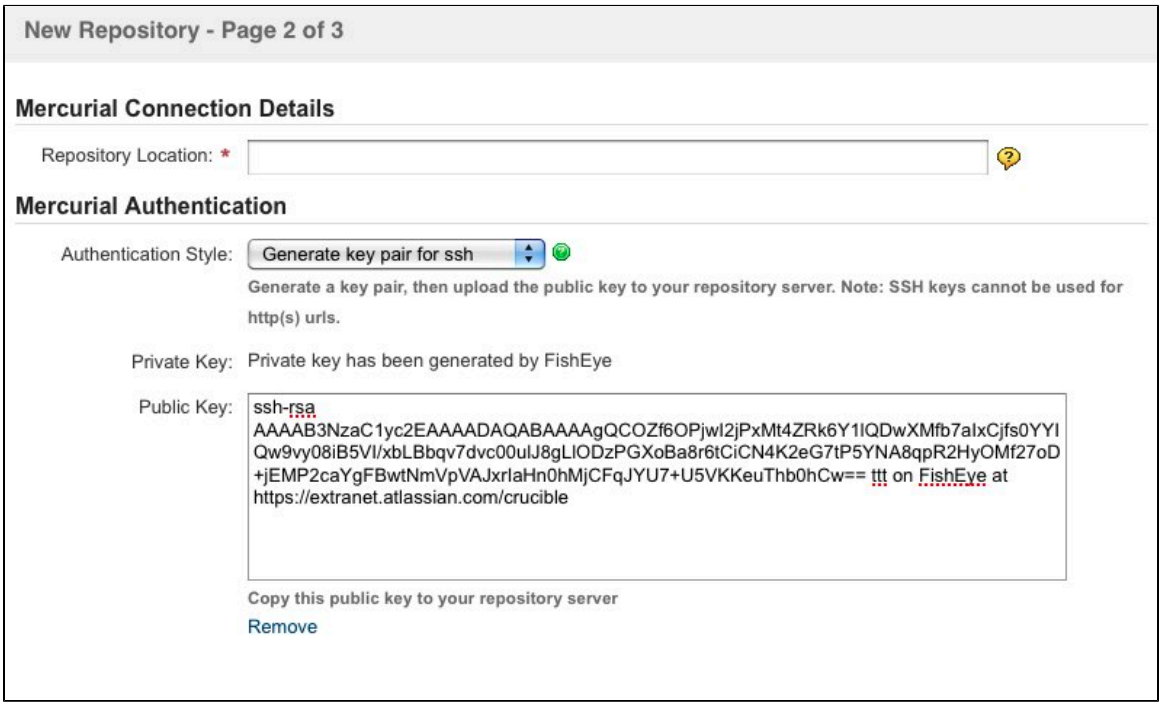



3

### 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.

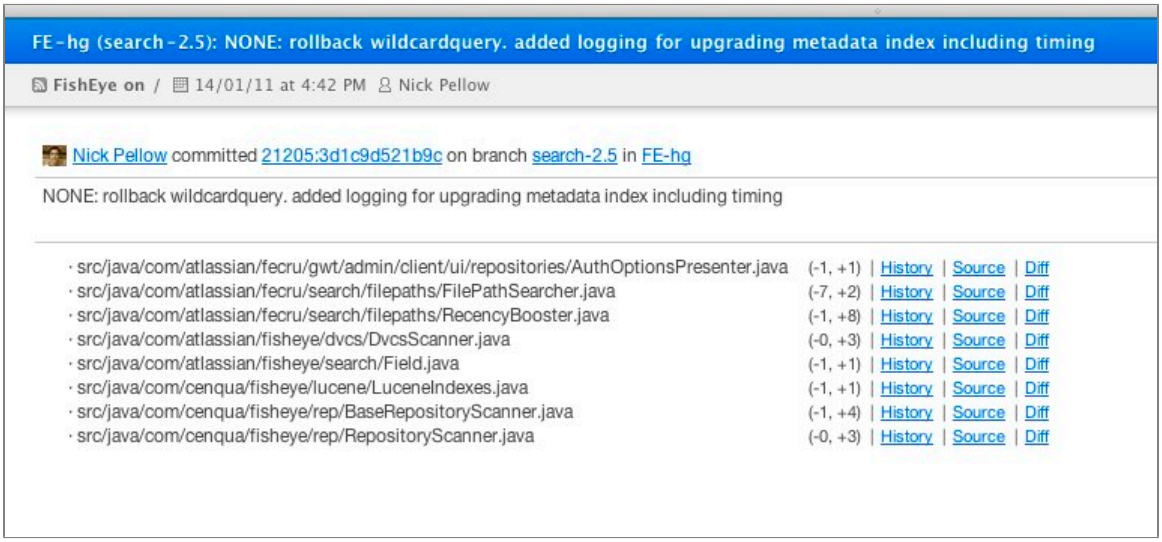
Read more about Authentication...



4

### RSS Improvements

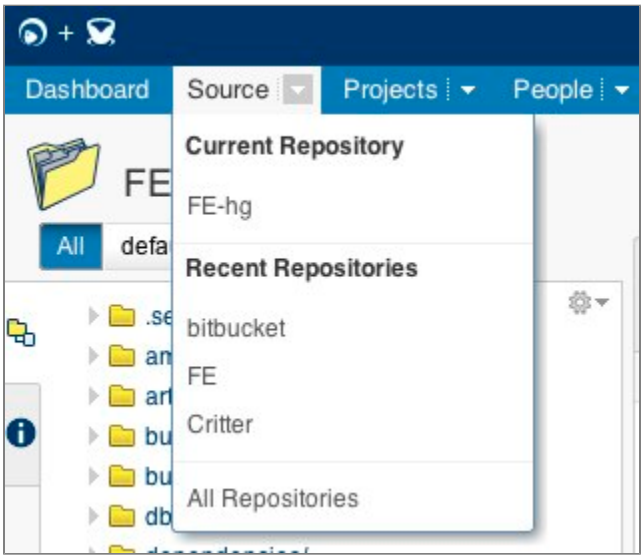
FishEye 2.5 introduces some fine tuning for rendered RSS feeds. RSS content is easier to digest (better titles) and allows you to consume FishEye activity through your favourite RSS reader. We render wiki-markup in RSS which will allow you to click through links and view images inline.




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

**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' button on Database Configuration, Edit Admin page doesn't save the new config | Tom Davies [Atlassian] | Tom Davies [Atlassian] | | Closed | Fixed | May 26, 2011 | Jun 19, 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 droping 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 element in lsactivity | Conor MacNeill [Atlassian] | Conor MacNeill [Atlassian] | | Closed | Fixed | May 17, 2011 | May 17, 2011 |
| | 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 Heemskerk [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 (6 issues)**

| Type | Key | Summary | Assignee | Reporter | Priority | Status | Resolution | Created | Updated |
|---|---|---|---|---|---|---|---|---|---|
| | 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 renderer breaks the source layout | Tom Davies [Atlassian] | Julien Hoarau | | Closed | Fixed | Apr 18, 2011 | May 09, 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 FF 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 arent storing the latest possible activity time, so its possible to get a lot of unnecessary data | Geoff Crain [Atlassian] | Geoff Crain [Atlassian] | | Closed | Fixed | Mar 28, 2011 | Mar 29, 2011 |
| | 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 (11 issues)**

| Type | Key | Summary | Assignee | Reporter | Priority | Status | Resolution | Created | Updated |
|---|---|---|---|---|---|---|---|---|---|
| | 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 | May 09, 2011 |
| | FE-3075 | NullPointerException in GroupEditAction | Pierre-Etienne Poirot [Atlassian] | None | | Closed | Fixed | Mar 20, 2011 | May 09, 2011 |
| | FE-3052 | Recent visited tabs spins forever when REST end points are disabled | Pierre-Etienne Poirot [Atlassian] | None | | Closed | Fixed | Mar 15, 2011 | May 09, 2011 |
| | FE-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 Heemskerk [Atlassian] | | Closed | Fixed | Mar 21, 2011 | Mar 24, 2011 |
| | FE-2985 | Typo: Sceduled Backups | Tim Pettersen [Atlassian] | Matt Doar (Consulting Toolsmiths) | | Closed | Fixed | Mar 18, 2011 | Mar 20, 2011 |
| | FE-2981 | HG indexing fails with "not found in manifest!" error | Matthew Watson [Atlassian] | Matthew Watson [Atlassian] | | Closed | Fixed | Mar 15, 2011 | Mar 23, 2011 |
| | FE-2974 | UCM ClearCase: indexing error 'No such file or directory' | Michael Heemskerk [Atlassian] | Michael Heemskerk [Atlassian] | | Closed | Fixed | Mar 08, 2011 | Mar 11, 2011 |
| | FE-2972 | Go-To-Changeset doesn't work | Matthew Watson [Atlassian] | Erik van Zijst [Atlassian] | | Closed | Fixed | Mar 06, 2011 | Mar 24, 2011 |
| | FE-2966 | Deadlock in Jetty connection timeout | Joe Xie [atlassian] | Gurleen Anand [Atlassian] | | Closed | Fixed | Feb 24, 2011 | Mar 21, 2011 |
| | FE-2954 | LineDecorator.decorateLine(int) never call | Tom Davies [Atlassian] | Julien Hoarau | | 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 Heemskerk [Atlassian] | Michael Heemskerk [Atlassian] | | Closed | Fixed | Feb 06, 2011 | May 09, 2011 |
| | FE-3157 | Git isn't detecting binary files at all | Michael Heemskerk [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 |
| | | | | Michael | | | | Mar 10, | May 18, |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FE-2976 | NPE during Mercurial indexing | Unassigned | Heemskerk [Atlassian] | | Closed | Fixed | 2011 | 2011 |
| | FE-2969 | Error trying to connect to HG repo using ssh on windows | Geoff Crain [Atlassian] | Matthew Watson [Atlassian] | | Closed | Fixed | Mar 03, 2011 | Mar 04, 2011 |
| | FE-2964 | ClearCase: FishEye doesn't handle versions that are deleted from a ClearCase VOB | Michael Heemskerk [Atlassian] | Michael Heemskerk [Atlassian] | | Closed | 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 Heemskerk [Atlassian] | Michael Heemskerk [Atlassian] | | Closed | Fixed | Feb 22, 2011 | Mar 03, 2011 |
| | FE-2956 | P4 Unicode Handling | Conor MacNeill [Atlassian] | Conor MacNeill [Atlassian] | | Closed | Fixed | Feb 17, 2011 | Mar 02, 2011 |
| | FE-2948 | java.lang.IllegalStateException - No match found exception thrown when the clearcase.processBranch command times out after 1 hour | Michael Heemskerk [Atlassian] | Gurleen Anand [Atlassian] | | Closed | Fixed | Feb 09, 2011 | Mar 03, 2011 |

**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 (16 issues)**

| Type | Key | Summary | Assignee | Reporter | Priority | Status | Resolution | Created | Updated |
|---|---|---|---|---|---|---|---|---|---|
| | CRUC-5612 | Make Oracle multidb 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 View" do not work in JIRA 4.2.3 and JIRA 4.3 | Joe Xie [atlassian] | Brenden Bain [Atlassian] | | Closed | Fixed | Feb 06, 2011 | Feb 13, 2011 |
| | CRUC-5594 | Adding an AppLink to a "standalone Crucible" creates a FishEye configuration that is unusable. | Luis Miranda [Atlassian] | Brenden Bain [Atlassian] | | Closed | 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] | | Closed | 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] | | Closed | Fixed | Feb 02, 2011 | Feb 15, 2011 |
| | CRUC-5526 | FishEyeInstance gets re-created on JIRA restart | Joe Xie [atlassian] | Luis Miranda [Atlassian] | | Closed | Fixed | Jan 30, 2011 | Feb 13, 2011 |
| | CRUC-5475 | Create func tests for all search result pages | Nick Pellow [Atlassian] | Nick Pellow [Atlassian] | | Closed | 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] | | Closed | Fixed | Jan 18, 2011 | Feb 15, 2011 |
| | CRUC-5400 | On Plugin startup, synchronise App Links | Joe Xie | Mark Lassau | | | Fixed | Jan 17, | Feb 13, |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | [atlassian] | [Atlassian] | | Closed | | 2011 | 2011 |
| | CRUC-5188 | Update admin UI | Mark Lassau [Atlassian] | Joe Xie [atlassian] | | Closed | Fixed | Dec 13, 2010 | Feb 13, 2011 |
| | CRUC-5183 | Extend oauth services to REST resources (such as gadgets) | Joe Xie [atlassian] | Joe Xie [atlassian] | | Closed | 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] | | Closed | Fixed | Dec 12, 2010 | Feb 13, 2011 |
| | CRUC-5154 | Add support for OAuth authentication | Joe Xie [atlassian] | Joe Xie [atlassian] | | Closed | Fixed | Dec 12, 2010 | Feb 13, 2011 |

## 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*

- **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 via 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

DOWNLOAD latest version

- Thank you for all your issues and votes. Keep logging issues to help us keep improving!
- Read the release notices for important information about this release.

## Highlights of FishEye 2.4

1

### 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.

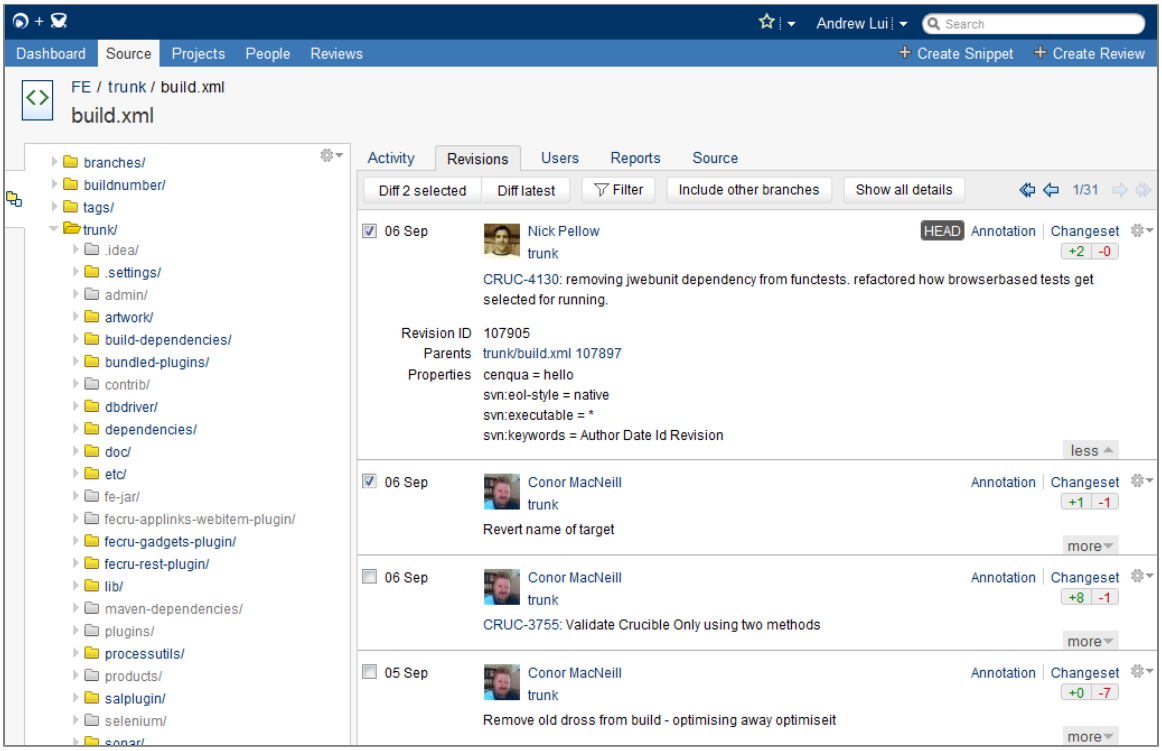
*Note, this feature currently does not work for ClearCase repositories due to technical constraints.*


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.

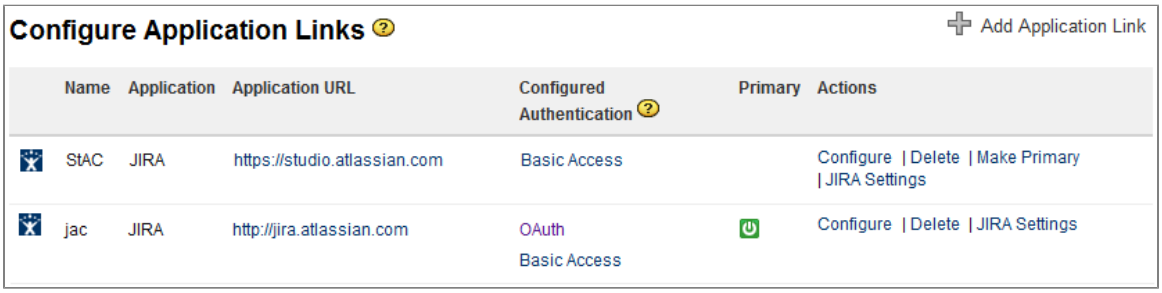



More....

## 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.

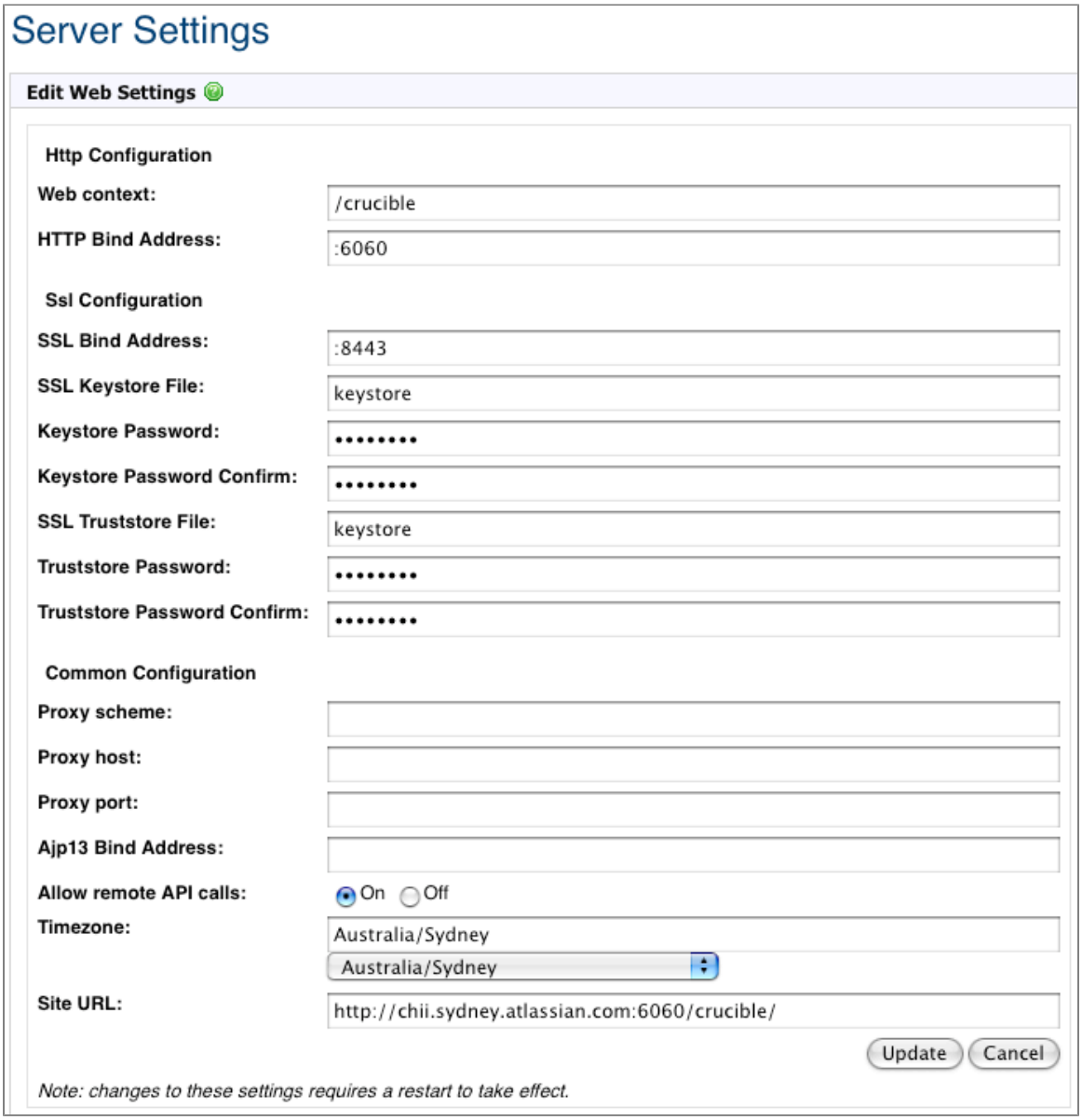


More....

4

### 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.

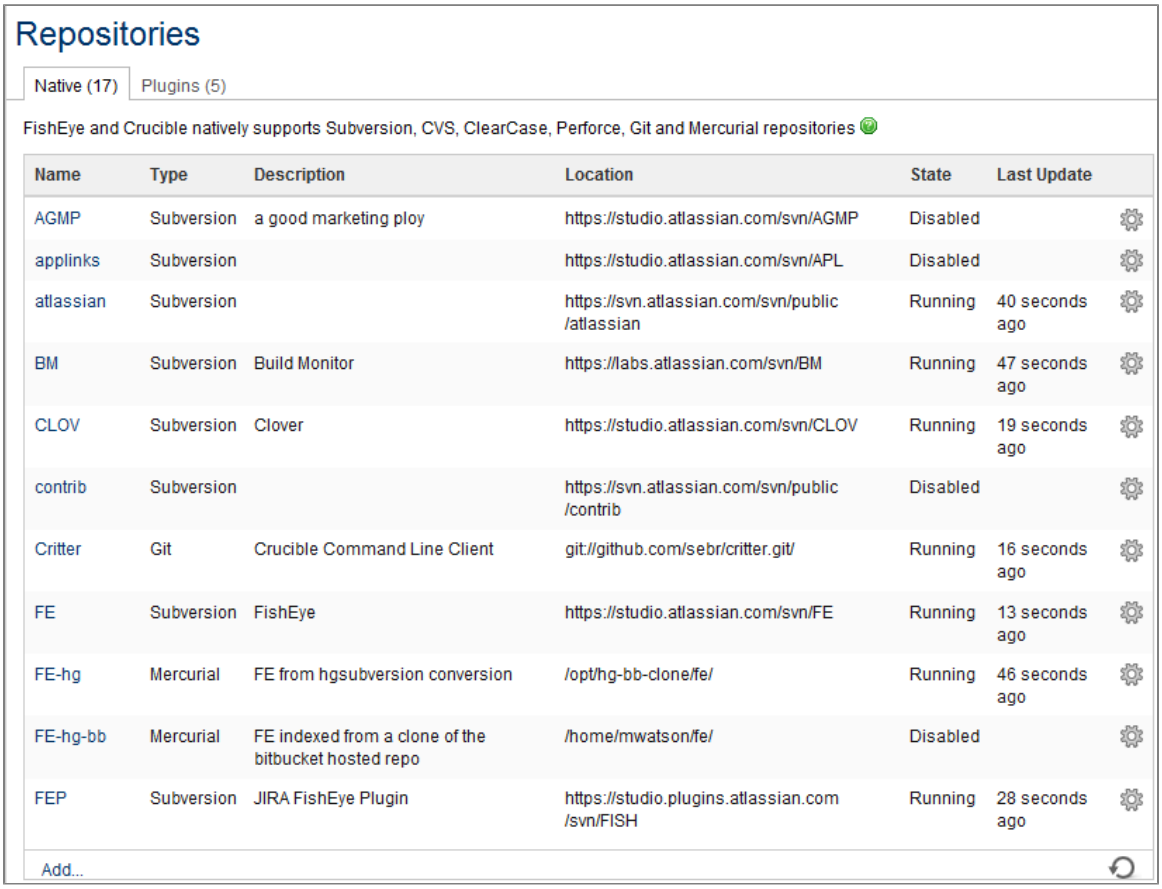



More....

5

### User Interface Improvements

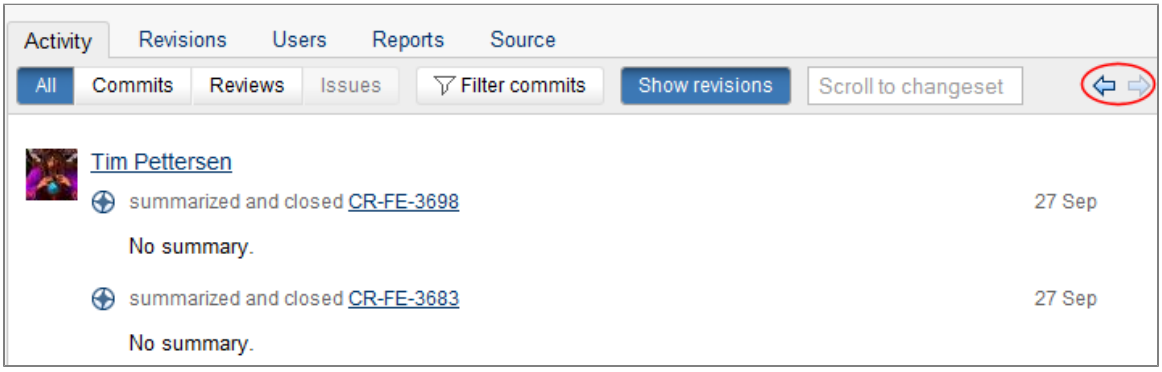
We're continuing our ongoing work to improve the FishEye user interface. This release includes a brand new repository administration console and a redesigned header. We've also improved the toolbars on a number of screens including the dashboard, as well as replaced dropdowns throughout the application with autocomplete controls.




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.



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

**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-5735 | XSS vulnerability in Crucible Author Mapping | Vitaly Osipov [Atlassian] | Andrew Lui [Atlassian] | | Closed | Fixed | Apr 20, 2011 | May 30, 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 Heemskerk [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 changeset comments in search results | Vitaly Osipov [Atlassian] | Andrew Lui [Atlassian] | | Closed | Fixed | Apr 20, 2011 | May 30, 2011 |
| | 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 | Feb 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 Heemskerk [Atlassian] | Michael Heemskerk [Atlassian] | | Closed | Fixed | Jan 03, 2011 | May 09, 2011 |
| | FE-3248 | ClearCase UCM activity support is brittle (detection of modified activities etc.) | Unassigned | Michael Heemskerk [Atlassian] | | Closed | Fixed | Aug 27, 2010 | May 09, 2011 |
| | FE-3187 | Base ClearCase: File revisions from excluded branches are not picked up when merged to an included branch | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Sep 16, 2010 | May 09, 2011 |
| | FE-3176 | Copy and rename detection for Git repositories is broken | Jason Hinch [Atlassian] | None | | Closed | 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 | | Closed | Fixed | Nov 12, 2010 | May 09, 2011 |
| | FE-3170 | ClearCase: ancestor relations are not created for CC file revisions | Michael Heemskerk [Atlassian] | None | | Closed | 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 Heemskerk [Atlassian] | None | | Closed | Fixed | Dec 08, 2010 | May 09, 2011 |
| | FE-3086 | ClearCase: checkins of different users may be grouped into the same changeset | Unassigned | None | | Closed | Fixed | Nov 03, 2010 | May 09, 2011 |
| | FE-3079 | Fisheye Incompatible with Git 1.7.3 on Windows | Jason Hinch [Atlassian] | None | | Closed | Fixed | Nov 29, 2010 | May 09, 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 Heemskerk [Atlassian] | Michael Heemskerk [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-5246 | activity stream shows duplicate commits in cvs when commits are aggregated. | Geoff Crain [Atlassian] | Geoff Crain [Atlassian] | | Closed | Fixed | Dec 19, 2010 | Jan 11, 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 Heemskerk [Atlassian] | Michael Heemskerk [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 found, stack overflows) | Michael Heemskerk [Atlassian] | Anna Buttfield [Atlassian] | | Closed | Fixed | Nov 29, 2010 | Jan 06, 2011 |
| | CRUC-5009 | Help links in admin are failing | Tom Davies [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Nov 24, 2010 | Jan 06, 2011 |
| | CRUC-4993 | Git - error when switch directories | Jason Hinch [Atlassian] | Adam Ahmed [Atlassian] | | Closed | Fixed | Nov 21, 2010 | Dec 12, 2010 |
| | CRUC-4951 | changeset view: when a file has a really long name/path the options overflow covering content | Craig Sharkie [Atlassian] | Brydie McCoy [Atlassian] | | Closed | Fixed | Nov 15, 2010 | Dec 13, 2010 |
| | | Trusted apps jira integration docs | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | CRUC-4904 | are missing paths critical to sub-task creation | Unassigned | Anna Lyons [Atlassian] | | Closed | 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] | | Closed | Fixed | Nov 11, 2010 | Dec 13, 2010 |
| | CRUC-4894 | font shiftiness on eyeql results | Craig Sharkie [Atlassian] | Tim Pettersen [Atlassian] | | Closed | Fixed | Nov 10, 2010 | Dec 13, 2010 |
| | CRUC-4891 | ie8 - black waiting screen with spinner off to the side before creating a review | Craig Sharkie [Atlassian] | Tim Pettersen [Atlassian] | | Closed | Fixed | Nov 10, 2010 | Dec 13, 2010 |
| | 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 Heemskerk [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 (23 issues)**

| Type | Key | Summary | Assignee | Reporter | Priority | Status | Resolution | Created | Upda |
|---|---|---|---|---|---|---|---|---|---|
| | 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 ( 2011 |
| | FE-3048 | "Inconsistent state while finding the revision for file" with Git repositories | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Nov 17, 2010 | May ( 2011 |
| | CRUC-5309 | XSS vulnerability in Crucible Review Reload | Unassigned | Andrew Lui [Atlassian] | | Closed | Fixed | Jan 04, 2011 | Jan 1 2011 |
| | CRUC-5008 | File history review links are misstyled | Seb Ruiz [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Nov 24, 2010 | Nov 2 2010 |
| | CRUC-5003 | Crucible and FishEye dashboard pages have missing or partial ghost features | Craig Sharkie [Atlassian] | Craig Sharkie [Atlassian] | | Closed | Fixed | Nov 23, 2010 | Nov 2 2010 |
| | CRUC-4998 | ChangesetTooltipAction.findChangeSet() eagerly creates cruRevisions | Seb Ruiz [Atlassian] | Brendan Humphreys [Atlassian] | | Closed | Fixed | Nov 22, 2010 | Nov 2 2010 |
| | CRUC-4996 | Map .json files to javascript syntax hilighting | Seb Ruiz [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Nov 22, 2010 | Nov 2 2010 |
| | CRUC-4991 | Manifest implementation has large memory footprint | Anna Buttfield [Atlassian] | Anna Buttfield [Atlassian] | | Closed | Fixed | Nov 21, 2010 | Nov 2 2010 |
| | CRUC-4987 | Git Indexing Issues | Conor MacNeill [Atlassian] | Conor MacNeill [Atlassian] | | Closed | Fixed | Nov 18, 2010 | Nov 2 2010 |
| | CRUC-4985 | Cannot Change Password in User Settings | Tom Davies [Atlassian] | Dylan Hansen [Atlassian] | | Closed | Fixed | Nov 18, 2010 | Dec ( 2010 |
| | CRUC-4983 | "You do not have permission to see all the search results." for custom search filter in Crucible | Unassigned | Gurleen Anand [Atlassian] | | Closed | Fixed | Nov 18, 2010 | Nov 1 2010 |
| | CRUC-4982 | Review splitter dragging highlights text | Craig Sharkie [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Nov 18, 2010 | Nov 2 2010 |
| | | ClassCastException in | Anna Buttfield | Anna Buttfield | | | | Nov 14, | Nov 2 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | CRUC-4928 | ParameterSetInsertion.checkAuthors due to concurrent map modification | [Atlassian] | [Atlassian] | | Closed | Fixed | 2010 | 2010 |
| | CRUC-4912 | NullPointerException when Checking the Author for the BaseLineCount | Anna Buttfield [Atlassian] | Chai Ying Chan [Atlassian] | | Closed | Fixed | Nov 12, 2010 | Jan 0 2011 |
| | CRUC-4906 | Diffs link gone from watch emails | Seb Ruiz [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Nov 11, 2010 | Nov 1 2010 |
| | CRUC-4879 | you are not prompted for a repo rescan after removing an excludes pattern | Joe Xie [atlassian] | Tim Pettersen [Atlassian] | | Closed | Fixed | Nov 10, 2010 | Nov 1 2010 |
| | CRUC-4872 | Rescan repo after adding an exclude fails | Tom Davies [Atlassian] | Tim Pettersen [Atlassian] | | Closed | Fixed | Nov 10, 2010 | Dec 0 2010 |
| | CRUC-4842 | Return the ability to add "favourites" | Tom Davies [Atlassian] | Tim Buntel [Atlassian] | | Closed | Fixed | Nov 08, 2010 | Nov 2 2010 |
| | CRUC-4576 | Review titles not truncated properly | Unassigned | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 14, 2010 | Jan 0 2011 |
| | CRUC-4575 | Review titles not truncated properly | Seb Ruiz [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 14, 2010 | Nov 2 2010 |
| | CRUC-4488 | When the QuickSearch fails, the spinner remains active and there is no failure indication to the user | Pierre-Etienne Poirot [Atlassian] | Pierre-Etienne Poirot [Atlassian] | | Closed | Fixed | Oct 11, 2010 | Jan 0 2011 |
| | 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 1 2010 |
| | CRUC-4175 | JspTagException is thrown when navigating annotated view | Unassigned | Ming Giet Chong [Atlassian] | | Closed | Fixed | Sep 01, 2010 | Nov 2 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 | Updat |
|---|---|---|---|---|---|---|---|---|---|
| | FE-3417 | Show the branch a changeset was committed to in watch emails | Seb Ruiz [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 25, 2010 | May 0 2011 |
| | FE-3393 | Sections of the code are unreadable in side-by-side diff in IE7 | Unassigned | None | | Closed | Fixed | Jun 28, 2010 | May 0 2011 |
| | FE-3392 | Javascript error occurs when clicking a folder when you have never selected a branch in the branch filter | Unassigned | None | | Closed | Fixed | Nov 05, 2010 | May 0 2011 |
| | FE-3391 | Watch email contains broken link when there is no context path | Unassigned | None | | Closed | Fixed | Jul 14, 2010 | May 0 2011 |
| | FE-3390 | Search results data sorting incorrect | Unassigned | None | | Closed | Fixed | Oct 15, 2010 | May 0 2011 |
| | FE-3388 | "IllegalStateException: Indexer is stopped" on 2.4 Upgrade | Unassigned | None | | Closed | Fixed | Oct 21, 2010 | May 0 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 0 2011 |
| | FE-3385 | Committer name bolds when you mouseover it somewhere else | Unassigned | None | | Closed | Fixed | Oct 13, 2010 | May 0 2011 |
| | FE-3379 | svn does not fully support the branch selector | Unassigned | None | | Closed | Fixed | Oct 26, 2010 | May 0 2011 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FE-3378 | plain text rendering of wiki text in the activity stream for changeset comments is wrong | Unassigned | None | | Closed | Fixed | Nov 02, 2010 | May 0 2011 |
| | FE-3377 | Quick file links in activity stream disappear if browser isn't wide enough | Unassigned | None | | Closed | Fixed | Oct 20, 2010 | May 0 2011 |
| | FE-3375 | ClearCaseRepositoryCache.getChangeSet throws IllegalStateException when the changeset does not exist. The cache for other SCM types don't | Unassigned | None | | Closed | Fixed | Oct 21, 2010 | May 0 2011 |
| | FE-3374 | [IE] More ie issues | Unassigned | None | | Closed | Fixed | Oct 12, 2010 | May 0 2011 |
| | FE-3369 | Path segments in Include/Exclude cannot be parsed in 2.4.0 | Unassigned | None | | Closed | Fixed | Oct 24, 2010 | May 0 2011 |
| | FE-3365 | Too many objects are kept in memory during changeset indexing, resulting in an OutOfMemoryException | Unassigned | None | | Closed | Fixed | Nov 04, 2010 | May 0 2011 |
| | FE-3364 | Activity tab for all repositories is gone in 2.4 | Unassigned | None | | Closed | Fixed | Oct 27, 2010 | May 0 2011 |
| | FE-3360 | GWT admin for ClearCase is slow because cleartool commands are being executed too often and too early | Michael Studman [Atlassian] | Michael Heemskerk [Atlassian] | | Closed | Fixed | Sep 27, 2010 | May 0 2011 |
| | FE-3348 | ClearCase: performance of incremental indexing has decreased in 2.4.0 (Base CC only) | Michael Heemskerk [Atlassian] | Michael Heemskerk [Atlassian] | | Closed | Fixed | Oct 21, 2010 | May 0 2011 |
| | FE-3346 | Reload of admin page causes CPU max out | Unassigned | Michael Heemskerk [Atlassian] | | Closed | Fixed | Sep 27, 2010 | May 0 2011 |
| | FE-3188 | Can't view projects in FishEye only mode | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Aug 09, 2010 | May 0 2011 |
| | FE-3179 | ClearCase: CPU usage has increased significantly between 2.3.6 and 2.4-M6 | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Oct 04, 2010 | May 0 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 Heemskerk [Atlassian] | None | | Closed | Fixed | Oct 26, 2010 | May 0 2011 |
| | FE-3163 | ClearCase: project exclude rules aren't being stored | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Oct 24, 2010 | May 0 2011 |
| | FE-3148 | Change the logging level of valid 'error' messages | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Jun 02, 2010 | May 0 2011 |
| | FE-3145 | Excludes cause indexing to fail for Git repositories | Unassigned | None | | Closed | Fixed | Nov 01, 2010 | May 0 2011 |
| | FE-3124 | Git: Use of file content-hash to map to file revision is insufficient | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Oct 18, 2010 | May 0 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 Heemskerk [Atlassian] | None | | Closed | Fixed | Oct 24, 2010 | May 0 2011 |
| | FE-3106 | ClearCase: FE maxes out CPU when the connection to the ClearCase server is lost | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Sep 21, 2010 | May 0 2011 |
| | FE-3095 | ClearCase interactive process logs timeout and forgets it has failed | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Aug 30, 2010 | May 0 2011 |
| | FE-3050 | ClearCase: nothing is indexed when branch excludes is non-empty and branch | Michael Heemskerk | None | | | Fixed | Oct 25, | May 0 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | includes is empty | [Atlassian] | | | Closed | | 2010 | 2011 |
| | FE-3049 | ClearCase: error in storing repository configuration after creating new repo | Michael Heemskerk [Atlassian] | None | | Closed | Fixed | Oct 04, 2010 | May 0 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 Heemskerk [Atlassian] | James Roper [Atlassian] | | Closed | Fixed | Sep 12, 2010 | Nov 1 2010 |
| | CRUC-4850 | Upgrade to JIRA Blockers 1.6.9 | Tim Pettersen [Atlassian] | Tim Pettersen [Atlassian] | | Closed | Fixed | Nov 09, 2010 | Nov 0 2010 |
| | CRUC-4845 | Map xslt files for xml syntax highlighting | Seb Ruiz [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Nov 08, 2010 | Nov 0 2010 |
| | CRUC-4716 | PermaID field is empty in REST JSON responses | Nick Pellow [Atlassian] | Nick Pellow [Atlassian] | | Closed | Fixed | Oct 29, 2010 | Nov 0 2010 |
| | CRUC-4710 | addGeneralComment fails when called outside a HTTP request. e.g. a plugin commit listener | Joe Xie [atlassian] | Nick Pellow [Atlassian] | | Closed | Fixed | Oct 27, 2010 | Nov 1 2010 |
| | CRUC-4693 | Snippet Page: Join Review issues | Adam Ahmed [Atlassian] | Adam Ahmed [Atlassian] | | Closed | Fixed | Oct 26, 2010 | Nov 1 2010 |
| | CRUC-4692 | Reviewers dropdown in Crucible Reviews spins forever in IE8 - cannot add reviewers | Adam Ahmed [Atlassian] | David Hergert [PAETEC] | | Closed | Fixed | Oct 26, 2010 | Nov 0 2010 |
| | CRUC-4691 | PropertyNotFoundException when Priting Snippet | Michael Heemskerk [Atlassian] | Chai Ying Chan [Atlassian] | | Closed | Fixed | Oct 26, 2010 | Nov 1 2010 |
| | CRUC-4690 | No Description for Repository LDAP Settings | Seb Ruiz [Atlassian] | Chai Ying Chan [Atlassian] | | Closed | Fixed | Oct 26, 2010 | Nov 0 2010 |
| | CRUC-4656 | "Edit Details", "Summarize" and "Tools" menu missing under "Reviews" tab in Opera in 2.4 | Craig Sharkie [Atlassian] | Gurleen Anand [Atlassian] | | Closed | Fixed | Oct 21, 2010 | Nov 0 2010 |
| | CRUC-4647 | Bring the "Deleted" lozenge in line with product/corporate styles | Unassigned | Craig Sharkie [Atlassian] | | Closed | Fixed | Oct 20, 2010 | Jun 19 2011 |
| | CRUC-4636 | Make more of GWT admin deep linkable | Michael Studman [Atlassian] | Michael Studman [Atlassian] | | Closed | Fixed | Oct 19, 2010 | Nov 1 2010 |
| | CRUC-4635 | Better linking in GWT Admin | Michael Studman [Atlassian] | Michael Studman [Atlassian] | | Closed | Fixed | Oct 19, 2010 | Nov 1 2010 |
| | CRUC-4614 | Annotation links from file history don't preserve wayback spec | Tom Davies [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 18, 2010 | Nov 1 2010 |
| | CRUC-4613 | Links to comments in the same review don't work | Tom Davies [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 18, 2010 | Nov 1 2010 |
| | CRUC-4612 | infinite redirects to login via dialog iframes | Tom Davies [Atlassian] | Geoff Crain [Atlassian] | | Closed | Fixed | Jun 20, 2010 | Nov 1 2010 |
| | CRUC-4597 | Clear filter button goes to changeset | Tom Davies [Atlassian] | Chris Lam [Atlassian] | | Closed | Fixed | Oct 15, 2010 | Nov 2 2010 |
| | | Crucible Inbox: Summarize state review | Tom Davies | Matthew | | | | Oct 14, | Nov 2 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | CRUC-4585 | with incomplete reviewers shows in no filter | [Atlassian] | Watson [Atlassian] | | Closed | Fixed | 2010 | 2010 |
| | CRUC-4547 | Join Review is duplicated | Craig Sharkie [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 12, 2010 | Nov 0 2010 |
| | CRUC-4541 | Sometimes can't edit objectives | Adam Ahmed [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 12, 2010 | Nov 2 2010 |
| | CRUC-4529 | Wiki links are broken | Pierre-Etienne Poirot [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 12, 2010 | Nov 0 2010 |
| | CRUC-4527 | [Admin] No 'test connection' button when creating repo | Michael Studman [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 12, 2010 | Nov 2 2010 |
| | CRUC-4491 | [Admin] Repo list needs pagination and filtering | Michael Studman [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 11, 2010 | Nov 1 2010 |
| | CRUC-4363 | Changeset header UI annoyances | Jason Hinch [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Sep 22, 2010 | Jan 0 2011 |
| | CRUC-4347 | Set min height on body of snippet page | Adam Ahmed [Atlassian] | Jason Hinch [Atlassian] | | Closed | Fixed | Sep 21, 2010 | Nov 0 2010 |
| | CRUC-4241 | selecting a tag on reports page won't work | Tom Davies [Atlassian] | Geoff Crain [Atlassian] | | Closed | Fixed | Sep 16, 2010 | Nov 1 2010 |
| | CRUC-4131 | Links in new admin ui are not real links | Michael Studman [Atlassian] | Anna Buttfield [Atlassian] | | Closed | Fixed | Aug 30, 2010 | Nov 1 2010 |
| | CRUC-4081 | [Stars] Stars in dropdowns are broken | Tom Davies [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Aug 26, 2010 | Nov 1 2010 |
| | CRUC-4047 | [IE] Problems on Snippet Page | Adam Ahmed [Atlassian] | Anna Lyons [Atlassian] | | Closed | Fixed | Aug 22, 2010 | Nov 0 2010 |
| | CRUC-3673 | Make the rename user functionality more visible | Pierre-Etienne Poirot [Atlassian] | Andrew Myers [Atlassian] | | Closed | Fixed | Jun 23, 2010 | Nov 0 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 perforce 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 [Atlassian] | 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 throw errors in 2.4.0 | Conor MacNeill [Atlassian] | Dylan Hansen [Atlassian] | | Closed | Fixed | Oct 26, 2010 | Oct 26, 2010 |
| | CRUC-4674 | PermId element can not be serialized to JSON for REST requests | Nick Pellow [Atlassian] | Nick Pellow [Atlassian] | | Closed | 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] | | Closed | Fixed | Oct 24, 2010 | Jan 05, 2011 |
| | CRUC-4651 | NullPointerException in SvnCache.getFileRevision | Michael Heemskerk [Atlassian] | Michael Heemskerk [Atlassian] | | Closed | 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] | | Closed | Fixed | Oct 20, 2010 | Nov 15, 2010 |
| | CRUC-4616 | SVN branch selector enabled with ajax loaded directories | Tom Davies [Atlassian] | Seb Ruiz [Atlassian] | | Closed | 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] | | Closed | Fixed | Oct 15, 2010 | Nov 15, 2010 |
| | CRUC-4591 | [IE7] - Horizontal Scrollbar on File History page | Adam Ahmed [Atlassian] | Adam Ahmed [Atlassian] | | Closed | Fixed | Oct 14, 2010 | Jan 05, 2011 |
| | CRUC-4538 | Changeset link changes server url | Tom Davies [Atlassian] | Seb Ruiz [Atlassian] | | Closed | Fixed | Oct 12, 2010 | Jan 05, 2011 |
| | CRUC-4534 | Changeset hover of CVS broken | Tom Davies [Atlassian] | Seb Ruiz [Atlassian] | | Closed | 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*

- **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.

DOWNLOAD latest version

See the documentation on Upgrading to this version.

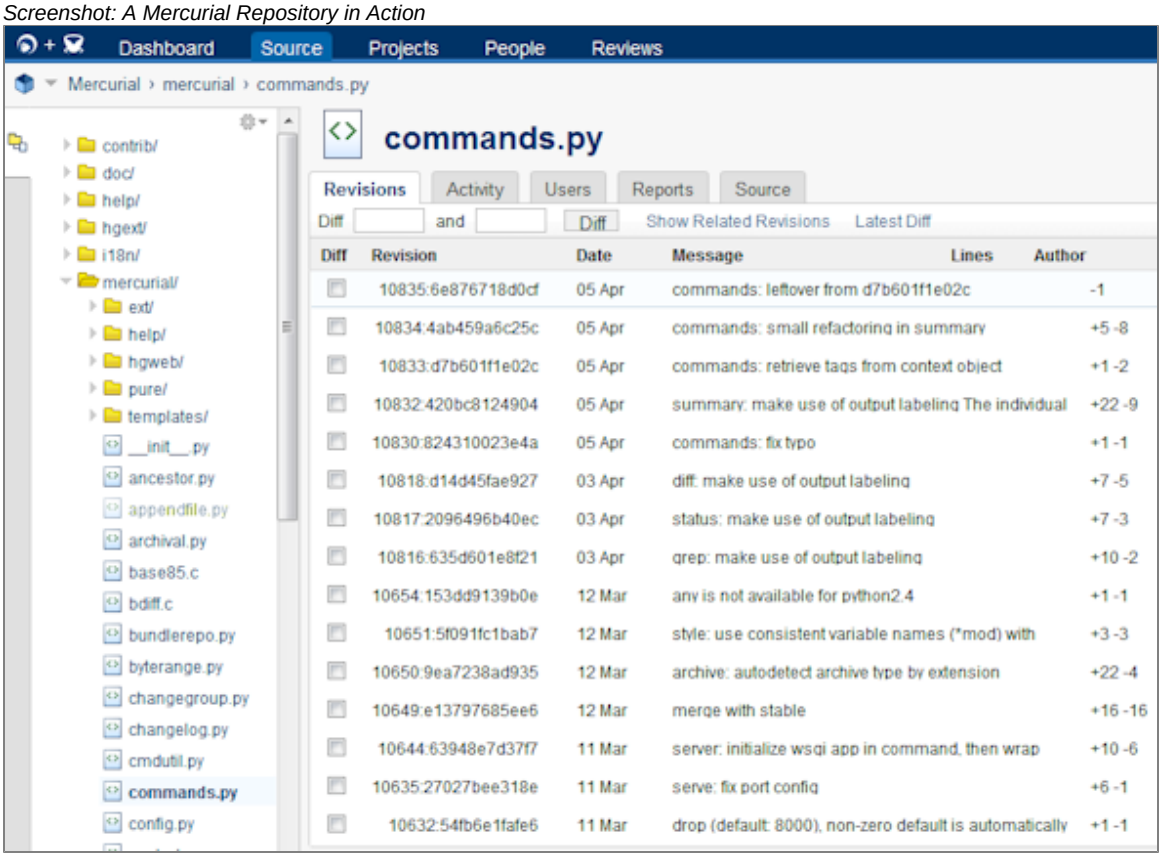
| Installing FishEye 2.3 |
| --- |
| Download FishEye 2.3 now. See the documentation on Upgrading to this version. |

## 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*



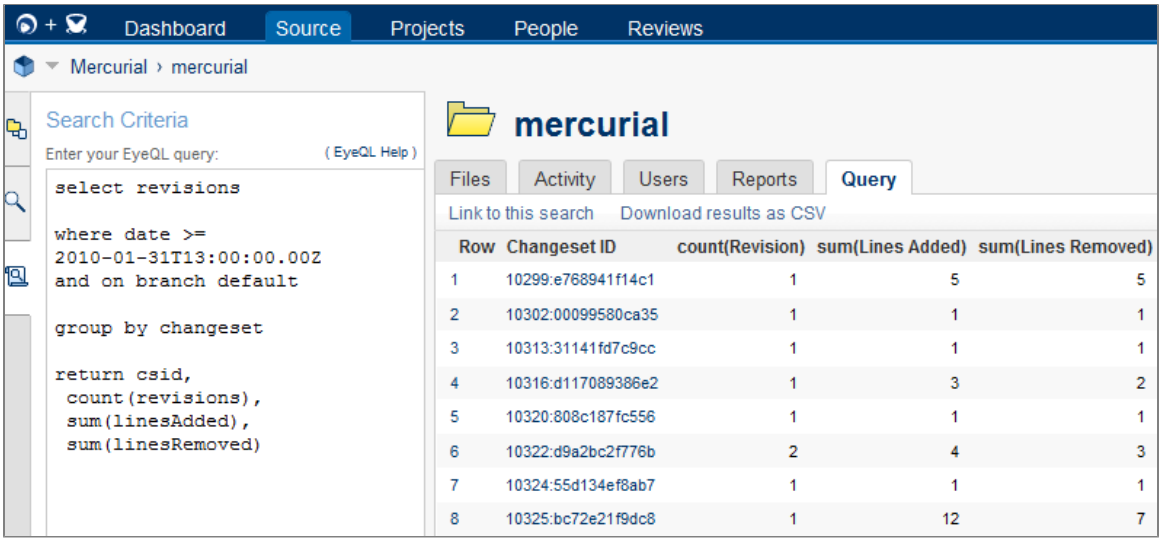

See the documentation for more details.

## 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:

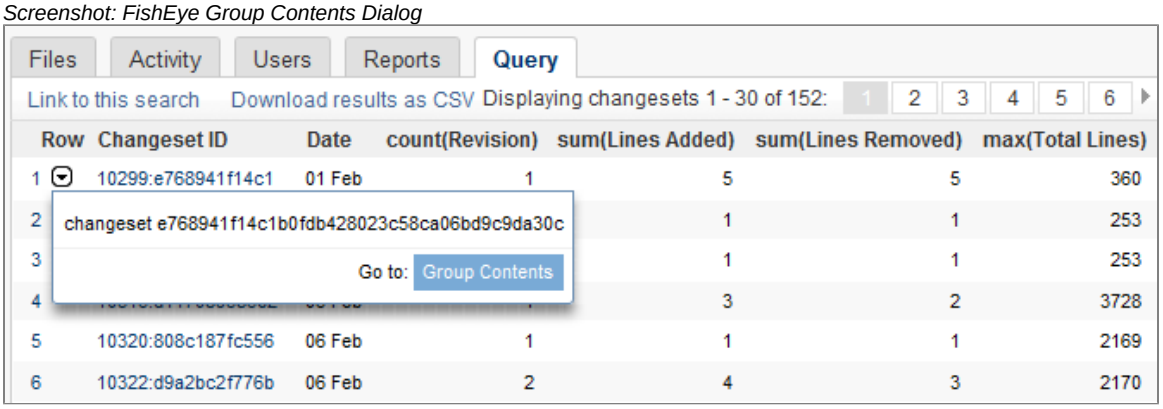
- Lines added or removed,
- Authors,
- Changesets,
- Tags,
- Reviews (and more).

*Screenshot: New EyeQL Functions*




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*

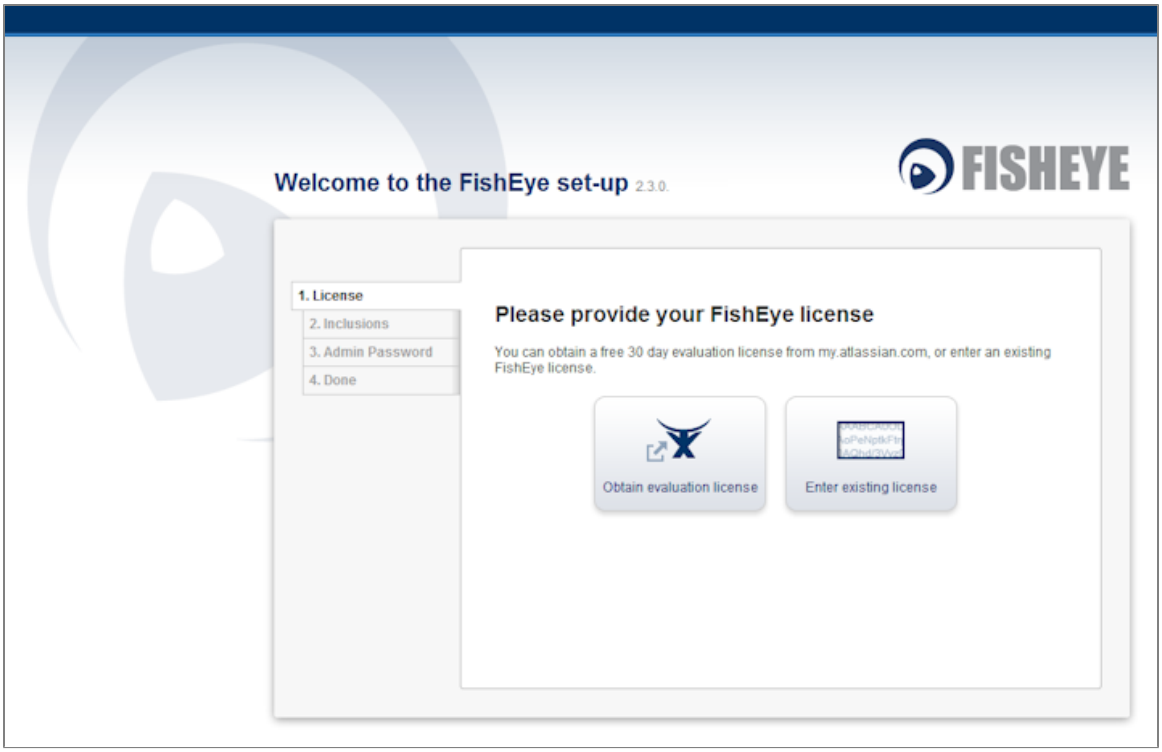



See the EyeQL documentation for more information.

### 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

**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.

| JIRA Issues (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.

| **JIRA Issues (10 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-3356 | ClearCase autocreate view broken if view already exists (in a different location) | | Closed |
| FE-3120 | ClearCase InteractiveProcess ignores error stream | | Closed |
| FE-3251 | ClearCase repository configuration in FishEye administration is overly complicated and incomplete | | Closed |
| FE-3196 | Test Repository option for ClearCase repositories needs to do more thorough checks | | 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-2543 | FishEye fails to process changes for a dynamic view | | Closed |
| FE-3418 | Check P4TICKETS environment variable when testing p4 connection, and on p4 login failure | | 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-2649 | Merge studio crowd classloader fix from 1.6.x | | Closed |
| FE-2676 | Search results preview dropdown not showing up | | Closed |
| FE-3419 | Support a url parameter to specify the REST return format | | Closed |
| FE-3373 | incorrect line count changes | | 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-3368 | Lucene indexer fails on null values | | 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.

| JIRA Issues (8 issues) | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-3286 | HG: Support http/s authentication for repos by placing username/password into URL | | Closed |
| FE-3285 | HG: Error on windows for files starting with "." | | Closed |
| FE-2612 | Sys-Info page does not show all the LDAP settings correctly | | Closed |
| FE-2610 | Rest login method is ambigous for CAPTCHA | | Closed |
| FE-2608 | within a repo, a quicksearch spins forever (NPE on server) | | Closed |
| FE-2607 | Fisheye 2.3.x is missing ability to rescan revisions | | Closed |

| | | | |
|---|---|---|---|
| FE-2592 | UCM Activity/Project information should be cached to stop FishEye re-requesting the same information | | Closed |
| FE-2380 | SASL support with bundled svnkit | | Closed |

**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.

| **JIRA Issues (7 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2613 | Document Clearcase branch inclusion/exclusion rules | | Closed |
| FE-2605 | Startup is broken for upgrades with licenses that have expired maintenance. | | Closed |
| FE-2597 | FishEye continually reindexes Base ClearCase changesets made at the same time by different users | | Closed |
| FE-2595 | Allow users to include/exclude specific UCM Streams from indexing | | Closed |
| FE-2573 | Duplicate comments shown for UCM Projects when 'Integration Streams Only' set to false | | Closed |
| FE-2570 | FishEye currently does not index version '0' | | Closed |
| FE-1766 | In annotated file view, line-level links to other revisions go to wrong line | | Closed |

## 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*

- 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

# FishEye 2.2 Release Notes

FishEye 2.5 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.

DOWNLOAD latest version

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.

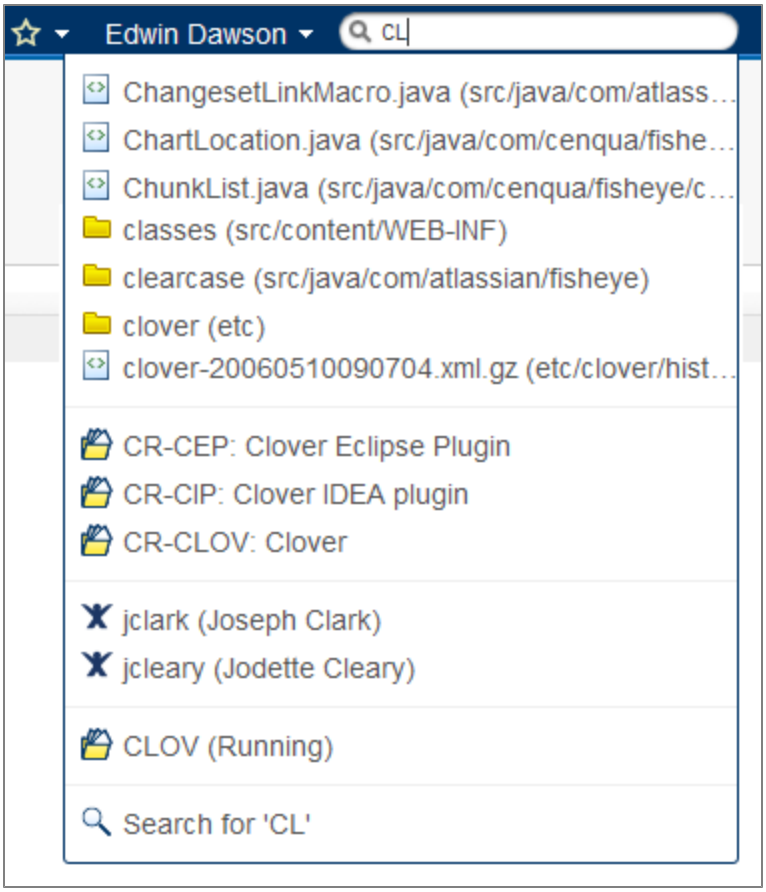
*Screenshot: Enhanced 'Side-by-Side Diff' Mode*

See the documentation for more details.

## 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*

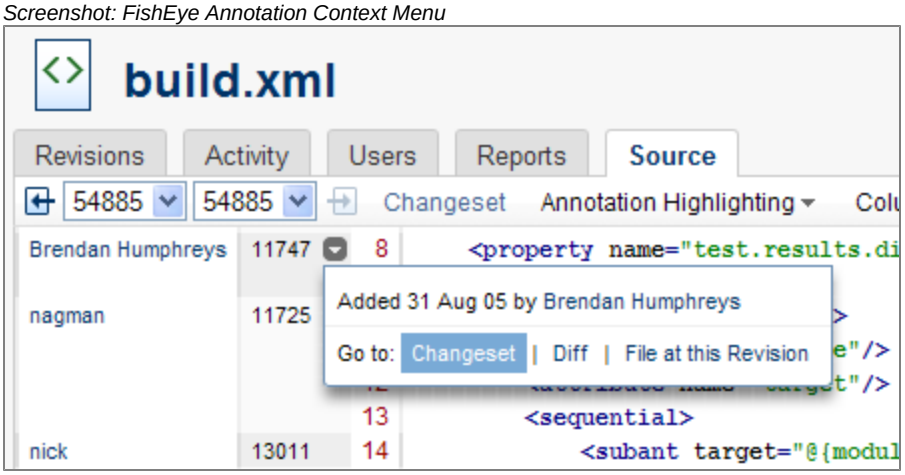



See the documentation for more details.


### 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**

**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.

| **JIRA Issues (34 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2572 | Repository rescan configuration options not visible | | Closed |
| FE-2567 | Add CAPTCHA-reset to admin interface | | Closed |
| FE-2566 | Repository reindex performed after aborted index using incorrect start date | | Closed |
| FE-2559 | Use HttpOnly cookies for session cookies | | Closed |
| FE-2558 | Base ClearCase support should handle processing VOB subpaths when reusing snapshot views | | Closed |
| FE-2557 | Use the Repository name when creating views for Base ClearCase projects | | Closed |
| FE-2554 | Base ClearCase code is not processing all branches of a VOB | | Closed |
| FE-2534 | Incorrect author information being used when processing version deletions | | Closed |
| FE-2527 | FileNotFoundException generated when content is processed | | Closed |
| FE-2524 | NullPointerException generated when repository index is processed | | Closed |
| FE-2508 | Permissions error with new projects - no Moderator role | | Closed |
| FE-2503 | When using the "initial import" feature, an anonymous committer is automatically created which counts against your license | | Closed |
| FE-2448 | Allow user to suppress specific branches | | Closed |
| FE-2401 | Support XML files for ClearCase repository | | Closed |
| FE-2385 | 'remember' cookies for users with characters needing to be url escaped (e.g. '@' or ' ') in their username do not work | | Closed |
| FE-2301 | Close LDAP NamingEnumerations | | Closed |
| FE-2016 | Repository link is broken when the file name with a hash in the P4 repository | | Closed |
| FE-1844 | if the file revision is marked as binary the "diff" checkbox should be disabled | | Closed |

| | | | |
|---|---|---|---|
| FE-1840 | user options has "File history view mode: Physical/Logical" — does this mean anything now? | | Closed |
| FE-1838 | user options has "Default annotation mode: Age/Author/None" — where is mixed? | | Closed |
| FE-1833 | STUPID - user options has "Length of tag list: Hide/Medium/Long" — "hide" isn't a length | | Closed |
| FE-1826 | funny line artefact in review statistics | | Closed |
| FE-1685 | activity stream sporadically display a wrong author for a commit activity item | | Closed |
| FE-1677 | hover near edge of page pushes page out of layout | | Closed |
| FE-1517 | Safari ignores content type for FishEye URLs ending in .tar.gz | | Closed |
| FE-1474 | Sidebar outstanding todos | | Closed |
| FE-1317 | Update existing report plugins | | Closed |
| FE-1315 | Scrap VelocityHelper & expose Velocity classes to plugins | | Closed |
| FE-1134 | Global Scheduler improvements | | Closed |
| FE-1060 | allow a way to set the misfire instruction | | Closed |
| FE-963 | allow a way to register job class names with the quartz manager | | Closed |
| FE-913 | continuing work on the fisheye api | | Closed |
| FE-883 | show partial matches in quicknav results | | Closed |
| FE-664 | Create a sanity/perf test of the FishEye REST api | | Closed |

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:

| **JIRA Issues (19 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2517 | Java errors when using ClearCase | | Closed |
| FE-2498 | Paths with upper-case characters break stats from changeset index | | Closed |
| FE-2495 | OOME on quicknavchangeset indexing | | Closed |
| FE-2492 | Include support for users whose names have spaces | | Closed |
| FE-2490 | Unable to open connection to Subverion version 1.6.6 | | Resolved |

| | | | |
|---|---|---|---|
| FE-2482 | Upgrade bundled SVNKit from 1.3.1 to 1.3.2 | | Closed |
| FE-2475 | git repo updater cannot be set | | Closed |
| FE-2472 | java.lang.IndexOutOfBoundsException generated when updating repository details | | Closed |
| FE-2462 | Listing Projects via REST is slow because we eagerly load the allowed reviewers, even though that field is 'expandable' | | Closed |
| FE-2459 | Incorrect location used when processing the Base ClearCase history for a multi-level VOB | | Closed |
| FE-2456 | jira keys are not autolinked in html email | | Closed |
| FE-2450 | Updater - Poll Period - Disable Poll always remains unchecked for 'never' | | Closed |
| FE-2432 | Trailing spaces added to some file names in scan, causing cleartool desc to fail | | Closed |
| FE-2425 | Configurable polling period for Git repositories | | Closed |
| FE-2405 | NPE reported when new directories/files are indexed | | Closed |
| FE-2402 | VCS error \| SVN repository client error accessing local file | | Closed |
| FE-2390 | FishEye becomes a CPU Hog (infinite loop) if you enter the wrong password for a repository | | Closed |
| FE-2385 | 'remember' cookies for users with characters needing to be url escaped (e.g. '@' or ' ') in their username do not work | | Closed |
| FE-2280 | ClearCase scan issue | | Closed |

## 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*

> **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 2.5 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.

DOWNLOAD latest version

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. |

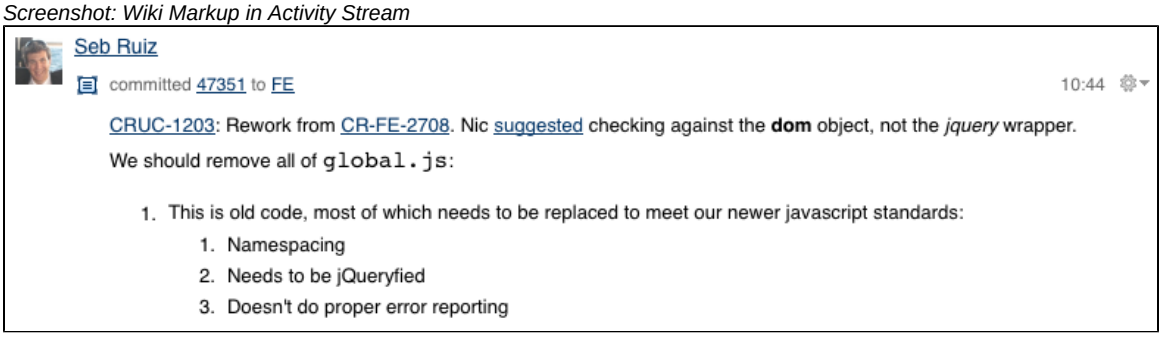
## Highlights of FishEye 2.1

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*

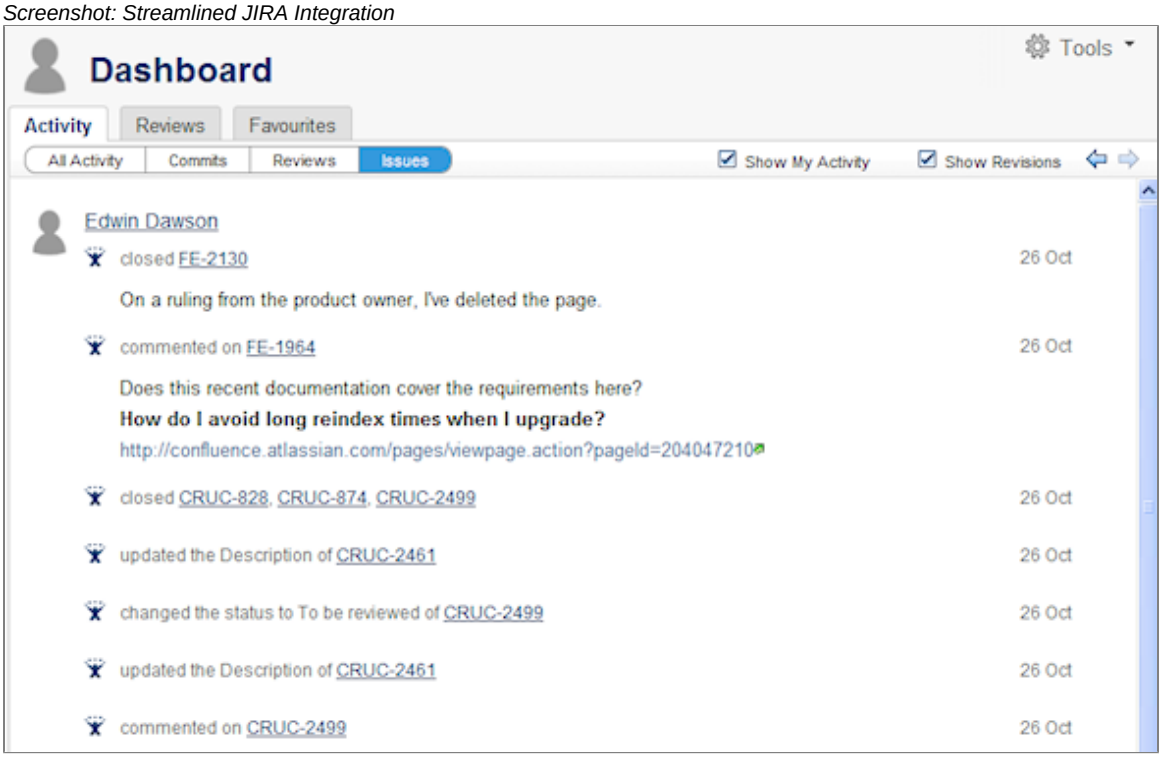


2

### 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*

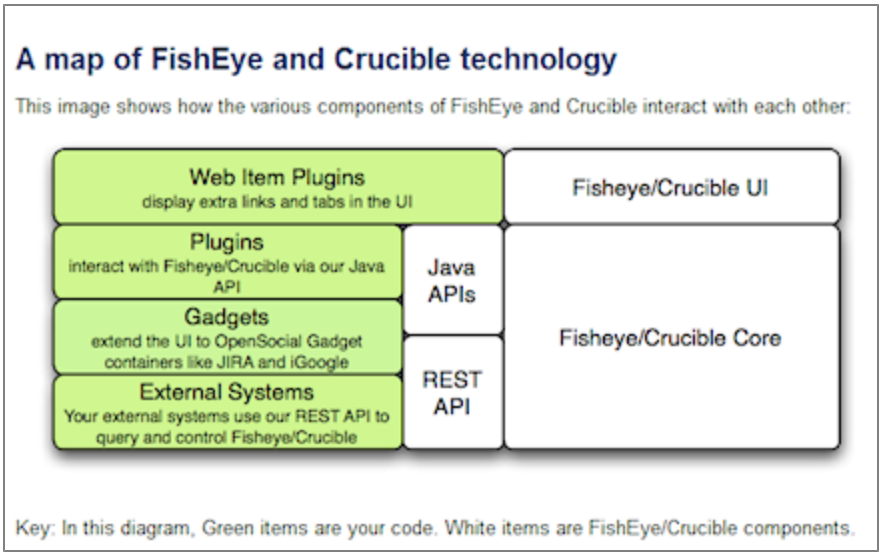



3

### 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*

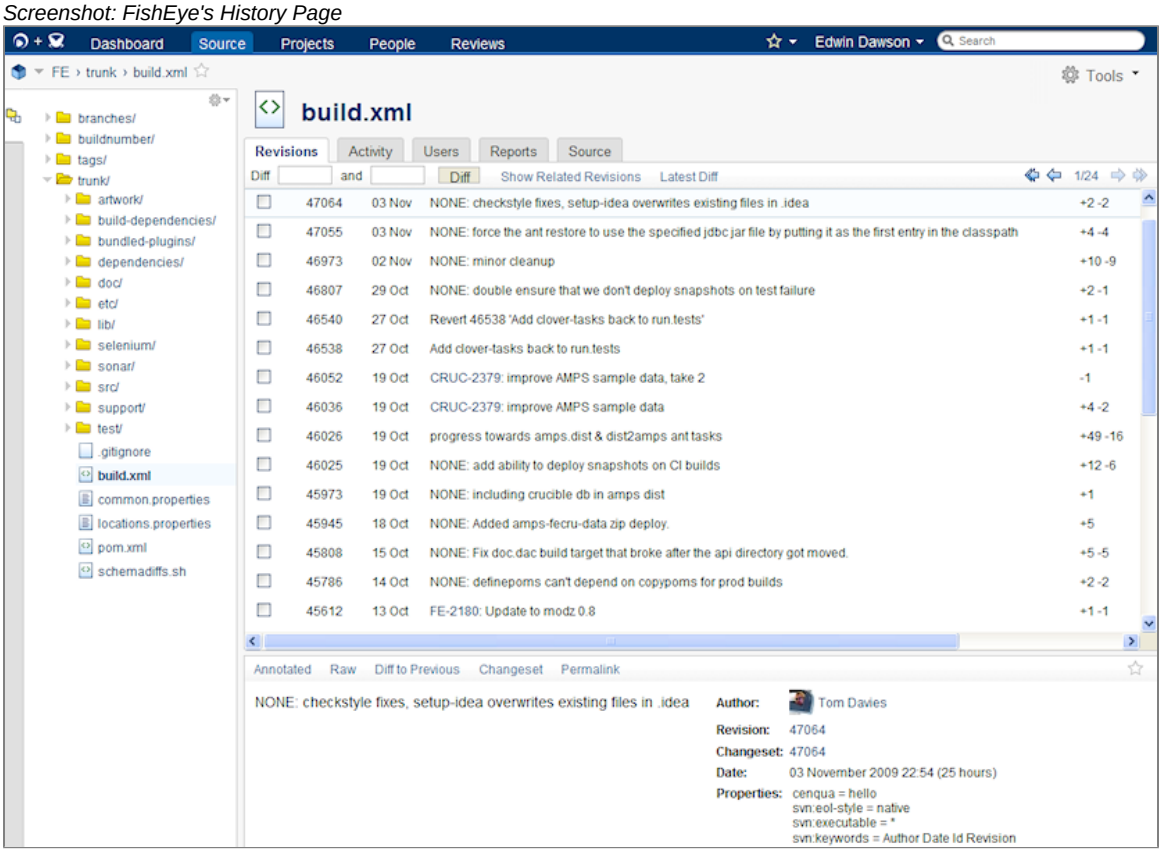



4

### History Page Performance Increases

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.

*Screenshot: FishEye's History Page*



5

### ClearCase Support Now in Beta

FishEye's support for IBM ClearCase is now in beta, following extensive feedback from our brave alpha testers.

6

### 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

**27th January 2010**

This is a bugfix release, addressing the following issues:

| JIRA Issues (9 issues) | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2415 | NullPointerException generated when viewing XML documents | | Closed |
| FE-2411 | cvs updater ignores period==0 if it finds a remote tag | | Closed |
| FE-2409 | Initial scan lshistory of large VOB never finishes | | Closed |
| FE-2404 | FishEye should re-examine Activity contents to determine if subsequent changes have been added | | Closed |
| FE-2403 | Information about VOB comparisons is being logged at INFO rather than DEBUG | | Closed |
| FE-2398 | Base ClearCase logic should apply the include/exclude rules when processing lshistory output | | Closed |
| FE-2386 | StringIndexOutOfBoundsException when viewing files in a ClearCase repo | | Closed |
| FE-2127 | EyeQL only returns one file per changeset | | Closed |
| FE-2017 | Cache directory not deleted when repository removed from Fisheye | | Closed |

#### 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:

| JIRA Issues (15 issues) | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2424 | Fisheye's default shell fails on Solaris | | Closed |
| FE-2366 | Make View Storage Location an optional field | | Closed |
| FE-2345 | Invoke cleartool in interactive mode rather than creating new instances for each invocation | | Closed |
| FE-2343 | Improve git rename detection | | Closed |
| FE-2342 | Allow git executable location to be configured | | Closed |
| FE-2337 | An cleartool exception is generated when the Admin UI is opened | | Closed |
| FE-2333 | The 'VOB to Include' field only displays non-PVOBs | | Closed |
| FE-2328 | Values entered in the VOB/UCM Inclusion rules are not displayed on the Repository Details page | | Closed |
| FE-2327 | NullPointerException in ClearCaseRepositoryEngine | | Closed |
| FE-2325 | Allow FishEye to index private ClearCase VOBs | | Closed |
| FE-2323 | Improve performance of Base ClearCase indexing | | Closed |
| FE-2312 | No vobsToReturn were found error scanning ClearCase even though the Vobs were shown in the UI when configuring the repo | | Closed |
| FE-2299 | Base ClearCase code generates incorrect view name on Windows | | Closed |
| FE-1921 | author mappings dont work in the settings dialog | | Closed |
| FE-1642 | Pasting a license string with an ampersand in the ui, breaks FE | | Closed |

**From 2.1.1 to 2.1.2**

### 19th November 2009

This is a bugfix release.

The following issues are addressed by this release:

| JIRA Issues (4 issues) | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2308 | Do not creat tags when files are copied into a tag area from outside the repository scope | | Closed |
| FE-2307 | Can't set the use built-in symbolic rules for Subversion repositories | | Closed |
| FE-2303 | revision icons not inline on p4 repo activity streams | | Closed |
| FE-2282 | Quicklaunch of repositories closes before repository selection. | | Closed |

**From 2.1.0 to 2.1.1**

**17th November 2009**

This is a bugfix release.

The following issues are addressed by this release:

| **JIRA Issues (8 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2302 | File History Diff navigation broken in 2.1 | | Closed |
| FE-2301 | Close LDAP NamingEnumerations | | Closed |
| FE-2300 | Column headers don't display correctly in IE7 on Repositories tab | | Closed |
| FE-2298 | out of bounds error? | | Closed |
| FE-2296 | Attempt to continue when errors occur in the content retrieval logic | | Closed |
| FE-2271 | Exceptions generated when users attempt to test ClearCase Repository connection | | Closed |
| FE-2255 | Activity details are being duplicated on the Activity tab | | Closed |
| FE-2176 | Enable users to specify if they want the includes/excludes rule to be exact matches | | Closed |

## 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*

> **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 2.5 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.

DOWNLOAD latest version

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. |

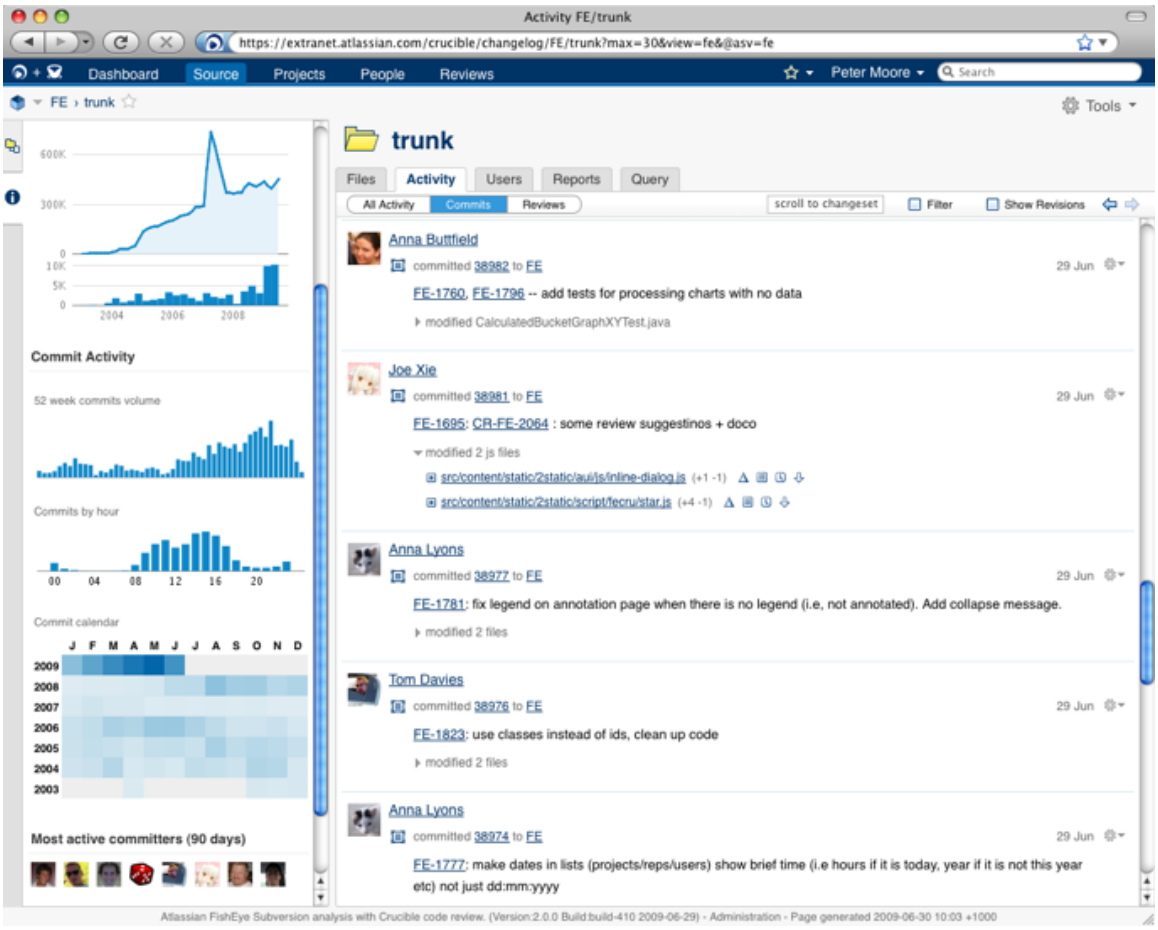
## Highlights of FishEye 2.0

1

### 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.

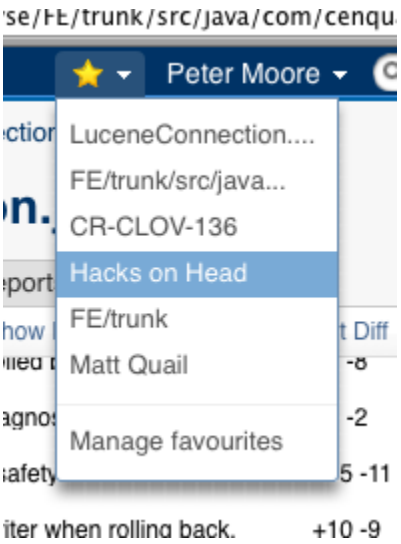
*Screenshot: Users on trunk sorted by activity*


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*

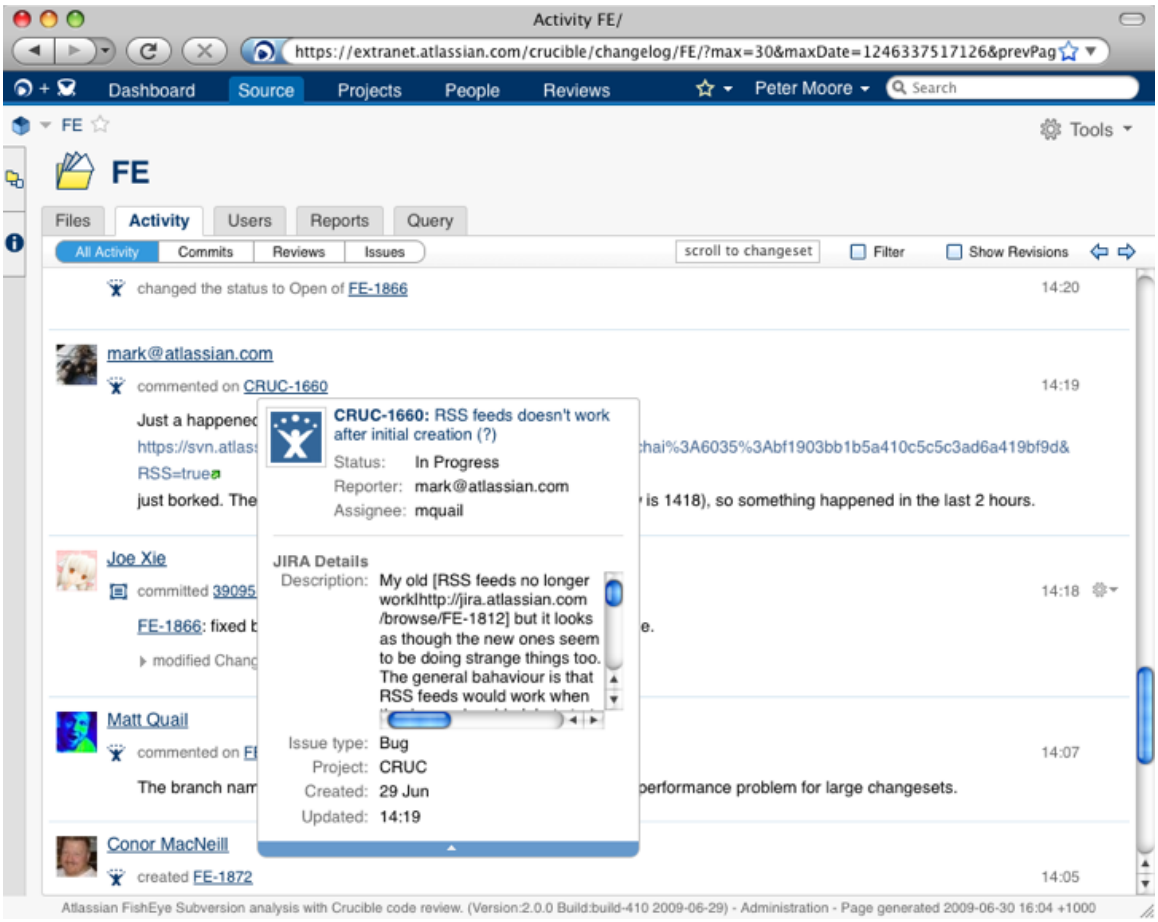


4

### 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*

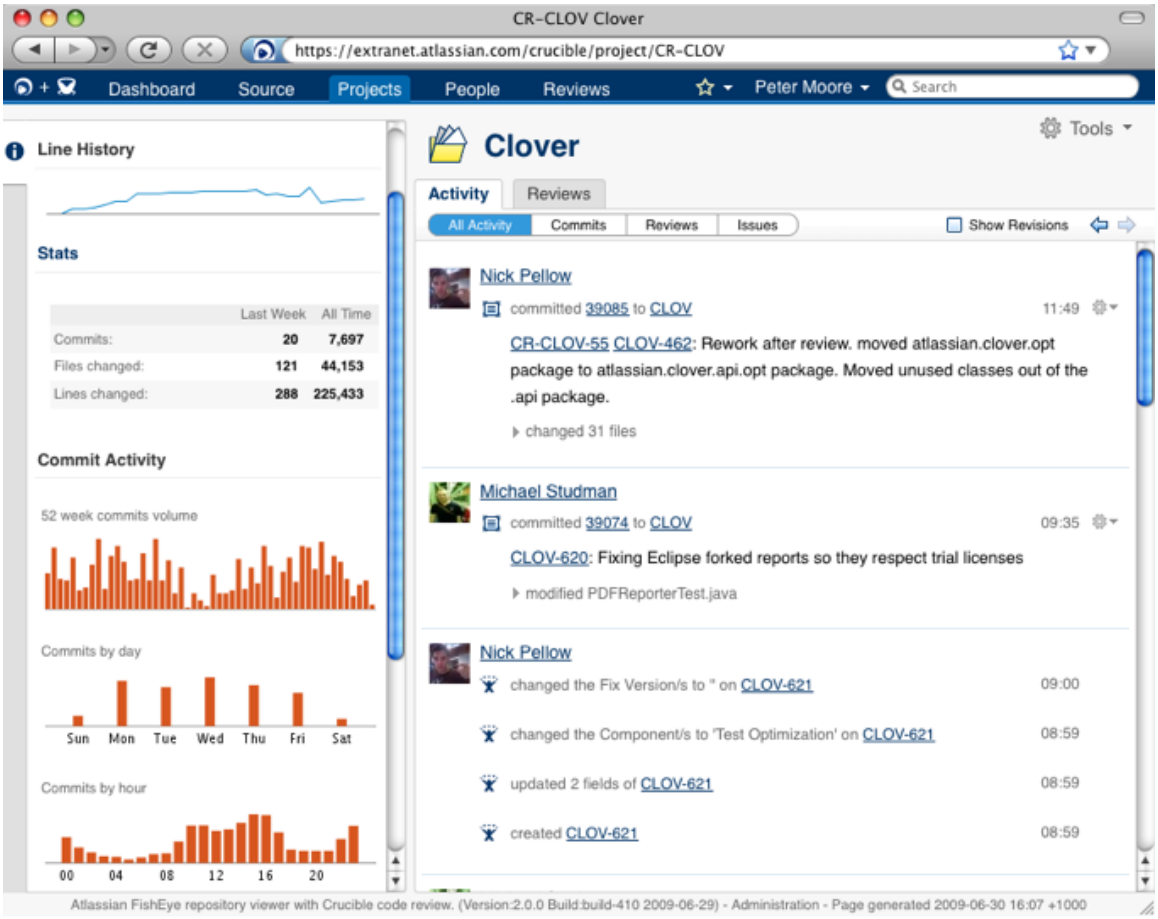



5

### 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*

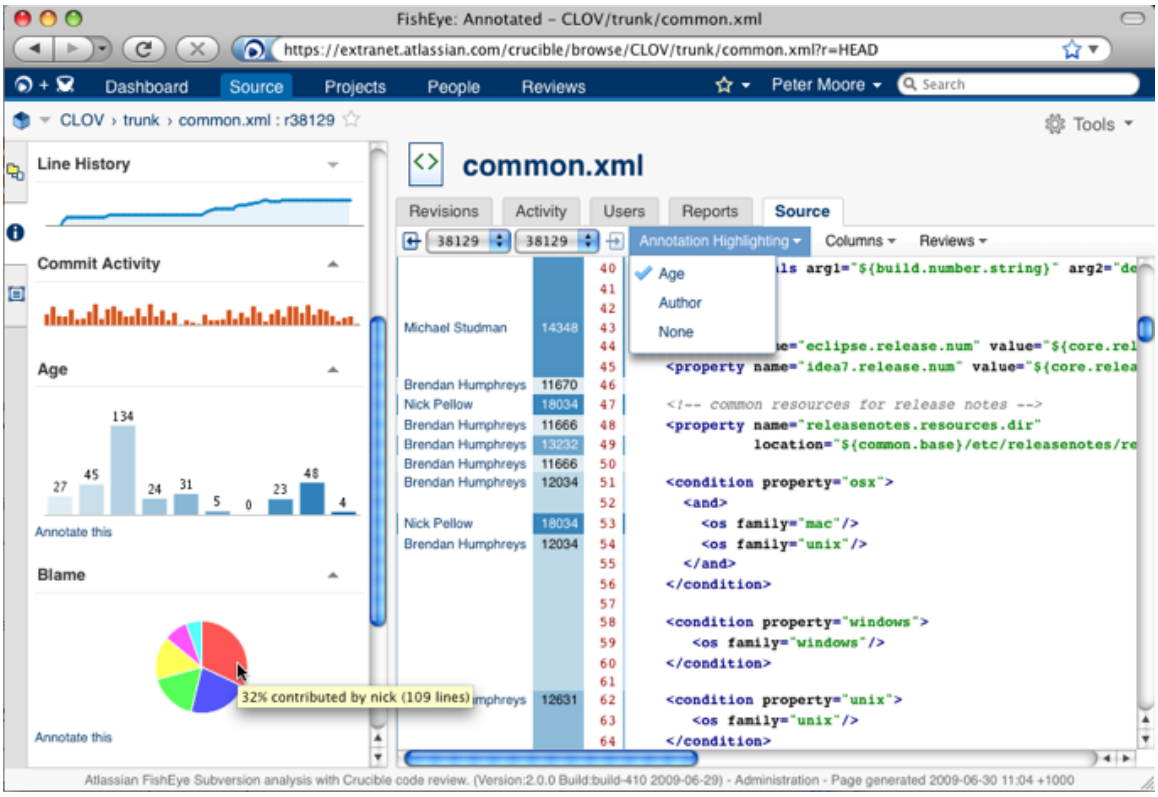



6

### 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*

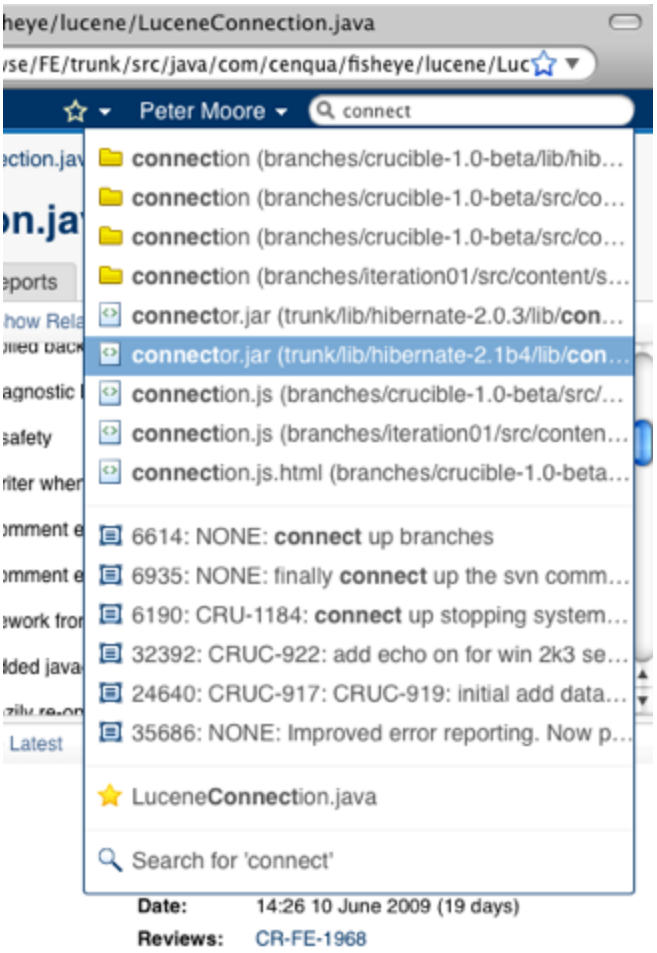



7

### 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*

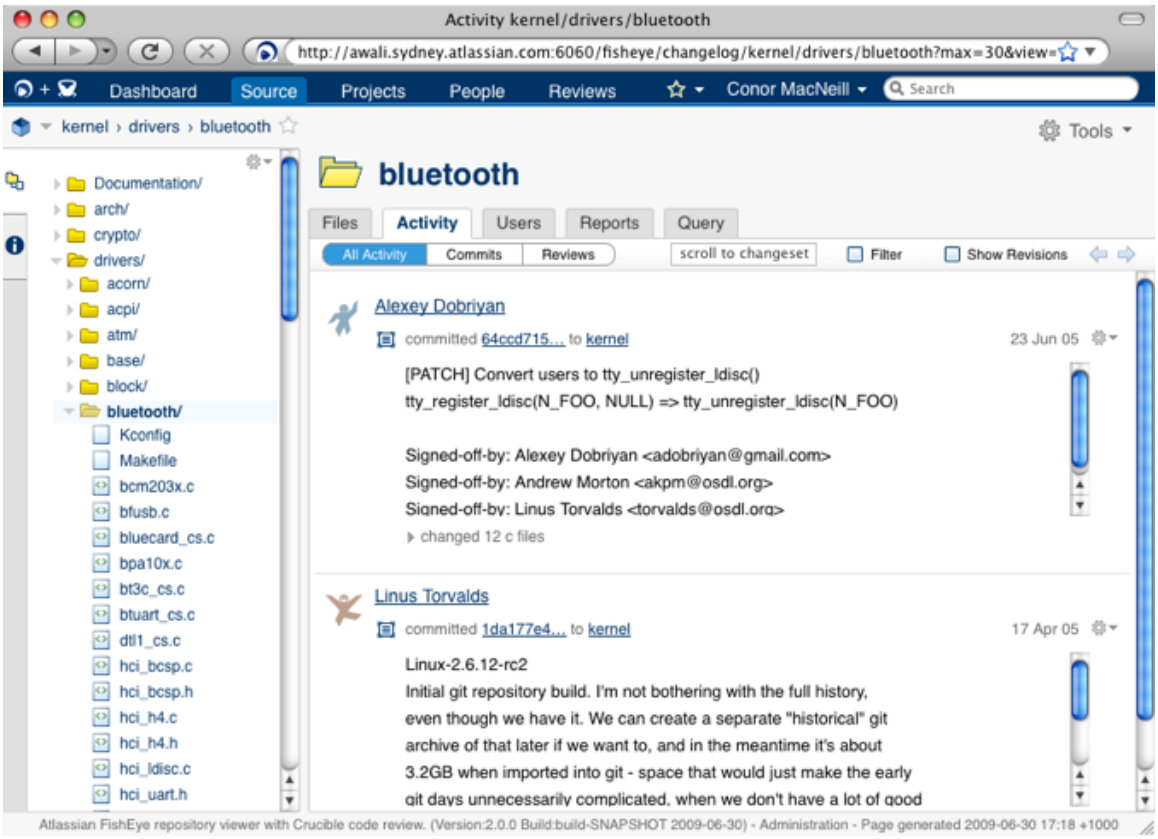




### Git Beta

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

**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:

| JIRA Issues (12 issues) | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2163 | Repository Edit form - Return key may cause 'Accept' button to become occluded | | Closed |
| FE-2157 | Atlassian Footer not appearing as a footer | | Closed |
| FE-2143 | Fisheye treats files as binary if they used to have a svn:mime-type property, even after the property has been removed | | Closed |
| FE-2095 | NPE when clicking on the 'Edit License' link | | Closed |
| FE-2054 | add FishEye/Crucible support in AMPS, continued | | Closed |
| FE-2029 | Issue names (and links) not shown in activities or issues tab after JIRA integration | | Closed |
| FE-1933 | Create personal license for FishEye | | Closed |
| FE-1851 | Remove a repository should also remove the caches, indexes and temporarily files | | Closed |
| FE-1425 | User browser should not use case-sensitive search on e-mail address | | Closed |
| FE-1367 | Add case sensitive flag to the includes | | Closed |
| FE-1277 | Fisheye automatic tag detection doesn't work well | | Closed |
| FE-712 | finish upgrade to atlassian licensing 2.0 | | Closed |

**From 2.0.3 to 2.0.4**

### 8th September 2009

This is a bugfix and improvement release. This release addresses the following issues:

| JIRA Issues (9 issues) | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2078 | Excludes doesn't work well with path | | Closed |
| FE-2050 | Merge AMPS support to branch | | Closed |
| FE-2034 | in changeset dropdown, "Context Permalink" becomes "Permalink" | | Closed |
| FE-2032 | any comma in the field for svn:author is preventing author constraint drop down from functional properly | | Closed |
| FE-2021 | Excludes do not work on checkouts | | Closed |
| FE-2020 | Email subject should say '[Fisheye]', not '[Crucible]' | | Closed |
| FE-2006 | Fisheye not compatible with crowd 2.0 | | Closed |

| | | | |
|---|---|---|---|
| FE-1939 | Improve performance for opening the changelog page when there is very large changesets | | Closed |
| FE-773 | Set the default permgen space to 128m if it hasn't been set | | Closed |

**From 2.0.2 to 2.0.3**

### 18th August 2009

This is a bugfix release which includes the following issues:

| **JIRA Issues (8 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-1995 | Create Support Zip and create support request via mail does not include SCM Light plugin configuration from 2.0 | | Closed |
| FE-1987 | Fisheye throws error when you add a svn repository with 0 revisions | | Closed |
| FE-1984 | Git not supported in FishEye/Crucible Rest API | | Closed |
| FE-1963 | Add link to FishEye 101 in Readme.html | | Closed |
| FE-1960 | refinements to author blame | | Closed |
| FE-1934 | Toggling debug on and off states a confusing message in the logs. | | Closed |
| FE-1901 | OOME with very large memory consumption by org.apache.lucene.document.Document | | Closed |
| FE-811 | File Explorer author constraint is case-sensitive | | Closed |

**From 2.0.1 to 2.0.2**

### 24th July 2009

This is a bugfix release which includes the following issues:

| **JIRA Issues (6 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-1957 | Excessive invocation of svn blame | | Closed |
| FE-1948 | Svnkit thorws "svn: authentication cancelled" for svn repositories accessed via svn+ssh | | Closed |
| FE-1943 | Option to hide user email addresses | | Closed |
| FE-1938 | NPE when viewing changelog with showid | | Closed |
| FE-1932 | NoSuchElementException when defining a start revision for a svn url + path that is a sub section of a bigger repository | | Closed |
| FE-778 | Automatically periodically reindex repositories, particularly with Perforce | | Closed |

**From 2.0 to 2.0.1**

### 14th July 2009

This is a bugfix release which includes the following issues:

**JIRA Issues (45 issues)**

| Key | Summary | Priority | Status |
|---|---|---|---|
| FE-1930 | Throwing NPE when loading charts | | Closed |
| FE-1900 | ELException when clicking on Report or Query tabs | | Closed |
| FE-1896 | Rescanning duplicates entries in the Changeset Index | | Closed |
| FE-1894 | Show My Activity toggle in Toolbar has a tooltip that says Exclude your activity, irrespective of what state the toggle is in | | Closed |
| FE-1891 | Can't view created projects in FE only mode | | Closed |
| FE-1888 | improve .idea project setup (move to a template) | | Closed |
| FE-1887 | set Findbugs config to threshold so current build passes - make build fail if we go backwards | | Closed |
| FE-1886 | improve concat Ant task so that it works when switching between concat/no-concat | | Closed |
| FE-1885 | Cannot delete projects when switching from Crucible + FIsheye, to just Fisheye and any added projects added after the crucible license is removed, do not show up in the projects tab | | Closed |
| FE-1882 | Avatar link creation doesn't lowercase email address first | | Closed |
| FE-1881 | NPE exception when indexing repository with Native Client | | Closed |
| FE-1875 | Corrupt @asv cookie preferences cause servlet exceptions on dashboard | | Closed |
| FE-1874 | poor z-index of crucible hover and the email review dialog box causing overlaps | | Closed |
| FE-1866 | scroll-to-changeset doesn't work if "Activity" is your sticky tab | | Closed |
| FE-1864 | Quick Search results doesn't warn when searching a repository that is being scanned. | | Closed |
| FE-1863 | (the new) RSS feeds should return 401 instead of the html login page | | Closed |
| FE-1857 | encoding problem on jira comment activity items | | Closed |
| FE-1849 | Profile dialog forget my name | | Closed |
| FE-1845 | the colours on the "blame" chart don't correspond to the colours of the blame in the annotation | | Closed |
| FE-1841 | cannot diff binary files" message could be prettier | | Closed |
| FE-1839 | inconsistent capitalisation | | Closed |
| FE-1836 | weird tooltip value (doesn't say it's a commit message) | | Closed |
| FE-1823 | Star widget causes duplicate ids | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1821 | Safari 3.21: tools menu dropdown borkage | | Closed |
| FE-1801 | IE: changeset activity stream menus too large and have transparent background | | Closed |
| FE-1798 | If you change from side-by-side to unified in a review then change from soft to hard wrap you are reverted back to side-by-side. | | Closed |
| FE-1797 | Shouldn't be able to submit empty quicksearch | | Closed |
| FE-1796 | NegativeArraySizeException in charts | | Closed |
| FE-1788 | Quicksearch Results Icon Alignment Poor | | Closed |
| FE-1786 | Only the top part of 'Search Reviews' on the custom review filter is clickable | | Closed |
| FE-1785 | 'All Activity' and 'Reviews' tabs on /committer/REPO/NAME appear active on mouseover | | Closed |
| FE-1783 | Deleting a repo doesn't delete committer mappings | | Closed |
| FE-1781 | Legend and Commit activity accordion segments don't add value when expanded | | Closed |
| FE-1780 | Breadcrumbs on /foo/admin/editAvatarType-default.do are wrong | | Closed |
| FE-1779 | gravitar admin page table has poor layout | | Closed |
| FE-1777 | Latest Activity on projects page should have time as well | | Closed |
| FE-1775 | Author/Moderator name at the top of the review should be a link | | Closed |
| FE-1767 | not selecting a repo in "Settings: User Mapping" and clicking "add" leads to empty user drop down | | Closed |
| FE-1764 | JIRA change aggregation shows same issue id twice when two files attached to the same issue | | Closed |
| FE-1741 | In the browse view, the 'show/hide deleted files' toggle is not respected when clicking on a node in the dir tree | | Closed |
| FE-1663 | hover popups' shadows dont expand with the popups, | | Closed |
| FE-1514 | Document disk space usage changes | | Closed |
| FE-1416 | CR-FE-1499 Rework | | Closed |
| FE-1402 | handle blank commiter names | | Closed |
| FE-1189 | SVN initialscan does not create changeset index until end of scan | | Closed |

**From 2.0 Beta3 to 2.0**

**30th June 2009**

Full list of issues in this release:

| **JIRA Issues (200 issues)** | | | |
|---|---|---|---|

| Key | Summary | Priority | Status |
|---|---|---|---|
| FE-2379 | Viewing a very large raw file triggers an OutOfMemoryError | | Closed |
| FE-1865 | removing a favourite and then clicking on it again causes an error | | Closed |
| FE-1854 | Activity stream on project page doesn't filter by mapped JIRA projects | | Closed |
| FE-1847 | 'star' instead of 'favourite' in cog menu | | Closed |
| FE-1818 | Links from files in a changeset to the file history page don't include the revision | | Closed |
| FE-1815 | a con: XML namespace is being added to saved config.xml | | Closed |
| FE-1809 | Enhance README.html to include links to Evaluator Guide, Installation Guide, and Upgrade Guide | | Closed |
| FE-1808 | FishEye RC1 tarball contains Readme.HTML that links to Crucible docs | | Closed |
| FE-1804 | we shouldn't show an empty star before a search has been performed - they throw an exception when clicked | | Closed |
| FE-1802 | with the blue background on the comment form you can't tell which button is active when tabbing through them | | Closed |
| FE-1799 | Collapse all files in review doesn't work if there is a comment anchor in the URL | | Closed |
| FE-1784 | Soft wrapping always selected in prefs menu | | Closed |
| FE-1778 | admin project page doesn't render default and allowed reviewers | | Closed |
| FE-1771 | Commit by hour chart time axis numbering incorrect | | Closed |
| FE-1770 | Inconsistent and unsatisfactory highlighting colours on review page | | Closed |
| FE-1768 | defect label on comments has no css, but draft does | | Closed |
| FE-1760 | Infinite loop in CalculatedBucketGraphXY | | Closed |
| FE-1754 | Cross-repo QS repository membership and ordering changes between pages | | Closed |
| FE-1703 | Typo error | | Closed |
| FE-1698 | Improve LDAP Authentication so it does not Abandon initial bind or do an CMP | | Closed |
| FE-1627 | Add option to disable the indexing of diffs | | Closed |
| FE-1519 | Change include/exclude parameters for Restore | | Closed |
| FE-1515 | p4 calls labels command without supplying path when it needs to get the info on a particular label | | Closed |
| FE-1513 | Upgrade to latest AGSL-1 | | Closed |

| | | | |
|---|---|---|---|
| FE-1506 | Truncate number of lines in scrolling log | | Closed |
| FE-1505 | Restore should not fail if run from a different directory | | Closed |
| FE-1504 | Update FE System Requirements - remove IE 6 from support browsers | | Closed |
| FE-1500 | Let BackupManager store the job data in config.xml | | Closed |
| FE-1492 | Anna's misc M7 non-ui tasks | | Closed |
| FE-1489 | Add user timezone to send request by support so that all support requests have a valid timezone. | | Closed |
| FE-1485 | chart tweaks and fixes | | Closed |
| FE-1484 | Add directory tree to other pages | | Closed |
| FE-1483 | Enhance directory tree to display file | | Closed |
| FE-1479 | Make scale/chart nicer on code-metrics-report | | Closed |
| FE-1477 | Fix code-metrics-report | | Closed |
| FE-1473 | Make report plugin urls pretty | | Closed |
| FE-1472 | Fix 'revisions' tabulation on code-metrics report | | Closed |
| FE-1467 | Problem accessing Admin Remote API if the global anonymous access is off | | Closed |
| FE-1465 | Refactor report plugin packages | | Closed |
| FE-1464 | Report Mode Navigation (similar to Search Mode Navigation) | | Closed |
| FE-1463 | Root (No Repository Context) Search Page | | Closed |
| FE-1455 | Date Pickers should be jQuery pickers (remove old script file when done) | | Closed |
| FE-1453 | Fix Structure (make search query controls in 'fixed/hard' section up top) | | Closed |
| FE-1452 | Remove blanks from Author box | | Closed |
| FE-1451 | Change "Search All Directories" to "Searching... <BREADCRUMB TRAIL>" | | Closed |
| FE-1449 | Remove .do from global quicksearch url | | Closed |
| FE-1448 | QuickSearch Improvements | | Closed |
| FE-1447 | Search Mode Navigation | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1441 | Make revision comments appear on just one line in revision table | | Closed |
| FE-1435 | No progress indicator on dir tree twiddle | | Closed |
| FE-1427 | Improve group membership management | | Closed |
| FE-1418 | Combine summarize and close emails | | Closed |
| FE-1417 | CR-FE-1480 Rework | | Closed |
| FE-1415 | fix up pagination in users/ page | | Closed |
| FE-1414 | Fix it | | Closed |
| FE-1412 | Fix it | | Closed |
| FE-1411 | Fix QuickNav | | Closed |
| FE-1410 | View Changeset Stream Context Fixes | | Closed |
| FE-1408 | activity-streams plugin wrangling | | Closed |
| FE-1401 | add a comment as one of the return columns in /search/FE/ | | Closed |
| FE-1392 | Change mset messages displayed in browse history table should not be truncated (except via css) | | Closed |
| FE-1391 | quicksearch pages continuously displays progress indicator | | Closed |
| FE-1390 | Fix error handling when loading directory subtrees via ajax | | Closed |
| FE-1389 | Poor Javascript performance on file history page | | Closed |
| FE-1388 | can sort by checkboxes in browse view | | Closed |
| FE-1387 | bad title name in the config servlet for light svn plugin. | | Closed |
| FE-1386 | Make physical/logical link labels indicate change to different state, not the current state | | Closed |
| FE-1385 | Make tooltip content more descriptive | | Closed |
| FE-1383 | RSS improvements | | Closed |
| FE-1369 | Use Session Expires due to Jetty bug (JETTY-688) | | Closed |
| FE-1365 | Page listing only source files to browse | | Closed |
| FE-1361 | just do it | | Closed |
| FE-1360 | Chart jsp refactorings | | Closed |

| | | | |
|---|---|---|---|
| FE-1357 | write tag | | Closed |
| FE-1356 | Consistent time and age formats | | Closed |
| FE-1355 | update stream jsps to render a changeset | | Closed |
| FE-1354 | update all stream actions to take a csid | | Closed |
| FE-1353 | Maintain stream context when viewing a changeset page | | Closed |
| FE-1352 | just do it | | Closed |
| FE-1351 | preference and toggle to exclude own activity from home page stream | | Closed |
| FE-1349 | just do it | | Closed |
| FE-1348 | Switch all hover popups to use ajs.hover | | Closed |
| FE-1347 | just do it | | Closed |
| FE-1346 | Open settings pages in ajs.dialog | | Closed |
| FE-1345 | gravatar servlet | | Closed |
| FE-1344 | config screen + jsps | | Closed |
| FE-1343 | Avatar config screens | | Closed |
| FE-1342 | make the committer page group by user if there are multiple committer with same user | | Closed |
| FE-1341 | User & committer list rationalisation | | Closed |
| FE-1340 | just do it | | Closed |
| FE-1339 | One jsp for global quicksearch and repo qsearch | | Closed |
| FE-1338 | just do it | | Closed |
| FE-1337 | Switch all dropdowns to ajs.dropdown | | Closed |
| FE-1336 | i.e. the old style changelog or what you get after you hit expand all | | Closed |
| FE-1335 | preference to show files rather than summary in actvity streams | | Closed |
| FE-1334 | legend unfucking | | Closed |
| FE-1333 | sparkline | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1332 | line history sparklines and user line history chart improvements | | Closed |
| FE-1331 | just do it | | Closed |
| FE-1330 | Integrate new html for diff/annotation pages | | Closed |
| FE-1329 | just do it | | Closed |
| FE-1322 | this should be done in a single query for all committers you're interested in, not once for every user or every committer | | Closed |
| FE-1321 | you could use Math.max/min in a few places here | | Closed |
| FE-1320 | you shouldn't need to set the commiters as a parameter, just set the BreakdownOption in ParameterSetQuery to AUTHOR | | Closed |
| FE-1313 | there is no real point paging if we are going to do these types of calculations for every user in the system. It's just not going to work imho. Is sorting by num-reviews a must have? If so do one query that counts reviews for all users, grouped by u | | Closed |
| FE-1312 | change the comparator BY_RECENT_ACTIVITY to sort nulls last -- remove have NoActivityItem | | Closed |
| FE-1311 | use a better UI than ^ and V, see craig/pete | | Closed |
| FE-1310 | Rework from CR-FE-1441: FE-1236: sortable and paged list of users at /users and committers | | Closed |
| FE-1307 | Star in Changeset page breadcrumbs doesn't appear in Safari 3.2.1 | | Closed |
| FE-1306 | Changeset page should show longer extract of Crucible review title -- there's plenty of room | | Closed |
| FE-1305 | Changeset page doesn't show spinner while loading diffs | | Closed |
| FE-1296 | EyeQL results which include "group by" clause disconnect when serving second page | | Closed |
| FE-1294 | -> ACTION | | Closed |
| FE-1293 | -> ACTION | | Closed |
| FE-1291 | refactor hover popup (cru/jira) linker | | Closed |
| FE-1289 | turn global.js for fisheye into the jquery equivalent, and to split it up into proper modules. | | Closed |
| FE-1288 | New UI treatment for extra change set page features | | Closed |
| FE-1287 | UI Rework | | Closed |
| FE-1286 | NPE viewing user-committer manager mappings in admin | | Closed |
| FE-1285 | Integrate new ui for file history page | | Closed |
| FE-1279 | fisheye code pointers should highlight the line they are pointing to | | Closed |
| FE-1276 | Add permission checks to /fe/ ajax actions | | |

| | | | |
|---|---|---|---|
| | | | Closed |
| FE-1272 | consolidate scripts into a single place, into head tag. | | Closed |
| FE-1271 | remove inline event handlers, replace with event binds | | Closed |
| FE-1265 | make tree open at current path | | Closed |
| FE-1264 | Integrate resizable column layout with dirlist.jsp | | Closed |
| FE-1262 | Layout in Repository plugin page is broken | | Closed |
| FE-1259 | just do it | | Closed |
| FE-1258 | plugin finangling | | Closed |
| FE-1257 | View review blockers report | | Closed |
| FE-1256 | plugin finangling | | Closed |
| FE-1255 | most active developers / directories | | Closed |
| FE-1254 | punch card chart | | Closed |
| FE-1253 | expose data via API | | Closed |
| FE-1252 | View code metrics report | | Closed |
| FE-1251 | create reports page | | Closed |
| FE-1250 | create webitem + decorator | | Closed |
| FE-1249 | Reports tab | | Closed |
| FE-1247 | jsp wrangling | | Closed |
| FE-1246 | action | | Closed |
| FE-1245 | Add a project page | | Closed |
| FE-1244 | update admin page | | Closed |
| FE-1243 | db schema change to store proj info | | Closed |
| FE-1242 | Add a collection of "content roots" across one or more repositories. | | Closed |
| FE-1241 | refactor javascript - remove prototype | | Closed |
| FE-1240 | Javascript refactor | | Closed |

| | | | |
|---|---|---|---|
| FE-1239 | jsp | | Closed |
| FE-1238 | new action to handle fetching / sorting | | Closed |
| FE-1237 | lucene searcher | | Closed |
| FE-1236 | List of users at /users and committers at /committers/REPO | | Closed |
| FE-1235 | UI integration | | Closed |
| FE-1234 | render chart table thingy | | Closed |
| FE-1233 | searchy extractor thingy | | Closed |
| FE-1232 | Activity calendar | | Closed |
| FE-1231 | UI integration | | Closed |
| FE-1230 | render chart | | Closed |
| FE-1229 | implement searchy extractor thingy | | Closed |
| FE-1228 | Activity histograms commits vs hour of day and day of week | | Closed |
| FE-1227 | UI integration | | Closed |
| FE-1226 | render the chart | | Closed |
| FE-1225 | implement searchy extractor thingy | | Closed |
| FE-1224 | Recent activity sparkline & chart | | Closed |
| FE-1223 | just do it | | Closed |
| FE-1222 | Show personal review summary on user home page | | Closed |
| FE-1221 | just do it | | Closed |
| FE-1220 | Show open review count on user pages | | Closed |
| FE-1218 | jiralinkspan span is created inside anchor tags | | Closed |
| FE-1217 | simple impl | | Closed |
| FE-1216 | Ubiquitous (cross repo) quicksearch/nav | | Closed |
| FE-1212 | just do it | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1211 | repo dropdown in breadcrumb bar | | Closed |
| FE-1209 | build new JSP | | Closed |
| FE-1207 | Reload the "files pane" with ajax from a click in the "tree pane" and update the breadcrumbs | | Closed |
| FE-1206 | ajax remaining data | | Closed |
| FE-1204 | Use a full tree to navigate the new browse page | | Closed |
| FE-1196 | make action | | Closed |
| FE-1195 | make a tag / javascript function / standardise | | Closed |
| FE-1194 | Built in smart Crucible linker | | Closed |
| FE-1183 | make repository and crucible defaults for jira issues that are not mapped redux | | Closed |
| FE-1140 | Syntax Highlighting | | Closed |
| FE-1138 | Review design of CommitterUserMapping.hmb.xml | | Closed |
| FE-1136 | Add the ability to return unique results in EyeQL | | Closed |
| FE-1038 | svn-connection getting wrong values out of the repository configuration | | Closed |
| FE-962 | allow execution contexts to be added to quartz jobs / triggers | | Closed |
| FE-935 | Allow users to remove the "Email Review" button via the GUI or add tool tip it to make its use clearer. | | Closed |
| FE-928 | Repository with long names does not wrap correctly in folder view listing | | Closed |
| FE-914 | fix diff text caching | | Closed |
| FE-906 | REST auth-tokens should be instance-wide | | Closed |
| FE-904 | Display commits for a branched file before the trunk commits that were performed prior to the branch operation | | Closed |
| FE-854 | add smart filename search to quicksearch (including CamelCase initals) | | Closed |
| FE-851 | Show related JIRA issues in annotated file view | | Closed |
| FE-838 | FE: No hyperlink for copies expander | | Closed |
| FE-835 | minify & combine javascript | | Closed |
| FE-831 | p4 Issue to do with line endings causes logs to fill up with unexpected line errors | | Closed |
| FE-800 | Date Constraint query combined with checking empty directories takes forever to return | | Closed |

| | | | |
|---|---|---|---|
| FE-796 | 'reviews' EyeQL return clause should return all reviews that the revision is included in | | Closed |
| FE-792 | fix images for quicksearch dropdown | | Closed |
| FE-788 | Improve annotation Colours | | Closed |
| FE-787 | refactor quicksearch jsp | | Closed |
| FE-758 | Upgrade to Atlassian-Plugins 2.1 | | Closed |

**From 2.0 Beta2 to 2.0 Beta3**

**5th June 2009**

Full list of issues in this release:

| **JIRA Issues (200 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2379 | Viewing a very large raw file triggers an OutOfMemoryError | | Closed |
| FE-1865 | removing a favourite and then clicking on it again causes an error | | Closed |
| FE-1854 | Activity stream on project page doesn't filter by mapped JIRA projects | | Closed |
| FE-1847 | 'star' instead of 'favourite' in cog menu | | Closed |
| FE-1818 | Links from files in a changeset to the file history page don't include the revision | | Closed |
| FE-1815 | a con: XML namespace is being added to saved config.xml | | Closed |
| FE-1809 | Enhance README.html to include links to Evaluator Guide, Installation Guide, and Upgrade Guide | | Closed |
| FE-1808 | FishEye RC1 tarball contains Readme.HTML that links to Crucible docs | | Closed |
| FE-1804 | we shouldn't show an empty star before a search has been performed - they throw an exception when clicked | | Closed |
| FE-1802 | with the blue background on the comment form you can't tell which button is active when tabbing through them | | Closed |
| FE-1799 | Collapse all files in review doesn't work if there is a comment anchor in the URL | | Closed |
| FE-1784 | Soft wrapping always selected in prefs menu | | Closed |
| FE-1778 | admin project page doesn't render default and allowed reviewers | | Closed |
| FE-1771 | Commit by hour chart time axis numbering incorrect | | Closed |
| FE-1770 | Inconsistent and unsatisfactory highlighting colours on review page | | Closed |
| FE-1768 | defect label on comments has no css, but draft does | | Closed |

| | | | |
|---|---|---|---|
| FE-1760 | Infinite loop in CalculatedBucketGraphXY | | Closed |
| FE-1754 | Cross-repo QS repository membership and ordering changes between pages | | Closed |
| FE-1703 | Typo error | | Closed |
| FE-1698 | Improve LDAP Authentication so it does not Abandon initial bind or do an CMP | | Closed |
| FE-1627 | Add option to disable the indexing of diffs | | Closed |
| FE-1519 | Change include/exclude parameters for Restore | | Closed |
| FE-1515 | p4 calls labels command without supplying path when it needs to get the info on a particular label | | Closed |
| FE-1513 | Upgrade to latest AGSL-1 | | Closed |
| FE-1506 | Truncate number of lines in scrolling log | | Closed |
| FE-1505 | Restore should not fail if run from a different directory | | Closed |
| FE-1504 | Update FE System Requirements - remove IE 6 from support browsers | | Closed |
| FE-1500 | Let BackupManager store the job data in config.xml | | Closed |
| FE-1492 | Anna's misc M7 non-ui tasks | | Closed |
| FE-1489 | Add user timezone to send request by support so that all support requests have a valid timezone. | | Closed |
| FE-1485 | chart tweaks and fixes | | Closed |
| FE-1484 | Add directory tree to other pages | | Closed |
| FE-1483 | Enhance directory tree to display file | | Closed |
| FE-1479 | Make scale/chart nicer on code-metrics-report | | Closed |
| FE-1477 | Fix code-metrics-report | | Closed |
| FE-1473 | Make report plugin urls pretty | | Closed |
| FE-1472 | Fix 'revisions' tabulation on code-metrics report | | Closed |
| FE-1467 | Problem accessing Admin Remote API if the global anonymous access is off | | Closed |
| FE-1465 | Refactor report plugin packages | | Closed |
| FE-1464 | Report Mode Navigation (similar to Search Mode Navigation) | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1463 | Root (No Repository Context) Search Page | | Closed |
| FE-1455 | Date Pickers should be jQuery pickers (remove old script file when done) | | Closed |
| FE-1453 | Fix Structure (make search query controls in 'fixed/hard' section up top) | | Closed |
| FE-1452 | Remove blanks from Author box | | Closed |
| FE-1451 | Change "Search All Directories" to "Searching... <BREADCRUMB TRAIL>" | | Closed |
| FE-1449 | Remove .do from global quicksearch url | | Closed |
| FE-1448 | QuickSearch Improvements | | Closed |
| FE-1447 | Search Mode Navigation | | Closed |
| FE-1441 | Make revision comments appear on just one line in revision table | | Closed |
| FE-1435 | No progress indicator on dir tree twiddle | | Closed |
| FE-1427 | Improve group membership management | | Closed |
| FE-1418 | Combine summarize and close emails | | Closed |
| FE-1417 | CR-FE-1480 Rework | | Closed |
| FE-1415 | fix up pagination in users/ page | | Closed |
| FE-1414 | Fix it | | Closed |
| FE-1412 | Fix it | | Closed |
| FE-1411 | Fix QuickNav | | Closed |
| FE-1410 | View Changeset Stream Context Fixes | | Closed |
| FE-1408 | activity-streams plugin wrangling | | Closed |
| FE-1401 | add a comment as one of the return columns in /search/FE/ | | Closed |
| FE-1392 | Change mset messages displayed in browse history table should not be truncated (except via css) | | Closed |
| FE-1391 | quicksearch pages continuously displays progress indicator | | Closed |
| FE-1390 | Fix error handling when loading directory subtrees via ajax | | Closed |
| FE-1389 | Poor Javascript performance on file history page | | Closed |
| FE-1388 | can sort by checkboxes in browse view | | Closed |

| | | | |
|---|---|---|---|
| FE-1387 | bad title name in the config servlet for light svn plugin. | | Closed |
| FE-1386 | Make physical/logical link labels indicate change to different state, not the current state | | Closed |
| FE-1385 | Make tooltip content more descriptive | | Closed |
| FE-1383 | RSS improvements | | Closed |
| FE-1369 | Use Session Expires due to Jetty bug (JETTY-688) | | Closed |
| FE-1365 | Page listing only source files to browse | | Closed |
| FE-1361 | just do it | | Closed |
| FE-1360 | Chart jsp refactorings | | Closed |
| FE-1357 | write tag | | Closed |
| FE-1356 | Consistent time and age formats | | Closed |
| FE-1355 | update stream jsps to render a changeset | | Closed |
| FE-1354 | update all stream actions to take a csid | | Closed |
| FE-1353 | Maintain stream context when viewing a changeset page | | Closed |
| FE-1352 | just do it | | Closed |
| FE-1351 | preference and toggle to exclude own activity from home page stream | | Closed |
| FE-1349 | just do it | | Closed |
| FE-1348 | Switch all hover popups to use ajs.hover | | Closed |
| FE-1347 | just do it | | Closed |
| FE-1346 | Open settings pages in ajs.dialog | | Closed |
| FE-1345 | gravatar servlet | | Closed |
| FE-1344 | config screen + jsps | | Closed |
| FE-1343 | Avatar config screens | | Closed |
| FE-1342 | make the committer page group by user if there are multiple committer with same user | | Closed |
| FE-1341 | User & committer list rationalisation | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1340 | just do it | | Closed |
| FE-1339 | One jsp for global quicksearch and repo qsearch | | Closed |
| FE-1338 | just do it | | Closed |
| FE-1337 | Switch all dropdowns to ajs.dropdown | | Closed |
| FE-1336 | i.e. the old style changelog or what you get after you hit expand all | | Closed |
| FE-1335 | preference to show files rather than summary in actvity streams | | Closed |
| FE-1334 | legend unfucking | | Closed |
| FE-1333 | sparkline | | Closed |
| FE-1332 | line history sparklines and user line history chart improvements | | Closed |
| FE-1331 | just do it | | Closed |
| FE-1330 | Integrate new html for diff/annotation pages | | Closed |
| FE-1329 | just do it | | Closed |
| FE-1322 | this should be done in a single query for all committers you're interested in, not once for every user or every committer | | Closed |
| FE-1321 | you could use Math.max/min in a few places here | | Closed |
| FE-1320 | you shouldn't need to set the commiters as a parameter, just set the BreakdownOption in ParameterSetQuery to AUTHOR | | Closed |
| FE-1313 | there is no real point paging if we are going to do these types of calculations for every user in the system. It's just not going to work imho. Is sorting by num-reviews a must have? If so do one query that counts reviews for all users, grouped by u | | Closed |
| FE-1312 | change the comparator BY_RECENT_ACTIVITY to sort nulls last -- remove have NoActivityItem | | Closed |
| FE-1311 | use a better UI than ^ and V, see craig/pete | | Closed |
| FE-1310 | Rework from CR-FE-1441: FE-1236: sortable and paged list of users at /users and committers | | Closed |
| FE-1307 | Star in Changeset page breadcrumbs doesn't appear in Safari 3.2.1 | | Closed |
| FE-1306 | Changeset page should show longer extract of Crucible review title -- there's plenty of room | | Closed |
| FE-1305 | Changeset page doesn't show spinner while loading diffs | | Closed |
| FE-1296 | EyeQL results which include "group by" clause disconnect when serving second page | | Closed |
| FE-1294 | -> ACTION | | Closed |
| FE-1293 | -> ACTION | | |

| | | | |
|---|---|---|---|
| | | | Closed |
| FE-1291 | refactor hover popup (cru/jira) linker | | Closed |
| FE-1289 | turn global.js for fisheye into the jquery equivalent, and to split it up into proper modules. | | Closed |
| FE-1288 | New UI treatment for extra change set page features | | Closed |
| FE-1287 | UI Rework | | Closed |
| FE-1286 | NPE viewing user-committer manager mappings in admin | | Closed |
| FE-1285 | Integrate new ui for file history page | | Closed |
| FE-1279 | fisheye code pointers should highlight the line they are pointing to | | Closed |
| FE-1276 | Add permission checks to /fe/ ajax actions | | Closed |
| FE-1272 | consolidate scripts into a single place, into head tag. | | Closed |
| FE-1271 | remove inline event handlers, replace with event binds | | Closed |
| FE-1265 | make tree open at current path | | Closed |
| FE-1264 | Integrate resizable column layout with dirlist.jsp | | Closed |
| FE-1262 | Layout in Repository plugin page is broken | | Closed |
| FE-1259 | just do it | | Closed |
| FE-1258 | plugin finangling | | Closed |
| FE-1257 | View review blockers report | | Closed |
| FE-1256 | plugin finangling | | Closed |
| FE-1255 | most active developers / directories | | Closed |
| FE-1254 | punch card chart | | Closed |
| FE-1253 | expose data via API | | Closed |
| FE-1252 | View code metrics report | | Closed |
| FE-1251 | create reports page | | Closed |
| FE-1250 | create webitem + decorator | | Closed |
| FE-1249 | Reports tab | | Closed |

| | | | |
|---|---|---|---|
| FE-1247 | jsp wrangling | | Closed |
| FE-1246 | action | | Closed |
| FE-1245 | Add a project page | | Closed |
| FE-1244 | update admin page | | Closed |
| FE-1243 | db schema change to store proj info | | Closed |
| FE-1242 | Add a collection of "content roots" across one or more repositories. | | Closed |
| FE-1241 | refactor javascript - remove prototype | | Closed |
| FE-1240 | Javascript refactor | | Closed |
| FE-1239 | jsp | | Closed |
| FE-1238 | new action to handle fetching / sorting | | Closed |
| FE-1237 | lucene searcher | | Closed |
| FE-1236 | List of users at /users and committers at /committers/REPO | | Closed |
| FE-1235 | UI integration | | Closed |
| FE-1234 | render chart table thingy | | Closed |
| FE-1233 | searchy extractor thingy | | Closed |
| FE-1232 | Activity calendar | | Closed |
| FE-1231 | UI integration | | Closed |
| FE-1230 | render chart | | Closed |
| FE-1229 | implement searchy extractor thingy | | Closed |
| FE-1228 | Activity histograms commits vs hour of day and day of week | | Closed |
| FE-1227 | UI integration | | Closed |
| FE-1226 | render the chart | | Closed |
| FE-1225 | implement searchy extractor thingy | | Closed |
| FE-1224 | Recent activity sparkline & chart | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1223 | just do it | | Closed |
| FE-1222 | Show personal review summary on user home page | | Closed |
| FE-1221 | just do it | | Closed |
| FE-1220 | Show open review count on user pages | | Closed |
| FE-1218 | jiralinkspan span is created inside anchor tags | | Closed |
| FE-1217 | simple impl | | Closed |
| FE-1216 | Ubiquitous (cross repo) quicksearch/nav | | Closed |
| FE-1212 | just do it | | Closed |
| FE-1211 | repo dropdown in breadcrumb bar | | Closed |
| FE-1209 | build new JSP | | Closed |
| FE-1207 | Reload the "files pane" with ajax from a click in the "tree pane" and update the breadcrumbs | | Closed |
| FE-1206 | ajax remaining data | | Closed |
| FE-1204 | Use a full tree to navigate the new browse page | | Closed |
| FE-1196 | make action | | Closed |
| FE-1195 | make a tag / javascript function / standardise | | Closed |
| FE-1194 | Built in smart Crucible linker | | Closed |
| FE-1183 | make repository and crucible defaults for jira issues that are not mapped redux | | Closed |
| FE-1140 | Syntax Highlighting | | Closed |
| FE-1138 | Review design of CommitterUserMapping.hmb.xml | | Closed |
| FE-1136 | Add the ability to return unique results in EyeQL | | Closed |
| FE-1038 | svn-connection getting wrong values out of the repository configuration | | Closed |
| FE-962 | allow execution contexts to be added to quartz jobs / triggers | | Closed |
| FE-935 | Allow users to remove the "Email Review" button via the GUI or add tool tip it to make its use clearer. | | Closed |
| FE-928 | Repository with long names does not wrap correctly in folder view listing | | Closed |
| FE-914 | fix diff text caching | | Closed |

| | | | |
|---|---|---|---|
| FE-906 | REST auth-tokens should be instance-wide | | Closed |
| FE-904 | Display commits for a branched file before the trunk commits that were performed prior to the branch operation | | Closed |
| FE-854 | add smart filename search to quicksearch (including CamelCase initals) | | Closed |
| FE-851 | Show related JIRA issues in annotated file view | | Closed |
| FE-838 | FE: No hyperlink for copies expander | | Closed |
| FE-835 | minify & combine javascript | | Closed |
| FE-831 | p4 Issue to do with line endings causes logs to fill up with unexpected line errors | | Closed |
| FE-800 | Date Constraint query combined with checking empty directories takes forever to return | | Closed |
| FE-796 | 'reviews' EyeQL return clause should return all reviews that the revision is included in | | Closed |
| FE-792 | fix images for quicksearch dropdown | | Closed |
| FE-788 | Improve annotation Colours | | Closed |
| FE-787 | refactor quicksearch jsp | | Closed |
| FE-758 | Upgrade to Atlassian-Plugins 2.1 | | Closed |

**From 1.6.6 to 2.0 Beta**

Full list of issues in this release:

| **JIRA Issues (200 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2379 | Viewing a very large raw file triggers an OutOfMemoryError | | Closed |
| FE-1865 | removing a favourite and then clicking on it again causes an error | | Closed |
| FE-1854 | Activity stream on project page doesn't filter by mapped JIRA projects | | Closed |
| FE-1847 | 'star' instead of 'favourite' in cog menu | | Closed |
| FE-1818 | Links from files in a changeset to the file history page don't include the revision | | Closed |
| FE-1815 | a con: XML namespace is being added to saved config.xml | | Closed |
| FE-1809 | Enhance README.html to include links to Evaluator Guide, Installation Guide, and Upgrade Guide | | Closed |
| FE-1808 | FishEye RC1 tarball contains Readme.HTML that links to Crucible docs | | Closed |
| FE-1804 | we shouldn't show an empty star before a search has been performed - they throw an exception when clicked | | Closed |

| | | | |
|---|---|---|---|
| FE-1802 | with the blue background on the comment form you can't tell which button is active when tabbing through them | | Closed |
| FE-1799 | Collapse all files in review doesn't work if there is a comment anchor in the URL | | Closed |
| FE-1784 | Soft wrapping always selected in prefs menu | | Closed |
| FE-1778 | admin project page doesn't render default and allowed reviewers | | Closed |
| FE-1771 | Commit by hour chart time axis numbering incorrect | | Closed |
| FE-1770 | Inconsistent and unsatisfactory highlighting colours on review page | | Closed |
| FE-1768 | defect label on comments has no css, but draft does | | Closed |
| FE-1760 | Infinite loop in CalculatedBucketGraphXY | | Closed |
| FE-1754 | Cross-repo QS repository membership and ordering changes between pages | | Closed |
| FE-1703 | Typo error | | Closed |
| FE-1698 | Improve LDAP Authentication so it does not Abandon initial bind or do an CMP | | Closed |
| FE-1627 | Add option to disable the indexing of diffs | | Closed |
| FE-1519 | Change include/exclude parameters for Restore | | Closed |
| FE-1515 | p4 calls labels command without supplying path when it needs to get the info on a particular label | | Closed |
| FE-1513 | Upgrade to latest AGSL-1 | | Closed |
| FE-1506 | Truncate number of lines in scrolling log | | Closed |
| FE-1505 | Restore should not fail if run from a different directory | | Closed |
| FE-1504 | Update FE System Requirements - remove IE 6 from support browsers | | Closed |
| FE-1500 | Let BackupManager store the job data in config.xml | | Closed |
| FE-1492 | Anna's misc M7 non-ui tasks | | Closed |
| FE-1489 | Add user timezone to send request by support so that all support requests have a valid timezone. | | Closed |
| FE-1485 | chart tweaks and fixes | | Closed |
| FE-1484 | Add directory tree to other pages | | Closed |
| FE-1483 | Enhance directory tree to display file | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1479 | Make scale/chart nicer on code-metrics-report | | Closed |
| FE-1477 | Fix code-metrics-report | | Closed |
| FE-1473 | Make report plugin urls pretty | | Closed |
| FE-1472 | Fix 'revisions' tabulation on code-metrics report | | Closed |
| FE-1467 | Problem accessing Admin Remote API if the global anonymous access is off | | Closed |
| FE-1465 | Refactor report plugin packages | | Closed |
| FE-1464 | Report Mode Navigation (similar to Search Mode Navigation) | | Closed |
| FE-1463 | Root (No Repository Context) Search Page | | Closed |
| FE-1455 | Date Pickers should be jQuery pickers (remove old script file when done) | | Closed |
| FE-1453 | Fix Structure (make search query controls in 'fixed/hard' section up top) | | Closed |
| FE-1452 | Remove blanks from Author box | | Closed |
| FE-1451 | Change "Search All Directories" to "Searching... <BREADCRUMB TRAIL>" | | Closed |
| FE-1449 | Remove .do from global quicksearch url | | Closed |
| FE-1448 | QuickSearch Improvements | | Closed |
| FE-1447 | Search Mode Navigation | | Closed |
| FE-1441 | Make revision comments appear on just one line in revision table | | Closed |
| FE-1435 | No progress indicator on dir tree twiddle | | Closed |
| FE-1427 | Improve group membership management | | Closed |
| FE-1418 | Combine summarize and close emails | | Closed |
| FE-1417 | CR-FE-1480 Rework | | Closed |
| FE-1415 | fix up pagination in users/ page | | Closed |
| FE-1414 | Fix it | | Closed |
| FE-1412 | Fix it | | Closed |
| FE-1411 | Fix QuickNav | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1410 | View Changeset Stream Context Fixes | | Closed |
| FE-1408 | activity-streams plugin wrangling | | Closed |
| FE-1401 | add a comment as one of the return columns in /search/FE/ | | Closed |
| FE-1392 | Change mset messages displayed in browse history table should not be truncated (except via css) | | Closed |
| FE-1391 | quicksearch pages continuously displays progress indicator | | Closed |
| FE-1390 | Fix error handling when loading directory subtrees via ajax | | Closed |
| FE-1389 | Poor Javascript performance on file history page | | Closed |
| FE-1388 | can sort by checkboxes in browse view | | Closed |
| FE-1387 | bad title name in the config servlet for light svn plugin. | | Closed |
| FE-1386 | Make physical/logical link labels indicate change to different state, not the current state | | Closed |
| FE-1385 | Make tooltip content more descriptive | | Closed |
| FE-1383 | RSS improvements | | Closed |
| FE-1369 | Use Session Expires due to Jetty bug (JETTY-688) | | Closed |
| FE-1365 | Page listing only source files to browse | | Closed |
| FE-1361 | just do it | | Closed |
| FE-1360 | Chart jsp refactorings | | Closed |
| FE-1357 | write tag | | Closed |
| FE-1356 | Consistent time and age formats | | Closed |
| FE-1355 | update stream jsps to render a changeset | | Closed |
| FE-1354 | update all stream actions to take a csid | | Closed |
| FE-1353 | Maintain stream context when viewing a changeset page | | Closed |
| FE-1352 | just do it | | Closed |
| FE-1351 | preference and toggle to exclude own activity from home page stream | | Closed |
| FE-1349 | just do it | | Closed |
| FE-1348 | Switch all hover popups to use ajs.hover | | Closed |

| | | | |
|---|---|---|---|
| FE-1347 | just do it | | Closed |
| FE-1346 | Open settings pages in ajs.dialog | | Closed |
| FE-1345 | gravatar servlet | | Closed |
| FE-1344 | config screen + jsps | | Closed |
| FE-1343 | Avatar config screens | | Closed |
| FE-1342 | make the committer page group by user if there are multiple committer with same user | | Closed |
| FE-1341 | User & committer list rationalisation | | Closed |
| FE-1340 | just do it | | Closed |
| FE-1339 | One jsp for global quicksearch and repo qsearch | | Closed |
| FE-1338 | just do it | | Closed |
| FE-1337 | Switch all dropdowns to ajs.dropdown | | Closed |
| FE-1336 | i.e. the old style changelog or what you get after you hit expand all | | Closed |
| FE-1335 | preference to show files rather than summary in actvity streams | | Closed |
| FE-1334 | legend unfucking | | Closed |
| FE-1333 | sparkline | | Closed |
| FE-1332 | line history sparklines and user line history chart improvements | | Closed |
| FE-1331 | just do it | | Closed |
| FE-1330 | Integrate new html for diff/annotation pages | | Closed |
| FE-1329 | just do it | | Closed |
| FE-1322 | this should be done in a single query for all committers you're interested in, not once for every user or every committer | | Closed |
| FE-1321 | you could use Math.max/min in a few places here | | Closed |
| FE-1320 | you shouldn't need to set the commiters as a parameter, just set the BreakdownOption in ParameterSetQuery to AUTHOR | | Closed |
| FE-1313 | there is no real point paging if we are going to do these types of calculations for every user in the system. It's just not going to work imho. Is sorting by num-reviews a must have? If so do one query that counts reviews for all users, grouped by u | | Closed |
| FE-1312 | change the comparator BY_RECENT_ACTIVITY to sort nulls last -- remove have NoActivityItem | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1311 | use a better UI than ^ and V, see craig/pete | | Closed |
| FE-1310 | Rework from CR-FE-1441: FE-1236: sortable and paged list of users at /users and committers | | Closed |
| FE-1307 | Star in Changeset page breadcrumbs doesn't appear in Safari 3.2.1 | | Closed |
| FE-1306 | Changeset page should show longer extract of Crucible review title -- there's plenty of room | | Closed |
| FE-1305 | Changeset page doesn't show spinner while loading diffs | | Closed |
| FE-1296 | EyeQL results which include "group by" clause disconnect when serving second page | | Closed |
| FE-1294 | -> ACTION | | Closed |
| FE-1293 | -> ACTION | | Closed |
| FE-1291 | refactor hover popup (cru/jira) linker | | Closed |
| FE-1289 | turn global.js for fisheye into the jquery equivalent, and to split it up into proper modules. | | Closed |
| FE-1288 | New UI treatment for extra change set page features | | Closed |
| FE-1287 | UI Rework | | Closed |
| FE-1286 | NPE viewing user-committer manager mappings in admin | | Closed |
| FE-1285 | Integrate new ui for file history page | | Closed |
| FE-1279 | fisheye code pointers should highlight the line they are pointing to | | Closed |
| FE-1276 | Add permission checks to /fe/ ajax actions | | Closed |
| FE-1272 | consolidate scripts into a single place, into head tag. | | Closed |
| FE-1271 | remove inline event handlers, replace with event binds | | Closed |
| FE-1265 | make tree open at current path | | Closed |
| FE-1264 | Integrate resizable column layout with dirlist.jsp | | Closed |
| FE-1262 | Layout in Repository plugin page is broken | | Closed |
| FE-1259 | just do it | | Closed |
| FE-1258 | plugin finangling | | Closed |
| FE-1257 | View review blockers report | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1256 | plugin finangling | | Closed |
| FE-1255 | most active developers / directories | | Closed |
| FE-1254 | punch card chart | | Closed |
| FE-1253 | expose data via API | | Closed |
| FE-1252 | View code metrics report | | Closed |
| FE-1251 | create reports page | | Closed |
| FE-1250 | create webitem + decorator | | Closed |
| FE-1249 | Reports tab | | Closed |
| FE-1247 | jsp wrangling | | Closed |
| FE-1246 | action | | Closed |
| FE-1245 | Add a project page | | Closed |
| FE-1244 | update admin page | | Closed |
| FE-1243 | db schema change to store proj info | | Closed |
| FE-1242 | Add a collection of "content roots" across one or more repositories. | | Closed |
| FE-1241 | refactor javascript - remove prototype | | Closed |
| FE-1240 | Javascript refactor | | Closed |
| FE-1239 | jsp | | Closed |
| FE-1238 | new action to handle fetching / sorting | | Closed |
| FE-1237 | lucene searcher | | Closed |
| FE-1236 | List of users at /users and committers at /committers/REPO | | Closed |
| FE-1235 | UI integration | | Closed |
| FE-1234 | render chart table thingy | | Closed |
| FE-1233 | searchy extractor thingy | | Closed |
| FE-1232 | Activity calendar | | Closed |
| FE-1231 | UI integration | | Closed |

| | | | |
|---|---|---|---|
| FE-1230 | render chart | | Closed |
| FE-1229 | implement searchy extractor thingy | | Closed |
| FE-1228 | Activity histograms commits vs hour of day and day of week | | Closed |
| FE-1227 | UI integration | | Closed |
| FE-1226 | render the chart | | Closed |
| FE-1225 | implement searchy extractor thingy | | Closed |
| FE-1224 | Recent activity sparkline & chart | | Closed |
| FE-1223 | just do it | | Closed |
| FE-1222 | Show personal review summary on user home page | | Closed |
| FE-1221 | just do it | | Closed |
| FE-1220 | Show open review count on user pages | | Closed |
| FE-1218 | jiralinkspan span is created inside anchor tags | | Closed |
| FE-1217 | simple impl | | Closed |
| FE-1216 | Ubiquitous (cross repo) quicksearch/nav | | Closed |
| FE-1212 | just do it | | Closed |
| FE-1211 | repo dropdown in breadcrumb bar | | Closed |
| FE-1209 | build new JSP | | Closed |
| FE-1207 | Reload the "files pane" with ajax from a click in the "tree pane" and update the breadcrumbs | | Closed |
| FE-1206 | ajax remaining data | | Closed |
| FE-1204 | Use a full tree to navigate the new browse page | | Closed |
| FE-1196 | make action | | Closed |
| FE-1195 | make a tag / javascript function / standardise | | Closed |
| FE-1194 | Built in smart Crucible linker | | Closed |
| FE-1183 | make repository and crucible defaults for jira issues that are not mapped redux | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1140 | Syntax Highlighting | | Closed |
| FE-1138 | Review design of CommitterUserMapping.hmb.xml | | Closed |
| FE-1136 | Add the ability to return unique results in EyeQL | | Closed |
| FE-1038 | svn-connection getting wrong values out of the repository configuration | | Closed |
| FE-962 | allow execution contexts to be added to quartz jobs / triggers | | Closed |
| FE-935 | Allow users to remove the "Email Review" button via the GUI or add tool tip it to make its use clearer. | | Closed |
| FE-928 | Repository with long names does not wrap correctly in folder view listing | | Closed |
| FE-914 | fix diff text caching | | Closed |
| FE-906 | REST auth-tokens should be instance-wide | | Closed |
| FE-904 | Display commits for a branched file before the trunk commits that were performed prior to the branch operation | | Closed |
| FE-854 | add smart filename search to quicksearch (including CamelCase initals) | | Closed |
| FE-851 | Show related JIRA issues in annotated file view | | Closed |
| FE-838 | FE: No hyperlink for copies expander | | Closed |
| FE-835 | minify & combine javascript | | Closed |
| FE-831 | p4 Issue to do with line endings causes logs to fill up with unexpected line errors | | Closed |
| FE-800 | Date Constraint query combined with checking empty directories takes forever to return | | Closed |
| FE-796 | 'reviews' EyeQL return clause should return all reviews that the revision is included in | | Closed |
| FE-792 | fix images for quicksearch dropdown | | Closed |
| FE-788 | Improve annotation Colours | | Closed |
| FE-787 | refactor quicksearch jsp | | Closed |
| FE-758 | Upgrade to Atlassian-Plugins 2.1 | | Closed |

## 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*

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.

DOWNLOAD latest version

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. |

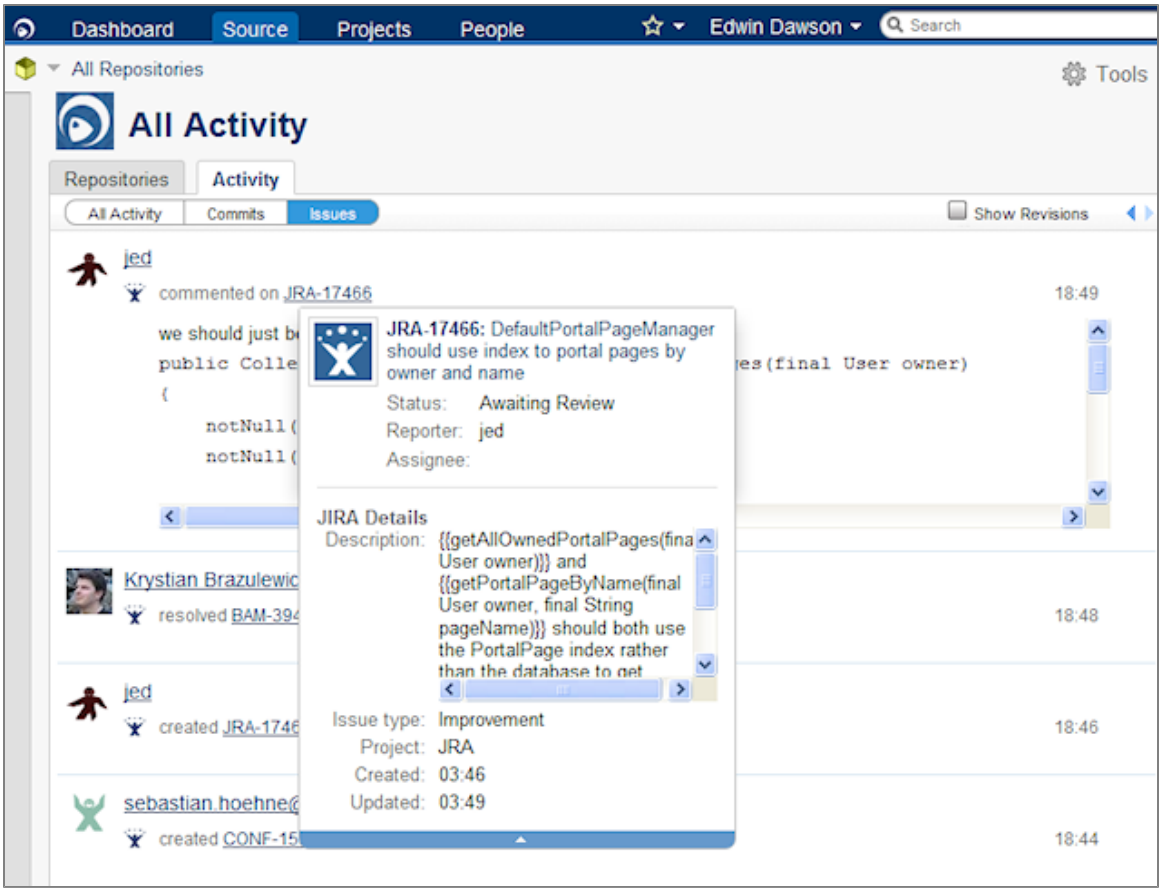
## Highlights of FishEye 2.0 Beta

1

### 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*

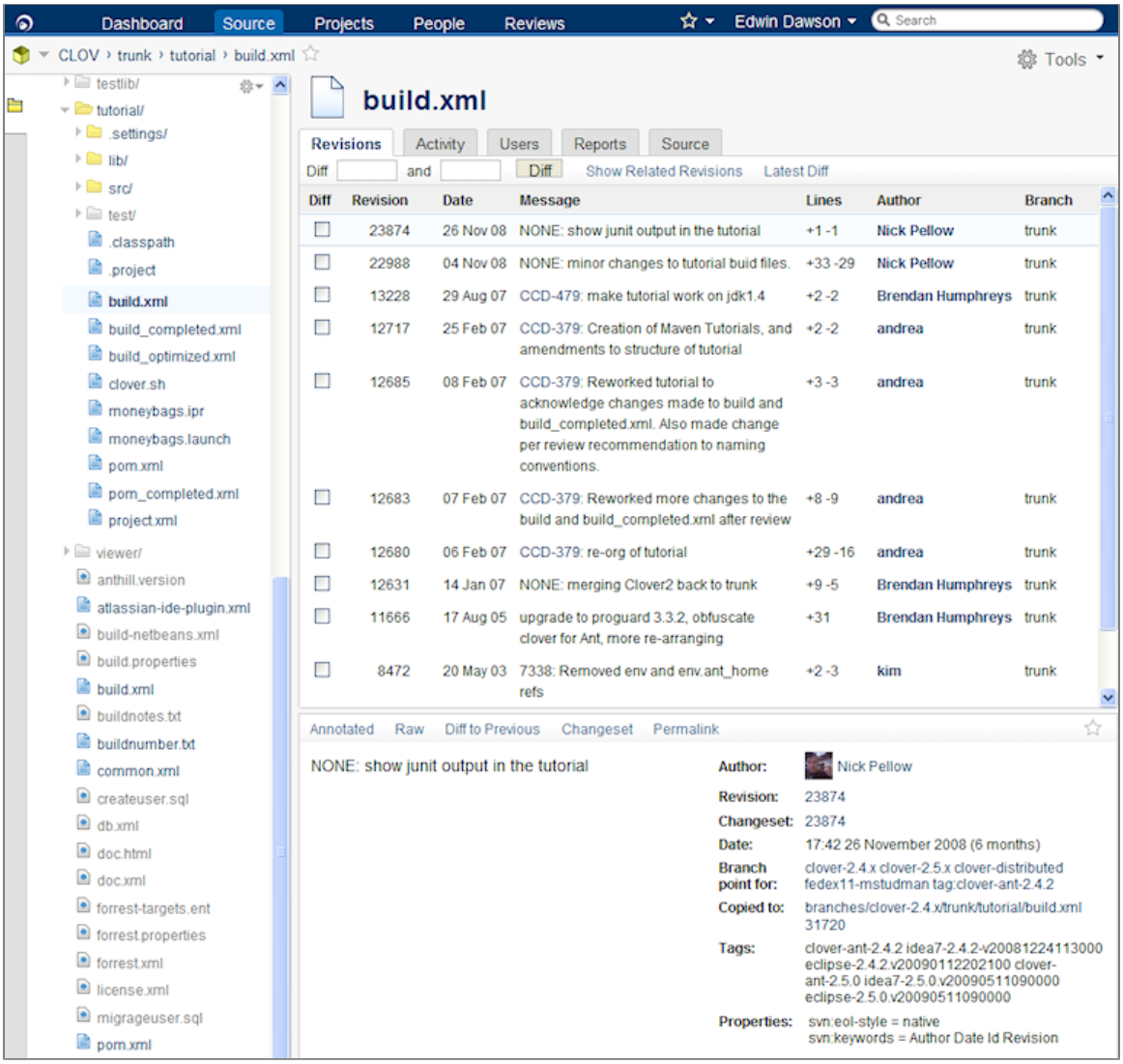



2

### 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*

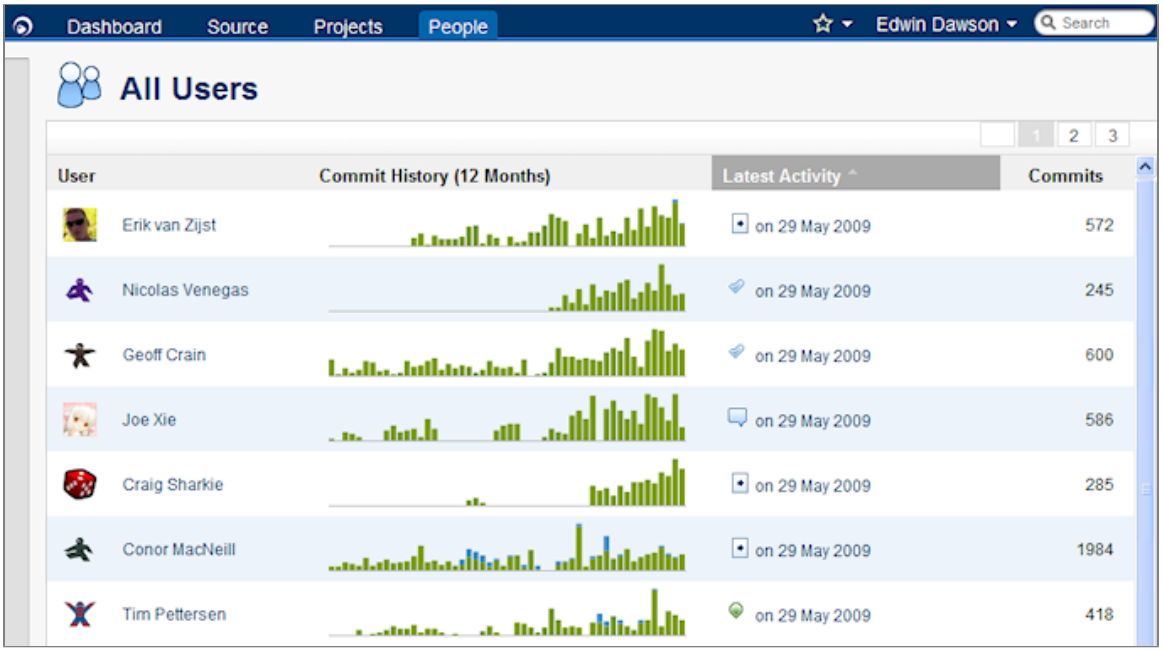



3

### 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 you 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*:



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

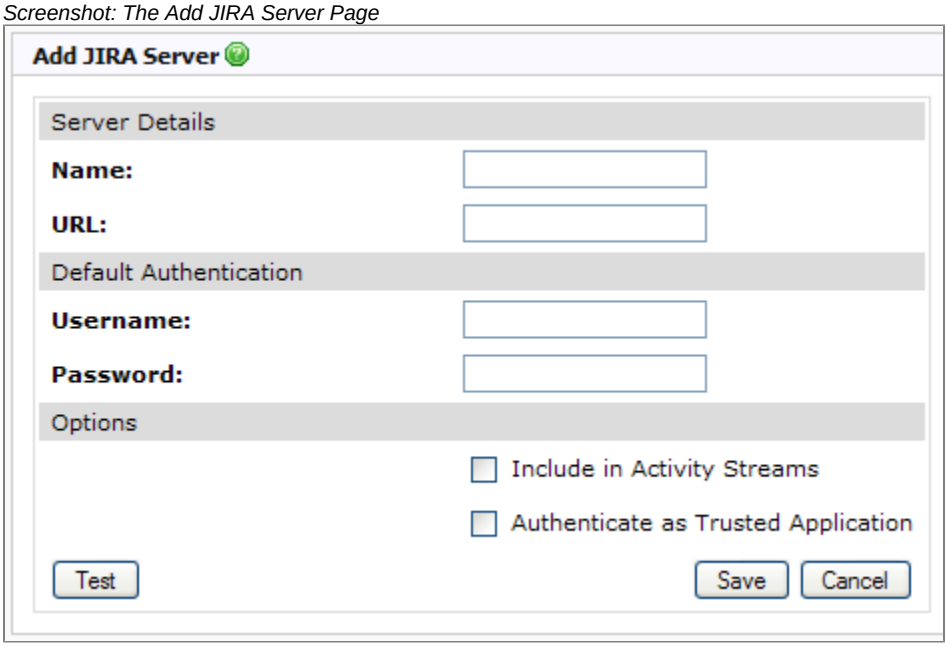
On the View JIRA Servers page, you can carry out a number of operations as listed on this page.

#### 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*




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 |

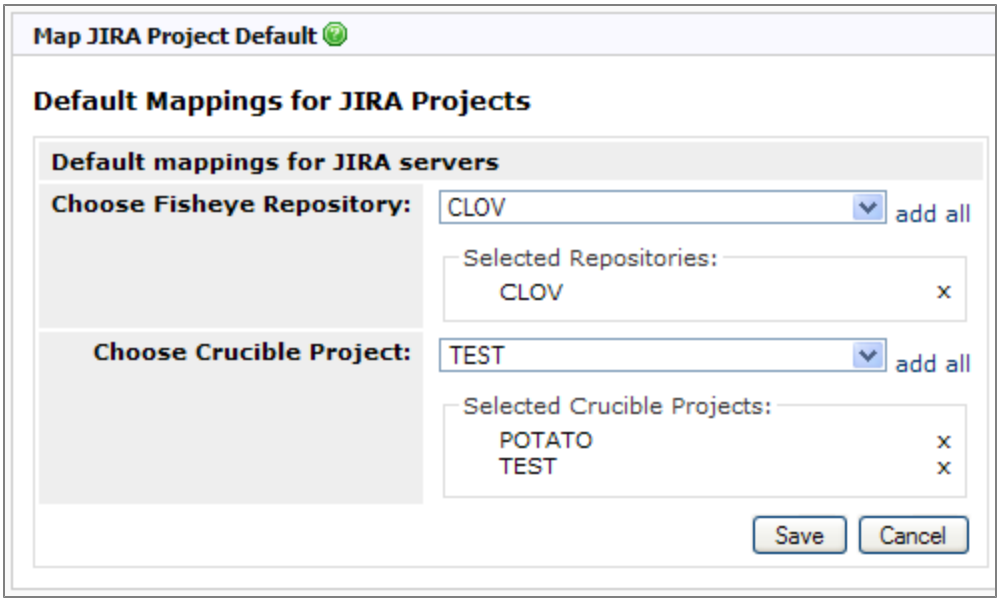
Once you've filled out the necessary fields, click '**Test**' to ensure that your details are correct. If you have a positive message return from the test, click '**Save**'.

#### Editing Default JIRA Server Mappings

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.

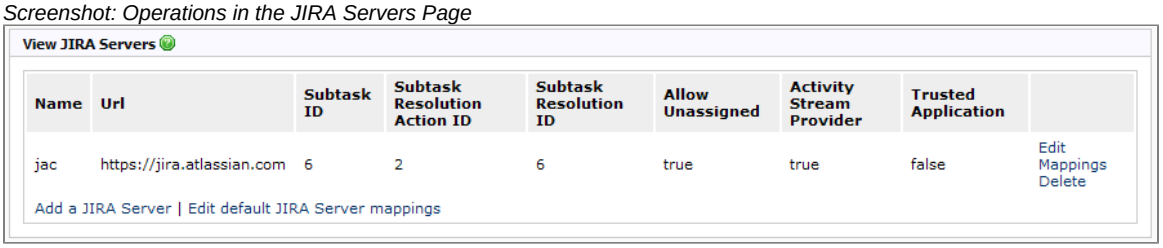
Once you've finished, click '**Save**'.

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*




##### *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:

1. There is no admin support yet (no UI controls for the feature).
2. There are missing features - notably author line count support.
3. Subsequent betas are likely to require re-indexing (although Git indexing is reasonably fast).
4. 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 2.5 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 adminstration 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
- Perforce Performance Tweaks
- Numerous improvements and bug-fixes

DOWNLOAD latest version

| 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 the date 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

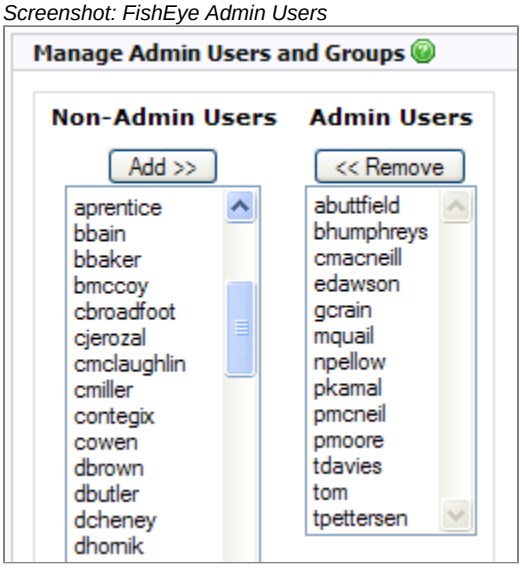
All of the improvements made to Quick Search and General Search are available when using Advanced Search and EyeQL.

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

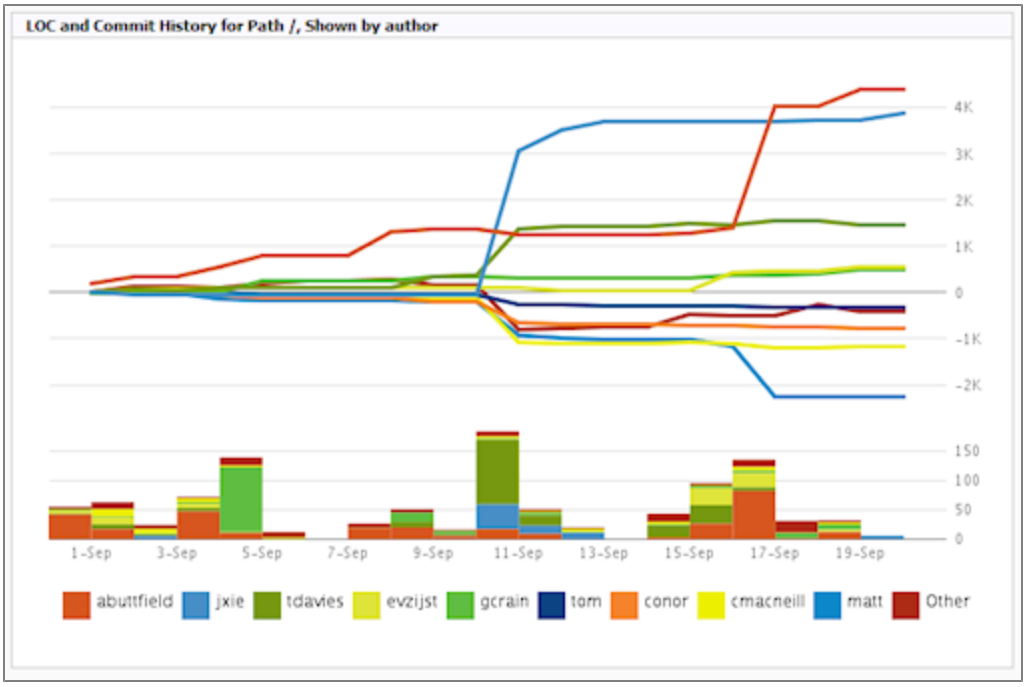
Enhancing the Remote API, the FishEye team have incorporated a **maxreturns** option, which lets you control the quantity of returned results. Also the range of functions is extended with the new **ancestor return** clause and **history search** items. Finally, Perforce data is now exposed in the remote 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

### Perforce Performance Tweaks

Perforce 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-483 | Ensure all user preferences are in user profile | | Closed |
| FE-494 | Add a link to the changeset on the annotation page | | Closed |
| FE-500 | Expose Changeset "Fixes Perforce Jobs" data in EyeQL (and REST api) | | Closed |
| FE-296 | Get Id button does not work in Trusted Application screen under IE | | Closed |
| FE-552 | Have an error page rather than 403 page when SVN permission denied | | Closed |
| FE-532 | upgrade to webwork 2.2.7 (fixes security problem) | | Closed |
| FE-617 | for new svn repositories, defaut for t/b/t should be None | | Closed |

| | | |
|---|---|---|
| FE-658 | Chart constraint dropped on second level subdir | Closed |
| FE-521 | CLONE -StackOverflow | Closed |
| FE-566 | diff-to-previous on annotate page 404s | Closed |
| FE-550 | improve catch-all svn symbolic regex | Closed |
| FE-393 | Use a single regularexpression to catch all tag/branch/trunk patterns | Closed |
| FE-554 | Add LIMIT clause to EyeQL documentation | Closed |
| FE-559 | Add 'ancestor' return clause to EyeQL | Closed |
| FE-578 | "Search just <repo:parh>" breadcrumb links are borked - escapes parameter separators | Closed |
| FE-536 | "List Repositories" method in the remote API documentation | Closed |
| FE-562 | resolve springsource/log4j versioning problem | 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-378 | review multithreading of RevCacheReader | Closed |
| FE-389 | only ask group to do group-membership tests for crowd users | Closed |
| FE-435 | Quick-search redo UI requirements | Closed |
| FE-434 | show match-in-context in quicksearch (hit highlighting) | Closed |
| FE-438 | Do content searches in quicksearch | Closed |
| FE-584 | Create the ability for customers to create support cases via Fisheye | Closed |
| FE-436 | Improved Quick Search | Closed |
| FE-301 | IndexOutOfBoundsException when opening annotated view | Closed |
| FE-503 | com.cenqua.fisheye.svn.SvnCache is throwing NPEs | Closed |
| FE-504 | command line reindex doens't work when loopback is not 127.0.0.1 | Closed |
| FE-674 | RSS Feed Title is missing space | Closed |
| FE-600 | TODO appearing in UI for comments | Closed |
| FE-511 | NPE when configuration file was not found | Closed |
| FE-668 | "Edit repository details" throws an NPE when a p4 repo has invalid info | Closed |

| | | | |
|---|---|---|---|
| FE-524 | upgrade to trusted apps 1.0, remove seraph dep | | Closed |
| FE-605 | added files appear as empty diffs | | Closed |
| FE-675 | RSS Feed Entries have almost no information in title | | Closed |
| FE-505 | Retrieve Password | | Closed |
| FE-437 | improve performance of filename searches in quicksearch | | Closed |
| FE-650 | Documentation: New 'Advanced' mode hides attributes in Add Repository screen | | Closed |
| FE-630 | Bundle SAL 1.1 in FishEye | | Closed |
| FE-540 | "Data Types and Structures" information in the Remote API page | | Closed |
| FE-479 | next and previous links on diff and annotation pages | | Closed |
| FE-555 | Documentation: Add maxReturn parameter to remote API | | Closed |
| FE-515 | Allow P4 label scanning to be skipped | | Closed |
| FE-82 | Re-index request: show message "Could not stop repository within 20 seconds. Re-index aborted." | | Closed |
| FE-288 | Repositories still occasionally get stuck in Stopping state | | Closed |
| FE-549 | Search tokenizes on underscores | | Closed |
| FE-338 | Please add ability to specify initial revision from which to begin initial scan | | Closed |
| FE-467 | Calculating the correct version for diffs doesn't work for perforce | | Closed |
| FE-541 | Allow limiting of number of results returned by remote API | | Closed |
| FE-636 | Make update polling configurable in admin section | | Closed |
| FE-491 | Hide optional fields in repo setup | | Closed |
| FE-495 | Self Signup layout borked | | Closed |
| FE-643 | "Request Garbage Collection" link on SysInfo/Support page redirects badly | | Closed |
| FE-649 | Cannot specify starting revision when creating perforce repo. The option only appears in the edit screen | | Closed |
| FE-651 | Weight quicksearch results by date | | Closed |
| FE-596 | CSS syntax highlighting omissions | | Closed |
| FE-513 | Redirected to javax.servlet.ServletException 500 error page when accessing to a disabled/stopped repository | | Closed |

| | | | |
|---|---|---|---|
| FE-319 | Can't access /admin/ when logged in via Crowd | | Closed |
| FE-607 | Create documentation for new feature, create support issues via fisheye Admin > Sysinfo screen | | Closed |
| FE-685 | Automatical user management from Crowd | | Closed |
| FE-687 | RSS Feed entry items truncate differently to 1.5.x | | Closed |
| FE-464 | upgrade to latest spring (at least 2.5.4) | | Closed |
| FE-428 | Check for updates option | | Closed |
| FE-482 | SVN repositories default to UTF-8 | | Closed |
| FE-487 | Remove the colon trailing repository names on the Fisheye home page | | Closed |
| FE-529 | Add an "Add Repository" link to the top of the repository list page in admin screen | | Closed |
| FE-678 | Link to file view in quicksearch results | | 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

**10th February 2009**

This release is a bugfix release which address the following issues:

| **JIRA Issues (19 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-1759 | Handle log message rewriting | | Closed |
| FE-1481 | Unable to open connection to Subverion version 1.6.1 | | Closed |
| FE-1141 | Print Stack Trace after exception in fisheye command line | | Closed |
| FE-1086 | Fisheye does not store the update time of p4 branches in the db, which means that upon a restart each branch is scanned for updates despite the fact it is not updated | | Closed |
| FE-1081 | illegal argument exception on all xwiki charts | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-1078 | EyeQL p4:jobsFixed return field produces invalid XML when used via REST | | Closed |
| FE-1046 | WrappedRequest does not delegate to super for all get* methods when the private params map is not null | | Closed |
| FE-845 | When viewing a changeset, perforce reports that there has been number of lines added/removed but "No Change" is report under the file rather than the diff. | | Closed |
| FE-583 | Chart the change in lines of code over a period | | Closed |
| FE-508 | A problem occurred starting your repository: Repository is not stopped (com.cenqua.fisheye.rep.RepositoryHandle$StateException) | | Closed |
| FE-499 | Broad search never returns results before timeout | | Closed |
| FE-493 | Trigger FishEye incremental scan via Email, or asychonous message queue'ing mechanism | | Closed |
| FE-473 | Unable to exclude a file | | Closed |
| FE-343 | Option for lines of code in all /trunk subdirectories | | Closed |
| FE-250 | What happens when SVN operation times out? | | Resolved |
| FE-194 | NPE when restarting repository | | Closed |
| FE-191 | FishEye update method: SVN hook script | | Closed |
| FE-127 | Sort repository lists on screens alphabetically | | Closed |
| FE-63 | Document custom content capability | | Closed |

**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:

| **JIRA Issues (22 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-1155 | Different Chinese Encoded Characters in Annotated view & Diff view | | Closed |
| FE-917 | Author constraints with usernames that have "\" fail | | Closed |
| FE-895 | Using request.getPathInfo() may not be accurate when servlet is called from <jsp:include> | | Closed |
| FE-894 | Generate Support Zip via the Administration console | | Closed |
| FE-888 | If SVNKit cannot authenticate successfully, it goes into a tight loop retrying | | Closed |

| | | | |
|---|---|---|---|
| FE-878 | Changing display name doesn't work | | Closed |
| FE-875 | document system.properties | | Closed |
| FE-873 | Documentation about Case Insensitive Perforce users | | Closed |
| FE-860 | add the ability to set the max HTTP header size via a system property | | Closed |
| FE-857 | Add doco for "path != word" | | Closed |
| FE-856 | Add "path = word" to eyeql | | Closed |
| FE-855 | Improve the performance of watch manager | | Closed |
| FE-852 | Diffs link shows up in HTML changeset emails for binary files | | Closed |
| FE-844 | Improve handling of Perforce encoding - especially Shift-JIS | | Closed |
| FE-841 | Files saved in another encoding beside UTF-8 (example shift-JIS) does not render properly in fisheye for perforce repositories. | | Closed |
| FE-840 | Email notification takes longer times to be sent | | Closed |
| FE-833 | Case of Sub Directories box doesn't match depot | | Closed |
| FE-830 | Need to update svnkit that resolves an NPE to do with NTLM authentication. | | Closed |
| FE-827 | Reading p4 annotate fails with "NumberFormatException" when customers annotate ouput does not being with <changeset>: | | Closed |
| FE-807 | Update the jsvn scripts to point to the new 1.2.1r5011.atlassian.jar version of svnkit | | Closed |
| FE-754 | Getting "Not Inserting Revision blah because a parent a parent directory is missing | | Closed |
| FE-624 | Indexing changesets which contain a large number of changes (e.g. brach copy, etc) can cause out of memory errors | | Closed |

**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 AtlasCamp 2008.

Full list of issues fixed in this release:

| **JIRA Issues (16 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-819 | Mail log doesn't show SMTP transcript in debug mode | | Closed |
| FE-808 | Sub Directories Ajax not refreshing | | Closed |
| FE-802 | FishEye no longer correctly detects \ characters in url constraints | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-801 | follow up js memory problems | | Closed |
| FE-799 | Rendering takes a long time when there are large number of files to display and these files have a number of revisions | | Closed |
| FE-795 | Please don't show: "You have 29 days before your maintenance expires ..." | | Closed |
| FE-791 | allow quicksearch to look for specific types of results | | Closed |
| FE-782 | Review use of classes copied mainly from lucene | | Closed |
| FE-727 | "Cannot open file: it does not appear to be a valid archive" error thrown for tarball downloads made in IE | | Closed |
| FE-661 | Spruce up Add Repository screen | | Closed |
| FE-538 | Deleting crowd-based user while the user is logged in results in user account being re-created on logout | | Closed |
| FE-537 | Deleting user while the user is logged in results in NPE when user tries to access his profile | | Closed |
| FE-361 | Sexy chart type switching | | Closed |
| FE-241 | Same time commits wrongly ordered in changelog | | Closed |
| FE-151 | FishEye continues to return 304 NOT MODIFIED for pages even after user logs out completely | | Closed |
| FE-145 | Browse page for trunk directory shows that a file is deleted, when deletion was only on branches | | Closed |

**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:

| **JIRA Issues (32 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-786 | escape urls in quicksearch | | Closed |
| FE-785 | Exception displayed in Crucible revisions - Not enough revisions to diff | | Closed |
| FE-777 | Document access to LOC data via JSON | | Closed |
| FE-775 | file search box not working | | Closed |
| FE-774 | quick search with csid presents broken HTML in results | | Closed |
| FE-768 | rework from CR-FE-678 CRUC-658 | | Closed |

| | | | |
|---|---|---|---|
| FE-749 | [admin] missing "Synchronise users with Crowd" on view section | | Closed |
| FE-747 | Add event system to FishEye | | Closed |
| FE-741 | Ognl will set the context to some value such as "ognl.OgnlContext@649af4c6" when you EditServerSettings and the context was originally empty | | Closed |
| FE-730 | implement CVS diff search | | Closed |
| FE-729 | FishEye should ship out-of-the-box JIRA linkers | | Closed |
| FE-726 | Include fisheye.out as part of the files that are sent via Raise Support Request in Administration > SysInfo/Support | | Closed |
| FE-723 | quicksearch jump to exact file match | | Closed |
| FE-721 | update Atlassian licensing to version 2.0 | | Closed |
| FE-714 | Don't enable starttls by default | | Closed |
| FE-711 | Some of the seach functionality we utilised in Fisheye no longer works in version 1.6.1 | | Closed |
| FE-710 | Provide Fisheye url for given file - remote API | | Closed |
| FE-709 | Cannot view a stopped repository | | Closed |
| FE-708 | highlighting in keywords should be done manually | | Closed |
| FE-707 | Strange svn blame issue | | Closed |
| FE-706 | Misc quicksearch improvements | | Closed |
| FE-703 | Quick search: respect quotes | | Closed |
| FE-697 | When there are lots of branches, not all path results are being returned | | Closed |
| FE-694 | Logout results in 500 Server error | | Closed |
| FE-689 | Delete group doesnt work, even with no users in it | | Closed |
| FE-672 | A certain set of steps in svn will cause the error "Not inserting revision path/to/file@9 because a parent directory is missing" | | Closed |
| FE-551 | Add support for local SVN 1.5 repositories using pure java client | | Closed |
| FE-457 | Syntax Highlighting is wrong (keywords match within identifiers - redux) | | Closed |
| FE-419 | loc data shoud be available as raw data | | Closed |
| FE-305 | stdcxx indexing problems(s) | | Closed |
| | | | |

| | | | |
|---|---|---|---|
| FE-267 | Document setting up a CVS mirror with rsync | | Closed |
| FE-240 | "1 more file in changeset" is useless | | Closed |

**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*

- 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 2.5 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

DOWNLOAD latest version

| Upgrading to FishEye 1.5 |
| --- |
| You can now download FishEye from here. If upgrading from a previous version, please follow the Upgrade Guide. |

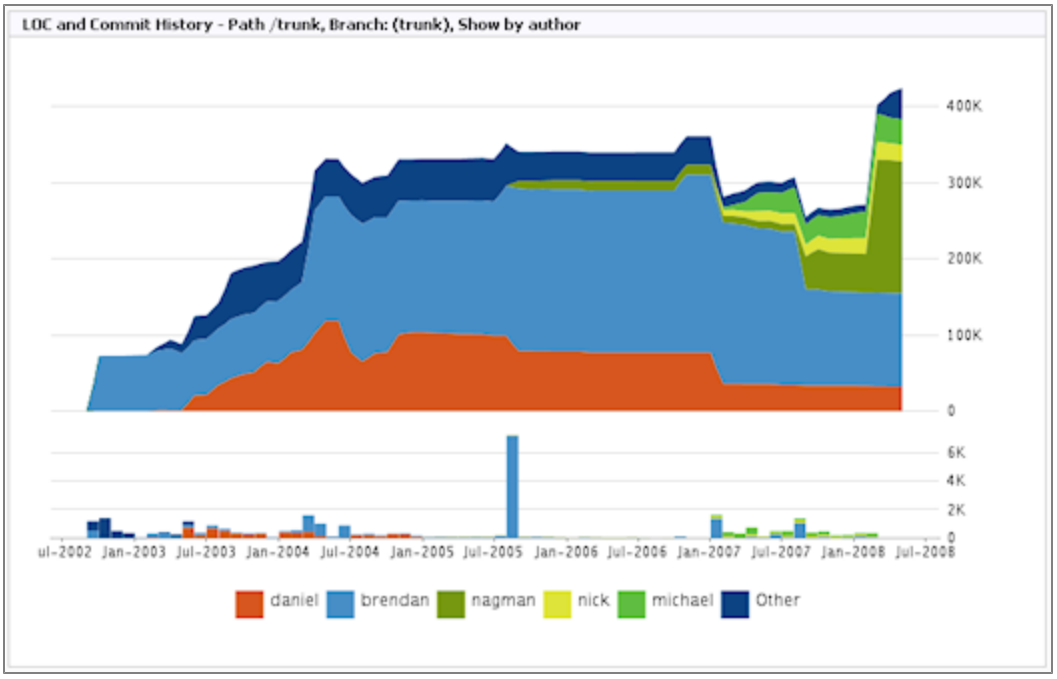
## Highlights of FishEye 1.5

### 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*



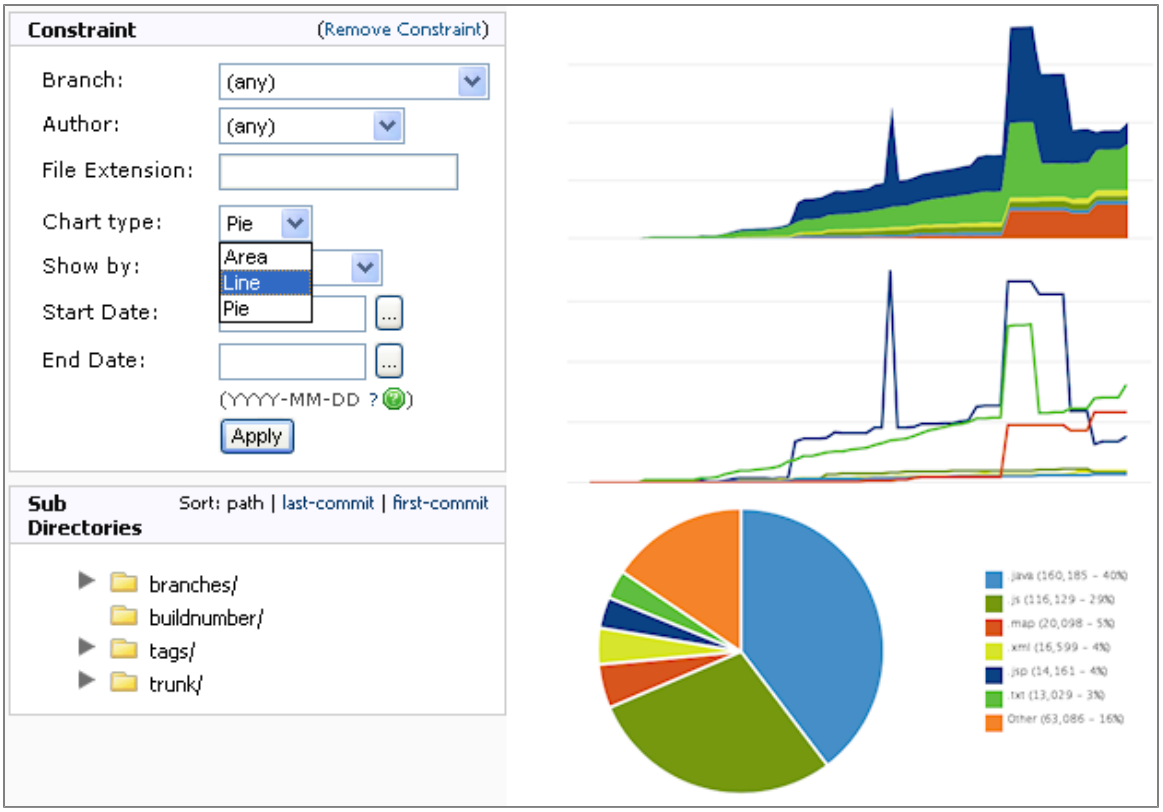

2

### 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

| JIRA Issues (56 issues) | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-2507 | Manually request incremental scan from commandline | | Closed |
| FE-2506 | CLONE -FishEye occasionally does not get the author and/or comment for change sets | | Closed |
| FE-394 | Can't open page on Multithreading without giving credentials | | Closed |
| FE-382 | Unable to index repository due to: org.tigris.subversion.javahl.ClientException: svn: PROPFIND of '/path': 403 Forbidden (https://subversion..company.com) | | Closed |
| FE-358 | Improve determination of first revision in an SVN repo | | Closed |
| FE-344 | Improving the chart page in fisheye | | Closed |

| | | | |
|---|---|---|---|
| FE-336 | Random order of operations on Trusted Applications | | Closed |
| FE-335 | Default Certificate Timeout value for Trusted Applications should not be 0 | | Closed |
| FE-333 | WARN - error parsing file with regexp | | Closed |
| FE-331 | User credentials are case-sensitive | | Closed |
| FE-326 | FishEye always sorts with oldest first | | Closed |
| FE-325 | Add syntax highlighting for ActionScript (.as) files | | Closed |
| FE-323 | Manually request incremental scan from commandline | | Closed |
| FE-321 | FishEye occasionally does not get the author and/or comment for change sets | | Closed |
| FE-320 | Showing files as directories in tags directories | | Closed |
| FE-317 | Improve email notification handling for commit comments with newlines | | Closed |
| FE-312 | Unable to display Japanese Character "Mojibake" under UTF-8 encoding | | Closed |
| FE-307 | LOC data should respect repository case sensitivity | | Closed |
| FE-303 | fix single-sign-off problem with Crowd | | Closed |
| FE-302 | Upgrade crowd support to 1.3 | | Closed |
| FE-299 | Upgrade Seraph to 0.36 | | Closed |
| FE-298 | Upgrade Seraph to 0.36 | | Closed |
| FE-295 | Ability to view full source code when creating a patch review | | Closed |
| FE-290 | FishEye/Crucible not correct supporting unlimited-user licenses | | Closed |
| FE-287 | Replace EDU.oswego concurrency classes with java,util.concurrent | | Closed |
| FE-282 | P4 Files of type "unicode" appear as binary | | Closed |
| FE-278 | Cannot edit or delete Trusted Application | | Closed |
| FE-273 | Upgrade to Cenqua Licensing 1.6 | | Closed |
| FE-269 | Editing repository details does not allways ends with a "you need to restart repository..." message | | Closed |
| FE-265 | Include appropriate licence/notice files | | Closed |
| FE-264 | When Crowd integration is enabled, Trusted Application requests should use the Crowd Db when determining if users exist | | Closed |

| | | | |
|---|---|---|---|
| FE-263 | XML-RPC calls generate responses non-conforming to XML-RPC spec | | Closed |
| FE-258 | Upgrade to seraph 0.36 when released | | Closed |
| FE-242 | autoadd login with crowd and max users creates spinning browser | | Closed |
| FE-237 | StackOverflow | | Closed |
| FE-235 | Don't "Index Content" on every server restart | | Closed |
| FE-234 | Add REST API docs to Confluence | | Closed |
| FE-229 | Duplicate LDAP users created with differing case | | Closed |
| FE-224 | Handle dependencies with Maven 2 | | Closed |
| FE-218 | ensure this NPE doesn't crash the watch mechanism | | Closed |
| FE-210 | doc: tag names in FE, eyeql help | | Closed |
| FE-209 | 404 page instead of diff view | | Closed |
| FE-200 | In Search Results, don't list every page | | Closed |
| FE-197 | Create a pom.xml for Fisheye/Crucible | | Closed |
| FE-185 | Add web resource plugin module support | | Closed |
| FE-182 | Lots of StackOverflow Errors in Log | | Closed |
| FE-181 | Multithread initial repository scan too | | Closed |
| FE-172 | Email feed unsubscribe & default format | | Closed |
| FE-170 | capitalisation on 'User Profile' page is a little inconsistent | | Closed |
| FE-164 | Fisheye does not seem to recognize mac os line ending \r when displaying diffs | | Closed |
| FE-155 | Documentation unclear on interaction of include/exclude & tag/branch configuration | | Closed |
| FE-139 | Please add the ability to watch a single file | | Closed |
| FE-122 | Allow Repositories to be indexed in parallel. | | Closed |
| FE-116 | FishEye is returning a bad response for a particular annotated file | | Closed |
| FE-62 | Admin screens for custom homepage and footer content | | Closed |
| FE-5 | [P4] need to consider "utf8" etc file types | | Closed |

# 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**

**1 August 2008**

This release contains minor improvements and bug fixes.

Errors were reported by the JIRA trusted connection.

- APP_UNKNOWN; Unknown Application: {0}; ["confluence:4557196"]

| JIRA Issues (21 issues) | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-662 | Customization of GUI | | Resolved |
| FE-593 | Certain Input Files causes excessive memory usage in Syntax Highlighting | | Closed |
| FE-579 | Directories mistakingly get indexed as files | | Closed |
| FE-577 | Force stop of a DiffFetcher thread in case of exceptions | | Closed |
| FE-571 | Username & groups that contain characters such as '&' could cause problems when viewing such groups/users via the UI due to the fact that the urls are not encoded. | | Closed |
| FE-568 | ArrayIndexOutOfBoundsException in CalculatedBucketGraphXY | | Closed |
| FE-565 | In system information include java version and os information | | Closed |
| FE-561 | Allow admins to turn on debugging via the admin UI | | Closed |
| FE-560 | Watches still not deleted | | Closed |
| FE-558 | Stop Fisheye repository does not work | | Closed |
| FE-557 | Crowd's crowd-integration-client-1.4.4.jar is not compatible with Fisheye 1.5.x | | Closed |
| FE-546 | -300,000 LOC?!?! | | Closed |
| FE-542 | force-lowercase option does not always appear in admin screens | | Closed |
| FE-517 | java.lang.ArrayIndexOutOfBoundsException in ChangelogServlet-doImageMap | | Closed |
| FE-475 | "java.lang.IncompatibleClassChangeError" with Crowd's client 1.4 in Fisheye/Crucible | | Closed |

| | | | |
|---|---|---|---|
| FE-474 | FishEye to handle/skip obliterated changelist in Perforce | | Closed |
| FE-472 | Customized display settings set back to default values after relogin | | Closed |
| FE-402 | Using scannow command or Scan Now button should trigger scanning ASAP | | Closed |
| FE-306 | To handle SVN operation hang during FishEye scanning | | Closed |
| FE-280 | Still see watch email log messages for deleted users (adds users back too!) | | Closed |
| FE-215 | Combined Author/Age Histogram labels are broken | | Closed |

**From 1.5.2 to 1.5.3**

### 23 June 2008

This release contains bug fixes.

| **JIRA Issues (2 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-512 | FishEye may stop sending emails after a backup | | Closed |
| FE-478 | "Bad format for response:jira" error when attempting to get ID from JIRA 3.12.3 | | Closed |

**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.

| **JIRA Issues (16 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-485 | update doco re p4:jobid regex | | Closed |
| FE-481 | Support regex p4:jobid eyeql search clause | | Closed |
| FE-476 | author blame error on svn file replace | | Closed |
| FE-465 | [mvn] upload jarjar 1.0rc7 into 3rdparty | | Closed |
| FE-455 | Charting Colors Broken | | Closed |
| FE-441 | Watches are not being deleted | | Closed |
| FE-439 | Author info/Store diff info Warning when disabled | | Closed |
| FE-431 | Email watch notification does not properly parse the checkin comments for links (to jira/cru for example) | | Closed |
| FE-429 | DownloadableClasspathResource passes null content type to GzipFilter | | Closed |
| FE-383 | Linecount graph calculation performance improvements | | Closed |
| FE-374 | number render bug in blame legends | | Closed |

| | | | |
|---|---|---|---|
| FE-359 | Date constraint should do more than clip | | Closed |
| FE-285 | Fisheye fails to search for files of a type | | Closed |
| FE-252 | Problem getting diff information from subversion repos | | Closed |
| FE-223 | Problem getting diff information for rev2232 | | Closed |
| FE-130 | Stop script hit error after set FISHEYE_ARGS | | Closed |

**From 1.5.0 to 1.5.1**

**24 April 2008**

This release contains bug fixes.

| **JIRA Issues (21 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-421 | update doco wrt to "enable/disable SSO" in crowd | | Closed |
| FE-415 | upgrade svnkit to 1.1.7 | | Closed |
| FE-413 | [crowd] logged out after 2 minutes when using another "incompatible" crowd app with FishEye/Crucible | | Closed |
| FE-412 | Add ability to turn off SSO in FishEye/Crucible's crowd support | | Closed |
| FE-409 | "Using the fisheye screens" doc page out of date | | Closed |
| FE-408 | in subdir breakdown charts the ". (this dir)" category is not self-explanatory | | Closed |
| FE-406 | "Files in Dir" entry in subdir chart | | Closed |
| FE-404 | Setting diff style to side-by-side in profile doesn't stick | | Closed |
| FE-403 | incrementalIndexThreads and initialIndexThreads are incorrectly stored in config.xml | | Closed |
| FE-392 | debug logging overly verbose in 1.5 | | Closed |
| FE-390 | User display preference setting in profile not saved | | Closed |
| FE-386 | Investigate UI Preference Behaviour | | Closed |
| FE-376 | Old charting code can probably be removed now | | Closed |
| FE-372 | charting title is wrong | | Closed |
| FE-369 | clicking (eg) Changelog from Chart tab loses directory constraint | | Closed |
| FE-360 | Small Line History Chart Tweaks | | Closed |
| FE-357 | Create an admin interface to edit the username force-lowercase configuration in security | | Closed |
| FE-350 | admin screen for force-lowercase | | Closed |
| FE-346 | static content is not being gz encoded | | Closed |
| FE-277 | Using "Test path" button on "Add repository" page saves the data and closes the form | | Closed |
| FE-26 | FishEye Quick Start Guide | | Closed |

## 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*

- 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 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 2.5 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. |

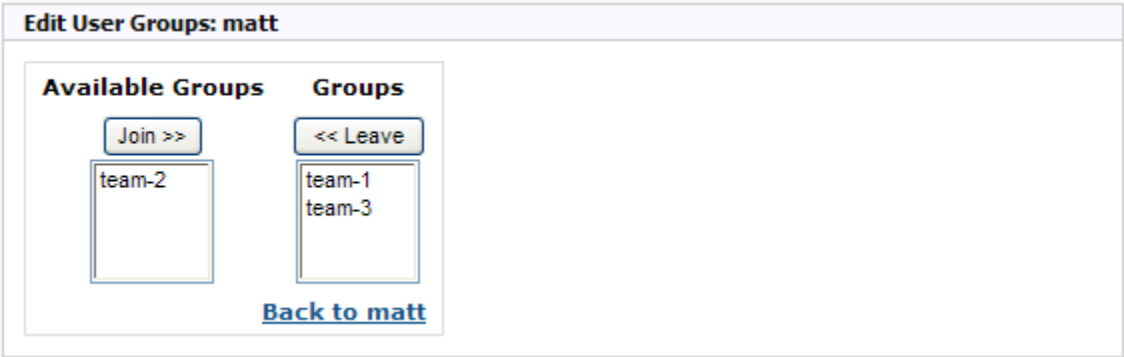
## Highlights of FishEye 1.4

1

### Enhancements to user management

In FishEye 1.3.7, we introduced support for public signup (self-registration). Now in release 1.4:

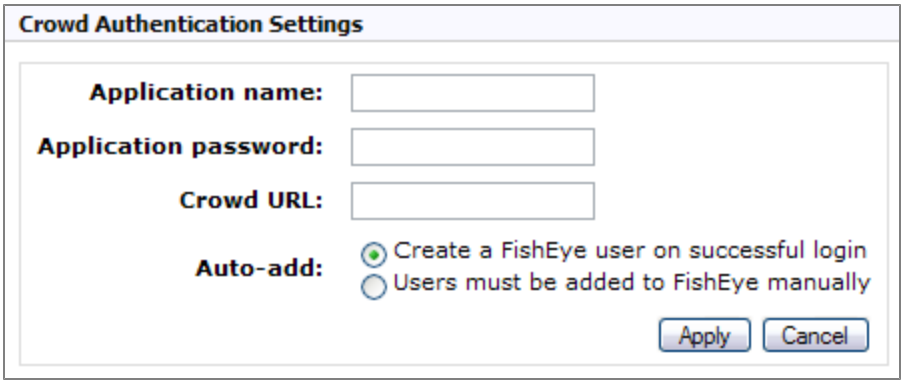
- Improved user interface makes user administration easier.
- Groups are supported.
- Renaming of users is supported.
- Read the documentation.




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.

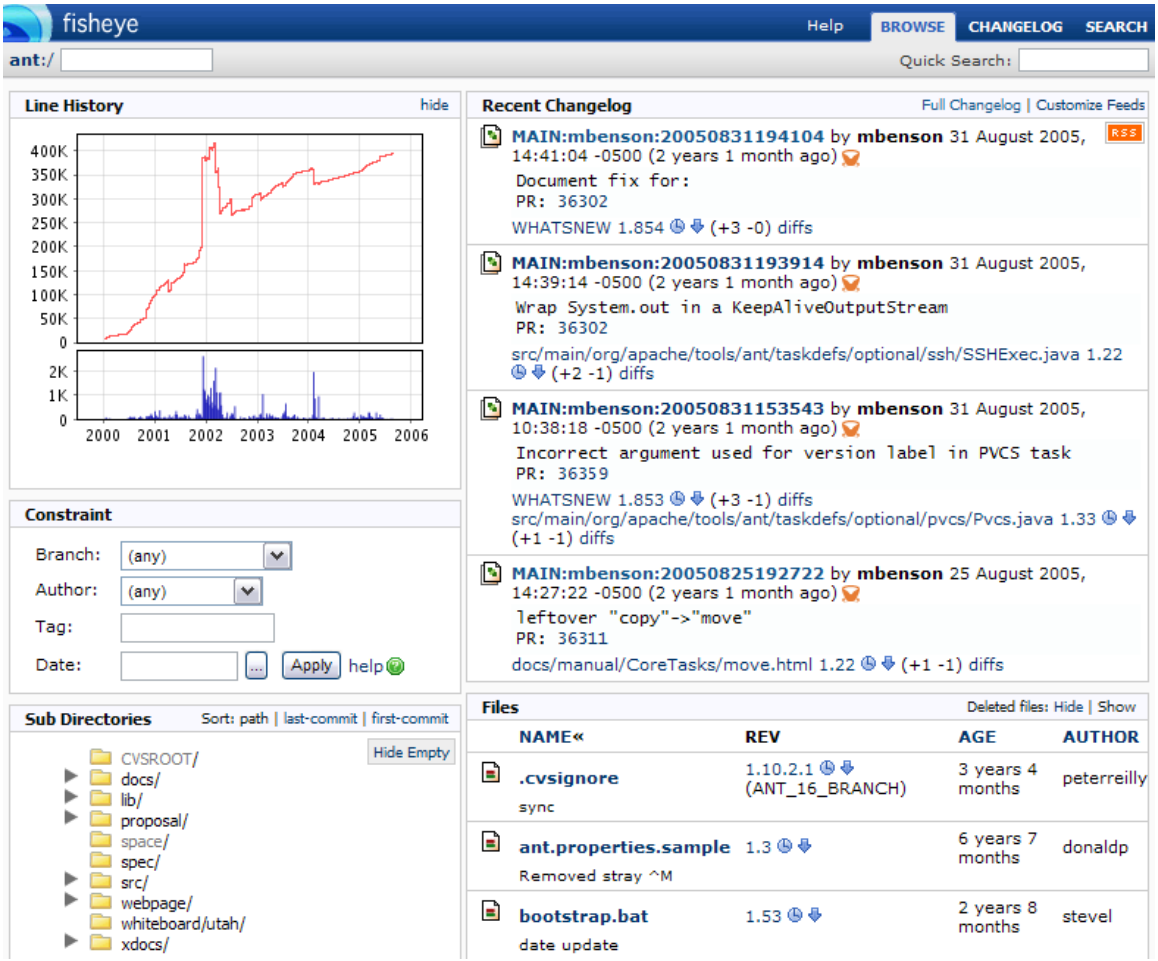



3

### 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 improvments:

- 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

| **JIRA Issues (40 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-163 | DbException: Problem getting diff information for rev1 | | Closed |
| FE-156 | Bug/feature request link at bottom of screens is wrong | | Closed |
| FE-149 | eyeql textbox too big in safari3.0.4 | | Closed |
| FE-148 | Ability to delete user groups | | Closed |
| FE-147 | Subversion revision indexing fails | | Closed |
| FE-146 | Ugrade to new version of yahoo library | | Closed |
| FE-141 | Hit NPE when trying to add new user (built-in) | | Closed |
| FE-138 | Upgrade atlassian-extras dependency to 1.10 | | Closed |
| FE-136 | suggestion: in the FishEye Admin menu, consider change 'Misc' to 'System Administration' | | Closed |
| FE-133 | Cascading documentation links within FishEye | | Closed |
| FE-132 | Report FishEye and/or API version via the API | | Closed |
| FE-128 | Make online help link to CAC documentation | | Closed |
| FE-126 | UI preferences (showing/hiding graph, directory sort order) don't work if user isn't logged in | | Closed |
| FE-121 | Excessively long debug and error log entry when using AJP auth, automatic fisheye user creation, and exceeding license limit. | | Closed |
| FE-120 | NPE when using ajp for authentication | | Closed |
| FE-119 | constraint in email watches can become corrupted by url escaping | | Closed |

| | | | |
|---|---|---|---|
| FE-118 | per-rep linkers don't necessarily trump default linkers when they match the same string | | Closed |
| FE-114 | Regex syntax highlighting StackOverflowError | | Closed |
| FE-111 | FishEye should ignore FISHEYE_HOME variable | | Closed |
| FE-108 | Syntax Highlighting is wrong (keywords match within identifiers) | | Closed |
| FE-103 | ability to rename users | | Closed |
| FE-100 | allow usernames to contain the @ character | | Closed |
| FE-99 | Linker Update requires a restarty | | Closed |
| FE-98 | groups-of-users support | | Closed |
| FE-97 | Subdirectories don't show up in sort orders other than "path" | | Closed |
| FE-93 | Change FishEye's default port | | Closed |
| FE-92 | Imporve PHP syntax highlighting (was WARN - error parsing file with regexp ) | | Closed |
| FE-86 | Default linkers don't warn that restart is required | | Closed |
| FE-81 | Make email optional for self-registered users | | Closed |
| FE-73 | Fisheye javadoc needs to be uploaded to docs.a.c | | Closed |
| FE-72 | Document "File History View Mode" | | Closed |
| FE-71 | Incorrect spelling 'Seach' on Simple Search screen | | Closed |
| FE-70 | Allow Crowd/SSO credentials to be used in the remote API | | Closed |
| FE-68 | Fisheye needs to support Crowd SSO | | Closed |
| FE-58 | RESTful admin api | | Closed |
| FE-57 | Add 'reindex repository' command to FishEyeCtl interface | | Closed |
| FE-54 | Diff colours too dark & noisy | | Closed |
| FE-51 | add listTagsForRevision() to remote api | | Closed |
| FE-21 | Index & expose P4 job information via remote API | | Closed |
| FE-1 | Branch dropdown breaks page layout when branch names are massive | | Closed |

## 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**

**7 February 2008**

This release contains bug fixes.

| **JIRA Issues (35 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-309 | Trusted Application not stored properly in configuration file | | Closed |
| FE-304 | Improve documentation on recommended hardware and software (JVM) settings for FishEye | | Closed |
| FE-284 | login error and logout pages return blank page | | Closed |
| FE-272 | Linker regex application is case-sensitive | | Closed |
| FE-266 | Support protocols such as pserver for remote CVS repositories | | Closed |
| FE-262 | trusted app admin screen doesn't support https:// urls | | Closed |
| FE-259 | certificateTimeout isn't saved to config.xml | | Closed |
| FE-253 | Adding large repository causes all other repositories to stop indexing | | Closed |
| FE-249 | Cancelling SVN Operation due to timeout: what operation? | | Closed |
| FE-247 | <properties> in config.xml not passed to custom authenticators | | Closed |
| FE-225 | List public FishEye instances in FishEye documentation | | Closed |
| FE-217 | Linecount graphs give incorrect results on antlr perforce database | | Closed |
| FE-211 | Linkers don't work in fisheye | | Closed |
| FE-208 | Linkers setting in Repository Defaults ignored | | Closed |
| FE-205 | CLONE -Custom Authenticator's init method is being passed an empty Properties object | | Closed |
| FE-195 | tmp folder fills up disk space rapidly | | Closed |
| FE-186 | Connect remote CVS | | Closed |
| FE-169 | Username is shown instead of Display Name under AJPv13 | | Closed |
| FE-162 | whenever i start fisheye my net stops working, i could not open other web pages except fisheye | | Closed |
| FE-124 | Subscribe to changelog RSS using permissions does not work as expected | | Closed |
| FE-83 | Update links etc to new FishEye docs on CAC | | Closed |
| FE-75 | Review FishEye docs on CAC | | Closed |
| FE-74 | Export the XML, PDF and HTML versions and upload to ALLDOCS space | | Closed |
| FE-60 | Ability to customize Fisheye welcome message | | Closed |
| FE-36 | New Diff UI (in 1.3.5) Is A Step Down | | Closed |
| FE-35 | apply new left-nav to Bamboo and Clover spaces | | Closed |

| | | | |
|---|---|---|---|
| FE-33 | Point online help links to new FishEye doc space | | Closed |
| FE-30 | Left-hand nav panel needs adaptation to Confluence 2.6 styles | | Closed |
| FE-28 | Move ' Improve FishEye Scan Performance' guide to FishEye permanent documentation space | | Closed |
| FE-27 | Switch to Atlassian Logos | | Closed |
| FE-25 | Move Fisheye docs to Confluence | | Closed |
| FE-22 | After Fisheye forums are moved, update all links in docs | | Closed |
| FE-18 | Allow FishEye to run inside Tomcat | | Closed |
| FE-14 | Woke up to stack overflow in fisheye | | Closed |
| FE-2 | I installed latest Crucible and fisheye.access = md5:... appeared to public frontend | | Closed |

**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.

| **JIRA Issues (7 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-207 | upgrade JFree chart | | Closed |
| FE-203 | NPE in api when calling getRevision() on a tag directory | | Closed |
| FE-193 | Confirm recent svnkit/javahl still works with subersion server 1.1.x | | Closed |
| FE-187 | clarify how groups are associated with repositories | | Closed |
| FE-177 | Add Application Trust Capability to Fisheye and Crucible | | Closed |
| FE-171 | please update help-paths.properties to accommodate some page-renumbering | | Closed |
| FE-55 | File has empty history in FishEye | | Closed |

**From 1.4 to 1.4.1**

This is a small bug-fix release.

| **JIRA Issues (1 issues)** | | | |
|---|---|---|---|
| **Key** | **Summary** | **Priority** | **Status** |
| FE-178 | CursorLengthException when indexing P4 | | Closed |

## 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 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 2.5 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

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*

- 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 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 1.3 Release Notes
FishEye 1.3 Changelog

## FishEye Release Summary

### *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
- Perforce 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.

# Security Advisories

This page lists security advisories for FishEye.



## 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

Atlassian rates the severity level of these vulnerabilities as **high**, according to the scale published in Severity Levels for Security Issues. The scale allows us to rank the severity as critical, high, moderate or low.

#### Risk Assessment

We have identified and fixed a number of cross-site scripting (XSS) vulnerabilities which may affect FishEye/Crucible instances, including publicly available instances (that is, internet-facing servers). XSS vulnerabilities potentially allow an attacker to embed their own JavaScript into a FishEye/Crucible page. You can read more about XSS attacks at cgisecurity.com, The Web Application Security Consortium and other places on the web.

#### Vulnerability

The table below describes the FishEye/Crucible versions and the specific functionality affected by the XSS vulnerabilities.

| FishEye/Crucible Feature | Affected FishEye/Crucible Versions | Issue Tracking |
|---|---|---|
| Crucible snippets | Crucible 2.4.5 to 2.5.0 | CRUC-5734 |
| Crucible author mapping | Crucible 2.4.5 to 2.5.0 | CRUC-5735 |
| Crucible changeset comments in search results | Crucible 2.3.0 to 2.5.0 | CRUC-5736 |
| Crucible comments search | Crucible 2.2.6 to 2.5.0 | CRUC-5737 |
| FishEye/Crucible dashboard - review activity | FishEye/Crucible 2.2.8 to 2.5.2 | FE-3031 |
| FishEye/Crucible reviews list | FishEye/Crucible 2.2.8 to 2.5.2 | FE-3032 |

#### Risk Mitigation

We recommend that you upgrade your FishEye/Crucible installation to fix these vulnerabilities.

Alternatively, if you are not in a position to upgrade immediately and you judge it necessary, you can disable public signup to your instance until you have applied the upgrade. For even tighter control, you could restrict access to trusted groups.

#### Fix

FishEye/Crucible 2.5.4 fixes all of these issues. View the issues linked above for information on earlier fix versions for each issue. For a full description of this release, see the FishEye 2.5 changelog and Crucible 2.5 Changelog. You can download the latest version of FishEye/Crucible from the download centre (FishEye download centre, Crucible download centre).

There are no patches available to fix these vulnerabilities. You must upgrade your FishEye/Crucible installation.

Our thanks to **Marian Ventuneac** of http://www.ventuneac.net, who reported FE-3031 and FE-3032. We fully support the reporting of vulnerabilities and we appreciate it when people work with us to identify and solve the problem.

## FishEye and Crucible Security Advisory 2011-01-12

This advisory announces a number of security vulnerabilities that we have found and fixed in recent versions of FishEye/Crucible. You need to upgrade your existing FishEye/Crucible installations to fix these vulnerabilities. Enterprise Hosted customers should request an upgrade by filing a ticket at http://support.atlassian.com. JIRA Studio is not vulnerable to any of the issues described in this advisory.

Atlassian is committed to improving product security. The vulnerabilities listed in this advisory have been discovered by Atlassian, unless noted otherwise. The reporter may also have requested that we do not credit them.

If you have questions or concerns regarding this advisory, please raise a support request at http://support.atlassian.com/.

**In this advisory:**



### *XSS Vulnerabilities*

#### Severity

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 2010-05-04

**In this advisory:**

## *Admin Escalation Vulnerability*

#### Severity

Atlassian rates this vulnerability as **critical**, according to the scale published in Severity Levels for Security Issues. The scale allows us to rank a vulnerability as critical, high, moderate or low.

#### Risk Assessment

We have identified and fixed an admin escalation vulnerability, which affects FishEye instances. This vulnerability has security implications and is especially important for anyone running publicly accessible instances of FishEye.

#### Vulnerability

This vulnerability allows a motivated attacker to perform admin actions.

All versions of FishEye from version 1.6.0-beta2 (including 1.6.0) through to 2.2.1 are affected by these admin escalation vulnerabilities.

| Affected FishEye Versions | Fix Availability | More Details | Severity |
|---|---|---|---|
| All versions up to and including 2.2.1 | 2.2.3 update, also available as patches for certain versions, listed on this page. | This vulnerability allows a motivated attacker to perform admin actions. | Critical |

#### Risk Mitigation

We strongly recommend either upgrading or patching your FishEye installation to fix this vulnerability. Please see the 'Fix' section below.

> Note: If you are an Atlassian JIRA Studio customer, we have assessed that your system is secure and implemented additional protections for it.

#### Fix

These issues have been fixed in FishEye 2.2.3 (see the changelog), which you can download from the download centre. Later versions will include protection from this vulnerability.

This fix is also provided as a patch for FishEye 2.1.4, 2.0.6 and 1.6.6, which you can download from this page. Customers on earlier point versions of FishEye will have to upgrade to version 2.1.4, 2.0.6 or 1.6.6 before applying the patch. We recommend you upgrade to FishEye 2.2.3.

## *XSS Vulnerabilities in FishEye*

#### Severity

Atlassian rates these vulnerabilities as **critical**, according to the scale published in Severity Levels for Security Issues. The scale allows us to rank a vulnerability as critical, high, moderate or low.

#### Risk Assessment

We have identified and fixed several cross-site scripting (XSS) vulnerabilities in FishEye, which may affect FishEye instances. These vulnerabilities have security implications and are especially important for anyone running publicly accessible instances of FishEye.

- The attacker might take advantage of the vulnerability to steal other users' session cookies or other credentials, by sending the credentials back to the attacker's own web server.
- The attacker's text and script might be displayed to other people viewing a FishEye page. This is potentially damaging to your company's reputation.

You can read more about XSS attacks at cgisecurity, CERT and other places on the web.

#### Vulnerability

All versions of FishEye are affected by these XSS vulnerabilities.

| Affected FishEye Versions | Fix Availability | More Details | Severity |
|---|---|---|---|
| All versions up to and including 2.2.1 | 2.2.3 only | An attacker could take advantage of this vulnerability to steal other users' session cookies or other credentials, or the attacker's text and script might be displayed to other people viewing a FishEye page. | Critical |

#### Risk Mitigation

We strongly recommend upgrading your FishEye installation to fix these vulnerabilities. Please see the 'Fix' section below.

#### Fix

These issues have been fixed in FishEye 2.2.3 (see the changelog), which you can download from the download centre.

### *Prevention of Brute Force Attacks*

#### Severity

Atlassian rates this vulnerability as **moderate**, according to the scale published in Severity Levels for Security Issues.

#### Risk Assessment

We have improved the security of the following areas in FishEye:

- Prevention of brute force attacks by requiring users to solve a CAPTCHA test after a maximum number of repeated login attempts.

#### Vulnerability

We have identified and fixed a problem where FishEye allows an unlimited number of repeated login attempts, potentially opening FishEye to a brute force attack. Details of this improvement are summarised below.

| Affected FishEye Versions | Fix Availability | More Details | Severity |
|---|---|---|---|
| All versions up to and including 2.2.1 | 2.2.3 only | FishEye allows an unlimited number of login attempts. This makes FishEye vulnerable to a brute force attack. | Moderate |

#### Risk Mitigation

We recommend that you upgrade your FishEye installation to fix these vulnerabilities. Please see the 'fix' section below.

You can also prevent brute force attacks by following our guidelines on using Fail2Ban to limit login attempts.

#### Fix

This issue has been fixed in FishEye 2.2.3 (see the changelog). Later versions will include protection from this vulnerability. You can download FishEye 2.2.3 from the download centre.

### *Changed Behaviour in FishEye*

In order to fix these issues, we have changed FishEye's behaviour as follows:

- After three consecutive failed login attempts, FishEye will display a CAPTCHA form asking the user to enter a given word when attempting to log in again. This will prevent brute force attacks via the login screen. The number of failed attempts needed to trigger the CAPTCHA testing is configurable. For more information, see the documentation for Brute Force Login Protection.

> In addition, after three consecutive failed login attempts via the FishEye remote API, an error message will be returned. Human intervention will then be required to reset that login account, i.e. solve the CAPTCHA test via the login screen.

### *Download Patches for Earlier FishEye / Crucible Versions*

> These patch releases contain security fixes, which apply to the shared FishEye architecture that is the basis of both FishEye and Crucible.

These patches fix the Admin Escalation vulnerability only. Please note that these patches are for specific older point versions of FishEye (2.1.4, 2.0.6 or 1.6.6). If you are running an earlier version than these, you will need to upgrade to a version specifically addressed by one of these patches. To update a more recent version of the product (2.1.5 through 2.2.1), please upgrade to FishEye 2.2.3 or later. Atlassian strongly recommends that you upgrade to FishEye 2.2.3 or later.

MD5 checksums are provided to allow verification of the downloaded files.

**Patch for FishEye / Crucible 2.1.4**

| File | FishEye / Crucible Version | Release Date | MD5 Checksum |
|---|---|---|---|
| fisheye-crucible-2.1.4-patch1.zip | 2.1.4 | 4th May, 2010 | 6062fa2e1ad93729527357fb97b0d2ea |

**Patch for FishEye / Crucible 2.0.6**

| File | FishEye / Crucible Version | Release Date | MD5 Checksum |
|---|---|---|---|
| fisheye-crucible-2.0.6-patch1.zip | 2.0.6 | 4th May, 2010 | 6aae75e2a5308121887bf9532473cf75 |

**Patch for FishEye 1.6.6**

| File | FishEye Version | Release Date | MD5 Checksum |
|---|---|---|---|
| fisheye-1.6.6-patch1.zip | 1.6.6 | 4th May, 2010 | 210ef3358aff83861733f8f22d331d7e |

**Patch for Crucible 1.6.6**

| File | Crucible Version | Release Date | MD5 Checksum |
|---|---|---|---|
| crucible-1.6.6-patch1.zip | 1.6.6 | 4th May, 2010 | 48e8e8ada0ddb3fc8671459051df1120 |

To acquire all of the fixes on this page, upgrade to FishEye 2.2.3, which you can download from the download centre.

## FishEye Security Advisory 2010-06-16

**In this advisory:**

## *Remote Code Exploit Vulnerability*

#### Severity

Atlassian rates this vulnerability as **critical**, according to the scale published in Severity Levels for Security Issues. The scale allows us to rank a vulnerability as critical, high, moderate or low.

#### Risk Assessment

We have identified and fixed a remote code exploit vulnerability which affects FishEye and Crucible instances.

#### Vulnerability

This vulnerability allows a motivated attacker to call remote code on the host server.

All versions of FishEye/Crucible up to version 2.3.2 are affected by this vulnerability.

| Affected FishEye Versions | Fix Availability | More Details | Severity |
|---|---|---|---|
| All versions up to and including 2.3.2. | 2.3.3 update, also available as patches for 2.3.2 and 2.2.3. | This vulnerability allows a motivated attacker to call remote code on the host server. | Critical |

This vulnerability has been discovered in XWork by OpenSymphony, a command pattern framework which is used by FishEye and Crucible.

**About the XWork Framework**:

- See the OpenSymphony XWork page for more information about XWork.

#### Risk Mitigation

We strongly recommend either upgrading or patching your FishEye/Crucible installation to fix this vulnerability. Please see the 'Fix' section below.

#### Fix

These issues have been fixed in FishEye 2.3.3 (see the changelog), which you can download from the download centre.

It has also been fixed in Crucible 2.3.3 (see the changelog), which you can download from the download centre.

Later versions will include protection from this vulnerability.

This fix is also provided as a patch for FishEye/Crucible 2.3.2 and 2.2.3, which you can download from links on this page. Customers on earlier point versions of FishEye/Crucible will have to upgrade to version 2.3.2 or 2.2.3 before applying the patch. Atlassian recommends you upgrade to FishEye/Crucible 2.3.3.

## *Download Patches for Earlier FishEye / Crucible Versions*

> These patch releases contain security fixes, which apply to the shared FishEye architecture that is the basis of both FishEye and Crucible.

Please note that these patches are for specific point versions of FishEye (2.3.2 and 2.2.3). If you are running an earlier version than these, you will need to upgrade to a version specifically addressed by one of these patches. Atlassian strongly recommends that you upgrade to FishEye 2.3.3 / Crucible 2.3.3 or later.

MD5 checksums are provided to allow verification of the downloaded files.

#### Patch for FishEye / Crucible 2.3.2

---

| File | FishEye / Crucible Version | Release Date | MD5 Checksum |
|---|---|---|---|
| fisheye-crucible-2.3.2-patch1.zip | 2.3.2 | 16th June, 2010 | 6fe98db821a6d26f26907688af2ccd84 |

**Patch for FishEye / Crucible 2.2.3**

| File | FishEye / Crucible Version | Release Date | MD5 Checksum |
|---|---|---|---|
| fisheye-crucible-2.2.3-patch1.zip | 2.2.3 | 16th June, 2010 | 6fe98db821a6d26f26907688af2ccd84 |

Our thanks to **Meder Kydyraliev** of the **Google Security Team** who discovered this vulnerability. Atlassian fully supports the reporting of vulnerabilities and appreciates it when people work with Atlassian to identify and solve the problem.

## 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

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 Installation Guide

This guide describes the advanced installation options that can be used when installing Fisheye.

> Evaluating FishEye for the first time? See the FishEye 101 page.

- Installing FishEye
- Configuring FishEye
- Best Practices for FishEye Configuration
- Setting JAVA_HOME
- Setting JVM System Properties

## Installing FishEye

This guide describes the advanced FishEye installation options.

> Evaluating FishEye for the first time? See the FishEye 101 page.

1. Download the FishEye zip file and extract it to a folder on your local computer.
   Folder names in the path to your FishEye executable should have no spaces in them.
   The directory where you have extracted FishEye is referred as `/FISHEYE_HOME/` below.
2. Ensure you have installed an appropriate Java runtime - see Supported Platforms.
3. Ensure that `java` is in the `PATH`, or that the `JAVA_HOME` environment variable is set.
4. Go to `/FISHEYE_HOME/` and run `bin/run.sh` if you are on Linux/MacOS and `bin\run.bat` on Windows
5. Wait for a few seconds for FishEye to start

6. Open your browser at http://localhost:8060/ (or type the http://hostname:8060/ of the machine where you extracted FishEye).
7. Enter your license, then the admin password to finish the setup
8. 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.
9. If you intend to use FishEye with Perforce, please ensure you read the Perforce client setup.

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

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

### Detailed installation guide

See the guidelines on Running FishEye for the First Time.

### FishEye Folder Layout

#### *FISHEYE_HOME (Default)*

By default, FishEye will run self-contained within the `/FISHEYE_HOME/` directory. The FishEye directory layout looks like this:

| | |
|---|---|
| `/FISHEYE_HOME/config.xml` | Configuration file. |
| `/FISHEYE_HOME/var/` | Directory under which FishEye stores its data. |
| `/FISHEYE_HOME/var/data/` | Persistent information. |
| `/FISHEYE_HOME/var/cache/` | Caches and indexes. |
| `/FISHEYE_HOME/var/log/` | Log files. |
| `/FISHEYE_HOME/var/tmp/` | Temporary files. |
| `/FISHEYE_HOME/cache/` | Caches and indexes. (in addition to `FISHEYE_HOME/var/cache`) |
| `/FISHEYE_HOME/bin/` | Scripts for controling FishEye. |
| `/FISHEYE_HOME/lib/` | FishEye's dependent libraries. |
| `/FISHEYE_HOME/ ...` | Remainder omitted for brevity. |

However, this self-contained layout results in tedious copying of files each time you upgrade FishEye. Also, if you want to run multiple instances of FishEye, you need multiple `/FISHEYE_HOME/` installations. These two issues can be avoided by setting a FISHEYE_INST ('FishEye Instance') location.

#### *FISHEYE_INST (Optional)*

FISHEYE_INST is a location where your repository data is stored, separate from the installation location of the FishEye application. This may be necessary for practical or administrative reasons.

A separate FISHEYE_INST location is recommended for production installations of FishEye.
Once you have have declared your FISHEYE_INST, you will need to copy your FISHEYE_HOME/config.xml file to your FISHEYE_INST/ directory.

When the `FISHEYE_INST` environment variable is set, FishEye's directory layout becomes:

| | |
|---|---|
| `$FISHEYE_INST/config.xml` | |
| `$FISHEYE_INST/var/` | All permanent and most temporary data is found under `$FISHEYE_INST/var/` |
| `$FISHEYE_INST/cache/` | Caches and indexes are found under `$FISHEYE_INST/var/` (in addition to `$FISHEYE_INST/var/cache`) |

| | |
|---|---|
| `$FISHEYE_INST/lib/` | Site-specific Java libraries (.jars) that FishEye should load on startup. (Do not copy the dependent `/FISHEYE_HOME/lib/` files into here.) |
| `$FISHEYE_INST/syntax/` | Site-specific syntax highlighting definitions. |
| `$FISHEYE_INST/system.properties` | 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 via FISHEYE_OPTS, e.g.: `export FISHEYE_OPTS=-Dpropname=value` ) |
| `/FISHEYE_HOME/lib/` | FishEye's dependent libraries. |
| `/FISHEYE_HOME/syntax/` | FishEye bundled highlighting definitions. |
| `/FISHEYE_HOME/bin/` | |
| `/FISHEYE_HOME/ ...` | Remaining files are found under `/FISHEYE_HOME/`. |

The rest of this *Installation Guide* refers to `$FISHEYE_INST/`, but if you have not set `FISHEYE_INST` then it defaults to `/FISHEYE_HOME/` (the directory where you extracted FishEye).

# Configuring FishEye

This page describes the initial configuration of FishEye, including the steps contained in the FishEye setup wizard.

## Initial Configuration

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 these addresses before starting FishEye by editing `config.xml`.

**On this page:**



## Running FishEye for the First Time

To run FishEye for the first time:

- On Windows:

```
C:\> cd FISHEYE_HOME\bin
C:\FISHEYE_HOME\bin> run.bat
```

- On UNIX-based systems:

```
$ cd /FISHEYE_HOME/bin
$ ./run.sh
```

Once started, FishEye will run its own HTTP web server, on port 8060 by default.

You can access FishEye immediately by going to `http://HOSTNAME:8060/` in a browser.

If you want to run FishEye as a Windows service, please refer to this document.

## Setting the Administration Password

The first time you run FishEye, when you access the FishEye web server you will be asked for:

- An administrator password. This password controls access to the FishEye Administration pages.
- A license key. You can get a trial license from the Atlassian website.

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.

### Integrating FishEye with JIRA

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.

For detailed instructions and troubleshooting, see Configuring JIRA Integration in the Setup Wizard.

### Accessing the Administration Pages

Once you have set up an administrator password, you can access the administration pages at `http://HOSTNAME:8060/admin/`.

One of your first steps will be to add a repository.

You will also want to read about the command-line options for controlling FishEye.

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

### What's Next?

Want a hands-on tour of the best FishEye features? See FishEye 101.

## Configuring JIRA Integration in the Setup Wizard

This page describes the '**Connect to JIRA**' tab of the FishEye setup wizard.

### *Overview*

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*

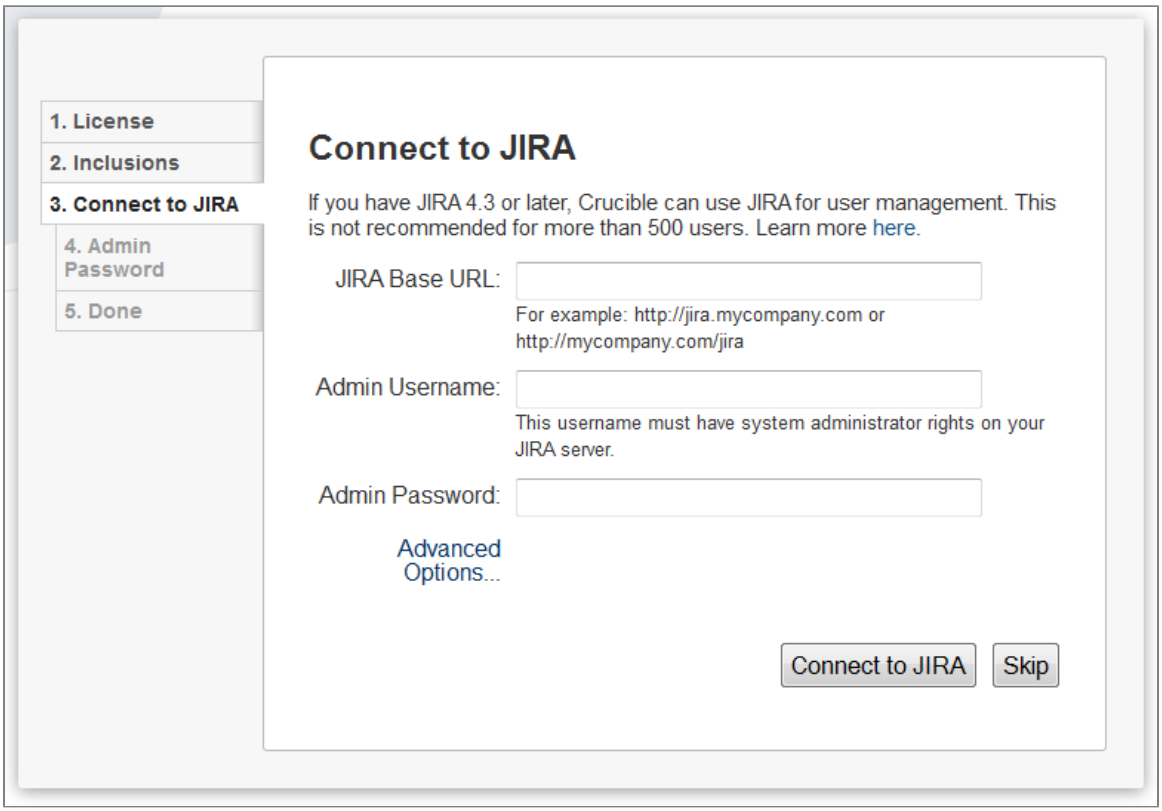
**To configure JIRA integration while running the FishEye setup wizard:**

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

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

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

2. Click the '**Connect to JIRA**' button.
3. Finish the setup process.
4. Configure the following setting in JIRA: Allow remote API access.




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

### *Troubleshooting*

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

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

| | | |
|---|---|---|
| The setup wizard displays one of the following error messages:<br>• *Failed to create application link from JIRA server at <URL> to this <application> server at <URL>.*<br>• *Failed to create application link from this <application> server at <URL> to JIRA server at <URL>.*<br>• *Failed to authenticate application link from JIRA server at <URL> to this <application> server at <URL>.*<br>• *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:<br>• *Failed to register <application> configuration in JIRA for shared user management. Received invalid response from JIRA: <response>*<br>• *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:<br>• *Error setting Crowd authentication* | The setup wizard successfully established the peer-to-peer link with JIRA, but could not persist the client-server link for user management in your `config.xml` file. This may be caused by a problem in your environment, such as a full disk. | Please investigate and fix the problem that prevented the application from saving the configuration file to disk. Then remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below. |
| The setup wizard displays the following error message:<br>• *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:<br>• *An error occurred: java.lang.IllegalStateException: Could not create the application in JIRA/Crowd (code: 500). Please refer to the logs for details.* | The setup wizard has not completed the integration of your application with JIRA. The links are only partially configured. The problem occurred because there is already a user management configuration in JIRA for this <application> URL. | Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below. |
| No users can log in after you have set up the application with JIRA integration. | Possible cause: There are no groups specified in the 'groups to synchronize' section of your administration console. | Go to the application administration screens and specify at least one group to synchronize. The default is '**jira-users**'. 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.
    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:
   Unable to render {include} Couldn't find a page to include called: _RemoveAppFromJIRAUserManagement
5. Go back to the setup wizard and click '**Connect to JIRA**' again.

**Solution 2: Removing a Partial Configuration – The Longer Way**

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

1. Skip the 'Connect to JIRA' step and continue with the setup wizard, to complete the initial configuration of the application.
2. Log in to JIRA as a user with the '**JIRA System Administrators**' global permission.
3. Click the '**Administration**' link on the JIRA top navigation bar.
4. Remove the application link from JIRA, if it exists:
    a. Click '**Application Links**' in the JIRA administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
    b. Look for a link to your application. It will have a base URL of the application linked to JIRA. For example: If you want to remove a link between JIRA and FishEye, look for the one where the '**Application URL**' matches the base URL of your FishEye server.
    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:
   Unable to render {include} Couldn't find a page to include called: _RemoveAppFromJIRAUserManagement
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. Select '**Other Applications**' from the '**Users, Groups & Roles**' section of the JIRA administration menu.
    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 FishEye/Crucible:
        a. Click '**Authentication**' on the FishEye/Crucible '**Admin Menu**'.
        b. Click '**Edit**' in the section titled '**JIRA or Crowd Authentication**'.
        c. Enter the following information:
            - **Application name** and **Application password** – Enter the values that you defined for your application in the settings on JIRA.
            - **JIRA/Crowd URL** – The web address of your JIRA server. Examples:

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

            - **Auto-add** – Select '**Create a FishEye/Crucible user on successful login**' (default) to ensure that your JIRA users will be automatically enrolled into FishEye/Crucible when they first log in via JIRA.
            - **Synchronize users with JIRA/Crowd** – Select '**Yes**' (default) to ensure that JIRA will synchronize all changes

in the user information on a regular basis. The **synchronization interval** is set to 60 minutes (1 hour) by default.
- **Single sign on (SSO)** – This option is not available when using JIRA for user management.
- **Groups of users to synchronize** – Select at least one group to synchronize. The default is '**jira-users**'.

d. Click '**Apply changes**'.

### *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.

Related Topics

User Management Limitations and Recommendations
Configuring FishEye
JIRA Integration in FishEye

## Best Practices for FishEye Configuration

***1. Ensure your FishEye scan performance is as fast as possible.***

Use the `file://` protocol for fastest indexing performance. Read more. (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 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.

***11. Do not use a 64 bit JVM.***

Refer to our Supported Platforms.

***See Also***

- Improve FishEye Scan Performance
- Tuning FishEye

## Setting JAVA_HOME

Once you have installed the JDK (see System Requirements), you need to set the JAVA_HOME environment variable.

### *To set the JAVA_HOME environment variable on Windows*

1. Right click on the '**My Computer**' icon on your desktop and select '**Properties**'.
2. Click the '**Advanced**' tab.
3. Click the '**Environment Variables**' button.
4. Click '**New**'.
5. In the '**Variable name**' field, enter 'JAVA_HOME'.
6. In the '**Variable value**' field, enter the directory (including its full path) where you installed the JDK.
7. Restart the computer.

### *To set the JAVA_HOME environment variable on Linux or UNIX based systems*

There are many ways you can do it on Linux or UNIX based systems (including Mac OS X). Here are two:

**For your current user,**

1. Open up a shell / terminal window
2. `vi ~/.profile` (replace vi with your favourite text editor)
3. Add `export JAVA_HOME=/path/to/java/home/dir` on its own line at the end of the file
4. Add `export PATH=$JAVA_HOME/bin:$PATH` on its own line immediately after
5. Save, and restart your shell
6. Running `java -version` should give you the desired results

**For all users in the system,**

1. Open up a shell / terminal window
2. `vi /etc/profile` (replace vi with your favourite text editor)
3. Add `export JAVA_HOME=/path/to/java/home/dir` on its own line at the end of the file
4. Add `export PATH=$JAVA_HOME/bin:$PATH` on its own line immediately after
5. Save, and restart your shell
6. Running `java -version` should give you the desired results

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.

## 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.

The recommended approach is to add the property to `FISHEYE_INST/system.properties`. Add this as a new text file, in the Java .properties format).

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).

# 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.

**On this page:**



## 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 via 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.

## 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 used to store repository data, separate from the installation location of the FishEye/Crucible application. It is *not* the installation directory. We recommend that you configure FishEye/Crucible to use a `FISHEYE_INST` directory for production instances. Read more about `FISHEYE_INST` in the FishEye Installation Guide.

### *Method 1 — Using a FISHEYE_INST Directory*

If you have FishEye/Crucible configured to use a `FISHEYE_INST` directory, then follow the instructions below:

1. Shut down your existing FishEye/Crucible server.
2. Make a backup of your `FISHEYE_INST` directory.
3. Extract the new FishEye/Crucible version to a directory, leaving your FISHEYE_INST environment variable set to its existing location.
4. Start FishEye/Crucible from the new installation.
5. Follow any version-specific instructions found in the FishEye Release Notes/Crucible Release Notes.

### *Method 2 — No 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:

*You will need to copy some files from your old FishEye/Crucible installation to your new one.*

1. Extract the new FishEye/Crucible instance into a directory such as `/NEW_FISHEYE/`.
2. Delete the `/NEW_FISHEYE/var` and `/NEW_FISHEYE/cache` directories.
3. Shut down the old FishEye/Crucible 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. Copy (or move) the `/OLD_FISHEYE/cache` directory to `/NEW_FISHEYE/cache`.
7. If you have a Cenqua-issued FishEye/Crucible license, copy your `fisheye.license` to `/NEW_FISHEYE/`. Atlassian-issued licenses are included within `config.xml`.
8. Start FishEye/Crucible from the new installation.

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:

1. Shut down the old FishEye/Crucible 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. Copy the `/OLD_FISHEYE/cache` directory to `/FISHEYE_INST`.
6. Copy the `/OLD_FISHEYE/data` directory to `/FISHEYE_INST`.
7. Extract the new FishEye/Crucible archive into a directory such as `/NEW_FISHEYE/`.
8. Start FishEye/Crucible from the new installation by running `NEW_FISHEYE/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 `FISHEYE_HOME` and `FISHEYE_INST`. Check 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 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.

## Supported Platforms

This page shows the supported platforms for **FishEye 2.6.x** and its minor releases.

**Key**: ✔ = Supported; ✖ = Not Supported

| **Java Version** | |
|---|---|
| JRE / JDK (1) | ✔ 1.6 |
| **Operating Systems** | |
| Microsoft Windows (2) | ✔ |
| Linux (2) | ✔ |
| Apple Mac OS X (2) | ✔ |
| **Databases** | |
| MySQL | ✔ MySQL Enterprise Server 5.0.21 or later<br>✔ MySQL Community Server 5.0.21 or later |
| PostgreSQL | ✔ 8.0, 8.1, 8.2, 8.3, 8.4 |
| Oracle | ✔ 11g |

| | |
|---|---|
| SQL Server | 2005 and 2008 |
| HSQLDB [3] | (bundled; for evaluation use only) |
| **Web Browsers** | |
| Microsoft Internet Explorer | 8.0, 9.0<br>6.0, 7.0 are not supported |
| Mozilla Firefox | 3.6, 4.0 |
| Safari | 4, 5 |
| Chrome | Stable Channel |
| **Version Control Systems** | |
| Subversion | Server 1.1, 1.2, 1.3, 1.4, 1.5, 1.6<br>Client uses included SVNkit. Native JavaHL 1.6 or later may be used. |
| CVS (and CVSNT) | All versions |
| Perforce | Client version 2007.3 or later |
| Git | 1.6 |
| Mercurial | 1.5.1 - 1.8.4 (1.9 coming soon) (Python 2.4.3 or later) |
| IBM ClearCase | 2003.06.10 or later<br><br>**NOTE**: The support for ClearCase will end on April 4th 2012, after which ClearCase functionality will be removed from newer versions of FishEye.<br><br>For more information please see End of Support Announcement for IBM ClearCase. |

### Supported Platform Notes

1. FishEye requires **Java Runtime** (JDK or JRE), version as noted in the table above. Pre-release/Early access versions of the Java Runtime are *not supported*.

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

2. FishEye is a pure Java application and should run on any platform provided the requirements for the JRE or JDK are satisfied.

3. The FishEye built-in database, running HSQLDB is somewhat susceptible to data loss during system crashes. External databases are generally more resistant to data loss during a system crash.

FishEye also supports a number of external databases, as noted in the table above. See the FishEye Database documentation for further details.

### Hardware Requirements

FishEye should ideally run on a standalone 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 (not NFS or SAN).

| 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). |

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 standalone Java program. It cannot be deployed to web application servers such as WebSphere, Weblogic or Tomcat.

### Single Sign On with Atlassian Crowd

FishEye is bundled with the Crowd client library, and operates with Crowd 1.3 or later.

**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 |
|---|---|
| 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 Web Browsers for FishEye (21 March 2011)

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

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

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

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

#### End of Life Announcement for Web Browser Support

| 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 `eol-announcement at atlassian dot com`.

#### End of Life Announcement for Java Platform Support

| Java Platform | Support End Date |
|---|---|

| | |
|---|---|
| **Java Platform 5 (JDK/JRE 1.5)** | When 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.

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

| 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 features.

### End of Support Announcement for IBM ClearCase

On **April 4th 2012**, we are ending FishEye support for IBM 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 **remove all functionality** related to ClearCase, from FishEye versions released after April 2nd 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.

# FishEye FAQ

**FishEye FAQ**

Answers to frequently asked questions about configuring and using FishEye.

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?<br>• How do I filter results?<br>• How do I find changes between two versions, showing separate histories?<br>• How do I find changes made between two version numbers?<br>• How do I find commits without comments?<br>• How do I find files on a branch, excluding deleted files?<br>• How do I find files removed from a given branch?<br>• How do I find revisions made by one author between versions?<br>• How do I select the most recent revisions in a given branch?<br>• 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.

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.

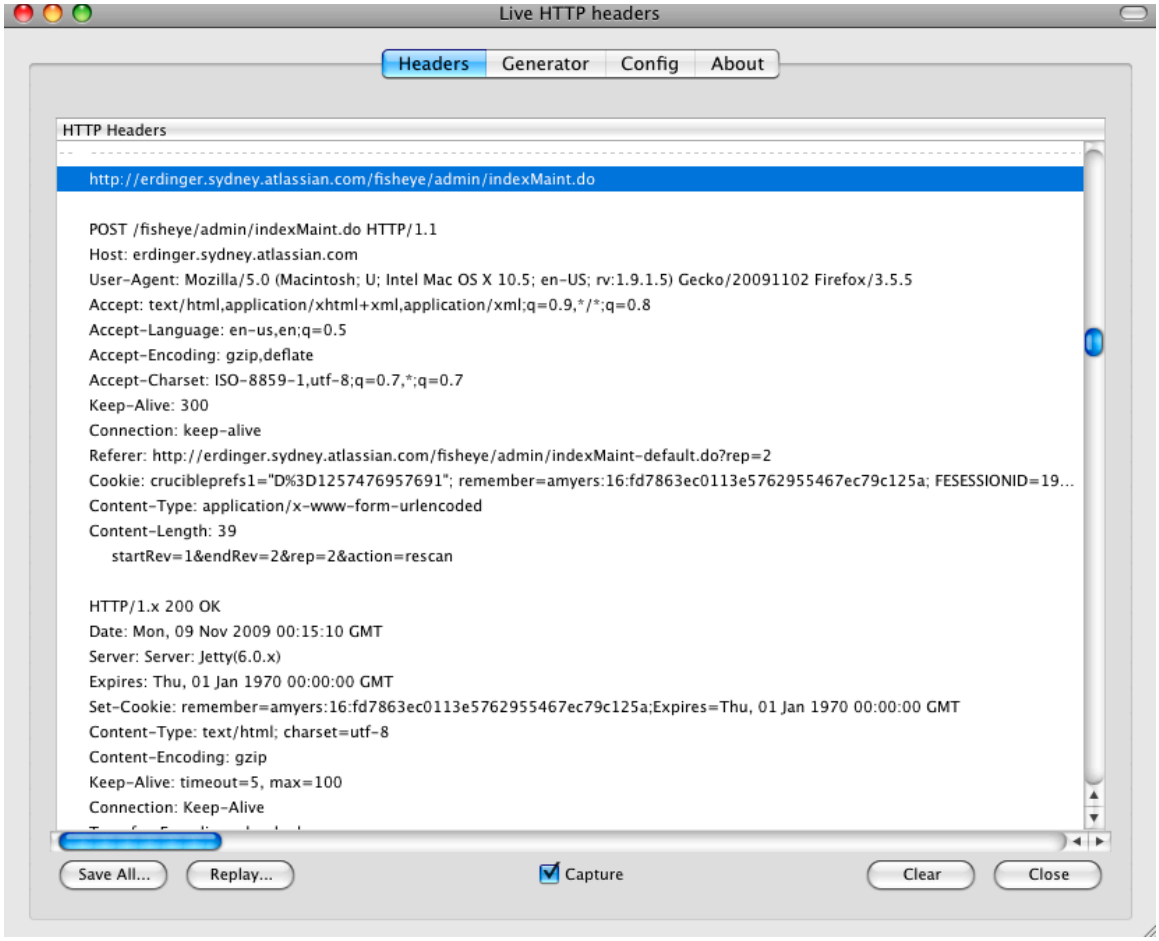
## General FAQ

| FishEye General FAQ |
| --- |
| • Automating Administrative Actions in Fisheye<br>• About Database Encoding<br>• About the Lines of Code metric<br>• Cannot View Lines of Code Information in FishEye<br>• Finding your Server ID<br>• How Do I Archive a Branch within Perforce<br>• How do I avoid long reindex times when I upgrade? — If re-indexing 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 re-indexing process.<br>• Mercurial Known Issues<br>• Ordering of Branches Important When Visualising Git Changesets<br>• Perforce Changesets and Branches<br>• What SCM systems are supported by FishEye? |

### 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 Adminstration 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.

## 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, a 'case sensitive collation'.

If your database was not created according to these instructions: Migrating to an External Database, then you may be using a case-insensitive and/or non-UTF8 collation. If so, 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 for FishEye/Crucible.

**On this page:**

- Oracle

### 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 CRU_STORED_PATH Table

Use the ALTER TABLE command, as follows:

```
alter table cru_stored_path convert to character set utf8 collate utf8_bin;
```

### 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 via the standard database migration procedure – instead of migrating from HSQLDB to PostgreSQL, you are migrating from a PostgreSQL DB with the incorrect encoding to one created with the correct encoding.

### Oracle

Oracle database 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.

## 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*

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 you use 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 and each branch as defined in the repository (see here for more information about defining branches in Subversion). Also, Fisheye 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** show 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

You cannot view lines of code information in FishEye, for example in charts or when viewing the statistics for a user.

*See About the Lines of Code metric for more information about the use of the lines of code metric in FishEye.*

#### Cause

There are three possible causes for this problem:

- Lines of code data will not be shown for users if the Store Diff Info setting is disabled. If you are viewing a page in FishEye that relates to a particular user or committer, and the Store Diff Info setting is disabled, you will not be able to view any lines of code information for the user.
- Lines of code data is currently not supported for Mercurial repositories. If you are viewing a page in FishEye that relates to a Mercurial repository, you will not be able to view any lines of code information.
- Lines of code data is currently not supported for Git repositories. If you are viewing a page in FishEye that relates to a Git repository, you will not be able to view any lines of code information.

**Resolution**

Cannot view lines of code information for specific users or committers:

- Enable the Store Diff Info setting in general settings for your repository. See Configuring Repository Details for further instructions. 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 note, the Store Diff Info setting is always enabled for CVS repositories.

Cannot view lines of code information for Mercurial repositories:

- We have an outstanding feature request for lines of code support in Mercurial. Please vote for this issue: CRUC-4505, if you are experiencing this problem.

Cannot view lines of code information for Git repositories:

- We have an outstanding feature request for lines of code support in Git. Please vote for this issue: CRUC-4524, if you are experiencing this problem.

## 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 Re-Index Times*

If re-indexing 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 re-indexing process.

### *Re-indexing With a Temporary Copy of Your FishEye Instance*

**To re-index 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 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, do the following. Use rsync (for CVS repositories in the Linux environment) or svnsync documentation (for Subversion only).

#### How to Re-Index a Single Repository on a Test Server

If you need to re-index 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 Crucible with the exact same name and details as that referenced by the production server.
3. Let that 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.

For this procedure, neither server needs to be shut down.

## Mercurial Known Issues

- CRUC-3466: FishEye does not support scanning repositories requiring http authentication, and only supports ssh based remote repos when run within a ssh-agent session with the correct keys added to authenticate against the remote repository.
- CRUC-3460: LOC changes can get counted twice when merged to another branch - once for the branch being merged, and once for the merge commit - this should only affect files modified in both parent trees (not files only modified on one branch).
- CRUC-3489: LOC counts for branches only show totals for the diffs made on the branches without accounting for the initial LOC that existed when the branch was created, thus can appear as negative values!
- CRUC-3467: FishEye will fail to detect files moving from binary to text until subsequent modifications are made.
- CRUC-3520: In the browse file view, FishEye will often fail to "show related revisions" due to copies or moves.
- CRUC-3408: If the watched repository has multiple heads on the same branch, the content searching can sometimes be relevant only for the first head seen.
- CRUC-3525: When entering a partial hash, if it doesn't match it can sometimes result in an internal error within FishEye.
- 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-3582: EyeQL searches based on mercurial tags do not work.
- 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).
- CRUC-3354: The revision ID is not displayed along with the hash ID for a commit in all locations, particularly within Crucible reviews.

## 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.

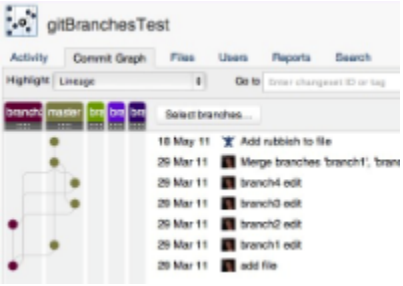
When you view the commit graph for a Git repository, FishEye works from the leftmost swimlane to the right doing the following:

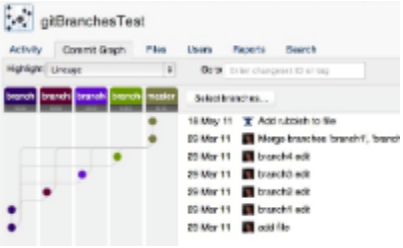
- For each swimlane, FishEye checks if the commit is in that branch. If the commit is in the branch, a dot is shown representing the commit.
- If the commit is not in the branch, the dot for the commit is moved to the next column on the right.

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
3

## 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.

# Installation & Configuration FAQ

**FishEye Installation & Configuration FAQ**



## How to reset the Admin password in Fisheye or Crucible

If you have forgotten/misplaced the admin password, you will need to reset it manually.

To 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 Administration page with the password: **admin**. Please change this password immediately from the Administration panel.

## How Do I Confugure 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:

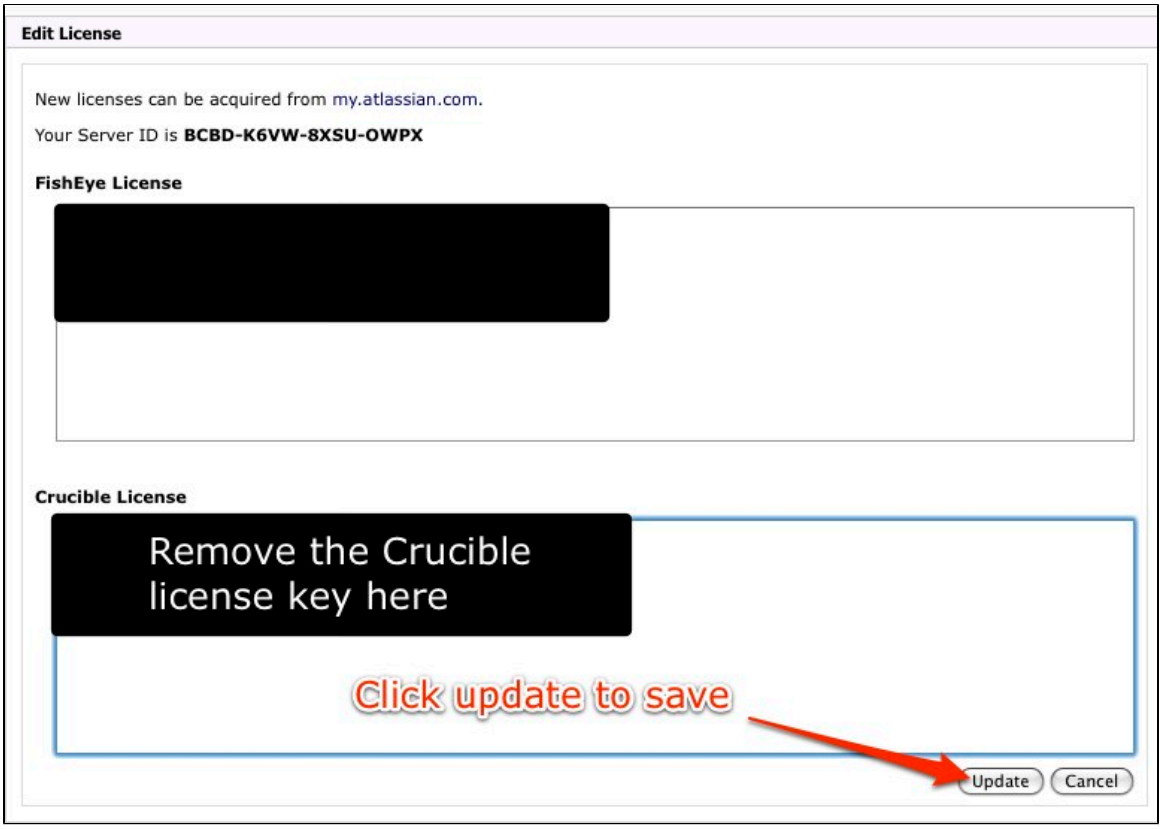
```
export FISHEYE_OPTS="-Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=8080
-Dhttp.nonProxyHosts=*.foo.com|localhost -Dhttp.proxyUser=username -Dhttp.proxyPassword=password"
```

See Environment Variables for instructions on how to set these parameters.

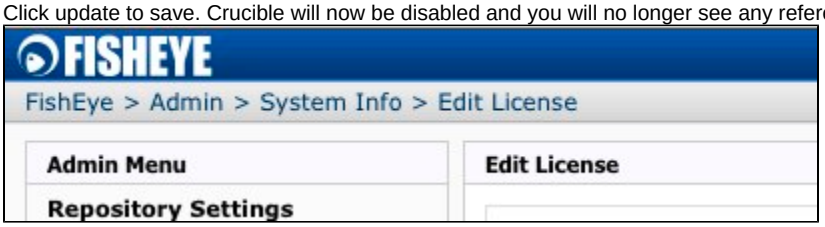
## How to Remove Crucible From FishEye 2.x or Later

To remove Crucible from a FishEye installation:

1. Go into Administation -> Sys-Info/Support -> Edit Licenses
2. Remove the Crucible license from this screen




3. Click update to save. Crucible will now be disabled and you will no longer see any reference to it in the application.




The process is the same if you would like to remove FishEye and keep Crucible, just remove the FishEye license instead.

## How to run Fisheye or Crucible on startup on Mac OS X

This article is only provided as a guide and has only been tested on Mac OS X 10.5

launchd does not provide support for service dependencies so if you are using an external database, this may not work for you. Fisheye assumes the database is available when it starts, and the startup scripts will not wait for the database to become available.

You need to create a .plist file to create items that will start at boot time. Please refer to the following page for details:

- 

Here is an example .plist that should work for Fisheye/Crucible:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
 <key>Label</key>
 <string>com.atlassian.fisheye</string>
 <key>ProgramArguments</key>
 <array>
  <string>/path/to/FISHEYE_HOME/bin/run.sh</string>
 </array>
        <key>OnDemand</key>
        <false/>
        <key>RunAtLoad</key>
        <true/>
 <key>StandardOutPath</key>
 <string>/path/to/FISHEYE_INST/var/log/fisheye.out</string>
 <key>StandardErrorPath</key>
 <string>/path/to/FISHEYE_INST/var/log/fisheye.out</string>
 <key>EnvironmentVariables</key>
 <dict>
  <key>FISHEYE_INST</key>  <string>/path/to/FISHEYE_INST/</string>
  <key>FISHEYE_OPTS</key>  <string>-Xms512m -Xmx1024m -XX:MaxPermSize=128m</string>
 </dict>

</dict>
</plist>
```

Customise the `/path/to/FISHEYE_INST/` and `/path/to/FISHEYE_HOME/` with the FISHEYE_INST and FISHEYE_HOME directories respectively and make any required modifications to FISHEYE_OPTS. Save the file as com.atlassian.fisheye.plist and then try and load it with:

```
[amyers@erdinger:fecru-2.1.3]$ launchctl load  com.atlassian.fisheye.plist
[amyers@erdinger:fecru-2.1.3]$ launchctl start com.atlassian.fisheye
```

Fisheye should now start up and you should be able to access it via your web browser.

## Can FishEye be run as a Windows service?

FishEye can be run as a service under Microsoft Windows.
To run FishEye as a service you can either use SRVANY and INSTSRV to run `java.exe` or create a Java Service Wrapper. A mechanism to run FishEye as a service will be incorporated at a later stage. In the meantime, example wrapper files written by FishEye users can be found here.

To install on Windows:

1. Unzip the wrapper zip file into your `FISHEYE_HOME` directory (Note, the end structure should be `FISHEYE_HOME/wrapper`, `FISHEYE_HOME/wrapper/bin`, etc and NOT `FISHEYE_HOME/wrapper/wrapper`, `FISHEYE_HOME/wrapper/wrapper/bin`. The location of the wrapper directory is important).
2. Run `Fisheye-Install-NTService.bat`, found in `FISHEYE_HOME/wrapper/bin`.
3. Start the Fisheye service under the Windows Control Panel.
4. Set your FISHEYE_INST within your `FISHEYE_HOME/wrapper/conf/wrapper.conf` as per the instructions below:

Please note, that if you do run as a service, then any Environment Variables that you want to set, need to be set in your `FISHEYE_HOME/wrapper/conf/wrapper.conf` file.

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

```
# JDK 1.5 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"
```

For example if you wish to add a FISHEYE_INST environment variable or add the java parameter "MaxPermSize", or the -Xrs options (should be used if running FishEye as a service under Windows, to prevent the JVM closing when an interactive user logs out) then it would be something like:

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

Your memory settings can also be found in this file:

```
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=32

# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=256
```

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

**Wrapper Configuration and "-server" Parameter**
Please note that the Wrapper configuration provided above uses the `-server` parameter to enable the Java HotSpotTM 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: no `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 unless you change the sequence to start at `wrapper.java.additional.1`. This is an issue with the Wrapper application.

In this situation it's likely best to install and run Fisheye/Crucible with the JDK to get all the advantages of the `-server` functionality.

## 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.

## Configuring 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 [FISHKB:Configuring 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.

## 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 (in some cases weeks), 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.

### *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 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`*

1) Shut down your current FishEye server completely.
2) Copy the `FISHEYE_HOME/config.xml` file and `FISHEYE_HOME/var` folder into to one folder on your destination server, called `fisheye_inst`.
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` env variable such as follows

```
export FISHEYE_INST=/path/to/fisheye_inst
```

replacing the `/path/to/fisheye_inst` with the fully qualified path to the `fisheye_inst` folder you set up 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.

**External Databases**
Please note that the steps above will only move your Fisheye data. If you are using an external database, it 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.

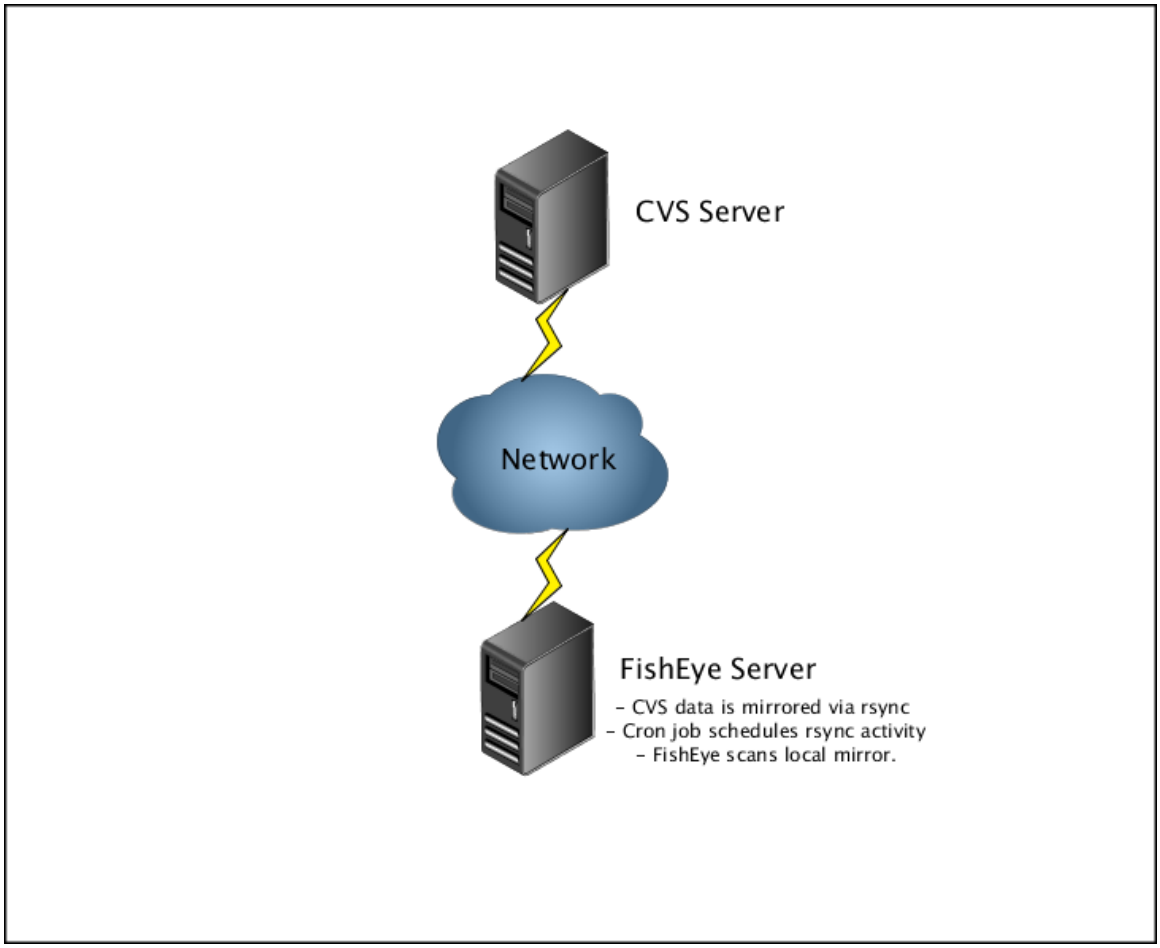
## Setting Up a CVS Mirror with rsync

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.

## 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?

Since version 2.5 of the FishEye JIRA plugin, it has provided a **Disable for unmapped JIRA Projects** option in the FishEye Configuration administration screen. If set to **true**, the FishEye tab panel will not display for projects that do not have explicit repository mappings.

Note there was a limitation regarding Crucible tab panels that was addressed in version **3.0.14** of the plugin.

### 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&le
```

To enable debug logging across JIRA restarts, update the **CLASS-SPECIFIC LOGGING LEVELS** section of your `WEB-INF/classes/log4j.properties` file by inserting the following:

```
log4j.category.com.atlassian.jira.ext.fisheye = DEBUG, console, filelog
log4j.additivity.com.atlassian.jira.ext.fisheye = false
```

and restarting JIRA.

**For Plugin versions 3.0+**

You can enable DEBUG logging for the JIRA FishEye plugin temporarily (until JIRA is shutdown) by accessing the following URL in a web browser when logged in as an administrator:

```
http://YOUR-JIRA-BASE-URL/secure/admin/ConfigureLogging.jspa?loggerName=com.atlassian.jirafisheyeplugin&l
```

To enable debug logging across JIRA restarts, update the **CLASS-SPECIFIC LOGGING LEVELS** section of your `WEB-INF/classes/log4j.properties` file by inserting the following:

```
log4j.category.com.atlassian.jirafisheyeplugin = DEBUG, console, filelog
log4j.additivity. com.atlassian.jirafisheyeplugin = false
```

and restarting JIRA.

## How do I uninstall the JIRA FishEye plugin?

**Plugin versions prior to 3.0**

To uninstall, simply remove the `jira-fisheye-plugin-*.jar` from your `WEB-INF/lib` directory.

**Plugin version 3.0+**

To uninstall, simply remove any `jira-fisheye-plugin-*.jar` jars from your `WEB-INF/classes/com/atlassian/jira/plugin/atlassian-bundled-plugins.zip` and `%JIRA_HOME%/plugins/installed-plugins` directory.

## How is the Crucible review tab panel for the FishEye JIRA plugin populated?

The Crucible 'Reviews' tab is populated differently depending on the version of the FishEye JIRA plugin you are using:

- **Prior to Version 2.3** of the plugin, a review is displayed on the Crucible tab panel if (and only if) it contains a revision that is displayed on the FishEye tab panel. Having the issue key in the description *will not* automatically link it to the issue.
- In **Version 2.3+**, you can configure the Crucible tab panel to use the method described above and/or *search for issue keys in the title/description of reviews*.

> If you associate reviews with JIRA issues from the Crucible user interface, they will not necessarily show 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.<br>• What are the Starter License restrictions? |

## 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 accomodated 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.

## What are the Starter License restrictions?

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?*

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

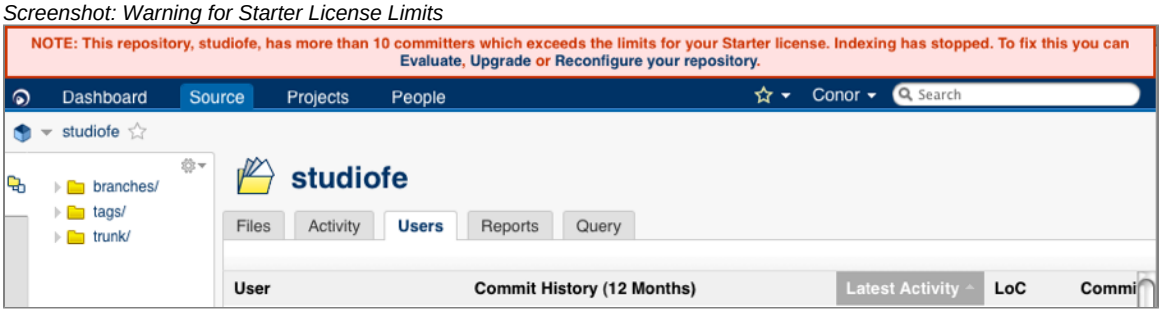
### *What happens if I exceed the Starter License limits?*

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 limts 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.

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 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 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. |

## 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 you have defined multiple repositories in FishEye for the same Subversion 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 you have defined multiple repositories in FishEye for the same Subversion server and those repositories use different credentials, FishEye may not use the correct credentials.

FishEye does not directly control when authentication information is used to access Subversion repositories. It delegates this operation to the JavaHL layer in use. JavaHL will 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.

If you experience this problem, you can configure FishEye to use the native JavaHL implementation, which will correctly apply the appropriate credentials.

## 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.

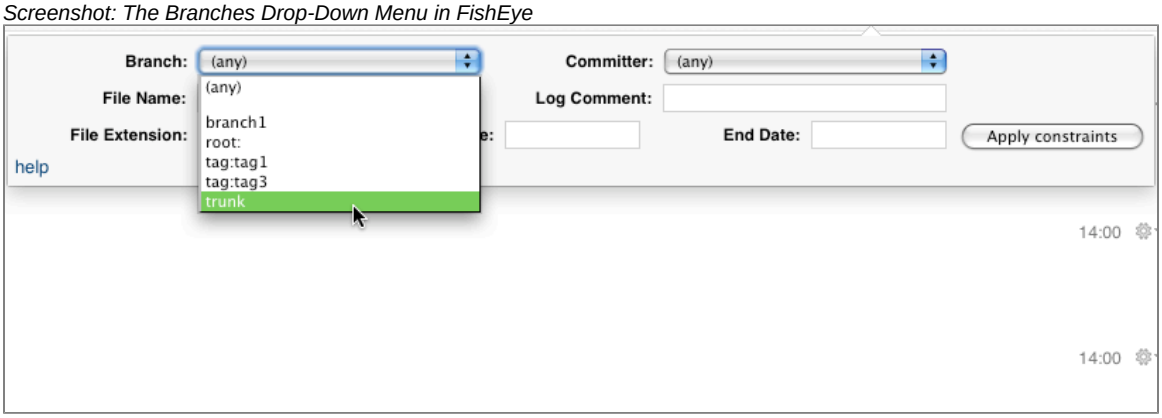
*On this page:*



### *Introduction*

When accessing Subversion via FishEye, you may see references to tags in the branches drop-down menu. In the example below, we can see **tag1** and **tag3** in the drop-down menu but not **tag2**:

*Screenshot: The Branches Drop-Down Menu in FishEye*




In actual fact, the branches drop-down menu shows only branch names. It does not show tags, but in some instances FishEye will synthesise a branch name to record certain operations. To understand how this occurs, you will need some background knowledge on Subversion tagging (introduced in the following segments of this page).

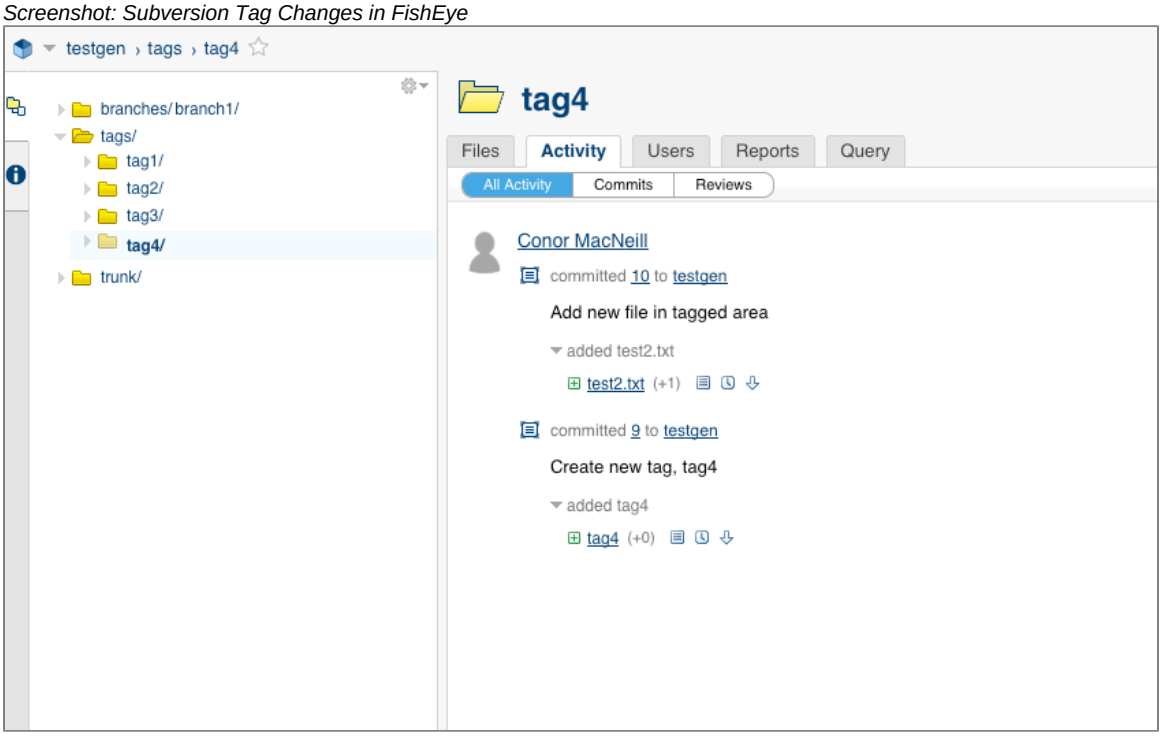
### *How Subversion Processes Tags and Branches*

> In Subversion, tags are only a convention and are typically the result of a copy operation from the trunk to a tag area in the tags directory. When FishEye processes this copy operation, it recognises that the destination is a tag directory and tags the source file on trunk with the name of the tag.
>
> i.e. FishEye is interpreting the Subversion copy to a tag directory as a tagging operation on the trunk files.
>
> For regular changes in your Subversion repository, FishEye records each change against a branch where the change took place. If, however, after tagging, you make a change to a file in the tagged area, you are making a change outside trunk or a recognized branch. FishEye records such changes by creating an artificial branch name and associating that branch name with the change. The branch name is derived from the tag name by prepending "tag:" (in other words, the characters "tag:" appear as the first part of the name). The same thing will occur if you create a new file in the tagged area which does not come from an existing branch or trunk.
>
> This is the reason you see some of your tags in the branch drop down. It means that for those tags, you have made a modification after the tagging operation.

### *An Example from a Live Subversion Repository*

For example, consider **tag4** in this screenshot:

*Screenshot: Subversion Tag Changes in FishEye*




There are two changes here. The first creates the tag and the second adds a new file in the tagged area. This will result in the creation of an artificial branch, called "**tag:tag4**" within FishEye.

### *Avoid Modifications in the Tag Area*

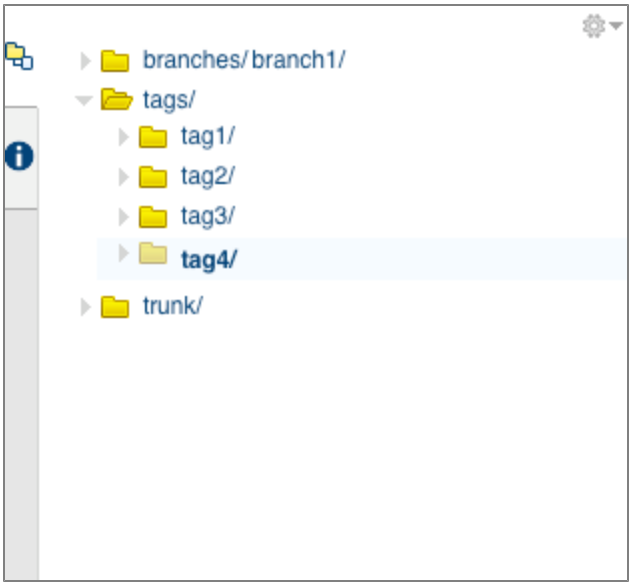
In general, it's not good practice to make changes in the tag areas of a Subversion repository. Such changes can easily get lost if they are not applied to trunk or a current branch. It is preferable to make the change in trunk or a branch and then create a new tag to capture the update. Nevertheless, since Subversion tagging is merely a convention, this is sometimes convenient. FishEye handles this situation as described above.

### *Conclusion*

In general a lot of systems have a large number of tags which would make the drop-down unworkable. This is the reason the tag field is a text-entry box below the branch drop-down menu in FishEye.

Since tags and branches are based on location convention in Subversion, the constraint is less effective than on other SCMs. You can always see the tag or branch you are interested in, based on its location in the repository. For example, the subdirectory list here shows all tags:

*Screenshot: Subdirectory Listing in FishEye*




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.

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 How to Get Legendary Support from Atlassian for more support-related information.

## How to Report a Security Issue

### Finding and Reporting a Security Vulnerability

If you find a security bug in the product, please open an issue on http://jira.atlassian.com in the relevant project.

- Set the priority of the bug to 'Blocker'.
- Provide as much information on reproducing the bug as possible.
- Set the security level of the bug to 'Developer and Reporters only'.

All communication about the vulnerability should be performed through JIRA, so that Atlassian can keep track of the issue and get a patch out as soon as possible.

> If you discover a security vulnerability, please attempt to create a test case that proves this vulnerability locally before opening either a bug or a support issue. When creating an issue, please include information on how the vulnerability can be reproduced; see our Bug Fixing Policy for general bug reporting guidelines. We will prioritise fixing the reported vulnerability if your report has information on how the vulnerability can be exploited.

### *Further reading*

See How to Get Legendary Support from Atlassian for more support-related information.

## New Features Policy

### *Summary*

- We encourage and display customer comments and votes openly in our issue tracking systems, http://jira.atlassian.com and http://studio.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:

- **Direct feedback** from face to face meetings with customers, and through our support and sales channels.
- **Availability of staff** to implement features.
- **Impact** of the proposed changes on the application and its underlying architecture.
- How **well defined** the requested feature is (some issues gain in popularity rapidly, allowing little time to plan their implementation).
- 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 systems, http://jira.atlassian.com and http://studio.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 our community forums.

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 How to Get Legendary Support from Atlassian 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 (e.g. JIRA 4.2.4) 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 How to Get Legendary Support from Atlassian for more support-related information.

## Security Advisory Publishing Policy

### Publication of Security Advisories

When a 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. This applies to all security advisories, including severity levels of critical, high, medium and low.
- We will send a copy of all 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 'Email Prefs' 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.

Early warning of critical security vulnerabilities:

- If the vulnerability is rated critical (see our criteria for setting severity levels) we will send an early warning to the 'Technical Alerts' mailing list approximately one week before releasing the fix. This early warning is in addition to the security advisory itself, described above.
- However, if the vulnerability is publicly known or being exploited, we will release the security advisory and patches as soon as possible, potentially without early warning.

### *Further reading*

See How to Get Legendary Support from Atlassian for more support-related information.

## Security Patch Policy

### Product Security Patch Policy

Atlassian makes it a priority to ensure the 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 Studio, since 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 How to Get Legendary Support from Atlassian 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 include:

- 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 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 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 How to Get Legendary Support from Atlassian 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 Enterprise Server 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 inital 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 Enterprise Server.
- 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 Enterprise Server 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(NewProxyPreparedStatement.java:1723)
 at org.hibernate.jdbc.BatchingBatcher.doExecuteBatch(BatchingBatcher.java:48)
 at org.hibernate.jdbc.AbstractBatcher.executeBatch(AbstractBatcher.java:246)
 ... 163 more
```

The FishEye error log can be found under `FISHEYE_INST/var/log/fisheye-error.log.YYYY-MM-DD` .

See the JIRA issue for more information.

### *Workaround*

Until the issue is solved, the suggested course of action is to restart your FishEye instance. This will return FishEye to normal operation.

The FishEye development team is actively working on a solution and this be part of an upcoming point release of FishEye.

### *Requesting Support*

If you require assistance in resolving the problem, please raise a support request under the FishEye project.

## Generating a Thread Dump Externally

If Fisheye stops responding or is showing poor performance, providing thread dumps to support can help diagnose the problem.

### *Generating a Thread Dump for Windows*

To take a thread dump:

1. Visit http://www.adaptj.com/root/main/download and click Launch
2. Click Run for any security warnings
3. Select Process -> Thread Dump
4. Under Process Id, select the '...' button.
5. From the drop-down list, select the Confluence process. Users running Fisheye, select the 'Java ...' option.
6. Ensure that the "Thread dump" and "Keep Remote Thread Running" is selected.
7. Click OK to capture the thread dump.
8. Save the output to a file, eg 'threaddump.log'

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.

> Alternatively, 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.

### *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 inital scan is taking a long time. Is this normal?

When you add a repository, FishEye needs to scan through the repository to build up its index and cache. This scan can take some time. Until this scan is complete, you may find that some data is not displayed.

As a guide, FishEye should be able to process about 100KB-200KB per second on an averaged-size PC. If FishEye is accessing the repository over the network (e.g. over a NFS mount), then you should expect the initial scan to take longer.

For more details, see Improve FishEye Scan Performance.

## I have installed FishEye, but there is no data in the Changelog.

When you add a repository, FishEye needs to scan through the repository to build up its index and cache. This scan can take some time. Until this scan is complete, you may find that some data is not displayed.

As a guide, FishEye should be able to process about 100KB-200KB per second on an averaged-size PC. If FishEye is accessing the repository over the network (e.g. over a NFS mount), then you should expect the initial scan to take longer.

## Initial scan and page loads are slow on Subversion

### *Background Information*

When you add a repository, FishEye needs to perform a once-off scan through the repository to build up its initial index and cache. This scan can take some time. Until this scan is complete, you may find that some data is not displayed. As a guide, FishEye should be able to process about 100KB-200KB per second on an averaged-size PC. If FishEye is accessing the repository over the network (e.g. over a NFS mount), then you should expect the initial scan to take longer. Read on if your scan appears to be considerably slower than expected.

### *Solutions*

It's possible that you've mis-configured your tag and branch structure and caused FishEye to process some or all files as trunk files. You should recheck your tag configuration.

If that fails, then the Atlassian support team will be happy to help you with this issue. Please sign up for a support account if you don't have one already, then login and create a FishEye support request.

Users with very large or non-local repositories may be able to improve their FishEye scan performance.

## It seems that FishEye's HTTP Header is Too Small

If you are receiving errors about FishEye's HTTP header being too small, it is adjustable. See the page on Setting JVM System Properties for instructions.

## JIRA Integration Issues

**Users are mapped to their own accounts when using Trusted Applications.**

If you (or the general account used for JIRA access, if not using Trusted Applications) do not have the permissions to carry out the JIRA actions linked from FishEye, an error will occur. Depending on the error returned from JIRA, FishEye may not display the error correctly or display it at all, simply reporting that "An error has occurred". To investigate what the error was, you can access the FishEye debug log, named `fisheye-debug.log.YYYY-MM-DD` under the `dist.inst/var/log` folder of your FishEyee 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 Enterprise Server. 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 Enterprise Server, 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 Enterprise Server 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 Enterprise Server 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 Development Hub

This page is deprecated. Please see the new Developer Documentation Space.

# Developing Fisheye Plugins

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## Plugin Module Types

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

### FishEye Web Items

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

### Servlet Plugin Module

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

#### Fisheye Page Decorators

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

### Spring Component Plugin Module

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

# Documentation for FishEye Development

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## FishEye's URL Structure

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

# Fisheye Plugins

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## Crucible Command Line Interface

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

### Crucible CLI - Remote API support matrix

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## Developer Report Plugin

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## FishEye Client for Eclipse

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

# FishEye REST API

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## Data Types

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## Repositories Service

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## Revision Data Service

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## Search Service

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## The FishEye Remote API

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

## Writing SOAP Clients for Fisheye or Crucible RPC Plugins

The content on this page is deprecated. Please see the separate documentation space for developer reference material about FishEye and Crucible.

# 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
- Atlassian Plugin Exchange

### 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:*



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

Standing 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.

## 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

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

Our documentation wiki contains developer-focused documentation (such as API guides, plugin and gadget development guides and guides to other frameworks) as well as product documentation (user's guides, administrator's guides and installation guides). The wiki permissions are different for each type of documentation.

- If you want to update the FishEye developer documentation, the Developer Network or other developer-focused wiki spaces, just sign up for a wiki username then log in and make the change.
- If you want to update the FishEye product documentation, we ask you to sign the Atlassian Contributor License Agreement (ACLA)

before we grant you wiki permissions to update the documentation space. Please read the ACLA to see the terms of the agreement and the documentation it covers. Then sign and submit the agreement as described on the form attached to that page.

### Following 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, will make the necessary changes.
- **Wiki permissions.** We use wiki permissions to determine who can edit the various types of documentation spaces.
  - Developer documentation (API guides, plugin development and gadget development): Anyone can edit these spaces, provided they have signed up for a wiki username and logged in to the wiki.
  - Product documentation (user's guides, administrator's guides, installation guides): 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:**



| Technical Tips |
| --- |
| ***Using a reverse proxy to Fisheye from Tomcat 5+ – Blog***<br>• By: Dieter Wimberger, on his Coalevo blog.<br>• About: How to use a reverse proxy to Fisheye from Tomcat 5.<br>• Date and FishEye version: 21 Jan 2008; FishEye 1.4.2.<br>• Related documentation: Integrating with Other Web Servers. |

| Plugin Development |
| --- |
| ***Diary of a FishEye Hacker – Part 1 and Part 2***<br>• By: Dan Hardiker, on the Adaptavist blog<br>• About: How to write a FishEye reporting plugin that provides a framework for alternative views on the repository data<br>• Date and FishEye version: 6 May 2008; FishEye 1.5<br>• 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 forums
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:**



| None |
| --- |
| ***No guides yet***<br>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

# TreeNavigation

Index

# TreeNavigationVersions

Click for all versions

FishEye 2.6 Documentation
FishEye 2.5 Documentation
FishEye 2.4 Documentation
FishEye 2.3 Documentation
FishEye 2.2 Documentation
FishEye 2.1 Documentation
FishEye 2.0 Documentation
FishEye 1.6 Documentation
FishEye 1.5 Documentation
FishEye 1.4 Documentation
FishEye 1.3 Documentation
FishEye 1.2 and Older Documentation