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 2.3

There are no bookmarks to display.

FishEye Documentation Home

FishEye 2.3.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.

ℹ 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.

> ℹ 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

> ✓ **Work with FishEye inside your IDE**
> Use the Atlassian Connector for Eclipse or the Atlassian Connector for IntelliJ IDEA to work with FishEye and Crucible right there in your development environment.

- Date Expressions Reference Guide
- Antglob Reference Guide
- EyeQL Reference Guide
- Using the FishEye Screens
    - Browsing through a Repository
    - Viewing the Changelog

Date Expressions Reference Guide

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

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

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

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

`DATE` can be either of the following:

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

`DATECONSTANT` can be any of:

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

* The timezone used for server-time is part of the FishEye configuration

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

Examples

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

`now+PT1H2M3S`	One hour, two minutes and three seconds from now

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

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
```

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

- 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).
- *If you are using Crucible:* Click the '**Projects**' tab to see a list of all projects (see the Crucible documentation).
- Click the '**People**' tab to view statistics about commiters to your FishEye repositories (see Viewing People's Statistics (draft))
- *If you are using Crucible:* Click the '**Reviews**' tab to go to your code reviews (see the Crucible documentation) .
- 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).

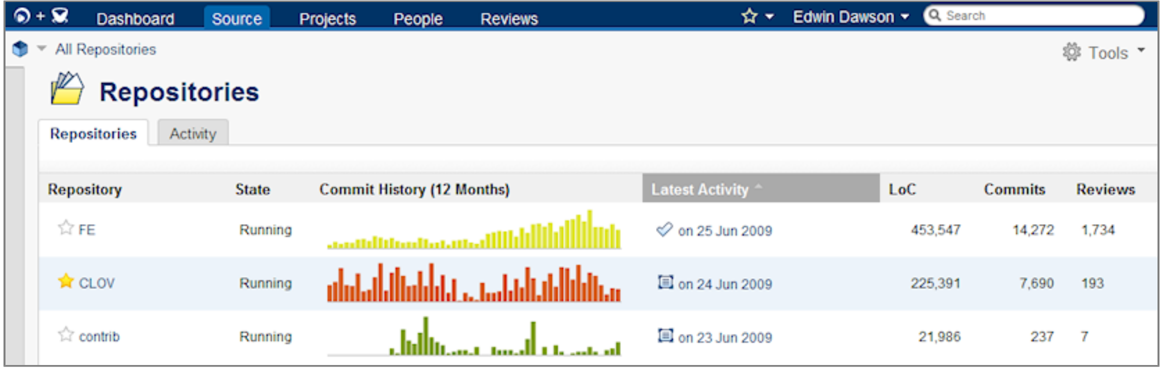
Repository Sub-Tabs

Once you have selected a repository, you can navigate through it by selecting files and folders on the tree in the left navigation bar. 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 into the repository:

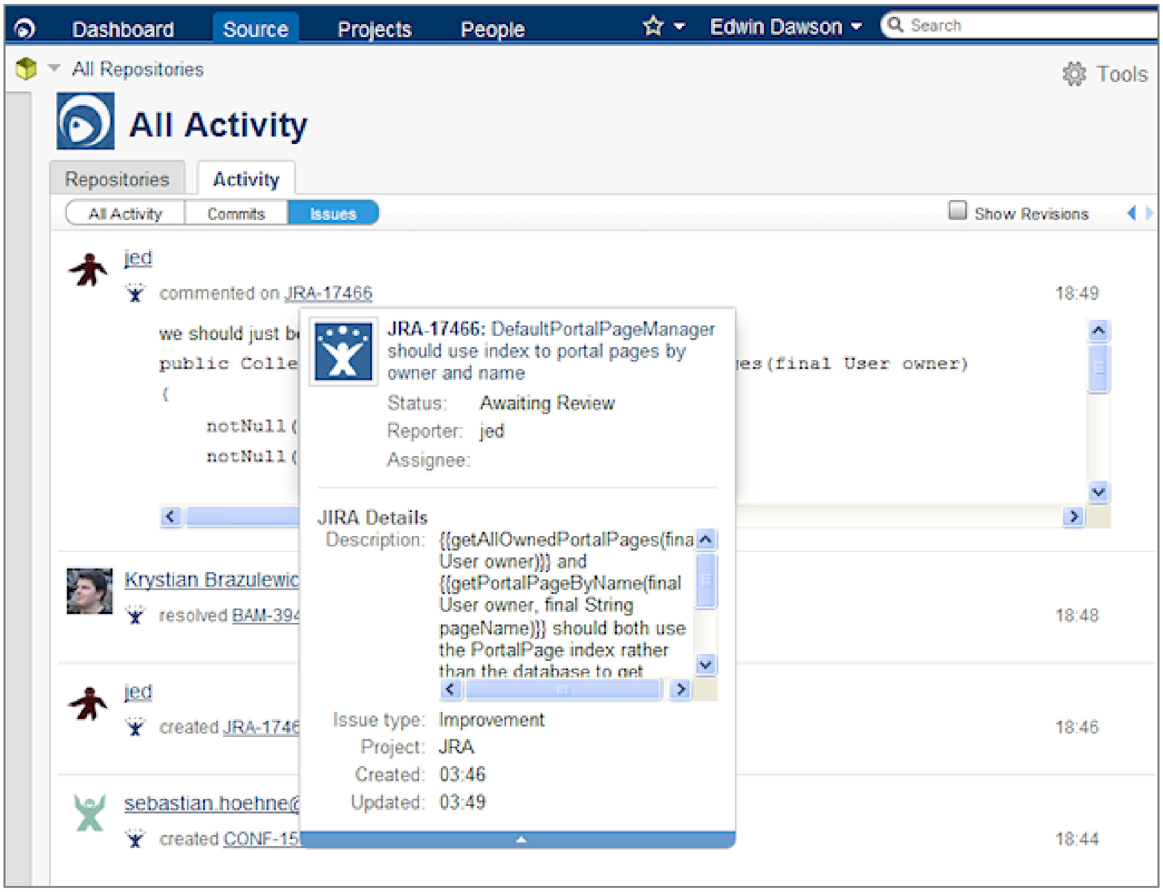
Sub-Tab Name	Description
Revisions	When viewing a file, shows the latest revisions of the file. See Viewing a File History.
Files	When viewing a folder, shows the contents of the directory.
Activity	Shows recent activity on the item. There are a number of sub-options here (see Viewing the Changelog):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

Screenshot: The Repositories View



Repository	State	Commit History (12 Months)	Latest Activity	LoC	Commits	Reviews
FE	Running		on 25 Jun 2009	453,547	14,272	1,734
CLOV	Running		on 24 Jun 2009	225,391	7,690	193
contrib	Running		on 23 Jun 2009	21,986	237	7

Screenshot: The Activity View



jed
commented on JRA-17466 18:49

we should just b
public Colle ...es(final User owner)
{
notNull(
notNull(

JRA-17466: DefaultPortalPageManager should use index to portal pages by owner and name

Status: Awaiting Review
Reporter: jed
Assignee:

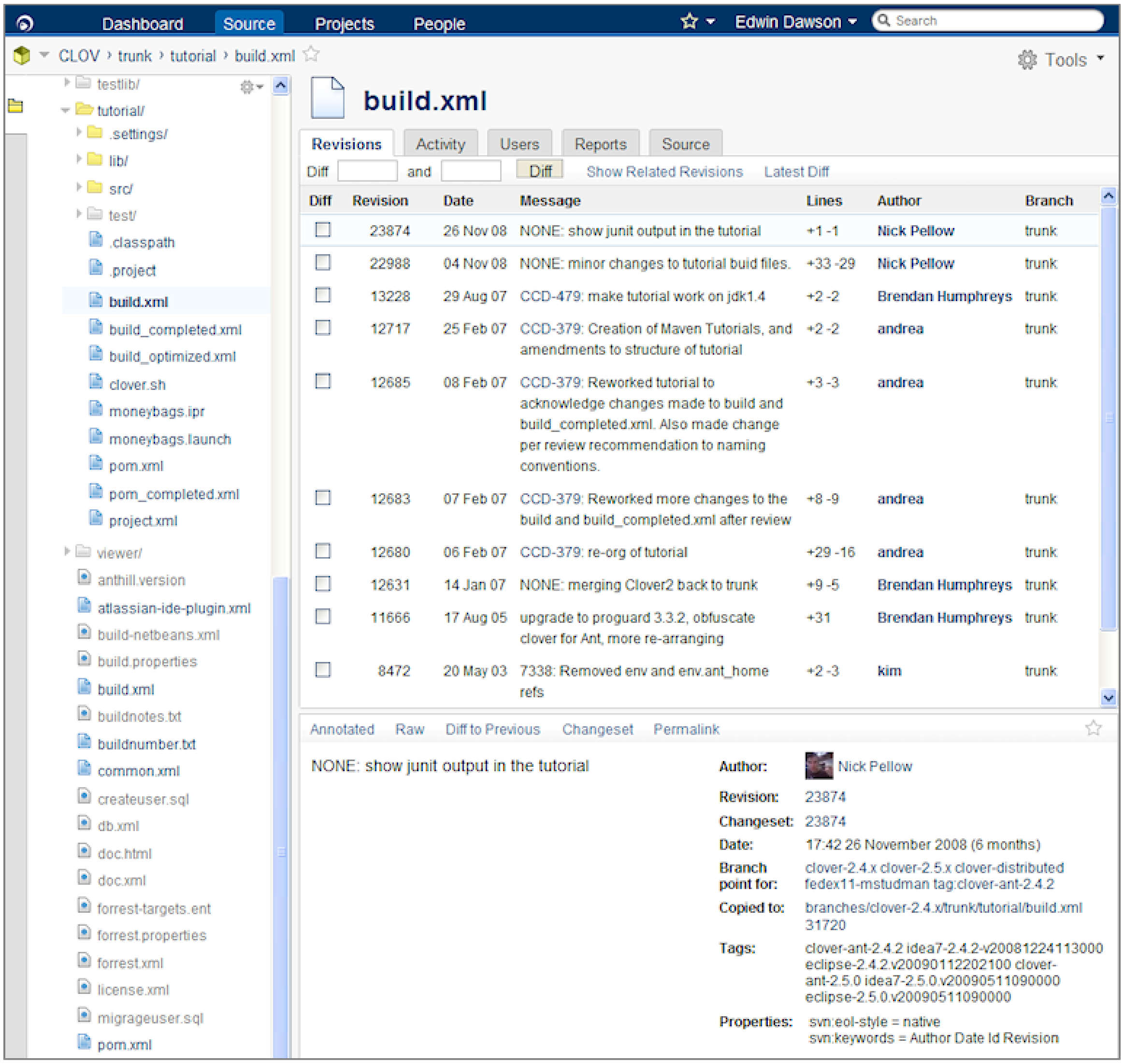
JIRA Details
Description: {{getAllOwnedPortalPages(fina User owner)}} and {{getPortalPageByName(final User owner, final String pageName)}} should both use the PortalPage index rather than the database to get
Issue type: Improvement
Project: JRA
Created: 03:46
Updated: 03:49

Krystian Brazulewic
resolved BAM-394 18:48

jed
created JRA-1746 18:46

sebastian.hoehne@
created CONF-15 18:44

Screenshot: Repository Sub-Tabs



Diff	Revision	Date	Message	Lines	Author	Branch
☐	23874	26 Nov 08	NONE: show junit output in the tutorial	+1 -1	Nick Pellow	trunk
☐	22988	04 Nov 08	NONE: minor changes to tutorial buid files.	+33 -29	Nick Pellow	trunk
☐	13228	29 Aug 07	CCD-479: make tutorial work on jdk1.4	+2 -2	Brendan Humphreys	trunk
☐	12717	25 Feb 07	CCD-379: Creation of Maven Tutorials, and amendments to structure of tutorial	+2 -2	andrea	trunk
☐	12685	08 Feb 07	CCD-379: Reworked tutorial to acknowledge changes made to build and build_completed.xml. Also made change per review recommendation to naming conventions.	+3 -3	andrea	trunk
☐	12683	07 Feb 07	CCD-379: Reworked more changes to the build and build_completed.xml after review	+8 -9	andrea	trunk
☐	12680	06 Feb 07	CCD-379: re-org of tutorial	+29 -16	andrea	trunk
☐	12631	14 Jan 07	NONE: merging Clover2 back to trunk	+9 -5	Brendan Humphreys	trunk
☐	11666	17 Aug 05	upgrade to proguard 3.3.2, obfuscate clover for Ant, more re-arranging	+31	Brendan Humphreys	trunk
☐	8472	20 May 03	7338: Removed env and env.ant_home refs	+2 -3	kim	trunk

NONE: show junit output in the tutorial

Author: Nick Pellow
Revision: 23874
Changeset: 23874
Date: 17:42 26 November 2008 (6 months)
Branch point for: clover-2.4.x clover-2.5.x clover-distributed fedex11-mstudman tag:clover-ant-2.4.2
Copied to: branches/clover-2.4.x/trunk/tutorial/build.xml 31720
Tags: clover-ant-2.4.2 idea7-2.4.2-v20081224113000 eclipse-2.4.2.v20090112202100 clover-ant-2.5.0 idea7-2.5.0.v20090511090000 eclipse-2.5.0.v20090511090000
Properties: svn:eol-style = native
svn:keywords = Author Date Id Revision

Browsing through a Repository

Once you have selected a repository, you can drill down into a subdirectory using the directory tree in the left-hand column.

You can use the '**Info**' icon at the extreme left of the screen to view the following repository details:

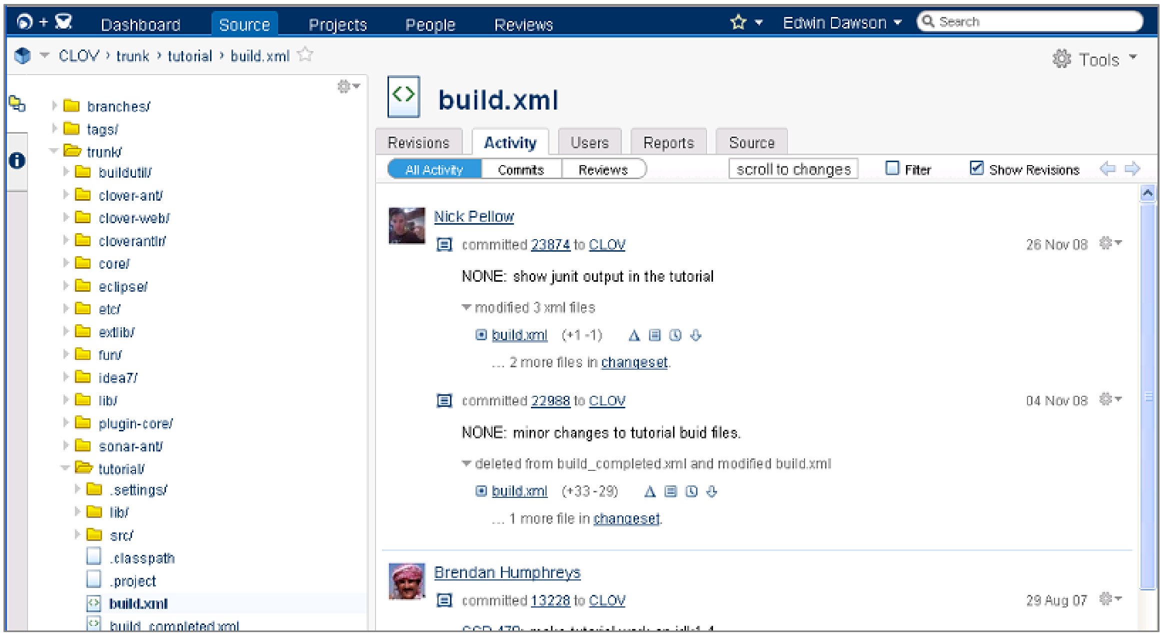
- '**Statistics**' — Total number of committers; last commit; commits this week; total Lines of Code (Loc); change in LoC this week.
- '**Line History**' — This graph shows the total line-count of MAIN or trunk over time for this directory subtree. This line-count does not include binary files, but does include every other file. If you have a branch-constraint specified, then the line-count history of that branch is also shown.
- '**Commit Activity**'
- '**Most Active Committers**' — Over the previous 90 days
- '**Subversion Details**'

(The '**Folder**' icon will toggle you back to the directory tree.)

At the right of the directory tree, the '**Tools**' menu allows you to:

- 'Watch' (ie. receive email notifications about changes to) the current repository/branch/folder.
- Subscribe to an RSS feed of changes to the current repository/branch/folder/file.

Screenshot: Browsing a Repository - Folder View



Nick Pellow

committed 23874 to CLOV 26 Nov 08

NONE: show junit output in the tutorial

modified 3 xml files

build.xml (+1 -1)

... 2 more files in changeset.

committed 22988 to CLOV 04 Nov 08

NONE: minor changes to tutorial buid files.

deleted from build_completed.xml and modified build.xml

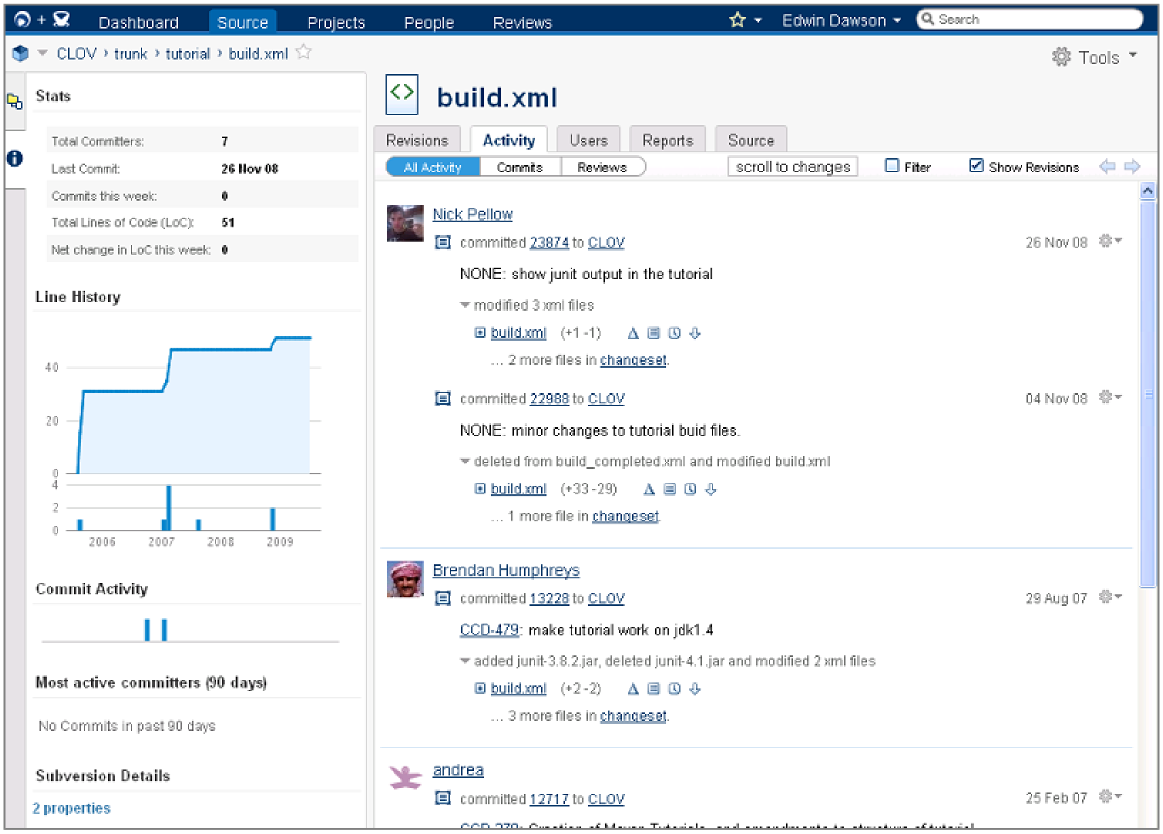
build.xml (+33 -29)

... 1 more file in changeset.

Brendan Humphreys

committed 13228 to CLOV 29 Aug 07

Screenshot: Browsing a Repository - Statistics View



Total Committers:	7
Last Commit:	26 Nov 08
Commits this week:	0
Total Lines of Code (LoC):	51
Net change in LoC this week:	0

Most active committers (90 days)

No Commits in past 90 days

Subversion Details

2 properties

Nick Pellow

committed 23874 to CLOV 26 Nov 08

NONE: show junit output in the tutorial

modified 3 xml files

build.xml (+1 -1)

... 2 more files in changeset.

committed 22988 to CLOV 04 Nov 08

NONE: minor changes to tutorial buid files.

deleted from build_completed.xml and modified build.xml

build.xml (+33 -29)

... 1 more file in changeset.

Brendan Humphreys

committed 13228 to CLOV 29 Aug 07

CCD-479: make tutorial work on jdk1.4

added junit-3.8.2.jar, deleted junit-4.1.jar and modified 2 xml files

build.xml (+2 -2)

... 3 more files in changeset.

andrea

committed 12717 to CLOV 25 Feb 07

Viewing the Changelog

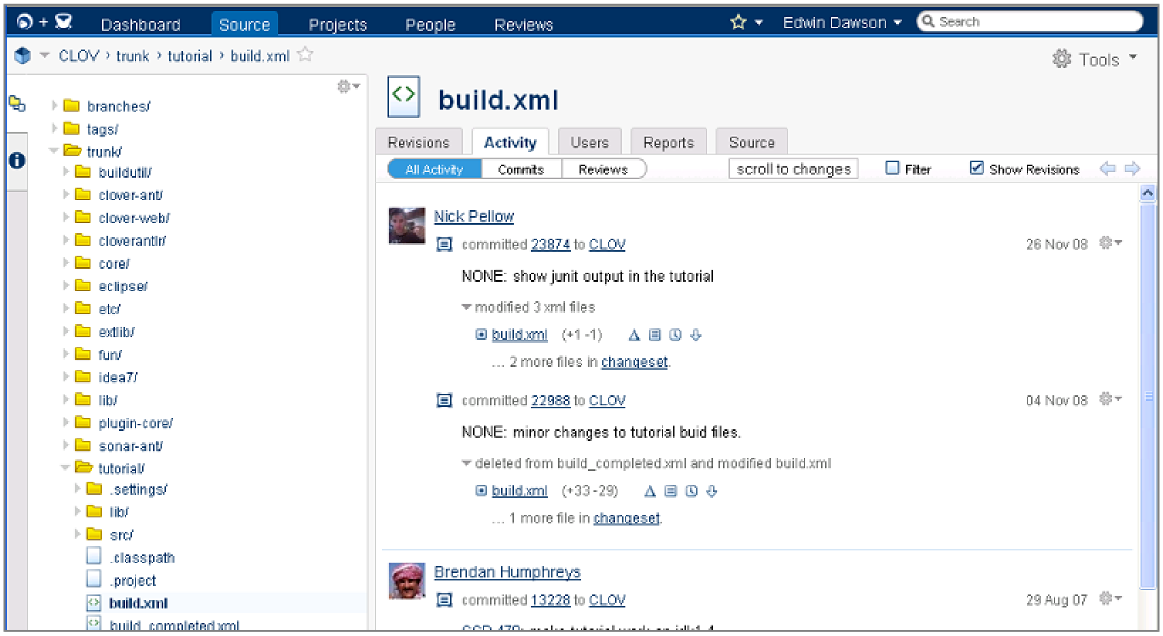
When browsing a repository, the '**Activity**' sub-tab in the right-hand column displays the changes made to your repository/branch/directory/file (ordered with the most recent first). The '**Activity**' sub-tab provides the following options:

- **All Activity** — The default view, showing commits, reviews¹ and JIRA issues².
- **Commits** — Shows commits in the activity stream.
- **Reviews**¹ — Shows review activity in the activity stream.
- **Scroll to Changeset** — Opens the changeset ID specified in the text field (press Enter to carry out the action).
- **Filter** *(see below)* — Applies constraints to the current activity stream.

- **Show Revisions** — If this is selected, then changeset items are automatically expanded to show modified files.
- **Earlier Activity** (Left Arrow icon) — Loads a page of earlier activity.
- **Later Activity** (Right Arrow icon) — Loads a page of later activity.

[1] If you are using Crucible
[2] If you are using JIRA

Screenshot: Viewing the 'Activity' sub-tab



Nick Pellow
committed 23874 to CLOV 26 Nov 08
NONE: show junit output in the tutorial
modified 3 xml files
build.xml (+1 -1)
... 2 more files in changeset.

committed 22988 to CLOV 04 Nov 08
NONE: minor changes to tutorial buid files.
deleted from build_completed.xml and modified build.xml
build.xml (+33 -29)
... 1 more file in changeset.

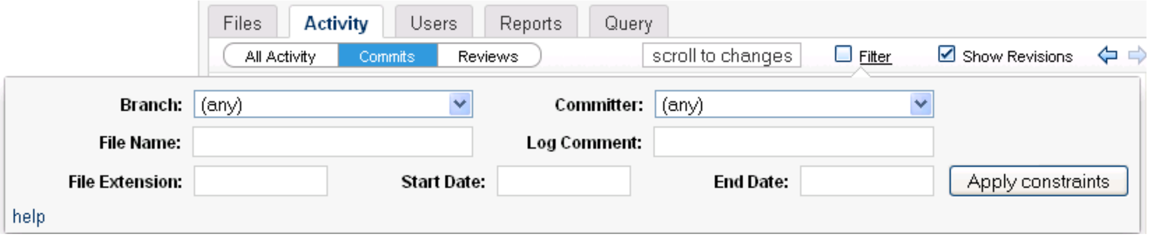
Brendan Humphreys
committed 13228 to CLOV 29 Aug 07

Filter mode

You can specify a '**Filter**' to control the information that is shown in the '**Activity**' sub-tab:

Constraint	Description
Branch	Shows only changesets on that branch.
Committer	Shows only changesets checked in by the given committer/author.
File Name	Shows only changesets that contain a given file.
Log Comment	Shows only changesets where the commit comment matches the given text.
Start Date	Shows only changesets created on or after that date. Must be of the form 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-DD, YYYY-MM or YYYY (you can use '/' instead of '-').

Screenshot: Using the Filter

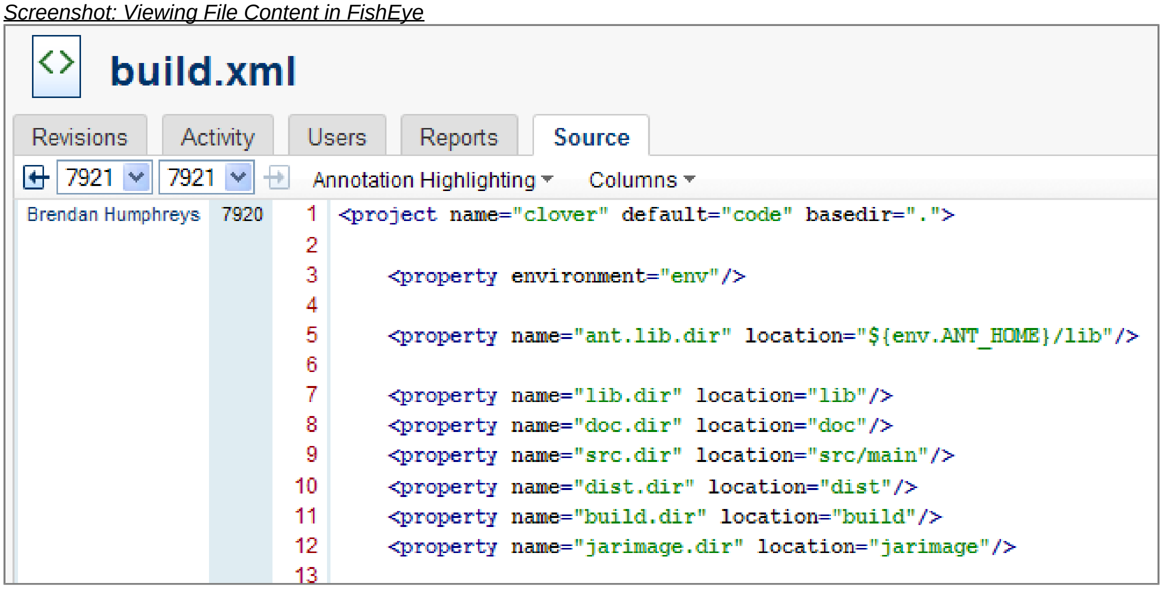


Branch: (any)
Committer: (any)
File Name:
Log Comment:
File Extension:
Start Date:
End Date:

Viewing File Content

When browsing a repository, clicking the 'Source' sub-tab in the right-hand column will display the contents of a file.

Screenshot: Viewing File Content in FishEye



```
<project name="clover" default="code" basedir=".">

    <property environment="env"/>

    <property name="ant.lib.dir" location="${env.ANT_HOME}/lib"/>

    <property name="lib.dir" location="lib"/>
    <property name="doc.dir" location="doc"/>
    <property name="src.dir" location="src/main"/>
    <property name="dist.dir" location="dist"/>
    <property name="build.dir" location="build"/>
    <property name="jarimage.dir" location="jarimage"/>
```

Arbitrary Diffs

The numbered drop-down options in the toolbar allow you to request a diff between any two revisions of the selected file. You can use revision numbers or tag names.

Annotation Highlighting

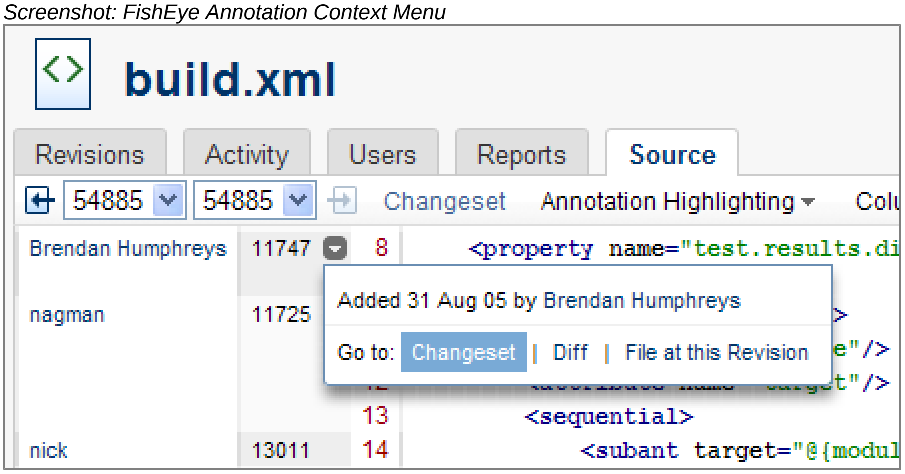
Annotation highlighting can be configured via a drop-down menu button on the toolbar. You can choose from one of the following three options:

- Age,
- Author,
- None.

Annotation Context Menu

You can drill down into the history of any specific 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



Added 31 Aug 05 by Brendan Humphreys

```
        <property name="test.results.di
            <sequential>
                <subant target="@{modul
```

Columns

You can configure the amount of information displayed on the page's columns. You can toggle the following items on and off:

- Author(s),
- Revision(s),
- Line Numbers.

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.

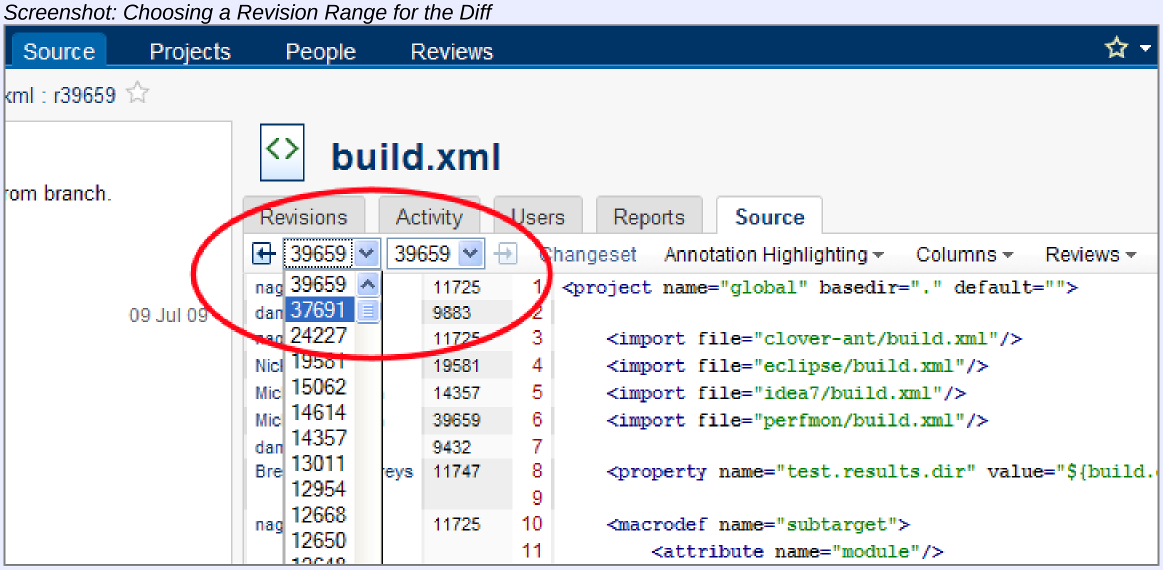
On this page:

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.

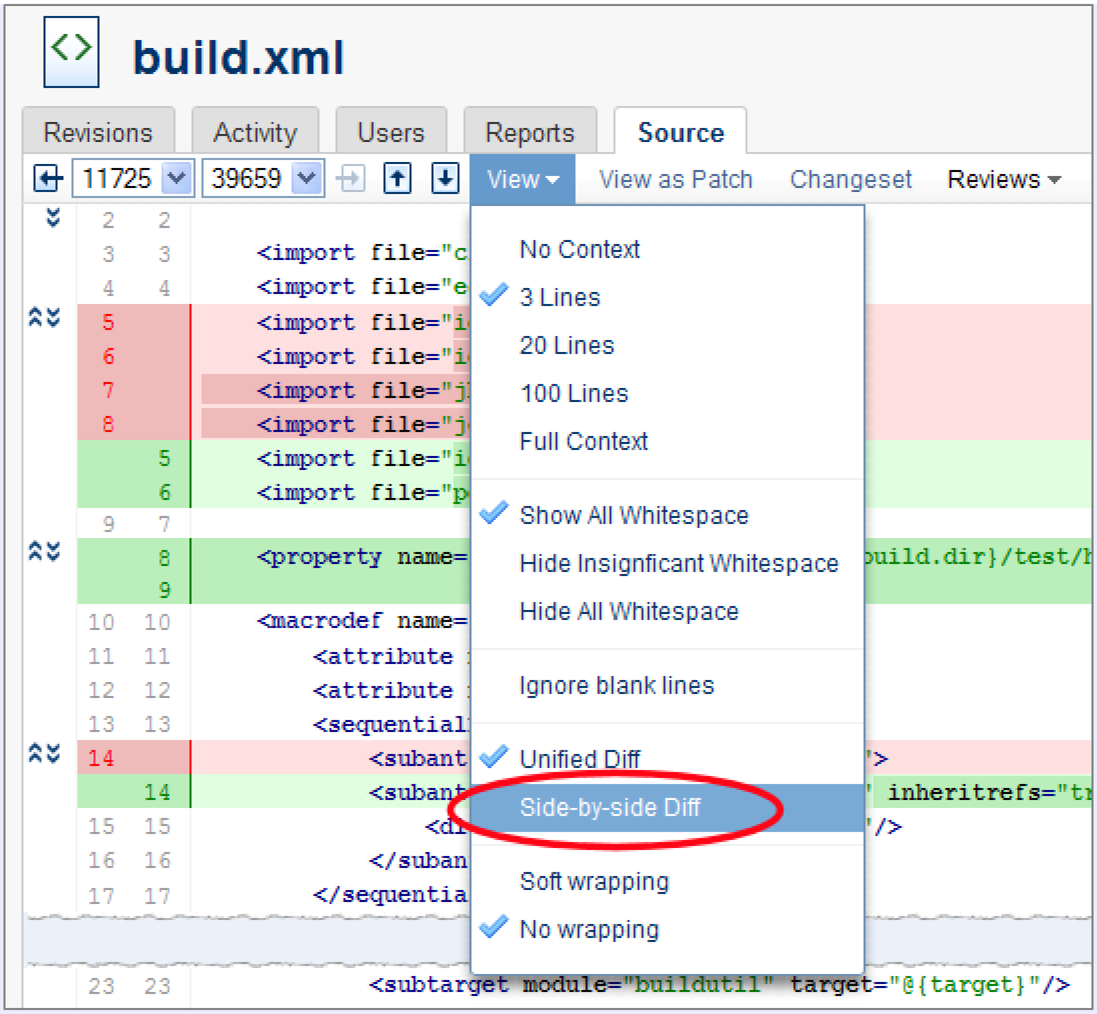
Screenshot: Choosing a Revision Range for the Diff



```
<project name="global" basedir="." default="">

    <import file="clover-ant/build.xml"/>
    <import file="eclipse/build.xml"/>
    <import file="idea7/build.xml"/>
    <import file="perfmon/build.xml"/>

    <property name="test.results.dir" value="${build.

    <macrodef name="subtarget">
        <attribute name="module"/>
```

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

Screenshot: Choosing Side by Side Diff Mode

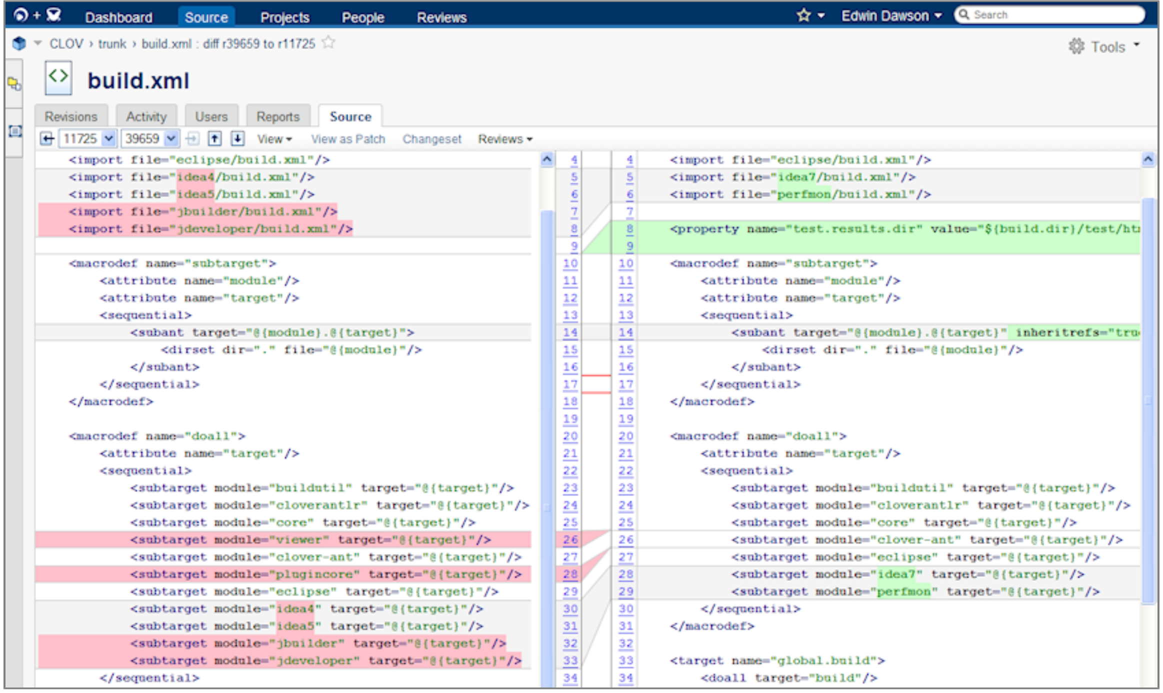


```
    <import file="c
    <import file="e
    <import file="i
    <import file="i
    <import file="j
    <import file="j
    <import file="i
    <import file="p

    <property name=                              uild.dir}/test/h

    <macrodef name=
        <attribute
        <attribute
        <sequential
            <subant                              '>
            <subant                              ' inheritrefs="tr
                <d                               '/>
            </suban
        </sequentia

            <subtarget module="buildutil" target="@{target}"/>
```

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



```
    <import file="eclipse/build.xml"/>
    <import file="idea4/build.xml"/>
    <import file="idea5/build.xml"/>
    <import file="jbuilder/build.xml"/>
    <import file="jdeveloper/build.xml"/>

    <macrodef name="subtarget">
        <attribute name="module"/>
        <attribute name="target"/>
        <sequential>
            <subant target="@{module}.@{target}">
                <dirset dir="." file="@{module}"/>
            </subant>
        </sequential>
    </macrodef>

    <macrodef name="doall">
        <attribute name="target"/>
        <sequential>
            <subtarget module="buildutil" target="@{target}"/>
            <subtarget module="cloverantlr" target="@{target}"/>
            <subtarget module="core" target="@{target}"/>
            <subtarget module="viewer" target="@{target}"/>
            <subtarget module="clover-ant" target="@{target}"/>
            <subtarget module="plugincore" target="@{target}"/>
            <subtarget module="eclipse" target="@{target}"/>
            <subtarget module="idea4" target="@{target}"/>
            <subtarget module="idea5" target="@{target}"/>
            <subtarget module="jbuilder" target="@{target}"/>
            <subtarget module="jdeveloper" target="@{target}"/>
        </sequential>
```

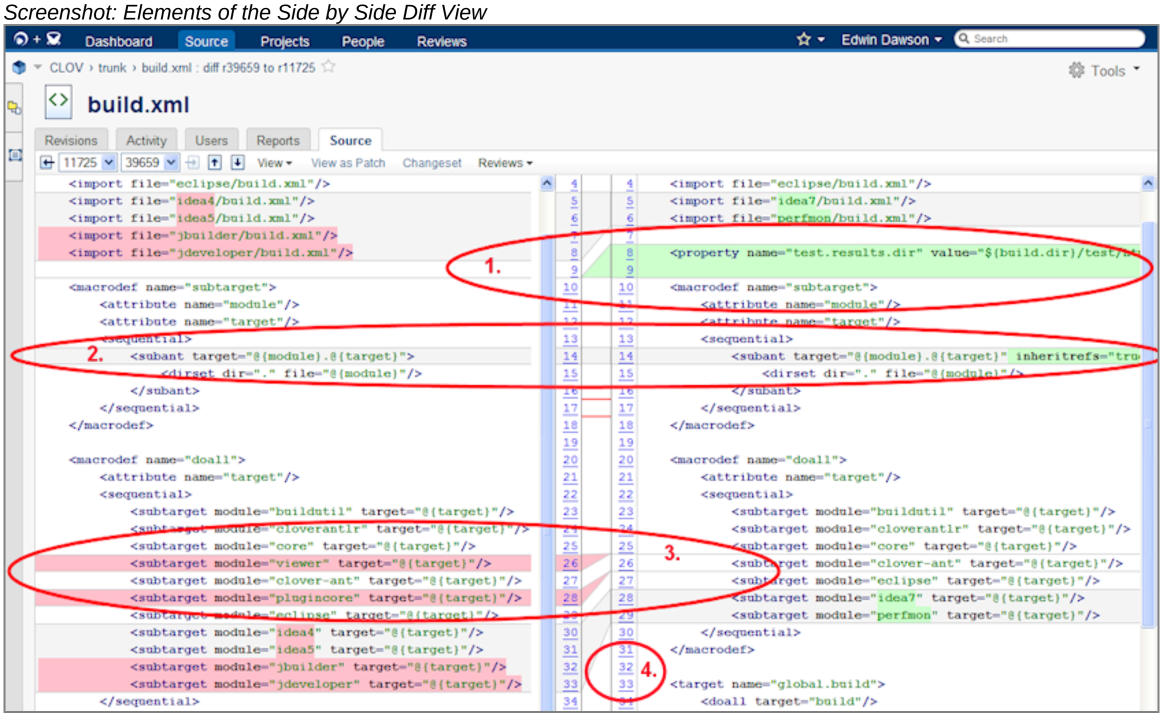
```
    <import file="eclipse/build.xml"/>
    <import file="idea7/build.xml"/>
    <import file="perfmon/build.xml"/>

    <property name="test.results.dir" value="${build.dir}/test/ht

    <macrodef name="subtarget">
        <attribute name="module"/>
        <attribute name="target"/>
        <sequential>
            <subant target="@{module}.@{target}" inheritrefs="tru
                <dirset dir="." file="@{module}"/>
            </subant>
        </sequential>
    </macrodef>

    <macrodef name="doall">
        <attribute name="target"/>
        <sequential>
            <subtarget module="buildutil" target="@{target}"/>
            <subtarget module="cloverantlr" target="@{target}"/>
            <subtarget module="core" target="@{target}"/>
            <subtarget module="clover-ant" target="@{target}"/>
            <subtarget module="eclipse" target="@{target}"/>
            <subtarget module="idea7" target="@{target}"/>
            <subtarget module="perfmon" target="@{target}"/>
        </sequential>
    </macrodef>

    <target name="global.build">
        <doall target="build"/>
```

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



```
    <import file="eclipse/build.xml"/>
    <import file="idea4/build.xml"/>
    <import file="idea5/build.xml"/>
    <import file="jbuilder/build.xml"/>
    <import file="jdeveloper/build.xml"/>

    <macrodef name="subtarget">
        <attribute name="module"/>
        <attribute name="target"/>
        <sequential>
            <subant target="@{module}.@{target}">
                <dirset dir="." file="@{module}"/>
            </subant>
        </sequential>
    </macrodef>

    <macrodef name="doall">
        <attribute name="target"/>
        <sequential>
            <subtarget module="buildutil" target="@{target}"/>
            <subtarget module="cloverantlr" target="@{target}"/>
            <subtarget module="core" target="@{target}"/>
            <subtarget module="viewer" target="@{target}"/>
            <subtarget module="clover-ant" target="@{target}"/>
            <subtarget module="plugincore" target="@{target}"/>
            <subtarget module="eclipse" target="@{target}"/>
            <subtarget module="idea4" target="@{target}"/>
            <subtarget module="idea5" target="@{target}"/>
            <subtarget module="jbuilder" target="@{target}"/>
            <subtarget module="jdeveloper" target="@{target}"/>
        </sequential>
```

```
    <import file="eclipse/build.xml"/>
    <import file="idea7/build.xml"/>
    <import file="perfmon/build.xml"/>

    <property name="test.results.dir" value="${build.dir}/test/ht

    <macrodef name="subtarget">
        <attribute name="module"/>
        <attribute name="target"/>
        <sequential>
            <subant target="@{module}.@{target}" inheritrefs="tru
                <dirset dir="." file="@{module}"/>
            </subant>
        </sequential>
    </macrodef>

    <macrodef name="doall">
        <attribute name="target"/>
        <sequential>
            <subtarget module="buildutil" target="@{target}"/>
            <subtarget module="cloverantlr" target="@{target}"/>
            <subtarget module="core" target="@{target}"/>
            <subtarget module="clover-ant" target="@{target}"/>
            <subtarget module="eclipse" target="@{target}"/>
            <subtarget module="idea7" target="@{target}"/>
            <subtarget module="perfmon" target="@{target}"/>
        </sequential>
    </macrodef>

    <target name="global.build">
        <doall target="build"/>
```

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

When browsing a repository, the '**Revisions**' sub-tab in the right-hand column displays the different revisions of a file.

Arbitrary Diffs

The '**Diff ... and ...**' boxes allow you to request a diff between any two revisions of the selected file. You can use revision numbers or tag names.

Show and Hide Related Revisions

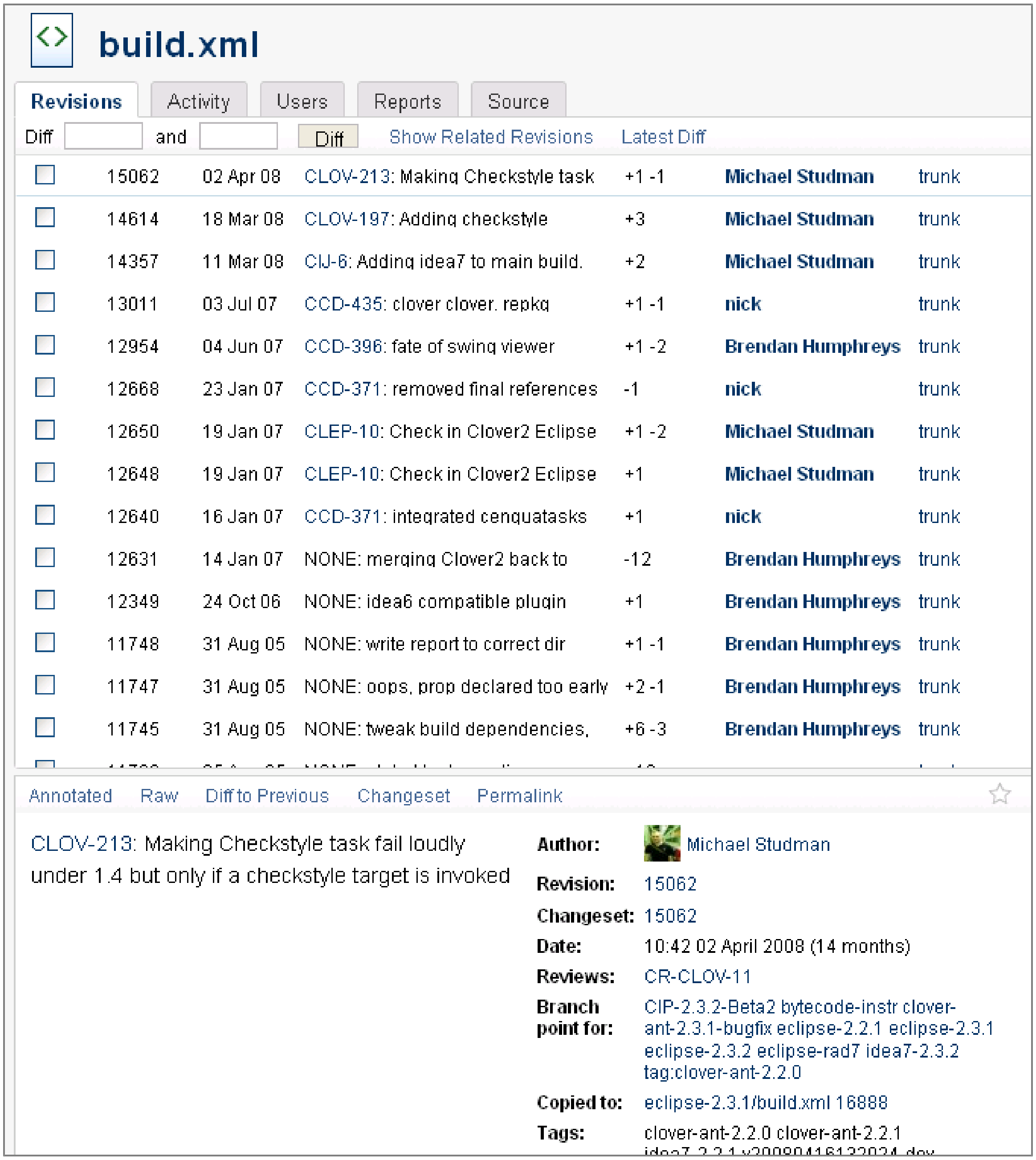
Click the '**Show Related Revisions**' link to show all of the current file's counterparts (revisions) of the same name, within your repository's logical structure. Click '**Hide Related Revisions**' to toggle the display back to default (showing only revisions from the current logical structure).

ⓘ 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 especially applies to Subversion's branching and tagging directory structure.

Latest Diff

Click the '**Latest Diff**' link to open the diff (source view) from the current version of this file to the previous version.

Screenshot: Revisions



□	15062	02 Apr 08	CLOV-213: Making Checkstyle task	+1 -1	Michael Studman	trunk
□	14614	18 Mar 08	CLOV-197: Adding checkstyle	+3	Michael Studman	trunk
□	14357	11 Mar 08	CIJ-6: Adding idea7 to main build.	+2	Michael Studman	trunk
□	13011	03 Jul 07	CCD-435: clover clover. repkg	+1 -1	nick	trunk
□	12954	04 Jun 07	CCD-396: fate of swing viewer	+1 -2	Brendan Humphreys	trunk
□	12668	23 Jan 07	CCD-371: removed final references	-1	nick	trunk
□	12650	19 Jan 07	CLEP-10: Check in Clover2 Eclipse	+1 -2	Michael Studman	trunk
□	12648	19 Jan 07	CLEP-10: Check in Clover2 Eclipse	+1	Michael Studman	trunk
□	12640	16 Jan 07	CCD-371: integrated cenquatasks	+1	nick	trunk
□	12631	14 Jan 07	NONE: merging Clover2 back to	-12	Brendan Humphreys	trunk
□	12349	24 Oct 06	NONE: idea6 compatible plugin	+1	Brendan Humphreys	trunk
□	11748	31 Aug 05	NONE: write report to correct dir	+1 -1	Brendan Humphreys	trunk
□	11747	31 Aug 05	NONE: oops, prop declared too early	+2 -1	Brendan Humphreys	trunk
□	11745	31 Aug 05	NONE: tweak build dependencies,	+6 -3	Brendan Humphreys	trunk

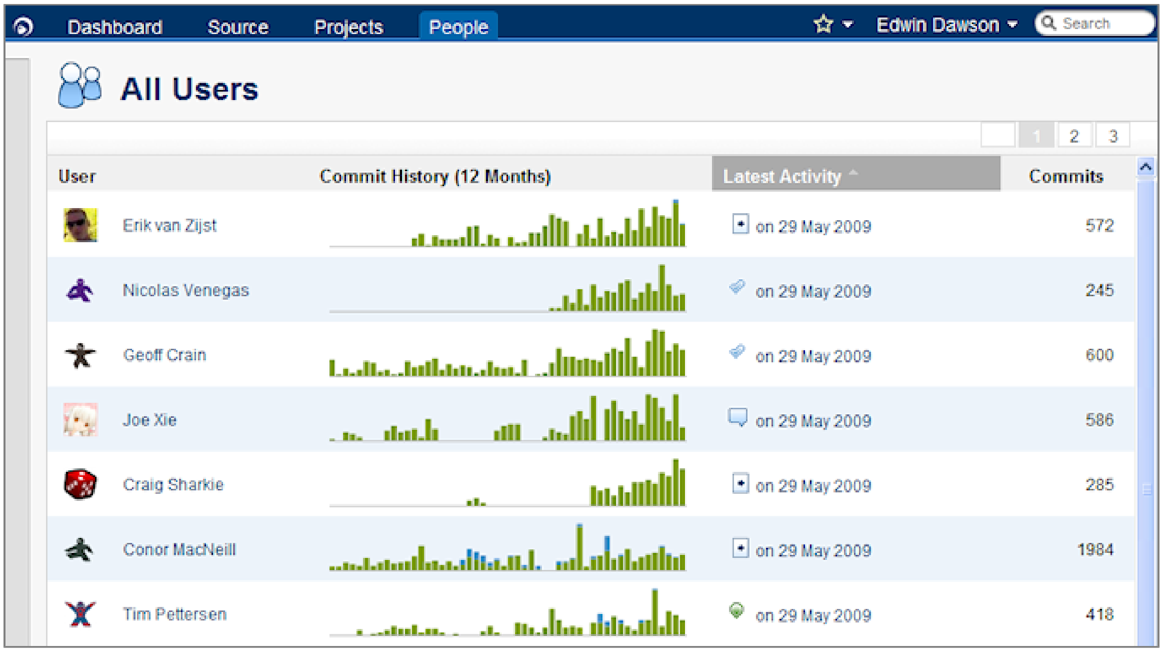
CLOV-213: Making Checkstyle task fail loudly under 1.4 but only if a checkstyle target is invoked

Author:	Michael Studman
Revision:	15062
Changeset:	15062
Date:	10:42 02 April 2008 (14 months)
Reviews:	CR-CLOV-11
Branch point for:	CIP-2.3.2-Beta2 bytecode-instr clover-ant-2.3.1-bugfix eclipse-2.2.1 eclipse-2.3.1 eclipse-2.3.2 eclipse-rad7 idea7-2.3.2 tag:clover-ant-2.2.0
Copied to:	eclipse-2.3.1/build.xml 16888
Tags:	clover-ant-2.2.0 clover-ant-2.2.1 idea7-2.2.1 v20080416132024-dev

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



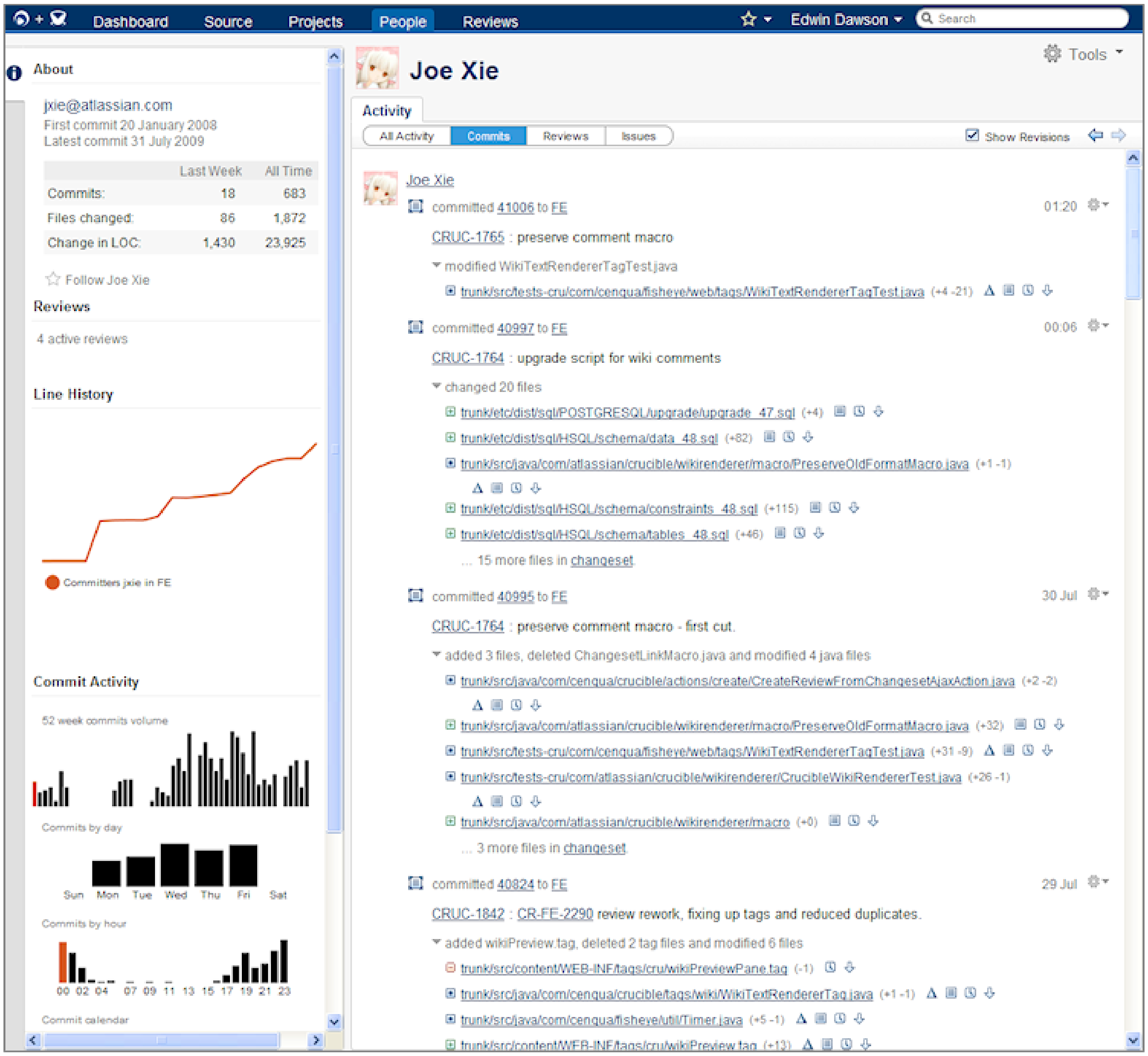
All Users

User	Commit History (12 Months)	Latest Activity	Commits
Erik van Zijst		on 29 May 2009	572
Nicolas Venegas		on 29 May 2009	245
Geoff Crain		on 29 May 2009	600
Joe Xie		on 29 May 2009	586
Craig Sharkie		on 29 May 2009	285
Conor MacNeill		on 29 May 2009	1984
Tim Pettersen		on 29 May 2009	418

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



About

jxie@atlassian.com
First commit 20 January 2008
Latest commit 31 July 2009

	Last Week	All Time
Commits:	18	683
Files changed:	86	1,872
Change in LOC:	1,430	23,925

☆ Follow Joe Xie

Reviews

4 active reviews

Joe Xie

Joe Xie

committed 41006 to FE 01:20

CRUC-1765 : preserve comment macro

modified WikiTextRendererTagTest.java

trunk/src/tests-cru/com/cenqua/fisheye/web/tags/WikiTextRendererTagTest.java (+4 -21)

committed 40997 to FE 00:06

CRUC-1764 : upgrade script for wiki comments

changed 20 files

trunk/etc/dist/sql/POSTGRESQL/upgrade/upgrade_47.sql (+4)
trunk/etc/dist/sql/HSQL/schema/data_48.sql (+82)
trunk/src/java/com/atlassian/crucible/wikirenderer/macro/PreserveOldFormatMacro.java (+1 -1)
trunk/etc/dist/sql/HSQL/schema/constraints_48.sql (+115)
trunk/etc/dist/sql/HSQL/schema/tables_48.sql (+46)
... 15 more files in changeset.

committed 40995 to FE 30 Jul

CRUC-1764 : preserve comment macro - first cut.

added 3 files, deleted ChangesetLinkMacro.java and modified 4 java files

trunk/src/java/com/cenqua/crucible/actions/create/CreateReviewFromChangesetAjaxAction.java (+2 -2)
trunk/src/java/com/atlassian/crucible/wikirenderer/macro/PreserveOldFormatMacro.java (+32)
trunk/src/tests-cru/com/cenqua/fisheye/web/tags/WikiTextRendererTagTest.java (+31 -9)
trunk/src/tests-cru/com/atlassian/crucible/wikirenderer/CrucibleWikiRendererTest.java (+26 -1)
trunk/src/java/com/atlassian/crucible/wikirenderer/macro (+0)
... 3 more files in changeset.

committed 40824 to FE 29 Jul

CRUC-1842 : CR-FE-2290 review rework, fixing up tags and reduced duplicates.

added wikiPreview.tag, deleted 2 tag files and modified 6 files

trunk/src/content/WEB-INF/tags/cru/wikiPreviewPane.tag (-1)
trunk/src/java/com/cenqua/crucible/tags/wiki/WikiTextRendererTag.java (+1 -1)
trunk/src/java/com/cenqua/fisheye/util/Timer.java (+5 -1)
trunk/src/content/WEB-INF/tags/cru/wikiPreview.tag (+13)

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.

Searching the Repository

FishEye allows you a number of ways to search through the repository to find particular changesets or files.

Methods of searching are described below:

- Quick Nav
- Quick Search
- Simple Search
- Advanced Search

Quick Nav

Quick Nav results appear as soon as you start typing into the **'Quick Search'** box in FishEye, before you've pressed Enter to visit the full page of search results. Results are weighted by most recent edit date; files edited within the last twelve months are given greater weighting.

Quick Nav supports the following powerful search features:

CamelCase detection
You can type a string like "**UpA**" to quickly find a name such as "**UnplannedArchitecture**". This is a common feature in IDE search systems.

Multiple directory selection
You can type the following:

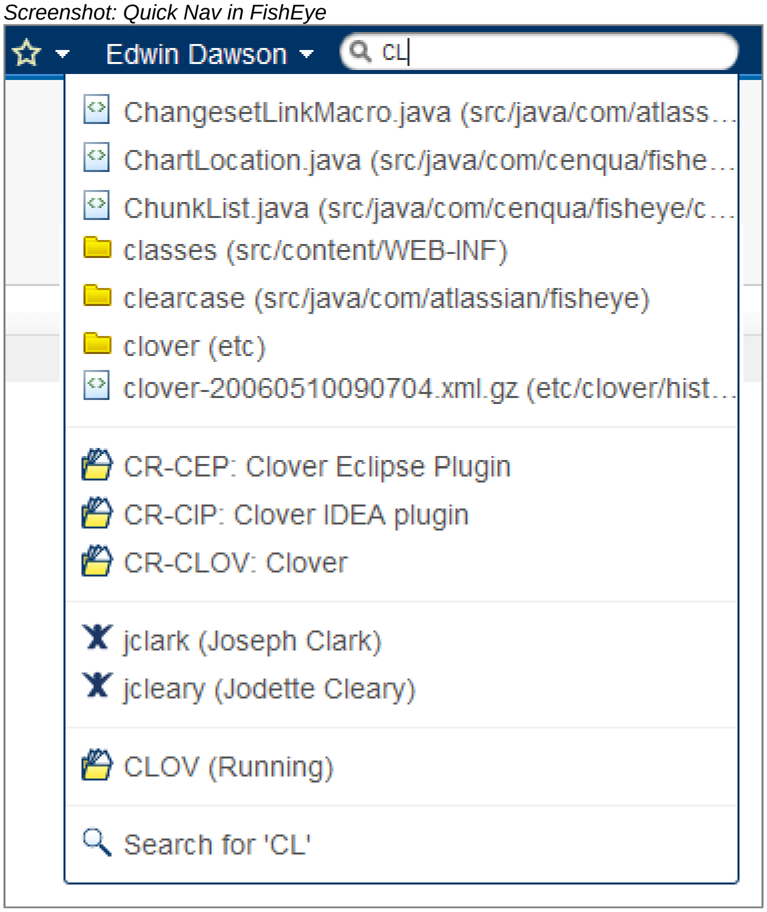
```
common/final/Actions
```

to find a path like this:

```
/src/common/eu/systemworks/specialprojects/final/Actions.java
```

Results constrained to one repository
When you begin your quick search a page by clicking the '**Source**' tab, then choosing a repository (or any page below it), FishEye Quick Nav will only show you results from the repository you're currently browsing. When you're viewing the index of repositories, the Dashboard, or tabs other than the '**Source**' tab, Quick Nav will show you results from all the repositories connected to FishEye.

Screenshot: Quick Nav in FishEye


Edwin Dawson
CL
ChangesetLinkMacro.java (src/java/com/atlass…
ChartLocation.java (src/java/com/cenqua/fishe…
ChunkList.java (src/java/com/cenqua/fisheye/c…
classes (src/content/WEB-INF)
clearcase (src/java/com/atlassian/fisheye)
clover (etc)
clover-20060510090704.xml.gz (etc/clover/hist…
CR-CEP: Clover Eclipse Plugin
CR-CIP: Clover IDEA plugin
CR-CLOV: Clover
jclark (Joseph Clark)
jcleary (Jodette Cleary)
CLOV (Running)
Search for 'CL'

Quick Search

To use this search, enter your search term in the '**Quick Search**' box in the top right hand corner of the FishEye screens.

Screenshot: Quick Search Box

You can search for the following:

- Authors
- Branch names
- Commit comments
- Changeset ids
- Filenames/paths
- File content
- Tags
- Date (YYYY-MM-DD format, or any substring of that)
- Added/Removed diff text. Requires a re-index of the repository.
- Type a CSID to go directly to that page.

Results are sorted by relevance, with limited results per page. Click '**Next**' to load following pages. Search matches inside the results are highlighted in yellow.

Restricting searches by prefixing database field
You can search matches against a given field, by using a search in this format:

```
author:anna
```

This would return all results from the **author** field that match the string 'anna'.

Searches can be specifically restricted to the following available fields:

- Author
- Comment
- Contents
- File (You can use Antglobs)
- Branch
- Tag
- Cs
- Date
- AddedLine (searches diff text)
- RemovedLine (searches diff text).

Searching for discrete strings with precise case
To search for a specific string that appears discretely, with specific capital or non-capital letters, search with quotation marks, as in the following example:

```
"Monkey"
```

This search will ignore occurrences of the string that appear embedded in other strings, have different case, or are part of a path or disk location. The example above would not return "ProjectMonkey", "monkey", or "/zoo/mammals/monkey/archie/".
Note that regular quick searches do not take case into account and phrases cannot be searched in Quick Search at present.

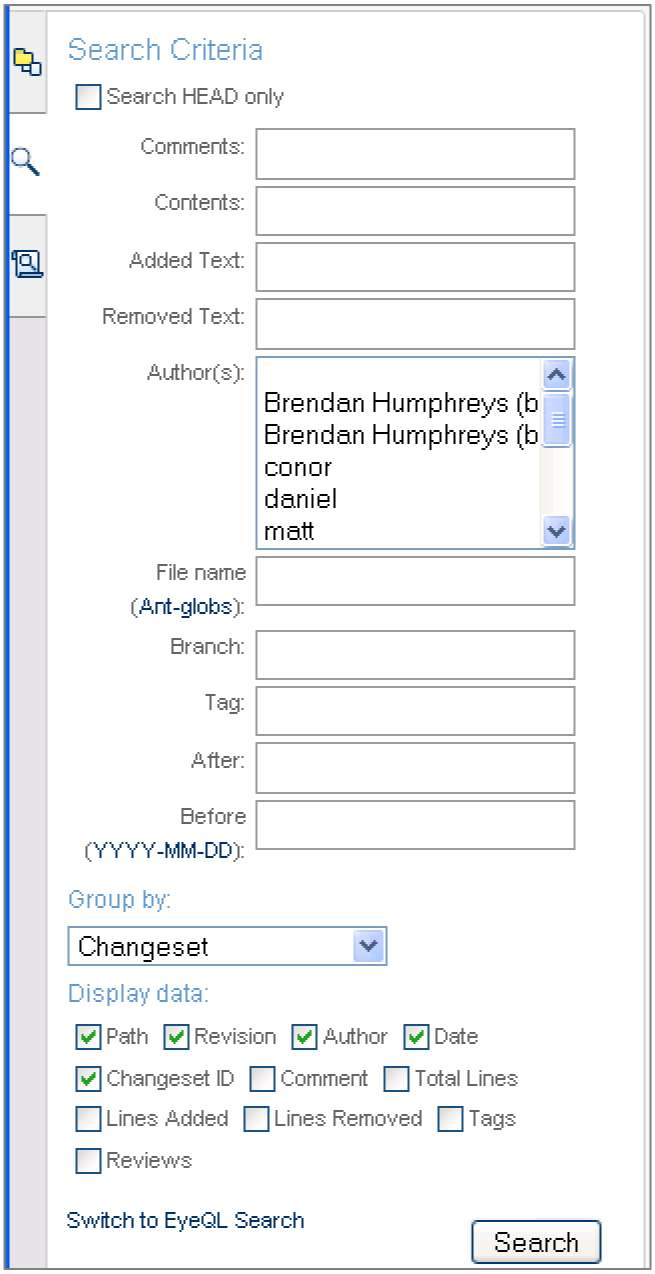
File content Quick Search and Diff Text searching are new features in FishEye 1.6 and will continue to be improved. If you want access to full diff text and historical file contents in the repository, you will need to re-index it.

> **A note about searching multiple repositories:**
> Cross-repository searching has a 5-second limitation on searches, to prevent it from becoming unresponsive and consuming server resources on FishEye instances that have large numbers of repositories. This means that cross-repository quicksearch is not an exhaustive search, and may not include all repositories in a large Fisheye instance. Preference is given to repositories with the most recent changes. For exhaustive searches, you should:
>
> 1. Limit your search to a particular repository, if possible.
> 2. Perform a full search, rather than a QuickSearch.

Simple Search

To access the simple search screen, click the '**Query**' sub-tab when browsing a repository.

Screenshot: FishEye Simple Search panel



Search Criteria

□ Search HEAD only

Comments:

Contents:

Added Text:

Removed Text:

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

File name (Ant-globs):

Branch:

Tag:

After:

Before (YYYY-MM-DD):

Group by: Changeset

Display data:

☑ Path ☑ Revision ☑ Author ☑ Date
☑ Changeset ID □ Comment □ Total Lines
□ Lines Added □ Lines Removed □ Tags
□ Reviews

You can use this search to retrieve a list of changesets/files using the filters that are available. You can search using one or more of the following filters:

- Commit comments
- Contents of files — files must be non-binary, less than 5MB, and located on the trunk/head
- Added text/removed text
- File names/paths — Antglobs can be used
- Authors
- Branch names
- Tag names
- Revision dates before and after.

Results can be grouped by the following:

- Changeset
- Revision
- File
- Directory.

You can choose to include any or all of the following fields in the results:

- Path

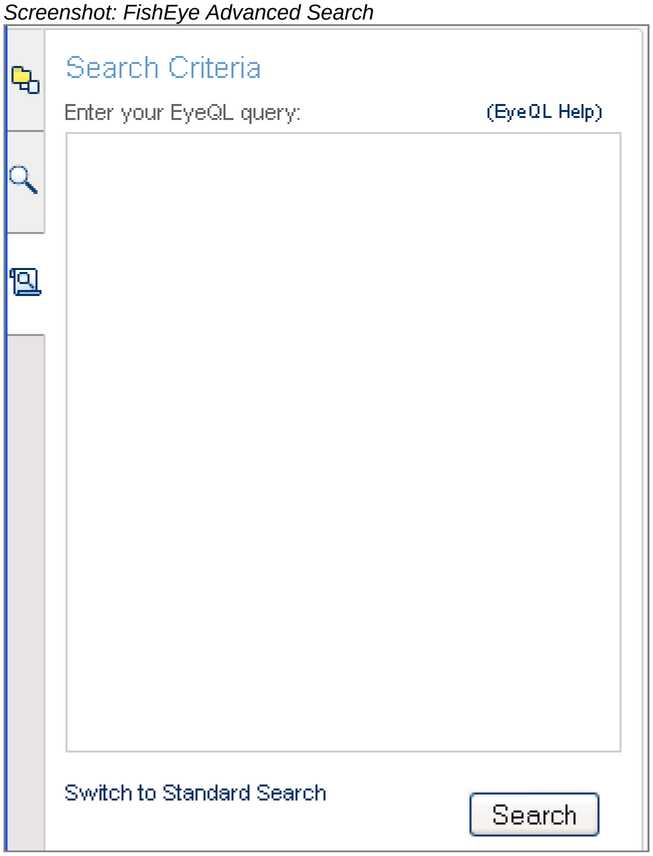
- Revision
- Author
- Date
- Comment
- Changeset
- Total lines
- Total lines added
- Total lines removed
- Tags
- Reviews *(if you are using Crucible)*.

The results are shown in a tabular format. You can link to the search results, and you can save the results to a CSV file.

Advanced Search

In some circumstances the results of a simple search may not be specific enough. Using the advanced search, you can create your own complex searches using FishEye's powerful query language called EyeQL.

Screenshot: FishEye Advanced Search



To do an advanced search, click the '**Switch to EyeQL Search**' link found at the bottom of the Simple Search screen.

> ✓ **Use Simple Search to build your basic query first**
> You can flick between Simple and Advanced Search. The EyeQL statement will contain the basics of the statement and you can adapt it as required.

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

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	Limits the chart to the selected branch, defined as the repository's trunk (unless the directory displayed is a branch directory).	Any branch from the current repository.	Displays the trunk, unless the current directory has been identified as a branch.
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)

ℹ *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

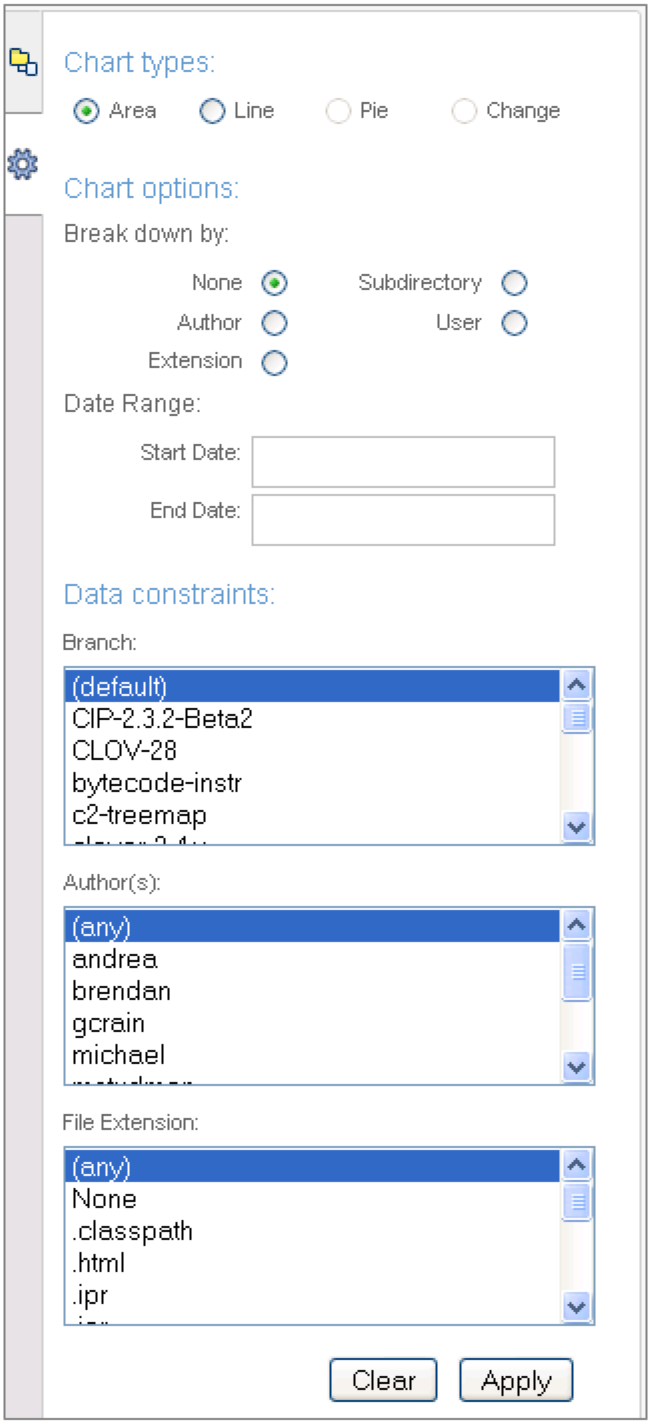


Chart types:

● Area ○ Line ○ Pie ○ Change

Chart options:

Break down by:

None ● Subdirectory ○

Author ○ User ○

Extension ○

Date Range:

Start Date:

End Date:

Data constraints:

Branch:

- (default)
- CIP-2.3.2-Beta2
- CLOV-28
- bytecode-instr
- c2-treemap

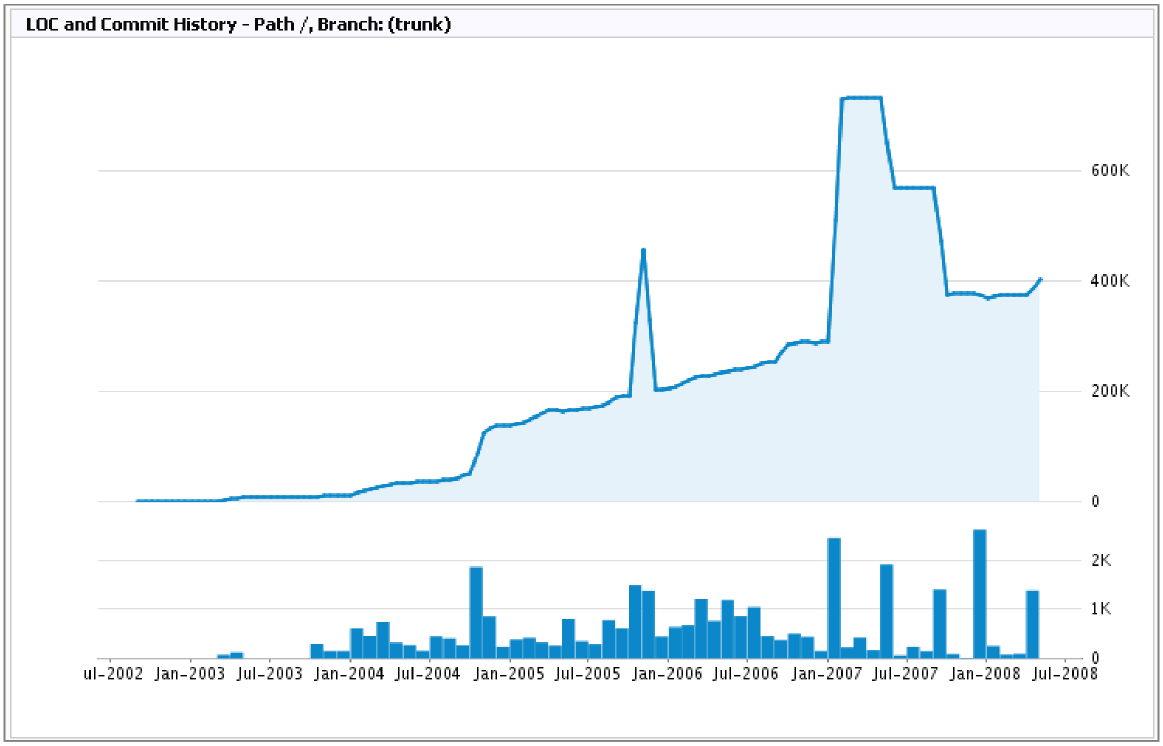
Author(s):

- (any)
- andrea
- brendan
- gcrain
- michael

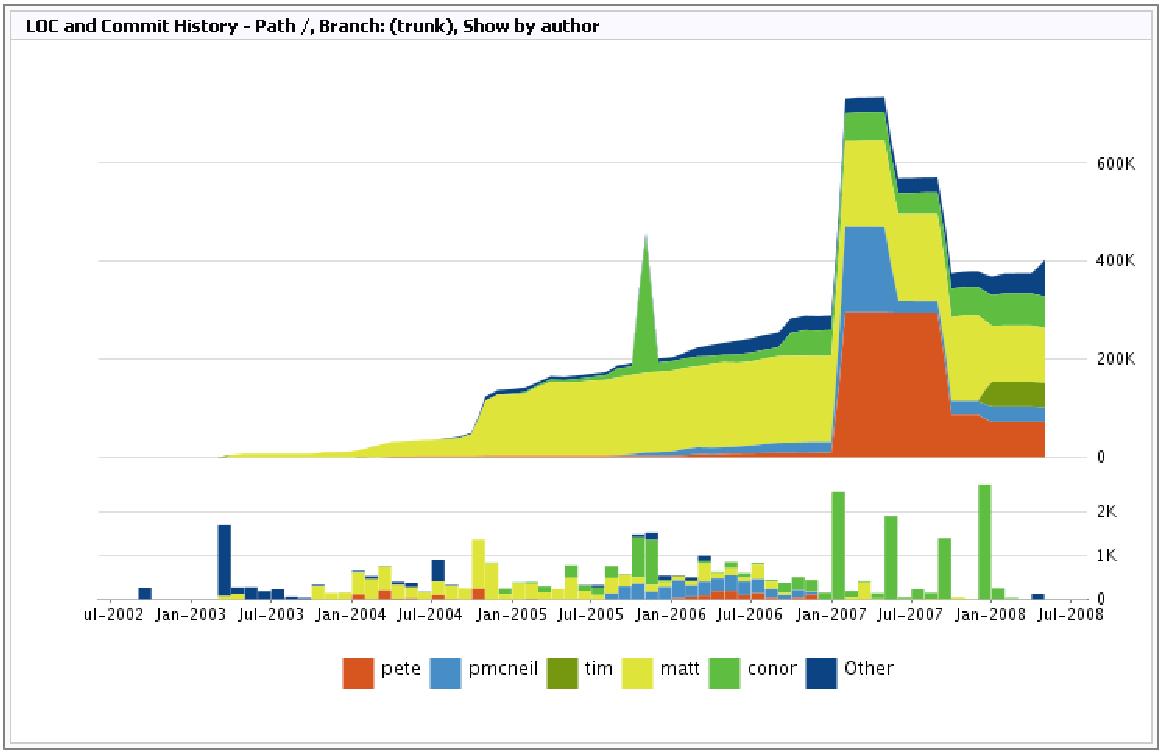
File Extension:

- (any)
- None
- .classpath
- .html
- .ipr

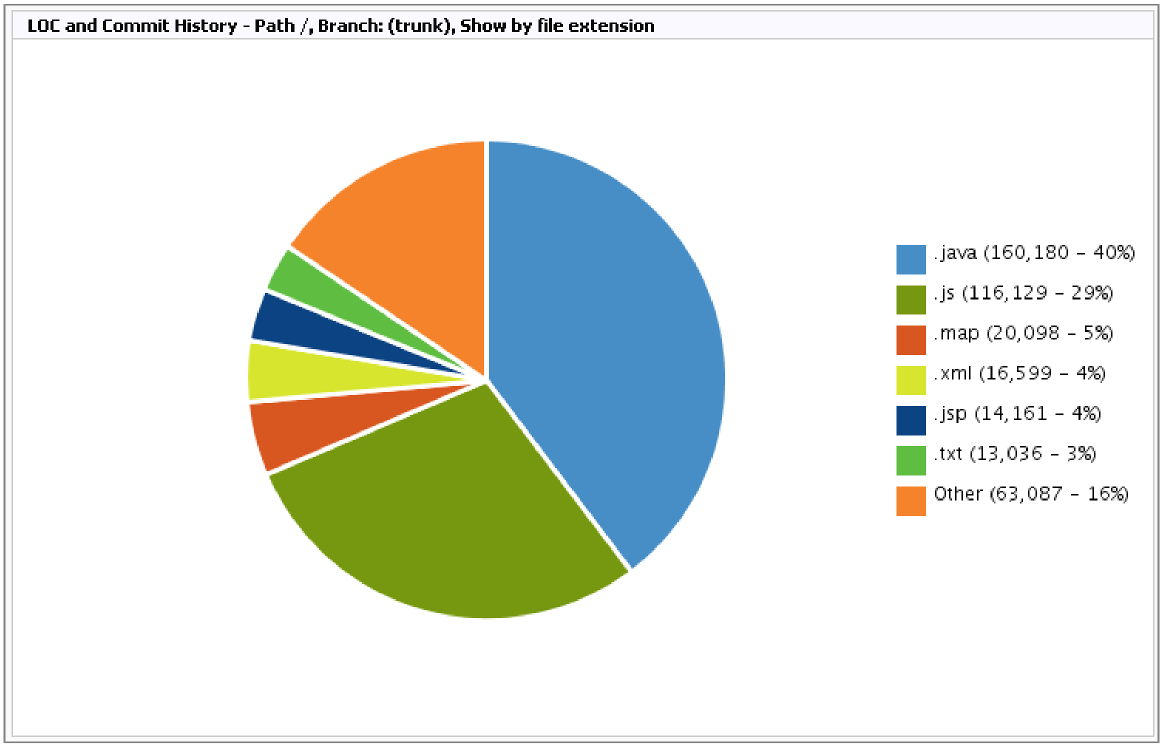
Screenshot: FishEye per-author LOC chart



Screenshot: FishEye per-author LOC chart showing multiple authors



Screenshot: FishEye LOC chart by file extension



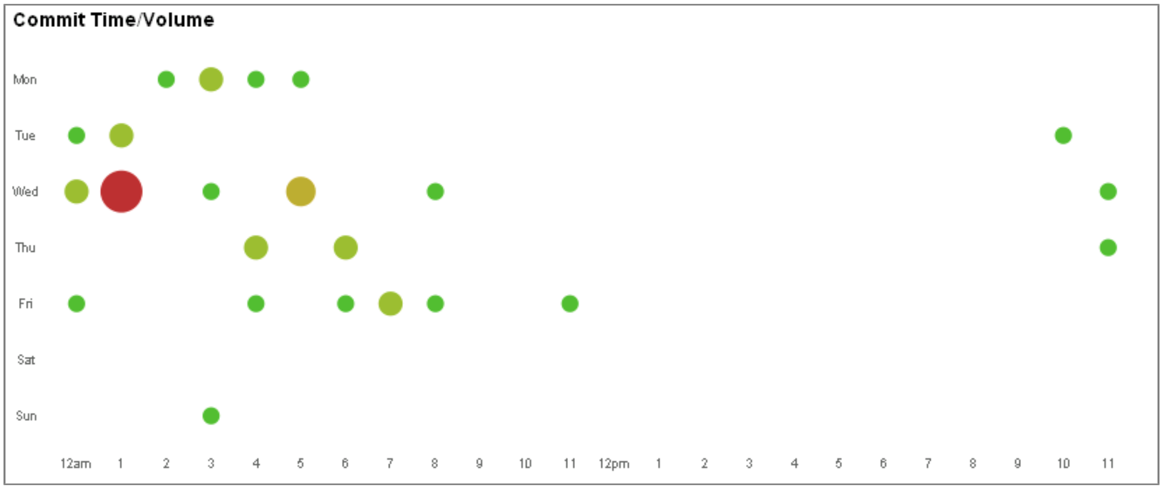
Per-Author Lines of Code Statistics

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.

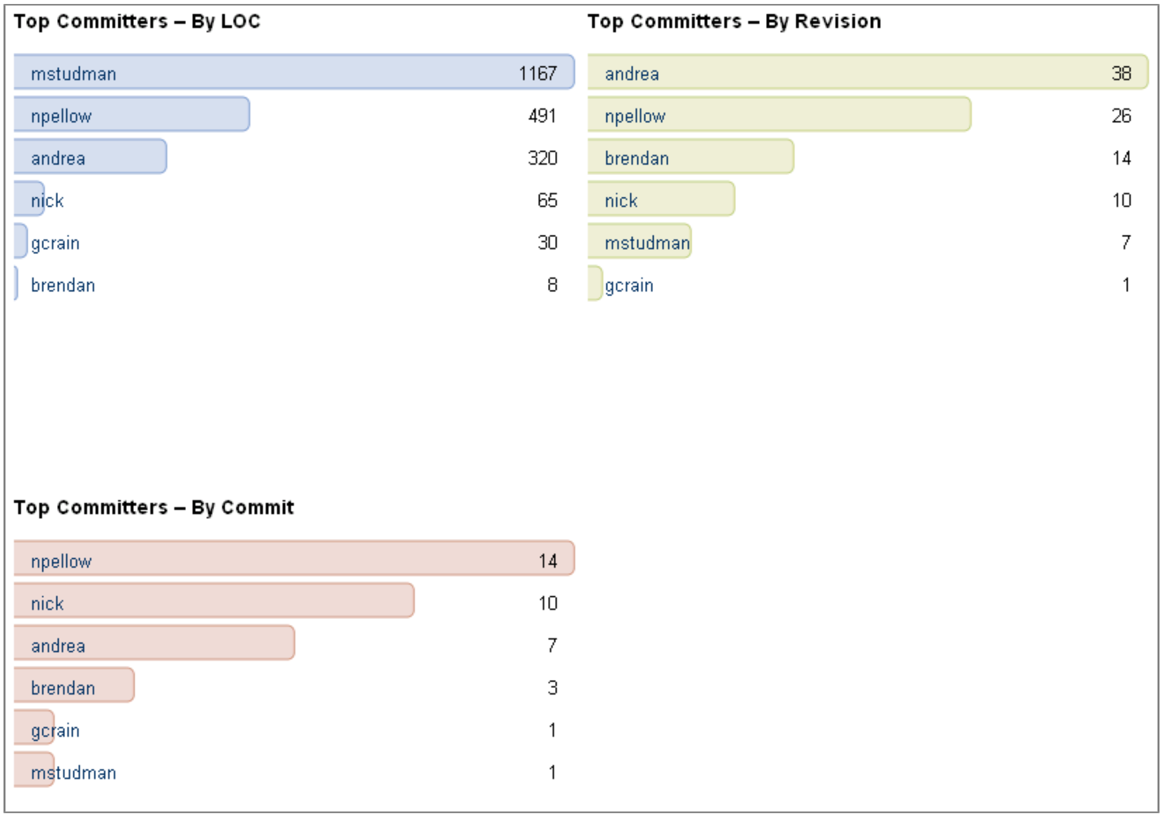
Code Metrics

A number of built-in reports are also provided:

Screenshot: Commit Time/Volume



Screenshot: Top Committers



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.

> ℹ 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.

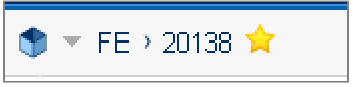
Adding a Person to Your Favourites

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



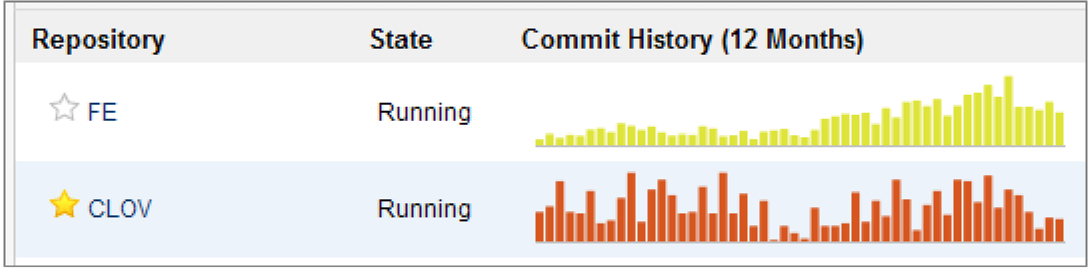
Adding a File or Folder to Your Favourites

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

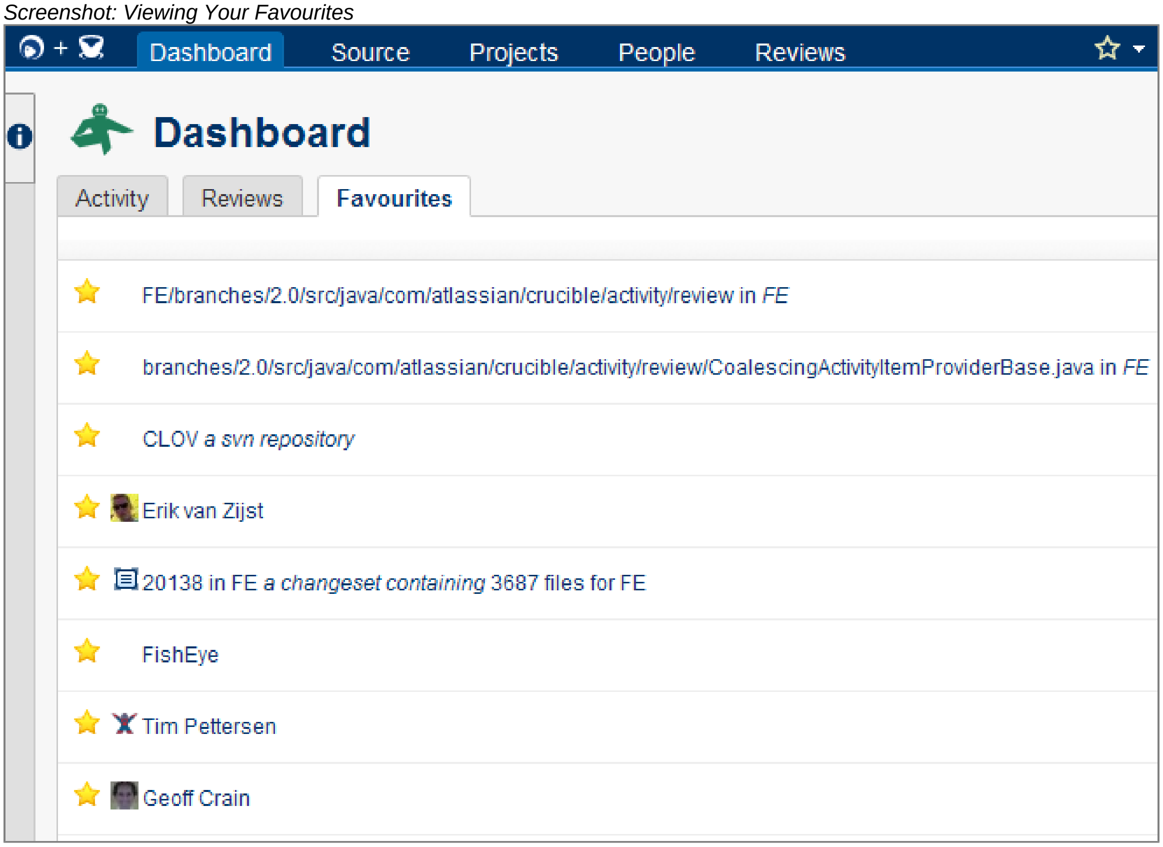


Repository	State	Commit History (12 Months)
☆ FE	Running	
★ CLOV	Running	

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



★ FE/branches/2.0/src/java/com/atlassian/crucible/activity/review in *FE*

★ branches/2.0/src/java/com/atlassian/crucible/activity/review/CoalescingActivityItemProviderBase.java in *FE*

★ CLOV *a svn repository*

★ Erik van Zijst

★ 20138 in FE *a changeset containing* 3687 files for FE

★ FishEye

★ Tim Pettersen

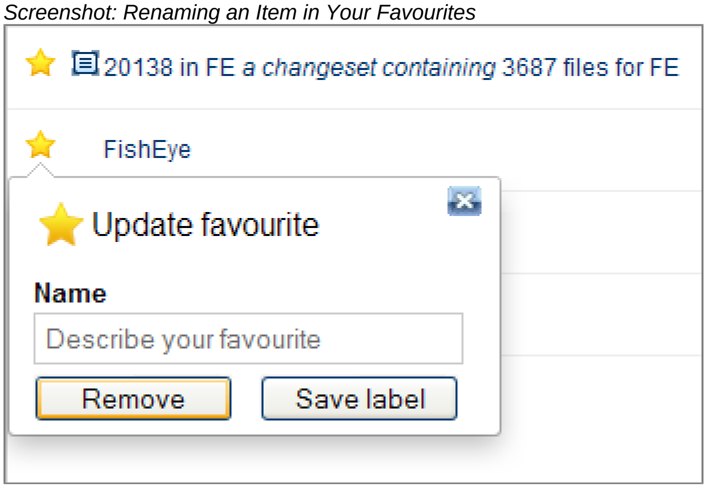
★ Geoff Crain

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

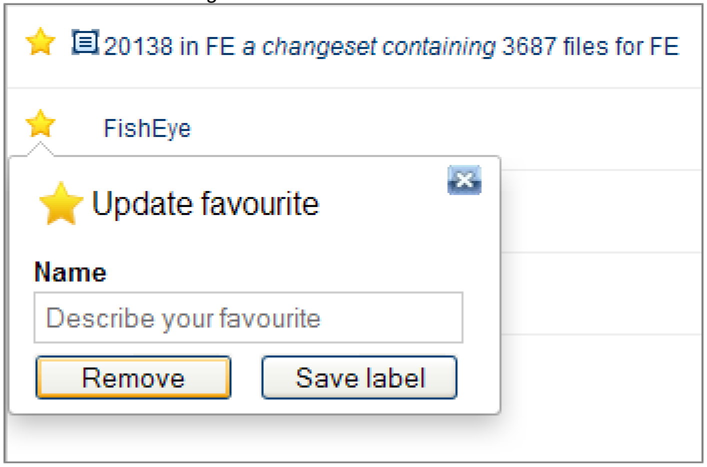
Screenshot: Renaming an Item in Your Favourites



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



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:

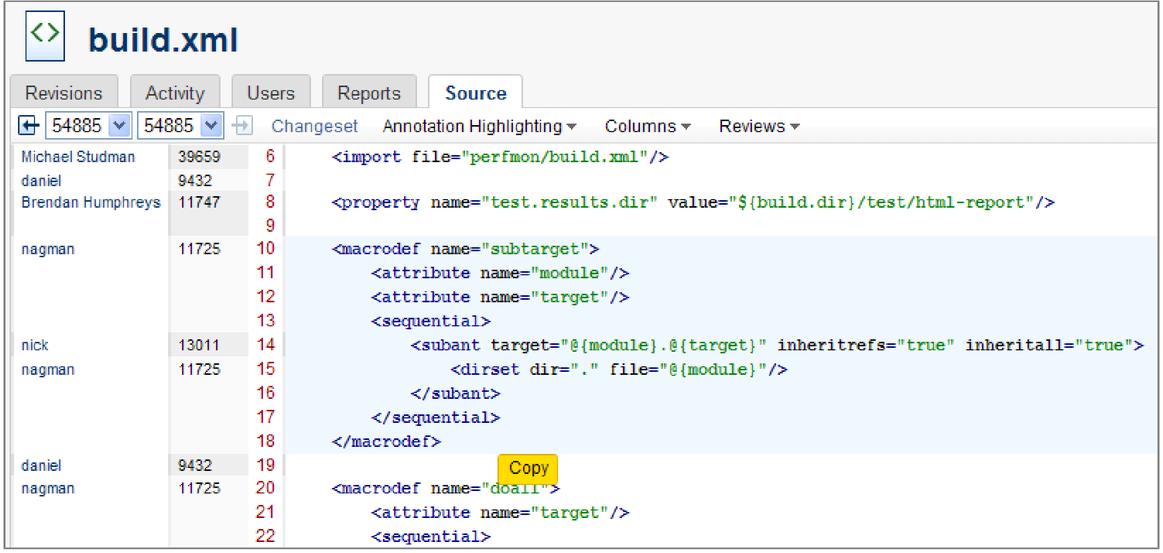
1. Click and drag in the browser window to select the desired lines.
2. A small temporary button labelled '**copy**' appears.
3. Click the '**copy**' button.
4. The selected code is added to the system clipboard. You can now use your operating system's regular paste function to make use of the code elsewhere.

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



```
Michael Studman     39659   <import file="perfmon/build.xml"/>
daniel              9432
Brendan Humphreys   11747   <property name="test.results.dir" value="${build.dir}/test/html-report"/>

nagman              11725   <macrodef name="subtarget">
                                <attribute name="module"/>
                                <attribute name="target"/>
                                <sequential>
nick                13011           <subant target="@{module}.@{target}" inheritrefs="true" inheritall="true">
nagman              11725               <dirset dir="." file="@{module}"/>
                                    </subant>
                                </sequential>
                            </macrodef>
daniel              9432
nagman              11725   <macrodef name="doall">
                                <attribute name="target"/>
                                <sequential>
```

Changeset Discussions

Please see the Crucible documentation for instructions on this feature.

Changing your User Profile

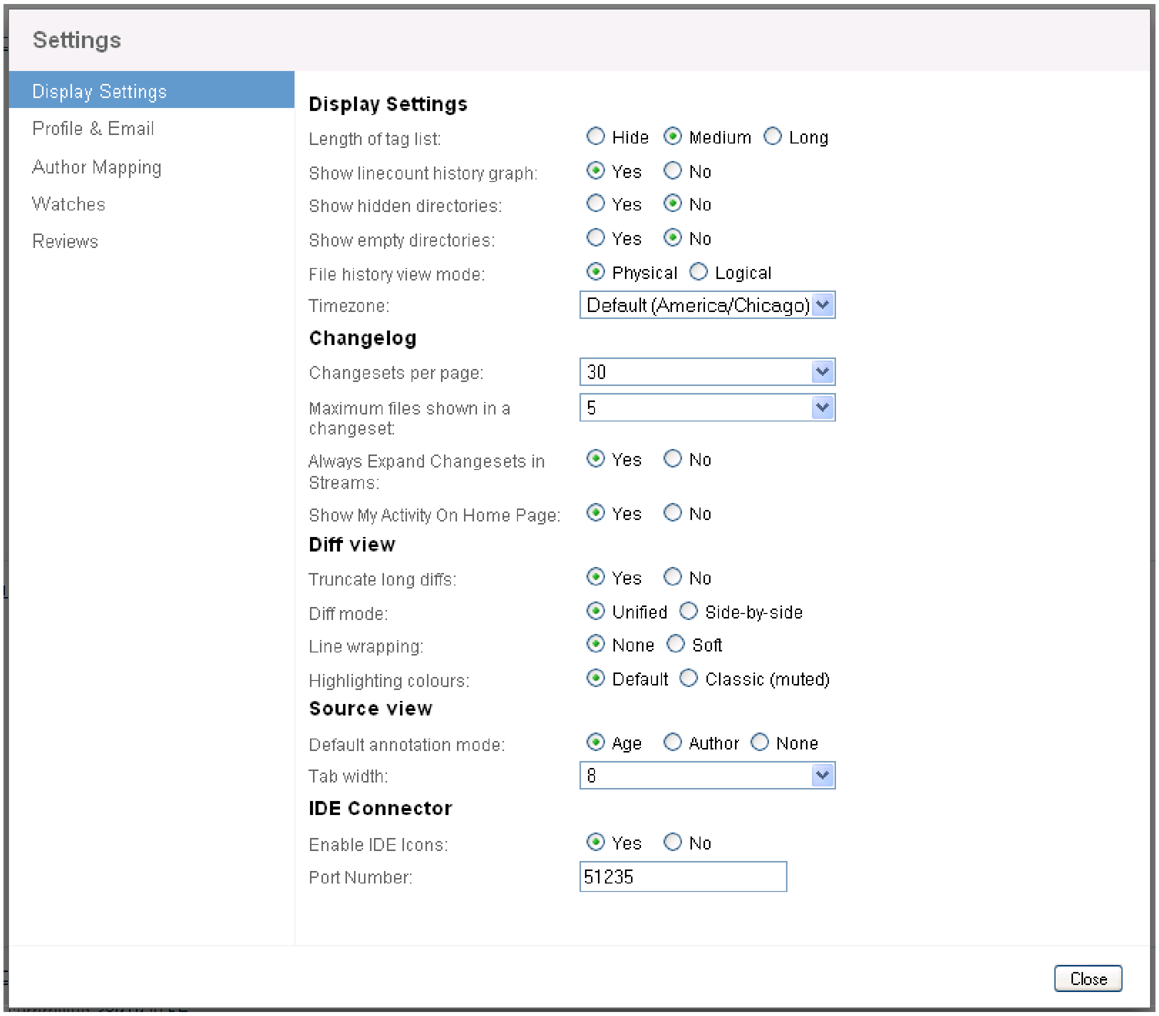
You can change FishEye settings such as password, notifications and display settings.

To change these settings, log into FishEye and click the User Menu (labelled with your username) at the top of the screen, then select '**Settings**'.

Below is a description of each tab and its contents.

- Display Settings Tab
- Profile and Email Tab
- Change Password Tab
- Author Mapping Tab
- Watches Tab
- Reviews tab
- Customising Your User Avatar

Screenshot: Settings



Display Settings

Length of tag list: ○ Hide ● Medium ○ Long

Show linecount history graph: ● Yes ○ No

Show hidden directories: ○ Yes ● No

Show empty directories: ○ Yes ● No

File history view mode: ● Physical ○ Logical

Timezone: Default (America/Chicago)

Changelog

Changesets per page: 30

Maximum files shown in a changeset: 5

Always Expand Changesets in Streams: ● Yes ○ No

Show My Activity On Home Page: ● Yes ○ No

Diff view

Truncate long diffs: ● Yes ○ No

Diff mode: ● Unified ○ Side-by-side

Line wrapping: ● None ○ Soft

Highlighting colours: ● Default ○ Classic (muted)

Source view

Default annotation mode: ● Age ○ Author ○ None

Tab width: 8

IDE Connector

Enable IDE Icons: ● Yes ○ No

Port Number: 51235

Display Settings Tab

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

General

Length of tag list	Default is 'Medium'. The option to show the list of tags for a file. This can be changed to show none ('Hide') or all ('Long').
Show Linecount History Graph	Default is 'Yes'. This is the graph that appears on the left hand side of the Browse and Changelog screen.
Show hidden directories	Default is 'No'. Do not show the hidden directories within any folder lists.
Show empty directories	Default is 'Yes'. The option to see any empty directories within any folder lists.
File History View Mode	Default is 'Logical'. In Subversion repositories, FishEye is able to show all operations on a single logical file spread across a number of physical paths - i.e. operations in different branches. When this is set to 'Logical', FishEye will show all the operations across all branches. In 'Physical' mode, only the operations related to the physical path whose history is being viewed are shown.
Timezone	Default is the timezone of the FishEye server.

Changelog

Changesets per page	The default is 30 per page.
Maximum files shown in a changeset	Default is 5.

Diff View

Truncate long diffs	Default is 'Yes'. Only show part of the diff, if the diff contains many lines of code.
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.
Highlighting Colours	The default colour scheme uses bright colours for highlighting diffs in the code. If you prefer more muted colours, select '**Classic (muted)**'.

Source View

Default annotation mode	Default is 'Age'. It can be changed to 'Author' or 'None'.
Tab width	Default is 8. Can be changed to a number between 1 and 10.

Profile and Email Tab

The settings in this tab allow you to change your email address and your display name.

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	Default is 'Immediately'. Can be changed to 'Daily'.

Change Password Tab

Option to be able to change your password if required.

> ⚠ The passwords are case sensitive.

Author Mapping Tab

This functionality is used by Crucible. Refer to the Crucible documentation.

Watches Tab

> ✓ **Add a 'watch' on the Browse, File History or Changelog page**
>
> By adding a 'watch', you can ask to receive emails about changes made to the repository. To add a watch, click on the ✉ icon at the top right of any Browse, File History or Changelog page.

The '**Watches**' tab in your Profile allows you to change the frequency at which the 'watch' emails are sent.

- 'Immediately' - the email is sent every time a change is made.
- 'Daily' - you will receive a daily email detailing these changes.

The default is 'Immediately'.

The option to add a watch may only be available if the administrator has enabled watches for the repository.

Reviews tab

This functionality is used by Crucible. Refer to the Crucible documentation.

Customising Your User Avatar

If your administrator has enabled an external avatar server (e.g. Gravatar), you can upload an avatar image of your choice. Note that your login name to the external server must be the email address that is specified in your User Profile.

Re-setting your Password

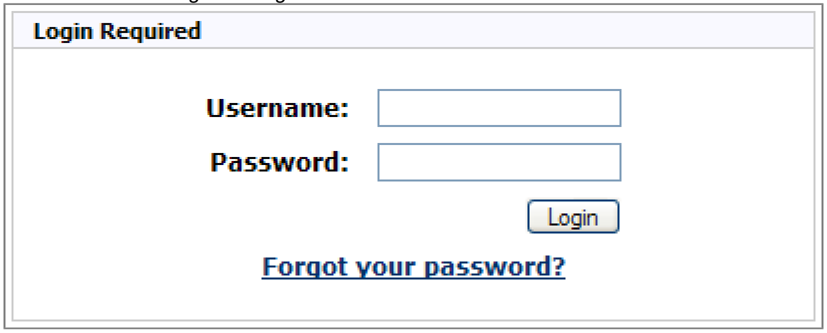
If you need to reset your password, FishEye has an integrated mechanism to generate a new password and send it to the email address in your profile.

To reset your password:

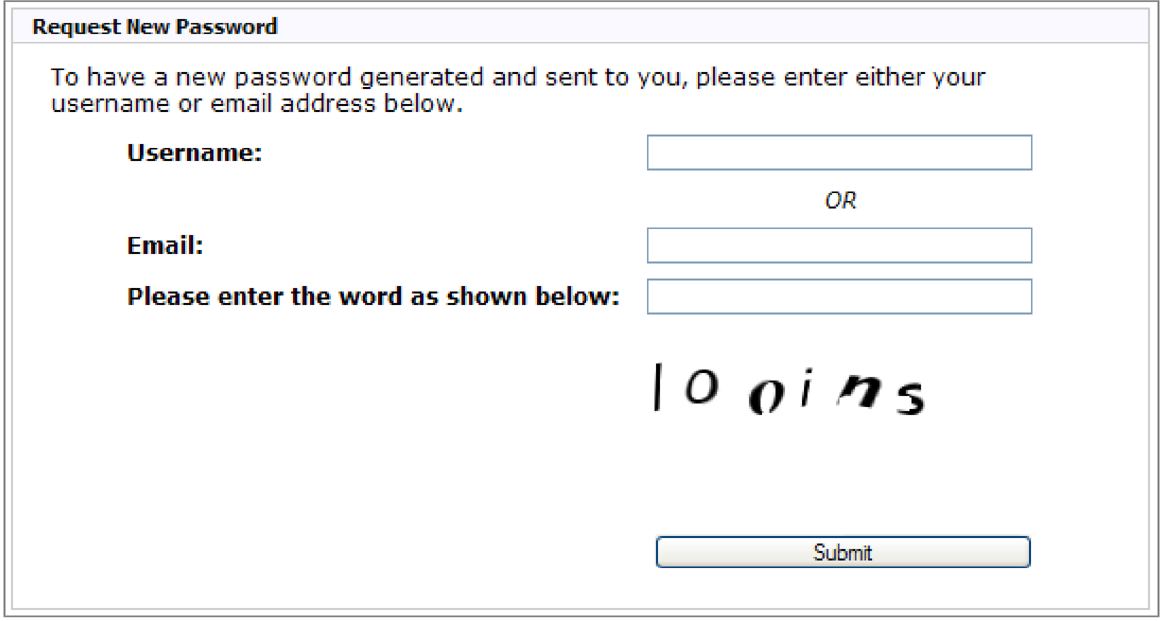
1. On the log in screen, click the '**Forgot your password?**' link. The '**Request New Password**' screen opens.
2. Fill out your username or email address and the Captcha step. That is, click in the form field labelled '**Please enter the word as shown below**' and type the graphical letters shown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



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

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.

FishEye Administrator's Guide

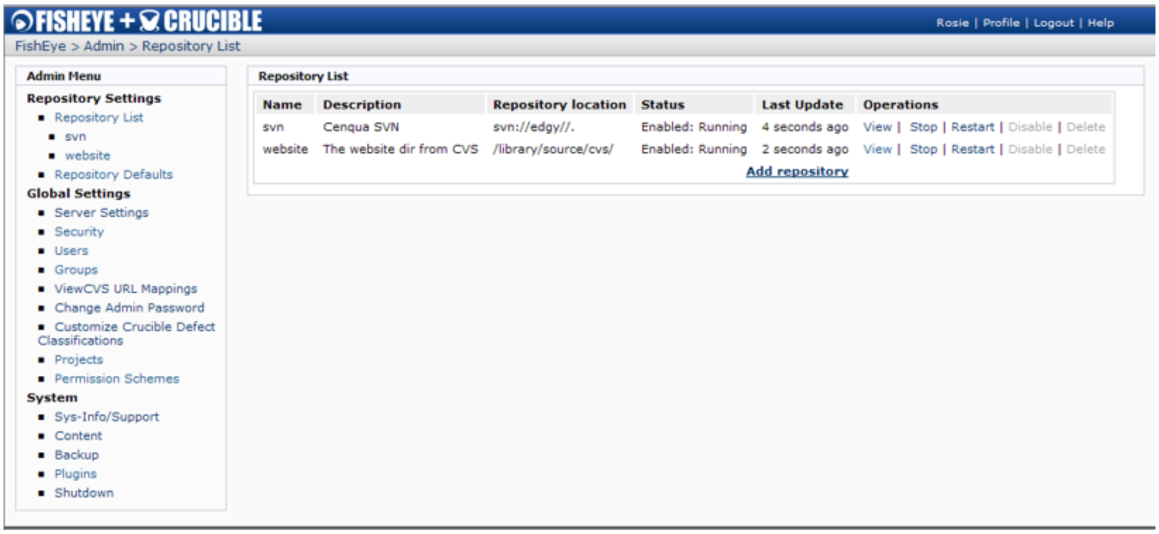
Once you have installed and configured FishEye, you can access the **Administration** pages at http://HOSTNAME:8060/admin/.

The FishEye '**Admin Menu**' allows you to administer your FishEye instance and manage your repositories, as shown in the screenshot below.

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.

Screenshot: FishEye Admin Menu and Repository List



Name	Description	Repository location	Status	Last Update	Operations
svn	Cenqua SVN	svn://edgy//.	Enabled: Running	4 seconds ago	View | Stop | Restart | Disable | Delete
website	The website dir from CVS	/library/source/cvs/	Enabled: Running	2 seconds ago	View | Stop | Restart | Disable | Delete

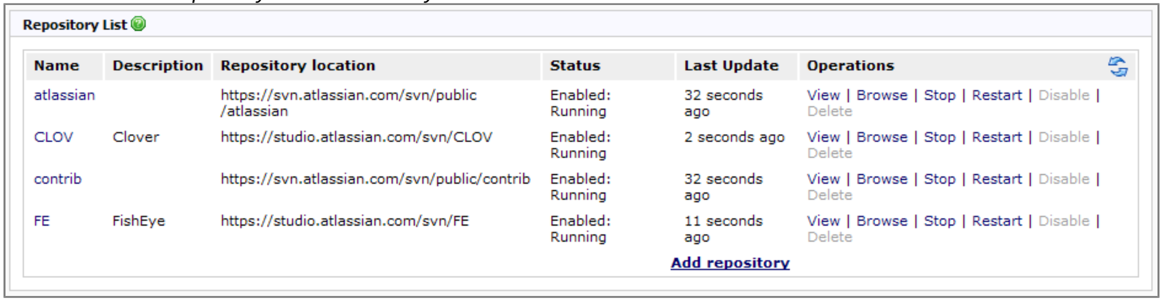
Information in this *Administrator's Guide*:

- Managing your Repositories
- Configuring ViewVC Compatibility
- Configuring SMTP
- Setting up a Repository Client
- Setting up your Web Server
- Contacting Support
- Migrating to an External Database
- Software Update Notifications
- Running Scheduled Events
- JIRA Integration in FishEye
- Customising Email Notifications
- Setting up your Users and Security
- Trusted Applications
- Advanced Administration Options
- Managing Plugins
- Customising the Welcome Message
- Backing Up and Restoring FishEye Data
- Enabling Access Logging in FishEye

Managing your Repositories

You can see a summary of your current repositories at the top of the first screen of the FishEye '**Administration**' options.

Screenshot: The Repository List in the FishEye Administration Screen



Repository List

Name	Description	Repository location	Status	Last Update	Operations
atlassian		https://svn.atlassian.com/svn/public/atlassian	Enabled: Running	32 seconds ago	View | Browse | Stop | Restart | Disable | Delete
CLOV	Clover	https://studio.atlassian.com/svn/CLOV	Enabled: Running	2 seconds ago	View | Browse | Stop | Restart | Disable | Delete
contrib		https://svn.atlassian.com/svn/public/contrib	Enabled: Running	32 seconds ago	View | Browse | Stop | Restart | Disable | Delete
FE	FishEye	https://studio.atlassian.com/svn/FE	Enabled: Running	11 seconds ago	View | Browse | Stop | Restart | Disable | Delete

See the links below for documentation on the FishEye repository options.

FishEye Repository Options

- Adding a Repository
  - ClearCase
  - CVS
  - Git
  - Mercurial
  - Perforce
  - Subversion
    - SVN fisheye.access
    - SVN Tag and Branch Structure
      - How Tags Work in Subversion
- Repository Options
  - Configuring Repository Details
  - Store Diff Info
  - Operations
  - Symbolic
  - Indexer
  - Updater
  - Linkers
  - Permissions
  - Watches
  - Allow (Process)
  - Hidden Directories
  - Tarball Settings
  - Properties
  - Commit Message Syntax

Adding a Repository

Adding a repository to FishEye is a simple matter. Once a repository has been added, further configuration options are available, depending upon the repository type.

Note that FishEye needs to build an index and cache of your repository. This begins when you first **enable** a repository, and may take some time to complete.

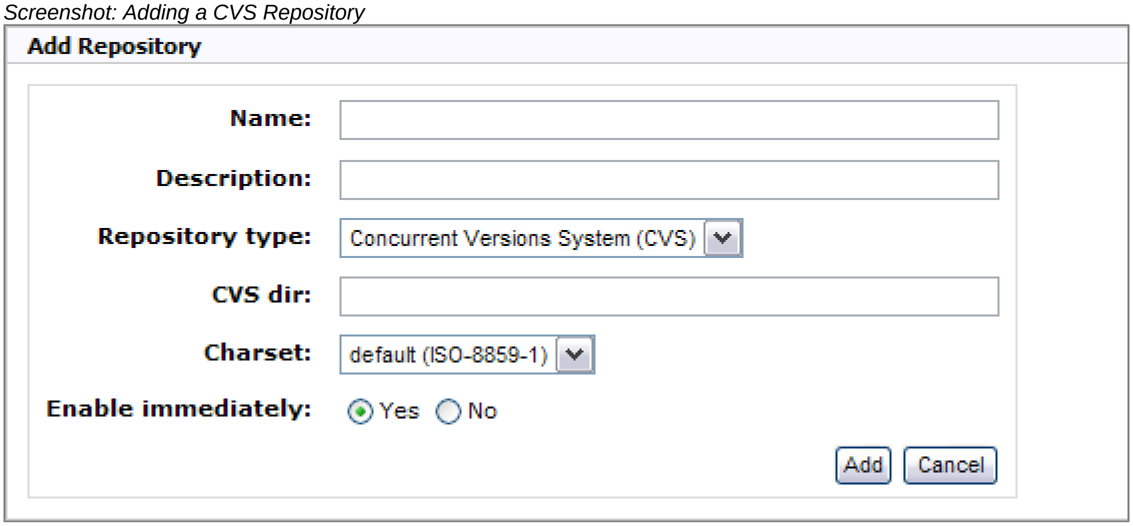
To add a repository,

1. From the '**Admin Menu**', click the '**New**' link next to '**Repository List**', OR, click through to the '**Repository List**' page and then click '**Add Repository**'.
2. Select a '**Repository type**' from the dropdown list.
3. Specific fields will appear on the '**Add Repository**' screen, depending on the chosen repository type. Enter the repository details as prompted. You will find more information in the specific sections listed below.

FishEye currently supports the following repository types:

- ClearCase
- CVS
- Git
- Mercurial
- Perforce
- Subversion

Screenshot: Adding a CVS Repository



Add Repository

Name:	
Description:	
Repository type:	Concurrent Versions System (CVS)
CVS dir:	
Charset:	default (ISO-8859-1)
Enable immediately:	● Yes ○ No

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

On this page:

Requirements

The instructions on this page require the following applications:

- IBM ClearCase 2003.06.10 or later

Setting up a ClearCase Repository

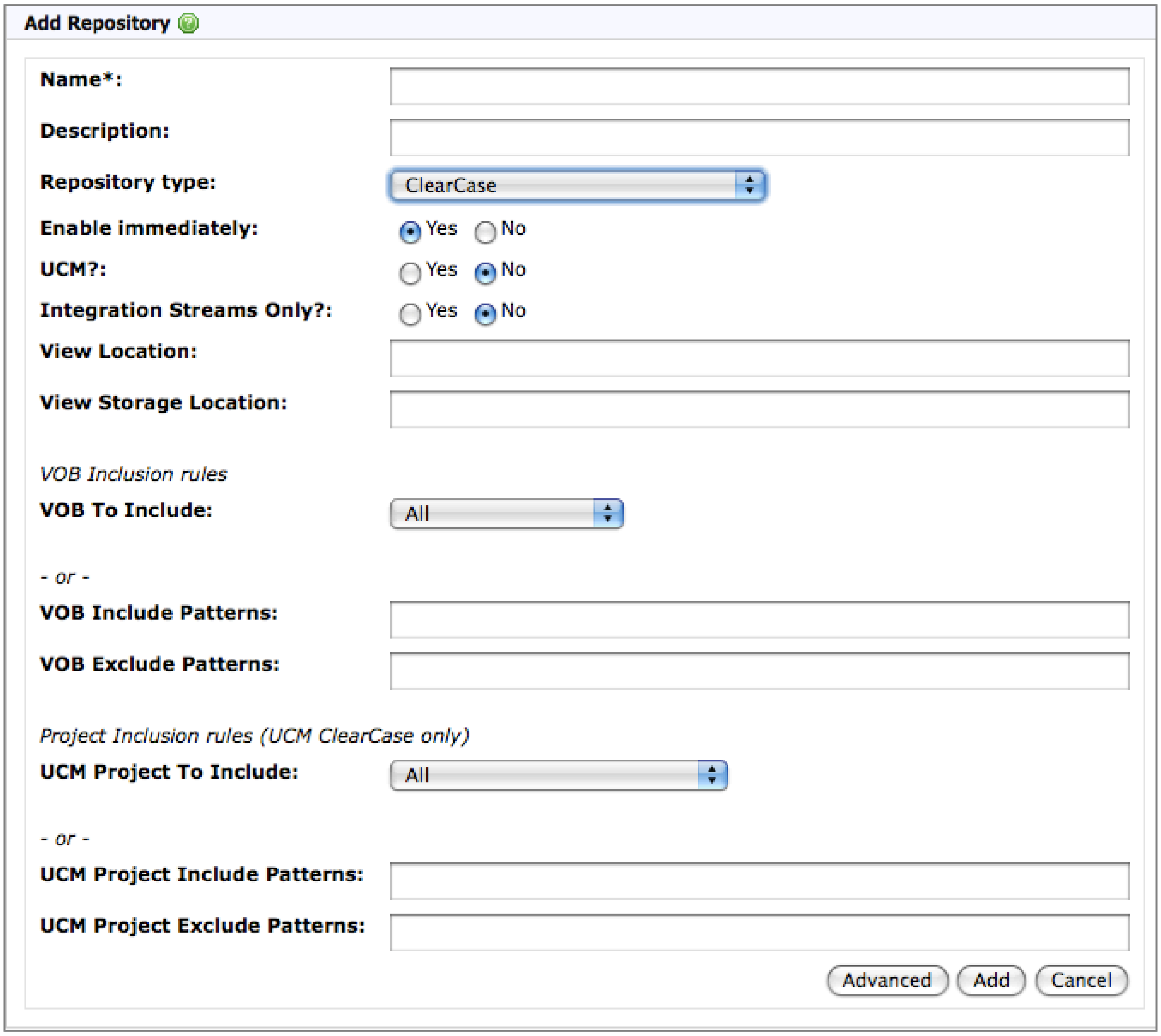
When adding or managing a ClearCase repository, carry out the following steps:
1. Open FishEye's '**Add Repository**' dialog, by choosing '**Administration**' > '**Repository List**' > '**New**'.
2. Set your repository details, as described below.

ClearCase Repository Details

Field	Description	Allowed values
Name	A name chosen by you to be displayed in the list of FishEye repositories.	Free text
Description	A description for this repository, if required.	Free text
Repository Type	Defines the repository type; these instructions apply to ClearCase, so select ClearCase.	ClearCase

Enable Immediately	Defines whether the repository will be accessible in FishEye right away.	Yes / No
UCM	Indicates whether the underlying ClearCase repository uses UCM or Base ClearCase.	Yes / No
Integration Streams Only	Specifies whether FishEye should index content that has been delivered to development and streams or only integration streams. It is recommended that users choose 'Yes' for this option.	Yes / No
Auto Create View	If set, FishEye will create views for each VOB being indexed. If not set, FishEye will use an existing view (which has been specified in the View Location field)	True/False
View Location	If Auto Create View is set, the location of a directory accessible to the FishEye instance where views can be created. If Auto Create View is not set, the location of an existing ClearCase view	A system path
View Storage Location	The location where view storage files are stored.	A system path
VOB to Include	A drop down list displaying all the non-UCM VOBs found in the ClearCase installation. If users only require that FishEye index a single VOB, they should select the VOB to index from this drop down list	Auto populated
VOB Includes	Specifies the pattern that should be used to determine whether a VOB should be included in the indexing logic. Multiple inclusion patterns can be separated with a comma	Free text
VOB Excludes	Specifies the pattern that should be used to determine whether a VOB should be excluded from the indexing logic. Multiple exclusion patterns can be separated with a comma	Free text
UCM 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	Auto populated
UCM Project Include Patterns	Specifies the pattern that should be used to determine whether a UCM Project should be included in the indexing logic. Multiple inclusion patterns can be separated with a comma	Free text
UCM Project Exclude Patterns	Specifies the pattern that should be used to determine whether a UCM Project should be excluded from the indexing logic. Multiple exclusion patterns can be separated with a comma	Free text

Screenshot: Adding a ClearCase Repository



Add Repository

Name*:

Description:

Repository type: ClearCase

Enable immediately: ● Yes ○ No

UCM?: ○ Yes ● No

Integration Streams Only?: ○ Yes ● No

View Location:

View Storage Location:

VOB Inclusion rules

VOB To Include: All

- or -

VOB Include Patterns:

VOB Exclude Patterns:

Project Inclusion rules (UCM ClearCase only)

UCM Project To Include: All

- or -

UCM Project Include Patterns:

UCM Project Exclude Patterns:

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.

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.

UCM ClearCase

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 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.

ℹ PVOB stands for *Project Versioned Object Base*.

Base ClearCase

The logic for the Base ClearCase support is similar to the UCM ClearCase support,

- All non-UCM VOBs that are available are identified.
- Find the check-ins for each VOB and index the version inforamtion.
- 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.

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

Known Issues

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

- 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

- 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:

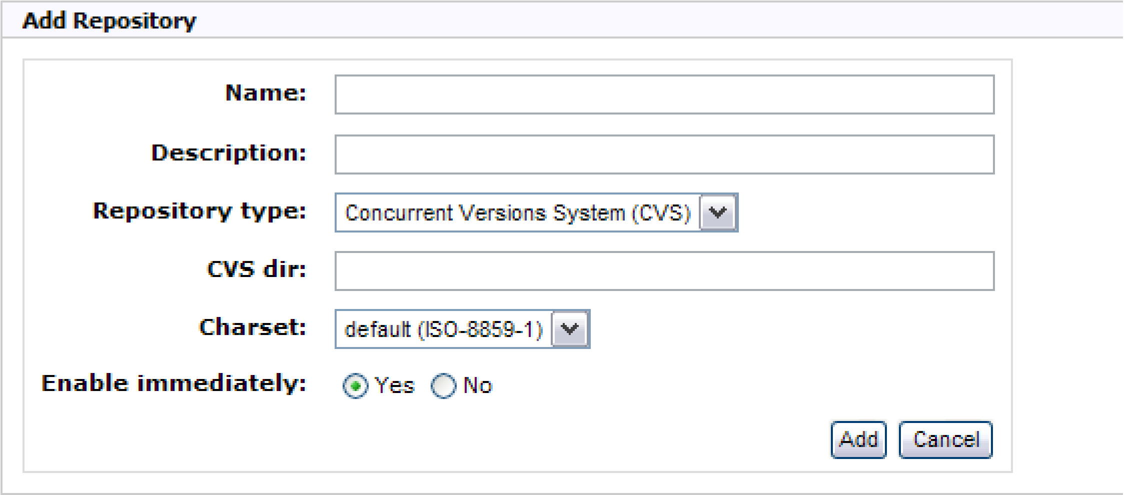
- Updater.
- Indexer.

> ⚠ 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



Add Repository

Name:

Description:

Repository type: Concurrent Versions System (CVS)

CVS dir:

Charset: default (ISO-8859-1)

Enable immediately: ● Yes ○ No

Git

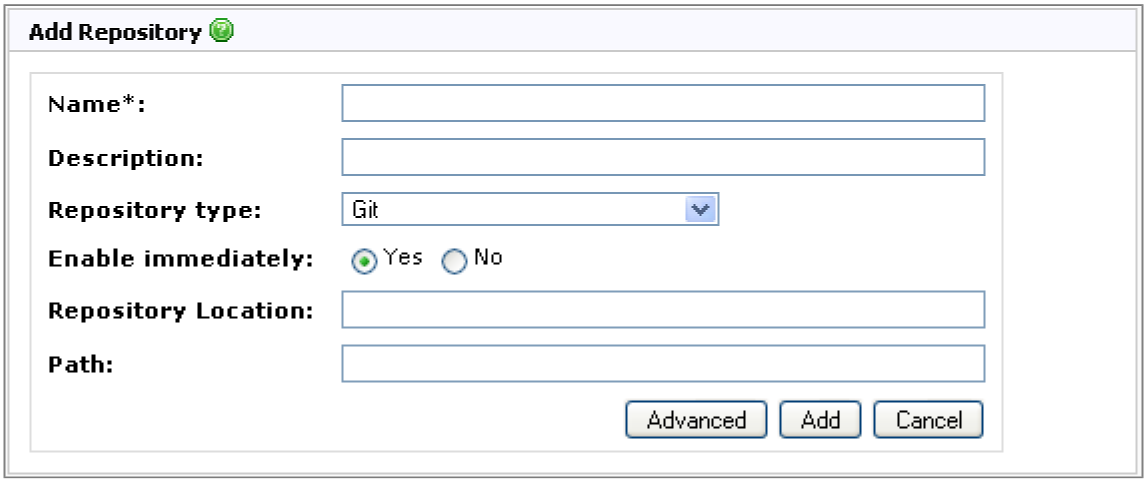
When adding or managing a Git repository, you can do the following:

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

Git Repository Details

Name	A name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters.
Description	A short description of this repository.
Location	The URL describing the git repository location. FishEye will clone this repository for indexing purposes.
Path	The path within the Git repository that you wish to have FishEye index.
Block Size	Controls how many commits FishEye will process in one batch. ℹ This setting appears under '**Advanced**'.

Screenshot: Adding a Git Repository



Add Repository

Name*:

Description:

Repository type: Git

Enable immediately: ● Yes ○ No

Repository Location:

Path:

Related Links

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

Mercurial

> ⚠ The information on this page relates to FishEye's alpha support for Mercurial. Do not use in production.

Introduction and Disclaimer

Atlassian stresses that this is not final support for Mercurial which means the following:

1. Subsequent updates to the technology are likely to require re-indexing.
2. 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 Mercurial support in FishEye.

See the Mercurial Known Issues page for details on the features that are not complete.

Configuration

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

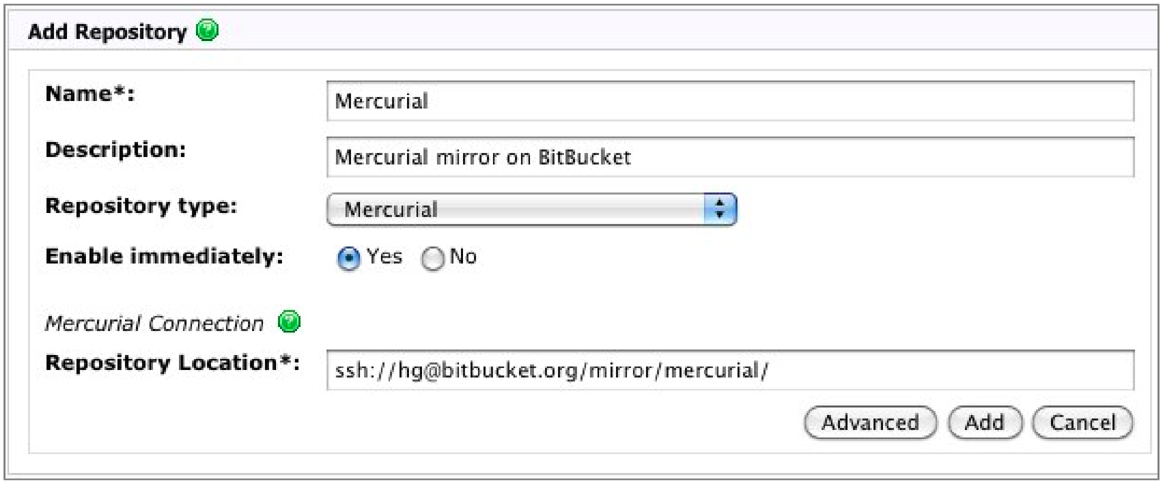
When adding or managing a Mercurial repository, you can do the following:

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

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. ℹ This URL must be compatible with the hg clone command.
Block Size	Controls how many commits FishEye will process in one batch. ℹ This setting appears under '**Advanced**'.

Screenshot: Adding a Mercurial Repository



Add Repository

Name*: Mercurial

Description: Mercurial mirror on BitBucket

Repository type: Mercurial

Enable immediately: ● Yes ○ No

Mercurial Connection

Repository Location*: ssh://hg@bitbucket.org/mirror/mercurial/

Submitting Feedback

We're very interested in your feedback. The best place for submitting feedback is the FishEye forums.

Related Links

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

Perforce

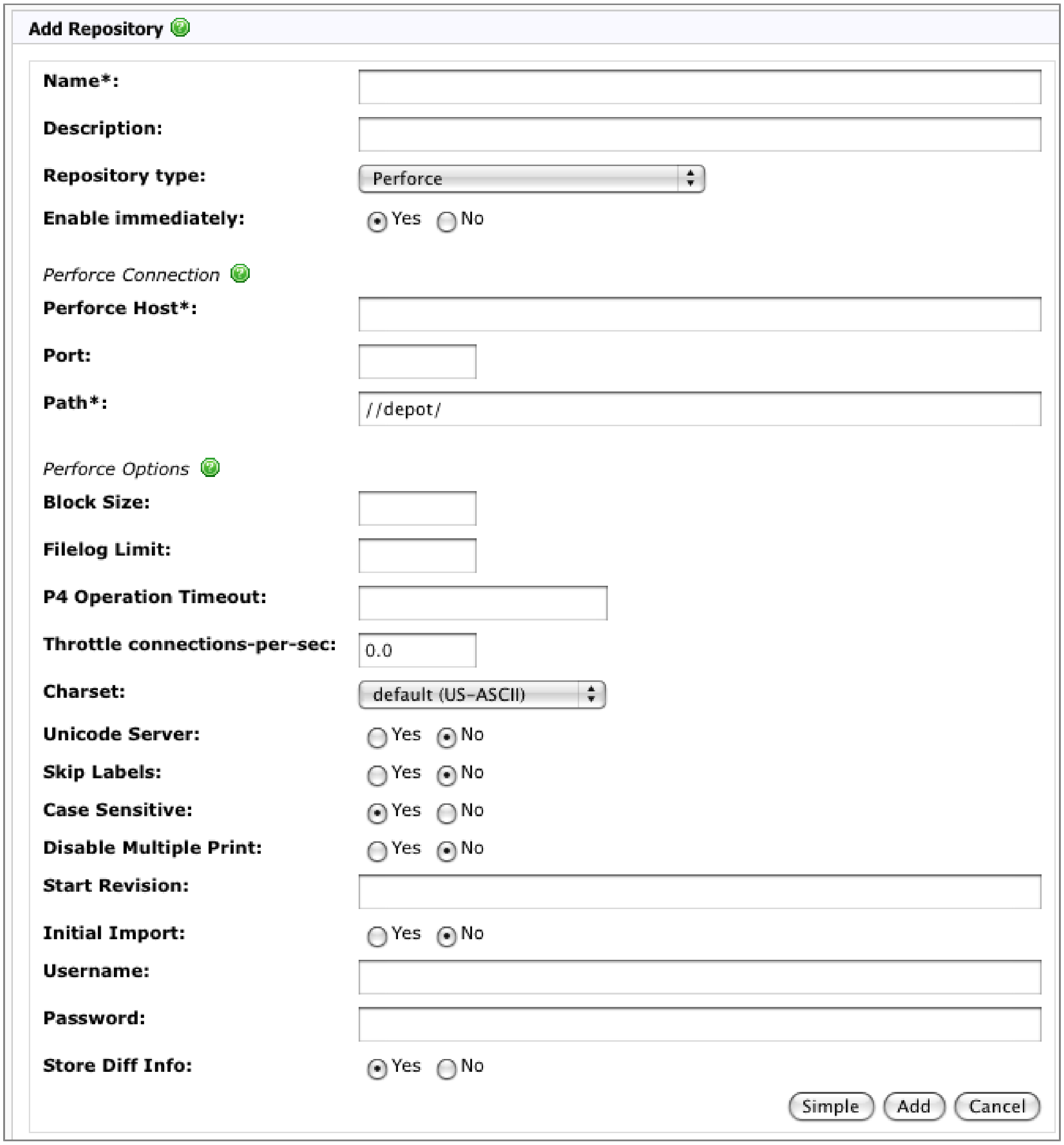
When adding or managing a Perforce repository, you can:

- 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.
Throttle connections-per-sec	If set, this allows FishEye to throttle how many connections it makes per second to the Perforce server. The default is blank (do not throttle). You may enter fractional values such as 2.5.
Charset	The character set used to interpret and display text files.
Unicode Server	This field indicates whether the Perforce Server is running in internationalised mode.
Skip Labels	When true, FishEye will not scan Perforce Labels for FishEye tag information.
Case Sensitive	This field indicates whether the Perforce Server metadata is case sensitive. You should set this to 'false' for servers running on Windows platforms.
Disable Multiple Print	When FishEye needs file content from Perforce, it uses a p4 print operation. Normally FishEye will request multiple files in one operation but this can cause problems in some Perforce instances. Set this value to true to disable printing multiple files in one operation.
Start Revision	If you wish to set this, please enter a **changelist** number. If set, the revision number from which FishEye will start indexing the repository. The default is to start scanning from the first revision in the repository.
Initial Import	When a Start Revision is set, this setting controls how FishEye establishes the initial state of the repository. When true, FishEye will import the repository content as it existed one revision prior to the start revision. FishEye will create a single synthetic revision to hold the initial state. The comment associated with this revision will be 'Created by FishEye for initial repository import'. False means that FishEye will only process the revisions from the start revision onwards. The repository state prior to this revision is ignored.
Username/Password	The credentials to use if your repository requires authentication.
Store Diff Info	Enable this option if you are using the Subversion or Perforce SCM systems and making use of per-author line counts. Otherwise, enabling this option is not necessary. Read more information
Enable immediately	Controls whether FishEye will immediately enable this repository, which starts the initial scan. If you wish to do some further configuration before the scan starts, then select 'No'. You can enable a repository later from the Repository List.

Screenshot: Adding a Perforce Repository



Add Repository

Name*:	
Description:	
Repository type:	Perforce
Enable immediately:	● Yes ○ No
Perforce Connection	
Perforce Host*:	
Port:	
Path*:	//depot/
Perforce Options	
Block Size:	
Filelog Limit:	
P4 Operation Timeout:	
Throttle connections-per-sec:	0.0
Charset:	default (US-ASCII)
Unicode Server:	○ Yes ● No
Skip Labels:	○ Yes ● No
Case Sensitive:	● Yes ○ No
Disable Multiple Print:	○ Yes ● No
Start Revision:	
Initial Import:	○ Yes ● No
Username:	
Password:	
Store Diff Info:	● Yes ○ No

Subversion

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

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

There are also the following SVN-specific actions:

- Updater.
- Indexer.
- Store Diff Info.

It is particularly important that you set up the correct branch and tag structure for your Subversion repositories. If FishEye does not know which files are tags and branches, it will treat all files as trunk files. This can significantly increase the effective size of your repository. **This will increase initial scan time and impact runtime performance.** Please refer to the instructions on [tag and branch configuration].
In the majority of cases, indexing a small repository shouldn't take hours, and certainly not days. However, if you have a giant repository, have a slow remote host, you're using HTTP or HTTPS protocols, or if there is a problem with the symbolic setup of your repository, it could potentially take hours or even days. If in doubt, schedule the indexing to run over a weekend or extended maintenance period.

Using the 'file://' protocol to access your Subversion repository can be much faster than the other network protocols. 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". (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. '**Do not import**' means that FishEye will only process the revisions from the start revision onwards. The repository state prior to this revision is ignored. '**Import without tag information**' means that FishEye will import the repository content as it existed one revision prior to the start revision. FishEye will create a single synthetic revision to hold the initial state. The comment associated with this revision will be 'Created by FishEye for initial repository import'. Tags created prior to the start revision are ignored.

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
```

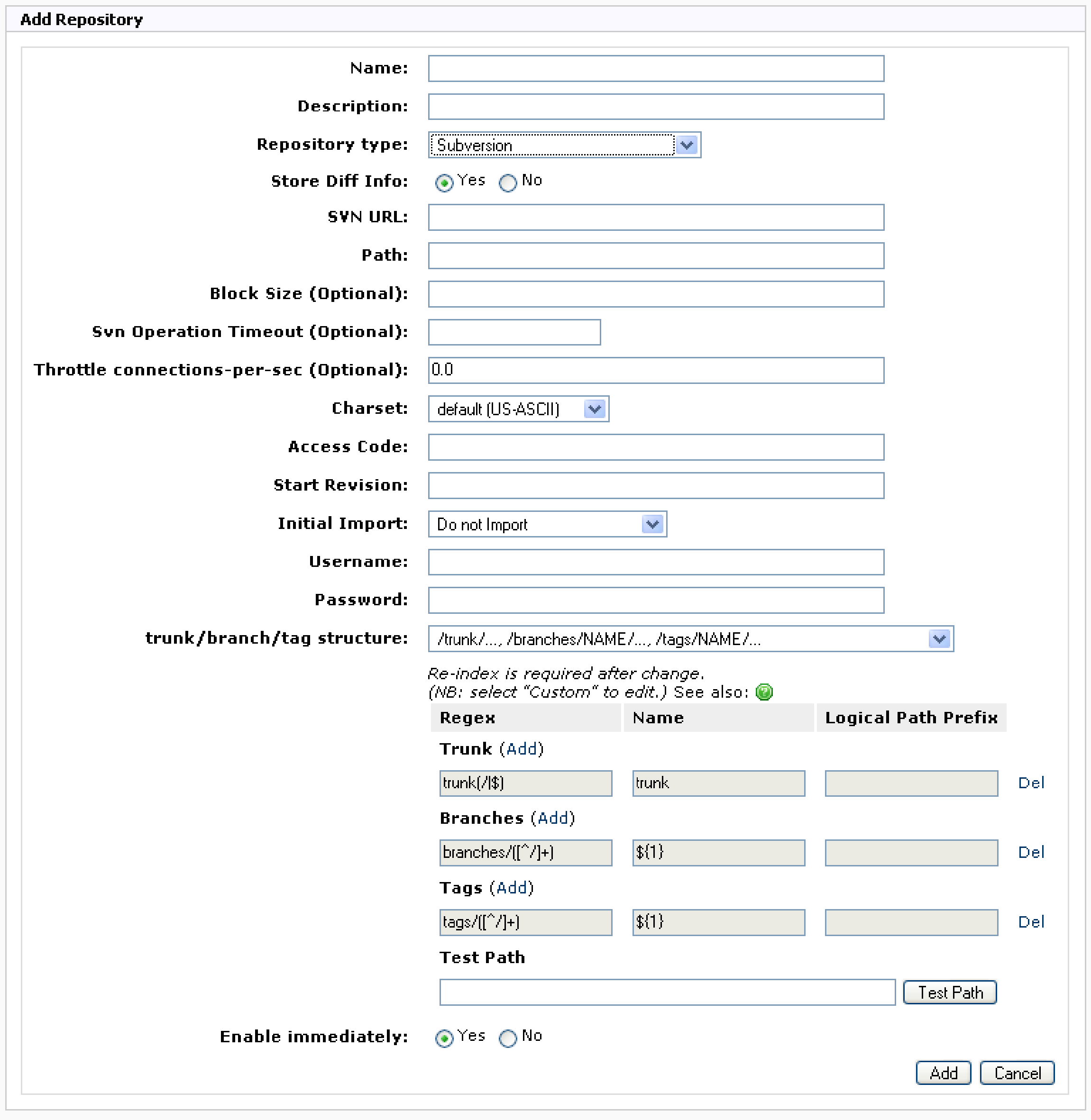
Where SVN_URL is the complete URL of the repository you want to add.

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



Add Repository

Name:	
Description:	
Repository type:	Subversion
Store Diff Info:	● Yes ○ No
SVN URL:	
Path:	
Block Size (Optional):	
Svn Operation Timeout (Optional):	
Throttle connections-per-sec (Optional):	0.0
Charset:	default (US-ASCII)
Access Code:	
Start Revision:	
Initial Import:	Do not Import
Username:	
Password:	
trunk/branch/tag structure:	/trunk/..., /branches/NAME/..., /tags/NAME/...

Re-index is required after change.
(NB: select "Custom" to edit.) See also:

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

Test Path

Enable immediately:	● Yes ○ No

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

<table>
<tr><th>Regular Expression</th><th>Name</th><th>Logical Path Prefix</th></tr>
<tr><td><code>trunk(/|$)</code></td><td><code>trunk</code></td><td>N/A</td></tr>
<tr><td><code>branches/([SVN Tag and Branch Structure^/]+)</code></td><td><code>${1}</code></td><td>N/A</td></tr>
</table>

`(tags|releases)/([SVN Tag and Branch Structure^/]+)`	`${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(/|$)`	`trunk`	`core`
trunk	`plugins/([SVN Tag and Branch Structure^/]+)/trunk(/|$)`	`trunk`	`plugin_${1}`
branches	`core/branches/([SVN Tag and Branch Structure^/]+)`	`core-${1}`	`core`
branches	`plugins/([SVN Tag and Branch Structure^/]+)/branches/([SVN Tag and Branch Structure^/]+)`	`${1}-${2}`	`plugin_${1}`

tags	`core/tags/([SVN Tag and Branch Structure^/]+)`	`core-${1}`	`core`
tags	`plugins/([SVN Tag and Branch Structure^/]+)/tags/([SVN Tag and Branch Structure^/]+)`	`${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/([SVN Tag and Branch Structure^/]+)`	`${1}`	N/A
`branches/([SVN Tag and Branch Structure^/]+)/([SVN Tag and Branch Structure^/]+)`	`${1}-${2}`	N/A
`tags/([SVN Tag and Branch Structure^/]+)/([SVN Tag and Branch Structure^/]+)`	`${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.

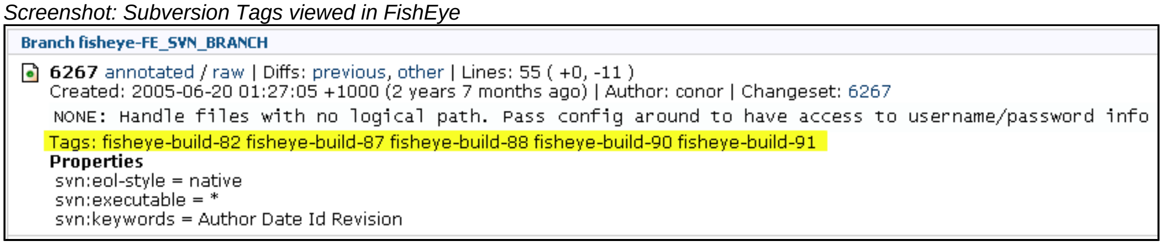
Discovering your Subversion Tag & Branch Structure

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



Branch fisheye-FE_SVN_BRANCH

6267 annotated / raw | Diffs: previous, other | Lines: 55 (+0, -11)
Created: 2005-06-20 01:27:05 +1000 (2 years 7 months ago) | Author: conor | Changeset: 6267

```
NONE: Handle files with no logical path. Pass config around to have access to username/password info
```

Tags: fisheye-build-82 fisheye-build-87 fisheye-build-88 fisheye-build-90 fisheye-build-91

Properties
svn:eol-style = native
svn:executable = *
svn:keywords = Author Date Id Revision

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

FishEye has configuration options for each repository and default settings that will affect all repositories.

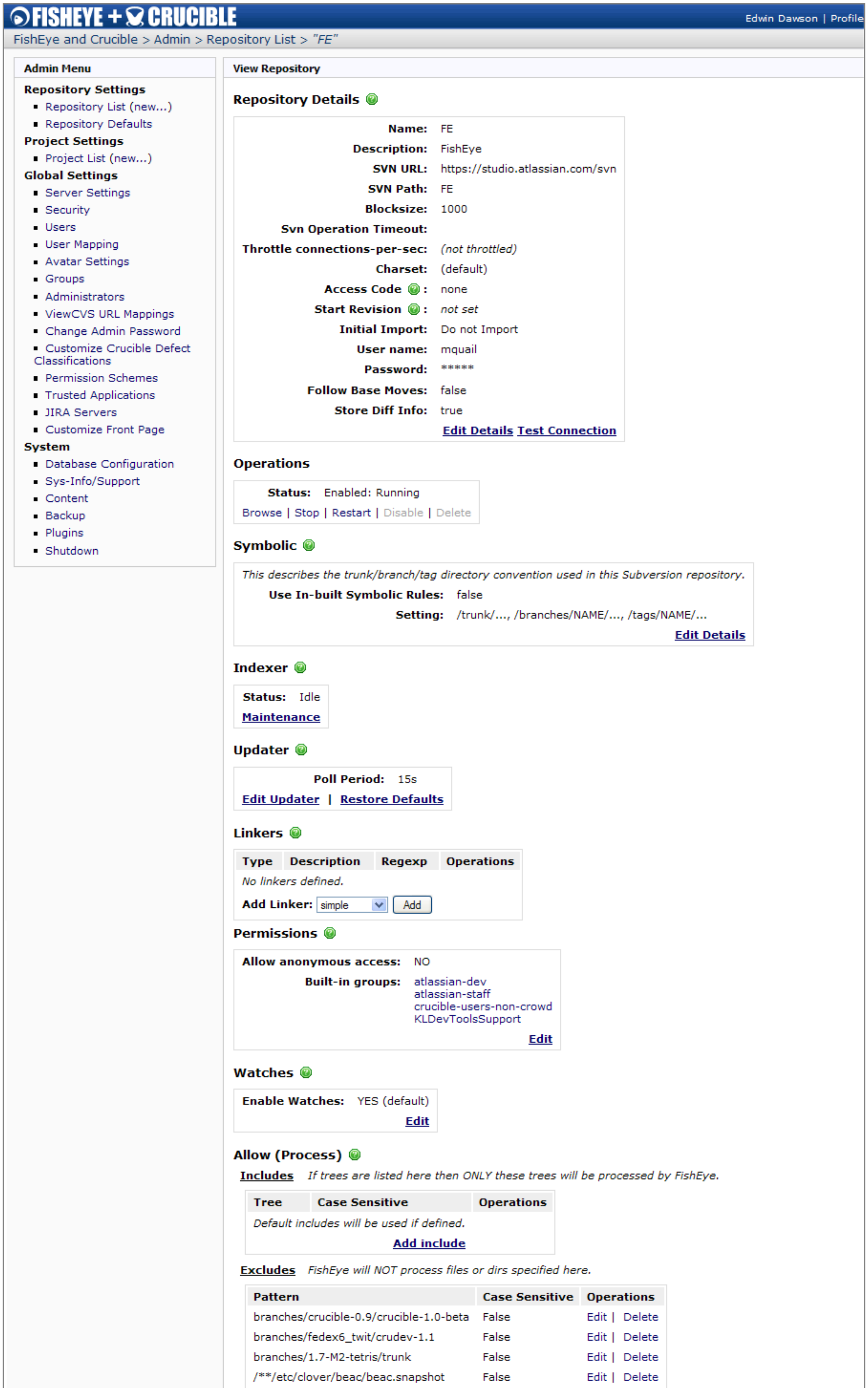
- To access the settings for a specific repository, click the name of the repository in the '**Admin Menu**', or click '**Repository List**' and then click '**View**' next to the repository name.
- To change settings that will affect all repositories, click '**Repository Defaults**' in the '**Admin Menu**'.

> ⚠ 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 '**Repository List**'.

Repository options:

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

Screenshot: Repository Options



Admin Menu

Repository Settings

- Repository List (new...)
- Repository Defaults

Project Settings

- Project List (new...)

Global Settings

- Server Settings
- Security
- Users
- User Mapping
- Avatar Settings
- Groups
- Administrators
- ViewCVS URL Mappings
- Change Admin Password
- Customize Crucible Defect Classifications
- Permission Schemes
- Trusted Applications
- JIRA Servers
- Customize Front Page

System

- Database Configuration
- Sys-Info/Support
- Content
- Backup
- Plugins
- Shutdown

View Repository

Repository Details

Name:	FE
Description:	FishEye
SVN URL:	https://studio.atlassian.com/svn
SVN Path:	FE
Blocksize:	1000
Svn Operation Timeout:	
Throttle connections-per-sec:	*(not throttled)*
Charset:	(default)
Access Code :	none
Start Revision :	*not set*
Initial Import:	Do not Import
User name:	mquail
Password:	*****
Follow Base Moves:	false
Store Diff Info:	true

Edit Details Test Connection

Operations

Status: Enabled: Running

Browse | Stop | Restart | Disable | Delete

Symbolic

This describes the trunk/branch/tag directory convention used in this Subversion repository.

Use In-built Symbolic Rules: false

Setting: /trunk/..., /branches/NAME/..., /tags/NAME/...

Edit Details

Indexer

Status: Idle

Maintenance

Updater

Poll Period: 15s

Edit Updater | Restore Defaults

Linkers

Type	Description	Regexp	Operations
No linkers defined.			

Add Linker: simple [Add]

Permissions

Allow anonymous access: NO

Built-in groups: atlassian-dev
atlassian-staff
crucible-users-non-crowd
KLDevToolsSupport

Edit

Watches

Enable Watches: YES (default)

Edit

Allow (Process)

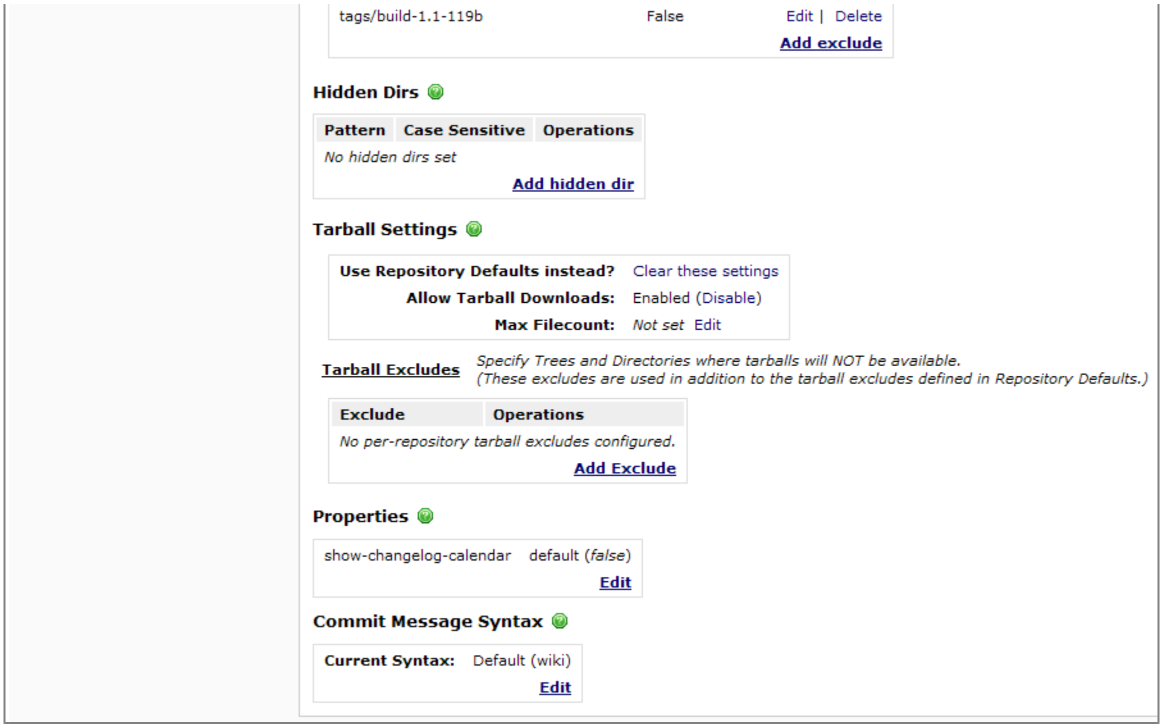
Includes *If trees are listed here then ONLY these trees will be processed by FishEye.*

Tree	Case Sensitive	Operations
Default includes will be used if defined.		

Add include

Excludes *FishEye will NOT process files or dirs specified here.*

Pattern	Case Sensitive	Operations
branches/crucible-0.9/crucible-1.0-beta	False	Edit | Delete
branches/fedex6_twit/crudev-1.1	False	Edit | Delete
branches/1.7-M2-tetris/trunk	False	Edit | Delete
/**/etc/clover/beac/beac.snapshot	False	Edit | Delete



tags/build-1.1-119b False Edit | Delete

Add exclude

Hidden Dirs

Pattern	Case Sensitive	Operations
No hidden dirs set		

Add hidden dir

Tarball Settings

Use Repository Defaults instead?	Clear these settings
Allow Tarball Downloads:	Enabled (Disable)
Max Filecount:	*Not set* Edit

Tarball Excludes *Specify Trees and Directories where tarballs will NOT be available. (These excludes are used in addition to the tarball excludes defined in Repository Defaults.)*

Exclude	Operations
No per-repository tarball excludes configured.	

Add Exclude

Properties

show-changelog-calendar default (*false*)

Edit

Commit Message Syntax

Current Syntax: Default (wiki)

Edit

Configuring Repository Details

When adding or managing a repository, you can:

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

To access the details of a repository,

1. From the '**Admin Menu**', choose one of the following:
   - Click '**Repository List**' then click '**View**' next to the repository name.
   - Or just click the repository name in the '**Admin Menu**'.
2. Then click '**Edit Details**' on the '**View Repository**' page.

Repository details differ depending on the repository type. FishEye currently supports the following repository types:

- ClearCase
- CVS
- Git
- Mercurial
- Perforce
- Subversion

Store Diff Info

This page explains the '**Store Diff Info**' setting in FishEye and how to switch it on and off.

On this page:

About the Store Diff Setting

Store diff info means that FishEye is caching the summary of what lines are added and removed between subsequent versions of the same file in its database. In other words, we are storing the info of a diff, not the diff itself (thus you will still be able to view diffs if this value is set to off).

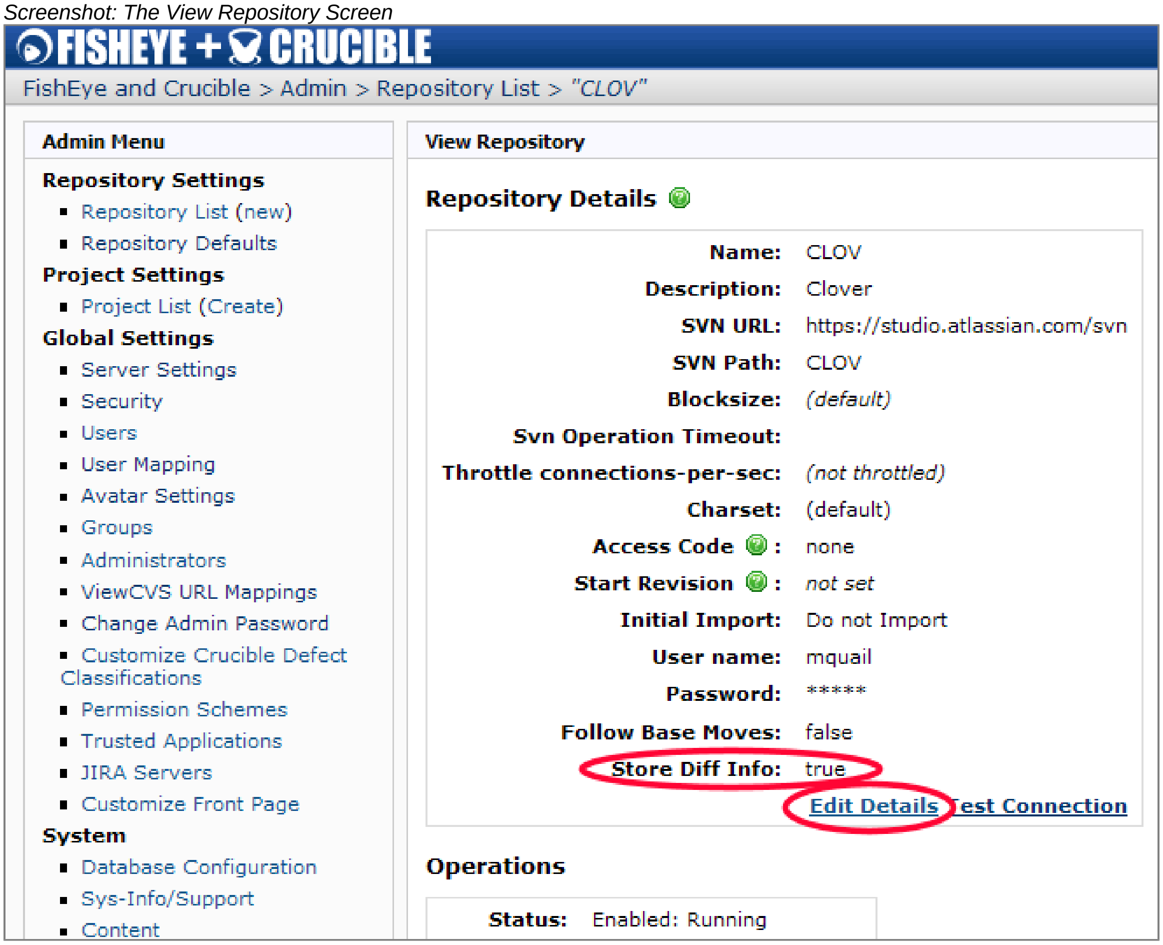
After setting this value to on, a full re-index of your repository is required so that FishEye can collect diff info for all revisions in your repository.

Turning 'Store Diff Info' on and off

To find this setting, open the FishEye **Admininistration Screen**, then click '**Repository List**' under '**Repository Settings**'.

For the desired repository, click '**View**'. The '**View Repository**' screen opens. In the section labelled '**Repository Details**', click '**Edit Details**'. The '**Edit Repository Details**' screen opens.

Screenshot: The View Repository Screen

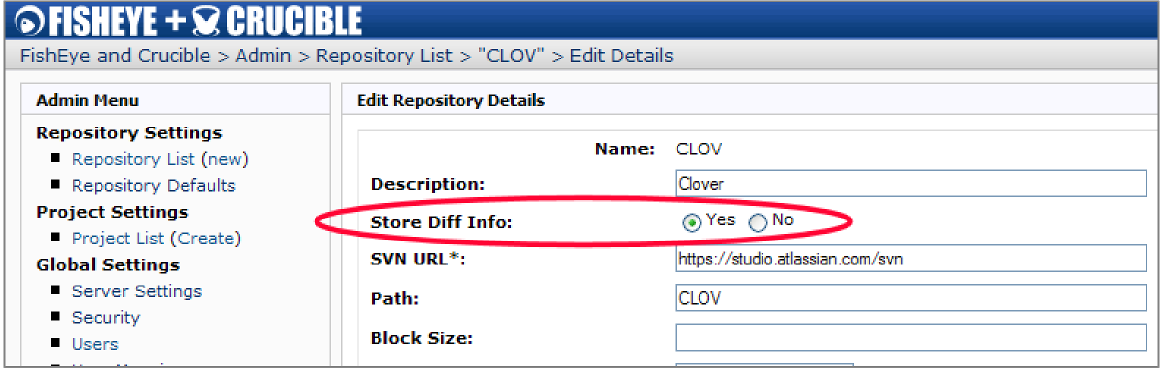


Name: CLOV
Description: Clover
SVN URL: https://studio.atlassian.com/svn
SVN Path: CLOV
Blocksize: *(default)*
Svn Operation Timeout:
Throttle connections-per-sec: *(not throttled)*
Charset: (default)
Access Code: none
Start Revision: *not set*
Initial Import: Do not Import
User name: mquail
Password: *****
Follow Base Moves: false
Store Diff Info: true

Status: Enabled: Running

In the '**Edit Repository Details**' screen, '**Store Diff Info**' is a value that can be toggled on or off. On is the default setting for new repositories.

Screenshot: The Edit Repository Details Screen



Name: CLOV
Description: Clover
Store Diff Info: ● Yes ○ No
SVN URL*: https://studio.atlassian.com/svn
Path: CLOV
Block Size:

Considerations for the Store Diff Info Setting

Required for Per-author Line Graphs

Leaving this option off will disable per-author line graphs.

Required for Displaying Context of Search Results

Leaving this option off will disable the display of context of found terms in search results.

Re-index Required for Per-author Line Graphs

Diff info 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.

Perforce Repository Indexing Performance Impact

Turning '**Store Diff Info**' on for Perforce repositories requires 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.

ℹ Repositories created before FishEye 1.5 will default to '**Store Diff Info**' off.

ℹ CVS and Subversion repository scan times are not affected by this setting.

Operations

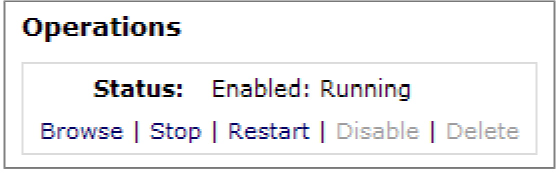
You can carry out various operations on a FishEye repository in the Repository Options screen.

To get to this page:

1. Open FishEye '**Administration**',
2. Find the desired repository on that page and click '**View**' in the repository controls. The Repository Options screen opens.
3. Scroll down to the '**Operations**', section in the repository options. A summary of the current status is shown.
4. Here, you have a number of operation actions you can carry out by clicking:

- **Browse**: This will allow you to browse the repository.
- **Stop**: This will stop the repository scan.
- **Restart**: This will restart the repository scan.
- **Disable**: This will disable the repository.
- **Delete**: This will delete the repository profile from FishEye.

Screenshot: Operations Panel in FishEye Repository Options



Operations

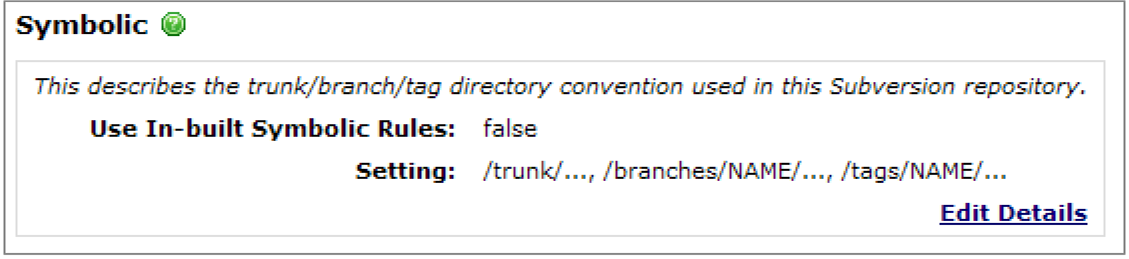
Status: Enabled: Running

Symbolic

In the Repository Options screen, you can view the symbolic settings for the current repository.
You can click '**Edit**' to adjust these settings.

For more information, see SVN Tag and Branch Structure.

Screenshot: Symbolic Settings in FishEye Repository Options



Symbolic

This describes the trunk/branch/tag directory convention used in this Subversion repository.

Use In-built Symbolic Rules: false

Setting: /trunk/..., /branches/NAME/..., /tags/NAME/...

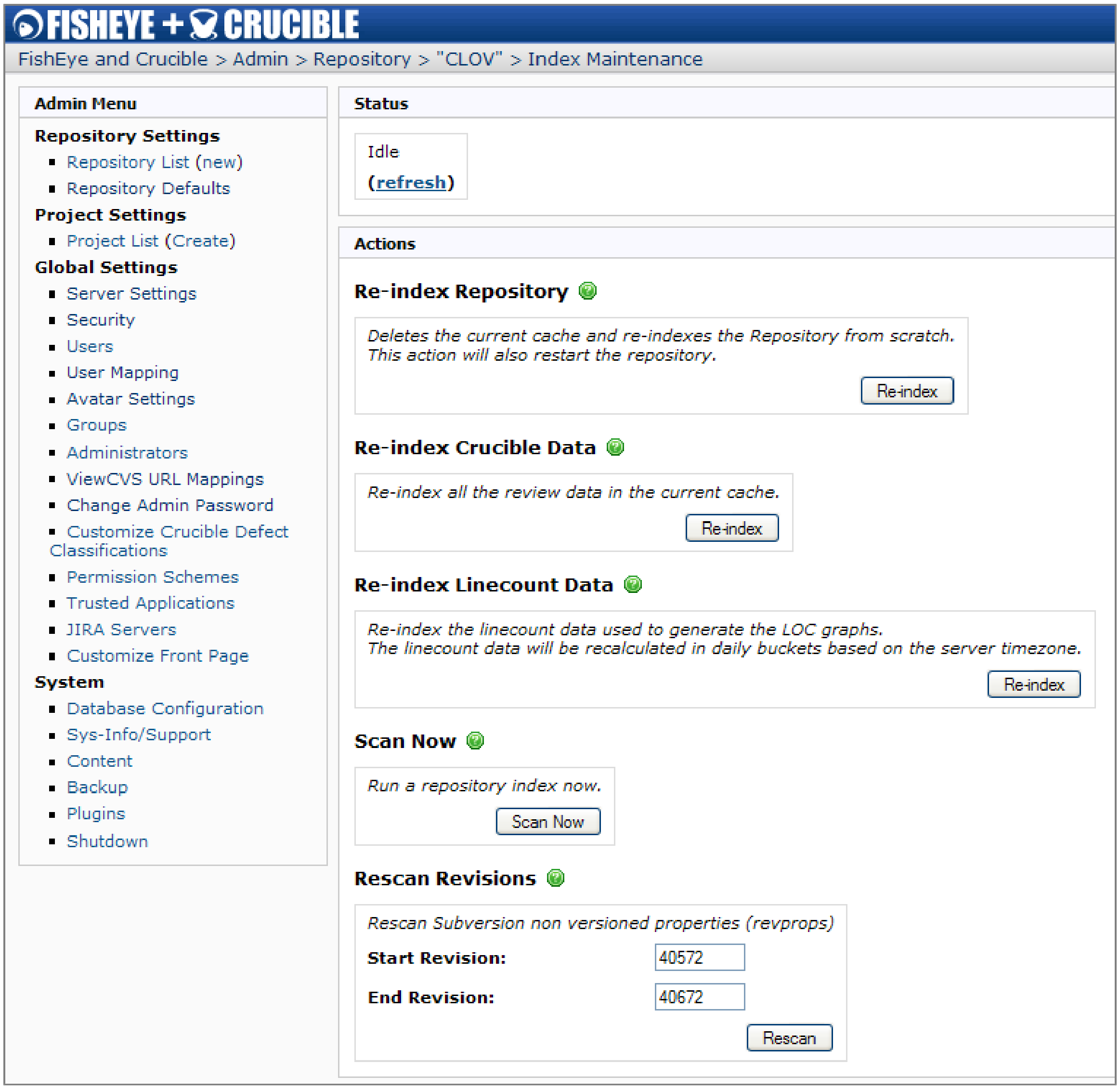
Indexer

To get to this page, go to '**Administration**' > '**Repository List**' > '**View**' (next to your repository name) > '**Indexer**' > '**Maintenance**'

The '**Indexer Maintenance**' screen opens, where an administrator can manually trigger the following actions:

Action	Description
Refresh	Refreshes the status string, in the Status box. This will show the name of any repository task that FishEye is currently carrying out (if any). This may be helpful to monitor the progress of an initial indexing task or to diagnose problems (such as your repository server being out of action).
Re-index Repository	Delete the current cache and re-index the repository from the beginning. This action will also restart the repository.
Re-index Crucible Data	(applies only when using Crucible with FishEye) Re-index all the Crucible review data associated with the current repository.
Re-index Linecount Data	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.
Scan Now	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.
Rescan Revisions	For a **Subversion** repository, you can 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. For a **Perforce**, repository, you can rescan changelists. This will rescan changelist metadata (author, date, commit message) and any job fixes associated with changelists in the the given changelist range.

Screenshot: FishEye Index Maintenance menu



FishEye and Crucible > Admin > Repository > "CLOV" > Index Maintenance

Status

Idle
(refresh)

Actions

Re-index Repository

Deletes the current cache and re-indexes the Repository from scratch. This action will also restart the repository.

Re-index Crucible Data

Re-index all the review data in the current cache.

Re-index Linecount Data

Re-index the linecount data used to generate the LOC graphs.
The linecount data will be recalculated in daily buckets based on the server timezone.

Scan Now

Run a repository index now.

Rescan Revisions

Rescan Subversion non versioned properties (revprops)

Start Revision: 40572

End Revision: 40672

Updater

The '**Updater**' repository option allows an administrator to manually trigger the actions described below, depending upon your repository type.

- Updater (Affects all version-control repositories)
- Updater (CVS)

Updater (Affects all version-control repositories)

Poll Period	How often FishEye will check to see if there have been any new commits into the SVN or Perforce repository. The default is 60 seconds. It is possible to set the period by units. For example: 10second, 1week. Valid units are 'second', 'minute', 'hour', 'day', 'week', 'month', 'year'. The default unit is days if only a number is added. ⚠ You can also set the value 'never', which creates a situation where scanning is purely manually controlled via the command line, or the Scan Now option on the Indexer Maintenance page, accessed by clicking '**Maintenance**' on the '**View**' page for each repository.

Updater (CVS)

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.

The default values should be fine for most repositories. Leave a value blank to use the default value.

Value	Description
History file	The location of the CVS history file. If relative, then it is relative to the CVS directory specified for this repository. Defaults to `./CVSROOT/history`.
Full scan period	How often FishEye will do a full scan of the repository. Defaults to 15 minutes. Specify an interval, such as '15 min', '2 hours', etc. A value of '0' disables the periodic full scan. (You can still use `fisheyectl fullscan` to cause a full scan to occur.)
Strip prefix	Prefix to strip off files found in the history file, to make them relative to this repository's CVS directory. Necessary 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 specified `/usr/local/cvsroot/foo/bar` as the CVS directory of this repository. You will need to give the history file as `../../CVSROOT/history` and set a strip prefix of `foo/bar`.

⚠ Once you have changed the value, you will need to restart fisheye. The period begins from when the initial index completes, i.e. when you restart, your repos will be scanned in order (depending on the number of threads you have configured) and when this scan completes this is the start of the period.

For example, if you set one hour, then your next scan will begin one hour after your initial scan is complete.

Linkers

When browsing commit messages or comments in FishEye, any issue IDs or Bug IDs that appear will be turned into hyperlinks, allowing you to easily click to see those referenced issues or pages. This capability is due to FishEye's ability to detect special substrings in commit messages, and hyperlink those substrings to other systems.

This is particularly useful if you use an issue tracking system, and put the issue identifiers into your commit messages. The '**Linkers**' repository option (Administration > Repository Defaults > Linkers, or Administration > View Repository List > View (next to your REPO) > Linkers) allows you to define the substrings and their related URLs.

Any linkers defined in the repository defaults are added to each individual repository.

Example Linkers

Here are some examples of how to create simple linkers.

JIRA Examples

⚠ If you have already set up JIRA integration, ignore the following instructions.

- 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.

About FishEye Regular Expressions

FishEye uses the Java regular expression language, which is based on Perl 5 regular expressions.

Note: If you want your regex to be case insensitive, put `(?i)` at the start of the regex.

To try out your regular expressions, you can use this online test page.

Example: The SyntaxDef Field

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

- 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.

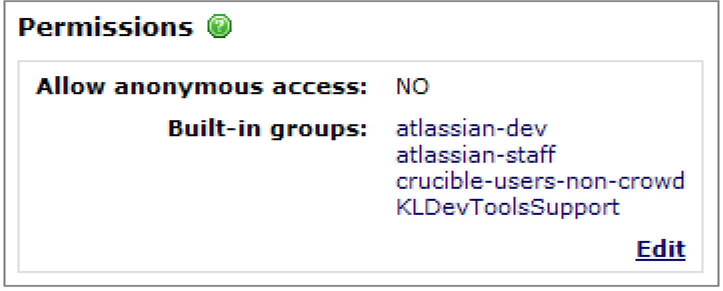
Permissions

For each repository in FishEye, you can adjust per-repository permission settings.

To do this:

1. Open FishEye '**Administration**',
2. Find the desired repository on that page and click '**View**' in the repository controls. The Repository Options screen opens.
3. Scroll down to the '**Permissions**', section in the repository options.
4. Click '**Edit**' to change the permissions. You can enable or disable anonymous access, as well as adding or removing groups from those that are available. For more information, see Associating a Group with a Repository.

Screenshot: Per-Repository Permissions Summary in FishEye

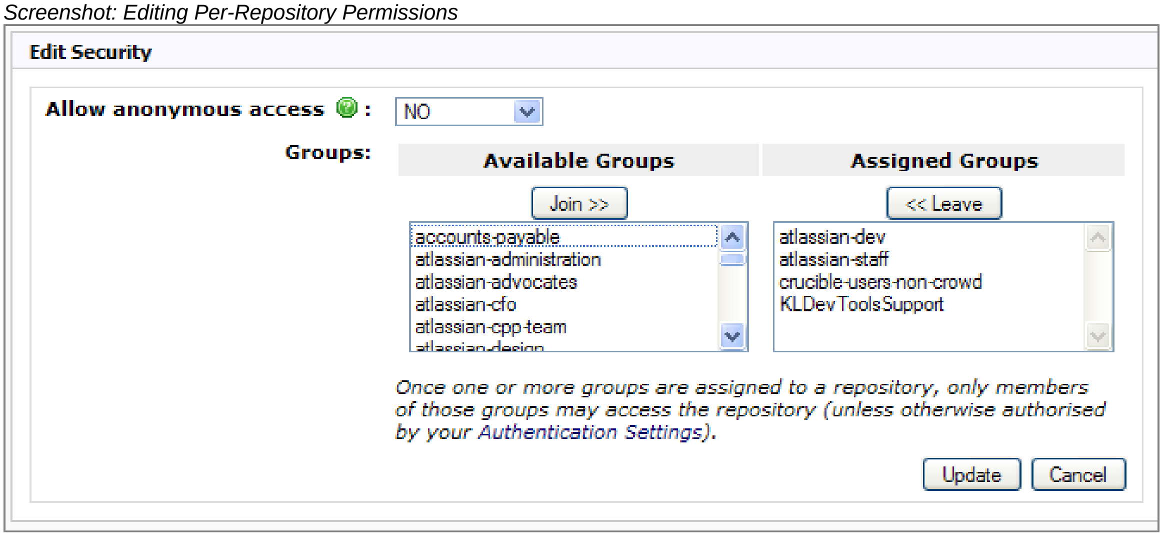


Permissions

Allow anonymous access: NO

Built-in groups: atlassian-dev
atlassian-staff
crucible-users-non-crowd
KLDevToolsSupport

Screenshot: Editing Per-Repository Permissions



Edit Security

Allow anonymous access : NO

Groups:

Available Groups	Assigned Groups
accounts-payable atlassian-administration atlassian-advocates atlassian-cfo atlassian-cpp-team atlassian-design	atlassian-dev atlassian-staff crucible-users-non-crowd KLDevToolsSupport

Once one or more groups are assigned to a repository, only members of those groups may access the repository (unless otherwise authorised by your Authentication Settings).

Watches

FishEye has a watch notification system that allows users to receive email notifications when commits are detected. The '**Watches**' repository option allows you to disable this functionality in the Repository Defaults and on a per-repository basis.

> ⚠ Watch functionality requires a valid SMTP server to be configured.

Allow (Process)

By default, FishEye will cache and index your whole repository, and present all of this information to users. You can control what parts of your repository FishEye will access, by setting the '**Allow (Process)**' repository option.

Includes

The '**Includes**' subsection 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.

ⓘ 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.
Examples:

- 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.

Excludes

The '**Excludes**' subsection 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. 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.

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.

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.

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.

Hidden Directories

The '**Hidden Dirs**' repository option allows you to mark unused (deprecated) directories as 'hidden'. Hidden directories will not appear in the FishEye user interface unless the user has specifically toggled 'Show hidden directories'. FishEye will still index and cache these directories.

This can be useful if you have old directories that you don't want cluttering the screens 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.

You can selectively disable the creation of tarballs for certain directories or directory trees.

Properties

The '**Properties**' repository option allows you to customise the behaviour of FishEye. Specifically, you can remove the graph and calendars from certain screens.

A property may be set either per repository or globally as a repository default. A repository default property is inherited by all repositories. A default property may be overridden at the repository level.

The following properties are supported:

Name	Possible Values	Default Value	Description
show-changelog-calendar	true, false	false	If set to false, the calendar is disabled on the Changelog page. This may be required for performance reasons. The revision totals displayed per calendar day, month and year may be expensive to calculate. For repositories with a lot of historical data, disabling the calendar can result in significant performance improvements when viewing the Changelog page.
enable-line-history	true, false	true	Allows you to disable (hide) the line-count history graph on the Browse and Changelog pages. This may be desirable if you have a large repository and generating the line graphs takes a long time.

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.

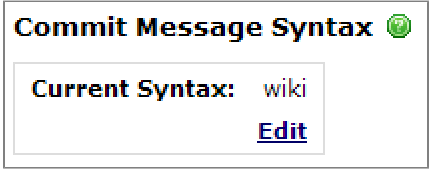
Adjusting the Global Default Setting

1. Go to '**Administration**' > '**Repository Defaults**', then scroll down to the '**Commit Message Syntax**' option.
2. Click '**Edit**' to make changes. You can choose from the following options:

- **Wiki Markup**: Wiki Markup is rendered in Commit messages.
- **Plain text**: Wiki Markup is not rendered in FishEye commit messages.

The factory setting is '**plain**', i.e. FishEye will not render commit messages.

Screenshot: Setting Global Wiki Rendering for Commit Messages



Commit Message Syntax

Current Syntax: wiki

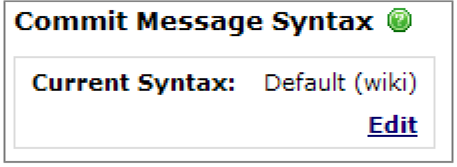
Adjusting the Per-Repository Setting

To adjust this, go to '**FishEye Administration**' > '**Repository Options**' > '**Commit Message Syntax**'.

You can choose from the following options:

- **Wiki Markup**: Wiki Markup is rendered in Commit messages.
- **Plain text**: Wiki Markup is not rendered in FishEye commit messages.

Screenshot: Setting Per-Repository Wiki Rendering for Commit Messages



Commit Message Syntax

Current Syntax: Default (wiki)

Edit

Configuring ViewVC Compatibility

For backwards-compatibility and legacy system support, FishEye contains a URL-compatibility mode with the ViewVC (formerly known as ViewCVS) and CVSWeb tools. This allows FishEye to supplant or extend ViewVC, making use of the URLs set up for ViewVC.

To configure ViewVC compatibility, click '**ViewCVS URL Mappings**' on the '**Admin Menu**'.

For example, a ViewVC URL of the form

```
http://host/viewcvs.cgi/x/y/z
```

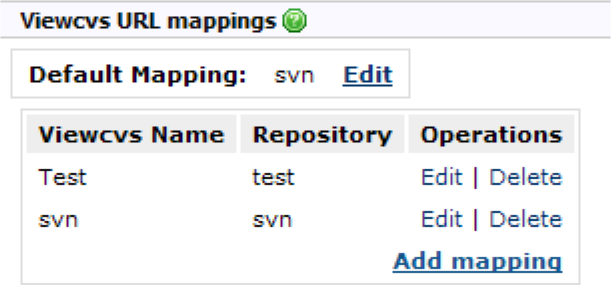
can be viewed in FishEye at

```
http://fisheyehost/viewcvs/x/y/z
```

FishEye can be configured to determine exactly how it provides this compatibility mode. In particular, you can configure how to map ViewVC repository names (`cvsroot` or `root` in the query parameter) to FishEye repository names.

The **Default Mapping** can be used to configure which repository to use if no repository is specified in the URL. If a repository name is given in the URL, you can tell FishEye how to translate that to the name of a FishEye repository. Otherwise, FishEye will attempt to use the repository name given in the URL directly.

Screenshot: Configuring ViewVC Compatibility



Viewcvs URL mappings

Default Mapping: svn Edit

Viewcvs Name	Repository	Operations
Test	test	Edit | Delete
svn	svn	Edit | Delete

Add mapping

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

Configuring SMTP

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

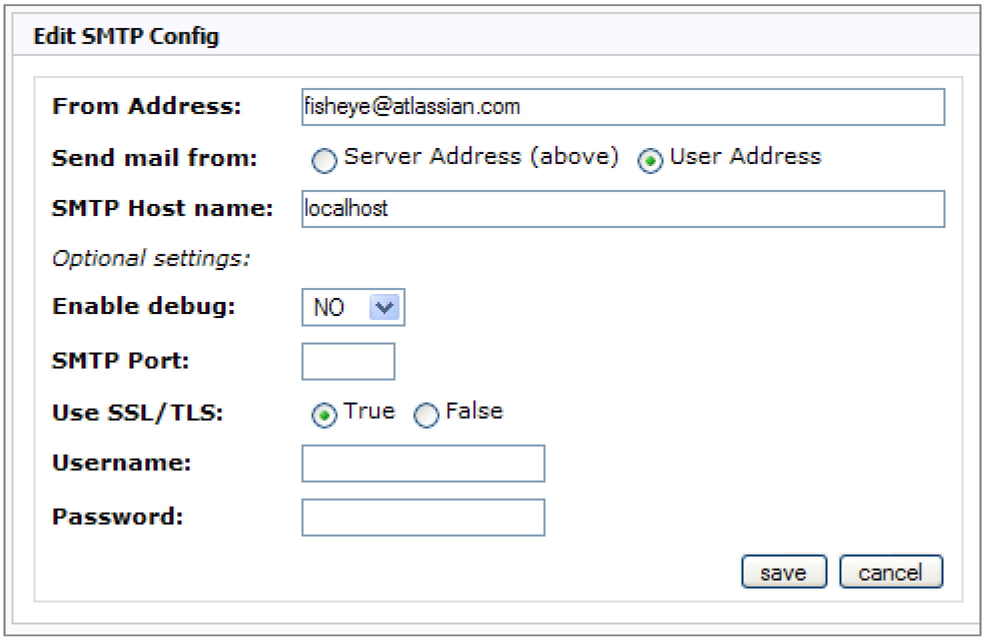
You can enter the following parameters:

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. (**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



Edit SMTP Config

From Address: fisheye@atlassian.com

Send mail from: ○ Server Address (above) ● User Address

SMTP Host name: localhost

Optional settings:

Enable debug: NO

SMTP Port:

Use SSL/TLS: ● True ○ False

Username:

Password:

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.

Submitting Feedback

We're very interested in your feedback. The best place for submitting feedback is the FishEye forums.

Related Links

- Git Configuration
- Crucible Repository Configuration
- General Crucible Configuration

Mercurial Client

⚠ The information on this page relates to FishEye's alpha support for Mercurial. Do not use in production.

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

Introduction and Disclaimer

Atlassian stresses that this is not final support for Mercurial which means the following:

1. Subsequent updates to the technology are likely to require re-indexing.
2. 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 Mercurial support in FishEye.

Configuration

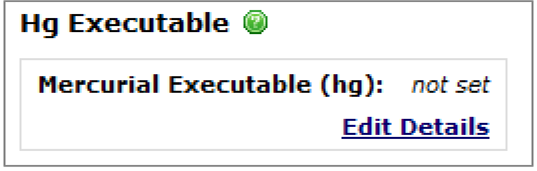
To set the location of the Mercurial executable, carry out the following steps:

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

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

ℹ FishEye requires Mercurial 1.5.1 or later.

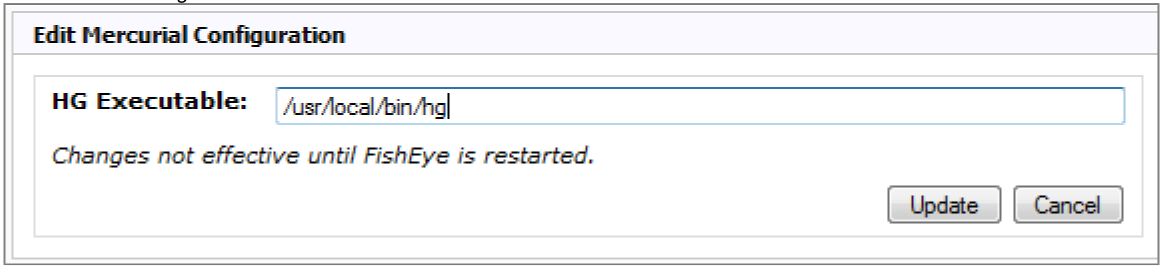
Screenshot: Mercurial Executable Settings



Hg Executable

Mercurial Executable (hg): *not set*

Screenshot: Setting the Mercurial Executable Location



Edit Mercurial Configuration

HG Executable: /usr/local/bin/hg

Changes not effective until FishEye is restarted.

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. ⚠ 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

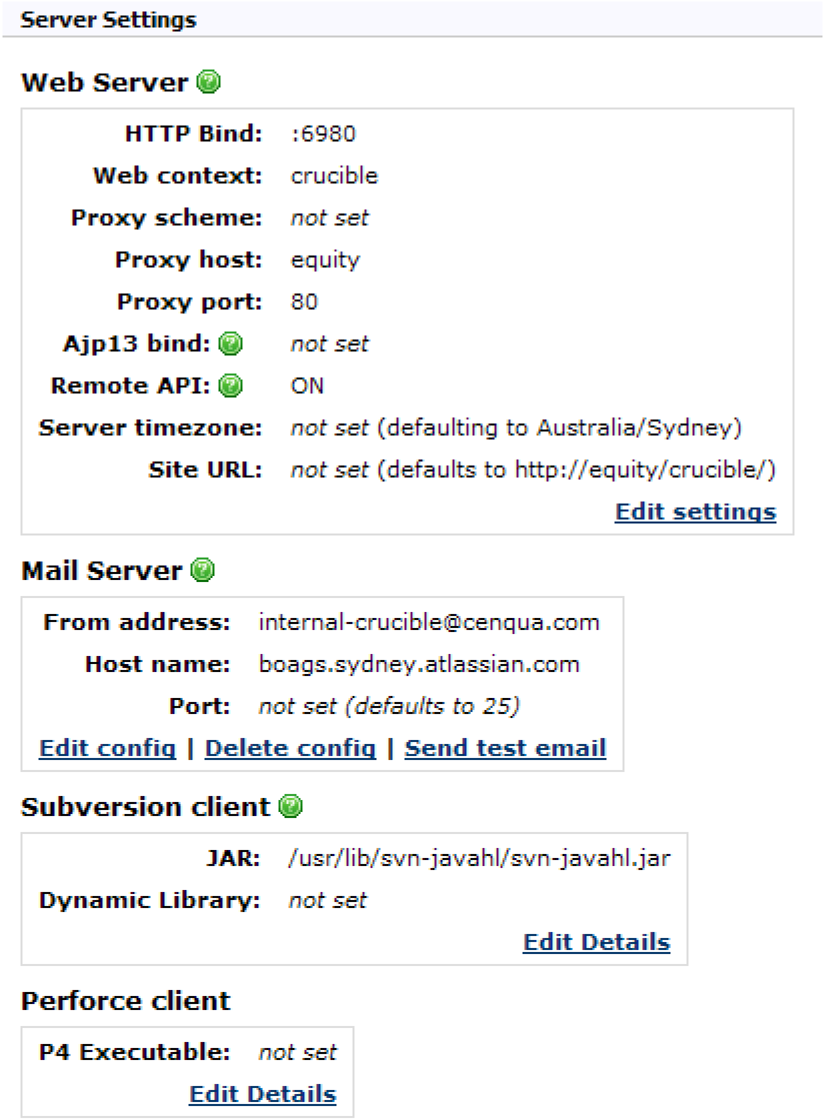
Native Subversion Clients are also supported by FishEye as an alternative to SVNkit. This process requires additional configuration. Read the Native Subversion Client instructions for more information.

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



Server Settings

Web Server

HTTP Bind: :6980
Web context: crucible
Proxy scheme: *not set*
Proxy host: equity
Proxy port: 80
Ajp13 bind: *not set*
Remote API: ON
Server timezone: *not set* (defaulting to Australia/Sydney)
Site URL: *not set* (defaults to http://equity/crucible/)

Edit settings

Mail Server

From address: internal-crucible@cenqua.com
Host name: boags.sydney.atlassian.com
Port: *not set (defaults to 25)*

Edit config | Delete config | Send test email

Subversion client

JAR: /usr/lib/svn-javahl/svn-javahl.jar
Dynamic Library: *not set*

Edit Details

Perforce client

P4 Executable: *not set*

Edit Details

Configuring the FishEye Web Server

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

> ⚠ 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:
	- **Host name and port number**:
	`hostname:8060`
	- **port number only** (requires a leading colon):
	`:8060`
	- **IP address and port number**:
	`10.0.0.11:60`
	(*At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.*)
	⚠ Do not add http:// to the front i.e. do not add http://hostname:8060 or http://10.0.0.11:60. Simply define hostname:port or IpAddress:port

Restart required	Yes

Setting	Web context
Description	By default, the FishEye application can be accessed via `http://HOST:PORT/`, where HOST and PORT are defined as above. You can force the FishEye application to be hosted on a different 'context' or 'path' by specifying a value here.
Example	If you specify a web context of '**fisheye**' then FishEye will be accessible from `http://HOST:PORT/fisheye/` instead of `http://HOST:PORT/`.
Restart required	Yes

Setting	Proxy scheme
Description	Can be set if you are forwarding through to FishEye from another web server.
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:
	• **Host name and port number**:
	`hostname:8009`
	• **port number only** (requires a leading colon):
	`:8009`
	• **IP address and port number**:
	`10.0.0.11:8009`
	(*At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.*)
	⚠ Do not add http:// to the front i.e. do not add http://hostname:8060 or http://10.0.0.11:60. Simply define hostname:port or IpAddress:port

Restart required	Yes

Setting	**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.

Integrating with Other Web Servers

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.

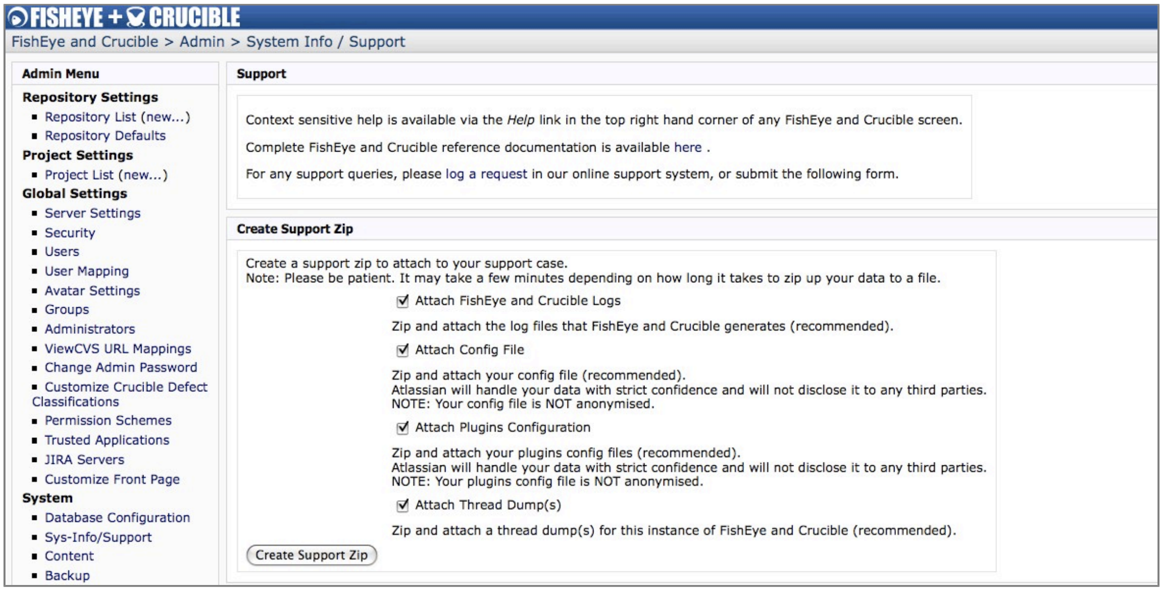
Support Utility

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.

The Create Support Zip form is shown below:



Context sensitive help is available via the *Help* link in the top right hand corner of any FishEye and Crucible screen.

Complete FishEye and Crucible reference documentation is available here .

For any support queries, please log a request in our online support system, or submit the following form.

Create a support zip to attach to your support case.
Note: Please be patient. It may take a few minutes depending on how long it takes to zip up your data to a file.

☑ Attach FishEye and Crucible Logs

Zip and attach the log files that FishEye and Crucible generates (recommended).

☑ Attach Config File

Zip and attach your config file (recommended).
Atlassian will handle your data with strict confidence and will not disclose it to any third parties.
NOTE: Your config file is NOT anonymised.

☑ Attach Plugins Configuration

Zip and attach your plugins config files (recommended).
Atlassian will handle your data with strict confidence and will not disclose it to any third parties.
NOTE: Your plugins config file is NOT anonymised.

☑ Attach Thread Dump(s)

Zip and attach a thread dump(s) for this instance of FishEye and Crucible (recommended).

Raising a Support Ticket via the Fisheye Administration Console

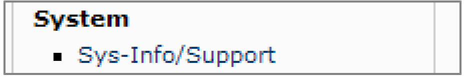
Ensure that SMTP email is set up on your Fisheye instance and your mail server allows zip files.

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



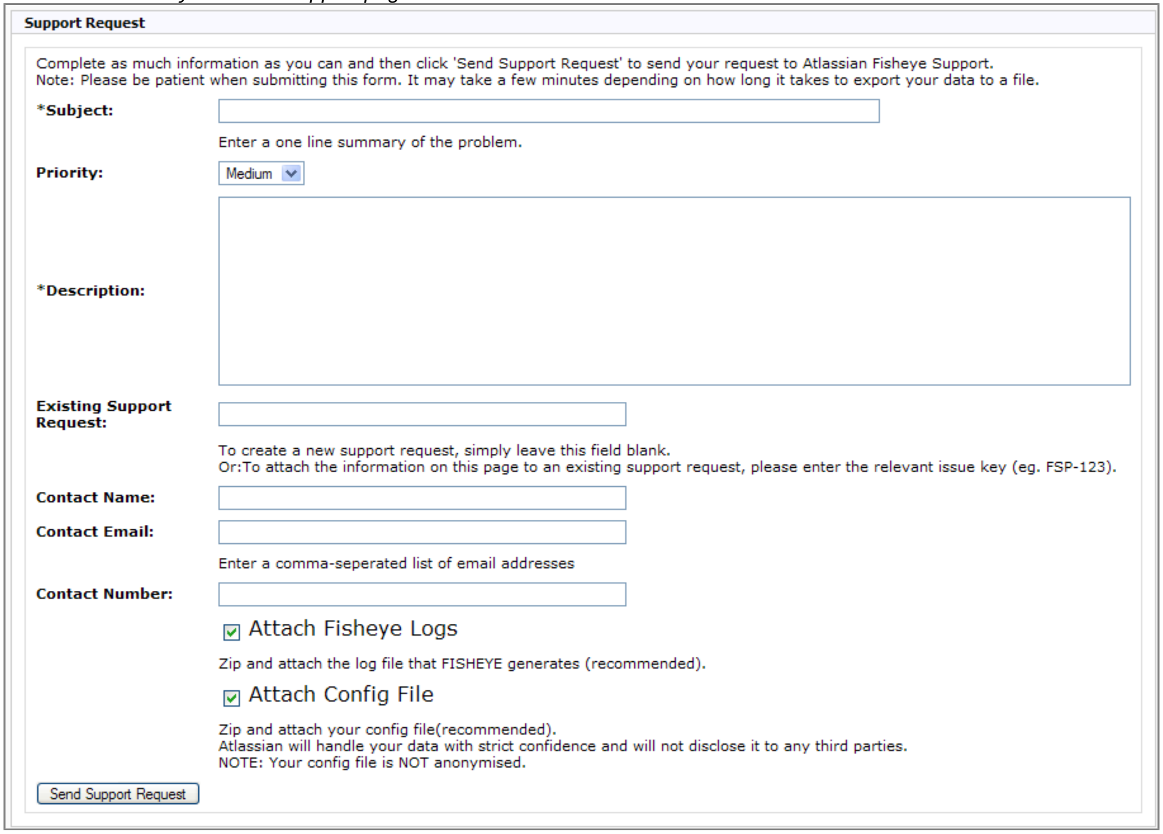
System

- Sys-Info/Support

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



Support Request

Complete as much information as you can and then click 'Send Support Request' to send your request to Atlassian Fisheye Support.
Note: Please be patient when submitting this form. It may take a few minutes depending on how long it takes to export your data to a file.

***Subject:**

Enter a one line summary of the problem.

Priority: Medium

***Description:**

Existing Support Request:

To create a new support request, simply leave this field blank.
Or:To attach the information on this page to an existing support request, please enter the relevant issue key (eg. FSP-123).

Contact Name:

Contact Email:

Enter a comma-seperated list of email addresses

Contact Number:

☑ Attach Fisheye Logs

Zip and attach the log file that FISHEYE generates (recommended).

☑ Attach Config File

Zip and attach your config file(recommended).
Atlassian will handle your data with strict confidence and will not disclose it to any third parties.
NOTE: Your config file is NOT anonymised.

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.

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.

Migrating to an External Database

This page contains instructions on migrating your FishEye database from its default embedded form to an external database. This may be useful for the following reasons:

- **Improved Protection Against Data Loss**: 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.
- **Performance & Scalability**: If you have many 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:

Overview

FishEye and Crucible offer alternatives to the built-in HSQLDB database for storing its relational data. At the time of writing, MySQL Enterprise Server and PostgreSQL are supported (see Supported Platforms for version numbers). This page outlines the steps required for switching to an external database.

Migrating to MySQL Enterprise Server

To switch from the built-in HSQLDB database to MySQL Enterprise Server, install MySQL Enterprise Server and follow the steps below.

1. The JDBC drivers for MySQL Enterprise Server are bundled with FishEye. 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 or Crucible to have it pick up the new driver.

2. Create a UTF-8 Database:

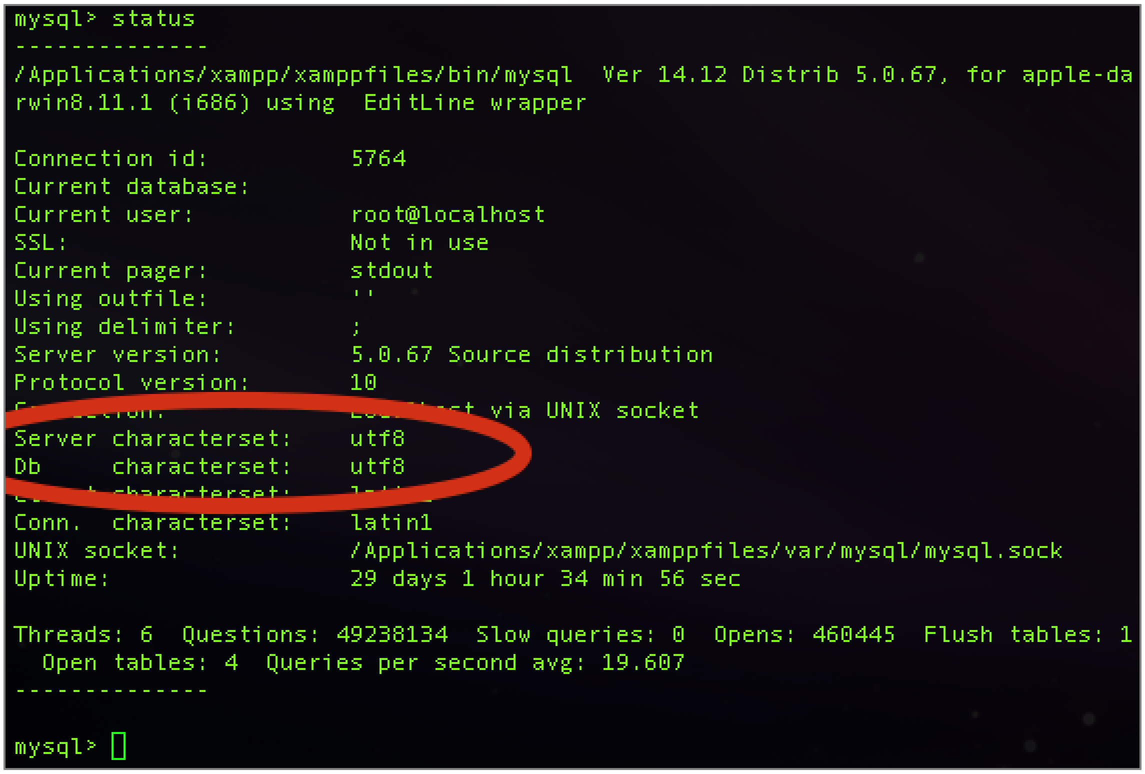
```
CREATE DATABASE fisheye CHARACTER SET utf8 COLLATE utf8_bin;
```

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

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

4. Use the **`status`** command to verify database character encoding information:

Screenshot: Using the MySQL Enterprise Server Status Command



```
mysql> status
--------------
/Applications/xampp/xamppfiles/bin/mysql  Ver 14.12 Distrib 5.0.67, for apple-da
rwin8.11.1 (i686) using  EditLine wrapper

Connection id:          5764
Current database:
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.0.67 Source distribution
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    utf8
Db     characterset:    utf8
Client characterset:    latin1
Conn.  characterset:    latin1
UNIX socket:            /Applications/xampp/xamppfiles/var/mysql/mysql.sock
Uptime:                 29 days 1 hour 34 min 56 sec

Threads: 6  Questions: 49238134  Slow queries: 0  Opens: 460445  Flush tables: 1
  Open tables: 4  Queries per second avg: 19.607
--------------

mysql> 
```

5. Create a user that can log in from the host that Crucible or FishEye 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.

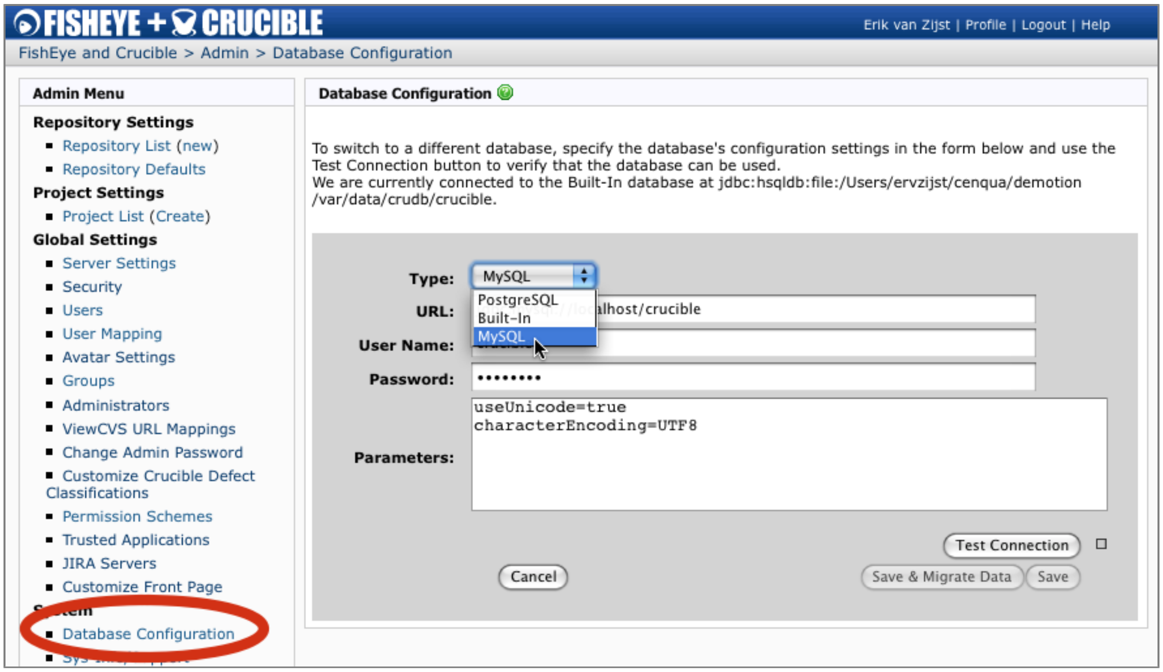
For instance, when FishEye and MySQL Enterprise Server run on the same machine (accessible through `localhost`), issue the following commands (replacing username and password with the appropriate values):

```
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)
```

6. With the database prepared, navigate to the '**Database Configuration**' section in the admin interface, select MySQL Enterprise Server from the drop down and fill out the database URL, username and password.

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

Screenshot: Testing the Connection



Database Configuration

To switch to a different database, specify the database's configuration settings in the form below and use the Test Connection button to verify that the database can be used.
We are currently connected to the Built-In database at jdbc:hsqldb:file:/Users/ervzijst/cenqua/demotion /var/data/crudb/crucible.

Type: MySQL

URL: lhost/crucible

User Name:

Password: ••••••••

Parameters:

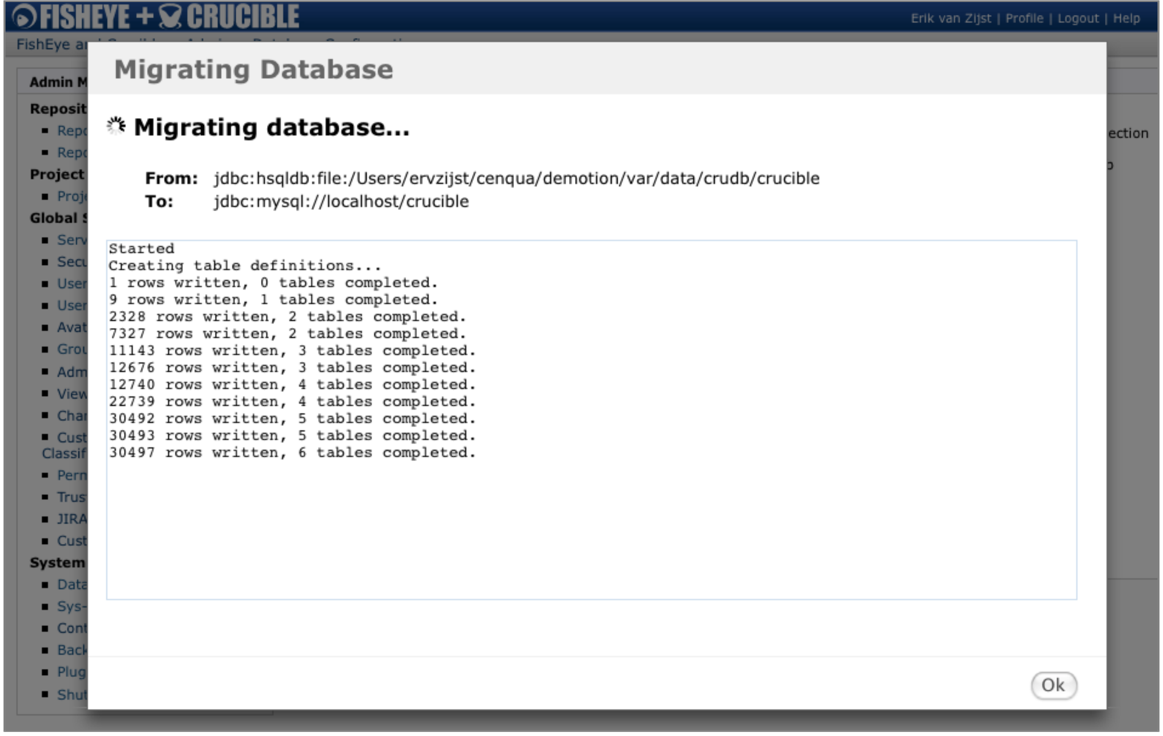
```
useUnicode=true
characterEncoding=UTF8
```

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 Crucible or FishEye is running on and that all the required privileges are present.

7. 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 will not switch to the new database and report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration.

Screenshot: Migrating the Database



Migrating Database

Migrating database...

From: jdbc:hsqldb:file:/Users/ervzijst/cenqua/demotion/var/data/crudb/crucible
To: jdbc:mysql://localhost/crucible

```
Started
Creating table definitions...
1 rows written, 0 tables completed.
9 rows written, 1 tables completed.
2328 rows written, 2 tables completed.
7327 rows written, 2 tables completed.
11143 rows written, 3 tables completed.
12676 rows written, 3 tables completed.
12740 rows written, 4 tables completed.
22739 rows written, 4 tables completed.
30492 rows written, 5 tables completed.
30493 rows written, 5 tables completed.
30497 rows written, 6 tables completed.
```

Migrating to PostgreSQL

To switch from the built-in HSQLDB database to PostgreSQL, install PostgreSQL and follow the steps below.

1. The JDBC drivers for PostgreSQL are bundled with FishEye. 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 or 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 fisheye ENCODING 'UTF-8' OWNER username;
```

4. Make sure the user has full access to the database:

```
> grant all on database fisheye to username;
```

5. With the database prepared, navigate to the '**Database Configuration**' section in the admin interface, select PostgreSQL from the drop down and fill in the details for database URL, username and password.

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

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 Crucible or FishEye 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 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.

Support for other Databases

FishEye and Crucible currently ship with support for MySQL Enterprise Server and PostgreSQL as external databases (see Supported Platforms for version numbers).

If you are looking for support for Oracle or Microsoft SQL Server, please vote for the issues below. Your vote will help us prioritise them.

- **Request Oracle Support**: CRUC-1489
- **Request MS-SQL Support**: CRUC-1407

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.

Software Update Notifications

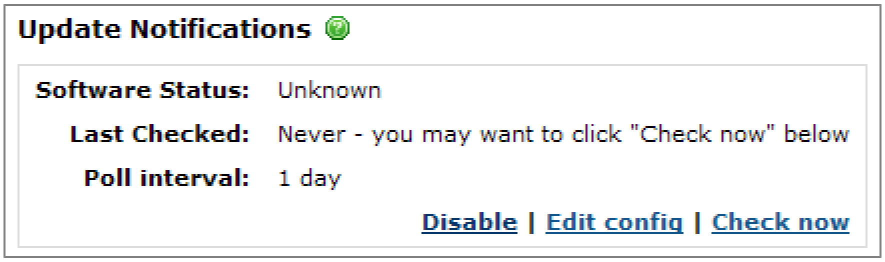
FishEye (and Crucible) can detect and notify you of new versions of the program. In the Admin screen, you can set a background thread to poll Atlassian's servers and report when a new version of FishEye or Crucible has been released.

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



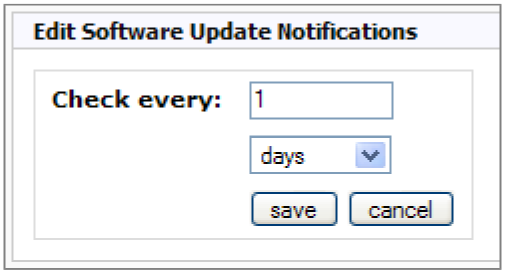
Update Notifications

Software Status: Unknown

Last Checked: Never - you may want to click "Check now" below

Poll interval: 1 day

Screenshot: Changing the Update Notification Interval



Edit Software Update Notifications

Check every: 1 days

Running Scheduled Events

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

JIRA Integration in FishEye

This page contains instructions for setting up JIRA integration.

JIRA is Atlassian's issue tracking product, which can be used to manage projects and associated work. JIRA issues can be viewed in the main Dashboard view. This requires you to enter details on the required JIRA server(s) via the administration screens.

Before you begin: Ensure that you configure your JIRA instance to enable sub-tasks and allow Remote API access; it is also recommended that you enable unassigned issues.

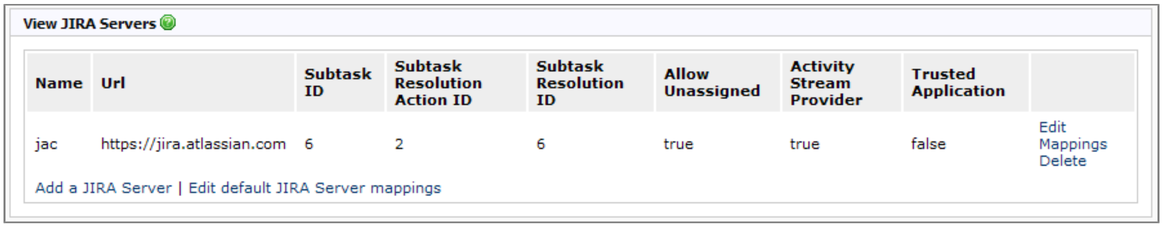
The instructions on this page have been tested with JIRA 4.0.0.

On this page:

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

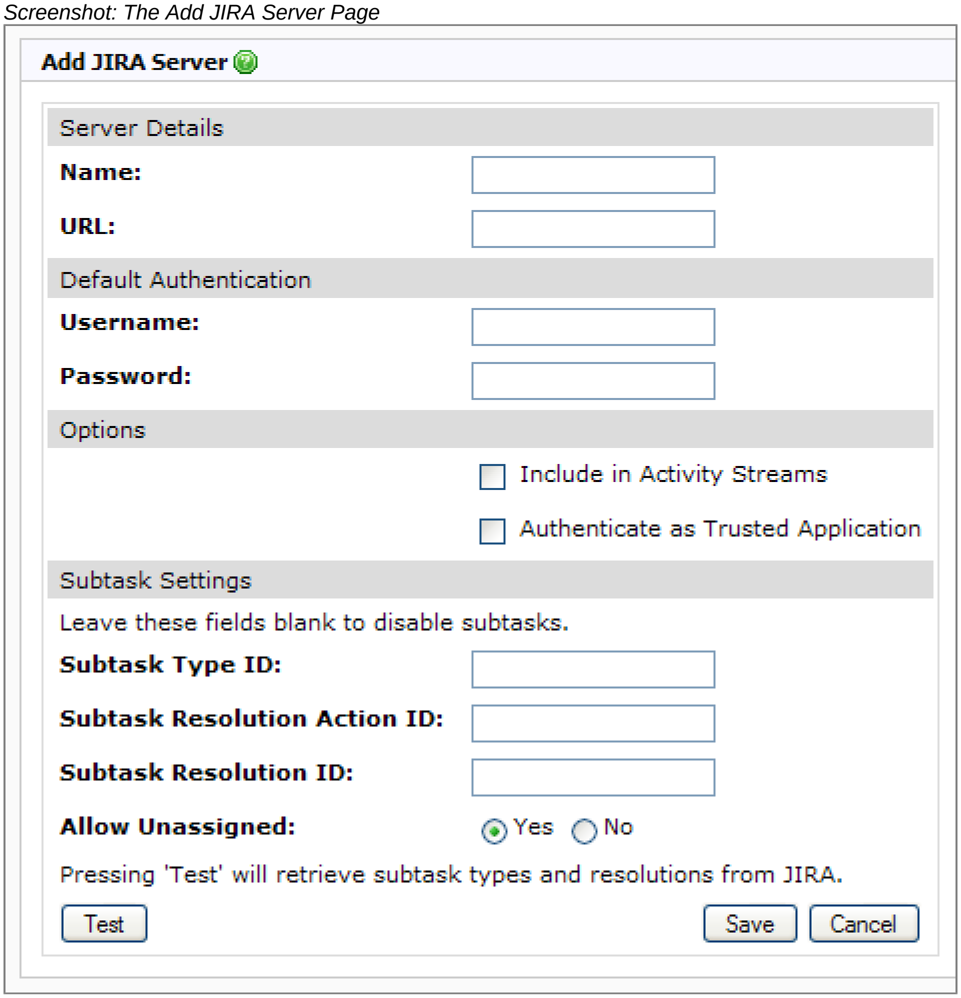
On the View JIRA Servers page, you can carry out a number of operations as described below.

Adding a New JIRA Server

To add a new JIRA server from the View JIRA Servers page, click '**Add JIRA Server**'.

The '**Add JIRA Server**' page opens.

Screenshot: The Add JIRA Server Page



Add JIRA Server

Server Details

Name:

URL:

Default Authentication

Username:

Password:

Options

□ Include in Activity Streams

□ Authenticate as Trusted Application

Subtask Settings

Leave these fields blank to disable subtasks.

Subtask Type ID:

Subtask Resolution Action ID:

Subtask Resolution ID:

Allow Unassigned: ● Yes ○ No

Pressing 'Test' will retrieve subtask types and resolutions from JIRA.

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 base URL of the JIRA server, e.g. http://jira.atlassian.com	Yes
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 in the Activity Streams.	No
Authenticate as Trusted Application	Check Box	Allows the system to interface with JIRA, letting users log on with their own accounts (and use their own accounts on the JIRA server). If you enable subtasks using a trusted JIRA instance, you will also need to add FishEye as a trusted application to your JIRA instance, and then add the root path (denoted by '**/**') to the **URL Paths to Allow** field on the trusted applications screen in the administration interface of your JIRA instance. For more information, see the FishEye documentation and JIRA documentation about trusted applications. Also see the Known Issues.	No
Subtask Type ID	Number	This is required to enable creating issues from a Crucible comment (not required for standalone FishEye). See the instructions on obtaining this number.	No
Subtask Resolution Action ID	Number	This is required to enable creating issues from a Crucible comment (not required for standalone FishEye). See the instructions on obtaining this number (not required for standalone FishEye).	No
Subtask Resolution ID	Number	This is required to enable creating issues from a Crucible comment (not required for standalone FishEye). See the instructions on obtaining this number (not required for standalone FishEye).	No
Allow Unassigned	True/False Button	Allow unassigned sub-tasks. This setting must match your 'Allow Unassigned Issues ' setting in JIRA. .	No

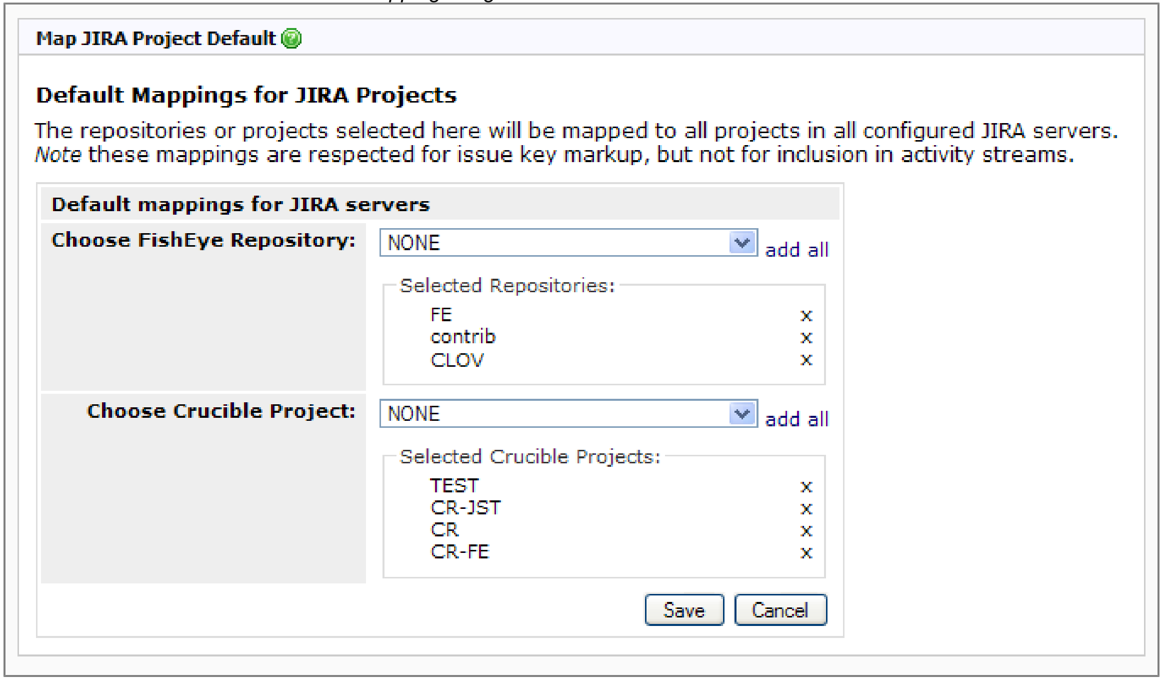
Once you've filled out the necessary fields, click '**Test**' to ensure that your details are correct. If you have a positive message return from the test, click '**Save**'.

Editing Default JIRA Server Mappings

ⓘ This setting has two effects; it enables the feature that shows JIRA information in a dynamic window when you hover the mouse over a JIRA issue key; it will also turn every issue key into a hyperlink to that issue.

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



Map JIRA Project Default

Default Mappings for JIRA Projects

The repositories or projects selected here will be mapped to all projects in all configured JIRA servers. *Note* these mappings are respected for issue key markup, but not for inclusion in activity streams.

Default mappings for JIRA servers

Choose FishEye Repository: NONE add all

Selected Repositories:
FE x
contrib x
CLOV x

Choose Crucible Project: NONE add all

Selected Crucible Projects:
TEST x
CR-JST x
CR x
CR-FE x

On this page, select the FishEye Repositories or Crucible Projects that you wish to associate with all the JIRA servers that you have configured for use in FishEye/Crucible. You can click '**add all**' to quickly include them all in this category. You can remove individual items by clicking the small 'X' marks.

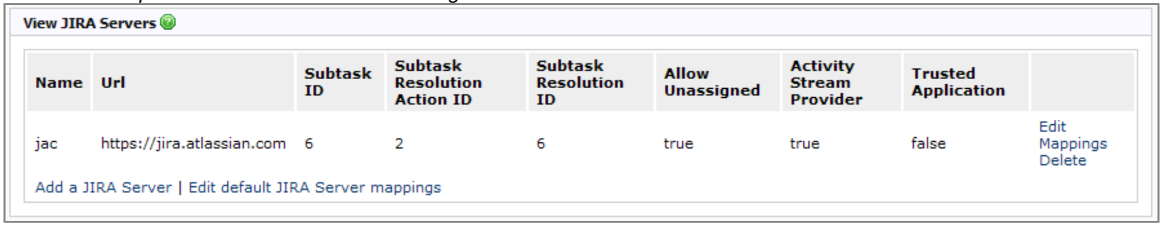
Once you've finished, click '**Save**'.

⚠ You should disable any existing Crucible linkers you have set up for JIRA, as they will override this feature and prevent the dynamic dialog box from appearing when you mouse over an issue.

Operations on Existing Servers

Once you have configured an existing JIRA server, there are three main operations you can carry out on it: '**Edit**', '**Mappings**' and '**Delete**'. These options appear on the far right of the screen.

Screenshot: Operations in the JIRA Servers Page



View JIRA Servers

Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete

Add a JIRA Server | Edit default JIRA Server mappings

Edit settings for an existing JIRA server

When you click '**Edit**', you can adjust any of the general settings you configured when you first added the server.

Edit mappings for an existing JIRA server

When you click '**Mappings**', a page is loaded that is almost identical to the '**Default Mapping**' screen, but allows you to choose mappings only for that specific JIRA server.

Delete an existing JIRA server

Clicking '**Delete**' will remove the server from the list.

Known Issues

If you elect to use Trusted Applications for authentication with your JIRA server, activity streams will be generated using the currently logged in user. However JIRA project mapping and issue key linking (including the associated "hovering" content) will be retrieved using the user specified on the JIRA Server configuration page in the FishEye administration section.

We are working towards supporting Trusted Applications for issue key linking and project mapping. If this issue is important to you, please vote for CRUC-1910.

See Also

- The JIRA documentation on Integrating JIRA with FishEye, which enables you to view FishEye data from within JIRA.
- The Crucible documentation on JIRA Integration in Crucible, which enables you to view JIRA data from within Crucible.

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.

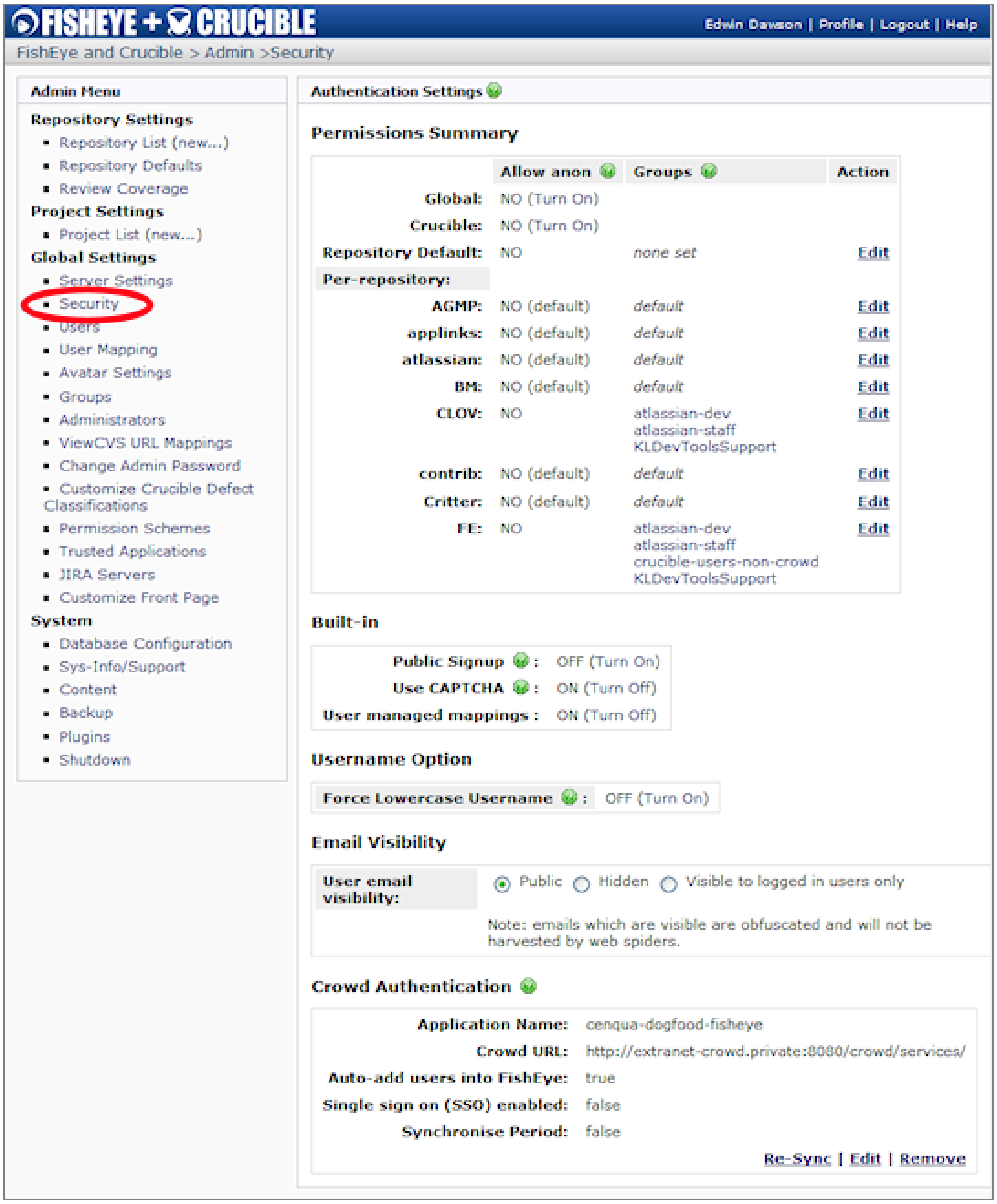
To configure your authentication settings, click '**Users/Security**' on the '**Admin Menu**'.

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
- Load all users from Active Directory, LDAP or Atlassian Crowd
- Understanding security

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

Screenshot: Authentication Settings



Authentication Settings

Permissions Summary

	Allow anon	Groups	Action
Global:	NO (Turn On)		
Crucible:	NO (Turn On)		
Repository Default:	NO	*none set*	Edit
Per-repository:			
AGMP:	NO (default)	*default*	Edit
applinks:	NO (default)	*default*	Edit
atlassian:	NO (default)	*default*	Edit
BM:	NO (default)	*default*	Edit
CLOV:	NO	atlassian-dev atlassian-staff KLDevToolsSupport	Edit
contrib:	NO (default)	*default*	Edit
Critter:	NO (default)	*default*	Edit
FE:	NO	atlassian-dev atlassian-staff crucible-users-non-crowd KLDevToolsSupport	Edit

Built-in

Public Signup : OFF (Turn On)
Use CAPTCHA : ON (Turn Off)
User managed mappings : ON (Turn Off)

Username Option

Force Lowercase Username : OFF (Turn On)

Email Visibility

User email visibility: ● Public ○ Hidden ○ Visible to logged in users only

Note: emails which are visible are obfuscated and will not be harvested by web spiders.

Crowd Authentication

Application Name: cenqua-dogfood-fisheye
Crowd URL: http://extranet-crowd.private:8080/crowd/services/
Auto-add users into FishEye: true
Single sign on (SSO) enabled: false
Synchronise Period: false

Re-Sync | Edit | Remove

Adding a User to a Group

There are two types of FishEye user groups:

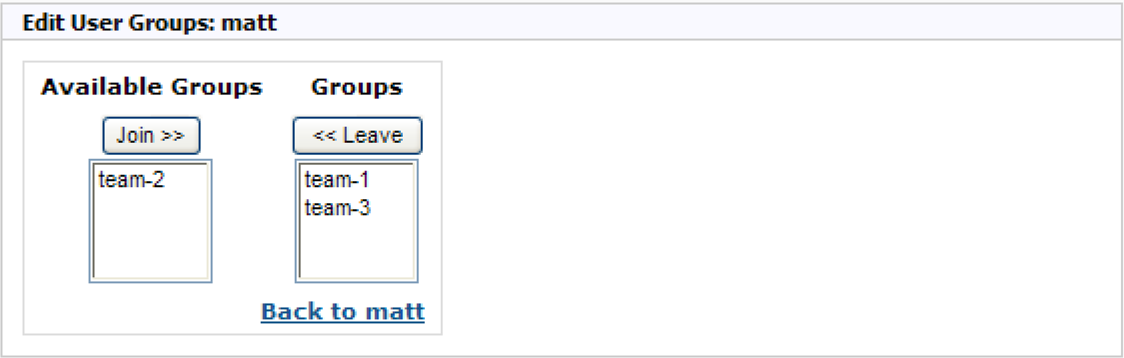
- 'Built-in' groups — these are stored in the FishEye database.
- 'External' groups — these are stored in an external directory (e.g. Crowd), if any are configured. See Configuring External Authentication Sources.

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 '**Edit**' link.
   ℹ If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click 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.
   ℹ 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

Unable to render embedded object: File (AuthenticationSettings.png) not found.

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.

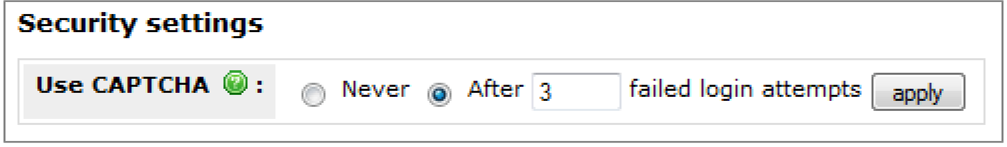
Configuring brute force login protection

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



Security settings

Use CAPTCHA : ○ Never ● After 3 failed login attempts

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

For each user who has a FishEye/Crucible account, an avatar image is displayed next to their name. By default, each user has a plain grey avatar.

Alternatively, you can allow users to upload their own avatar image by uploading an image to an external service, such as Gravatar.

To allow avatars from an external server,

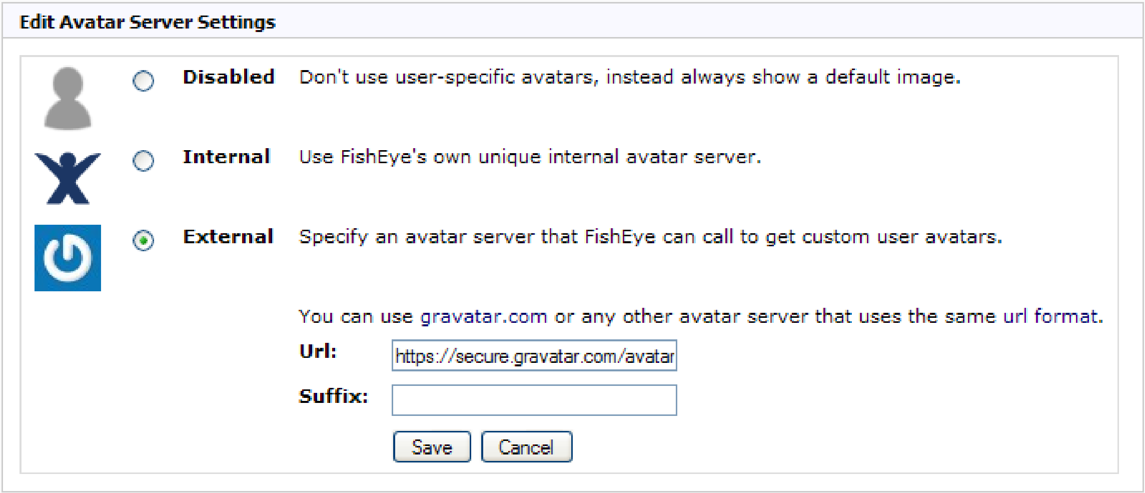
1. Go to the FishEye '**Admin Menu**'.
2. Click the '**Avatar Settings**' link in the left navigation column.
3. The '**Edit Avatar Server Settings**' page will be displayed (see screenshot below).
4. Change '**Select Type**', to '**External**'.
5. In the '**Url**' field, type the address of the external service, e.g. http://gravatar.com/avatar/
   For additional security, it is recommended that you specify the following:

   ```
   https://secure.gravatar.com/avatar/
   ```

   > When using an external avatar service, you should use the same protocol as is used to access FishEye. i.e. if FishEye uses the secure HTTP protocol then your Gravatar URL would include that prefix, as in the previous example. Conversely, if your FishEye instance uses regular HTTP, then your Gravatar URL should do the same.

6. In the '**Suffix**' field, you can optionally specify parameters as 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. For example "&r=g"
7. Click the '**Save**' button.
8. Users can now upload their own images to the external server.
   The user's login name to the external server must be the email address that is specified in their FishEye/Crucible User Profile .

Screenshot: 'Edit Avatar Server Settings'



Edit Avatar Server Settings

○ **Disabled** Don't use user-specific avatars, instead always show a default image.

○ **Internal** Use FishEye's own unique internal avatar server.

● **External** Specify an avatar server that FishEye can call to get custom user avatars.

You can use gravatar.com or any other avatar server that uses the same url format.

Url: https://secure.gravatar.com/avatar

Suffix:

> If you are using a web server in front of FishEye, you will have 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.

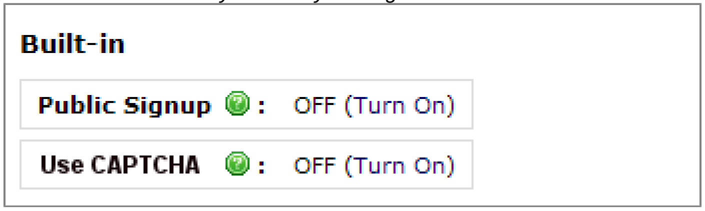
You may want to turn CAPTCHA off if you are serving FishEye or Crucible behind a firewall.

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



Built-in

Public Signup : OFF (Turn On)

Use CAPTCHA : OFF (Turn On)

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 '**Security**' on the '**Admin Menu**'. Only one external authentication source can be setup and each repository has the choice of authenticating against that or the internal login. To change authentication sources, you will need to remove the settings that are already configured, by clicking the '**Remove**' link. You will then be presented with the option to add a different authentication. FishEye currently supports:

- LDAP Authentication
- Host-Based Authentication - Implemented using PAM on Linux/Solaris/OS-X and Local/Domain Accounts on Windows
- AJPv13 Authentication
- Custom Authentication

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
```

Crowd Authentication

Atlassian's Crowd identity management system can be integrated with FishEye. Please see the document Integrating Crowd with FishEye in the *Crowd Administrator's Guide*.
Note:

- In FishEye versions 1.4 and later, support for Crowd is built in and configuration is greatly simplified.
- In FishEye versions 1.3.x and earlier, using Crowd required installing a custom authentication plugin on your FishEye server. In version 1.4, the custom authentication plugin is no longer required or supported.

⚠ The page System Requirements does not exist.

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:
	`# comments`
	`name1=value1`
	`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.

> 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.

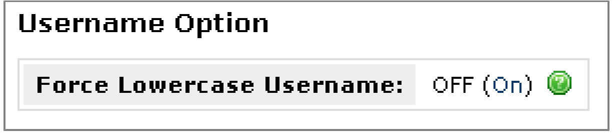
Forcing Imported Usernames to be Lowercase

When importing users from external authentication sources, you can set FishEye to force the usernames to become lowercase. This solves an issue with some sources adding duplicate users to the FishEye database.

To force lowercase usernames, carry out the following steps:

1. Log into FishEye's Admin Interface.
2. Under **Authentication settings**, the option **Force Lowercase Username** can be toggled on and off.
3. With this setting switched **On**, when new users are added to FishEye from an external authentication source, all usernames will be converted to lowercase.

Screenshot: Forcing Lowercase Usernames



Username Option

Force Lowercase Username: OFF (On)

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 Linux/Solaris/OS-X, and NT-based authentication on Windows.

Group Restrictions

FishEye can be configured to check if a user belongs to a group (or groups) before allowing access.

You can list one group name, or join several group names into a boolean expression like `group1 & (group2 | group3)`.

If your group name contains spaces or non-ASCII characters, then you need to use quotes. For example: `"Power Users" | Administrators`.

Windows

> If you are using Active Directory, you can configure FishEye to use LDAP as an alternative to host-based authentication.

If the computer FishEye is running on is **not** a member of a domain, then the Domain attribute is ignored.

When the computer is a member of a domain, you need to enter the full DNS name of the domain (e.g. `corp.example.com`). If you enter the short version of the domain (e.g. `corp`), then group-based restrictions may fail.

Once you have configured your settings, we recommend you use the '**Test**' function to ensure your access control behaves correctly.

PAM

On Linux, Solaris and OS-X, host-based authentication uses PAM (Pluggable Authentication Modules) to check users' passwords.

FishEye needs to be configured with the *service name* to use when conversing with PAM. You can create a new service name in the PAM configuration (typically `/etc/pam.conf` or `/etc/pam.d/`), or configure FishEye to use an existing service name (such as `other`, `login` or `xscreensaver`).

Some general operating-system specific tips are given below, but you should consult the PAM documentation for your operating system.

Once you have configured your settings, we recommend you use the '**Test**' function to ensure your access control behaves correctly.

Linux

On many Linux distributions, you may need to create a `/etc/pam.d/fisheye` file containing:

```
auth       required     pam_stack.so service=system-auth
```

Mac OS-X

On a default OS-X installation, you may need to create a `/etc/pam.d/fisheye` file containing:

```
auth       sufficient     pam_securityserver.so
auth       required       pam_deny.so
```

Solaris

If you are using the default `pam_unix_auth` PAM configuration on Solaris, then you may need to add a line like this to your `/etc/pam.conf` file:

```
fisheye auth requisite          pam_authtok_get.so.1
fisheye auth required           pam_unix_auth.so.1
```

If you test this and it does not work, it is probably because when using `pam_unix_auth` on Solaris, the process doing the password check needs read access to `/etc/shadow`.

Giving the FishEye process read access to this file may solve this problem, but using permissions other than `0400` for `/etc/shadow` is not recommended. You should discuss this with your system administrators first, and possibly change to a PAM module other than `pam_unix_auth`.

Global Settings

Global settings are:

Domain/Service name	**Windows:** the name of the domain. Leave blank to use the local computer. **PAM:** The service name in your PAM configuration to use. If blank, `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.

LDAP Authentication

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

On this page:

- Active Directory

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 `${REP}` variable is replaced with the name of the repository in question.
Match Type	One of 'user' (default) or 'any'. This setting modifies the meaning of LDAP restriction. If set to 'user', then FishEye expects the filter to match the exact DN of the current user. If it does match, then the user has access to the repository. Commonly, if your user object contains the list of groups the user has access to, then you would use a 'user' match. If set to 'any', then the filter just needs to match one result for the user to have access to the repository. Commonly, if your group object contains the list of UID members, then you would use an 'any' match. In such a case, your LDAP restriction filter may look like this: `(&(uniquemember=${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:

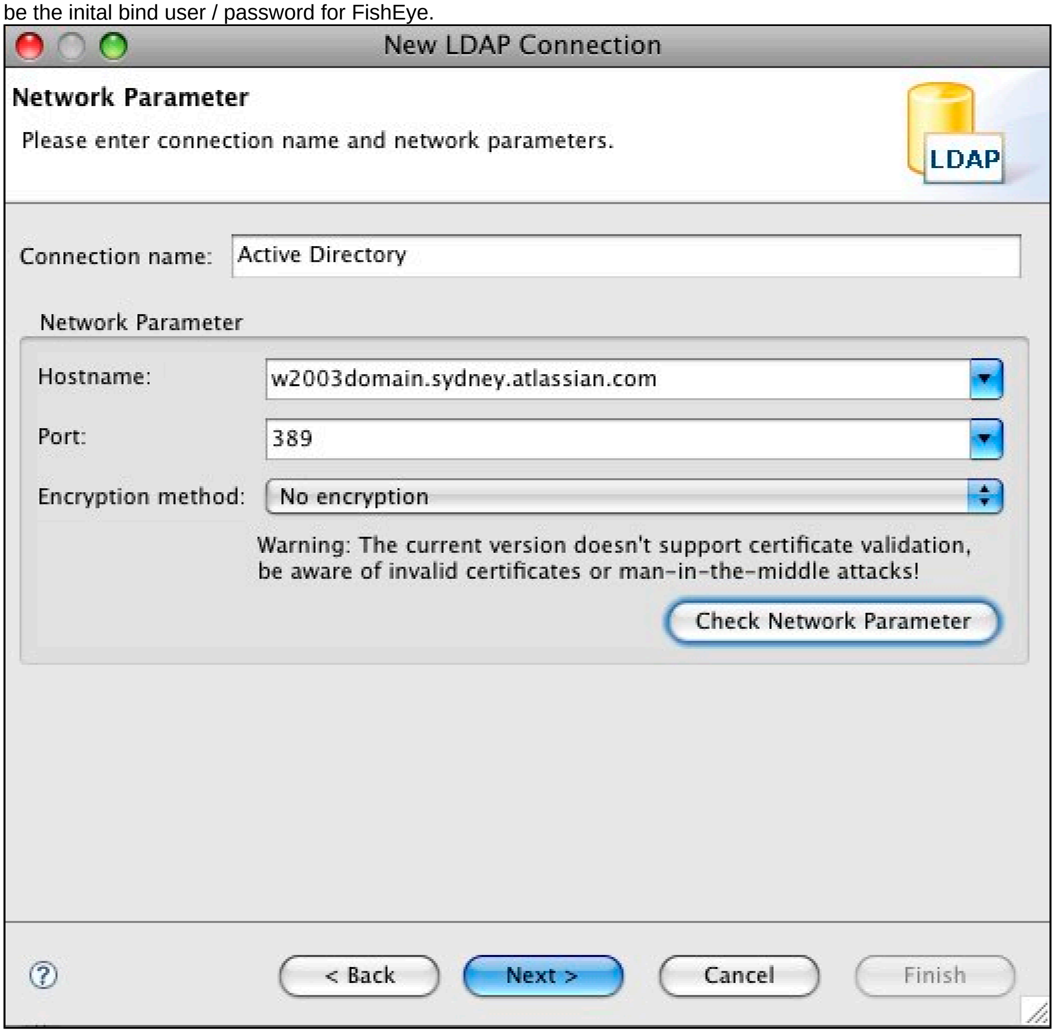
URL	`ldap://HOSTNAME:389`
Base DN	`DC=corp,DC=example,DC=com`
User Filter	`(sAMAccountName=${USERNAME})`
UID Attribute	`sAMAccountName`
Email attribute	`mail`
Initial bind DN	`corp.example.com/Users/SomeUser`

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

Network Parameter

Please enter connection name and network parameters.

Connection name: Active Directory

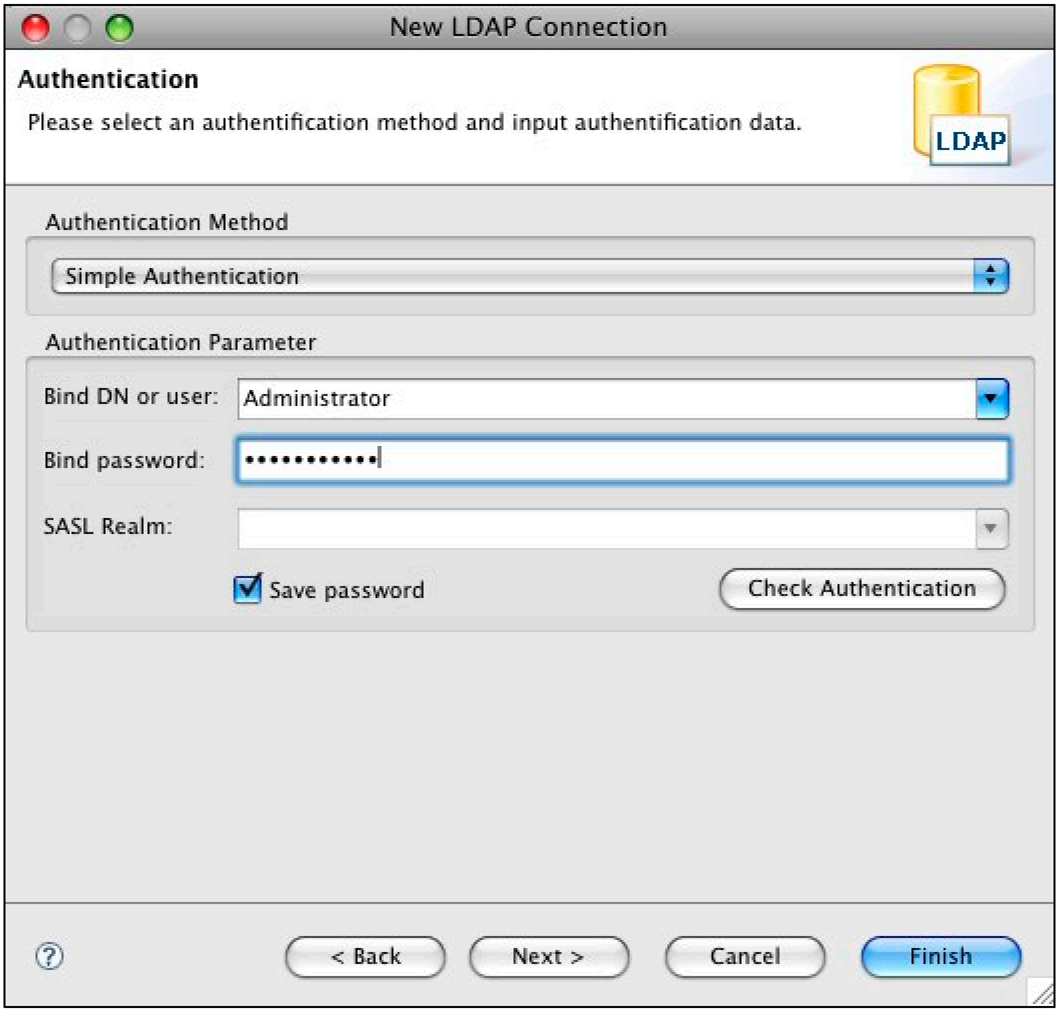
Network Parameter

Hostname: w2003domain.sydney.atlassian.com

Port: 389

Encryption method: No encryption

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



New LDAP Connection

Authentication

Please select an authentification method and input authentification data.

Authentication Method

Simple Authentication

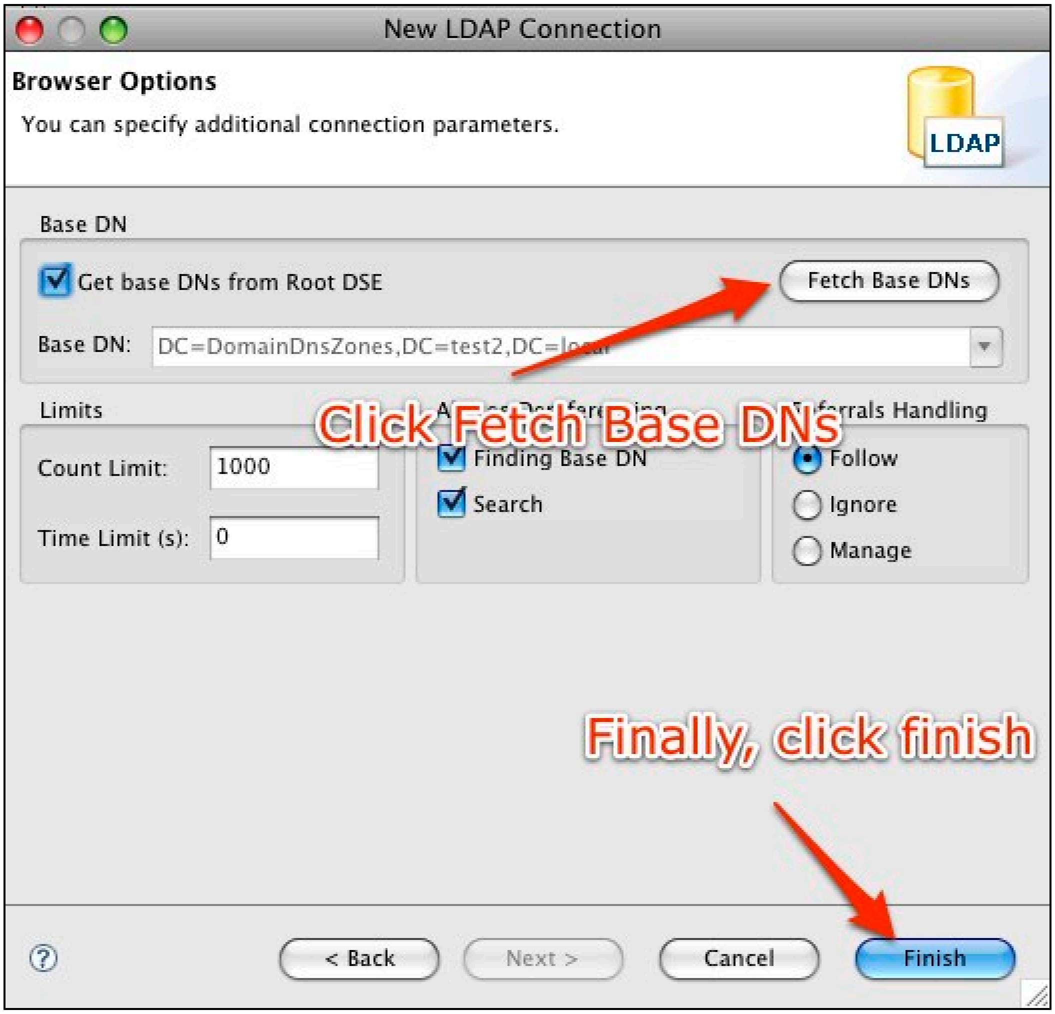
Authentication Parameter

Bind DN or user: Administrator

Bind password: ••••••••••••

SASL Realm:

☑ Save password



New LDAP Connection

Browser Options

You can specify additional connection parameters.

Base DN

☑ Get base DNs from Root DSE

Base DN: DC=DomainDnsZones,DC=test2,DC=local

Limits

Count Limit: 1000

Time Limit (s): 0

Aliases Dereferencing

☑ Finding Base DN

☑ Search

Referrals Handling

● Follow

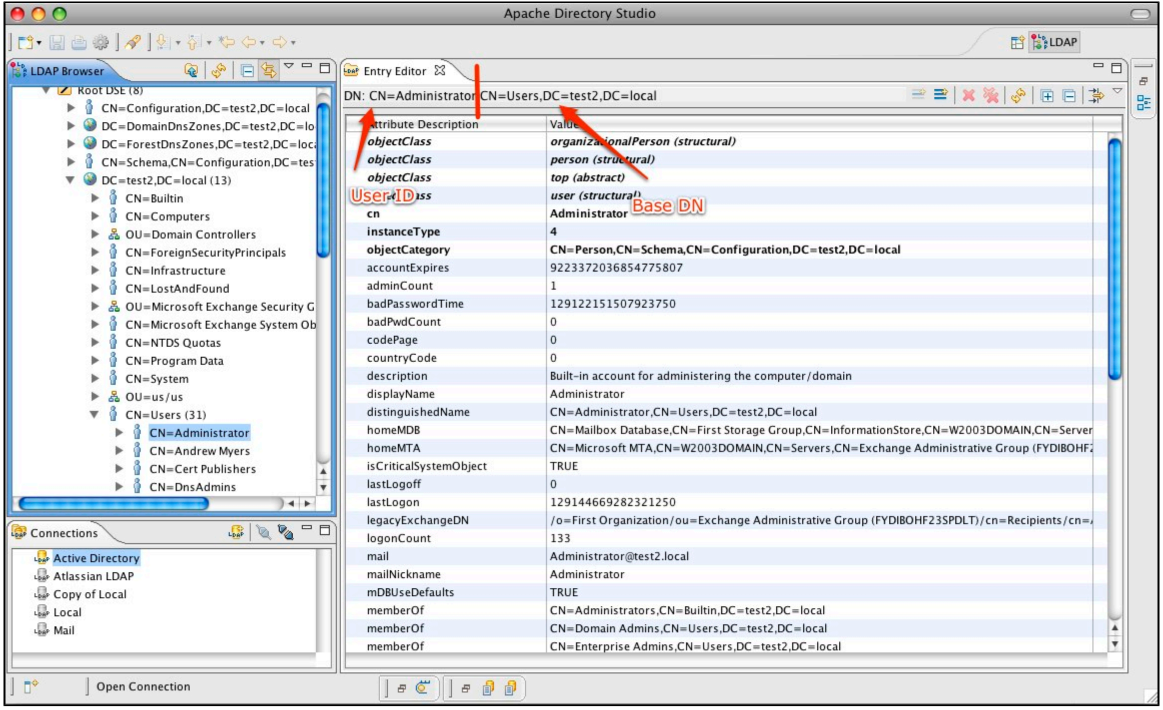
○ Ignore

○ Manage

Click Fetch Base DNs

Finally, click 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.



Apache Directory Studio

LDAP Browser

- Root DSE (8)
  - CN=Configuration,DC=test2,DC=local
  - DC=DomainDnsZones,DC=test2,DC=lo
  - DC=ForestDnsZones,DC=test2,DC=loca
  - CN=Schema,CN=Configuration,DC=tes
  - DC=test2,DC=local (13)
    - CN=Builtin
    - CN=Computers
    - OU=Domain Controllers
    - CN=ForeignSecurityPrincipals
    - CN=Infrastructure
    - CN=LostAndFound
    - OU=Microsoft Exchange Security G
    - CN=Microsoft Exchange System Ob
    - CN=NTDS Quotas
    - CN=Program Data
    - CN=System
    - OU=us/us
    - CN=Users (31)
      - CN=Administrator
      - CN=Andrew Myers
      - CN=Cert Publishers
      - CN=DnsAdmins

Connections

- Active Directory
- Atlassian LDAP
- Copy of Local
- Local
- Mail

Entry Editor

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

Attribute Description	Value
objectClass	*organizationalPerson (structural)*
objectClass	*person (structural)*
objectClass	*top (abstract)*
objectClass	*user (structural)*
cn	**Administrator**
instanceType	**4**
objectCategory	**CN=Person,CN=Schema,CN=Configuration,DC=test2,DC=local**
accountExpires	9223372036854775807
adminCount	1
badPasswordTime	129122151507923750
badPwdCount	0
codePage	0
countryCode	0
description	Built-in account for administering the computer/domain
displayName	Administrator
distinguishedName	CN=Administrator,CN=Users,DC=test2,DC=local
homeMDB	CN=Mailbox Database,CN=First Storage Group,CN=InformationStore,CN=W2003DOMAIN,CN=Server
homeMTA	CN=Microsoft MTA,CN=W2003DOMAIN,CN=Servers,CN=Exchange Administrative Group (FYDIBOHF2
isCriticalSystemObject	TRUE
lastLogoff	0
lastLogon	129144669282321250
legacyExchangeDN	/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=/
logonCount	133
mail	Administrator@test2.local
mailNickname	Administrator
mDBUseDefaults	TRUE
memberOf	CN=Administrators,CN=Builtin,DC=test2,DC=local
memberOf	CN=Domain Admins,CN=Users,DC=test2,DC=local
memberOf	CN=Enterprise Admins,CN=Users,DC=test2,DC=local

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})`.

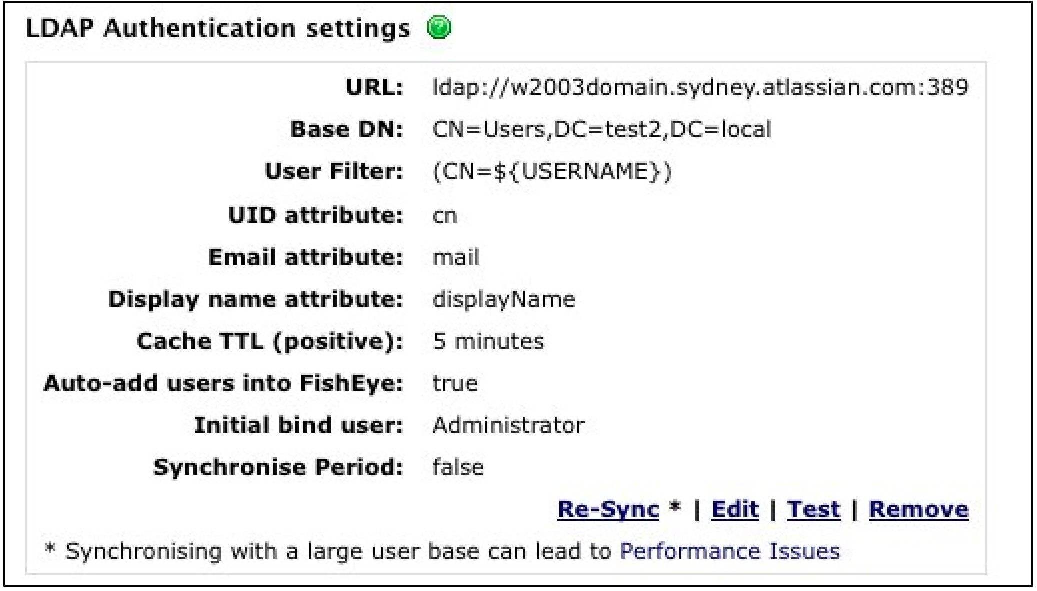
4. For the UID attribute, email attribute and the display name attribute, I can get that information from the right hand pane above. In my case

UID attribute is `cn`, the email attribute is `mail` and the display name attribute is `displayName`.

5. With all that information we are now ready to connect FishEye to AD. To summarise we have:

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

6. Placing that information into FishEye:



LDAP Authentication settings

URL:	ldap://w2003domain.sydney.atlassian.com:389
Base DN:	CN=Users,DC=test2,DC=local
User Filter:	(CN=${USERNAME})
UID attribute:	cn
Email attribute:	mail
Display name attribute:	displayName
Cache TTL (positive):	5 minutes
Auto-add users into FishEye:	true
Initial bind user:	Administrator
Synchronise Period:	false

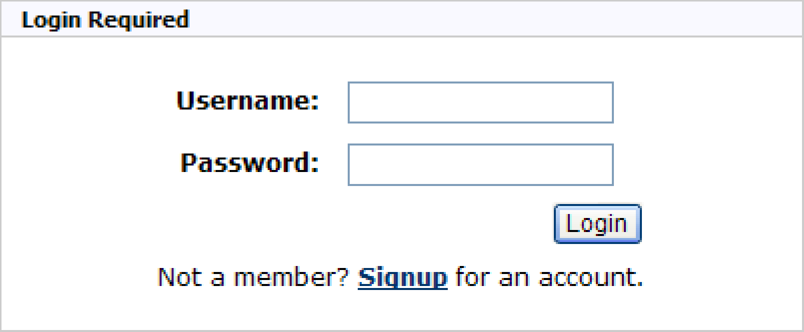
* Synchronising with a large user base can lead to Performance Issues

Useful References

- LDAP Authentication
- http://www.computerperformance.co.uk/Logon/LDAP_attributes_active_directory.htm

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:

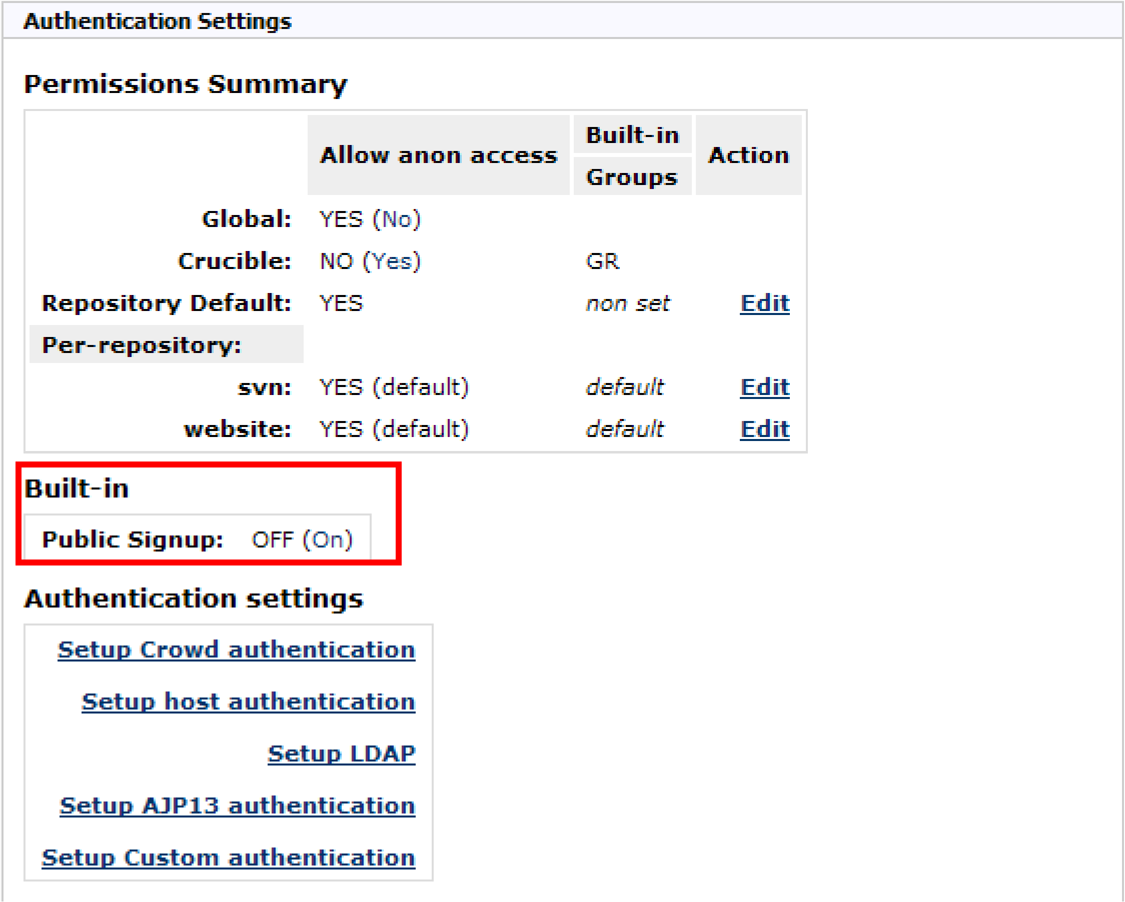


Public signup is disabled by default.

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'



Authentication Settings

Permissions Summary

	Allow anon access	Built-in Groups	Action
Global:	YES (No)		
Crucible:	NO (Yes)	GR	
Repository Default:	YES	*non set*	Edit
Per-repository:			
svn:	YES (default)	*default*	Edit
website:	YES (default)	*default*	Edit

Built-in

Public Signup: OFF (On)

Authentication settings

Setup Crowd authentication

Setup host authentication

Setup LDAP

Setup AJP13 authentication

Setup Custom authentication

Creating a Group

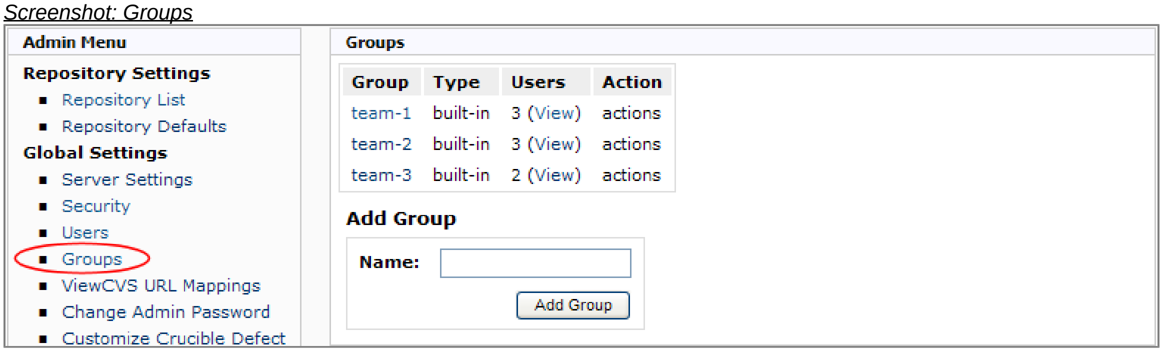
There are two types of FishEye user groups:

- '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



Group	Type	Users	Action
team-1	built-in	3 (View)	actions
team-2	built-in	3 (View)	actions
team-3	built-in	2 (View)	actions

Name:

Add Group

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.

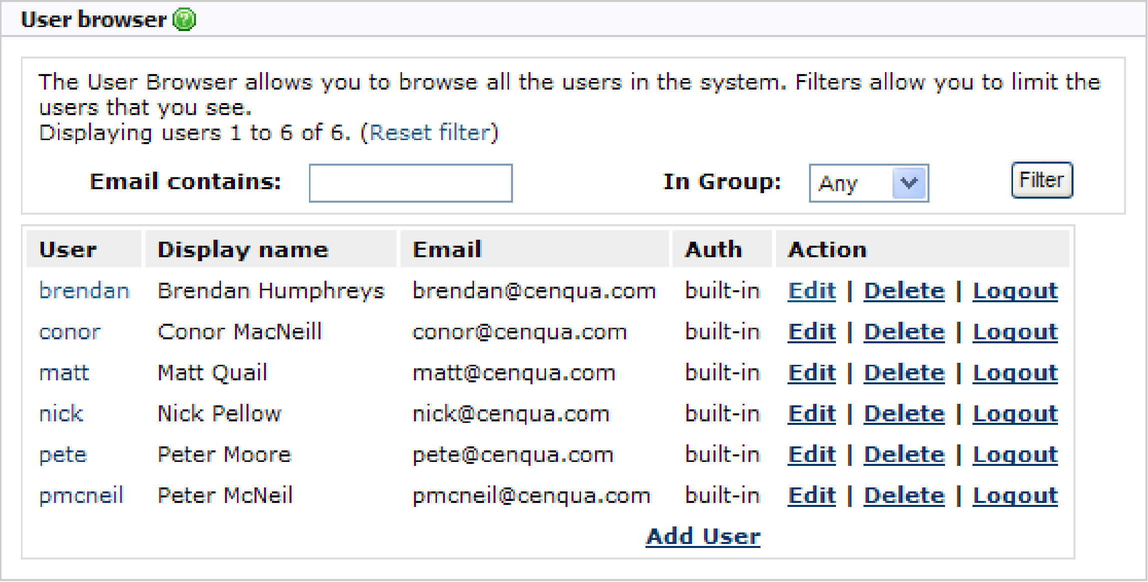
⚠ **Note re external directories:**

- 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.
   ℹ The user can easily change their own password later.
9. Click the '**Add**' button.

Screenshot: User Browser



User browser

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

Email contains: In Group: Any

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

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.

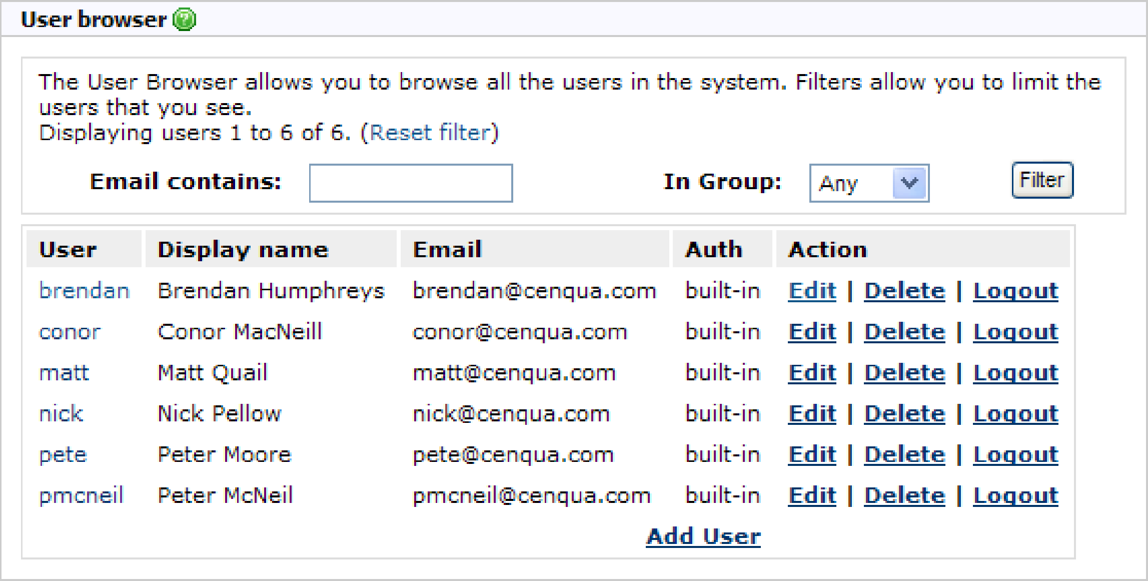
To deactivate a user's Crucible account,

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.
   ℹ If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click 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.
   ℹ If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click the '**Filter**' button.

Screenshot: Authentication Settings



User browser

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

Email contains: [ ] In Group: Any

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

Editing a User's Details

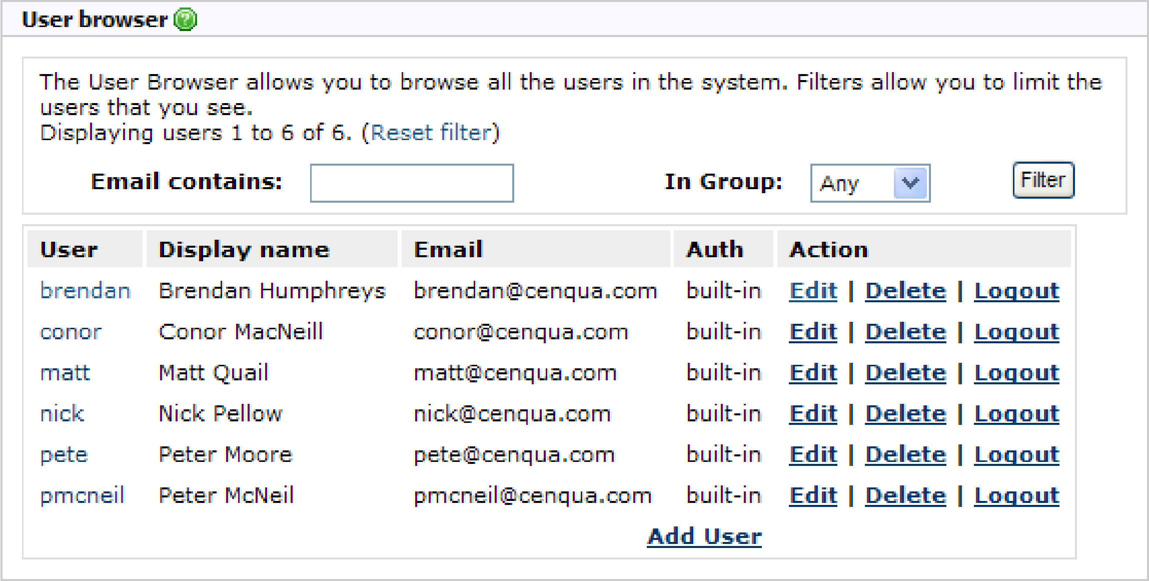
There are two types of user accounts:

- '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



User browser

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

Email contains: ______ **In Group:** Any

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

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.

ℹ 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.
   ℹ If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click 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.

ℹ 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.

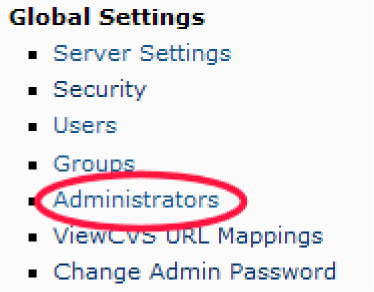
Granting Administrator Privileges

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:

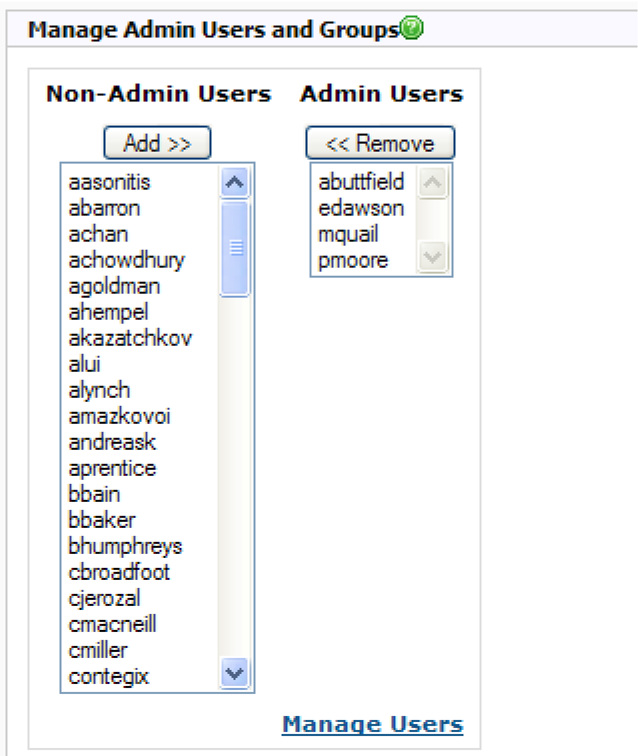
- Select the username in the '**Non-Admin Users**' column.
- Click the '**Add**' button.
- The username appears in the '**Admin Users**' column.

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:

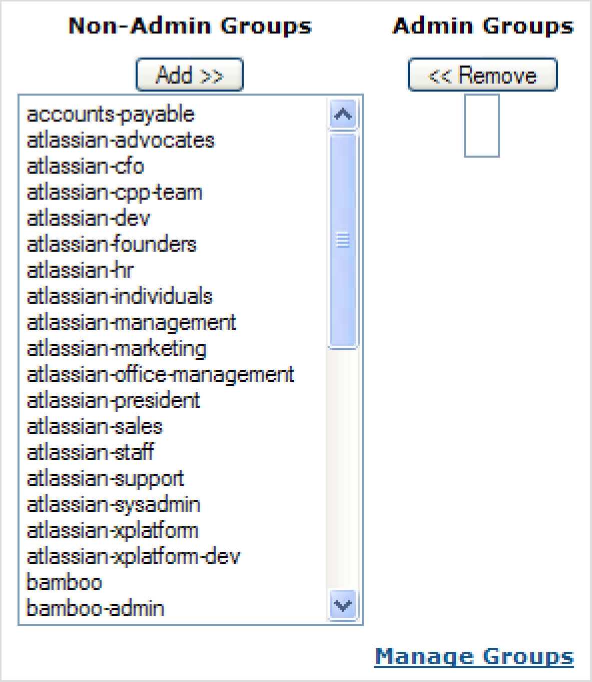
- Select the group name in the '**Non-Admin Groups**' column.
- Click the '**Add**' button.
- The group name appears in the '**Admin Groups**' column.

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



accounts-payable
atlassian-advocates
atlassian-cfo
atlassian-cpp-team
atlassian-dev
atlassian-founders
atlassian-hr
atlassian-individuals
atlassian-management
atlassian-marketing
atlassian-office-management
atlassian-president
atlassian-sales
atlassian-staff
atlassian-support
atlassian-sysadmin
atlassian-xplatform
atlassian-xplatform-dev
bamboo
bamboo-admin

Load all users from Active Directory, LDAP or Atlassian Crowd

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

1. Configuring your external source

You must firstly set up FishEye/Crucible to connect to your external user repository, either LDAP-based (including Active Directory) or Atlassian Crowd. Follow the steps in the documentation pages listed below and then return to Step 2 on this page.

- LDAP Authentication
- Crowd Authentication

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.

2. Loading external users

To load users from Atlassian Crowd;

1. From the '**Admin Menu**', click '**Global Settings**' then '**Security**'.
2. Under Security, find the option '**Synchronise users with Crowd**'. Select the '**Yes**' option by clicking the radio button. Click '**Apply**' to complete the process.
3. As soon as an option has been selected and '**Apply**' has been clicked, users are immediately added to the user list.

Note that 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.

3. Setting the synchronise period

The '**synchronise period**' option allows you to set the time interval when Fisheye or Crucible will synchronise with the LDAP directory. You can use intuitive settings such as '**1m**' for one minute, '**1h**' for one hour, and so on. Simply enter the time interval and click '**Apply**'.

Synchronising Large User Repositories
If you have a rather 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.
```

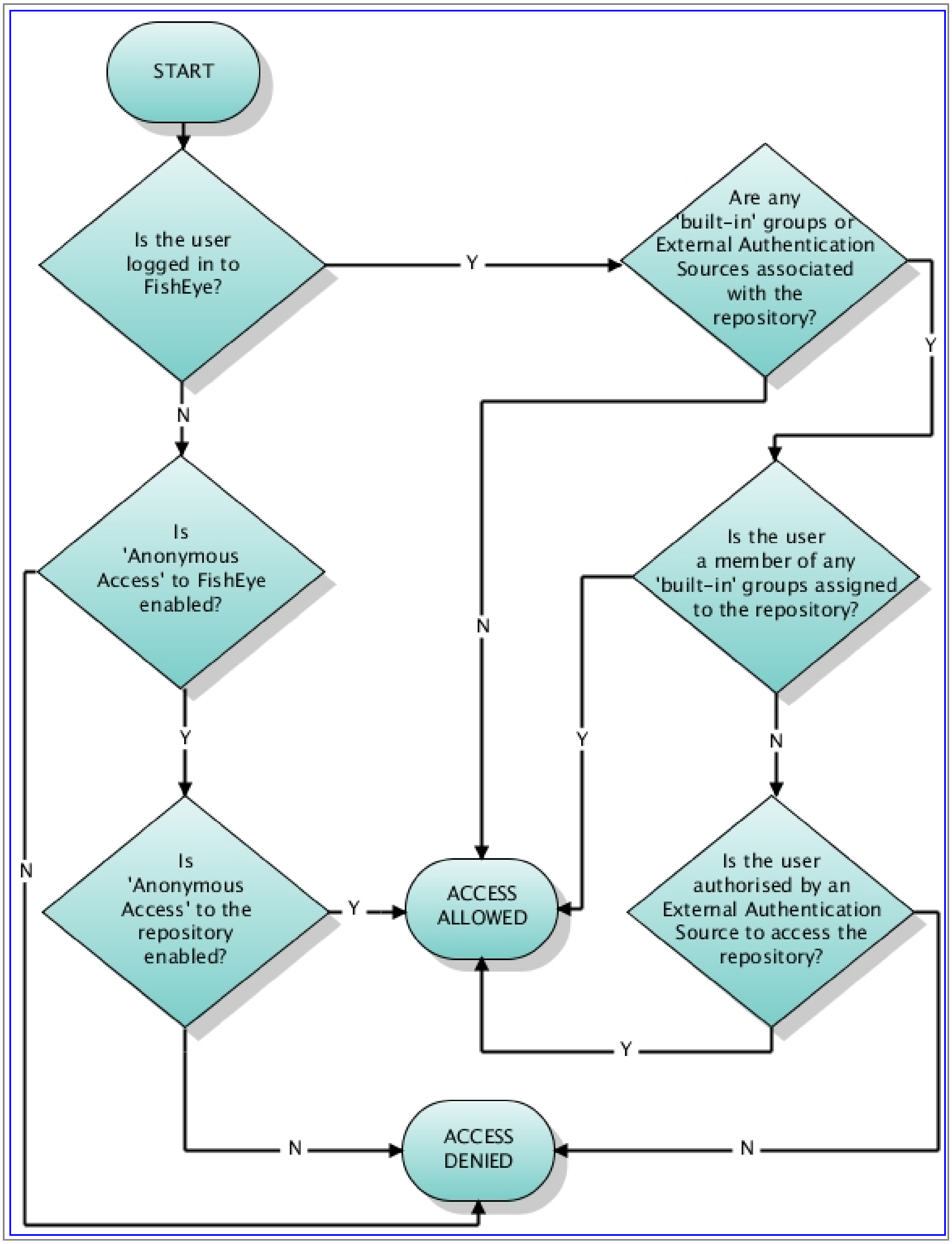
Additionally, when setting up external authentication for large repositories: set "Synchronise users with LDAP" to no. Next turn on "Auto-add" users to on, so that any External user that matches your defined user filter who logs into FishEye, will be added as a user (NOTE: Auto-add does not mean any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


START
Is the user logged in to FishEye?
Y
N
Are any 'built-in' groups or External Authentication Sources associated with the repository?
Y
N
Is 'Anonymous Access' to FishEye enabled?
Y
N
Is the user a member of any 'built-in' groups assigned to the repository?
Y
N
Is 'Anonymous Access' to the repository enabled?
Y
N
ACCESS ALLOWED
Is the user authorised by an External Authentication Source to access the repository?
Y
N
ACCESS DENIED

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. Trusted applications is a new feature in Crucible 1.2.2 and FishEye 1.4.2. At this time, JIRA 3.12 and Confluence 2.7 can be configured as trusted applications.

Note: FishEye and Crucible share the same trusted applications — an application trusted by FishEye is also trusted by Crucible.

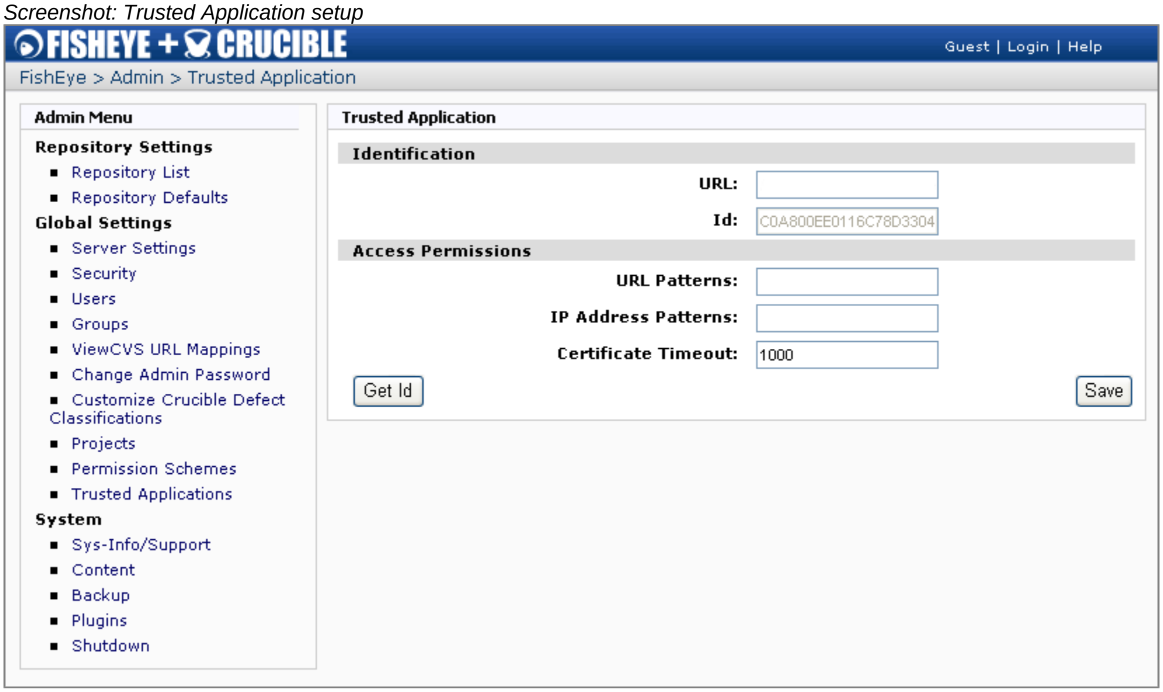
> Before you begin, note that configuring a trusted application requires the transmission of sensitive data. To prevent 'man-in-the-middle attacks', it is recommended that you use an encrypted SSL connection while configuring a trusted application.

Adding a Trusted Application

To add a trusted application to FishEye:

1. Access the '**Administration Screen**'.
2. Click '**Trusted Applications**' under '**Global Settings**' on the left navigation bar.
3. Click '**Add a Trusted Application**'. The '**Trusted Application**' screen opens.

Screenshot: Trusted Application setup



URL:

Id: C0A800EE0116C78D3304

URL Patterns:

IP Address Patterns:

Certificate Timeout: 1000

On this page, there are two areas, the '**Identification**' area and the '**Access Permissions**' area.

Configuring Identification Settings

Under the '**Identification**' heading, there are two fields, '**URL**' and '**Id**'.

URL field

In this field is where you will enter the Trusted Application Public Key URL of the application you wish to trust. For example, if your application's base URL is;

```
'http://www.mycompany/jira/'
```

you would enter that into the URL field. Once you've done this, click the Get Id button. FishEye will then retrieve the Trust Certificate Id from the other application and display it in the Id field. If this step fails, you may not have not entered the correct URL for the other application.

Id field

This field contains the **Trust Certificate ID**, once you have filled out the URL field correctly (see above) and clicked the Get Id button. The contents of this field are not editable.

(Note: The application you are trusting must support Trusted Applications also. JIRA 3.12 and Confluence 2.7 support this.)

Configuring Access Permissions

Under the **Access Permissions** heading, there are three fields, **URL Patterns**, **IP Address Patterns** and **Certificate Timeout**. These allow you to further restrict requests from a trusted application.

URL Patterns field

With this field, you can limit the access a trusted application has to FishEye. It it not necessary to specify anything for this field; in fact a blank value is a sensible default. The default behaviour is no restriction.

The text that you specify should not include your hostname, IP address or port number, rather it relates to folders on the server, that start with the text you provide.

For example, if you use this setting:

```
/foo
```

then FishEye will trust only the requests to FishEye URLs starting with `/foo`, e.g. `/foo/bar`, `/foobar` and `/foo/bar/baz/x`. You can specify multiple URLs by separating them with a comma.

> URL Patterns do not support wildcard characters or regular expressions in FishEye.

IP Address Patterns field

With this field, you can limit the trusted network addresses for other applications. You can use wildcards to specify a number range, and multiple addresses can be separated with commas. For example, if you use this setting:

```
192.168.*.*,127.0.0.0
```

then FishEye will only trust requests from machines with the IP addresses 192.168.*anything*.*anything*
(a group of network addresses) and 127.0.0.0 (a single host). The default is no restriction.

Certificate Timeout field

With this field, you can set the number of milliseconds before the certificate times out. This feature's purpose is to prevent 'replay attacks'. For example, if an attacker intercepts a request, they may attempt to extract the certificate and send it again independently. With the certificate timeout, the application will be able to tell that this is no longer a valid request. The default value is 1000 (one second).

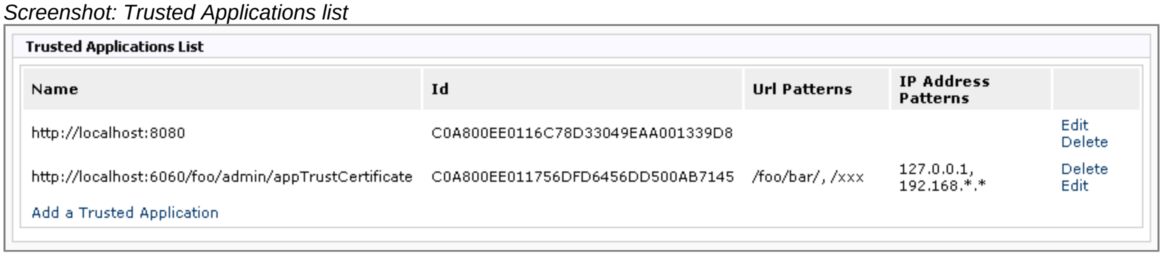
> A shorter time out is more secure, but if set too short, it may cause valid requests to be rejected on slower networks.

Once you've finished entering the settings for the Trusted Application, click Save to confirm and activate the trust relationship.

Editing Trusted Application Settings

Once you have configured your trusted application(s), you can view the settings on the main **'Trusted Applications'** page.

Screenshot: Trusted Applications list



Trusted Applications List

Name	Id	Url Patterns	IP Address Patterns	
http://localhost:8080	C0A800EE0116C78D33049EAA001339D8			Edit Delete
http://localhost:6060/foo/admin/appTrustCertificate	C0A800EE011756DFD6456DD500AB7145	/foo/bar/, /xxx	127.0.0.1, 192.168.*.*	Delete Edit
Add a Trusted Application				

From this screen, you can click **'Edit'** to make changes to the trusted application settings, or click **'Delete'** to remove the trust relationship for that application.

Advanced Administration Options

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. ⚠ 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.

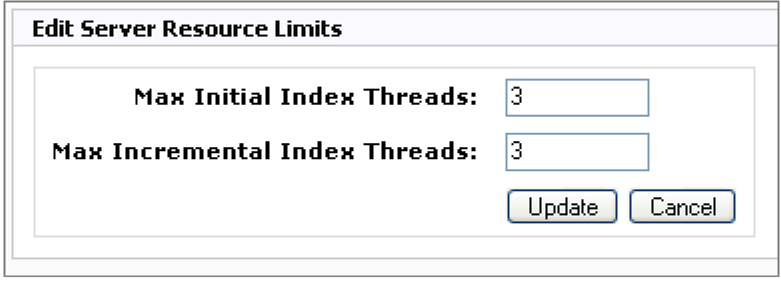
The number of threads available for these indexing operations is now configurable.

ℹ 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



Edit Server Resource Limits

Max Initial Index Threads: 3

Max Incremental Index Threads: 3

!

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.

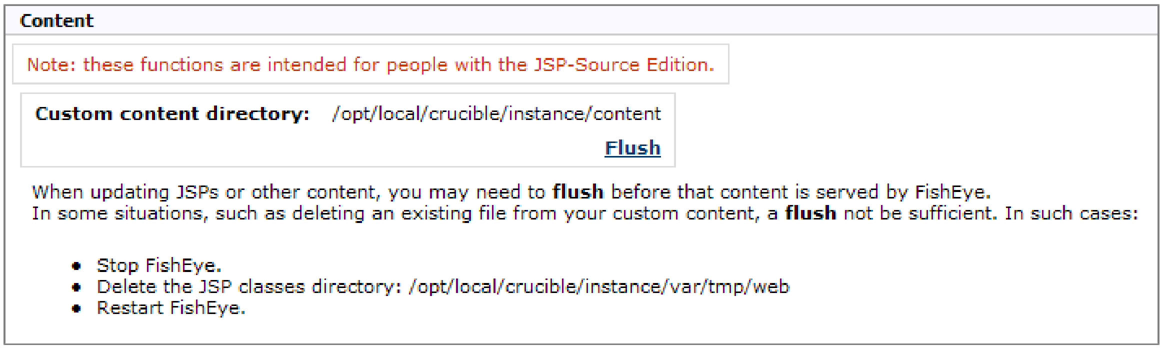
If you have a commercial license assigned to your account, you will see a '**source download**' option on your download page.

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



Content

Note: these functions are intended for people with the JSP-Source Edition.

Custom content directory: /opt/local/crucible/instance/content

When updating JSPs or other content, you may need to **flush** before that content is served by FishEye.
In some situations, such as deleting an existing file from your custom content, a **flush** not be sufficient. In such cases:

- Stop FishEye.
- Delete the JSP classes directory: /opt/local/crucible/instance/var/tmp/web
- Restart FishEye.

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.

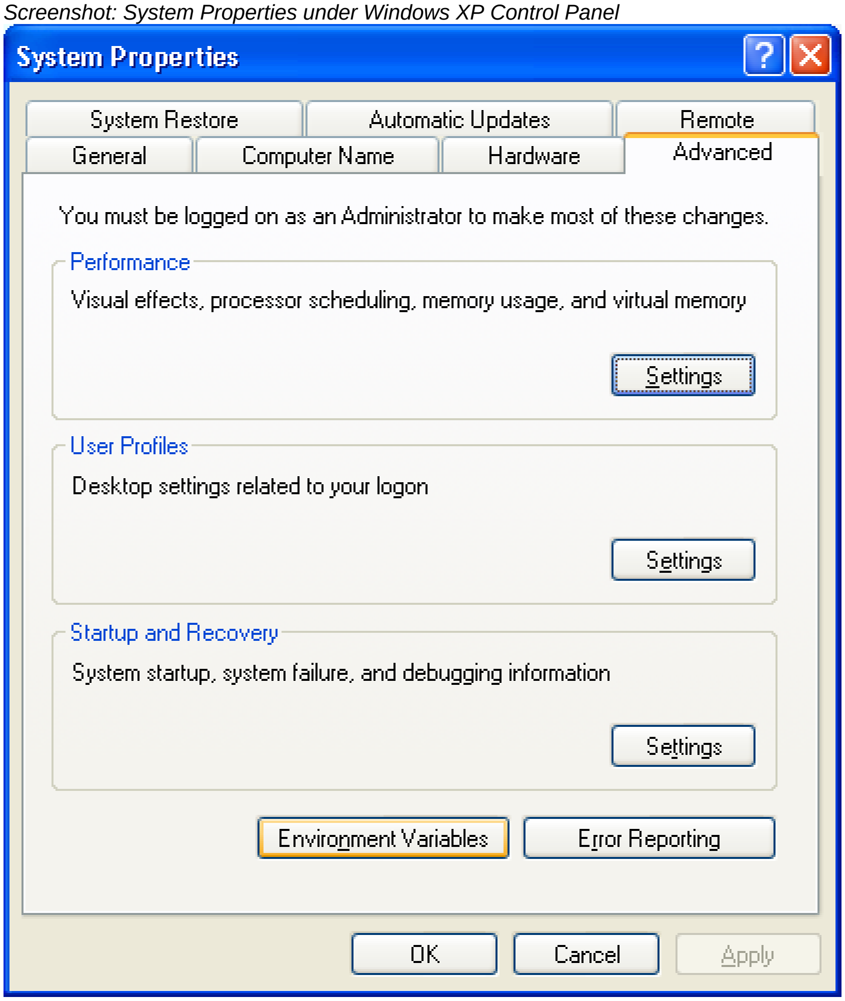
Read more about using FISHEYE_INST in the Installation Guide.

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



You must be logged on as an Administrator to make most of these changes.

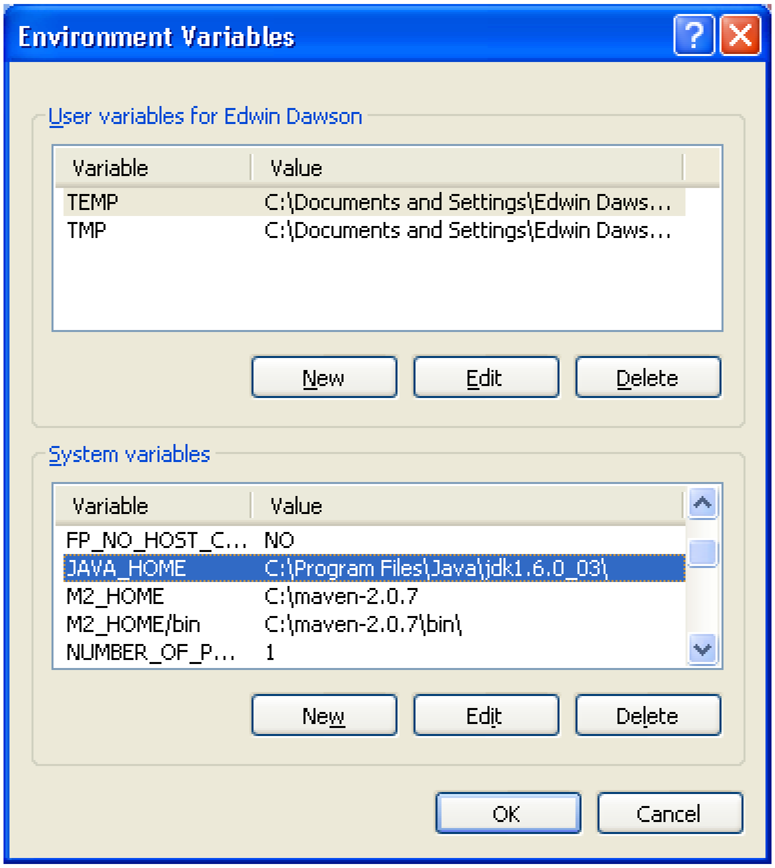
Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your logon

System startup, system failure, and debugging information

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

Screenshot: Environment Variables under Windows XP Control Panel



User variables for Edwin Dawson

Variable	Value
TEMP	C:\Documents and Settings\Edwin Daws...
TMP	C:\Documents and Settings\Edwin Daws...

System variables

Variable	Value
FP_NO_HOST_C...	NO
JAVA_HOME	C:\Program Files\Java\jdk1.6.0_03\
M2_HOME	C:\maven-2.0.7
M2_HOME/bin	C:\maven-2.0.7\bin\
NUMBER_OF_P...	1

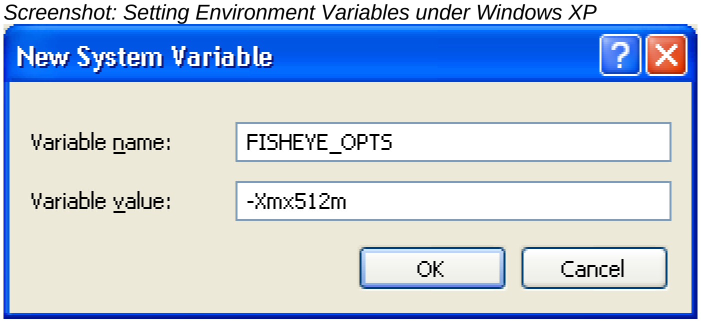
4. Click 'New'.
5. In the 'Variable name' field, enter the name of the environment variable, for example

```
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



Variable name: FISHEYE_OPTS

Variable value: -Xmx512m

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

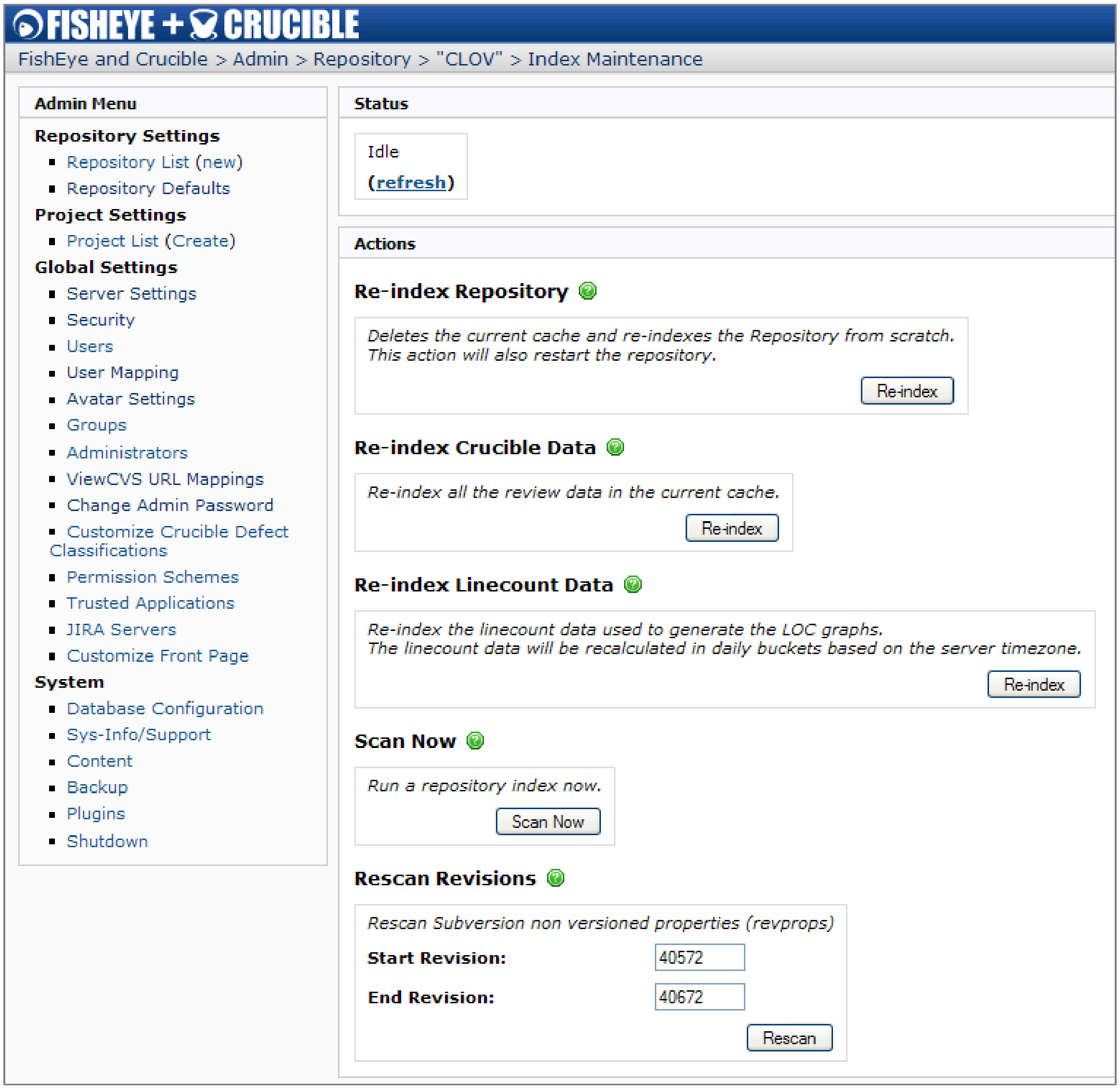
Settings and administrative operations for repository indexing are located on the Indexer page.

To get to this page, go to '**Administration**' > '**Repository List**' > '**View**' (next to your repository name) > '**Indexer**' > '**Maintenance**'

The '**Indexer Maintenance**' screen opens, where an administrator can manually trigger the following actions:

Action	Description
Refresh	Refreshes the status string, in the Status box. This will show the name of any repository task that FishEye is currently carrying out (if any). This may be helpful to monitor the progress of an initial indexing task or to diagnose problems (such as your repository server being out of action).
Re-index Repository	Delete the current cache and re-index the repository from the beginning. This action will also restart the repository.
Re-index Crucible Data	(applies only when using Crucible with FishEye) Re-index all the Crucible review data associated with the current repository.
Re-index Linecount Data	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.
Scan Now	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.
Rescan Revisions	For a **Subversion** repository, you can 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. For a **Perforce**, repository, you can rescan changelists. This will rescan changelist metadata (author, date, commit message) and any job fixes associated with changelists in the the given changelist range.

Screenshot: FishEye Index Maintenance menu



Re-index Repository

Deletes the current cache and re-indexes the Repository from scratch. This action will also restart the repository.

Re-index Crucible Data

Re-index all the review data in the current cache.

Re-index Linecount Data

Re-index the linecount data used to generate the LOC graphs. The linecount data will be recalculated in daily buckets based on the server timezone.

Scan Now

Run a repository index now.

Rescan Revisions

Rescan Subversion non versioned properties (revprops)

Start Revision: 40572

End Revision: 40672

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.

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 FishEye plugin is a program that provides an extra piece of FishEye functionality.

You can create your own plugins (see the FishEye Development Hub), then install them into your FishEye system as follows:

To add a plugin,

1. Copy the `JAR` file to the `/FISHEYE_INST/var/plugins/user` directory. You can see the absolute path name of this directory under the plugins list.
2. Access the '**Admin Menu**' and click '**Plugins**'.
3. The '**Plugins**' screen will be displayed (see screenshot below). Click the '**Check for new plugins in...**' link.
4. Your newly added plugin will now appear in the list on the '**Plugins**' screen.

To enable a plugin,

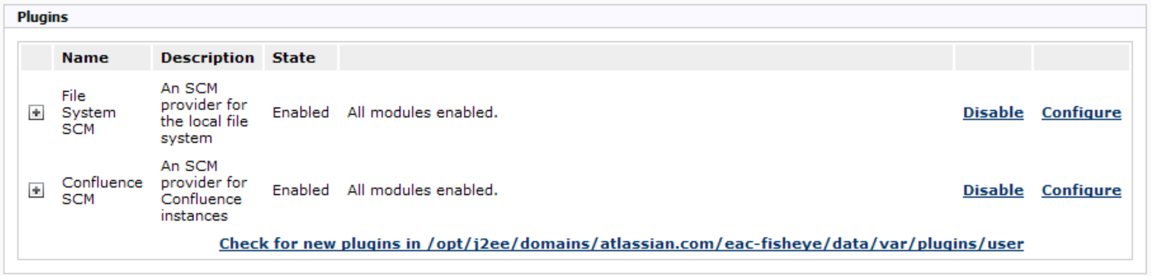
1. Click '**Plugins**' on the '**Admin Menu**'.
2. The '**Plugins**' screen will be displayed (see screenshot below).
3. Locate the plugin you wish to enable, and click the '**Enable**' link.
    - To view, enable or disable individual modules within a plugin, click the '**+**' sign at the left of the plugin's name.

To configure a plugin,

1. Click '**Plugins**' on the '**Admin Menu**'.
2. The '**Plugins**' screen will be displayed (see screenshot below).
3. Locate the plugin you wish to enable, and click the '**Configure**' link. This will show the configuration page for that plugin. If the plugin does not require configuration there will not be a '**Configure**' link.
4. Read the documentation for that particular plugin to understand the process of configuring the plugin.

Some plugins require further configuration after they are enabled.

Screenshot: Plugin Management Screen



Plugins

	Name	Description	State			
⊞	File System SCM	An SCM provider for the local file system	Enabled	All modules enabled.	Disable	Configure
⊞	Confluence SCM	An SCM provider for Confluence instances	Enabled	All modules enabled.	Disable	Configure

Check for new plugins in /opt/j2ee/domains/atlassian.com/eac-fisheye/data/var/plugins/user

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.

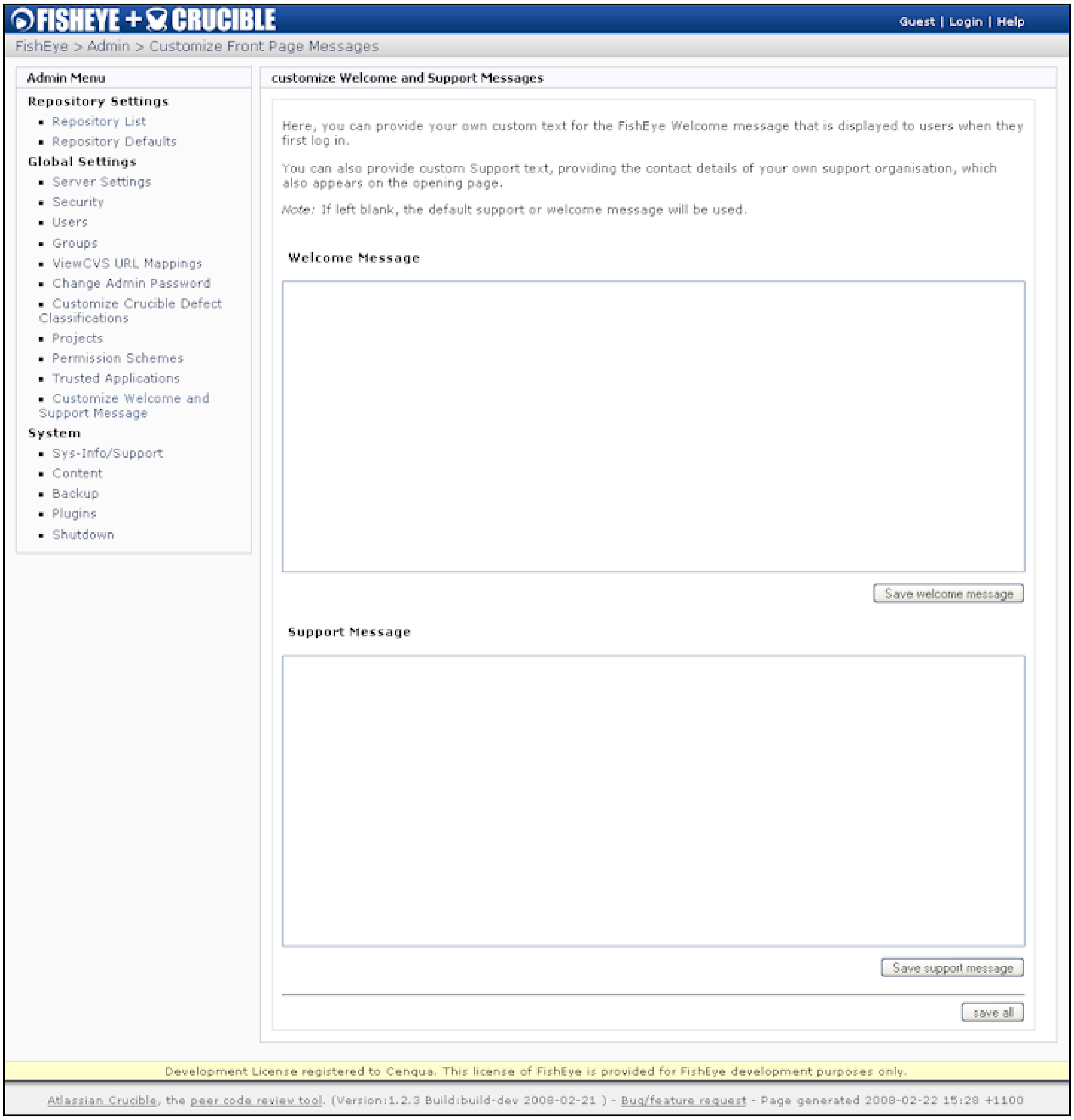
To customise the welcome message, access the administration page, and click '**Customize Welcome & Support Message**' under '**Global Settings**' on the left navigation bar.

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



customize Welcome and Support Messages

Here, you can provide your own custom text for the FishEye Welcome message that is displayed to users when they first log in.

You can also provide custom Support text, providing the contact details of your own support organisation, which also appears on the opening page.

Note: If left blank, the default support or welcome message will be used.

Development License registered to Cenqua. This license of FishEye is provided for FishEye development purposes only.

Atlassian Crucible, the peer code review tool. (Version:1.2.3 Build:build-dev 2008-02-21) - Bug/feature request - Page generated 2008-02-22 15:28 +1100

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

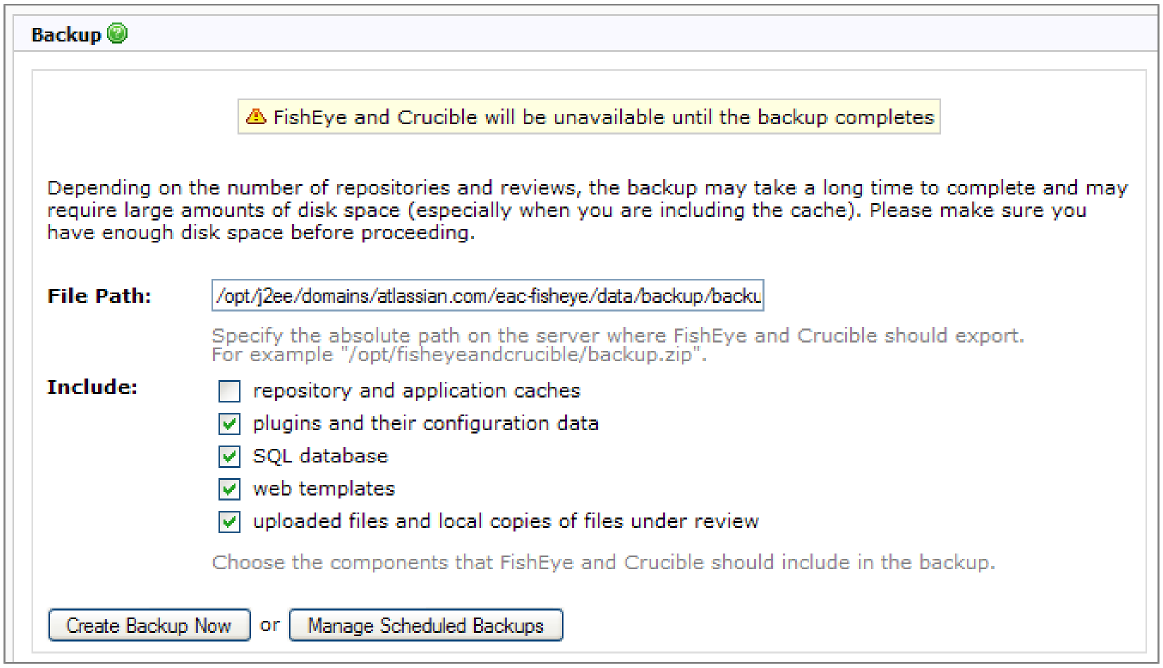
The FishEye Admin Interface Backup Process

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.
  ℹ 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



⚠ FishEye and Crucible will be unavailable until the backup completes

Depending on the number of repositories and reviews, the backup may take a long time to complete and may require large amounts of disk space (especially when you are including the cache). Please make sure you have enough disk space before proceeding.

File Path: /opt/j2ee/domains/atlassian.com/eac-fisheye/data/backup/backu

Specify the absolute path on the server where FishEye and Crucible should export.
For example "/opt/fisheyeandcrucible/backup.zip".

Include:

- ☐ repository and application caches
- ☑ plugins and their configuration data
- ☑ SQL database
- ☑ web templates
- ☑ uploaded files and local copies of files under review

Choose the components that FishEye and Crucible should include in the backup.

The FishEye Command Line Backup Process

ℹ 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.
   ℹ Note that if your FishEye instance uses a custom `FISHEYE_INST` directory, make sure the environment variable is properly set when running the backup command.

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. ℹ 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. ℹ 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

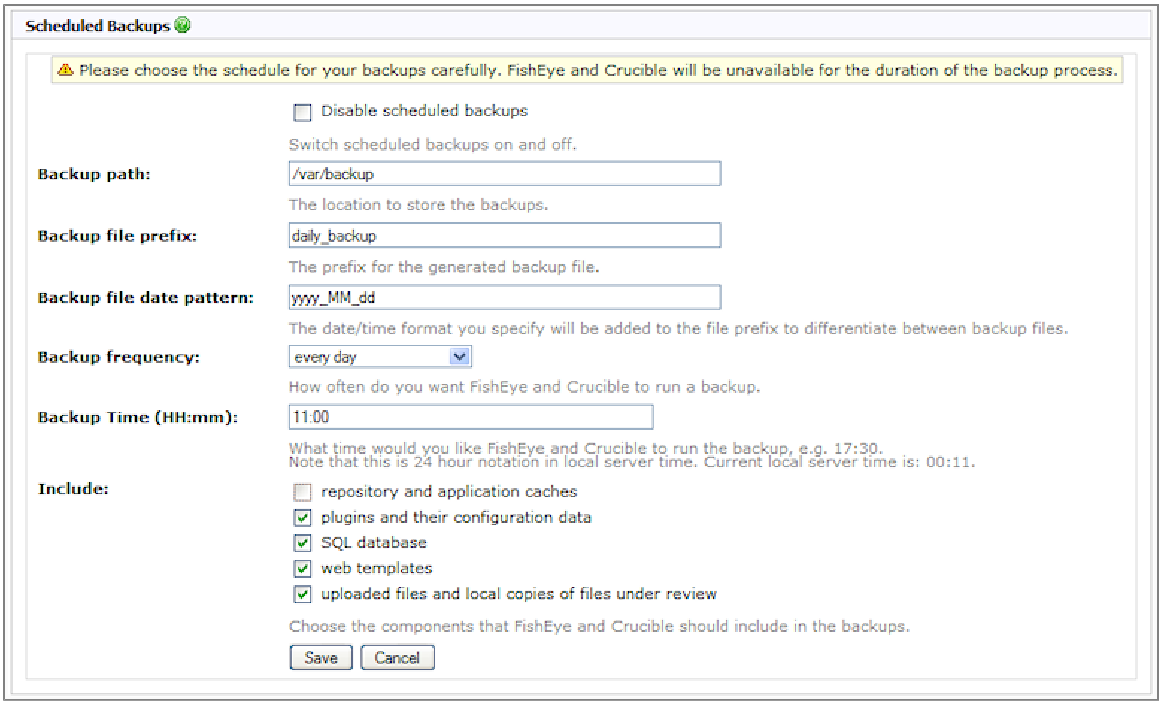
To set a schedule for automatic backups, open the administration screen and click '**Backup**' under '**System**' on the left navigation bar. The 'Backup' page opens. Now, click the link '**Manage Scheduled Backups**' at the bottom of the page. The 'Scheduled Backups' page opens.

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



Scheduled Backups

⚠ Please choose the schedule for your backups carefully. FishEye and Crucible will be unavailable for the duration of the backup process.

□ Disable scheduled backups

Switch scheduled backups on and off.

Backup path: /var/backup

The location to store the backups.

Backup file prefix: daily_backup

The prefix for the generated backup file.

Backup file date pattern: yyyy_MM_dd

The date/time format you specify will be added to the file prefix to differentiate between backup files.

Backup frequency: every day

How often do you want FishEye and Crucible to run a backup.

Backup Time (HH:mm): 11:00

What time would you like FishEye and Crucible to run the backup, e.g. 17:30.
Note that this is 24 hour notation in local server time. Current local server time is: 00:11.

Include:

□ repository and application caches
☑ plugins and their configuration data
☑ SQL database
☑ web templates
☑ uploaded files and local copies of files under review

Choose the components that FishEye and Crucible should include in the backups.

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

Restoring FishEye Data

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' 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.

FishEye Installation and Upgrade Guide

- FishEye Upgrade Guide
- FishEye Installation Guide

FishEye Upgrade Guide

The first time you run a new version of FishEye, it will automatically upgrade its data. This may involve a complete re-index of your repository.

- Before you start
- Upgrade Procedure
  - Method 1: Using a Separate FISHEYE_INST Directory
  - Method 2: No Separate FISHEYE_INST Directory
  - Method 3 - Without a FISHEYE_INST Directory, but would like to set one up
- FishEye 2.0 Upgrade Notes
- FishEye 1.6 Upgrade Notes
- FishEye 1.5 Upgrade Notes
- FishEye 1.4 Upgrade Notes
- FishEye 1.3 Upgrade Notes

Before you start

- Before upgrading you should always read the Release Notes and Changelog 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.

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` and `/NEW_FISHEYE/cache` directories.
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. Copy (or move) the `/OLD_FISHEYE/cache` directory to `/NEW_FISHEYE/cache`.
7. If you have a Cenqua-issued FishEye license, copy your `fisheye.license` to `/NEW_FISHEYE/`. (Atlassian-issued licenses are included within `config.xml`.)
8. Start FishEye from the new installation.
9. 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. Copy the `/OLD_FISHEYE/cache` directory to `/FISHEYE_INST`.
6. Extract the new Fisheye archive into a directory such as `/NEW_FISHEYE/`.
7. Start Fisheye from the new installation by running `NEW_FISHEYE/bin/run.sh`. (Use `run.bat` on Windows).
8. 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 Upgrade 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.
- Supported browsers are: Safari 3+, FireFox 3+ and Internet Explorer 7+ (not IE6).

FishEye 1.6 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.

FishEye 1.5 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**.

FishEye 1.4 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**.

FishEye 1.3 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**.

FishEye Installation Guide

This guide describes the advanced installation options that can be used when installing Fisheye.

ℹ Evaluating FishEye for the first time? See the FishEye 101 page.

- Supported Platforms
- Installing FishEye
- Configuring FishEye
- Best Practices for FishEye Configuration

Supported Platforms

This page shows the supported platforms for **FishEye 2.3.x** and its minor releases.

Key: ✓ = Supported; ✗ = Not Supported

Java Version	
JRE / JDK [1]	✓ 1.5 or later
Operating Systems	
Microsoft Windows [2]	✓
Linux [2]	✓
Apple Mac OS X [2]	✓
Databases	
MySQL	✓ MySQL Enterprise Server 5.0.21 or later ✓ MySQL Community Server 5.0.21 or later
PostgreSQL	✓ 8.0 or later
HSQLDB [3]	✓ (bundled; default)
Web Browsers	
Microsoft Internet Explorer	✓ 7.0 or later, ✗ IE6 is not supported
Mozilla Firefox	✓ 3 or later
Safari	✓ 4 or later
Version Control Systems	
Subversion	✓ Server 1.1 or later, client uses SVNkit or native JavaHL
CVS (and CVSNT)	✓ All versions
Perforce	✓ Client version 2007.3 or later
Git	✓ 1.6 or later
IBM ClearCase	✓ 2003.06.10 or later

Mercurial (alpha support)	✓ 1.5.1 or later (Python 2.4.3 or later)

Supported Platform Notes

1. FishEye requires **Java Runtime** (JDK or JRE) version **1.5** or later (Solaris requires **1.5.0_15** as a minimum).

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

ⓘ Once you have installed the JDK, you need to set the JAVA_HOME environment variable.

⚠ We strongly recommend the use of a 32-bit JDK/JRE rather than a 64-bit JDK/JRE. 64-bit JDK/JREs will consume the available RAM more rapidly, and this may result in poor performance.

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.

At this time, FishEye supports the following external databases:

- MySQL Enterprise Server 5.0.21 onwards and MySQL Community Server 5.0.21 onwards (see the FishEye Database documentation).
- PostGreSQL 8.x onwards (see the FishEye Database documentation).

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 SNVkit 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.

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).

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.
   This document assumes you have extracted FishEye to `/FISHEYE_HOME/`.
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. 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.
5. 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.

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).

Next Step - Initial Configuration

See the guidelines on configuring FishEye.

Configuring FishEye

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`.

Running FishEye for the First Time

To run FishEye for the first time, simply do the following:

- 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.

> ℹ If you want to run FishEye as a Windows service, please refer to this document

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 here.

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.

Accessing the Administration Pages

Once you have set up an administrator password (described above), 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.

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

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

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

FishEye Release Notes

⚠ FishEye 2.3 has now been released. Read the Release Notes.

FishEye Release Notes and Changelogs

- Security Advisories
  - FishEye Security Advisory 2010-06-16
  - FishEye Security Advisory 2010-05-04
- FishEye Release Summary
- FishEye 2.3 Release Notes
  - FishEye 2.3 Changelog
- FishEye 2.2 Release Notes
  - FishEye 2.2 Changelog
- FishEye 2.1 Release Notes
  - FishEye 2.1 Changelog
- FishEye 2.0 Release Notes
  - FishEye 2.0 Upgrade Notes
  - FishEye 2.0 Changelog
- FishEye 2.0 Beta Release Notes
  - Upgrading to the FishEye 2.0 Beta
  - JIRA Integration in FishEye 2.0 Beta
  - Git Alpha in FishEye 2.0 Beta
  - FishEye 2.0 Beta Reviewer's Guide
- FishEye 1.6 Release Notes
  - FishEye 1.6 Changelog
- FishEye 1.5 Release Notes
  - FishEye 1.5 Changelog
- FishEye 1.4 Release Notes
  - FishEye 1.4 Changelog
- FishEye 1.3 Release Notes
  - FishEye 1.3 Changelog

- For changes prior to 1.3, see:
  - 1.2.x Changelog
  - 1.1.x Changelog
  - 1.0.x Changelog

Installation

You can now download FishEye from here. Information on installing FishEye can be found here.

If upgrading from a previous version, please follow the Upgrade Guide.

FishEye 2.3 Release Notes

26 May 2010

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.

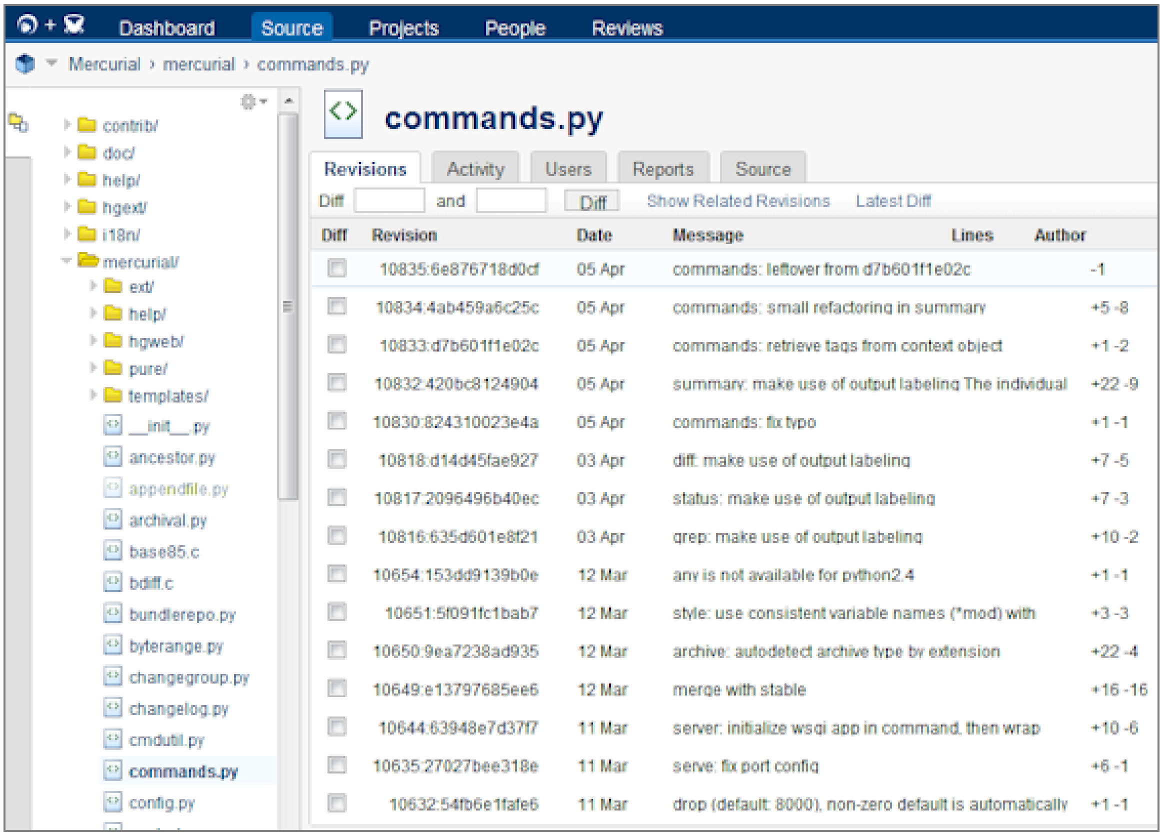
Highlights of FishEye 2.3

1

Mercurial SCM Alpha

FishEye 2.3 adds alpha support for Mercurial repositories. Atlassian is providing early access to this functionality for our customers. There are still a few kinks to be worked out, but it provides full access to FishEye.

Screenshot: A Mercurial Repository in Action



Diff	Revision	Date	Message	Lines	Author
□	10835:6e876718d0cf	05 Apr	commands: leftover from d7b601f1e02c	-1	
□	10834:4ab459a6c25c	05 Apr	commands: small refactoring in summary	+5 -8	
□	10833:d7b601f1e02c	05 Apr	commands: retrieve tags from context object	+1 -2	
□	10832:420bc8124904	05 Apr	summary: make use of output labeling The individual	+22 -9	
□	10830:824310023e4a	05 Apr	commands: fix typo	+1 -1	
□	10818:d14d45fae927	03 Apr	diff: make use of output labeling	+7 -5	
□	10817:2096496b40ec	03 Apr	status: make use of output labeling	+7 -3	
□	10816:635d601e8f21	03 Apr	grep: make use of output labeling	+10 -2	
□	10654:153dd9139b0e	12 Mar	any is not available for python2.4	+1 -1	
□	10651:5f091fc1bab7	12 Mar	style: use consistent variable names (*mod) with	+3 -3	
□	10650:9ea7238ad935	12 Mar	archive: autodetect archive type by extension	+22 -4	
□	10649:e13797685ee6	12 Mar	merge with stable	+16 -16	
□	10644:63948e7d37f7	11 Mar	server: initialize wsgi app in command, then wrap	+10 -6	
□	10635:27027bee318e	11 Mar	serve: fix port config	+6 -1	
□	10632:54fb6e1fafe6	11 Mar	drop (default: 8000), non-zero default is automatically	+1 -1	

See the documentation for more details.

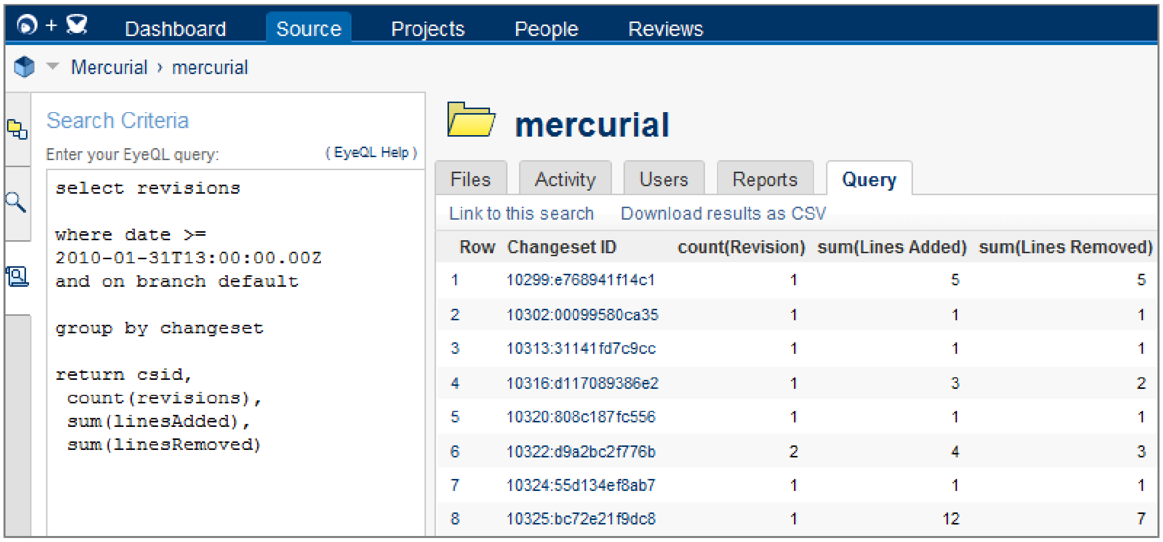
2

New 'Aggregate' functions in EyeQL query language

EyeQL, FishEye's sophisticated query language has become much more powerful with the addition of new aggregating functions. These allow you to quickly gather statistics such as counts, maximums, minimums and averages across the following:

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

Screenshot: New EyeQL Functions

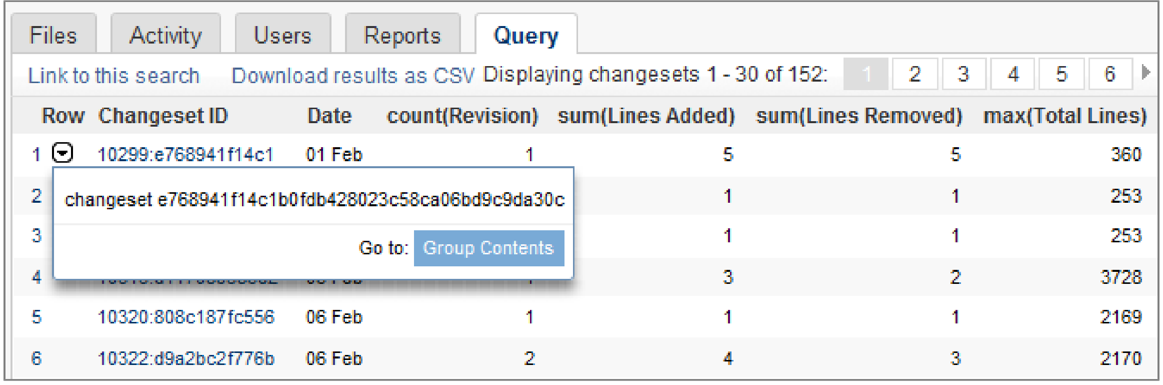


```
select revisions

where date >=
2010-01-31T13:00:00.00Z
and on branch default

group by changeset

return csid,
 count(revisions),
 sum(linesAdded),
 sum(linesRemoved)
```

Row	Changeset ID	count(Revision)	sum(Lines Added)	sum(Lines Removed)
1	10299:e768941f14c1	1	5	5
2	10302:00099580ca35	1	1	1
3	10313:31141fd7c9cc	1	1	1
4	10316:d117089386e2	1	3	2
5	10320:808c187fc556	1	1	1
6	10322:d9a2bc2f776b	2	4	3
7	10324:55d134ef8ab7	1	1	1
8	10325:bc72e21f9dc8	1	12	7

These data points can be calculated across your whole query result, or grouped by changeset, file or directory. When you're browsing the results, you can click in the results area to see the relevant group contents on an item.

Screenshot: FishEye Group Contents Dialog



Row	Changeset ID	Date	count(Revision)	sum(Lines Added)	sum(Lines Removed)	max(Total Lines)
1	10299:e768941f14c1	01 Feb	1	5	5	360
2				1	1	253
3				1	1	253
4	10316:d117089386e2	06 Feb	1	3	2	3728
5	10320:808c187fc556	06 Feb	1	1	1	2169
6	10322:d9a2bc2f776b	06 Feb	2	4	3	2170

changeset e768941f14c1b0fdb428023c58ca06bd9c9da30c

Go to: Group Contents

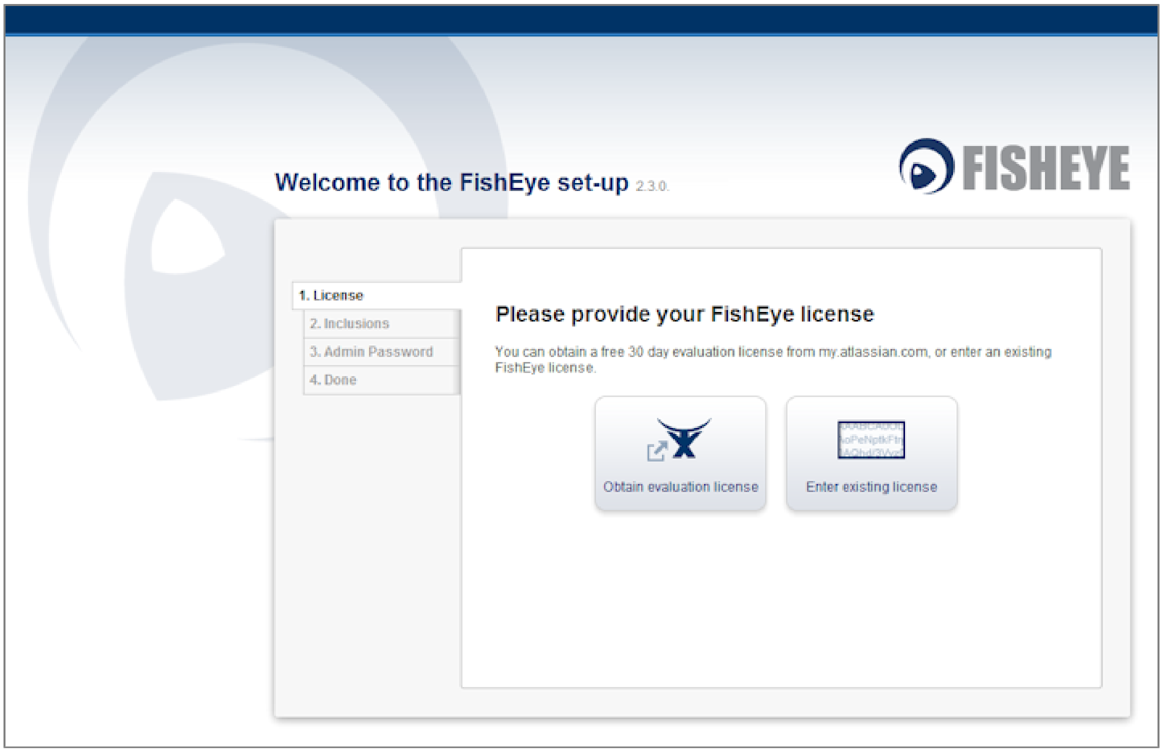
See the EyeQL documentation for more information.

3

Revamped Installation Process

FishEye's installation process has been given a thorough revision and a visual facelift. It's now smoother, faster and provides a better experience.

Screenshot: FishEye's New Installation Screen



Welcome to the FishEye set-up 2.3.0.

1. License
2. Inclusions
3. Admin Password
4. Done

Please provide your FishEye license

You can obtain a free 30 day evaluation license from my.atlassian.com, or enter an existing FishEye license.

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

On this page:

- From 2.3.2 to 2.3.3
- From 2.3.1 to 2.3.2
- From 2.3.0 to 2.3.1

From 2.3.2 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 (5 issues)

Key	Summary	Priority	Status
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

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 (6 issues)

Key	Summary	Priority	Status
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 2.2 Release Notes

> ⚠ FishEye 2.3 has now been released. Read the Release Notes.

18 February 2010

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.

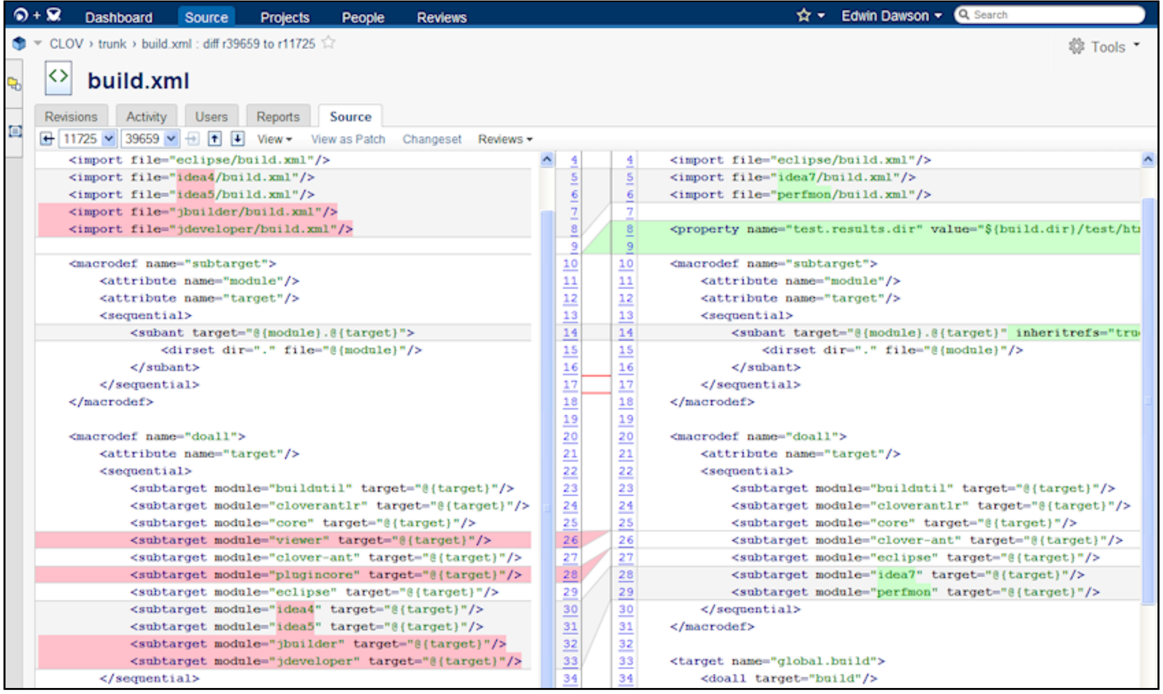
Highlights of FishEye 2.2

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



```
    <import file="eclipse/build.xml"/>
    <import file="idea4/build.xml"/>
    <import file="idea5/build.xml"/>
    <import file="jbuilder/build.xml"/>
    <import file="jdeveloper/build.xml"/>

    <macrodef name="subtarget">
        <attribute name="module"/>
        <attribute name="target"/>
        <sequential>
            <subant target="@{module}.@{target}">
                <dirset dir="." file="@{module}"/>
            </subant>
        </sequential>
    </macrodef>

    <macrodef name="doall">
        <attribute name="target"/>
        <sequential>
            <subtarget module="buildutil" target="@{target}"/>
            <subtarget module="cloverantlr" target="@{target}"/>
            <subtarget module="core" target="@{target}"/>
            <subtarget module="viewer" target="@{target}"/>
            <subtarget module="clover-ant" target="@{target}"/>
            <subtarget module="plugincore" target="@{target}"/>
            <subtarget module="eclipse" target="@{target}"/>
            <subtarget module="idea4" target="@{target}"/>
            <subtarget module="idea5" target="@{target}"/>
            <subtarget module="jbuilder" target="@{target}"/>
            <subtarget module="jdeveloper" target="@{target}"/>
        </sequential>
```

```
    <import file="eclipse/build.xml"/>
    <import file="idea7/build.xml"/>
    <import file="perfmon/build.xml"/>

    <property name="test.results.dir" value="${build.dir}/test/htı

    <macrodef name="subtarget">
        <attribute name="module"/>
        <attribute name="target"/>
        <sequential>
            <subant target="@{module}.@{target}" inheritrefs="tru
                <dirset dir="." file="@{module}"/>
            </subant>
        </sequential>
    </macrodef>

    <macrodef name="doall">
        <attribute name="target"/>
        <sequential>
            <subtarget module="buildutil" target="@{target}"/>
            <subtarget module="cloverantlr" target="@{target}"/>
            <subtarget module="core" target="@{target}"/>
            <subtarget module="clover-ant" target="@{target}"/>
            <subtarget module="eclipse" target="@{target}"/>
            <subtarget module="idea7" target="@{target}"/>
            <subtarget module="perfmon" target="@{target}"/>
        </sequential>
    </macrodef>

    <target name="global.build">
        <doall target="build"/>
```

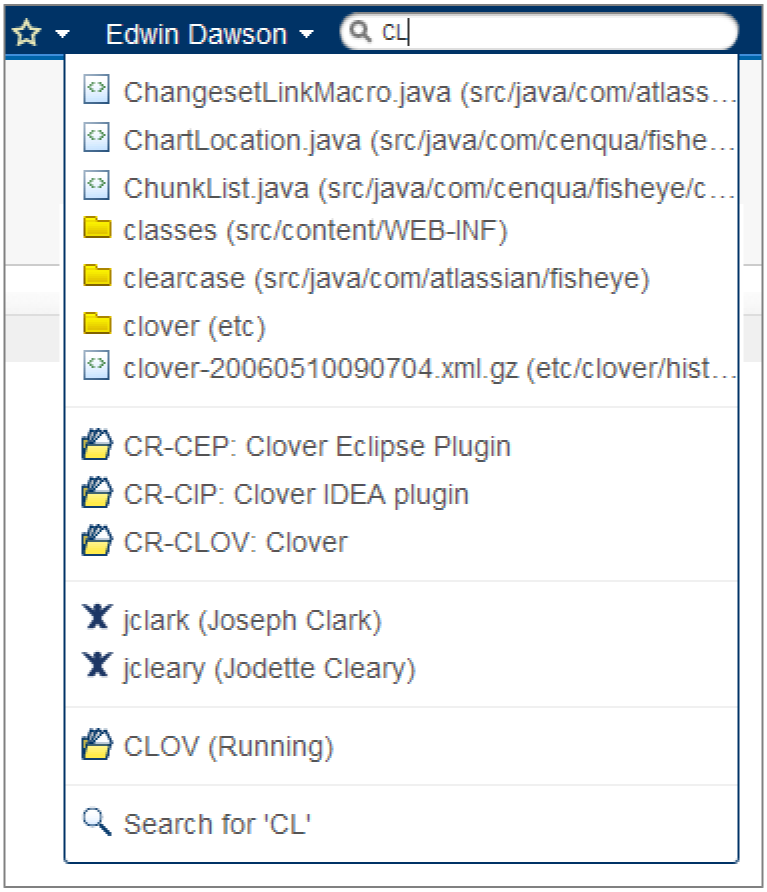
See the documentation for more details.

2

Improved Quick Navigation

FishEye's Quick Nav has been re-written to display results faster, and to return more relevant results. It still supports powerful IDE-like functions such as camelcase detection, multiple directory selection and cross-repository results. All these items appear in a handy preview window, when you type text into the search box.

Screenshot: Improved Quick Nav



Edwin Dawson

CL

ChangesetLinkMacro.java (src/java/com/atlass…

ChartLocation.java (src/java/com/cenqua/fishe…

ChunkList.java (src/java/com/cenqua/fisheye/c…

classes (src/content/WEB-INF)

clearcase (src/java/com/atlassian/fisheye)

clover (etc)

clover-20060510090704.xml.gz (etc/clover/hist…

CR-CEP: Clover Eclipse Plugin

CR-CIP: Clover IDEA plugin

CR-CLOV: Clover

jclark (Joseph Clark)

jcleary (Jodette Cleary)

CLOV (Running)

Search for 'CL'

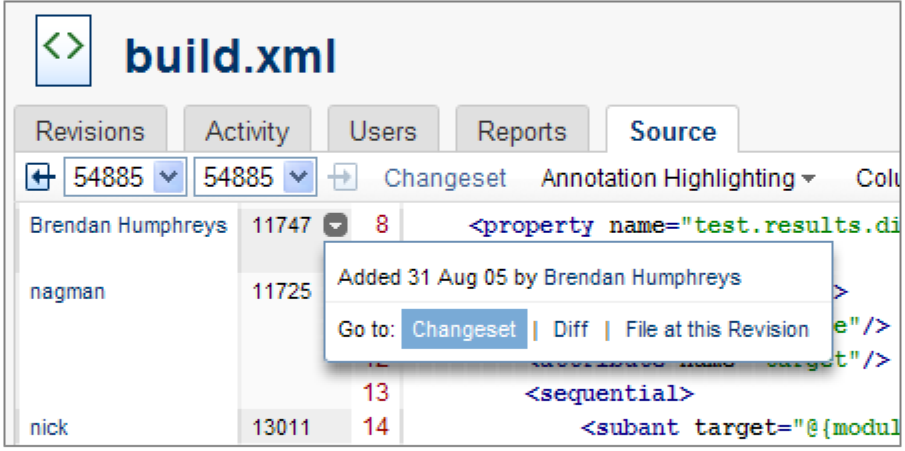
See the documentation for more details.

3

Annotation Context Menu

In FishEye, you can now drill down into the history of any line of code in a file. By hovering your mouse cursor next to the revision number in the left column, you can open the annotation context menu. From there, you can view the changeset where this file was changed, the diff of this line with the last change, or the full file from the time when this line changed. This speeds investigation of how certain lines have changed, when, and by whom.

Screenshot: FishEye Annotation Context Menu



build.xml

Brendan Humphreys	11747	8	`<property name="test.results.di`
nagman	11725		
		13	`<sequential>`
nick	13011	14	`<subant target="@{modul`

Added 31 Aug 05 by Brendan Humphreys

Go to: Changeset | Diff | File at this Revision

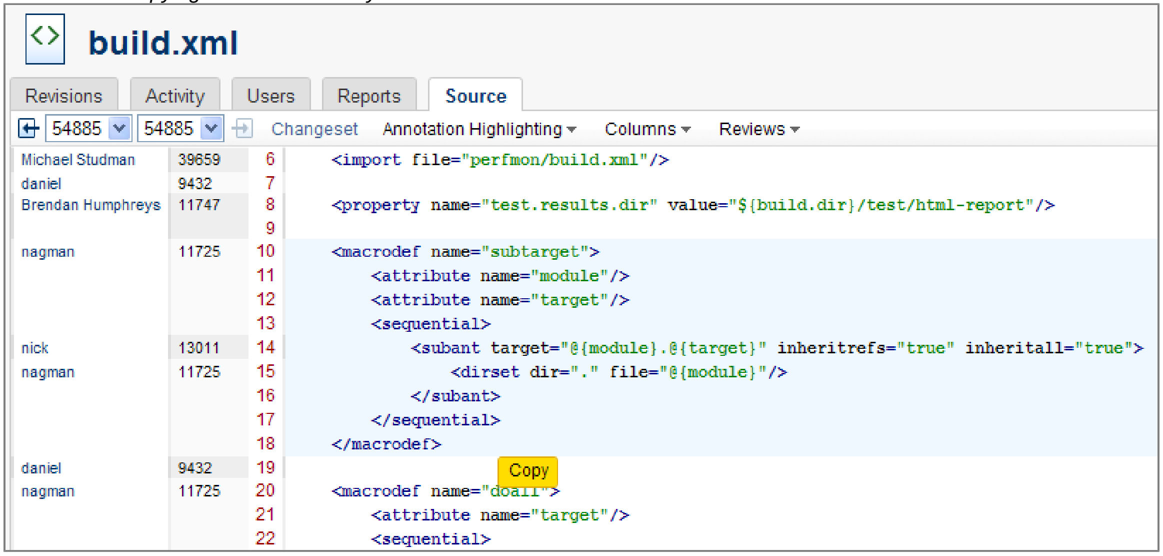
See the documentation for more details.

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



Michael Studman	39659	
daniel	9432	
Brendan Humphreys	11747	
nagman	11725	
nick	13011	
nagman	11725	
daniel	9432	
nagman	11725	

```
<import file="perfmon/build.xml"/>

<property name="test.results.dir" value="${build.dir}/test/html-report"/>

<macrodef name="subtarget">
    <attribute name="module"/>
    <attribute name="target"/>
    <sequential>
        <subant target="@{module}.@{target}" inheritrefs="true" inheritall="true">
            <dirset dir="." file="@{module}"/>
        </subant>
    </sequential>
</macrodef>

<macrodef name="doall">
    <attribute name="target"/>
    <sequential>
```

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

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 (32 issues)

Key	Summary	Priority	Status
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-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

ℹ 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	❗	Closed
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 2.1 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.

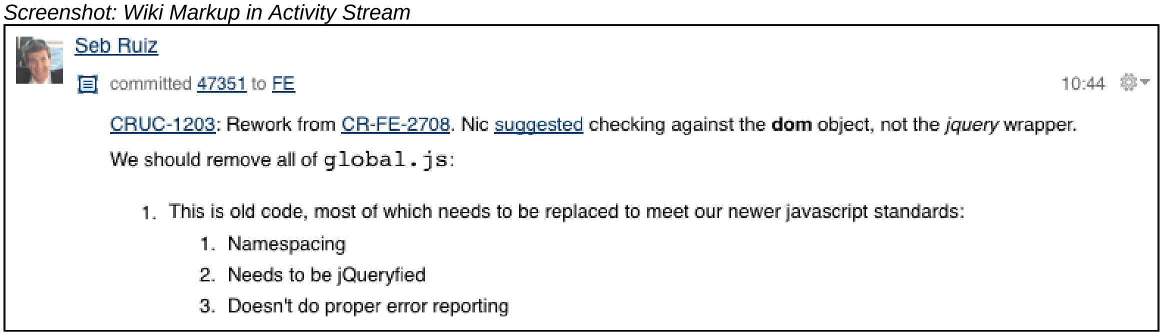
Highlights of FishEye 2.1

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



Seb Ruiz

committed 47351 to FE 10:44

CRUC-1203: Rework from CR-FE-2708. Nic suggested checking against the **dom** object, not the *jquery* wrapper.

We should remove all of `global.js`:

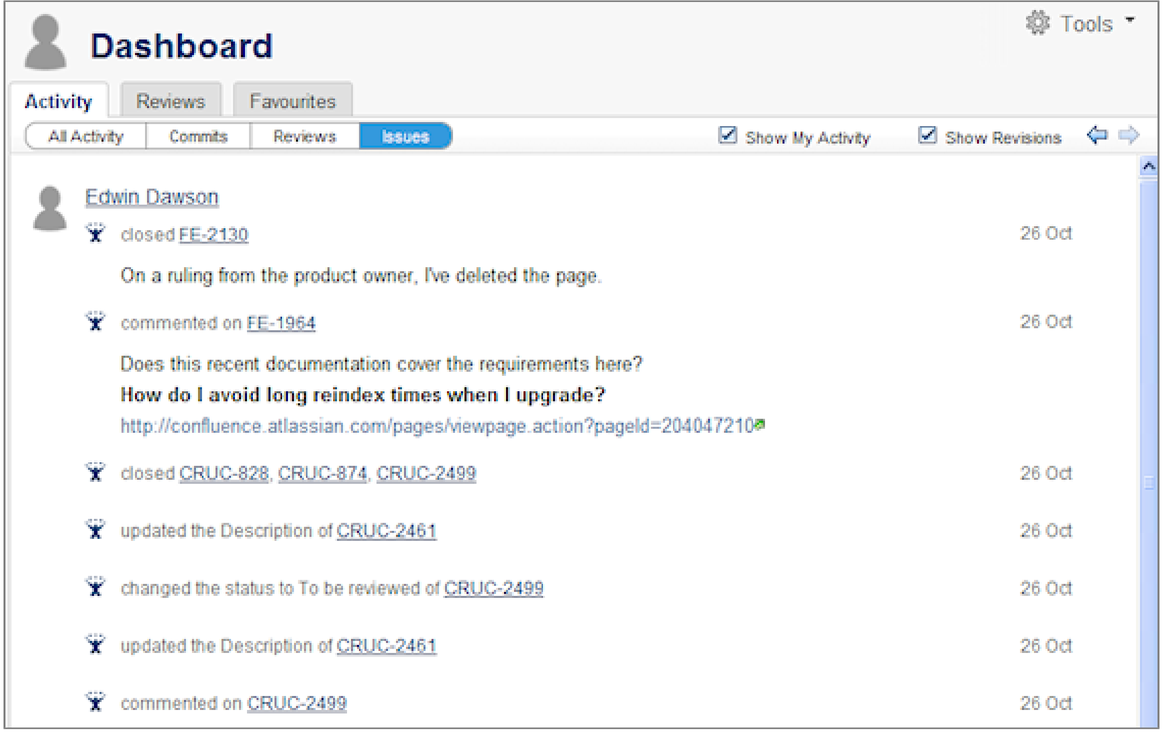
1. This is old code, most of which needs to be replaced to meet our newer javascript standards:
   1. Namespacing
   2. Needs to be jQueryfied
   3. Doesn't do proper error reporting

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



Edwin Dawson

closed FE-2130 26 Oct

On a ruling from the product owner, I've deleted the page.

commented on FE-1964 26 Oct

Does this recent documentation cover the requirements here?
How do I avoid long reindex times when I upgrade?
http://confluence.atlassian.com/pages/viewpage.action?pageId=204047210

closed CRUC-828, CRUC-874, CRUC-2499 26 Oct

updated the Description of CRUC-2461 26 Oct

changed the status to To be reviewed of CRUC-2499 26 Oct

updated the Description of CRUC-2461 26 Oct

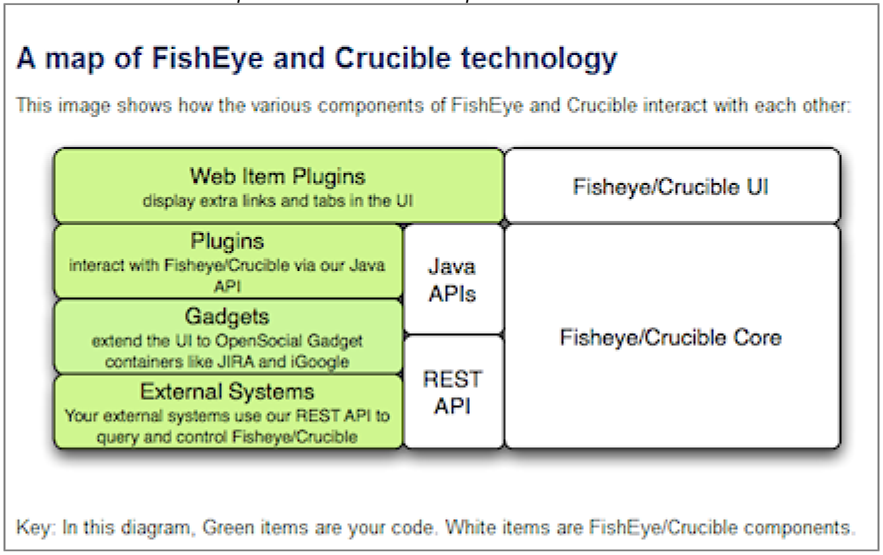
commented on CRUC-2499 26 Oct

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



A map of FishEye and Crucible technology

This image shows how the various components of FishEye and Crucible interact with each other:

Web Item Plugins
display extra links and tabs in the UI
Fisheye/Crucible UI
Plugins
interact with Fisheye/Crucible via our Java API
Java APIs
Gadgets
extend the UI to OpenSocial Gadget containers like JIRA and iGoogle
Fisheye/Crucible Core
External Systems
Your external systems use our REST API to query and control Fisheye/Crucible
REST API

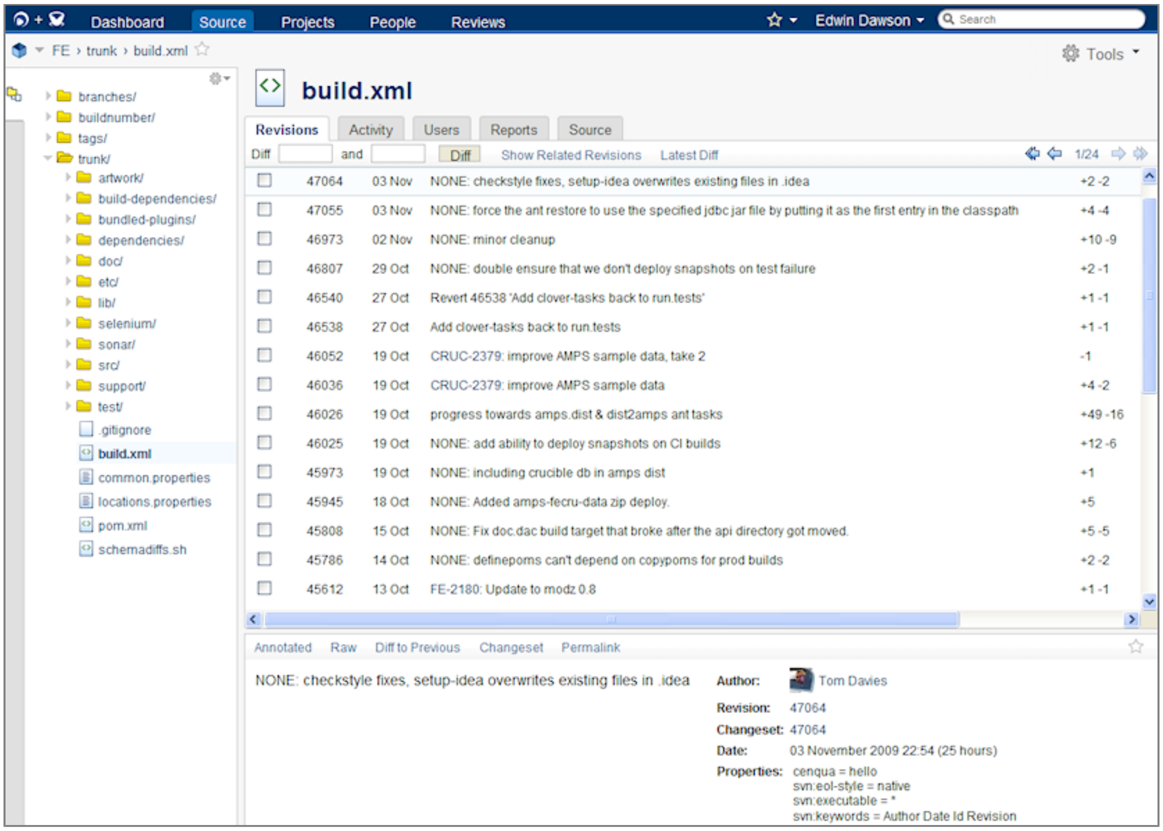
Key: In this diagram, Green items are your code. White items are FishEye/Crucible components.

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



□	47064	03 Nov	NONE: checkstyle fixes, setup-idea overwrites existing files in .idea	+2 -2
□	47055	03 Nov	NONE: force the ant restore to use the specified jdbc jar file by putting it as the first entry in the classpath	+4 -4
□	46973	02 Nov	NONE: minor cleanup	+10 -9
□	46807	29 Oct	NONE: double ensure that we don't deploy snapshots on test failure	+2 -1
□	46540	27 Oct	Revert 46538 'Add clover-tasks back to run.tests'	+1 -1
□	46538	27 Oct	Add clover-tasks back to run.tests	+1 -1
□	46052	19 Oct	CRUC-2379: improve AMPS sample data, take 2	-1
□	46036	19 Oct	CRUC-2379: improve AMPS sample data	+4 -2
□	46026	19 Oct	progress towards amps.dist & dist2amps ant tasks	+49 -16
□	46025	19 Oct	NONE: add ability to deploy snapshots on CI builds	+12 -6
□	45973	19 Oct	NONE: including crucible db in amps dist	+1
□	45945	18 Oct	NONE: Added amps-fecru-data zip deploy.	+5
□	45808	15 Oct	NONE: Fix doc.dac build target that broke after the api directory got moved.	+5 -5
□	45786	14 Oct	NONE: definepoms can't depend on copypoms for prod builds	+2 -2
□	45612	13 Oct	FE-2180: Update to modz 0.8	+1 -1

NONE: checkstyle fixes, setup-idea overwrites existing files in .idea

Author: Tom Davies
Revision: 47064
Changeset: 47064
Date: 03 November 2009 22:54 (25 hours)
Properties: cenqua = hello
svn:eol-style = native
svn:executable = *
svn:keywords = Author Date Id Revision

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

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 2.0 Release Notes

> ⚠ FishEye 2.3 has now been released. Read the Release Notes.

30 June 2009

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.

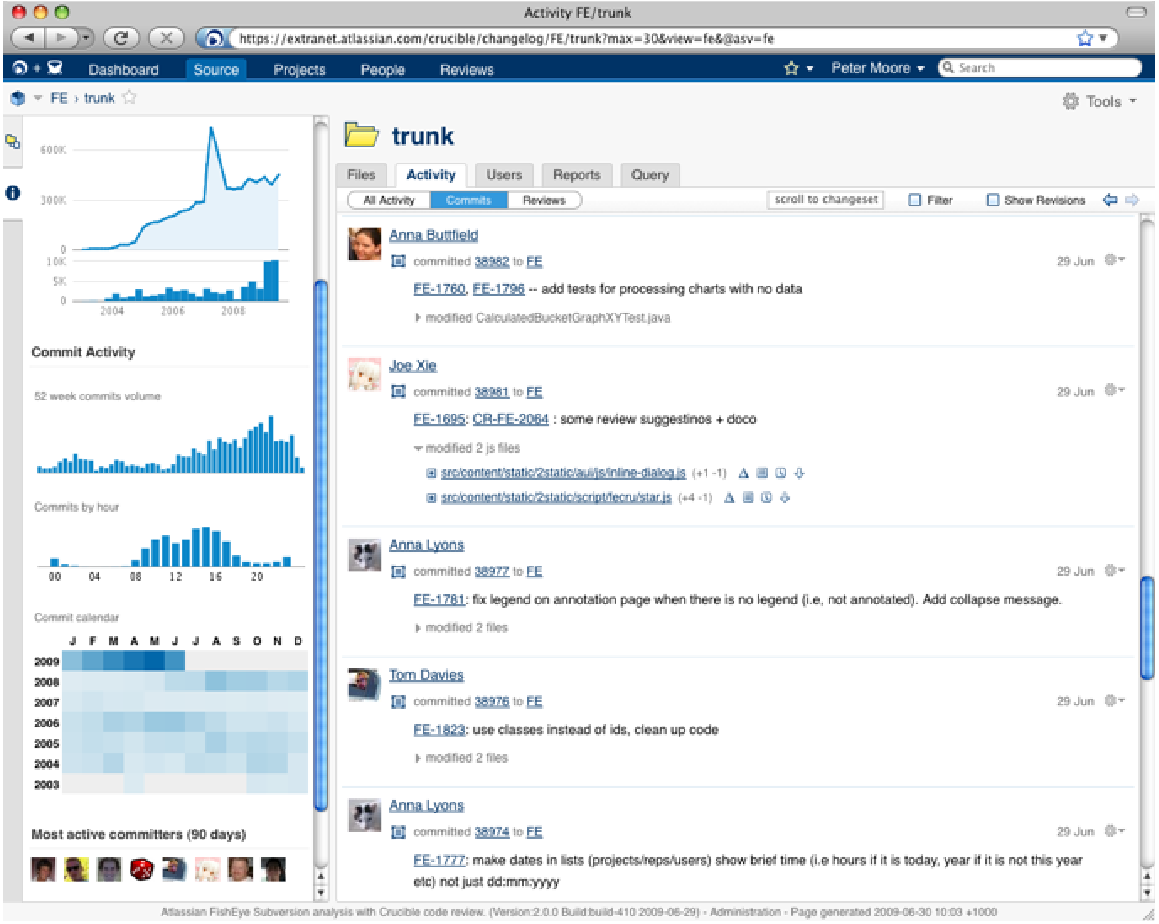
Highlights of FishEye 2.0

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



Anna Buttfield
committed 38982 to FE 29 Jun
FE-1760, FE-1796 -- add tests for processing charts with no data
modified CalculatedBucketGraphXYTest.java

Joe Xie
committed 38981 to FE 29 Jun
FE-1695: CR-FE-2064 : some review suggestinos + doco
modified 2 js files
src/content/static/2static/aui/js/inline-dialog.js (+1 -1)
src/content/static/2static/script/fecru/star.js (+4 -1)

Anna Lyons
committed 38977 to FE 29 Jun
FE-1781: fix legend on annotation page when there is no legend (i.e, not annotated). Add collapse message.
modified 2 files

Tom Davies
committed 38976 to FE 29 Jun
FE-1823: use classes instead of ids, clean up code
modified 2 files

Anna Lyons
committed 38974 to FE 29 Jun
FE-1777: make dates in lists (projects/reps/users) show brief time (i.e hours if it is today, year if it is not this year etc) not just dd:mm:yyyy

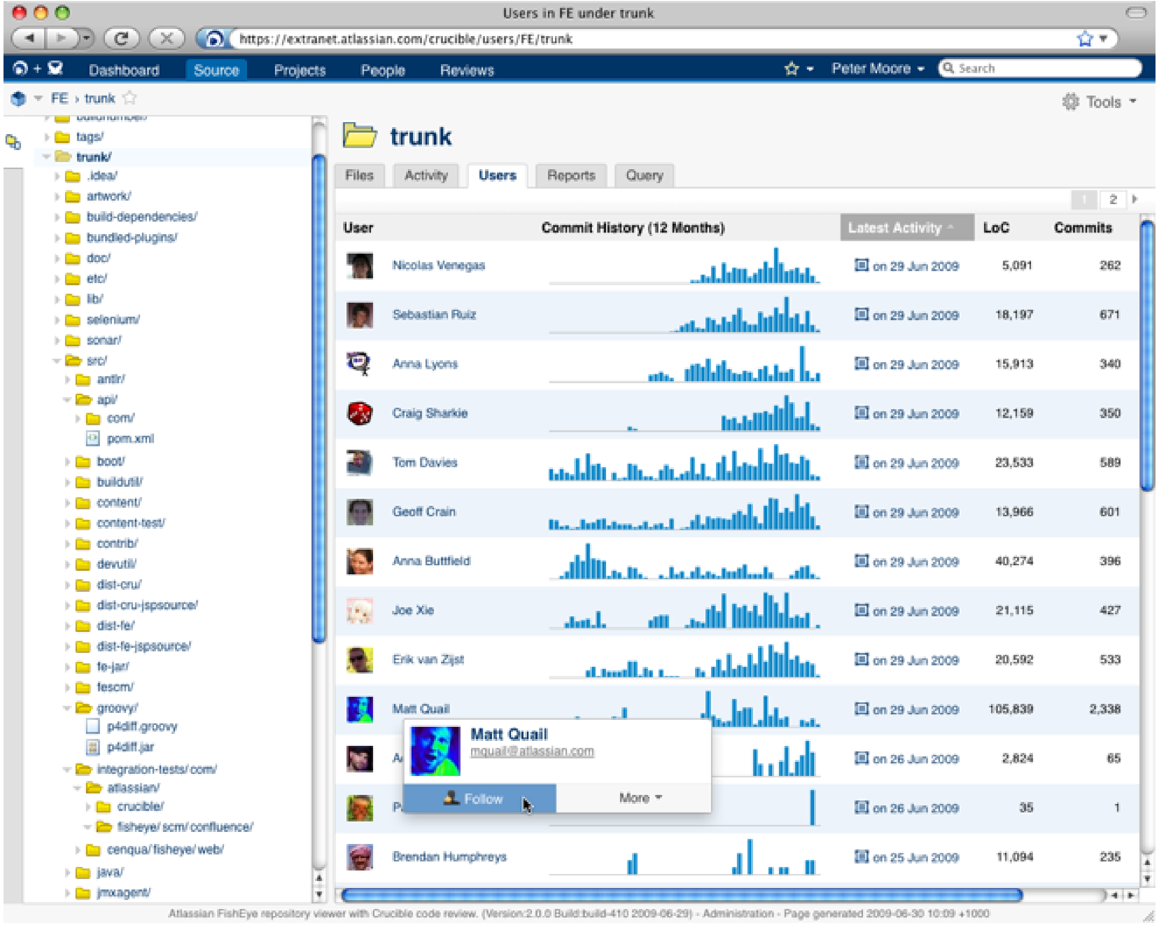
Atlassian FishEye Subversion analysis with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 10:03 +1000

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



User	Commit History (12 Months)	Latest Activity	LoC	Commits
Nicolas Venegas		on 29 Jun 2009	5,091	262
Sebastian Ruiz		on 29 Jun 2009	18,197	671
Anna Lyons		on 29 Jun 2009	15,913	340
Craig Sharkie		on 29 Jun 2009	12,159	350
Tom Davies		on 29 Jun 2009	23,533	589
Geoff Crain		on 29 Jun 2009	13,966	601
Anna Buttfield		on 29 Jun 2009	40,274	396
Joe Xie		on 29 Jun 2009	21,115	427
Erik van Zijst		on 29 Jun 2009	20,592	533
Matt Quail		on 29 Jun 2009	105,839	2,338
A		on 26 Jun 2009	2,824	65
P		on 26 Jun 2009	35	1
Brendan Humphreys		on 25 Jun 2009	11,094	235

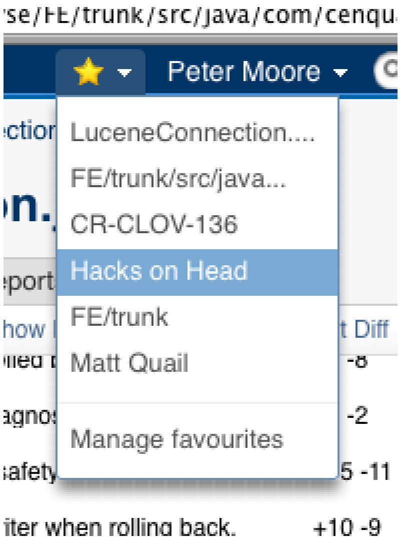
Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 10:09 +1000

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

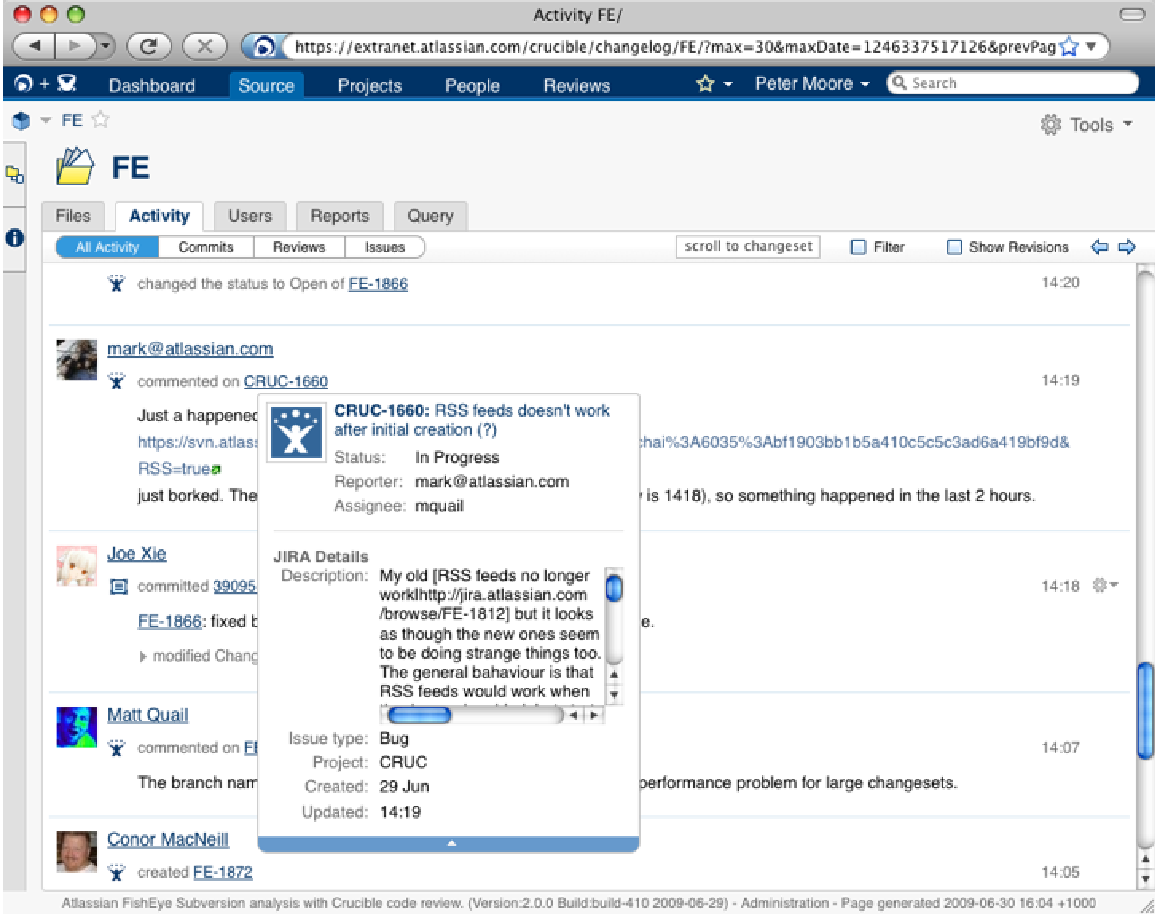


4

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



changed the status to Open of FE-1866 14:20

mark@atlassian.com
commented on CRUC-1660 14:19
Just a happened
https://svn.atlas hai%3A6035%3Abf1903bb1b5a410c5c5c3ad6a419bf9d&
RSS=true
just borked. The is 1418), so something happened in the last 2 hours.

Joe Xie
committed 39095 14:18
FE-1866: fixed b e.
modified Chang

Matt Quail
commented on FE 14:07
The branch nam performance problem for large changesets.

Conor MacNeill
created FE-1872 14:05

CRUC-1660: RSS feeds doesn't work after initial creation (?)

Status: In Progress
Reporter: mark@atlassian.com
Assignee: mquail

JIRA Details

Description: My old [RSS feeds no longer work|http://jira.atlassian.com/browse/FE-1812] but it looks as though the new ones seem to be doing strange things too. The general bahaviour is that RSS feeds would work when

Issue type: Bug
Project: CRUC
Created: 29 Jun
Updated: 14:19

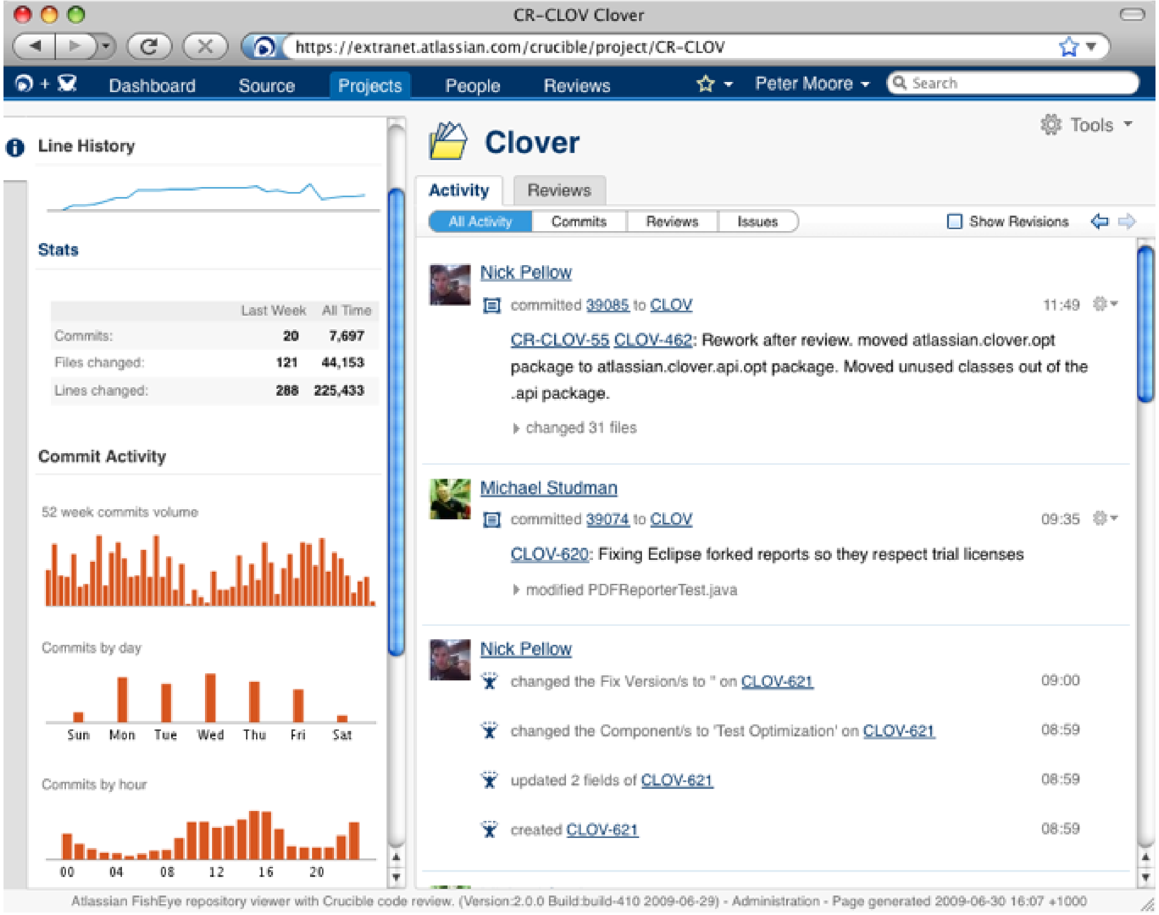
Atlassian FishEye Subversion analysis with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 16:04 +1000

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



	Last Week	All Time
Commits:	20	7,697
Files changed:	121	44,153
Lines changed:	288	225,433

Nick Pellow
committed 39085 to CLOV 11:49
CR-CLOV-55 CLOV-462: Rework after review. moved atlassian.clover.opt package to atlassian.clover.api.opt package. Moved unused classes out of the .api package.
changed 31 files

Michael Studman
committed 39074 to CLOV 09:35
CLOV-620: Fixing Eclipse forked reports so they respect trial licenses
modified PDFReporterTest.java

Nick Pellow
changed the Fix Version/s to '' on CLOV-621 09:00
changed the Component/s to 'Test Optimization' on CLOV-621 08:59
updated 2 fields of CLOV-621 08:59
created CLOV-621 08:59

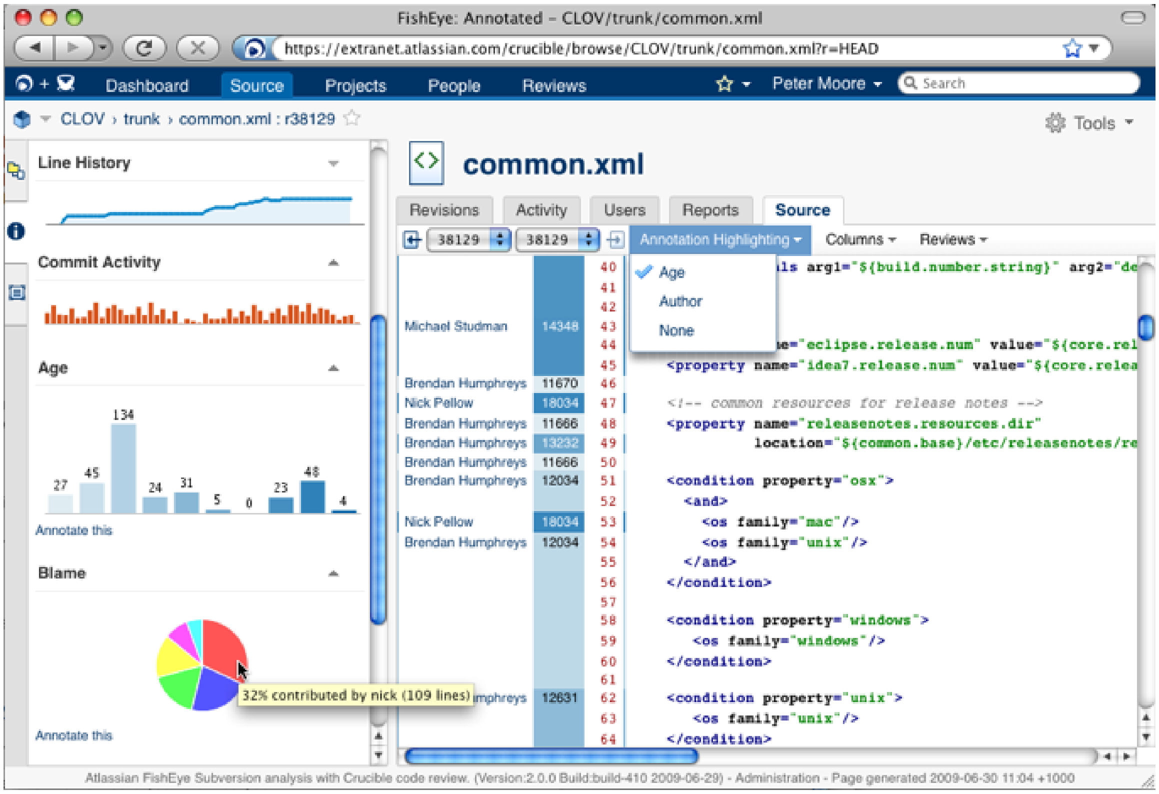
Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 16:07 +1000

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



```
                    ils arg1="${build.number.string}" arg2="de



                    e="eclipse.release.num" value="${core.rel
    <property name="idea7.release.num" value="${core.relea

    <!-- common resources for release notes -->
    <property name="releasenotes.resources.dir"
              location="${common.base}/etc/releasenotes/re

    <condition property="osx">
      <and>
        <os family="mac"/>
        <os family="unix"/>
      </and>
    </condition>

    <condition property="windows">
       <os family="windows"/>
    </condition>

    <condition property="unix">
       <os family="unix"/>
    </condition>
```

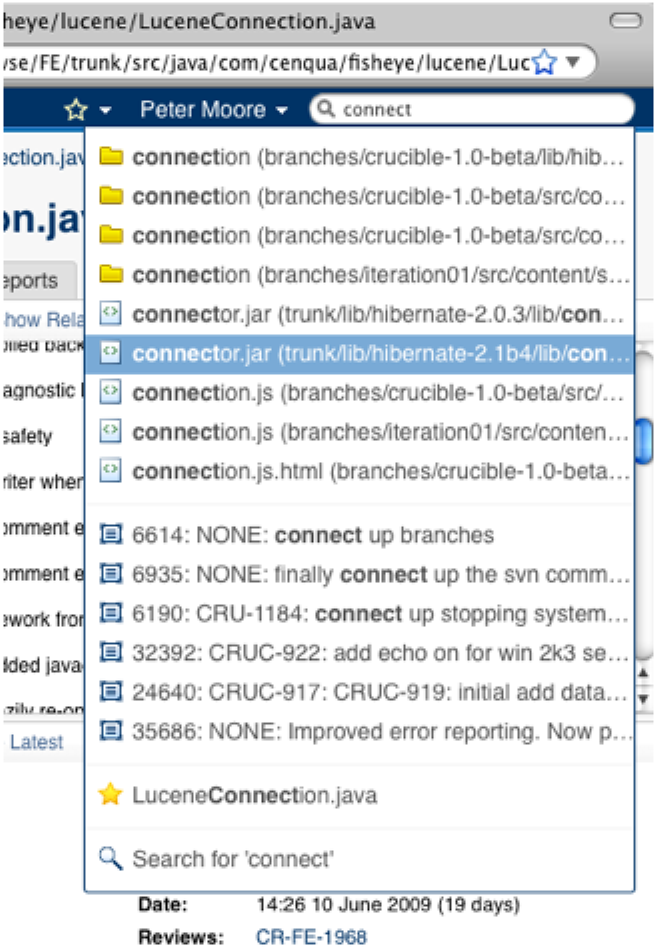
Atlassian FishEye Subversion analysis with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 11:04 +1000

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



heye/lucene/LuceneConnection.java

/se/FE/trunk/src/java/com/cenqua/fisheye/lucene/Luc

Peter Moore

connect

- connection (branches/crucible-1.0-beta/lib/hib…
- connection (branches/crucible-1.0-beta/src/co…
- connection (branches/crucible-1.0-beta/src/co…
- connection (branches/iteration01/src/content/s…
- connector.jar (trunk/lib/hibernate-2.0.3/lib/**con**…
- connector.jar (trunk/lib/hibernate-2.1b4/lib/**con**…
- connection.js (branches/crucible-1.0-beta/src/…
- connection.js (branches/iteration01/src/conten…
- connection.js.html (branches/crucible-1.0-beta…

- 6614: NONE: **connect** up branches
- 6935: NONE: finally **connect** up the svn comm…
- 6190: CRU-1184: **connect** up stopping system…
- 32392: CRUC-922: add echo on for win 2k3 se…
- 24640: CRUC-917: CRUC-919: initial add data…
- 35686: NONE: Improved error reporting. Now p…

- Lucene**Connect**ion.java

- Search for 'connect'

Date: 14:26 10 June 2009 (19 days)

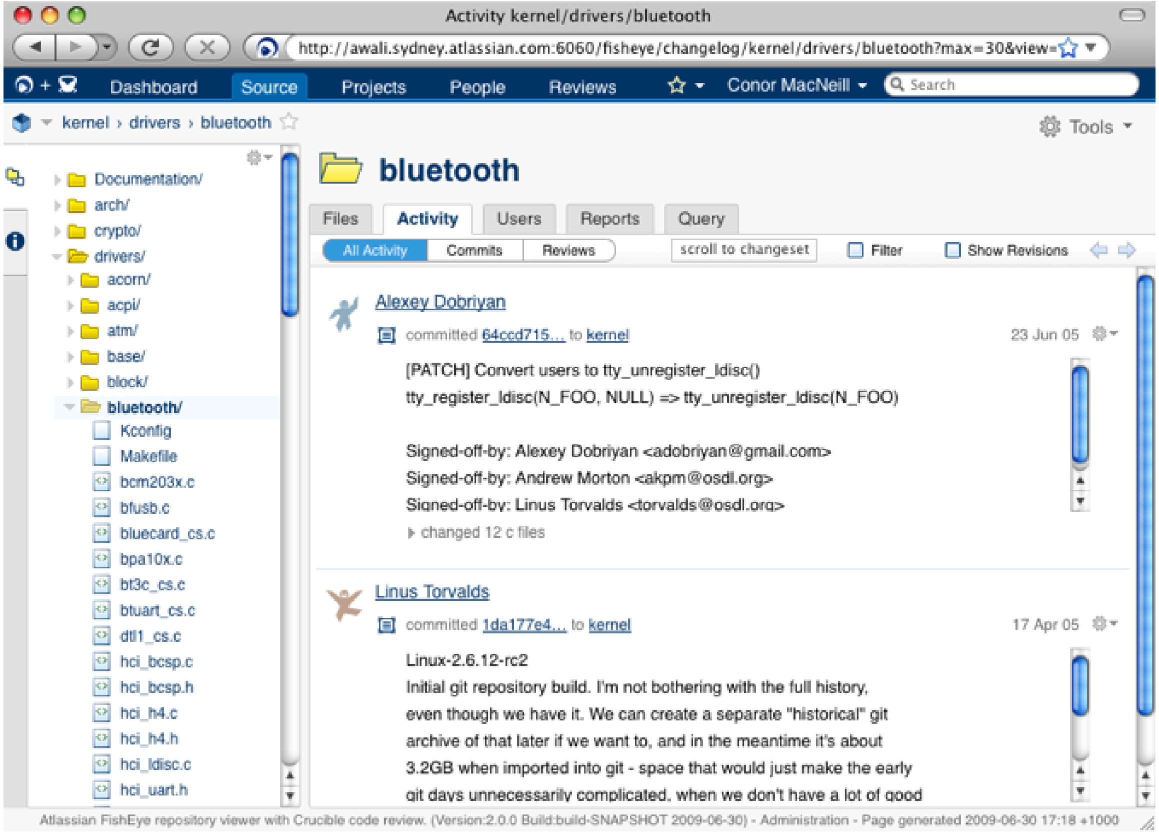
Reviews: CR-FE-1968

8

Git Beta

FishEye 2 adds beta support for Git repositories. There are still a few kinks to be worked out, but it provides full access to FishEye's functionality.

Screenshot: The linux kernel's Git repository in FishEye



Alexey Dobriyan
committed 64ccd715... to kernel 23 Jun 05

[PATCH] Convert users to tty_unregister_ldisc()
tty_register_ldisc(N_FOO, NULL) => tty_unregister_ldisc(N_FOO)

Signed-off-by: Alexey Dobriyan <adobriyan@gmail.com>
Signed-off-by: Andrew Morton <akpm@osdl.org>
Signed-off-by: Linus Torvalds <torvalds@osdl.org>

changed 12 c files

Linus Torvalds
committed 1da177e4... to kernel 17 Apr 05

Linux-2.6.12-rc2
Initial git repository build. I'm not bothering with the full history, even though we have it. We can create a separate "historical" git archive of that later if we want to, and in the meantime it's about 3.2GB when imported into git - space that would just make the early git days unnecessarily complicated, when we don't have a lot of good

Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0 Build:build-SNAPSHOT 2009-06-30) - Administration - Page generated 2009-06-30 17:18 +1000

9

Numerous improvements and bug fixes

Visit our issue tracker to see the full list of improvements and bug fixes.

FishEye 2.0 Changelog

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	⬇	Resolved
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	⬆	Resolved
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-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	⬆	Resolved
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	⬆	Resolved
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	↑	Resolved
FE-1483	Enhance directory tree to display file	↑	Resolved
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	↑	Resolved
FE-1464	Report Mode Navigation (similar to Search Mode Navigation)	↑	Resolved
FE-1463	Root (No Repository Context) Search Page	↑	Resolved
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	↑	Resolved
FE-1447	Search Mode Navigation	↑	Resolved
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	⬆	Resolved
FE-1410	View Changeset Stream Context Fixes	⬆	Resolved
FE-1408	activity-streams plugin wrangling	⬆	Closed
FE-1401	add a comment as one of the return columns in /search/FE/	⬆	Resolved
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	⬇	Resolved
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	⬆	Resolved
FE-1357	write tag	⬆	Closed
FE-1356	Consistent time and age formats	⬆	Resolved
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	⬆	Resolved
FE-1352	just do it	⬆	Closed
FE-1351	preference and toggle to exclude own activity from home page stream	⬆	Resolved

FE-1349	just do it	⇧	Closed
FE-1348	Switch all hover popups to use ajs.hover	⇧	Resolved
FE-1347	just do it	⇧	Closed
FE-1346	Open settings pages in ajs.dialog	⇧	Resolved
FE-1345	gravatar servlet	⇧	Closed
FE-1344	config screen + jsps	⇧	Closed
FE-1343	Avatar config screens	⇧	Resolved
FE-1342	make the committer page group by user if there are multiple committer with same user	⇧	Closed
FE-1341	User & committer list rationalisation	⇧	Resolved
FE-1340	just do it	⇧	Closed
FE-1339	One jsp for global quicksearch and repo qsearch	⇧	Resolved
FE-1338	just do it	⇧	Closed
FE-1337	Switch all dropdowns to ajs.dropdown	⇧	Resolved
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	⇧	Resolved
FE-1334	legend unfucking	⇧	Closed
FE-1333	sparkline	⇧	Closed
FE-1332	line history sparklines and user line history chart improvements	⇧	Resolved
FE-1331	just do it	⇧	Closed
FE-1330	Integrate new html for diff/annotation pages	⇧	Resolved
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
	there is no real point paging if we are going to do these types of calculations for every user in the system. It's		

FE-1313	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	⬆	Resolved
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	⬆	Resolved
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	⬆	Resolved
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	⇧	Resolved
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	⇧	Resolved
FE-1251	create reports page	⇧	Closed
FE-1250	create webitem + decorator	⇧	Closed
FE-1249	Reports tab	⇧	Resolved
FE-1247	jsp wrangling	⇧	Closed
FE-1246	action	⇧	Closed
FE-1245	Add a project page	⇧	Resolved
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.	⇧	Resolved
FE-1241	refactor javascript - remove prototype	⇧	Closed
FE-1240	Javascript refactor	⇧	Resolved
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	⇧	Resolved
FE-1235	UI integration	⇧	Closed
FE-1234	render chart table thingy	⇧	Closed
FE-1233	searchy extractor thingy	⇧	Closed

FE-1232	Activity calendar	⬆	Resolved
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	⬆	Resolved
FE-1227	UI integration	⬆	Closed
FE-1226	render the chart	⬆	Closed
FE-1225	implement searchy extractor thingy	⬆	Closed
FE-1224	Recent activity sparkline & chart	⬆	Resolved
FE-1223	just do it	⬆	Closed
FE-1222	Show personal review summary on user home page	⬆	Resolved
FE-1221	just do it	⬆	Closed
FE-1220	Show open review count on user pages	⬆	Resolved
FE-1218	jiralinkspan span is created inside anchor tags	⬆	Closed
FE-1217	simple impl	⬆	Closed
FE-1216	Ubiquitous (cross repo) quicksearch/nav	⬆	Resolved
FE-1212	just do it	⬆	Closed
FE-1211	repo dropdown in breadcrumb bar	⬆	Resolved
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	⬆	Resolved
FE-1206	ajax remaining data	⬆	Closed
FE-1204	Use a full tree to navigate the new browse page	⬆	Resolved
FE-1196	make action	⬆	Closed
FE-1195	make a tag / javascript function / standardise	⬆	Closed

FE-1194	Built in smart Crucible linker	⇧	Resolved
FE-1183	make repository and crucible defaults for jira issues that are not mapped redux	⇧	Resolved
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
FE-743	Introduce validation for Updater in fisheye or make configuration foolproof	⇩	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-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	⬆	Resolved
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	⇧	Resolved
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	⇧	Resolved
FE-1483	Enhance directory tree to display file	⇧	Resolved
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	⇧	Resolved
FE-1464	Report Mode Navigation (similar to Search Mode Navigation)	⇧	Resolved
FE-1463	Root (No Repository Context) Search Page	⇧	Resolved
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	⇧	Resolved

FE-1447	Search Mode Navigation	↑	Resolved
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	↑	Resolved
FE-1410	View Changeset Stream Context Fixes	↑	Resolved
FE-1408	activity-streams plugin wrangling	↑	Closed
FE-1401	add a comment as one of the return columns in /search/FE/	↑	Resolved
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	↓	Resolved
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	⬆	Resolved
FE-1357	write tag	⬆	Closed
FE-1356	Consistent time and age formats	⬆	Resolved
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	⬆	Resolved
FE-1352	just do it	⬆	Closed
FE-1351	preference and toggle to exclude own activity from home page stream	⬆	Resolved
FE-1349	just do it	⬆	Closed
FE-1348	Switch all hover popups to use ajs.hover	⬆	Resolved
FE-1347	just do it	⬆	Closed
FE-1346	Open settings pages in ajs.dialog	⬆	Resolved
FE-1345	gravatar servlet	⬆	Closed
FE-1344	config screen + jsps	⬆	Closed
FE-1343	Avatar config screens	⬆	Resolved
FE-1342	make the committer page group by user if there are multiple committer with same user	⬆	Closed
FE-1341	User & committer list rationalisation	⬆	Resolved
FE-1340	just do it	⬆	Closed
FE-1339	One jsp for global quicksearch and repo qsearch	⬆	Resolved
FE-1338	just do it	⬆	Closed
FE-1337	Switch all dropdowns to ajs.dropdown	⬆	Resolved
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	⬆	Resolved
FE-1334	legend unfucking	⬆	Closed

FE-1333	sparkline	⬆	Closed
FE-1332	line history sparklines and user line history chart improvements	⬆	Resolved
FE-1331	just do it	⬆	Closed
FE-1330	Integrate new html for diff/annotation pages	⬆	Resolved
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	⬆	Resolved
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	⬆	Resolved
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	⬆	Resolved
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	⇧	Resolved
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	⇧	Resolved
FE-1251	create reports page	⇧	Closed
FE-1250	create webitem + decorator	⇧	Closed
FE-1249	Reports tab	⇧	Resolved
FE-1247	jsp wrangling	⇧	Closed
FE-1246	action	⇧	Closed
FE-1245	Add a project page	⇧	Resolved
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.	⇧	Resolved
FE-1241	refactor javascript - remove prototype	⇧	Closed

FE-1240	Javascript refactor	⬆	Resolved
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	⬆	Resolved
FE-1235	UI integration	⬆	Closed
FE-1234	render chart table thingy	⬆	Closed
FE-1233	searchy extractor thingy	⬆	Closed
FE-1232	Activity calendar	⬆	Resolved
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	⬆	Resolved
FE-1227	UI integration	⬆	Closed
FE-1226	render the chart	⬆	Closed
FE-1225	implement searchy extractor thingy	⬆	Closed
FE-1224	Recent activity sparkline & chart	⬆	Resolved
FE-1223	just do it	⬆	Closed
FE-1222	Show personal review summary on user home page	⬆	Resolved
FE-1221	just do it	⬆	Closed
FE-1220	Show open review count on user pages	⬆	Resolved
FE-1218	jiralinkspan span is created inside anchor tags	⬆	Closed
FE-1217	simple impl	⬆	Closed
FE-1216	Ubiquitous (cross repo) quicksearch/nav	⬆	Resolved

FE-1212	just do it	⇧	Closed
FE-1211	repo dropdown in breadcrumb bar	⇧	Resolved
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	⇧	Resolved
FE-1206	ajax remaining data	⇧	Closed
FE-1204	Use a full tree to navigate the new browse page	⇧	Resolved
FE-1196	make action	⇧	Closed
FE-1195	make a tag / javascript function / standardise	⇧	Closed
FE-1194	Built in smart Crucible linker	⇧	Resolved
FE-1183	make repository and crucible defaults for jira issues that are not mapped redux	⇧	Resolved
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
FE-743	Introduce validation for Updater in fisheye or make configuration foolproof	⬇	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-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	⇧	Resolved
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	⇧	Resolved
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	⇧	Resolved
FE-1483	Enhance directory tree to display file	⇧	Resolved
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	⇧	Resolved
FE-1464	Report Mode Navigation (similar to Search Mode Navigation)	⇧	Resolved

FE-1463	Root (No Repository Context) Search Page	⬆	Resolved
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	⬆	Resolved
FE-1447	Search Mode Navigation	⬆	Resolved
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	⬆	Resolved
FE-1410	View Changeset Stream Context Fixes	⬆	Resolved
FE-1408	activity-streams plugin wrangling	⬆	Closed
FE-1401	add a comment as one of the return columns in /search/FE/	⬆	Resolved
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	⬇	Resolved
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	⬆	Resolved
FE-1357	write tag	⬆	Closed
FE-1356	Consistent time and age formats	⬆	Resolved
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	⬆	Resolved
FE-1352	just do it	⬆	Closed
FE-1351	preference and toggle to exclude own activity from home page stream	⬆	Resolved
FE-1349	just do it	⬆	Closed
FE-1348	Switch all hover popups to use ajs.hover	⬆	Resolved
FE-1347	just do it	⬆	Closed
FE-1346	Open settings pages in ajs.dialog	⬆	Resolved
FE-1345	gravatar servlet	⬆	Closed
FE-1344	config screen + jsps	⬆	Closed
FE-1343	Avatar config screens	⬆	Resolved
FE-1342	make the committer page group by user if there are multiple committer with same user	⬆	Closed
FE-1341	User & committer list rationalisation	⬆	Resolved

FE-1340	just do it	⬆	Closed
FE-1339	One jsp for global quicksearch and repo qsearch	⬆	Resolved
FE-1338	just do it	⬆	Closed
FE-1337	Switch all dropdowns to ajs.dropdown	⬆	Resolved
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	⬆	Resolved
FE-1334	legend unfucking	⬆	Closed
FE-1333	sparkline	⬆	Closed
FE-1332	line history sparklines and user line history chart improvements	⬆	Resolved
FE-1331	just do it	⬆	Closed
FE-1330	Integrate new html for diff/annotation pages	⬆	Resolved
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	⬆	Resolved
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	⇧	Resolved
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	⇧	Resolved
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	⇧	Resolved
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	⇧	Resolved
FE-1251	create reports page	⇧	Closed
FE-1250	create webitem + decorator	⇧	Closed

FE-1249	Reports tab	⬆	Resolved
FE-1247	jsp wrangling	⬆	Closed
FE-1246	action	⬆	Closed
FE-1245	Add a project page	⬆	Resolved
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.	⬆	Resolved
FE-1241	refactor javascript - remove prototype	⬆	Closed
FE-1240	Javascript refactor	⬆	Resolved
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	⬆	Resolved
FE-1235	UI integration	⬆	Closed
FE-1234	render chart table thingy	⬆	Closed
FE-1233	searchy extractor thingy	⬆	Closed
FE-1232	Activity calendar	⬆	Resolved
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	⬆	Resolved
FE-1227	UI integration	⬆	Closed
FE-1226	render the chart	⬆	Closed
FE-1225	implement searchy extractor thingy	⬆	Closed
FE-1224	Recent activity sparkline & chart	⬆	Resolved

FE-1223	just do it	↑	Closed
FE-1222	Show personal review summary on user home page	↑	Resolved
FE-1221	just do it	↑	Closed
FE-1220	Show open review count on user pages	↑	Resolved
FE-1218	jiralinkspan span is created inside anchor tags	↑	Closed
FE-1217	simple impl	↑	Closed
FE-1216	Ubiquitous (cross repo) quicksearch/nav	↑	Resolved
FE-1212	just do it	↑	Closed
FE-1211	repo dropdown in breadcrumb bar	↑	Resolved
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	↑	Resolved
FE-1206	ajax remaining data	↑	Closed
FE-1204	Use a full tree to navigate the new browse page	↑	Resolved
FE-1196	make action	↑	Closed
FE-1195	make a tag / javascript function / standardise	↑	Closed
FE-1194	Built in smart Crucible linker	↑	Resolved
FE-1183	make repository and crucible defaults for jira issues that are not mapped redux	↑	Resolved
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
FE-743	Introduce validation for Updater in fisheye or make configuration foolproof	⬇	✓ Closed

FishEye 2.0 Upgrade Notes

This page contains information about upgrading to FishEye 2.0.

On this page:

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.

Known Issues

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.

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.

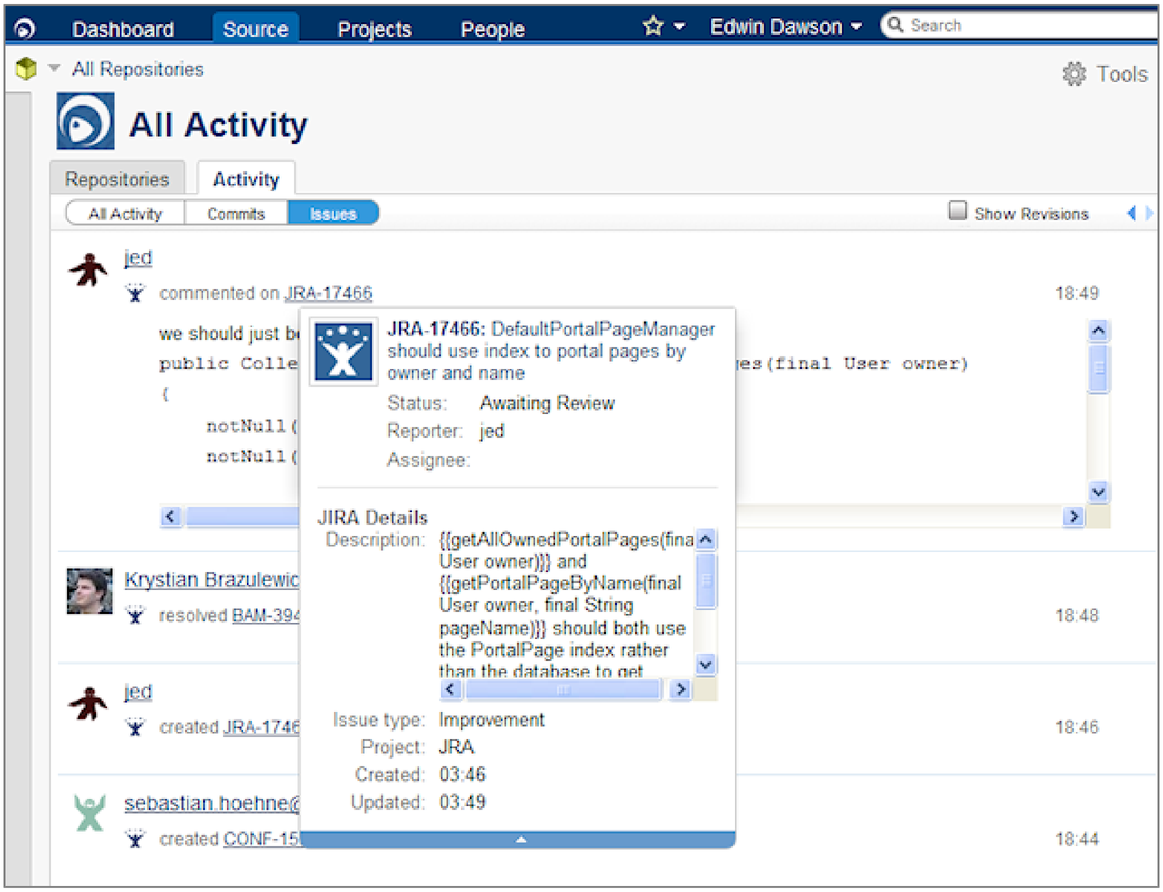
Highlights of FishEye 2.0 Beta

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



jed
commented on JRA-17466 18:49

we should just b
public Colle ... es(final User owner)
{
notNull(
notNull(

Krystian Brazulewic
resolved BAM-394 18:48

jed
created JRA-1746 18:46

sebastian.hoehne@
created CONF-15 18:44

JRA-17466: DefaultPortalPageManager should use index to portal pages by owner and name

Status: Awaiting Review
Reporter: jed
Assignee:

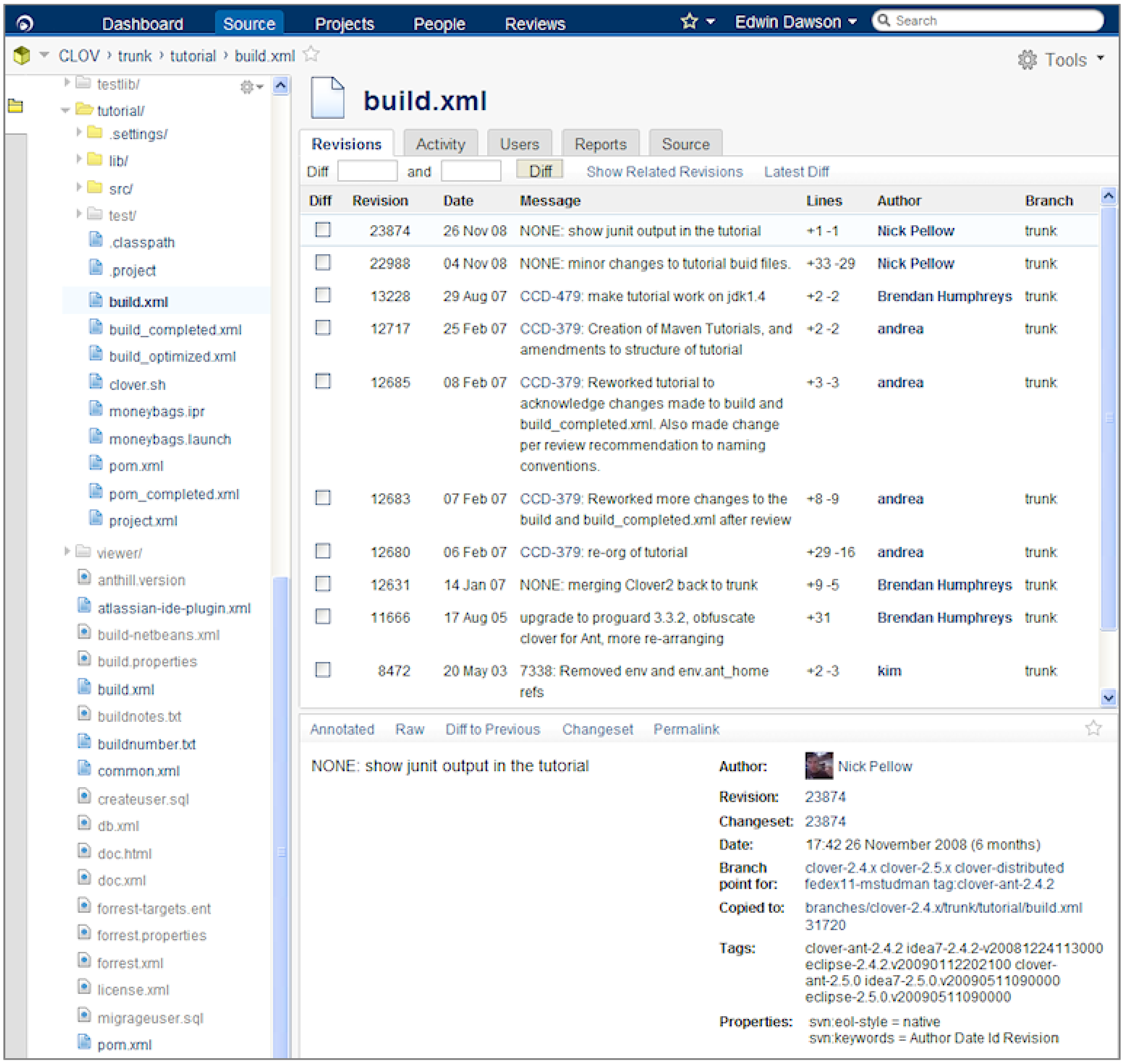
JIRA Details
Description: {{getAllOwnedPortalPages(fina User owner)}} and {{getPortalPageByName(final User owner, final String pageName)}} should both use the PortalPage index rather than the database to get
Issue type: Improvement
Project: JRA
Created: 03:46
Updated: 03:49

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



Diff	Revision	Date	Message	Lines	Author	Branch
□	23874	26 Nov 08	NONE: show junit output in the tutorial	+1 -1	Nick Pellow	trunk
□	22988	04 Nov 08	NONE: minor changes to tutorial buid files.	+33 -29	Nick Pellow	trunk
□	13228	29 Aug 07	CCD-479: make tutorial work on jdk1.4	+2 -2	Brendan Humphreys	trunk
□	12717	25 Feb 07	CCD-379: Creation of Maven Tutorials, and amendments to structure of tutorial	+2 -2	andrea	trunk
□	12685	08 Feb 07	CCD-379: Reworked tutorial to acknowledge changes made to build and build_completed.xml. Also made change per review recommendation to naming conventions.	+3 -3	andrea	trunk
□	12683	07 Feb 07	CCD-379: Reworked more changes to the build and build_completed.xml after review	+8 -9	andrea	trunk
□	12680	06 Feb 07	CCD-379: re-org of tutorial	+29 -16	andrea	trunk
□	12631	14 Jan 07	NONE: merging Clover2 back to trunk	+9 -5	Brendan Humphreys	trunk
□	11666	17 Aug 05	upgrade to proguard 3.3.2, obfuscate clover for Ant, more re-arranging	+31	Brendan Humphreys	trunk
□	8472	20 May 03	7338: Removed env and env.ant_home refs	+2 -3	kim	trunk

NONE: show junit output in the tutorial

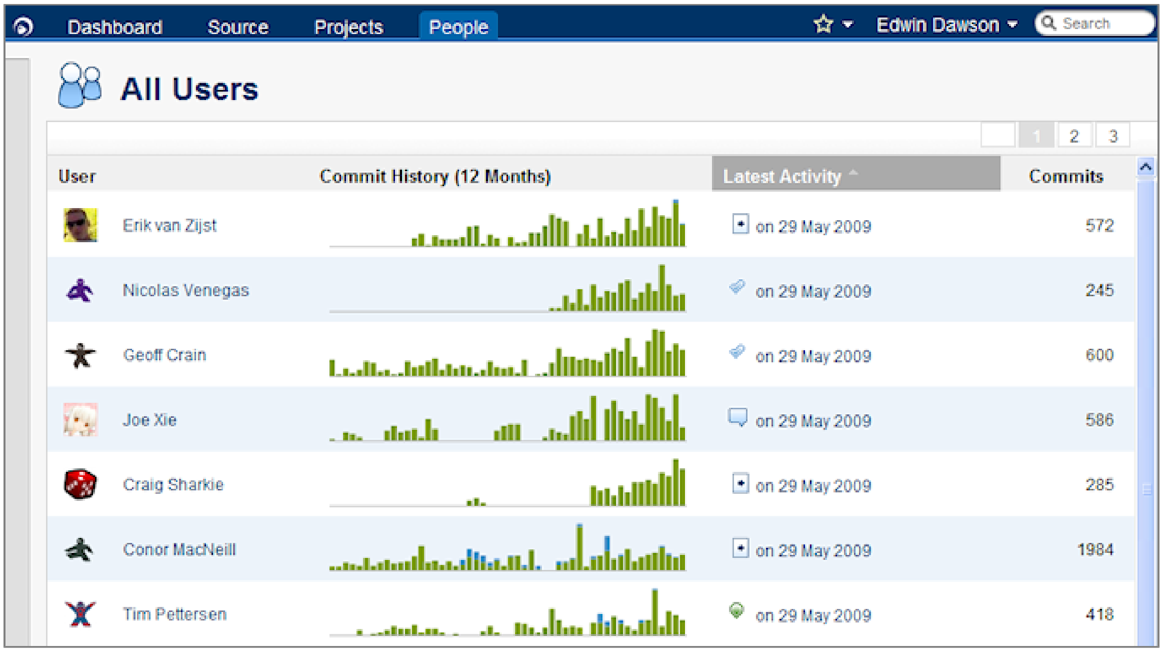
Author:	Nick Pellow
Revision:	23874
Changeset:	23874
Date:	17:42 26 November 2008 (6 months)
Branch point for:	clover-2.4.x clover-2.5.x clover-distributed fedex11-mstudman tag:clover-ant-2.4.2
Copied to:	branches/clover-2.4.x/trunk/tutorial/build.xml 31720
Tags:	clover-ant-2.4.2 idea7-2.4.2-v20081224113000 eclipse-2.4.2.v20090112202100 clover-ant-2.5.0 idea7-2.5.0.v20090511090000 eclipse-2.5.0.v20090511090000
Properties:	svn:eol-style = native svn:keywords = Author Date Id Revision

3

People View

You can now view detailed charts and activity statistics people who use your FishEye instance. You can compare number of commits charted over time and other activity in detail.

Screenshot: People View



All Users

User	Commit History (12 Months)	Latest Activity	Commits
Erik van Zijst		on 29 May 2009	572
Nicolas Venegas		on 29 May 2009	245
Geoff Crain		on 29 May 2009	600
Joe Xie		on 29 May 2009	586
Craig Sharkie		on 29 May 2009	285
Conor MacNeill		on 29 May 2009	1984
Tim Pettersen		on 29 May 2009	418

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:

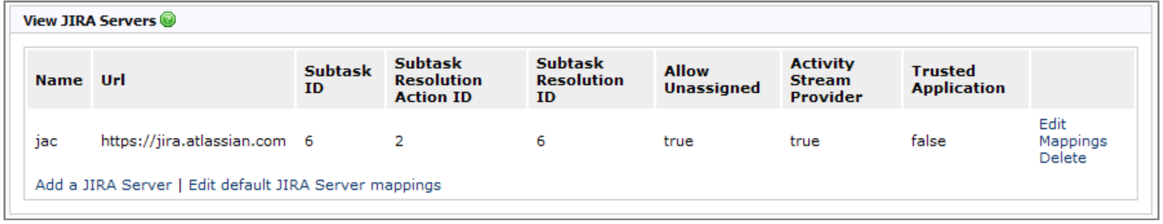
- Opening the Administration Screen for JIRA Integration
- Adding a New JIRA Server
- Editing Default JIRA Server Mappings
- Operations on Existing Servers
  - Edit settings for an existing JIRA server
  - Edit mappings for an existing JIRA server
  - Delete an existing JIRA server

JIRA issues can be viewed in the main Dashboard view in FishEye. This requires you to enter details on the required JIRA server(s) via the FishEye administration screens.

Opening the Administration Screen for JIRA Integration

To set up JIRA integration, open the Administration screen and then click '**JIRA Servers**' under the '**Global Settings**' sub-menu on the left navigation bar. The '**View JIRA Servers**' administration page opens.

Screenshot: The View JIRA Servers Page



View JIRA Servers

Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete

Add a JIRA Server | Edit default JIRA Server mappings

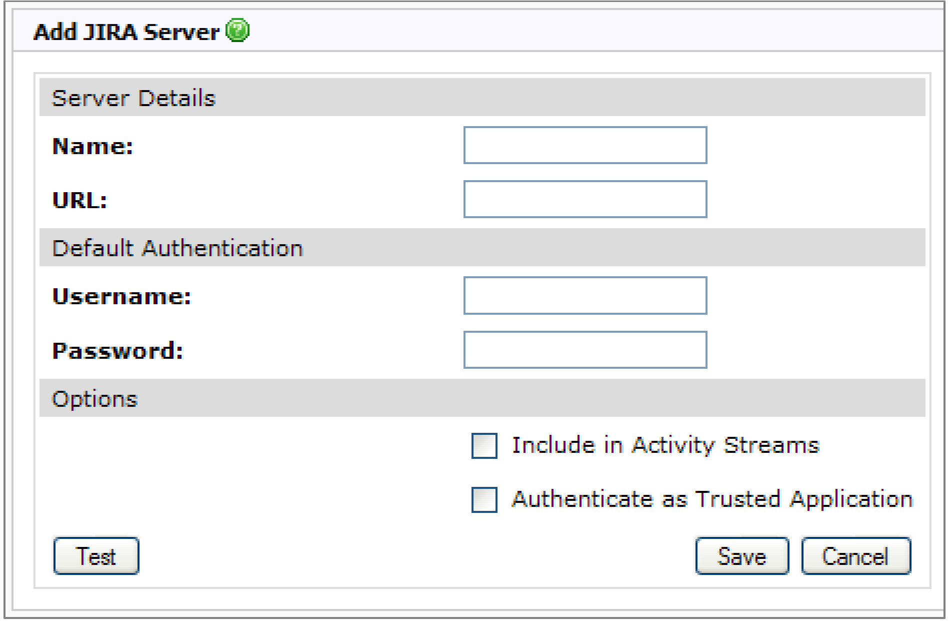
On the View JIRA Servers page, you can carry out a number of operations as listed on this page.

Adding a New JIRA Server

To add a new JIRA server from the View JIRA Servers page, click '**Add JIRA Server**'.

The '**Add JIRA Server**' page opens.

Screenshot: The Add JIRA Server Page



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

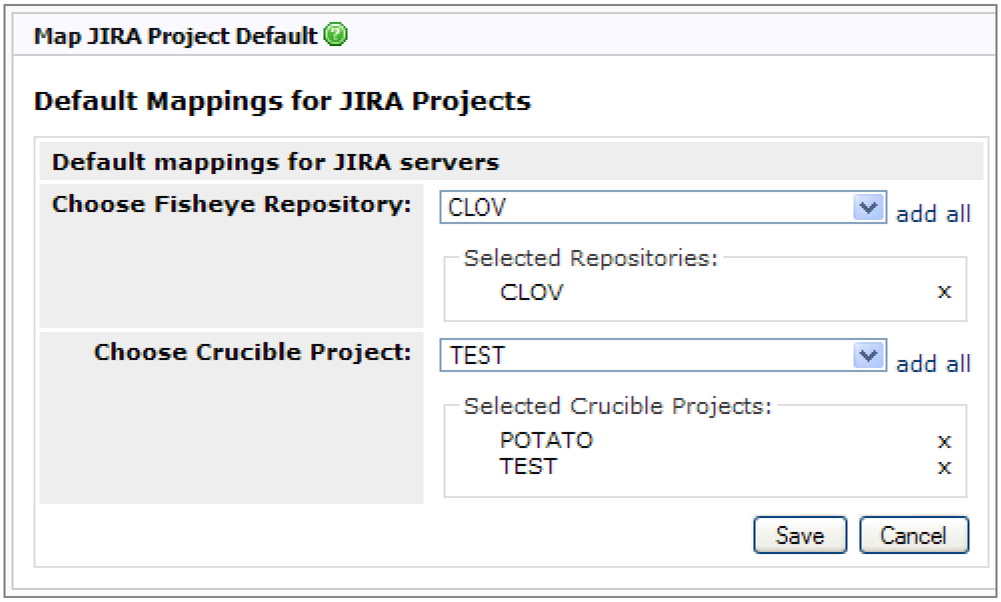
Once you've filled out the necessary fields, click '**Test**' to ensure that your details are correct. If you have a positive message return from the test, click '**Save**'.

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



Map JIRA Project Default

Default Mappings for JIRA Projects

Default mappings for JIRA servers

Choose Fisheye Repository: CLOV add all

Selected Repositories:
CLOV x

Choose Crucible Project: TEST add all

Selected Crucible Projects:
POTATO x
TEST x

On this page, select the FishEye repositories or Crucible Projects that you wish to associate with all the JIRA servers you have configured for use in FishEye. You can click '**add all**' to quickly include them all in this category. You can remove individual items by clicking the small 'X' marks.

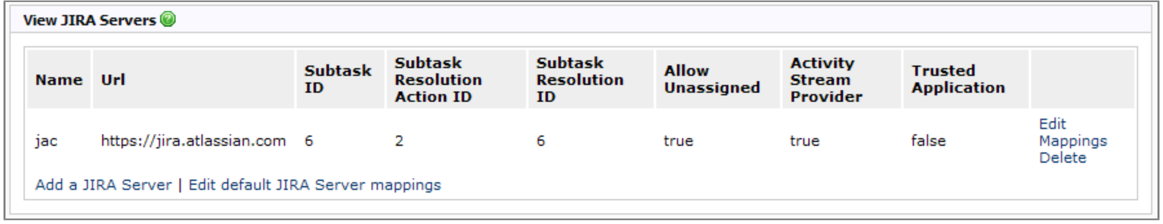
Once you've finished, click '**Save**'.

⚠ You should disable any existing FishEye linkers you have set up for JIRA, as they will override this feature and prevent the dynamic dialog box from appearing when you mouse over an issue.

Operations on Existing Servers

Once you have configured an existing JIRA server, there are three main operations you can carry out on it: '**Edit**', '**Mappings**' and '**Delete**'. These options appear on the far right of the screen.

Screenshot: Operations in the JIRA Servers Page



View JIRA Servers

Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete

Add a JIRA Server | Edit default JIRA Server mappings

Edit settings for an existing JIRA server

When you click '**Edit**', you can adjust any of the general settings you configured when you first added the server.

Edit mappings for an existing JIRA server

When you click '**Mappings**', a page is loaded that is almost identical to the '**Default Mapping**' screen, but allows you to choose mappings only for that specific JIRA server.

Delete an existing JIRA server

Clicking '**Delete**' will remove the server from the list.

Git Alpha in FishEye 2.0 Beta

FishEye 2.0 Beta is a public development release leading up to **FishEye 2.0**. For all production use and testing of FishEye, please use the latest official release.

> ⛔ Do not use in production.
> Beta releases should not be used in production environments.

Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing FishEye development process. As such:
    - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
    - Features in development releases may be incomplete, or may change or be removed before the next full release.

This page contains information about the alpha pre-release support for Git 1.6, which is activated (but not complete) in the FishEye 2.0 Beta.

Introduction and Disclaimer

Git Alpha support is included in this beta release for Git 1.6. Atlassian stresses that this is alpha level support which means the following:

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

23 September 2008

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

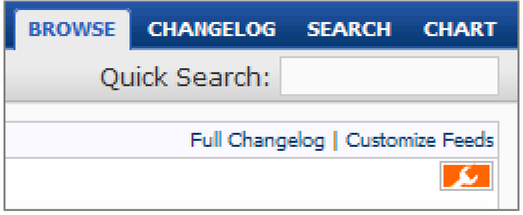
FishEye Search Enhancements

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

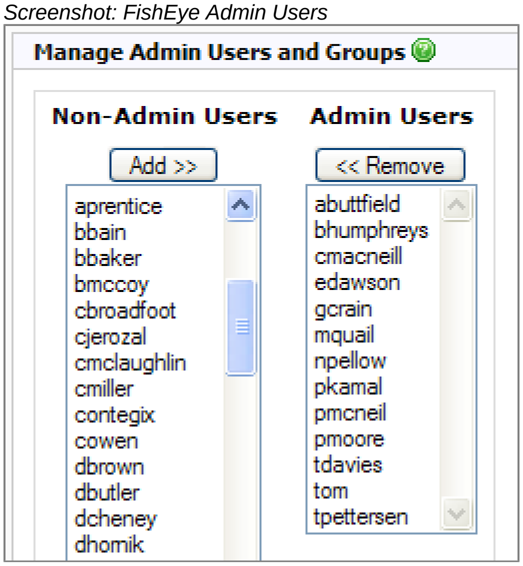
All of the improvements made to Quick Search and General Search are available when using Advanced Search and EyeQL.

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



Manage Admin Users and Groups

Non-Admin Users

Add >>

aprentice
bbain
bbaker
bmccoy
cbroadfoot
cjerozal
cmclaughlin
cmiller
contegix
cowen
dbrown
dbutler
dcheney
dhomik

Admin Users

<< Remove

abuttfield
bhumphreys
cmacneill
edawson
gcrain
mquail
npellow
pkamal
pmcneil
pmoore
tdavies
tom
tpettersen

3

Remote API Improvements

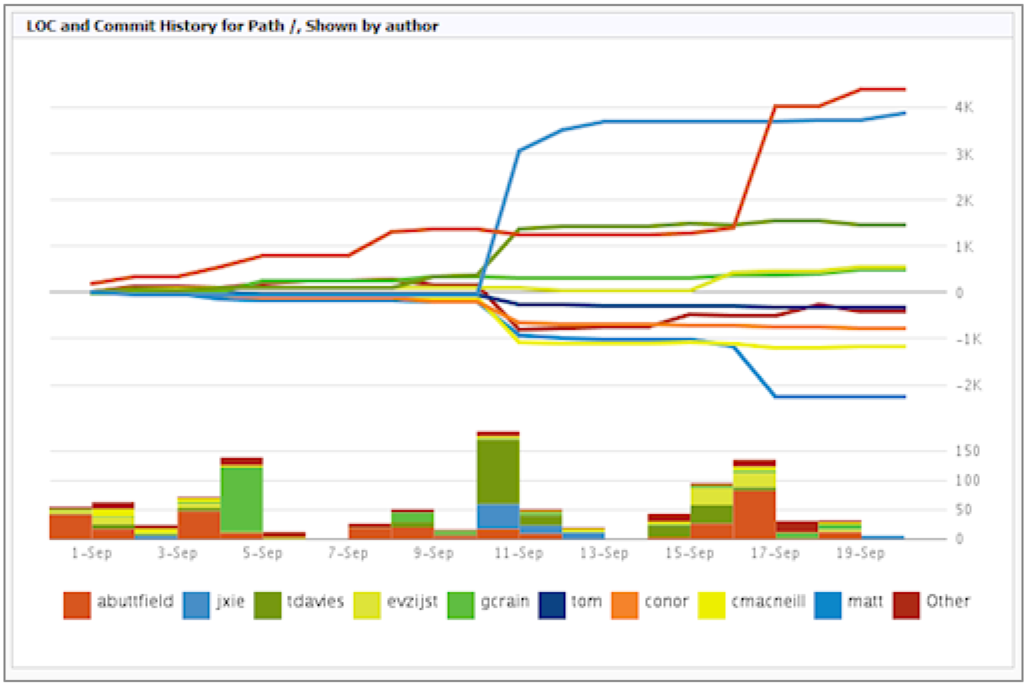
Enhancing the Remote API, the FishEye team have incorporated a **maxreturns** option, which lets you control the quantity of returned results. Also the range of functions is extended with the new **ancestor return** clause and **history search** items. Finally, Perforce data is now exposed in the remote 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-532	upgrade to webwork 2.2.7 (fixes security problem)	⬆	Closed
FE-500	Expose Changeset "Fixes Perforce Jobs" data in EyeQL (and REST api)	⬆	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-552	Have an error page rather than 403 page when SVN permission denied	⬆	Closed
FE-393	Use a single regularexpression to catch all tag/branch/trunk patterns	⬆	Closed
FE-296	Get Id button does not work in Trusted Application screen under IE	⬆	Closed
FE-554	Add LIMIT clause to EyeQL documentation	⬆	Closed
FE-559	Add 'ancestor' return clause to EyeQL	⬆	Closed
FE-541	Allow limiting of number of results returned by remote API	⬆	Closed
FE-578	"Search just <repo:parh>" breadcrumb links are borked - escapes parameter separators	⬆	Closed
FE-549	Search tokenizes on underscores	⬆	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-434	show match-in-context in quicksearch (hit highlighting)	⬆	Closed
FE-435	Quick-search redo UI requirements	⬆	Closed
FE-438	Do content searches in quicksearch	⬆	Closed
FE-338	Please add ability to specify initial revision from which to begin initial scan	⬆	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-504	command line reindex doens't work when loopback is not 127.0.0.1	⬆	Closed
FE-503	com.cenqua.fisheye.svn.SvnCache is throwing NPEs	⬆	Closed
FE-674	RSS Feed Title is missing space	⬆	Closed

FE-288	Repositories still occasionally get stuck in Stopping state	⬆	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-515	Allow P4 label scanning to be skipped	⬆	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-467	Calculating the correct version for diffs doesn't work for perforce	⬆	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-82	Re-index request: show message "Could not stop repository within 20 seconds. Re-index aborted."	⬆	Closed
FE-555	Documentation: Add maxReturn parameter to remote API	⬆	Closed
FE-491	Hide optional fields in repo setup	⬇	Closed
FE-495	Self Signup layout borked	⬇	Closed
FE-636	Make update polling configurable in admin section	⬇	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-428	Check for updates option	⬇	Closed
FE-482	SVN repositories default to UTF-8	⬇	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-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

On this page:

- From 1.6.5.a to 1.6.6
- From 1.6.4 to 1.6.5.a
- From 1.6.3 to 1.6.4
- From 1.6.2 to 1.6.3
- From 1.6.1 to 1.6.2
- From 1.6.0 to 1.6.1

From 1.6.5.a to 1.6.6

10th February 2009

This release is a bugfix release which address the following issues:

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?	↑	Closed
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 1.5 Release Notes

15 April 2008

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

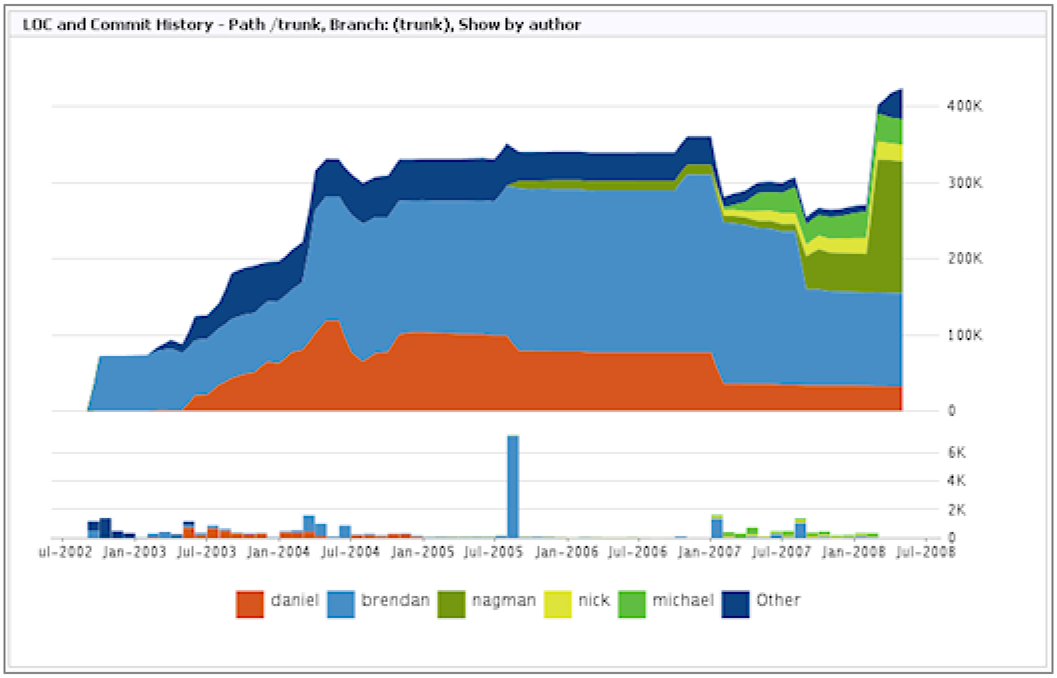
1

Per-author lines of code statistics

Statistics for lines of code is now broken down per-author, providing an all-new level of detail. This allows you to see how many lines of code were contributed to your project by each author, over time.

⚠ This requires changing a setting and re-scanning existing repositories. See the FishEye Upgrade Guide for more information.

Screenshot: FishEye Per-Author Line Count Chart

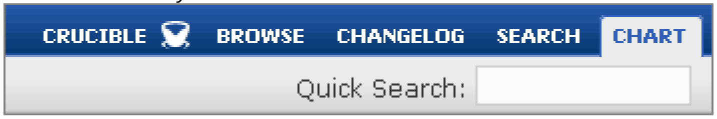


2

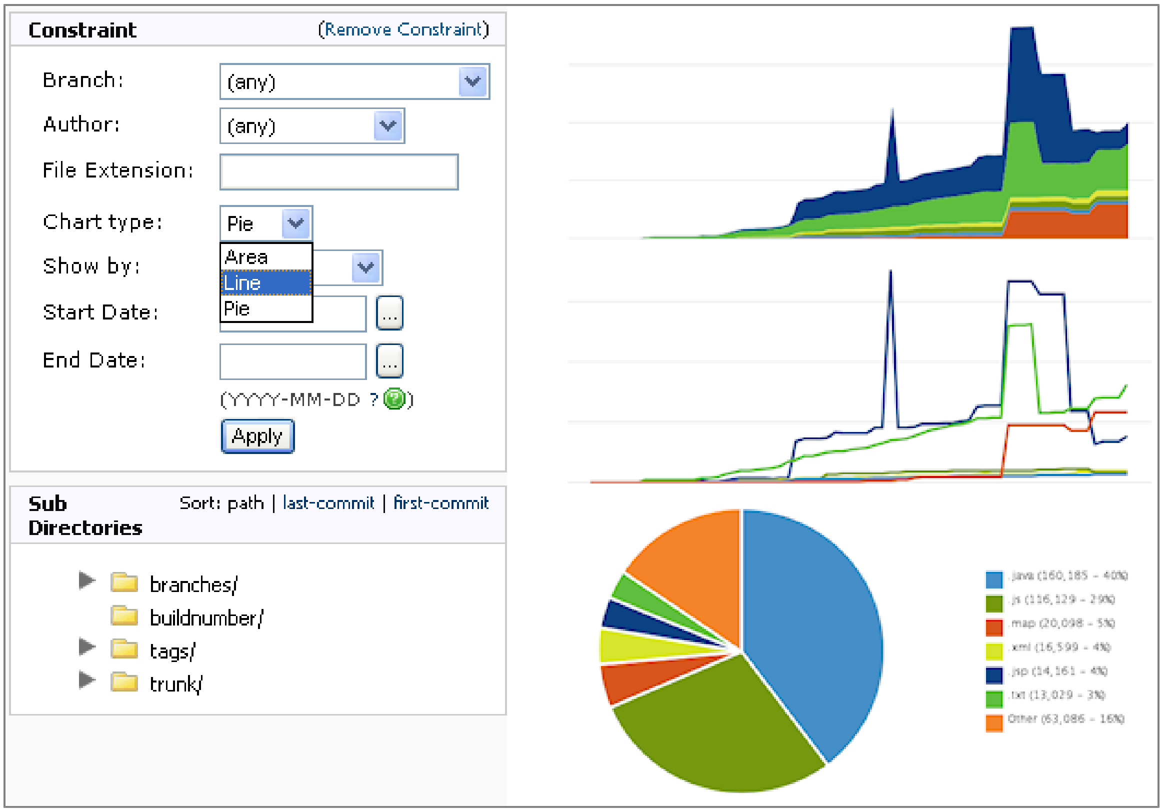
Charting improvements

The line graphs in FishEye have been improved, providing a better view of lines of code statistics from your project, as well as showing how this has grown.

Screenshot: FishEye Charts Tab



Screenshot: FishEye Chart Examples



Constraint (Remove Constraint)

Branch: (any)

Author: (any)

File Extension:

Chart type: Pie

Show by:

Start Date:

End Date:

(YYYY-MM-DD ?)

🟦	.java (160,185 - 40%)
🟩	.js (116,129 - 29%)
🟥	.map (20,098 - 5%)
🟨	.xml (16,599 - 4%)
🟦	.jsp (14,161 - 4%)
🟩	.txt (13,029 - 3%)
🟧	Other (63,086 - 16%)

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

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	↑	Closed
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 1.4 Release Notes

December 5, 2007.

The Atlassian FishEye team is delighted to present FishEye 1.4.
FishEye 1.4 is a major release which focuses on integration, user management and performance.

The updated FishEye Administration interface provides support for groups and improved user management screens. The new built-in integration with Atlassian Crowd extends your authentication and authorisation capabilities. You can now include users and groups from one or more Crowd directories, and provide single sign-on (SSO) across Atlassian products plus any other applications that support SSO.

The latest releases of FishEye and Crucible work together even more tightly than before, allowing you to see at a glance which files/changesets have been reviewed. You can also search within FishEye for files that have not yet been reviewed.

Highlights of this release:

- Enhancements to user management
- Crowd/SSO support
- Crucible integration
- Enhancements to JIRA plugin
- Plus over 30 improvements and bug-fixes

Responding to your feedback:

19 new feature requests/improvements implemented

Your [votes and issues|http://jira.atlassian.com/browse/FE] help us keep improving our products, and are much appreciated.

Upgrading to FishEye 1.4
You can now download FishEye from here. If upgrading from a previous version, please follow the Upgrade Guide.

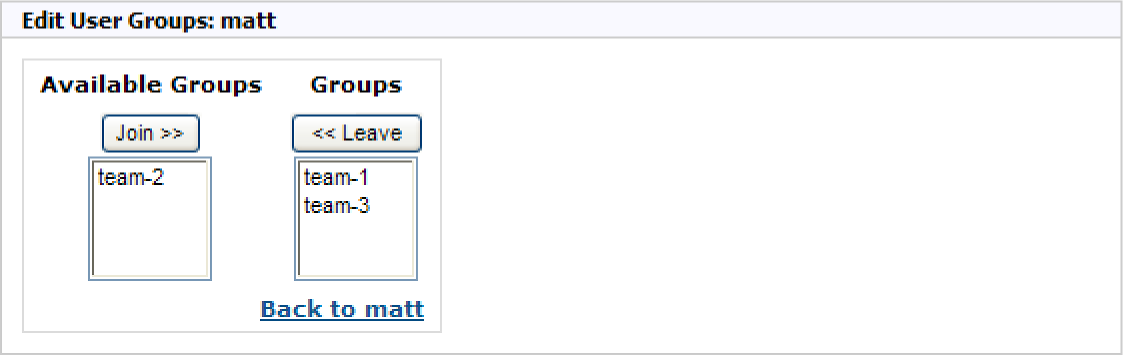
Highlights of FishEye 1.4

1

Enhancements to user management

In FishEye 1.3.7, we introduced support for public signup (self-registration). Now in release 1.4:

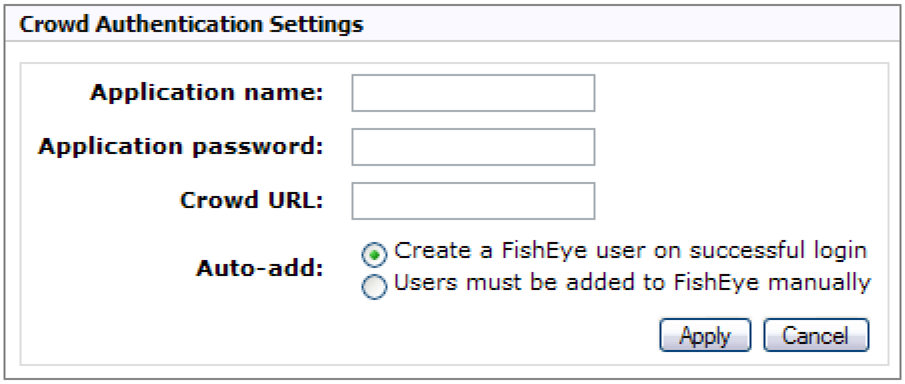
- Improved user interface makes user administration easier.
- Groups are supported.
- Renaming of users is supported.
- Read the documentation.


Edit User Groups: matt

2

Crowd/SSO support

- Inbuilt integration with Atlassian Crowd for authentication and authorisation.
- Users and groups in your Crowd directories now supported in FishEye.
- Single sign on (SSO) support via Crowd e.g. you can now sign in just once to access Atlassian JIRA, FishEye, Crucible, Confluence and Bamboo, and any other applications which support SSO.
- Read the documentation.



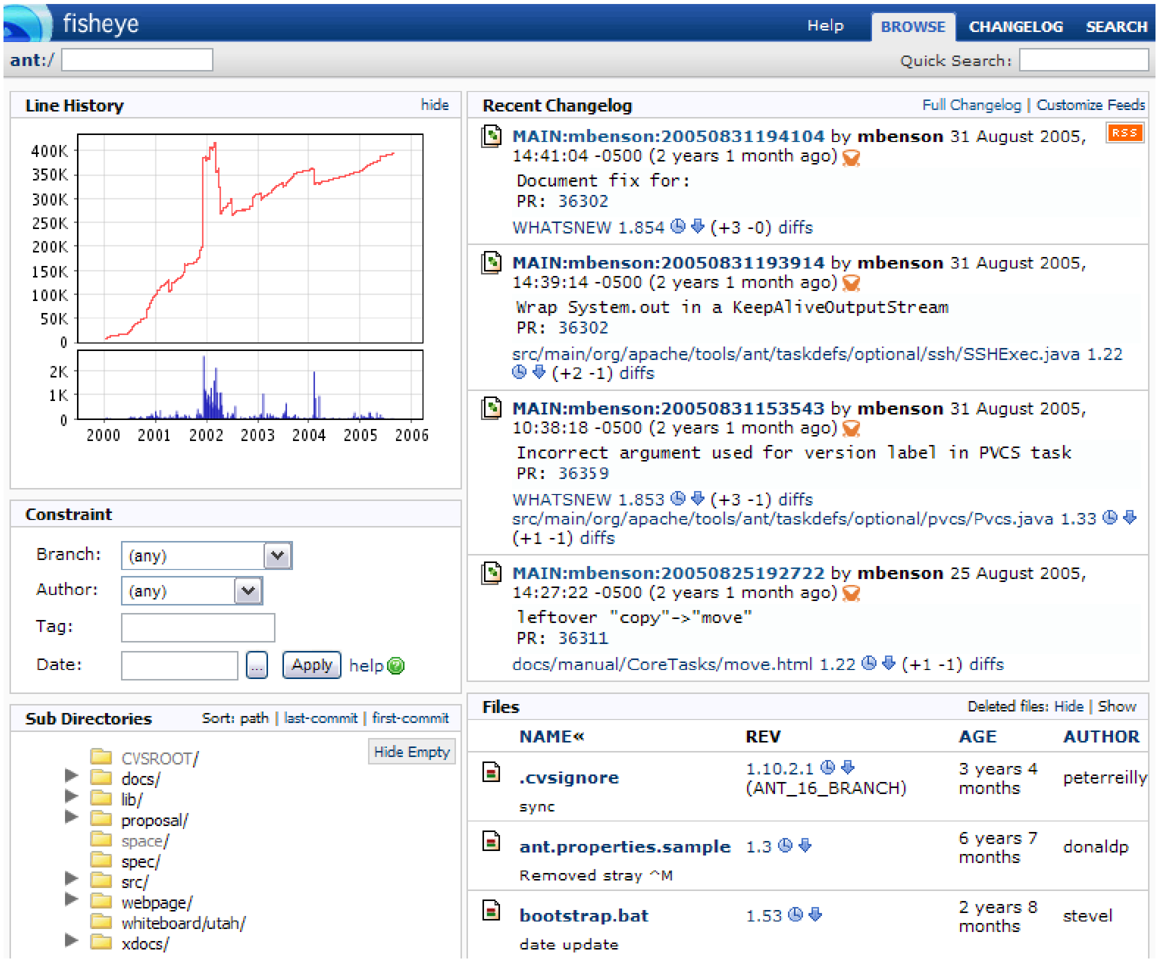
Crowd Authentication Settings

Application name:	
Application password:	
Crowd URL:	
Auto-add:	● Create a FishEye user on successful login ○ Users must be added to FishEye manually

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.



Constraint

Branch: (any)
Author: (any)
Tag:
Date:

Sub Directories Sort: path | last-commit | first-commit

CVSROOT/
docs/
lib/
proposal/
space/
spec/
src/
webpage/
whiteboard/utah/
xdocs/

Recent Changelog Full Changelog | Customize Feeds

MAIN:mbenson:20050831194104 by **mbenson** 31 August 2005, 14:41:04 -0500 (2 years 1 month ago)

```
Document fix for:
PR: 36302
```

WHATSNEW 1.854 (+3 -0) diffs

MAIN:mbenson:20050831193914 by **mbenson** 31 August 2005, 14:39:14 -0500 (2 years 1 month ago)

```
Wrap System.out in a KeepAliveOutputStream
PR: 36302
```

src/main/org/apache/tools/ant/taskdefs/optional/ssh/SSHExec.java 1.22 (+2 -1) diffs

MAIN:mbenson:20050831153543 by **mbenson** 31 August 2005, 10:38:18 -0500 (2 years 1 month ago)

```
Incorrect argument used for version label in PVCS task
PR: 36359
```

WHATSNEW 1.853 (+3 -1) diffs
src/main/org/apache/tools/ant/taskdefs/optional/pvcs/Pvcs.java 1.33 (+1 -1) diffs

MAIN:mbenson:20050825192722 by **mbenson** 25 August 2005, 14:27:22 -0500 (2 years 1 month ago)

```
leftover "copy"->"move"
PR: 36311
```

docs/manual/CoreTasks/move.html 1.22 (+1 -1) diffs

Files Deleted files: Hide | Show

NAME«	REV	AGE	AUTHOR
.cvsignore sync	1.10.2.1 (ANT_16_BRANCH)	3 years 4 months	peterreilly
ant.properties.sample Removed stray ^M	1.3	6 years 7 months	donaldp
bootstrap.bat date update	1.53	2 years 8 months	stevel

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

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 1.3 Release Notes

> ⚠ FishEye 2.3 has now been released. Read the Release Notes.

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

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 Release Summary

FishEye 2.3 (26-May-10)

- Mercurial SCM Alpha
- New 'Aggregate' functions in EyeQL query language

- Revamped Installation Process
- More in release notes.

FishEye 2.2 (18-Feb-10)

- 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-Nov-09)

- 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-Jun-09)

- Activity streams
- People statistics
- Favourites, bookmarks & saved search
- Enhanced JIRA integration
- New user interface
- Git beta
- More in release notes.

FishEye 1.6 (23-Sep-08)

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

FishEye 1.5 (14-Apr-08)

- Per-author lines of code statistics
- Charting improvements
- Customisable email templates
- More in release notes.

FishEye 1.4 (5-Dec-07)

- Enhancements to user management
- Crowd/SSO support
- Crucible integration
- Enhancements to JIRA plugin
- More in release notes.

FishEye 1.3 (1-Aug-07)

- 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 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.

> ℹ Note: If you are an Atlassian JIRA Studio customer, we have assessed that your system is secure and implemented additional protections for it.

Fix

These issues have been fixed in FishEye 2.2.3 (see the changelog), which you can download from the download centre. Later versions will include protection from this vulnerability.

This fix is also provided as a patch for FishEye 2.1.4, 2.0.6 and 1.6.6, which you can download from this page. Customers on earlier point versions of FishEye will have to upgrade to version 2.1.4, 2.0.6 or 1.6.6 before applying the patch. We recommend you upgrade to FishEye 2.2.3.

XSS Vulnerabilities in FishEye

Severity

Atlassian rates these vulnerabilities as **critical**, according to the scale published in Severity Levels for Security Issues. The scale allows us to rank a vulnerability as critical, high, moderate or low.

Risk Assessment

We have identified and fixed several cross-site scripting (XSS) vulnerabilities in FishEye, which may affect FishEye instances. These vulnerabilities have security implications and are especially important for anyone running publicly accessible instances of FishEye.

- The attacker might take advantage of the vulnerability to steal other users' session cookies or other credentials, by sending the credentials back to the attacker's own web server.
- The attacker's text and script might be displayed to other people viewing a FishEye page. This is potentially damaging to your company's reputation.

You can read more about XSS attacks at cgisecurity, CERT and other places on the web.

Vulnerability

All versions of FishEye are affected by these XSS vulnerabilities.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.2.1	2.2.3 only	An attacker could take advantage of this vulnerability to steal other users' session cookies or other credentials, or the attacker's text and script might be displayed to other people viewing a FishEye page.	Critical

Risk Mitigation

We strongly recommend upgrading your FishEye installation to fix these vulnerabilities. Please see the 'Fix' section below.

Fix

These issues have been fixed in FishEye 2.2.3 (see the changelog), which you can download from the download centre.

Prevention of Brute Force Attacks

Severity

Atlassian rates this vulnerability as **moderate**, according to the scale published in Severity Levels for Security Issues.

Risk Assessment

We have improved the security of the following areas in FishEye:

- Prevention of brute force attacks by requiring users to solve a CAPTCHA test after a maximum number of repeated login attempts.

Vulnerability

We have identified and fixed a problem where FishEye allows an unlimited number of repeated login attempts, potentially opening FishEye to a brute force attack. Details of this improvement are summarised below.

Affected FishEye Versions	Fix Availability	More Details	Severity
All versions up to and including 2.2.1	2.2.3 only	FishEye allows an unlimited number of login attempts. This makes FishEye vulnerable to a brute force attack.	Moderate

Risk Mitigation

We recommend that you upgrade your FishEye installation to fix these vulnerabilities. Please see the 'fix' section below.

You can also prevent brute force attacks by following our guidelines on using Fail2Ban to limit login attempts.

Fix

This issue has been fixed in FishEye 2.2.3 (see the changelog). Later versions will include protection from this vulnerability. You can download FishEye 2.2.3 from the download centre.

Changed Behaviour in FishEye

In order to fix these issues, we have changed FishEye's behaviour as follows:

- After three consecutive failed login attempts, FishEye will display a CAPTCHA form asking the user to enter a given word when attempting to log in again. This will prevent brute force attacks via the login screen. The number of failed attempts needed to trigger the CAPTCHA testing is configurable. For more information, see the documentation for Brute Force Login Protection.

> ⚠ In addition, after three consecutive failed login attempts via the FishEye remote API, an error message will be returned. Human intervention will then be required to reset that login account, i.e. solve the CAPTCHA test via the login screen.

Download Patches for Earlier FishEye / Crucible Versions

> ℹ 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

ℹ 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 FAQ

FishEye FAQ

Answers to frequently asked questions about configuring and using FishEye.

- Top Evaluator Questions
  - Does FishEye support GIT?
  - How do I fix problems with indexing my repository?
  - How do I setup JIRA integration?
  - How do I setup LDAP or external user management?
  - How do I speed up slow CVS updates?
  - How do I start FishEye as a Windows service?
  - How do I view changesets and diffs?
  - How is FishEye licensed?
  - What kind of search capabilities does FishEye have?
  - What progamming languages are supported?

- CVS FAQ
  - How does FishEye calculate CVS changesets?
- Example EyeQL Queries
  - How do find changes made to a branch after a given tag?
  - How do I filter results?
  - How do I find changes between two versions, showing separate histories?
  - How do I find changes made between two version numbers?
  - How do I find commits without comments?
  - How do I find files on a branch, excluding deleted files?
  - How do I find files removed from a given branch?
  - How do I find revisions made by one author between versions?
  - How do I select the most recent revisions in a given branch?
  - How do I show all changesets which do not have reviews?
- FishEye Developer FAQ
- General FAQ
  - Automating Administrative Actions in Fisheye
  - About the Lines of Code metric
  - How Do I Archive a Branch within Perforce
  - How do I avoid long reindex times when I upgrade?
  - Mercurial Known Issues
  - What SCM systems are supported by FishEye?
- Installation & Configuration FAQ
  - How to reset the Admin password in Fisheye or Crucible
  - How Do I Confugure an Outbound Proxy Server for FishEye
  - How to Remove Crucible From FishEye 2.x or Later
  - How to run Fisheye or Crucible on startup on Mac OS X
  - Can FishEye be run as a Windows service?
  - Can I deploy FishEye or Crucible as a WAR?
  - Configuring Web Proxy Support for FishEye or Crucible
  - Improve FishEye Scan Performance
  - Migrating FishEye Between Servers
  - Setting Up a CVS Mirror with rsync
  - What are the FishEye System Requirements?
- Licensing FAQ
  - Are anonymous users counted towards FishEye's licence limits?
  - What are the Starter License restrictions?
- Subversion FAQ
  - Errors 'SEVERE assert' or 'Checksum mismatch'
  - FishEye fails to connect to the Subversion repository after a short time of successful operation.
  - How can FishEye help with merging of branches in Subversion?
  - SVN Authentication Issues
  - Why do I need to describe the branch and tag structure for Subversion repositories?
  - Why don't all my tags show up in FishEye?
- Support Policies
  - Bug Fixing Policy
  - How to Report a Security Issue
  - New Features Policy
  - Patch Policy
  - Security Advisory Publishing Policy
  - Security Patch Policy
  - Severity Levels for Security Issues
- Troubleshooting
  - After I commit a change to my CVS repository, it takes a long time before it appears in FishEye.
  - FishEye freezes unexpectedly
  - Fix Out of Memory errors by increasing available memory
  - Generating a Thread Dump Externally
  - I have installed FishEye, and the inital scan is taking a long time. Is this normal?
  - I have installed FishEye, but there is no data in the Changelog.
  - Initial scan and page loads are slow on Subversion
  - It seems that FishEye's HTTP Header is Too Small
  - JIRA Integration Issues
  - Message 'org.tigris.subversion.javahl.ClientException svn Java heap space'
  - On my Red Hat Linux system, after running for several days FishEye freezes and does not accept any more connections.
  - Problems with very long comments and MySQL migration
  - URLs with encoded slashes don't work, especially in Author constraints

Do you have a question, or need help with FishEye? Please create a support request.

CVS FAQ

FishEye CVS FAQ

- How does FishEye calculate CVS changesets? — FishEye's goal is to allow changesets to be seen as a consistent stream of atomic commits. Revisions are collated into the same changeset provided that:

How does FishEye calculate CVS changesets?

FishEye's goal is to allow changesets to be seen as a consistent stream of atomic commits. Revisions are collated into the same changeset provided that:

- They have the same commit comment.
- They are by the same author.
- They are on the same branch.
- The changeset does not span more than 10 minutes.
- The same file does not appear in a changeset more than once.

Example EyeQL Queries

EyeQL

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

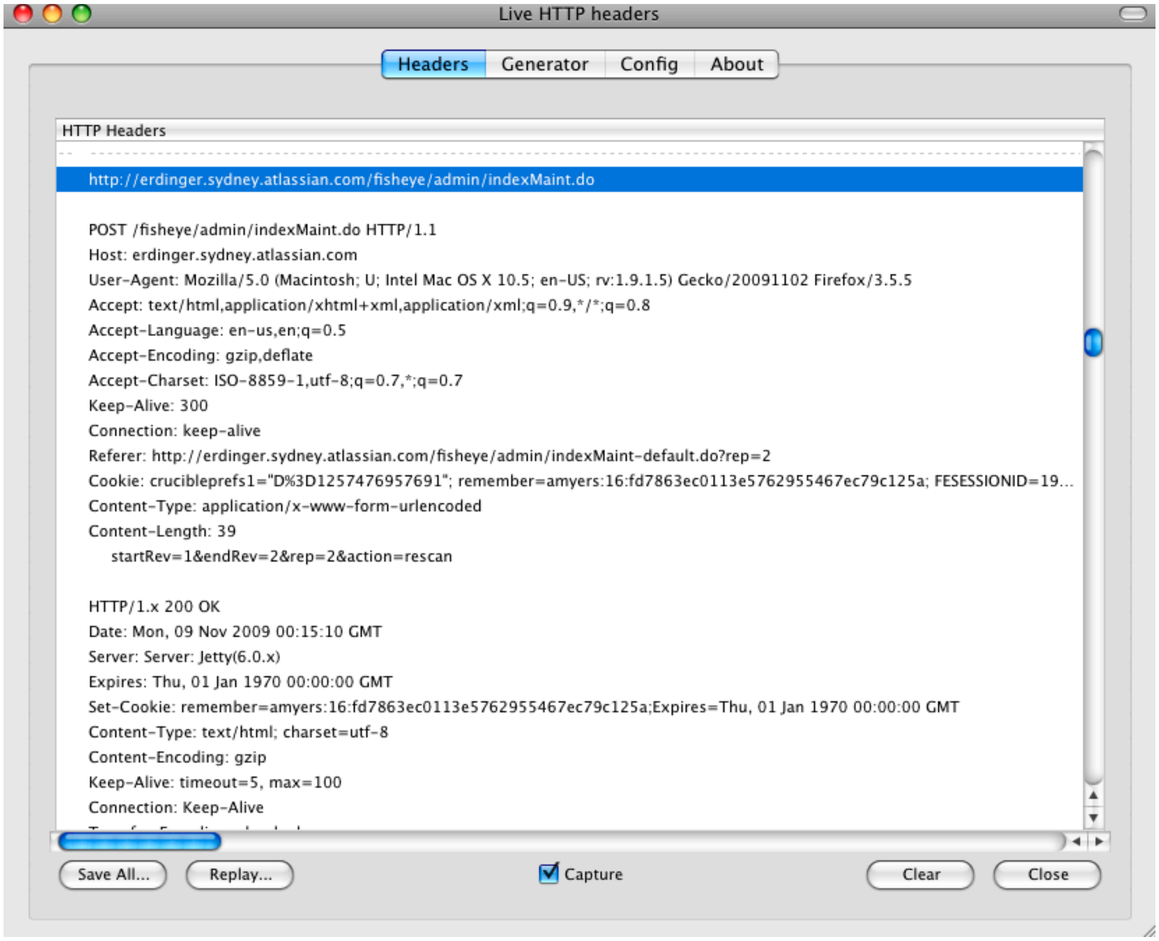
FishEye General FAQ

- Automating Administrative Actions in Fisheye
- About the Lines of Code metric
- How Do I Archive a Branch within Perforce
- 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.
- Mercurial Known Issues
- 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:



```
http://erdinger.sydney.atlassian.com/fisheye/admin/indexMaint.do

POST /fisheye/admin/indexMaint.do HTTP/1.1
Host: erdinger.sydney.atlassian.com
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://erdinger.sydney.atlassian.com/fisheye/admin/indexMaint-default.do?rep=2
Cookie: crucibleprefs1="D%3D1257476957691"; remember=amyers:16:fd7863ec0113e5762955467ec79c125a; FESESSIONID=19...
Content-Type: application/x-www-form-urlencoded
Content-Length: 39
  startRev=1&endRev=2&rep=2&action=rescan

HTTP/1.x 200 OK
Date: Mon, 09 Nov 2009 00:15:10 GMT
Server: Server: Jetty(6.0.x)
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Set-Cookie: remember=amyers:16:fd7863ec0113e5762955467ec79c125a;Expires=Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

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 the Lines of Code metric

This page contains information about the Lines of Code metric and how it is processed and represented by FishEye.

On this page:

- Definition
- Disadvantages
- LOC in Fisheye
- User-Specific LOC

Definition

Lines of Code or LOC (also known as Source Lines of Code - SLOC) is a quantitative measurement in computer programming for files that contains code from a computer programming language, in text form. The number of lines indicates the size of a given file and gives some indication of the work involved.

LOC is literally the count of the number of lines of text is 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.

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.
ℹ 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.
4. 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.
5. 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.
ℹ 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.

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
- How Do I Confugure an Outbound Proxy Server for FishEye
- How to Remove Crucible From FishEye 2.x or Later
- How to run Fisheye or Crucible on startup on Mac OS X
- Can FishEye be run as a Windows service? — FishEye can be run as a service under Microsoft Windows.
- Can I deploy FishEye or Crucible as a WAR? — Unfortunately FishEye and Crucible can not be deployed as a WAR.
- Configuring Web Proxy Support for FishEye or Crucible
- Improve FishEye Scan Performance — This page contains information about improving the performance of FishEye repository scans.
- Migrating FishEye Between Servers — This page describes the process for migrating FishEye between servers.
- Setting Up a CVS Mirror with rsync — In situations where running FishEye on the same server as your CVS repository is not practical or possible, you can use the Linux utility **rsync** to mirror the CVS repository contents onto the FishEye server.
- What are the FishEye System Requirements?

How to reset the 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:

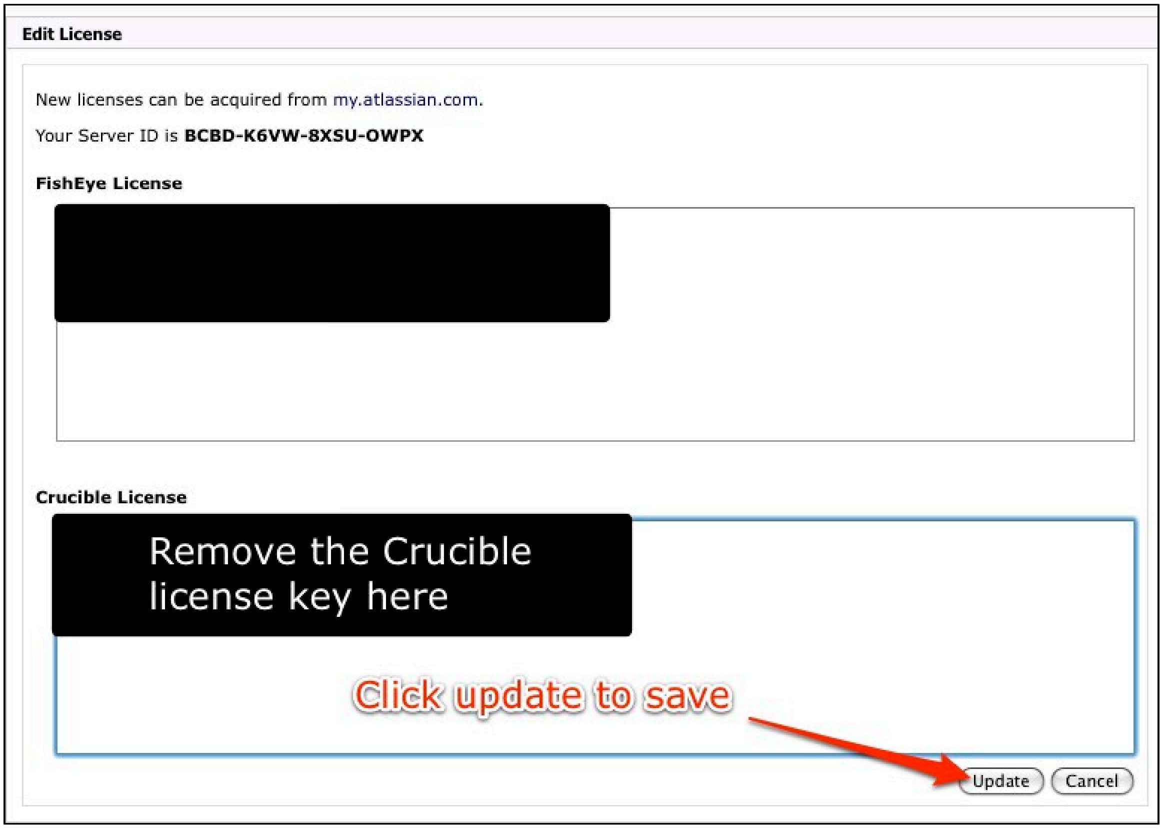
```
export FISHEYE_OPTS="-Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=8080
-Dhttp.nonProxyHosts=*.foo.com|localhost -Dhttp.proxyUser=username -Dhttp.proxyPassword=password"
```

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



Edit License

New licenses can be acquired from my.atlassian.com.

Your Server ID is **BCBD-K6VW-8XSU-OWPX**

FishEye License

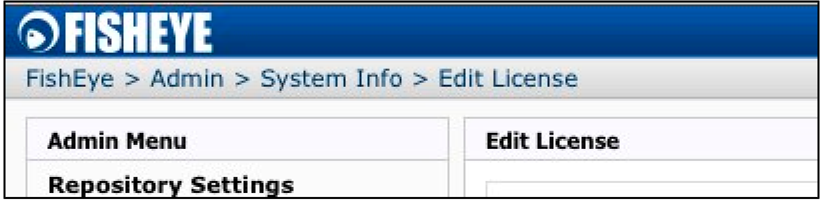
[redacted]

Crucible License

Remove the Crucible license key here

Click update to save

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).

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 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.

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.

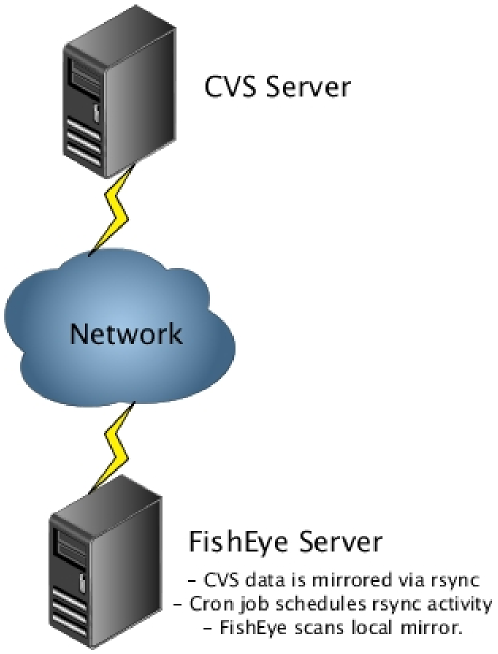
Setting Up a CVS Mirror with rsync

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


CVS Server
Network
FishEye Server
- CVS data is mirrored via rsync
- Cron job schedules rsync activity
- FishEye scans local mirror.

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.

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.
- What are the Starter License restrictions? — FishEye Starter Licenses are restricted to ten users, ten committers per repository, as well as a total of five repositories.

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 ten users, ten committers per repository, as well as a total of five repositories.

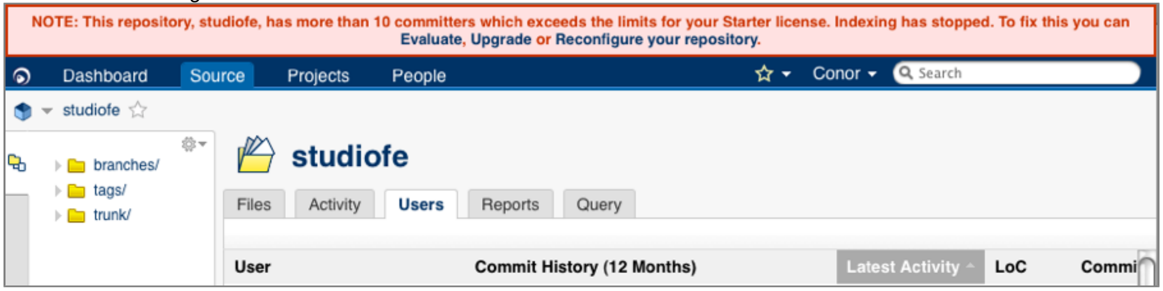
What happens if I exceed the Starter License limits?

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



NOTE: This repository, studiofe, has more than 10 committers which exceeds the limits for your Starter license. Indexing has stopped. To fix this you can Evaluate, Upgrade or Reconfigure your repository.

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.

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 inthe 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

- 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.
- 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.
- 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?

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.

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.

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?

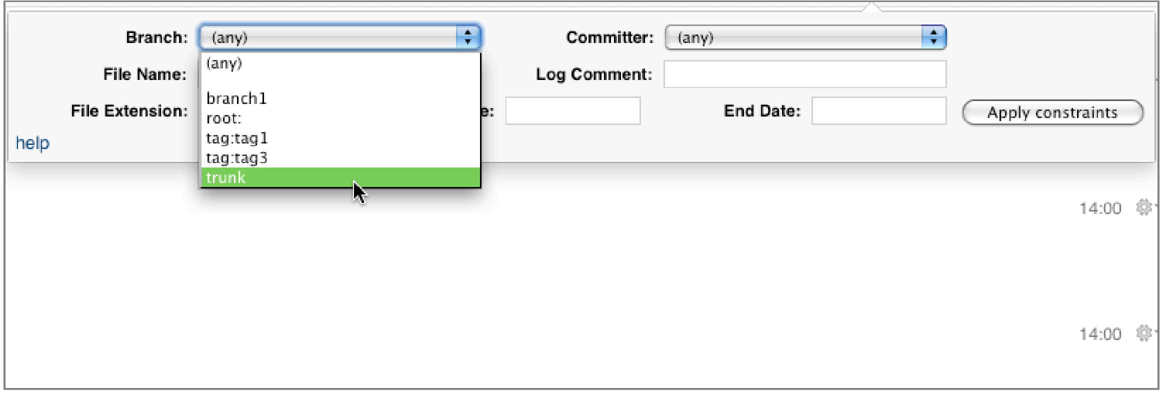
This page gives a detailed technical explanation of why certain issues affect Subversion users.

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.

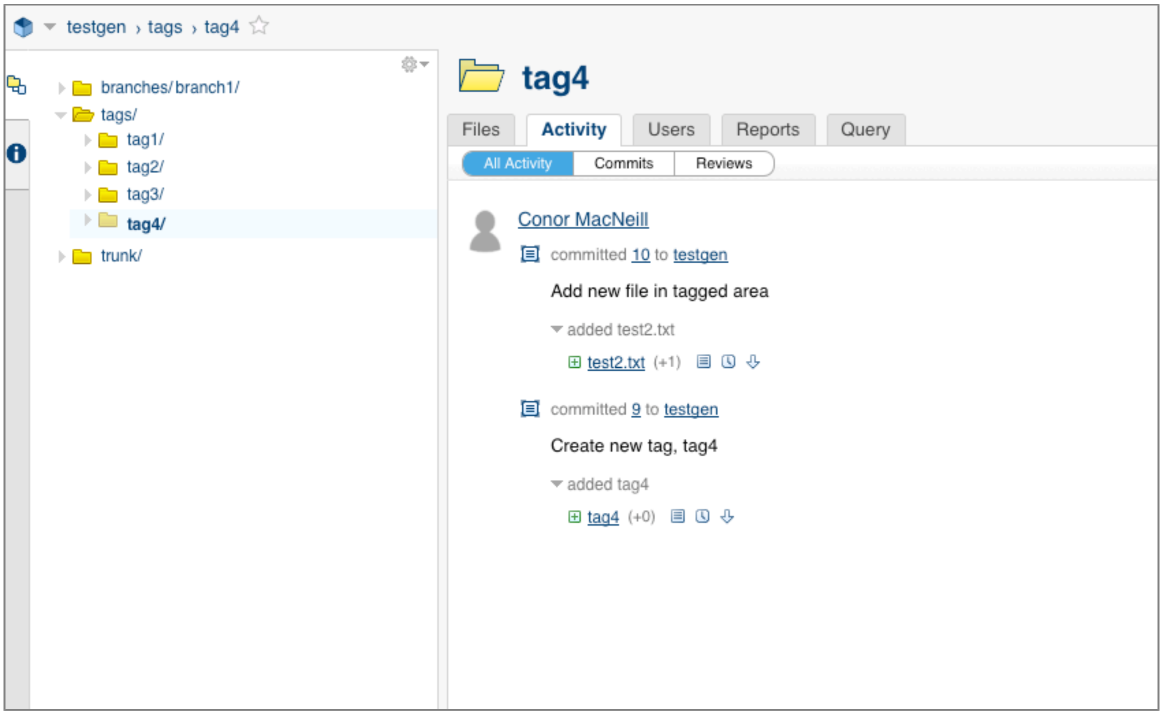
For regular changes in your Subversion repository, FishEye records each change against a branch where the change took place. If, however, after tagging, you make a change to a file in the tagged area, you are making a change outside trunk or a recognized branch. FishEye records such changes by creating 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



Conor MacNeill

committed 10 to testgen

Add new file in tagged area

added test2.txt

test2.txt (+1)

committed 9 to testgen

Create new tag, tag4

added tag4

tag4 (+0)

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

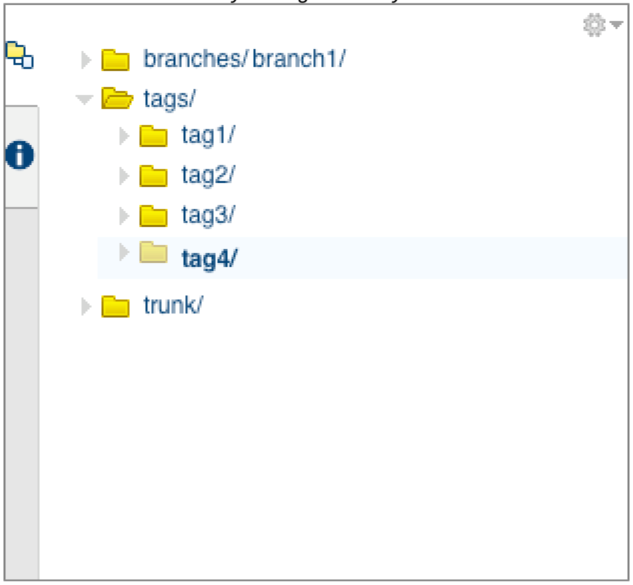
In general, it's not good practice to make changes in the tag areas of a Subversion repository. Such changes can easily get lost if they are not applied to trunk or a current branch. It is preferable to make the change in trunk or a branch and then create a new tag to capture the update. Nevertheless, 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.

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 (e.g. JIRA 4.0) and the last maintenance release of the last major version (e.g. JIRA 3.13.5) 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 (ie it doesn't require a major architectural change)
  OR
- The issue is a security issue, and falls under our Security 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 issue in an Atlassian product is discovered and resolved, Atlassian will inform customers through the following mechanisms:

- A security advisory will be posted in the documentation.
- A copy of the advisory will be sent to the product mailing-lists. These lists are mirrored on our forums.
- If the person who reported the issue wants to publish an advisory through some other agency (for example, CERT), Atlassian will assist in the production of that advisory, and link to it from our own.

Further reading

See How to Get Legendary Support from Atlassian for more support-related information.

Security Patch Policy

Our Security Patch Policy

When a security issue is discovered, Atlassian will endeavour to:

- Issue a new, fixed version as soon as possible,
- Issue a patch to the current release (e.g. JIRA 4.0) and the latest maintenance release for the last major version of a product (e.g. JIRA 3.13.5),
- Issue patches for older versions if feasible.

Patches will generally be attached to the relevant JIRA issue.

Visit our general Atlassian Patch Policy as well.

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, rating the vulnerability as one of the following:

- Critical
- High
- Moderate
- Low

Below is a summary of the factors which we use to decide on the severity level, and the implications for your installation.

Severity Level: Critical

We classify a vulnerability as critical if most or all of the following are true:

- 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.

Severity Level: High

We give a high severity level to those vulnerabilities which have the potential to become critical, but have one or more mitigating factors that make exploitation less attractive to attackers.

For example, given a vulnerability which has many characteristics of the critical severity level, we would give it a level of high if any of the following are true:

- The vulnerability is difficult to exploit.
- Exploitation does not result in elevated privileges.
- The pool of potential victims is very small.

Note: If the mitigating factor arises from a lack of technical details, the severity level would be elevated to critical if those details later became available. If your installation is mission-critical, you may want to treat this as a critical vulnerability.

Severity Level: Moderate

We give a moderate severity level to those vulnerabilities where the scales are slightly tipped in favour of the potential victim.

The following vulnerabilities are typically rated moderate:

- Denial of service vulnerabilities, since they do not result in compromise of a target.
- 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

We give a low severity level to those vulnerabilities which by themselves have typically very little impact on an organisation's infrastructure.

Exploitation of such vulnerabilities usually requires local or physical system access. Exploitation may result in client-side privacy or denial of service issues and leakage of information about organisational structure, system configuration and versions, or network topology.

> **Original ranking compiled by the SANS Institute**
> Our vulnerability ranking is based on a scale originally published by the SANS Institute.

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.
- Fix Out of Memory errors by increasing available memory — Since the default memory setting usually is around 64MB or 128MB, you might have to adjust the settings to run a bigger FishEye instance with sufficient memory.
- 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.

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.

Fix Out of Memory errors by increasing available memory

I am getting Out of Memory errors, how can I allocate more memory to FishEye?

Since the default memory setting usually is around 64MB or 128MB, you might have to adjust the settings to run a bigger FishEye instance with sufficient memory.
On this page:

- Out Of Memory Errors
  - OutOfMemoryError: Java Heap Space
  - OutOfMemoryError: PermGen space, or Permanent Generation Size
  - OutOfMemoryError: unable to create new native thread
  - OutOfMemoryError: GC overhead limit exceeded
  - java.lang.OutOfMemoryError: requested 32756 bytes for ChunkPool::allocate. Out of swap space?

Out Of Memory Errors

There are a number of different memory errors that the JVM will throw. The most common are listed as follows.
In the following, you will be required to set your memory settings via your FISHEYE_OPTS Environment Variables.
You will need to restart your server after setting your FISHEYE_OPTS.

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.

OutOfMemoryError: Java Heap Space

> ⚠ If you are running Fisheye/Crucible as a windows service, increasing memory needs to be done in the wrapper.conf file. Refere to the Can Fisheye be run as a Windows Service for instructions.

To solve this error, you will need to add the argument `-Xmx1024m` to FISHEYE_OPTS, in addition to any argument you use to set the heap size. Often you need to increase the amount of memory allocated to fisheye during the initial scan and period and once this is completed you can reduce back down.

```
FISHEYE_OPTS="-Xms128m -Xmx1024m -XX:MaxPermSize=256m"
```

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.

OutOfMemoryError: PermGen space, or Permanent Generation Size

If you get the error message: `java.lang.OutOfMemoryError: PermGen space` this means that you have exceeded Java's fixed 64MB block for loading class files. You will need to add the argument `-XX:MaxPermSize=256m` to FISHEYE_OPTS, in addition to any argument you use to set the heap size.

```
FISHEYE_OPTS="-Xms128m -Xmx512m -XX:MaxPermSize=256m"
```

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.

OutOfMemoryError: unable to create new native thread

This error occurs when the operating system is unable to create new threads. This is due to the JVM Heap taking up the available RAM.

> *Big heaps take away from the space that can be allocated for the stack of a new thread*

For Linux the maximum heap size of the JVM cannot be greater than 2GB. If you only have 2GB RAM in your server, it is **not** recommended to set the Max size of the JVM that high.
The size of the stack per thread can also contribute to this problem. The stack size can reduce the number of threads that can be created.

To fix this problem, you should reduce the size of your JVM Heap and also the size of the stack per thread.
The stack size can be changed with the following (example) parameter being added to your FISHEYE_OPTS:

```
FISHEYE_OPTS="-Xms128m -Xmx1024m -XX:MaxPermSize=256m -Xss512k"
```

Please refer to this guide as a reference for JVM tuning.

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.

OutOfMemoryError: GC overhead limit exceeded

This error indicates that the JVM took too long to free up memory during its GC process. This error can be thrown from the Parallel or Concurrent collectors.

> *The parallel collector will throw an OutOfMemoryError if too much time is being spent in garbage collection: if more than 98% of the total time is spent in garbage collection and less than 2% of the heap is recovered, an OutOfMemoryError will be thrown. This feature is designed to prevent applications from running for an extended period of time while making little or no progress because the heap is too small. If necessary, this feature can be disabled by adding the option -XX:-UseGCOverheadLimit to the command line.*

This kind of OutOfMemoryError can be caused if your java process is starting to use swapped memory for its heap. This will cause the JVM to take a lot longer than normal to perform normal GC operations. This can eventually cause a timeout to occur and cause this error.

To overcome this issue, you need to make sure that all processes can't allocate more memory than there is system memory. In practice this is impossible to do for all processes. At a minimum you should make sure that all your jvm's do not have a total maximum memory allocation than your normally available system memory.

Please refer to this guide for more information.

java.lang.OutOfMemoryError: requested 32756 bytes for ChunkPool::allocate. Out of swap space?

Essentially the native objects does not have enough memory to use. This is usually because you have allocated too much memory to your heap reducing the amount available for native objects. See this article.

The solution is to reduce the amount of heap memory you have allocated. For example if you have set -Xmx4096, you should consider reducing this to -Xmx2048m.

Remember if you are using a 32bit JVM you cannot allocate more than -Xmx2048m for linux (and even less for windows). Using a 64 bit JVM can resolve this problem, but is not recommended for fisheye/crucible instances (refer to System Requirements).

Read the Tuning FishEye page for more detail on adjusting resource limits and performance settings in FishEye.

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

IDE Connectors

- Use the Atlassian Connector for Eclipse or the Atlassian Connector for IntelliJ IDEA to work with FishEye and Crucible right there in your development environment.

Support

- Atlassian Support
- Support Policies

Forums

- FishEye General Forum | subscribe
- FishEye Developers Forum | subscribe

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....

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

Unable to render {include} Couldn't find a page to include called: ALLDOC:_Managing Community Contributions

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 it on your production site.

On this page:

Technical Tips

Using a reverse proxy to Fisheye from Tomcat 5+ – Blog

- By: Dieter Wimberger, on his Coalevo blog.
- About: How to use a reverse proxy to Fisheye from Tomcat 5.
- Date and FishEye version: 21 Jan 2008; FishEye 1.4.2.
- Related documentation: Integrating with Other Web Servers.

Plugin Development

Diary of a FishEye Hacker – Part 1 and Part 2

- By: Dan Hardiker, on the Adaptavist blog
- About: How to write a FishEye reporting plugin that provides a framework for alternative views on the repository data
- Date and FishEye version: 6 May 2008; FishEye 1.5
- Related documentation: FishEye Development Hub

Have you written a technical tip for FishEye?
Add a comment to this page, linking to your blog post or article. We will include it if the content fits the requirements of this page.

Feedback?
Your first port of call should be the author of the linked blog post. If you want to let us know how useful (or otherwise) a linked post is, please add a comment to this page.

Other Sources of Information

FishEye documentation
Atlassian website
Atlassian forums
Atlassian Blog
FishEye plugins

TreeNavigation

Index

TreeNavigationVersions

▷ Click for all versions
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