FishEye Documentation Home

Getting Started with FishEye 2.0

Release Notes (What's new)

Download FishEye | Feature Tour | About FishEye

Installation Guide | Upgrade Guide

Using/Administering FishEye 2.0

User's Guide

Administrator's Guide

Quick Start Guide

FishEye IDE Integration

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. Do you use JIRA or Bamboo too? With the connector you can manage your issues, code reviews and builds within your IDE, or move quickly between the IDE and a FishEye view of your source repository. **Hint:** The Atlassian IDE Connectors are free.

Resources

Support | Forums | FAQ

Development Hub

Crucible Documentation

EyeQL Reference Guide

FishEye Evaluator Resources

Previous Versions

FishEye 1.6 documentation
FishEye 1.5 documentation
FishEye 1.4 documentation
FishEye 1.3 documentation
FishEye 1.2 and older documentation

Offline Documentation

You can download the FishEye documentation in PDF, HTML or XML format for use offline.

Recently Updated

__newreleaseFishEye (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (3 minutes ago)
Viewing People's Statistics (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (36 minutes ago)
Changing your User Profile (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (39 minutes ago)

Configuring Avatar Settings (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (43 minutes ago)
edit_avatar_server_settings.png (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (50 minutes ago)
JIRA Integration in FishEye (FishEye 2.0)	by Tim Pettersen (17 hours ago)
JIRA Integration in FishEye (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (05 Jul)
Using Favourites (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (02 Jul)
Using the FishEye Screens (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (02 Jul)
Re: Can FishEye be run as a Windows service? (FishEye 2.0)	by Anonymous (02 Jul)
FishEye 2.0 Release Notes (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (30 Jun)
FishEye Release Summary (FishEye 2.0)	by Rosie Jameson [Atlassian Technical Writer] (30 Jun)
FishEye 2.0	by Rosie Jameson [Atlassian Technical Writer] (30 Jun)
FishEye Upgrade Guide (FishEye 2.0 Beta)	by Brendan Humphreys (30 Jun)
FishEye 2.0 Release Notes (FishEye 2.0)	by Brendan Humphreys (30 Jun)

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.
  - History charts showing branches and tags.
  - 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 FishEye Installation Guide.

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. Do you use JIRA or Bamboo too? With the connector you can manage your issues, code reviews and builds within your IDE, or move quickly between the IDE and a FishEye view of your source repository. **Hint:** The Atlassian IDE Connectors are free.

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 System Requirements. 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 requirements, 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 requirements 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 Evaluator Resources page.

Using the FishEye Screens

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

Header

The header along the top of the FishEye screen will remain the same as you browse through the different screens. You can

- 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 (draft)).
- Click your name to change your user settings (see Changing your User Profile).

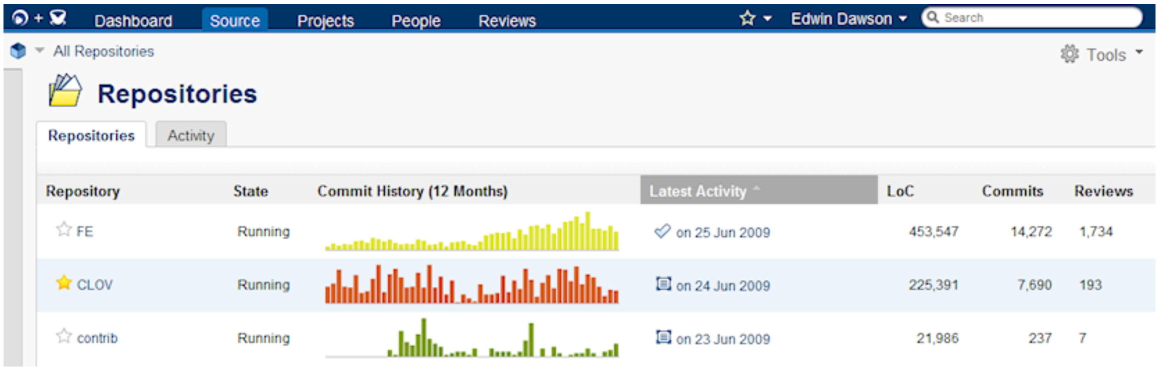
Repository Sub-Tabs

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:

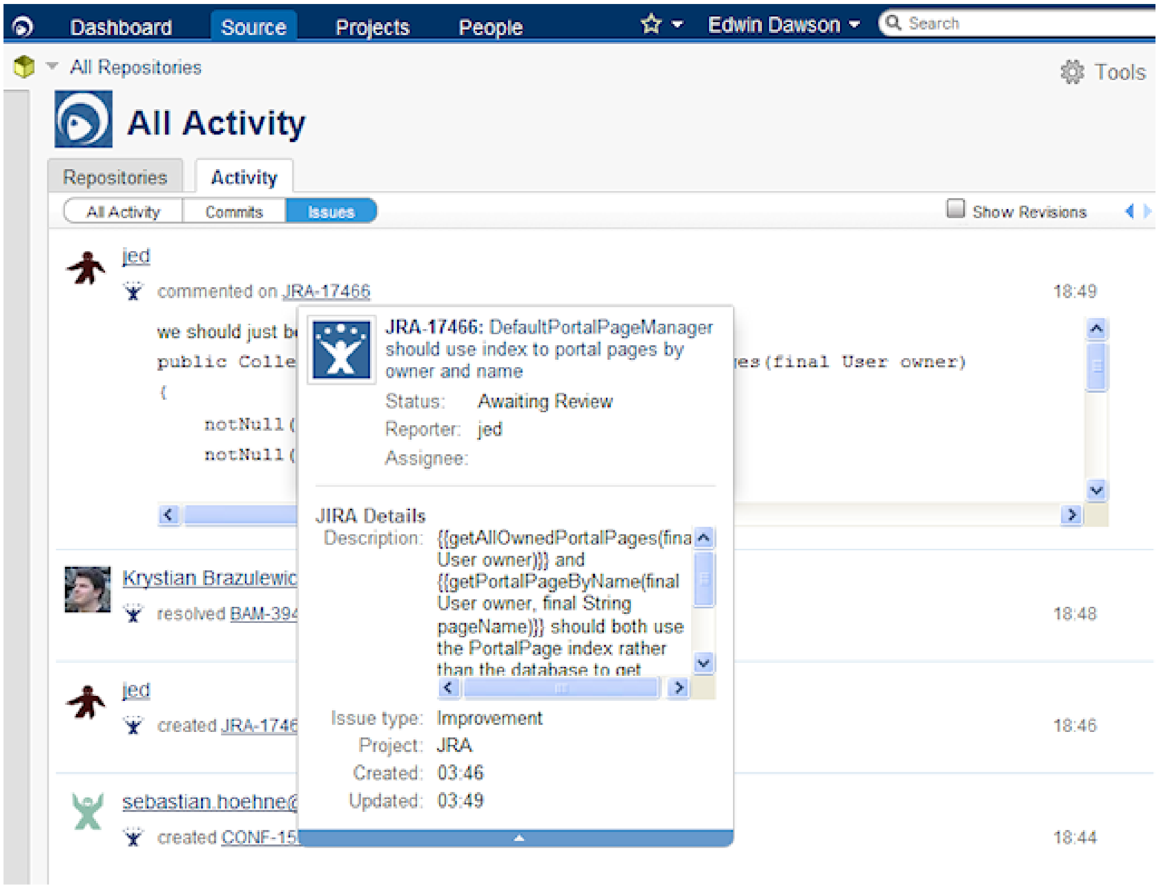
Sub-Tab Name	Description
Revisions	When viewing a file, shows the latest revisions of the file. See Viewing a File History.
Files	When viewing a folder, shows the contents of the directory.
Activity	Shows recent activity on the item. There are a number of sub-options here (see Viewing the Changelog): • All Activity — The default view, showing commits, reviews[1] and JIRA issues[2]. • 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 — 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



All Activity

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:

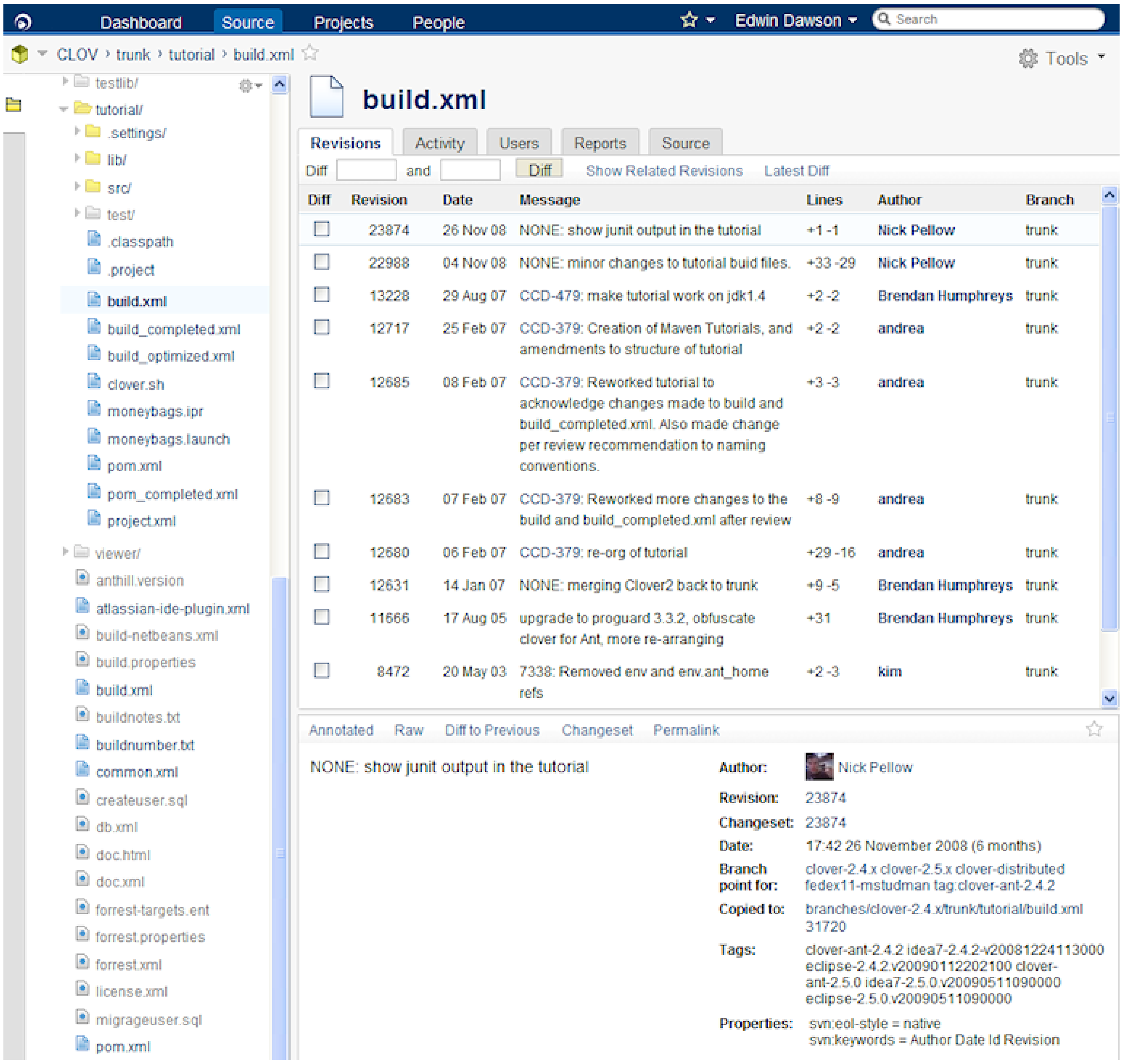
JIRA Details
Description: {{getAllOwnedPortalPages(fina User owner)}} and {{getPortalPageByName(final User owner, final String pageName)}} should both use the PortalPage index rather than the database to get
Issue type: Improvement
Project: JRA
Created: 03:46
Updated: 03:49

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:

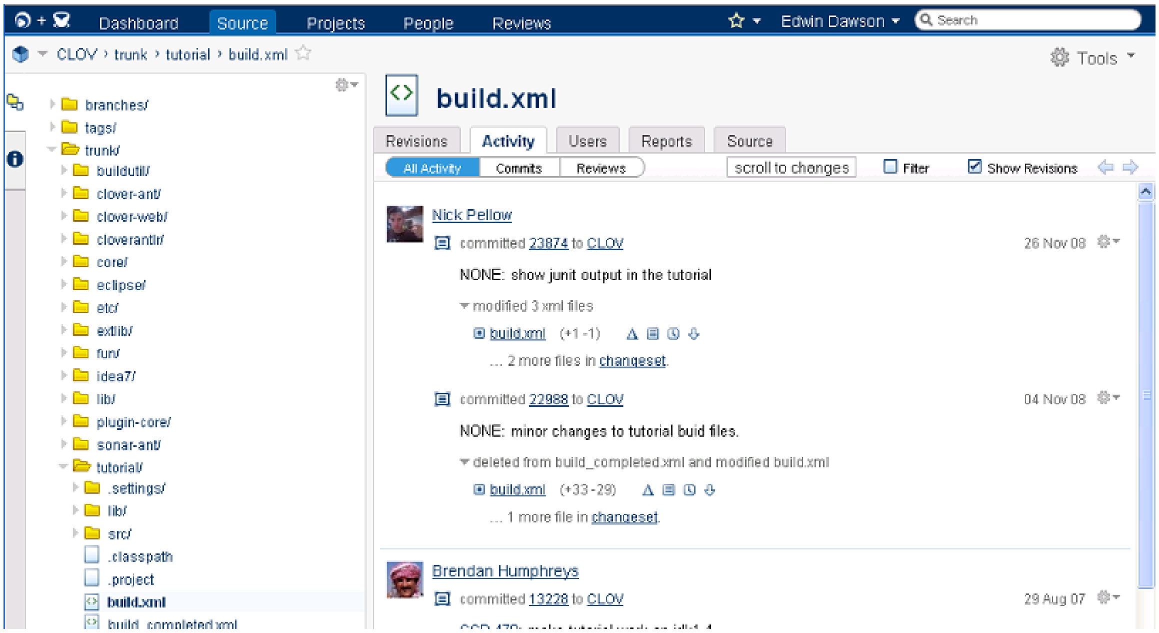
- '**Statistics**' — Total number of committers; last commit; commits this week; total Lines of Code (Loc); change in LoC this week.
- '**Line History**' — This graph shows the total line-count of MAIN or trunk over time for this directory subtree. This line-count does not include binary files, but does 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

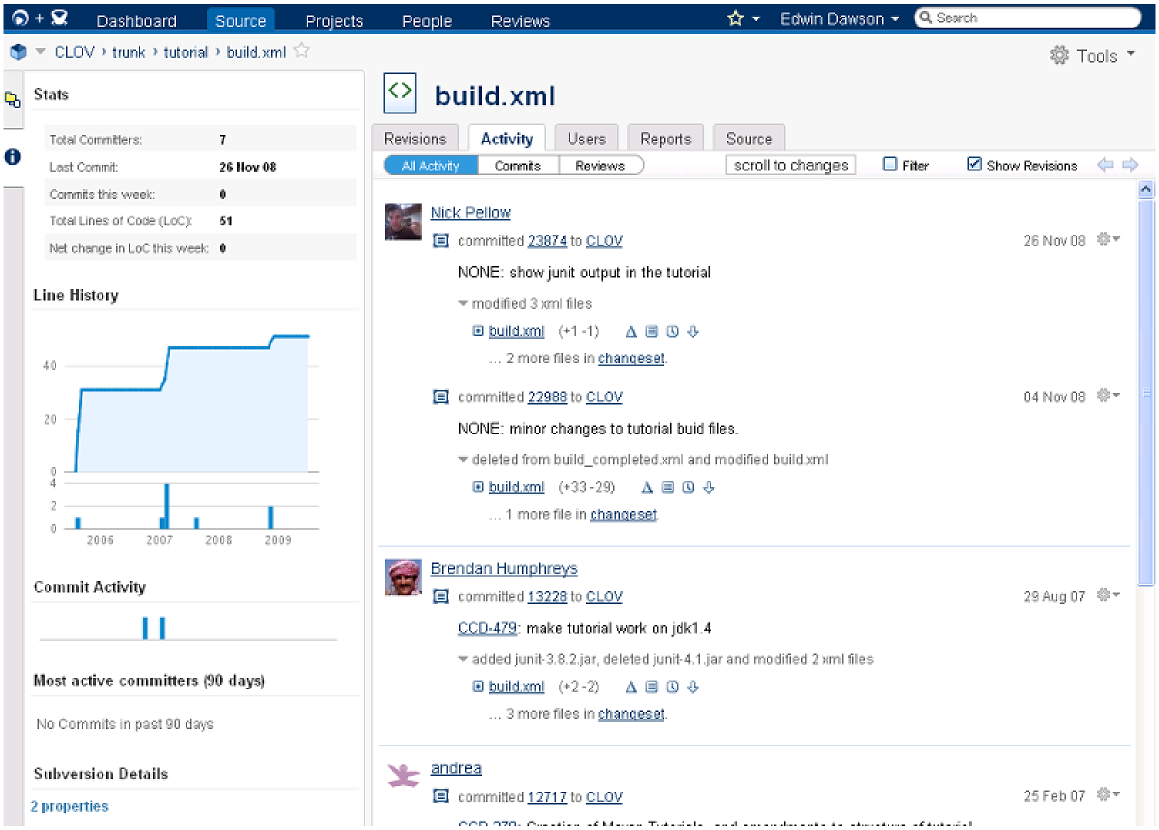
build.xml (+33 -29)

... 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

No Commits in past 90 days

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

The 'Activity' sub-tab in FishEye 2.0 has similar functionality to the '**Changelog**' in previous versions of FishEye.

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, reviews[1] and JIRA issues[2].

- **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

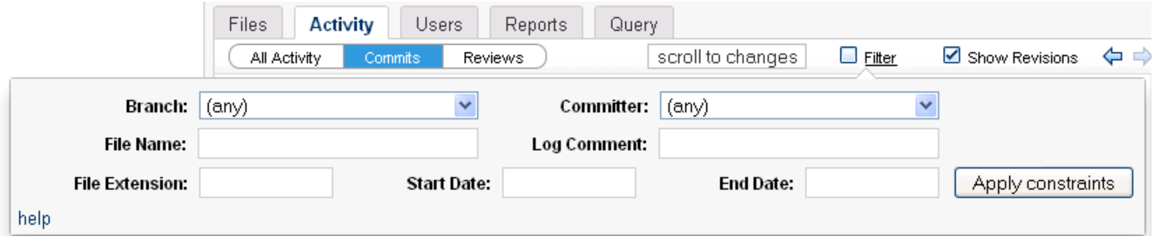
[2] If you are using JIRA

Filter mode

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

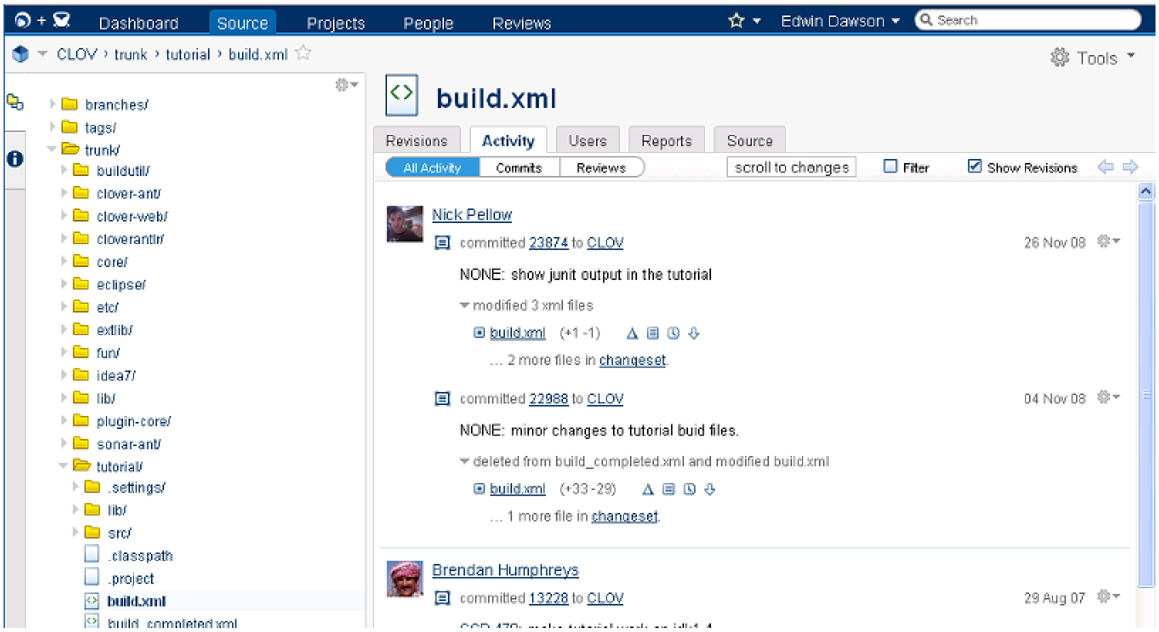
Constraint	Description
Branch	Shows only changesets on that branch.
Committer	Shows only changesets checked in by the given committer/author.
File Name	Shows only changesets that contain a given file.
Log Comment	Shows only changesets where the commit comment matches the given text.
Start Date	Shows 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:

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

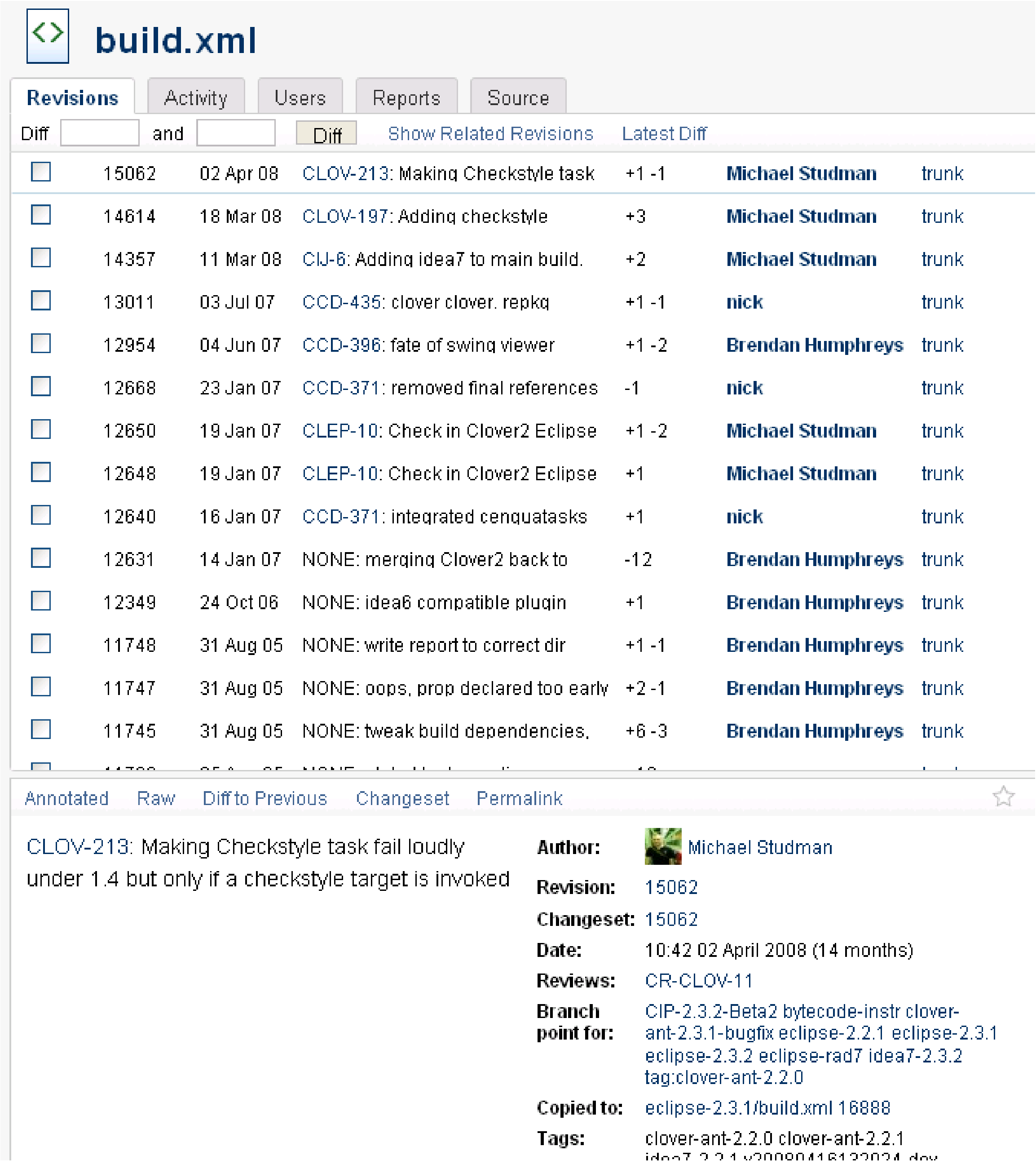
Viewing a File History

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.

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

Searching the Repository

FishEye allows you to search through the repository to find particular changesets or files.

Below we describe the three ways to search:

- Quick Search
- Simple Search
- Advanced Search

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

Search

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.h

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"
```

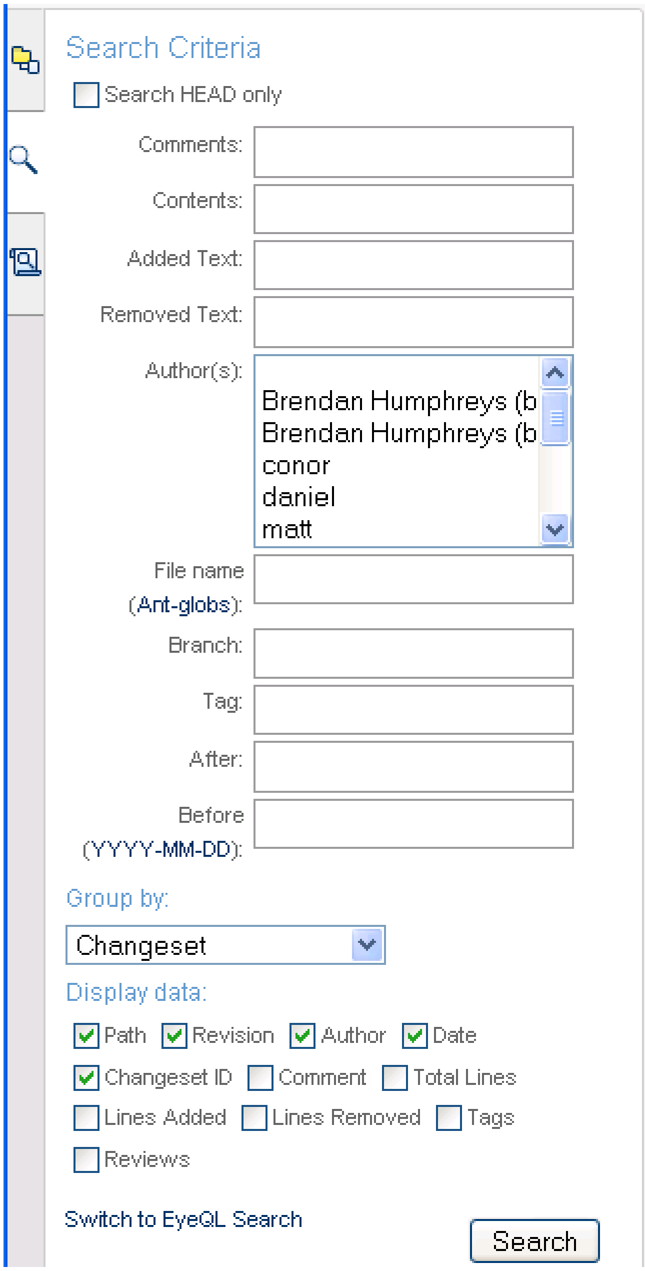
This search will ignore occurrences of the string that appear embedded in other strings, have different case, or are part of a path or disk location. The example above would not return "ProjectMonkey", "monkey", or "/zoo/mammals/monkey/archie/".
Note that regular quick searches do not take case into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



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

Screenshot: FishEye Simple Search panel

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.

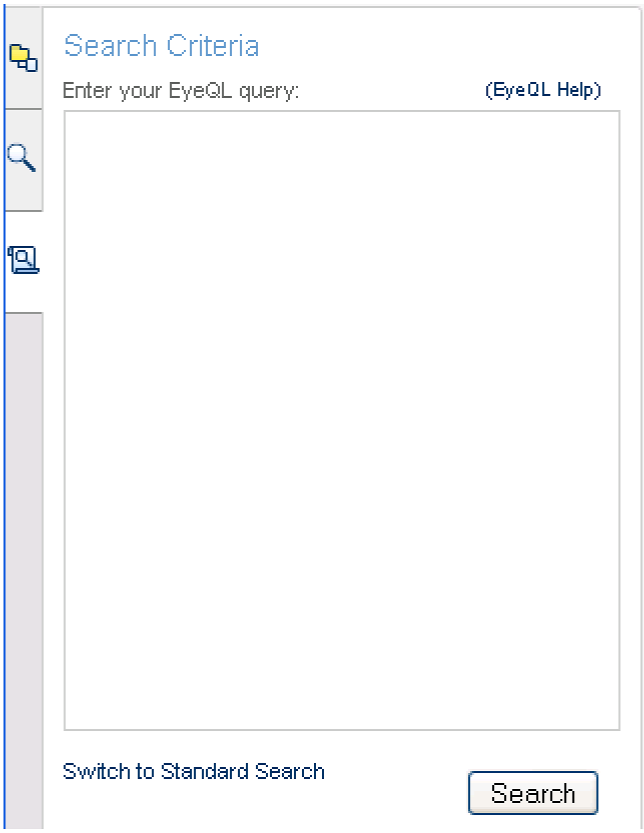
You can choose to include any or all of the following fields in the results:

- 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

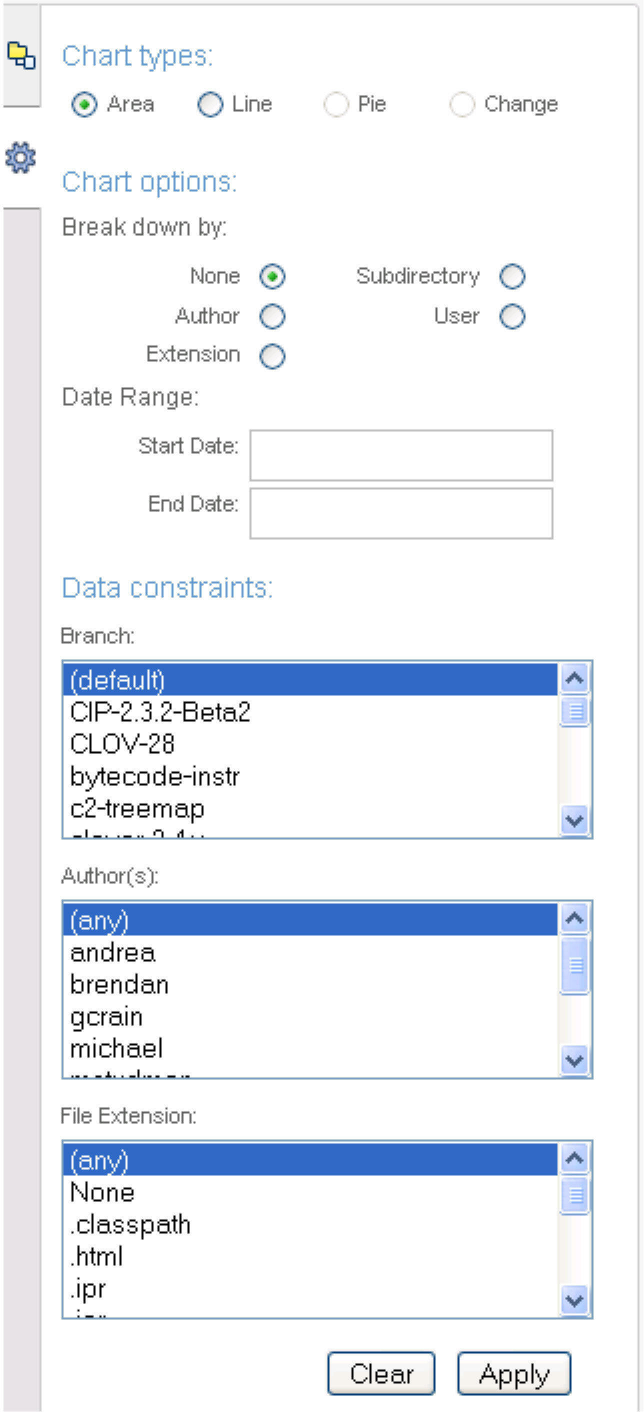


Chart types:

● 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

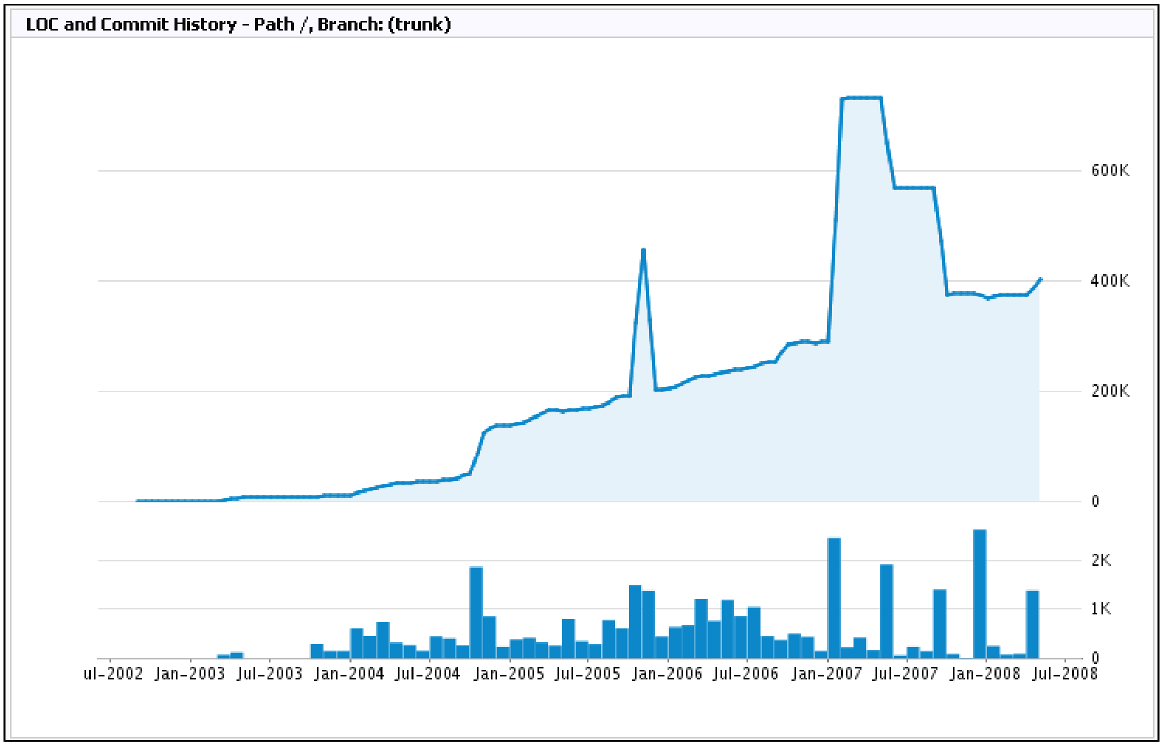
Author(s):

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

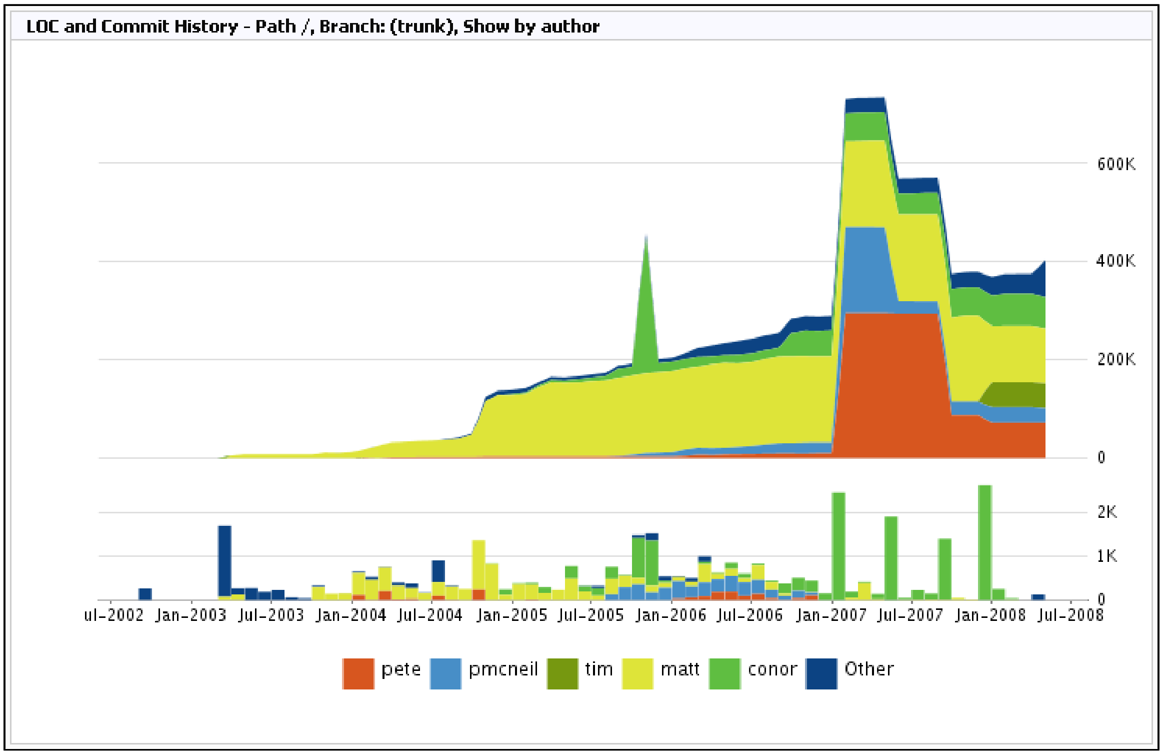
File Extension:

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

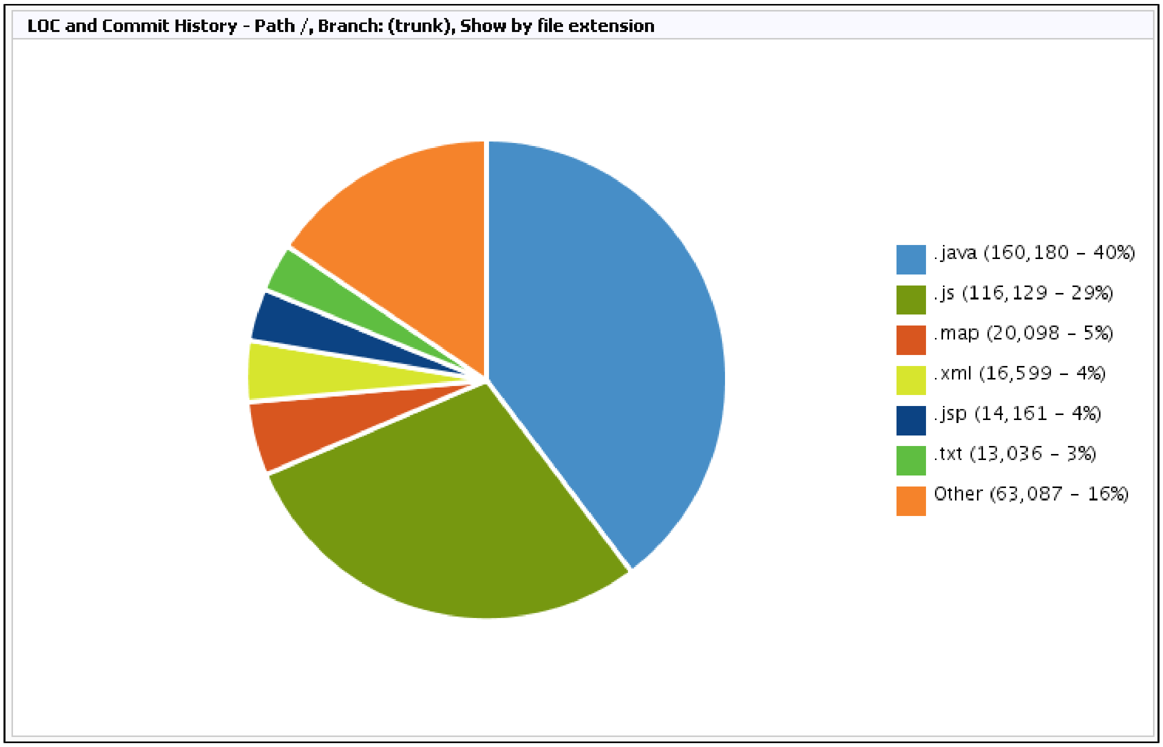
Screenshot: FishEye per-author LOC chart



Screenshot: FishEye per-author LOC chart showing multiple authors



Screenshot: FishEye LOC chart by file extension



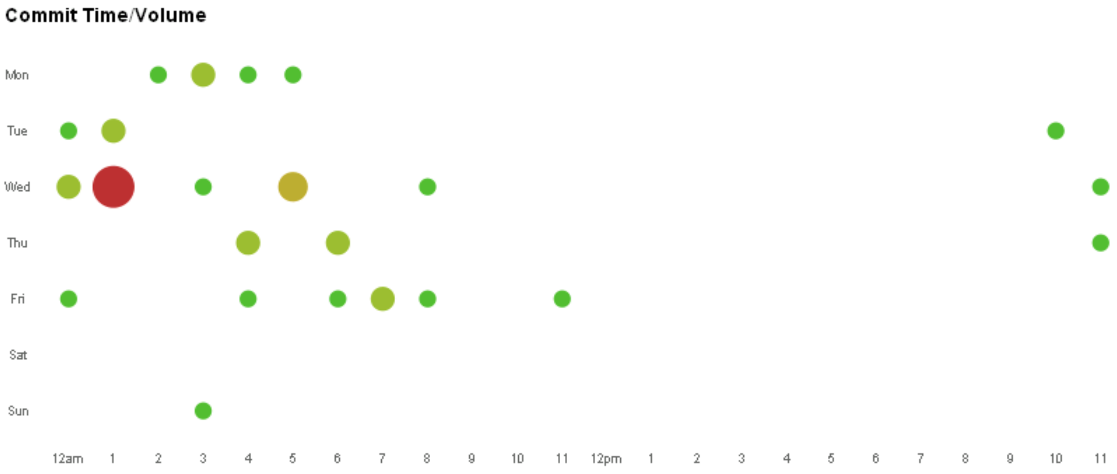
Per-Author Lines of Code Statistics

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.

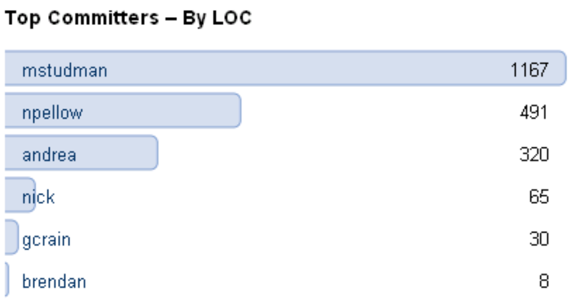
Code Metrics

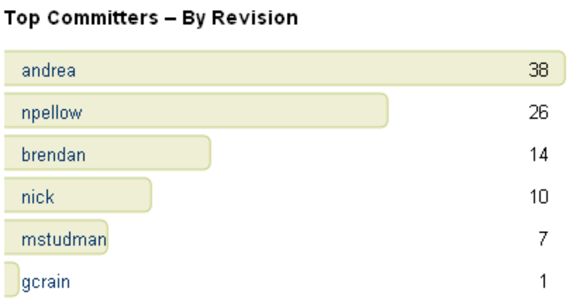
A number of built-in reports are also provided:

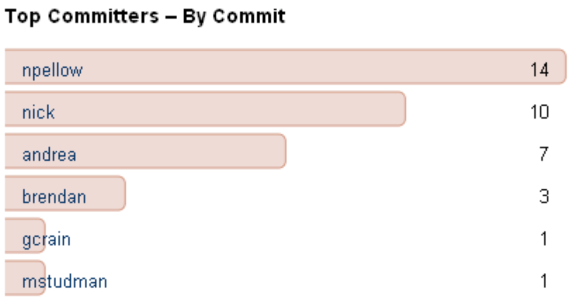
Screenshot: Commit Time/Volume



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.

> 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.

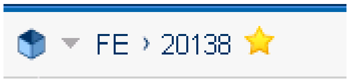
Adding a Person to Your Favourites

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



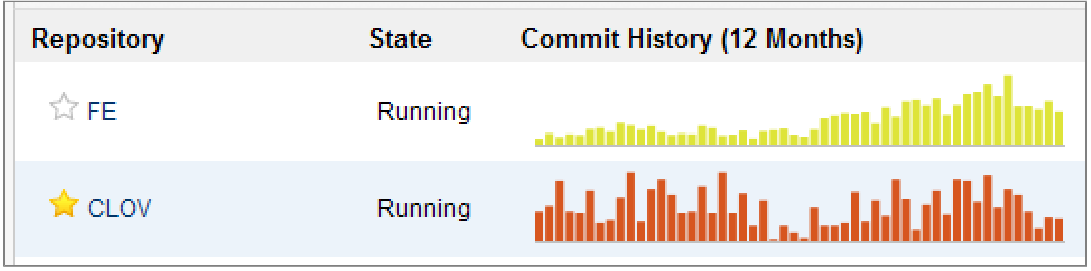
Adding a File or Folder to Your Favourites

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

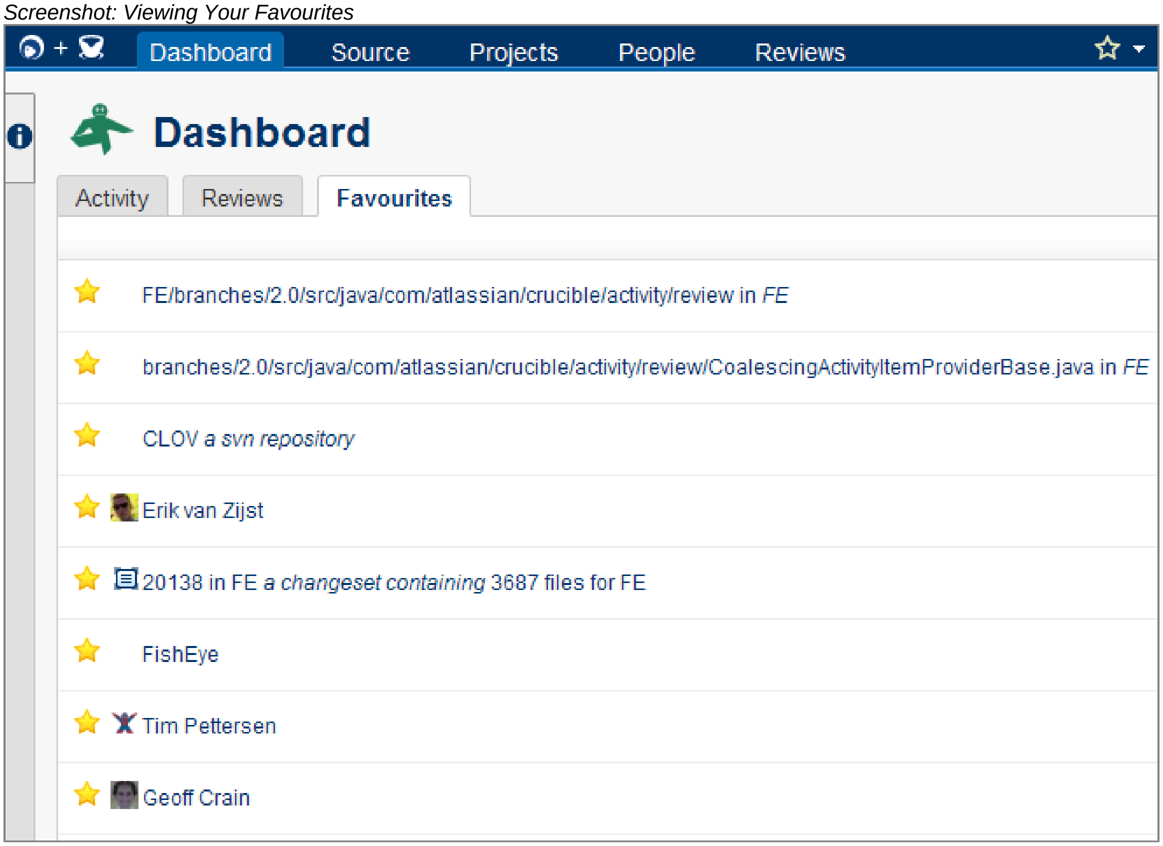


Repository	State	Commit History (12 Months)
☆ FE	Running	
★ CLOV	Running	

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

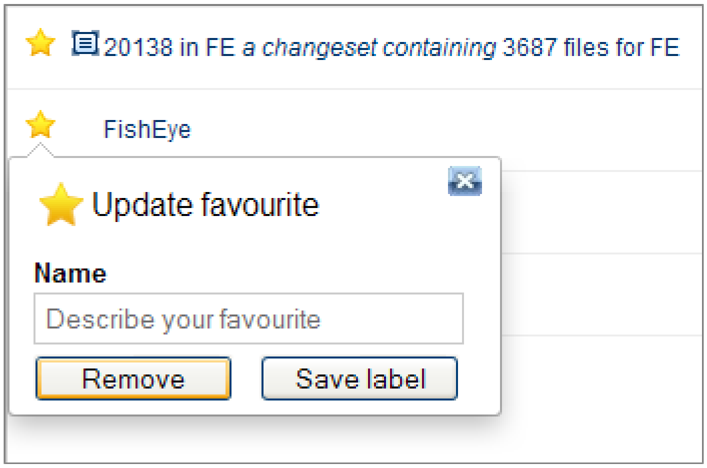
★ FishEye

★ Tim Pettersen

★ Geoff Crain

Renaming an Item In Your Favourites

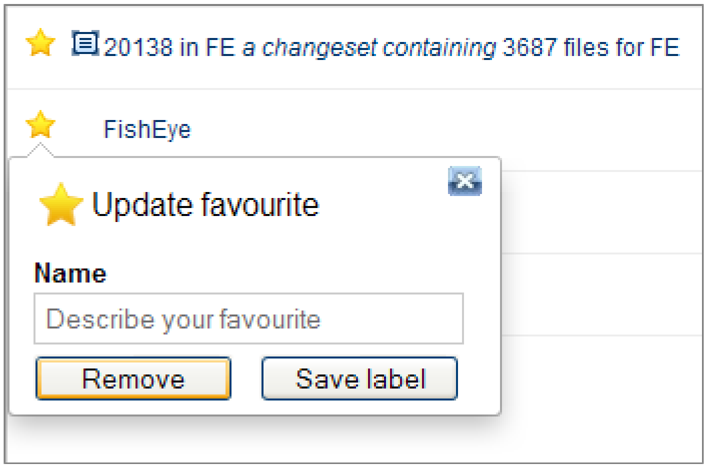
To rename an item in your favourites, open the Favourites drop-down menu (the gold star icon located at the top centre of the FishEye screen, next to your user menu). Select the option called '**Manage favourites**'. The Dashboard favourites page opens, showing all of your favourites in the system. To 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.



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.

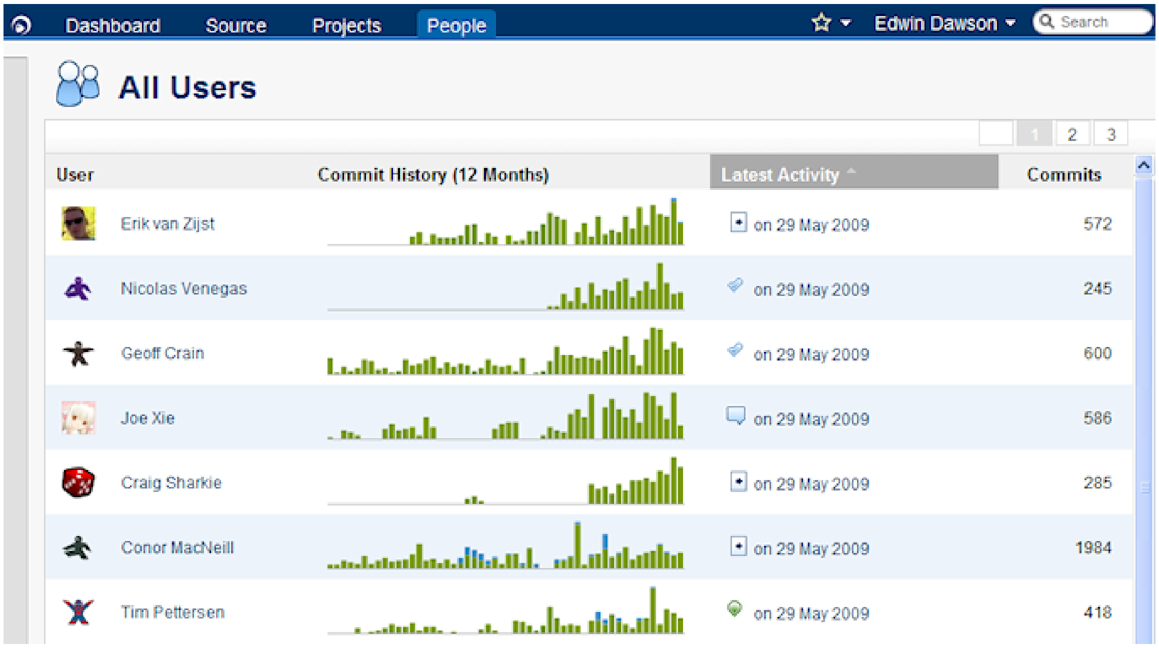


Screenshot: Removing an Item From Your Favourites

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



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

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

Changing your User Profile

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

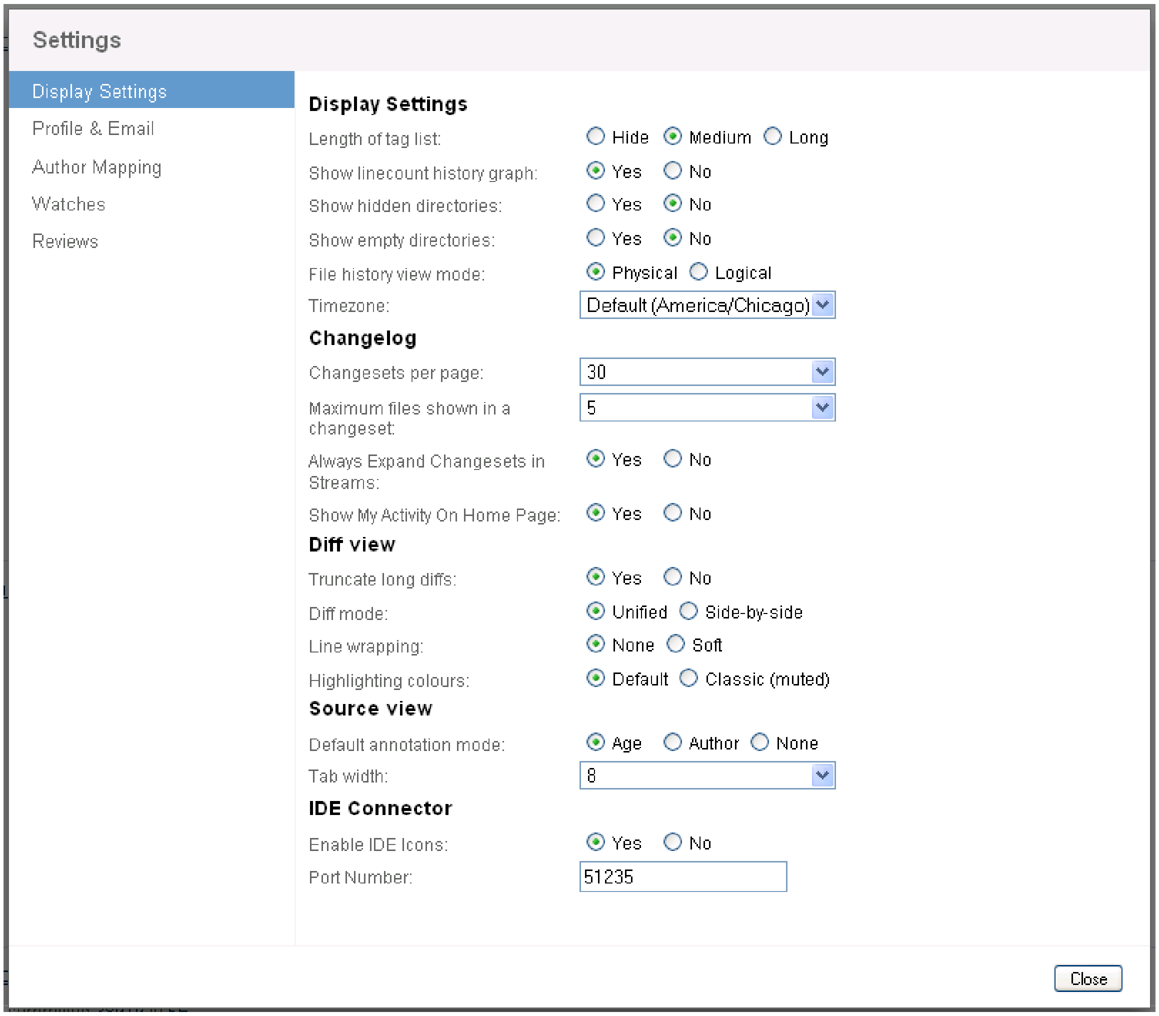
To change these settings, log into FishEye and click the User Menu (labelled with your username) at the top of the screen, then select '**Settings**'.

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

Always click 'Save' after making any changes.

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

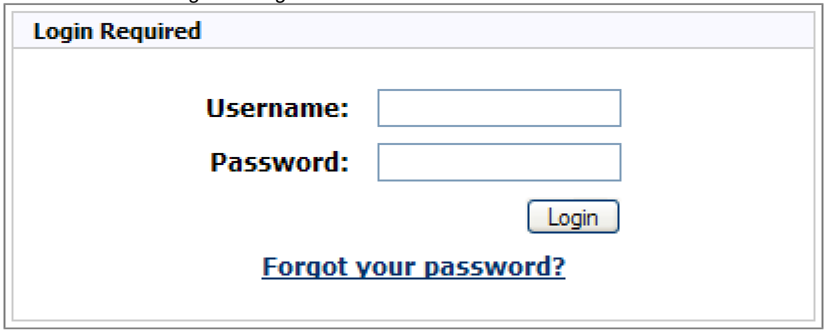
If you need to reset your password, FishEye has an integrated mechanism to generate a new password and send it to the email address in your profile.

To reset your password:

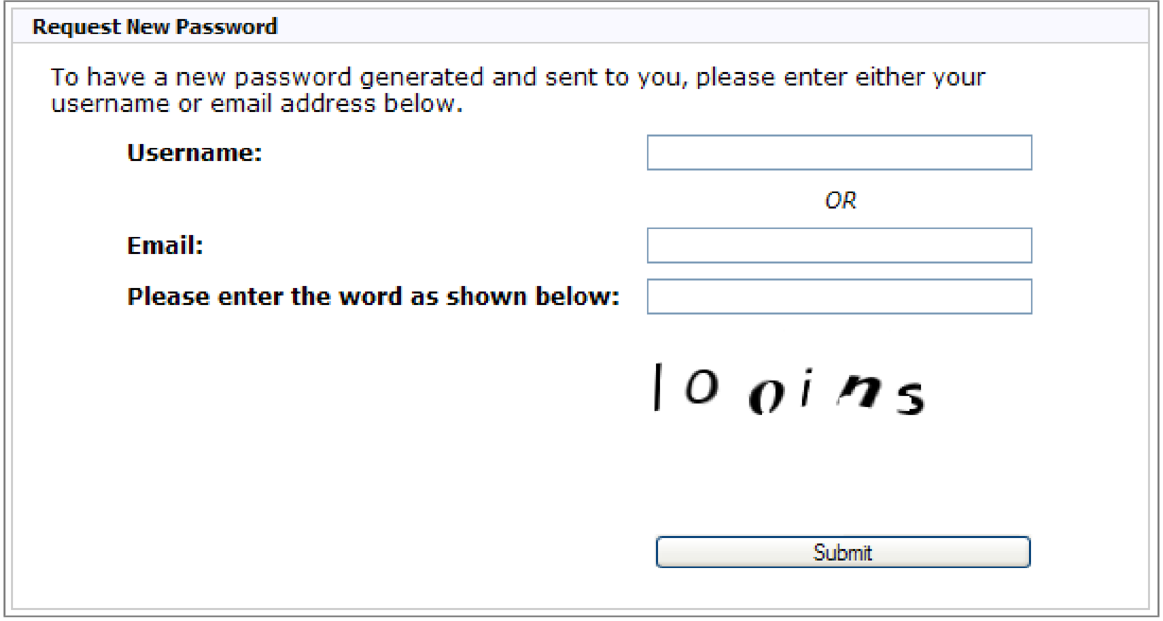
1. On the log in screen, click the '**Forgot your password?**' link. The '**Request New Password**' screen opens.
2. Fill out your username or email address and the Captcha step. That is, click in the form field labelled '**Please enter the word as shown below**' and type the graphical letters shown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

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

Screenshot: The Log In dialog



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.

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 from 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**|**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** | **revision** | **author** | **date** | **comment** | **csid** | **isBinary** | **totalLines** | **linesAdded** | **linesRemoved** | **isAdded** | **isDeleted** | **isCopied** | **isMoved** | **tags** | **reviews**)
(**as** *word*)?
The attribute to return, optionally followed by a name to use for the column.

Notes: **reviews** applies to Crucible reviews.

limit-clause:

(*duration* | *offset*, *duration* | *duration* **offset** *offset*)

Notes: Limits the number of results to return. *offset* specifies the starting point of the truncated result set and *duration* 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)
```

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/`

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

FishEye Administrator's Guide

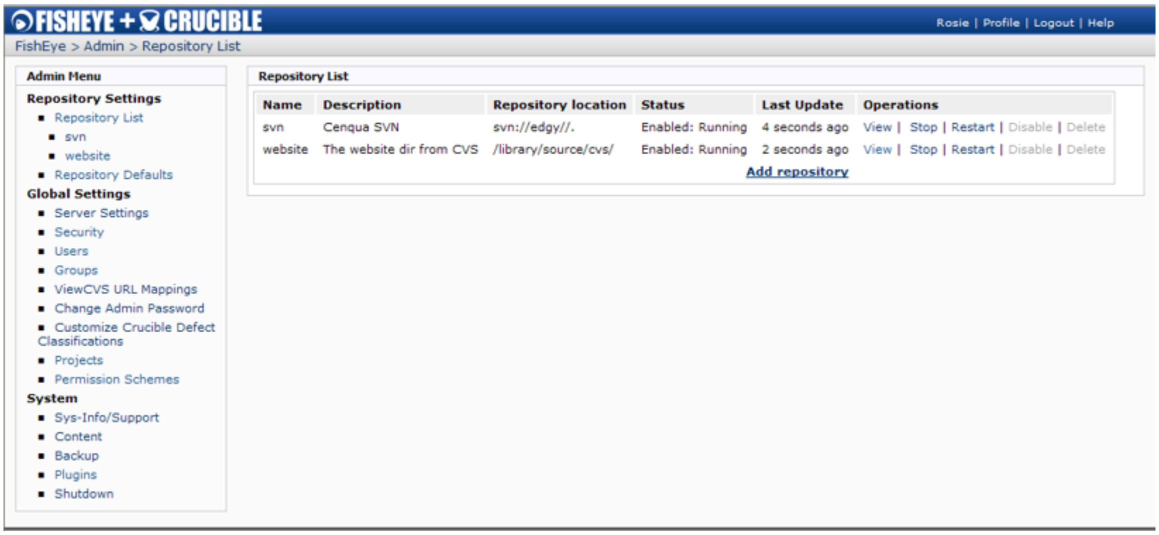
Once you have installed and configured FishEye, you can access the **Administration** pages at http://HOSTNAME:8060/admin/.

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
- Setting up a Repository Client
- Configuring ViewVC Compatibility
- Setting up your Web Server
- Configuring SMTP
- Setting up your Users and Security
- Backing Up and Restoring Config Data
- Advanced Administration Options
- Managing Plugins
- Trusted Applications
- Customising the Welcome Message
- Customising Email Notifications

Managing your Repositories

Adding a Repository

Adding a repository to FishEye is a simple matter. Further configuration options are available once a repository has been added, 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.

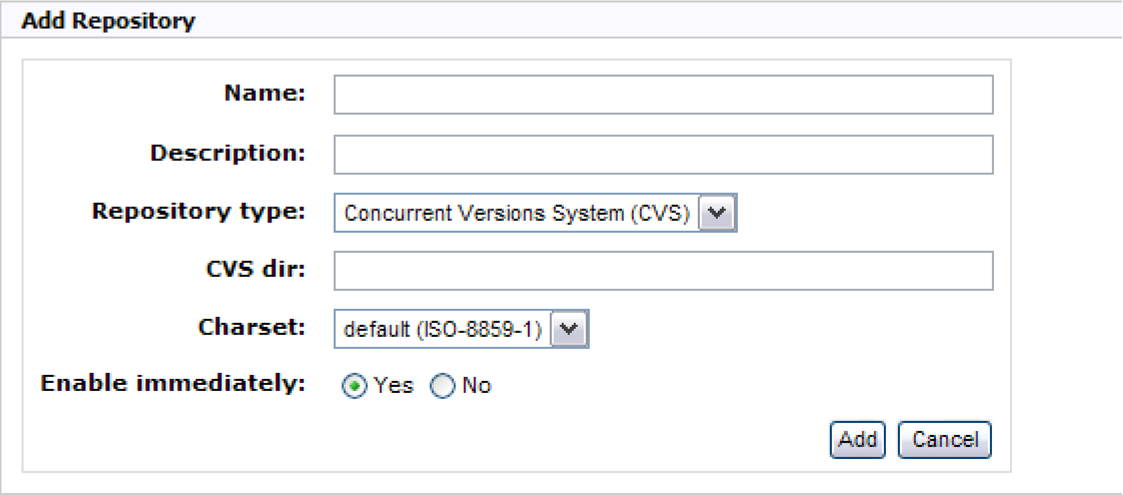
To add a repository,

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:

- CVS
- Git
- 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

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.

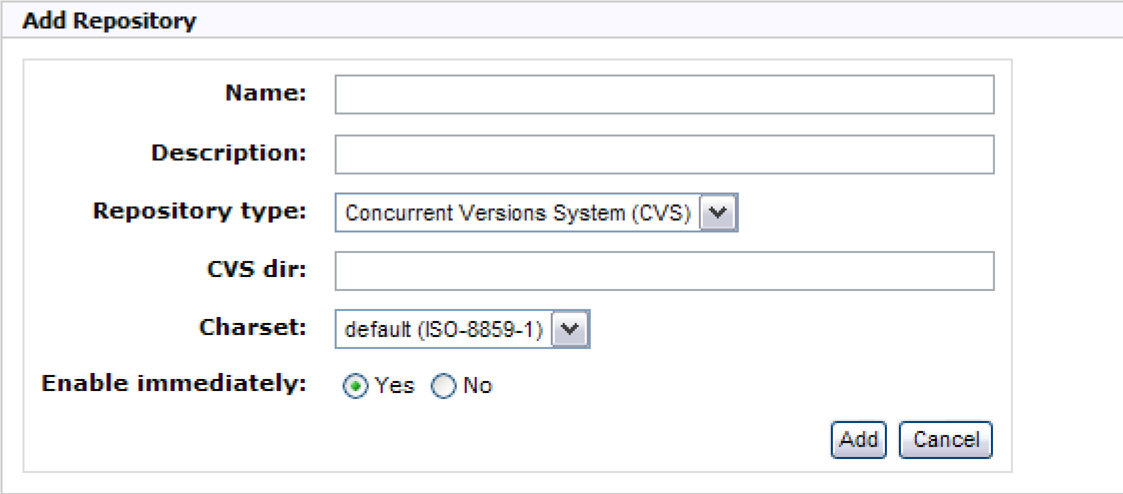
Known Limitations

- 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

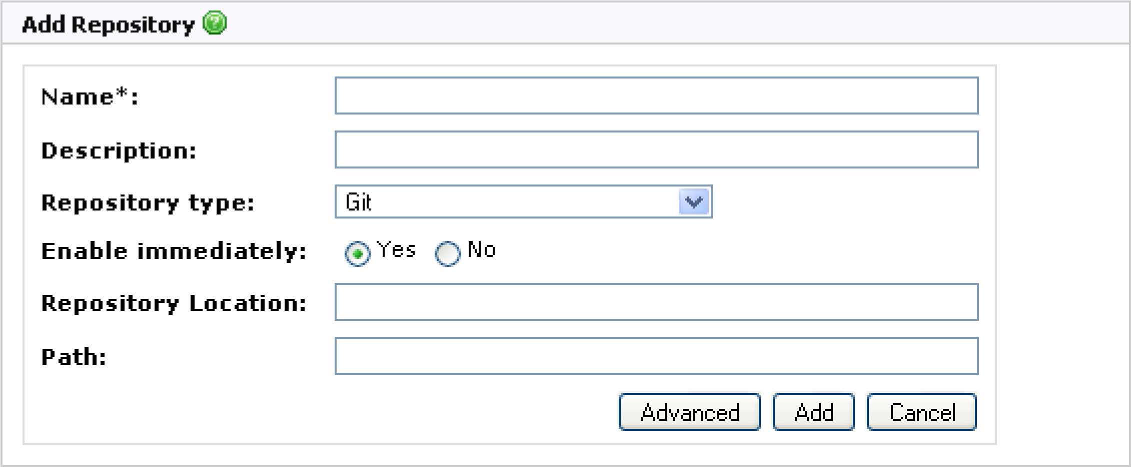
When adding or managing a Git repository, you can:

- 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.

Screenshot: Adding a Git Repository



Add Repository

Name*:

Description:

Repository type: Git

Enable immediately: ● Yes ○ No

Repository Location:

Path:

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.
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

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

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 has 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.
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.

Finding your Repository Root.

Run the following command:

```
svn info SVN_URL
```

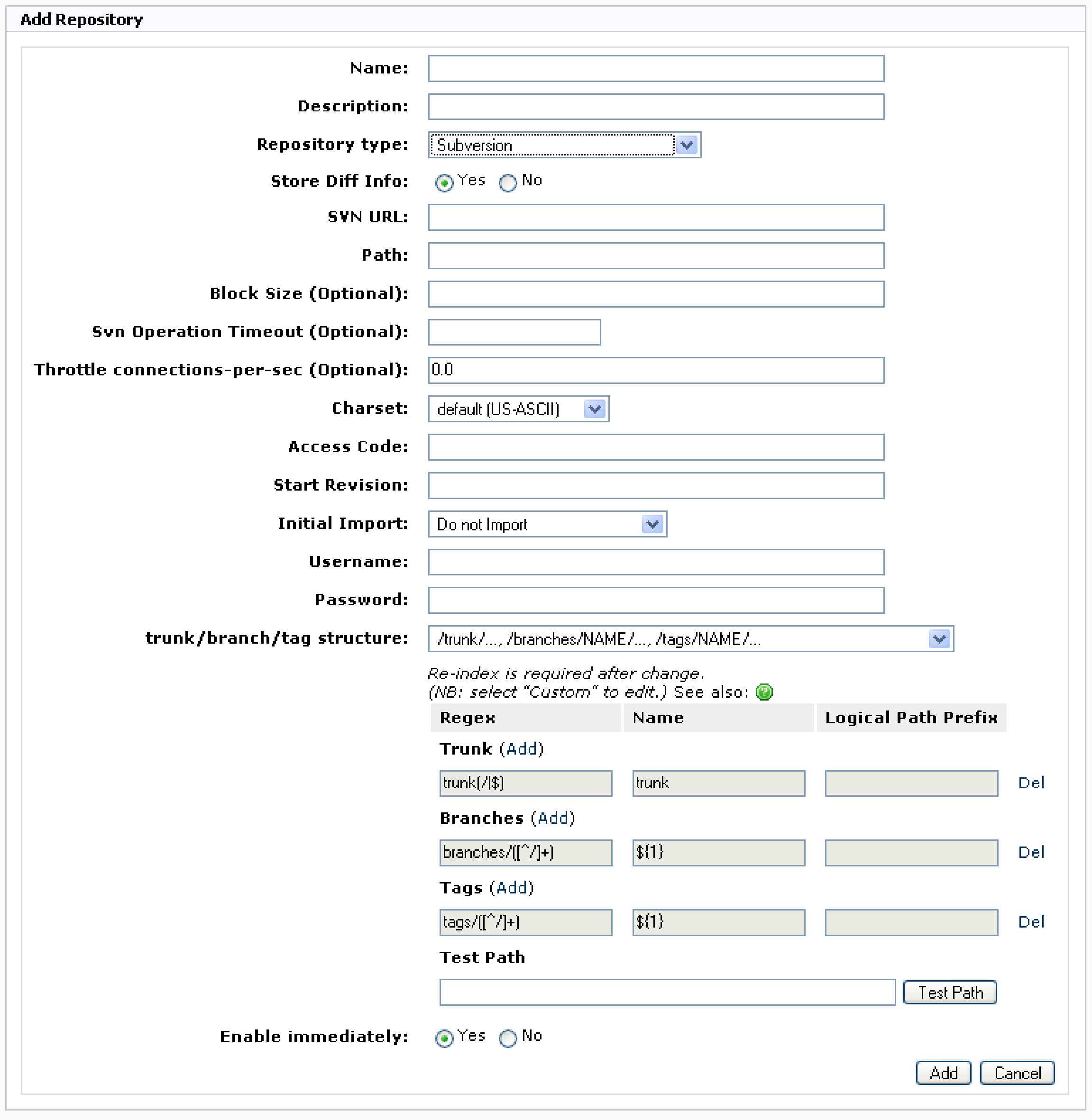
Where SVN_URL is the complete URL of the repository you want to add.

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 & Branch Structure

Since tags and branches in Subversion are implemented via directory copies, they are not really first-class concepts. You can describe what your tag/branch structure looks like, and FishEye will display that information as it would for CVS. 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 in the repository. Each path is classified as either a trunk, branch, tag or root path. 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.

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 will need to do a complete re-scan of the repository. You can do this from the Indexer option.

For more information on tag/branch layout, see Repository Layout in the Subversion documentation, or How Tags Work in Subversion if you are having trouble.

Common layouts

There are two common repository layouts that you can choose from in FishEye. These layouts are described in Repository Layout in the Subversion documentation.

The first is where there are top level `trunk`, `branches` and `tags` directories. This is called '/trunk/..., /branches/NAME/..., /tags/NAME/...' in FishEye.

The second is where the `trunk`, `branches` and `tags` directories are one level down, under each top-level project directory. This is called '/project/trunk/..., /project/branches/NAME/..., /project/tags/NAME/...' in FishEye.

Custom layouts

You can describe to FishEye any custom tag/branch structure you have. If you want to use one of the common layouts as a basis, first select it from the dropdown, then select 'Custom' to edit/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/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.

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.

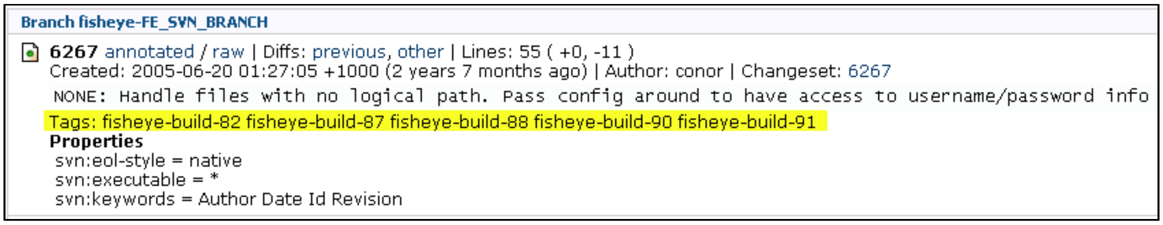
Discovering your Subversion Tag & Branch Structure

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 & Branch Structure.

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:

- CVS
- Git
- Perforce
- Subversion

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**'.

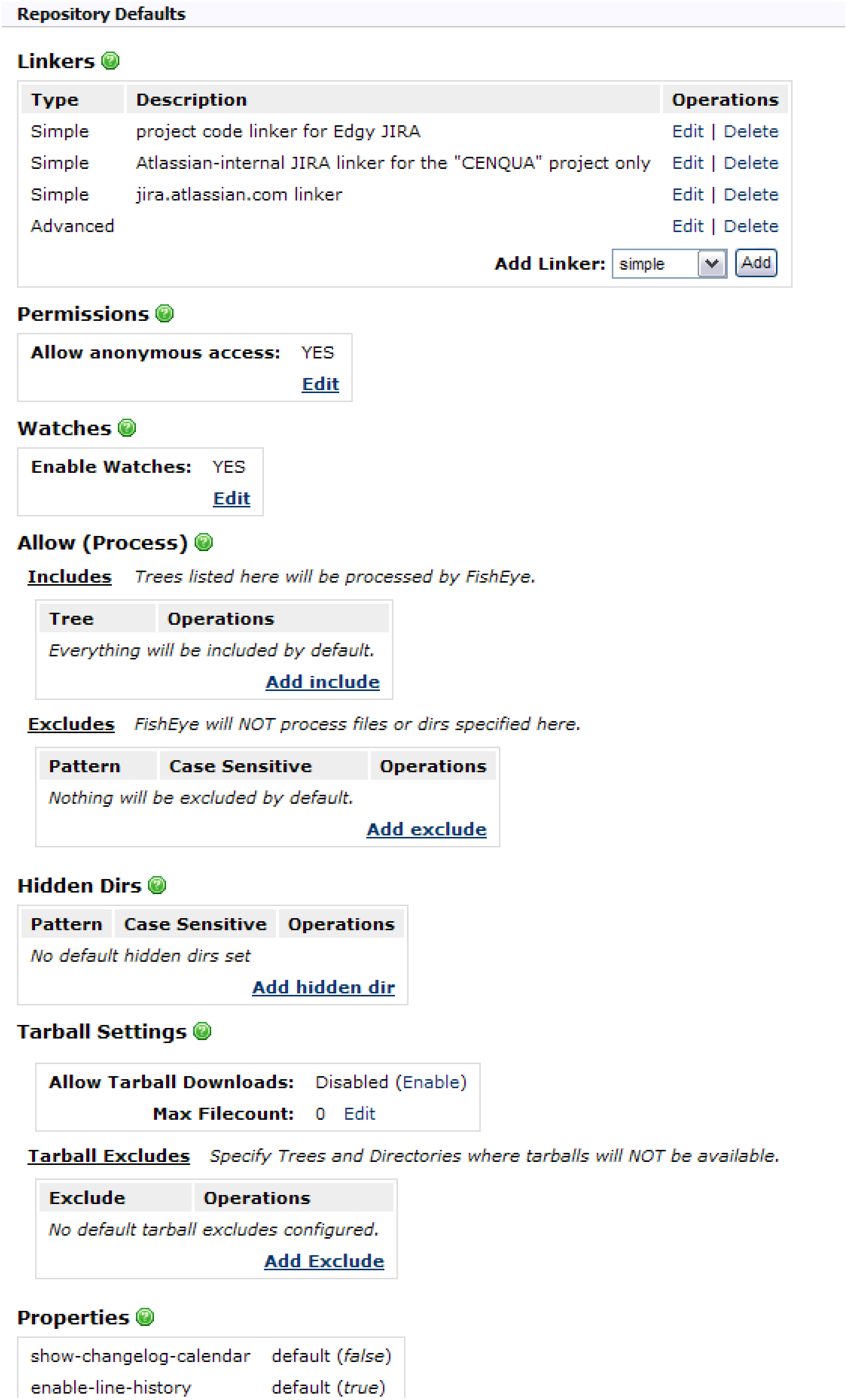
> ⚠ Some changes will require the repository to be restarted, while others will require the repository to be re-indexed. FishEye will advise you if this is the case when you make the change. You can restart a repository from the '**Repository List**'.

Repository options:

- Allow (Process)
- Hidden Directories
- Indexer
- Linkers
- Properties
- Store Diff Info
- Tarball Settings
- Updater
- Watches

* Permissions

Screenshot: Repository Defaults



Repository Defaults

Linkers

Type	Description	Operations
Simple	project code linker for Edgy JIRA	Edit | Delete
Simple	Atlassian-internal JIRA linker for the "CENQUA" project only	Edit | Delete
Simple	jira.atlassian.com linker	Edit | Delete
Advanced		Edit | Delete

Add Linker: simple Add

Permissions

Allow anonymous access: YES

Edit

Watches

Enable Watches: YES

Edit

Allow (Process)

Includes *Trees listed here will be processed by FishEye.*

Tree	Operations
Everything will be included by default.	

Add include

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

Pattern	Case Sensitive	Operations
Nothing will be excluded by default.		

Add exclude

Hidden Dirs

Pattern	Case Sensitive	Operations
No default hidden dirs set		

Add hidden dir

Tarball Settings

Allow Tarball Downloads: Disabled (Enable)

Max Filecount: 0 Edit

Tarball Excludes *Specify Trees and Directories where tarballs will NOT be available.*

Exclude	Operations
No default tarball excludes configured.	

Add Exclude

Properties

show-changelog-calendar	default (*false*)
enable-line-history	default (*true*)

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.
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.

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.

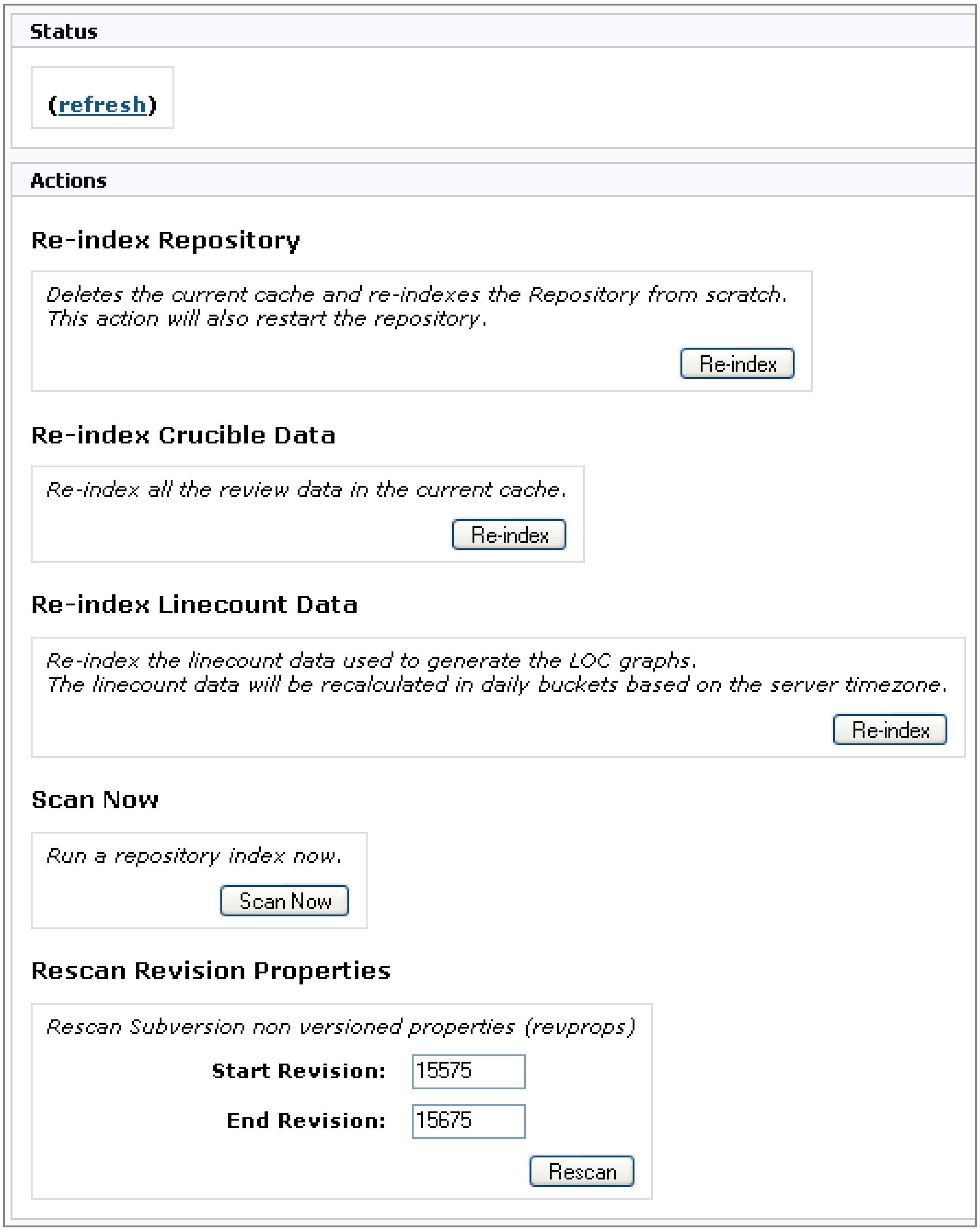
Indexer

To get to this page, go to Administration > Repository List > View (next to your repo) > Indexer > Maintenance

The '**Indexer**' repository option allows an administrator to 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	Re-index all the review data in the current cache.
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 set the repository to not poll automaticially, or it is set with a long poll (interval) period.
Rescan Revision Properties	Rescan Subversion non versioned properties (revprops). **SVN only.** In Subversion it is possible to allow non-versioned properties (e.g. the check-in comment) 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.
Full Scan	**CVS only.** Scan the whole repository for any changes since the last scan.

Screenshot: FishEye Index Maintenance menu


Status

(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 Revision Properties

Rescan Subversion non versioned properties (revprops)

Start Revision: 15575

End Revision: 15675

Linkers

FishEye can detect special substrings in commit messages, and hyperlink those substrings to other systems. The result is, 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 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

- 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.

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.

Store Diff Info

To find this setting, open the FishEye **Admininistration Screen**, then **Repository Settings**.

'**Store Diff Info**' is a value that can be toggled on or off. On is the default setting for new repositories. Repositories created before FishEye 1.5 will default to off.

Store diff info means that we're caching in our database the summary of what lines are added and removed between subsequent versions of the same file. I.e 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.

Considerations for the Store Diff Info Setting

1. Required for Per-author Line Graphs

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

2. Required for Displaying Context of Search Results

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

3. 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.

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

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.

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. E.g. you put 1 hour, then your next scan will begin 1 after your initial scan is complete.

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.

Setting up a Repository Client

- CVS Client
- Git 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

This feature is currently in alpha, released with the product for testing purposes. Do not use in production.

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

Introduction and Disclaimer

Alpha support of Git indexing is included in this release. Atlassian stresses that this is alpha level support which means the following:

1. There may be missing features.
2. Subsequent betas are likely to require re-indexing (although Git indexing is reasonably fast).
3. 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 Git support in FishEye.

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.

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.

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
```

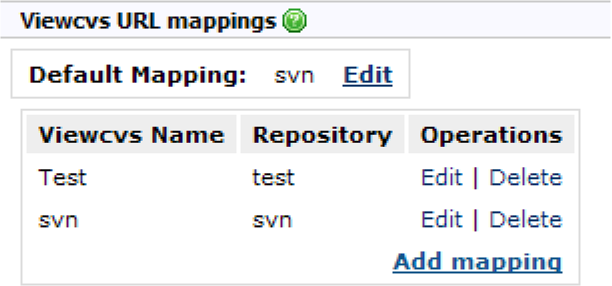
can be viewed in FishEye at

```
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

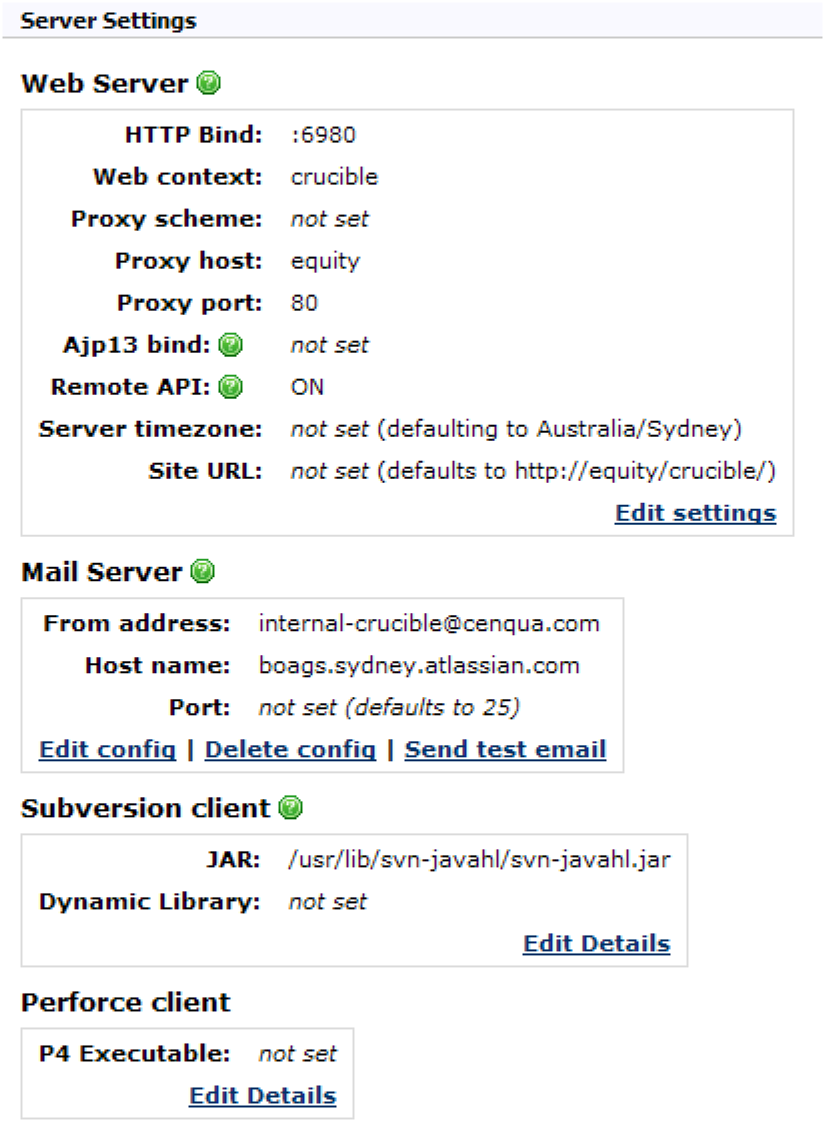
Once you have entered the mappings you will need to **restart** Fisheye.

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/)

Mail Server

From address: internal-crucible@cenqua.com

Host name: boags.sydney.atlassian.com

Port: *not set (defaults to 25)*

Subversion client

JAR: /usr/lib/svn-javahl/svn-javahl.jar

Dynamic Library: *not set*

Perforce client

P4 Executable: *not set*

Configuring the FishEye Web Server

The page 4. Setting up your Web Server does not exist.

> ⚠ 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.

<table>
<tr><td><b>Examples</b></td><td>You can use bindings like the following:
<ul>
<li><b>Host name and port number</b>:
<pre>hostname:8060</pre></li>
<li><b>port number only</b> (requires a leading colon):
<pre>:8060</pre></li>
<li><b>IP address and port number</b>:
<pre>10.0.0.11:60</pre></li>
</ul>
(<i>At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.</i>)

⚠ 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</td></tr>
<tr><td><b>Restart required</b></td><td>Yes</td></tr>
</table>

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.

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

Unable to render embedded object: File (FishEye_1.6.3_SMTP_Configuration.png) not found.

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.

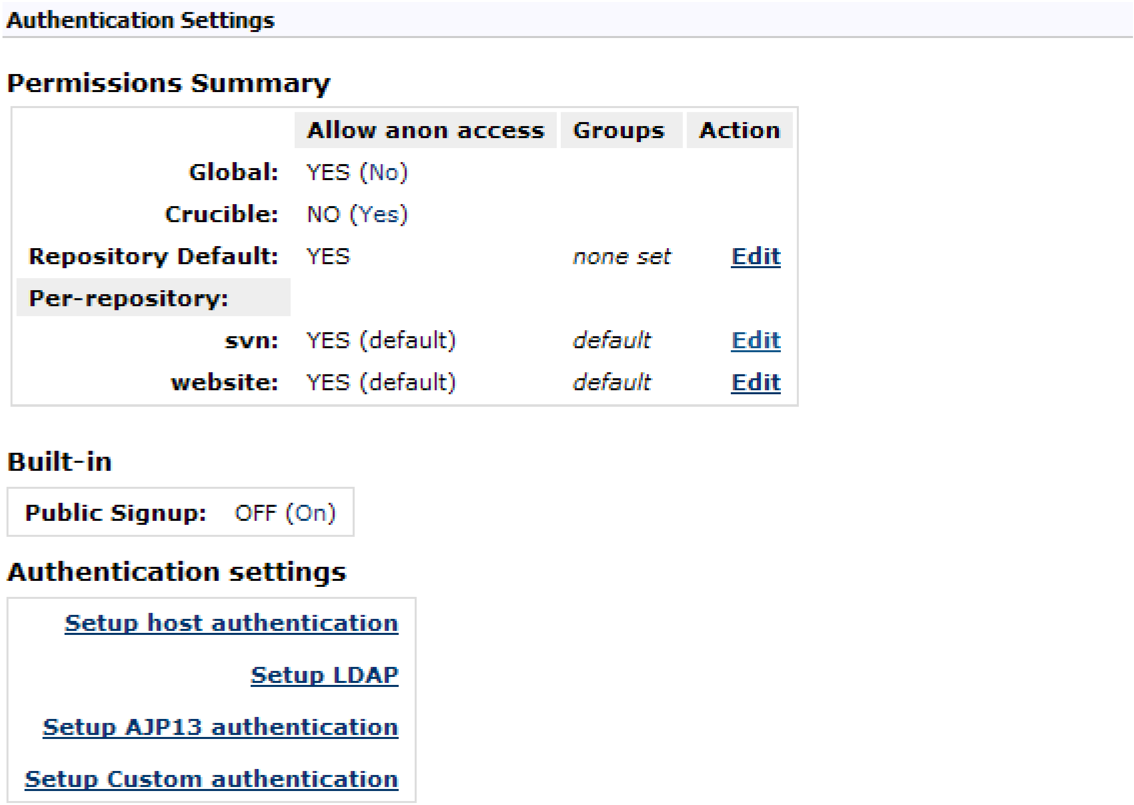
Anonymous access to FishEye is allowed by default. You can disable anonymous access at a global level and per repository.

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

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

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

Screenshot: Authentication Settings



Authentication Settings

Permissions Summary

	Allow anon access	Groups	Action
Global:	YES (No)		
Crucible:	NO (Yes)		
Repository Default:	YES	*none set*	Edit
Per-repository:			
svn:	YES (default)	*default*	Edit
website:	YES (default)	*default*	Edit

Built-in

Public Signup: OFF (On)

Authentication settings

Setup host authentication

Setup LDAP

Setup AJP13 authentication

Setup Custom authentication

Adding a User to a Group

There are two types of FishEye user groups:

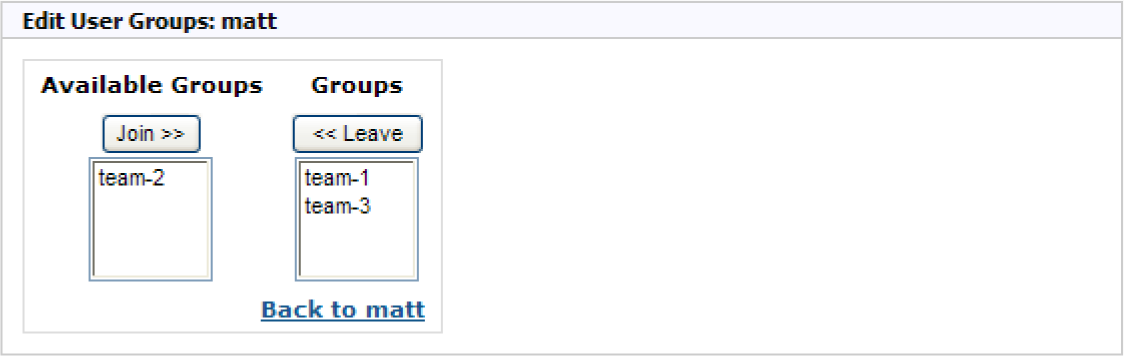
- 'Built-in' groups — these are stored in the FishEye database.
- 'External' groups — these are stored in an external directory (e.g. Crowd), if any are configured. See Configuring External Authentication Sources.

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.
   If the user doesn't initially appear on the screen, type part of their email address and/or select a group to which they belong, and click the '**Filter**' button.
4. The '**Edit User**' screen will be displayed (for more information please see Editing a User's Details). Click the '**Edit Groups**' link.
5. The '**Edit User Groups**' screen will be displayed as shown in the screenshot below.
   - To add the user to a group, select the group in the '**Available Groups**' column at left and click the '**Join**' button.
   - To remove the user from a group, select the group in the '**Groups**' column at right and click the '**Leave**' button.
6. Click the '**Back to user**' button.

Screenshot: Edit User Groups



Associating a Group with a Repository

Associating a group with a repository means that only members of that group will be able to access the repository. For more information, please see Understanding security.

To associate a group with a repository,

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

Screenshot: Authentication Settings

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

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 unique avatar that is randomly formed from the text in their email address.

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

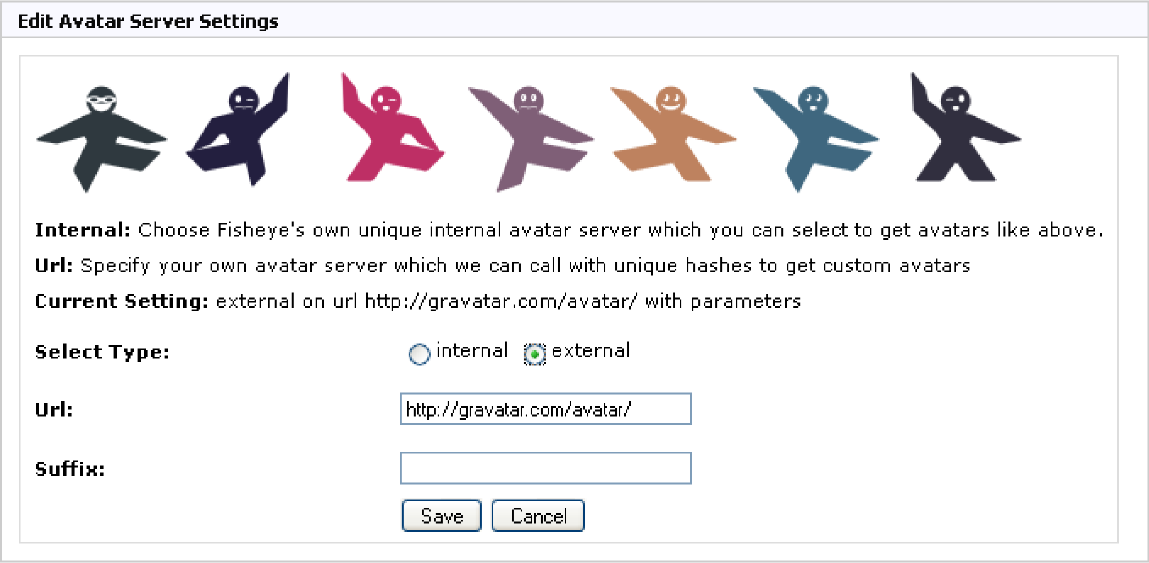
To allow avatars from an external server,

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/
   ℹ For additional security, it is recommended that you specify the following:

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

6. In the '**Suffix**' field, you can optionally specify a constraint 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").
7. Click the '**Save**' button.
8. Users can now upload their own images to the external server.
   ℹ Note that 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

Internal: Choose Fisheye's own unique internal avatar server which you can select to get avatars like above.

Url: Specify your own avatar server which we can call with unique hashes to get custom avatars

Current Setting: external on url http://gravatar.com/avatar/ with parameters

Select Type: ○ internal ● external

Url: http://gravatar.com/avatar/

Suffix:

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 '**Users/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 1. System Requirements does not exist.

(hidden draft) Crowd Authentication

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**' -> '**Users/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.

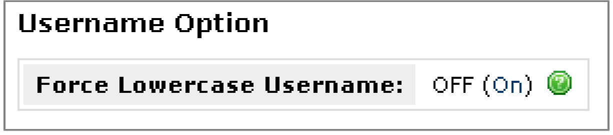
Forcing Imported Usernames to be Lowercase

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:

- Global Settings
- Per-Repository Settings
- 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.

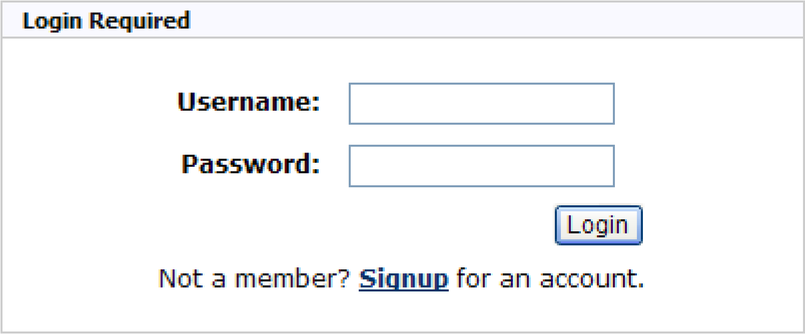
Active Directory

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`

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:



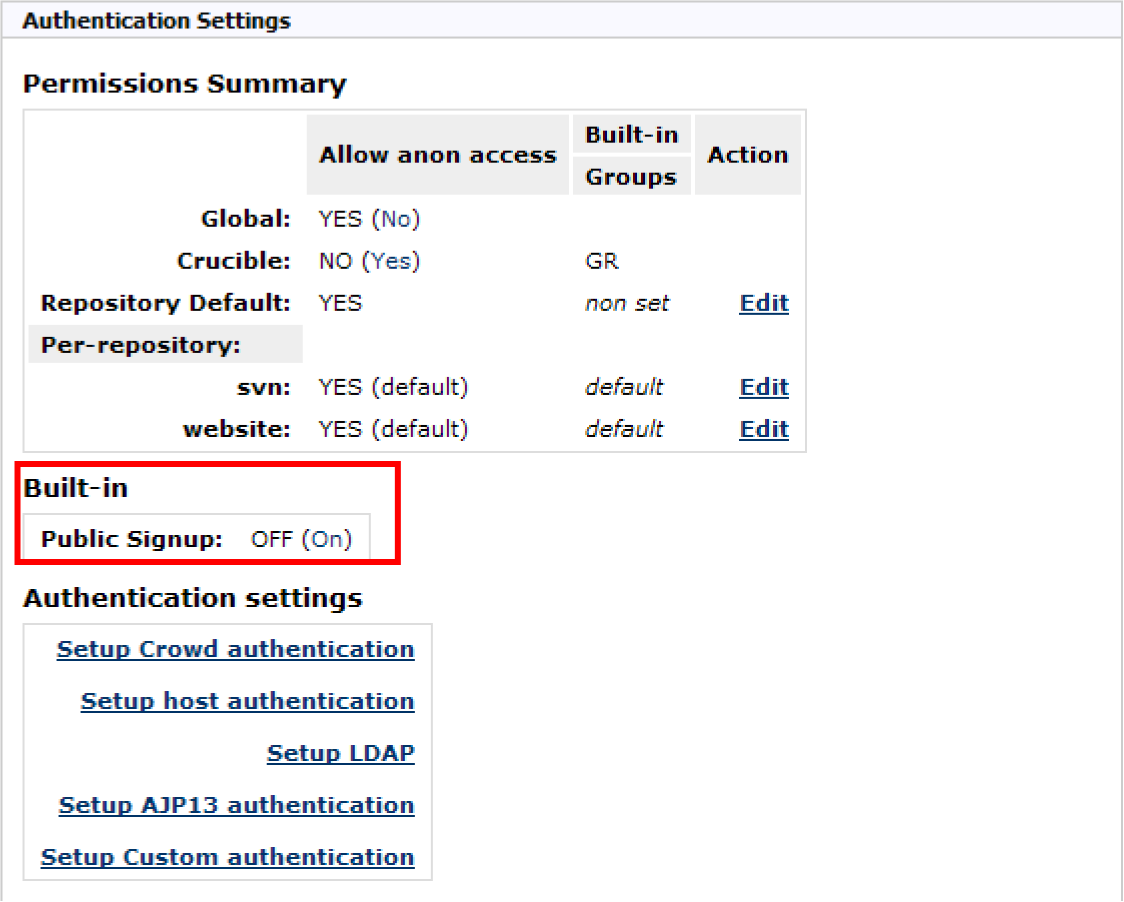
Not a member? Signup for an account.

Public signup is disabled by default.

To enable public signup,

1. Go to the FishEye '**Admin Menu**'.
2. Click 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.

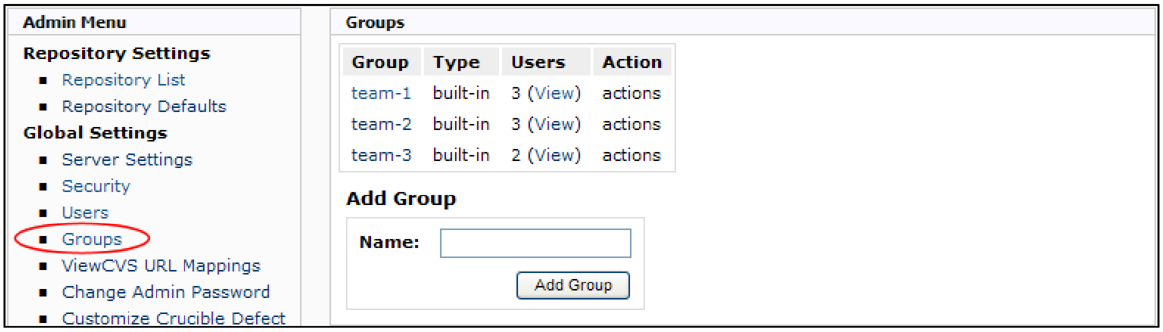
  Note that 'external' groups can only be created in your external directory.

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

Creating a User

There are two types of FishEye user accounts:

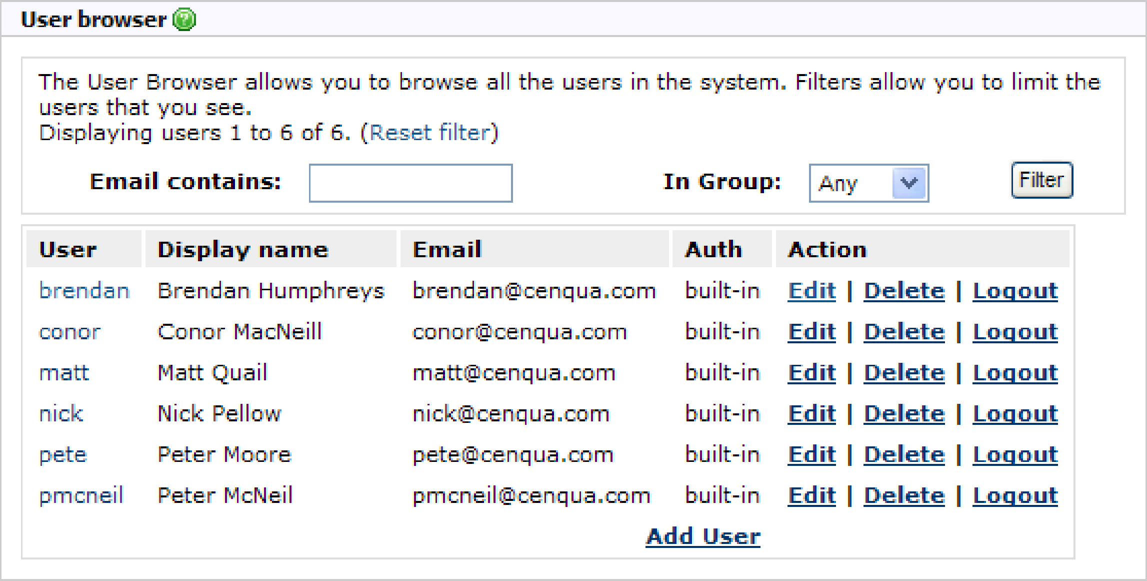
- 'Built-in' user accounts — these are stored in the FishEye database.
- 'External' user accounts — these are stored in an external directory (e.g. LDAP), if any are configured. See Configuring External Authentication Sources.

> **Note re external directories:**
> - new users can only be added in FishEye if they already exist in the external directory. FishEye does not modify your external directory.
> - if you have enabled 'auto-add' for your external directory, users who don't exist in FishEye will be automatically added the first time they login to FishEye.

To add a new user,

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

Screenshot: User Browser



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.

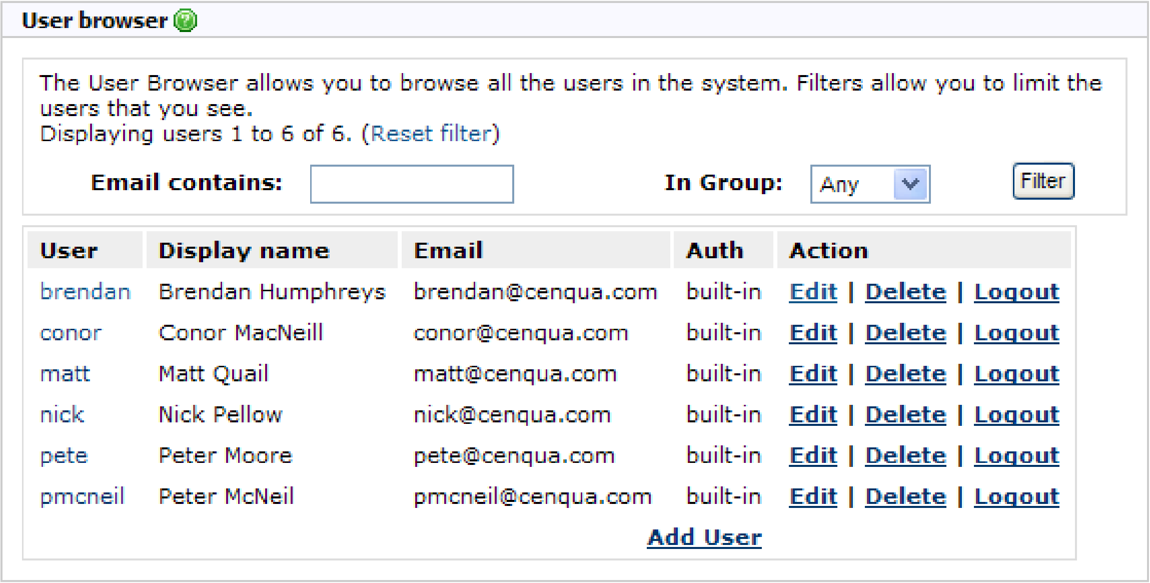
To deactivate a user's Crucible account,

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)

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 FishEye user accounts:

- 'Built-in' user accounts — these are stored in the FishEye 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,

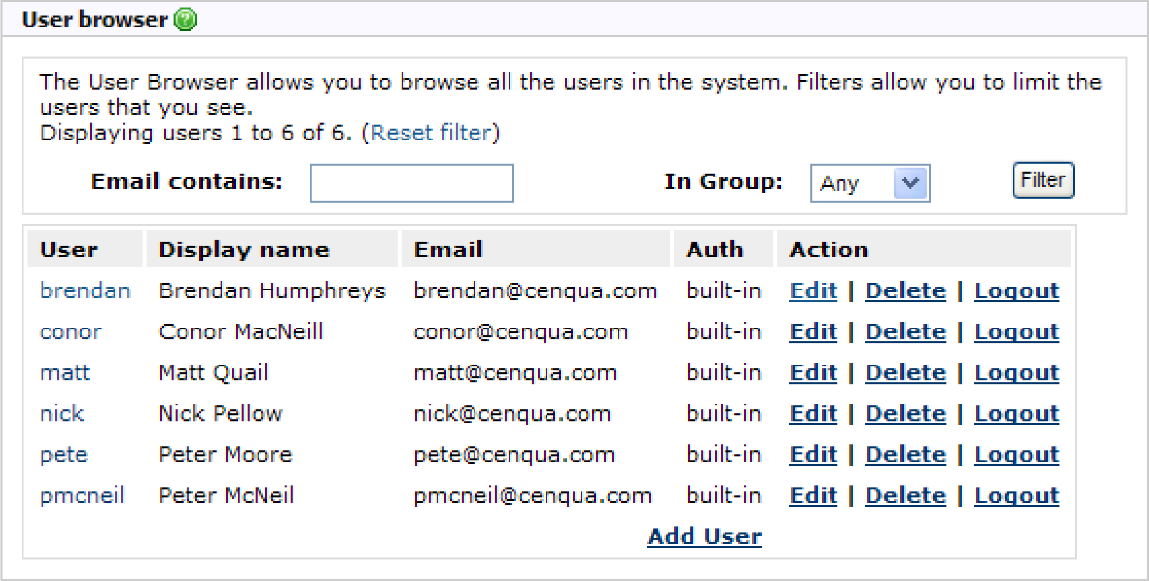
1. Click '**Users**' on the '**Admin Menu**'.
2. The '**User Browser**' screen will be displayed (see screenshot below), showing a list of users.
3. Locate the user and click the corresponding '**Edit**' link.

   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. You can update the following fields:
   - '**Display Name**' — type the user's display-name.

     ℹ 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.

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

Screenshot: 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 FishEye user accounts:

- 'Built-in' user accounts — these are stored in the FishEye 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 FishEye user accounts:

- 'Built-in' user accounts — these are stored in the FishEye 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.
   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 '**Rename**' link.
5. The '**Rename user**' screen will be displayed. Type the new username and click the '**Rename**' button.

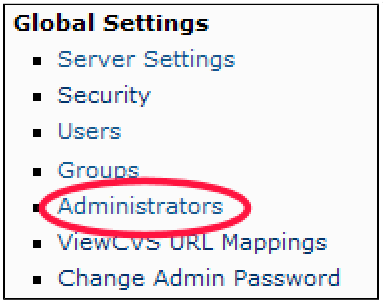
Granting Administrator Privileges

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:

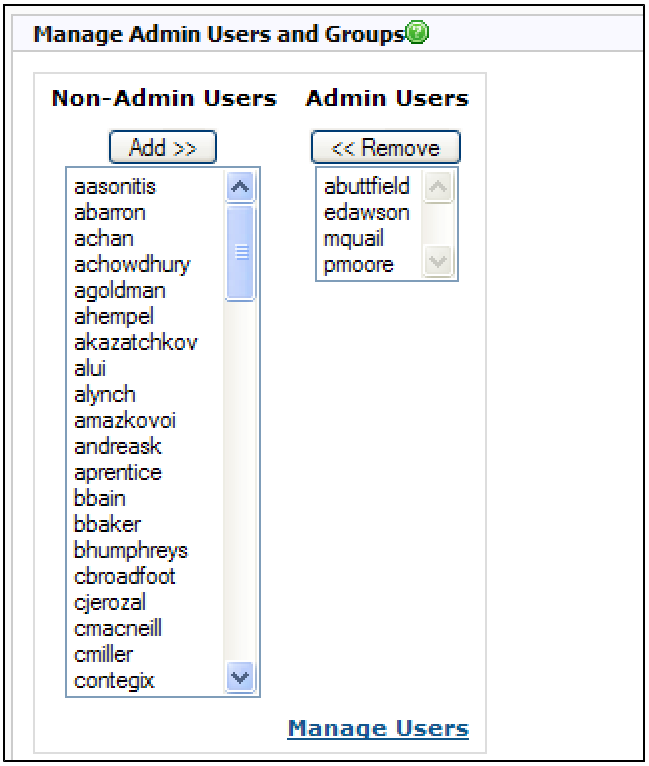
- Select the username in the '**Non-Admin Users**' column.
- Click the '**Add**' button.
- The username appears in the '**Admin Users**' column.

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



Manage Admin Users and Groups

Non-Admin Users	Admin Users
aasonitis	abuttfield
abarron	edawson
achan	mquail
achowdhury	pmoore
agoldman	
ahempel	
akazatchkov	
alui	
alynch	
amazkovoi	
andreask	
aprentice	
bbain	
bbaker	
bhumphreys	
cbroadfoot	
cjerozal	
cmacneill	
cmiller	
contegix	

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:

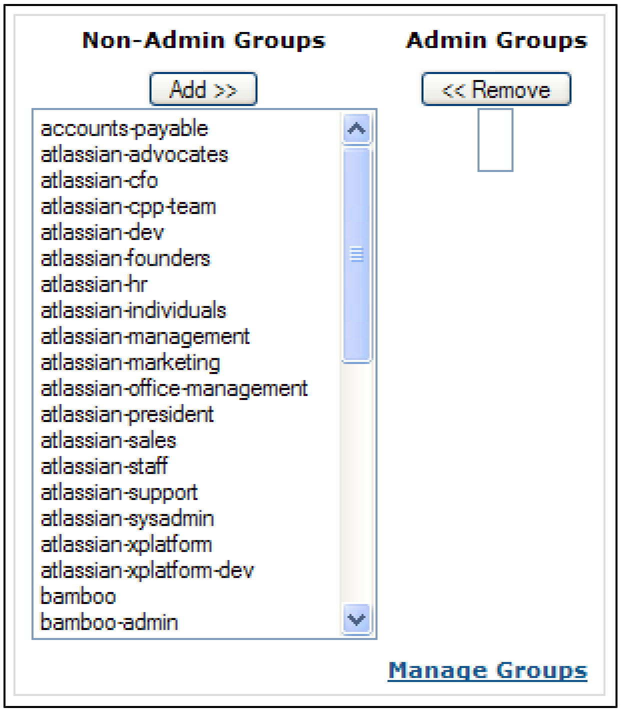
- Select the group name in the '**Non-Admin Groups**' column.
- Click the '**Add**' button.
- The group name appears in the '**Admin Groups**' column.

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.

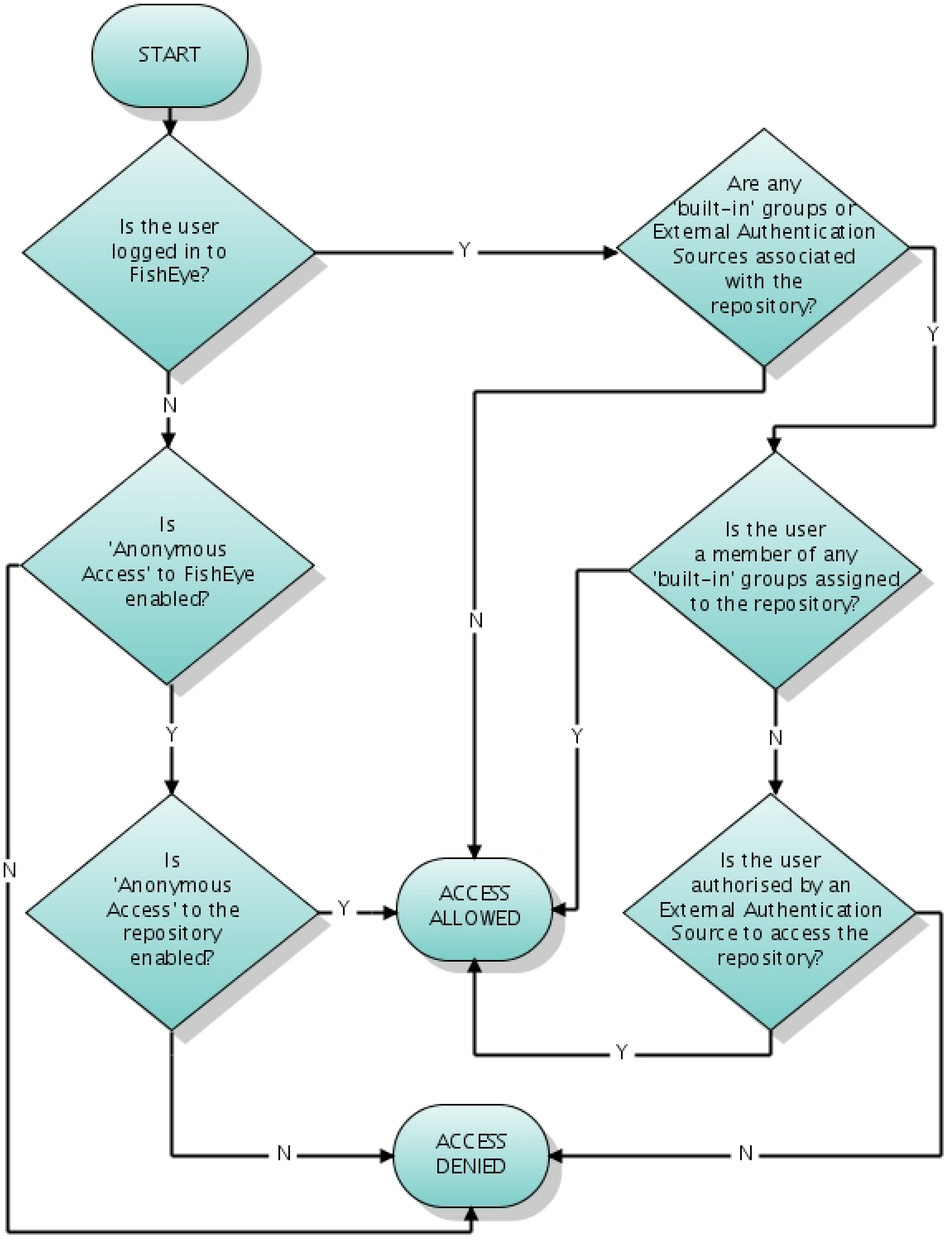
Note that users with identical names to existing users will not be imported. In the Users list, you can check whether each user is from the local database or loaded externally.

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**'.

Understanding security

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


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

Backing Up and Restoring Config Data

You can use one of the following methods to create a zip archive of all FishEye configuration files:

- Click '**Backup**' on the '**Admin Menu**'.
- Or use the `fisheyectl` script.

The FishEye backup and restore procedure requires you to use the `FISHEYE_INST` system variable.

A backup and restore allows you to move your FishEye instance to another location or host. It also allows you to upgrade to another version of

FishEye without losing any configuration or user data.

Backup

The backup includes all configuration settings, all FishEye user accounts and the FishEye license.
(it also includes reviews and comments, when running Crucible and FishEye together).

The following files will be backed up (when running FishEye only):

- `config.xml` (contains the FishEye license)
- `var/data/data0.bin`

> ⚠ No repository cache data will be backed up. This means that you will need to re-index the repository once you have restored the FishEye backup data.

Backup via the FishEye Administration Pages

1. Click '**Backup**' on the '**Admin Menu**'.
2. Click the '**Create Archive**' button to create a .zip file in the `$FISHEYE_INST/backup` directory.

Backup via the Command Line

The `fisheyectl` script takes a `backup` command and an optional `filename` for the backup archive. See Command-Line Options.

Scheduled Backup

See Running Scheduled Events

Restore

To restore from a backup:

1. Stop the FishEye server.
2. Unzip the backup file (created above) into the `$FISHEYE_INST` directory.

For example, say you have a `backup_20060101120000.zip` in `/tmp` and you have stopped FishEye, the restore procedure would be something like this:

```
$ cd $FISHEYE_INST
$ unzip /tmp/backup_20060101120000.zip
```

Advanced Administration Options

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

Command-Line Options

A FishEye instance can be managed using the `fisheyectl` script. Before running this script you need to ensure that you have set the JAVA_HOME environment variable, or that `java` is on the path.

Unix usage:

```
/FISHEYE_HOME/bin/fisheyectl.sh command [options]
```

Windows usage:

```
\FISHEYE_HOME\bin\fisheyectl.bat command [options]
```

The *command* parameter can be one of `run`, `start` or `stop` (see below). You can also find convenience scripts for running each of these commands, such as `run.sh` or `run.bat`.

run

The `run` command starts FishEye. This command runs FishEye in the foreground. It does not fork a background process.

start

The `start` command has the same options as `run`, but starts FishEye in the background.

Windows: FishEye will be run in a separate `cmd.exe` window.

Unix: FishEye will be run with `nohup` and the console output will be redirected to `$FISHEYE_INST/var/log/fisheye.out`.

Options (for both run and start):

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

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

`--Xtab-width nchars`	Specifies the number of spaces to use to represent a tab character. The default is 8.
`--Xdisable-dirtree-empty-checks`	When rendering the directory tree on some pages, FishEye calculates if each directory subtree is empty. For massive repositories, this calculation can cause the page to take a long time to render. This option disables the calculation that determines emptiness. ⚠ 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 or Perforce 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 [filename]
```

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

Options:

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

Configuring Indexing

Understanding How FishEye Indexes Repositories

FishEye's administration interface now allows you to configure the number of threads used for repository indexing. FishEye uses two types of indexing operation:

- **Initial indexing** - the complete scan of the repository that FishEye does when FishEye first accesses the repository, and
- **Incremental indexing** - the quick, background process that regularly scans repositories for changes.

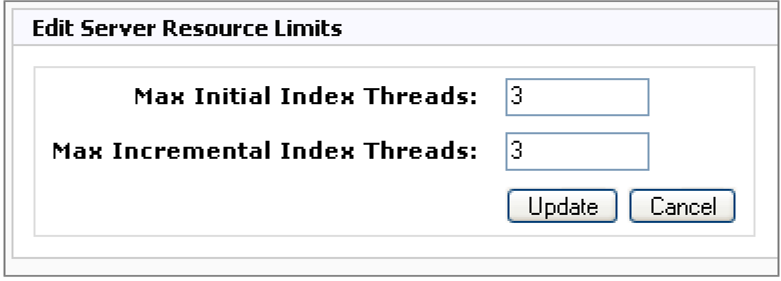
The number of threads available for these indexing operations is now configurable.

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

Setting FishEye's Total Threads

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

Screenshot: FishEye Multi-threading Configuration



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.

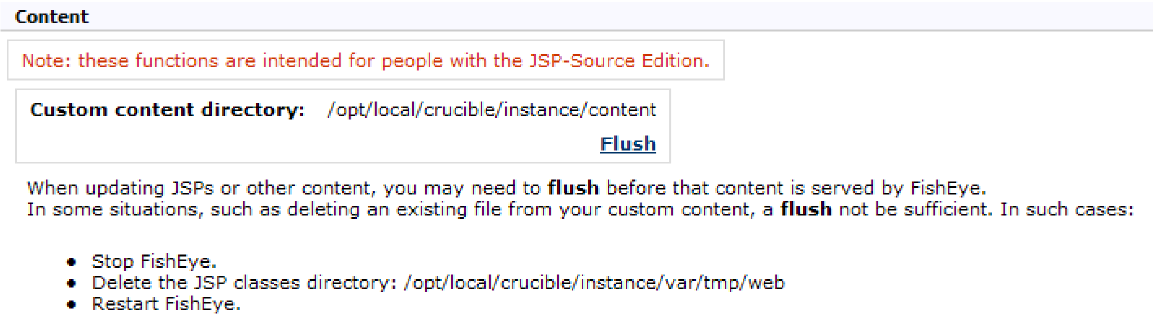
If you have a commercial license assigned to your account, you will see a '**source download**' option on your download page.

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.

Properties for FISHEYE_INST:

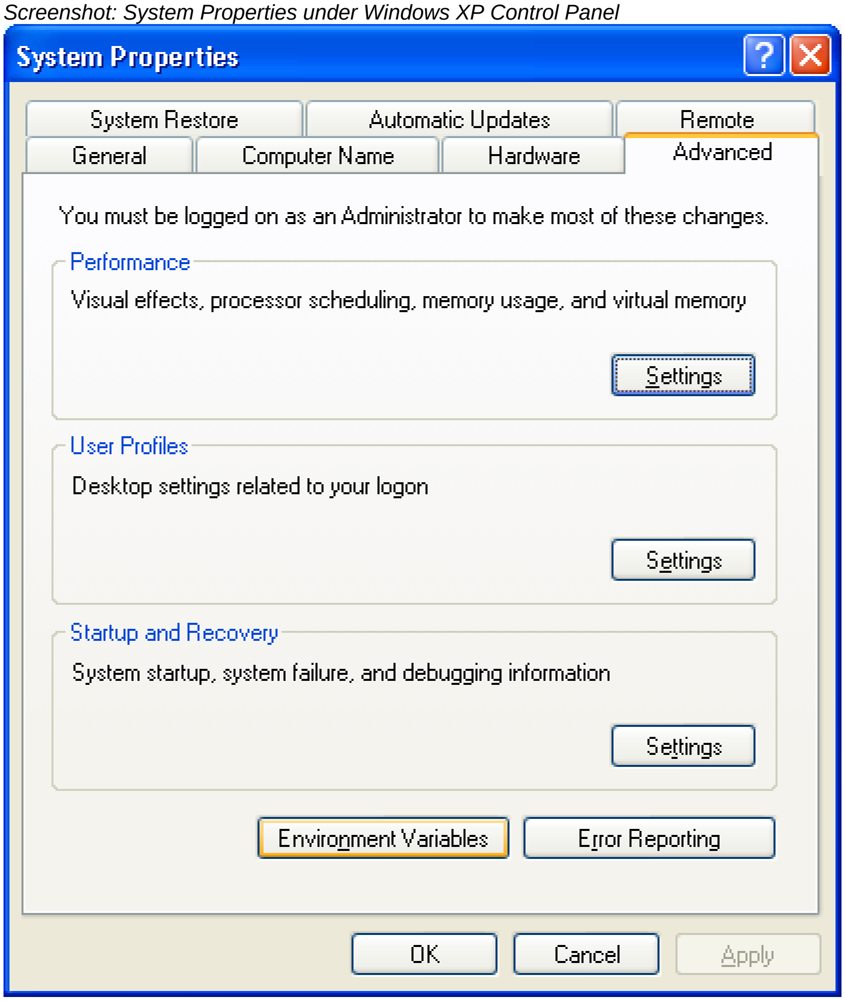
ⓘ Read more about using FISHEYE_INST in the Installation Guide.

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.

Performance

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

User Profiles

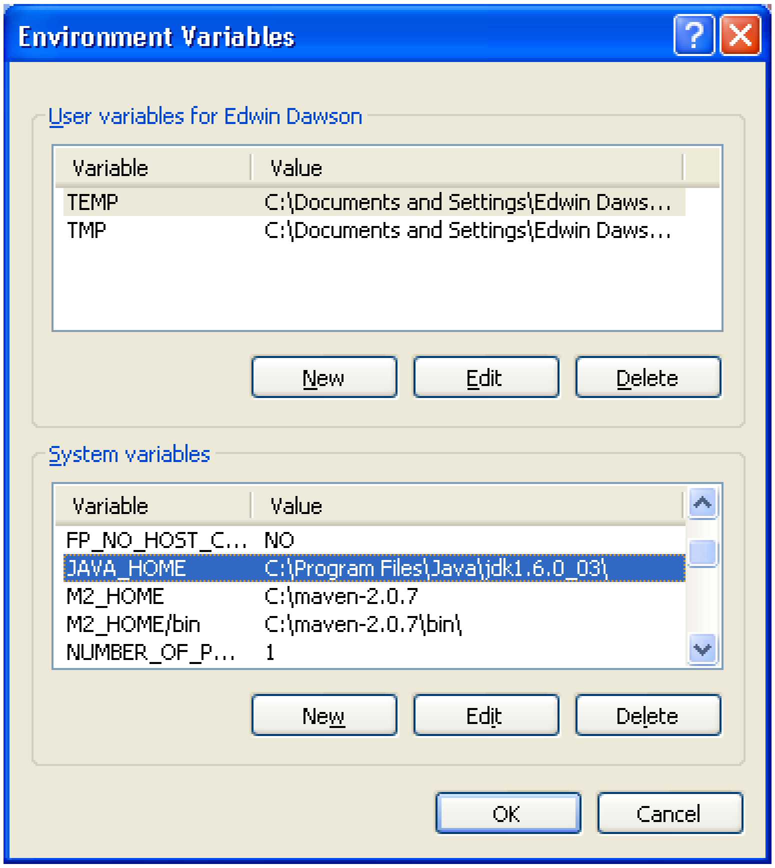
Desktop settings related to your logon

Startup and Recovery

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

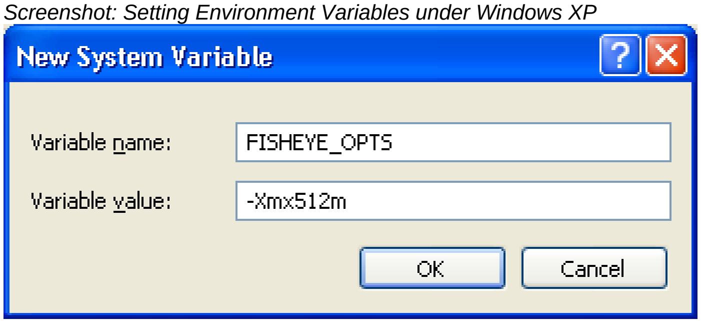
4. Click 'New'.
5. In the 'Variable name' field, enter the name of the environment variable, for example

```
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.

ℹ If you are experiencing memory errors in FishEye, see Fix Out of Memory errors by increasing available memory.

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, you will need to increase the heap size and restart FishEye. In this situation, try increasing your FISHEYE_OPTS variable to 512MB. Setting FISHEYE_OPTS is similar to the instructions for setting JAVA_HOME.
You can follow the same procedures, only using **FISHEYE_OPTS** in the '**Variable name**' field, and using the following '**Variable value**':

```
-Xmx512m
```

(this requires a reboot under Windows)

To do the same thing under the Linux console, you can type the following:

```
export FISHEYE_OPTS=-Xmx512m
```

This would need to be set to run on boot, or set in your FishEye startup script, if you have one.

For more information, read the detailed instructions on setting environment variables.

Improving Initial Scan Performance

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,

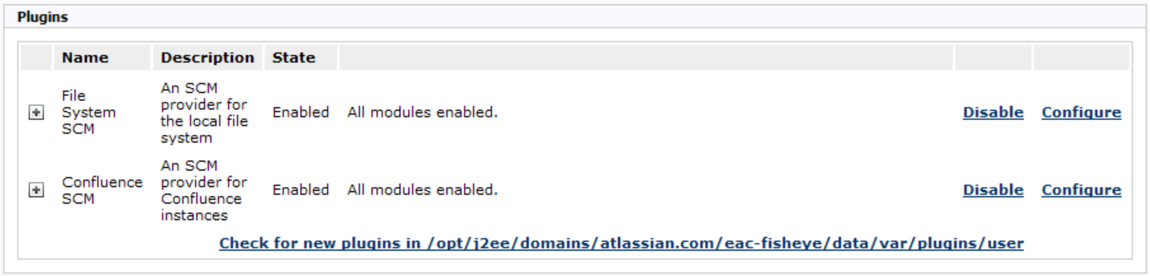
1. Click '**Plugins**' on the '**Admin Menu**'.
2. The '**Plugins**' screen will be displayed (see screenshot below).
3. Locate the plugin you wish to enable, and click the '**Enable**' link.
    - To view, enable or disable individual modules within a plugin, click the '**+**' sign at the left of the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

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.

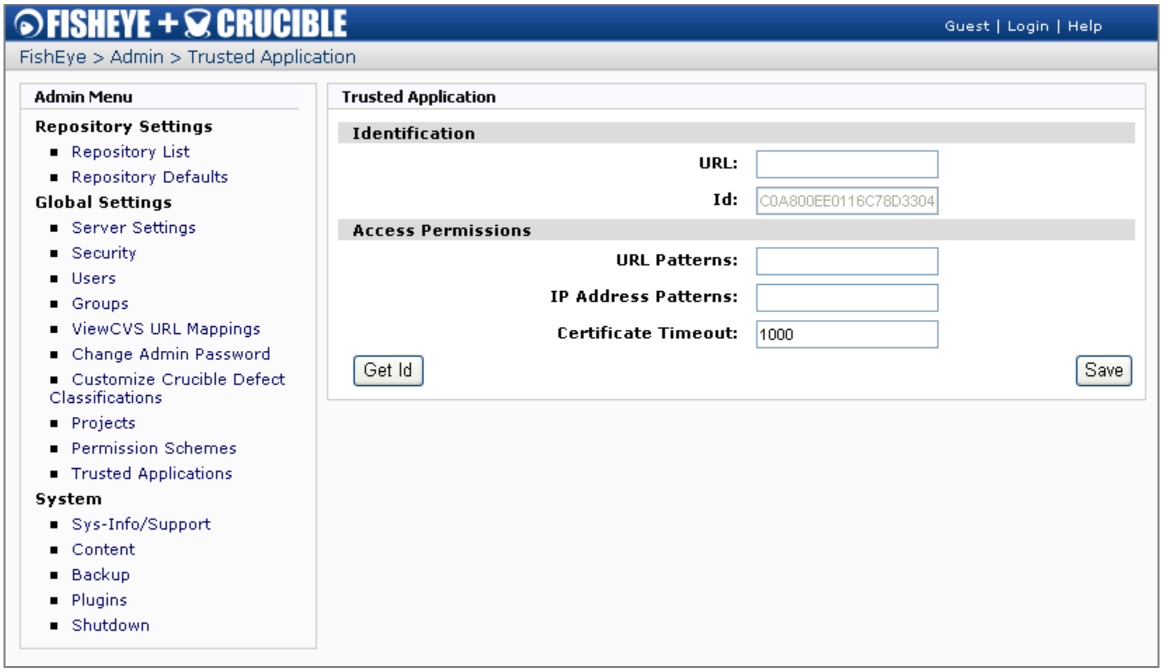
Before you begin, note that configuring a trusted application requires the transmission of sensitive data. To prevent ' man-in-the-middle attacks', it is recommended that you use an encrypted SSL connection while configuring a trusted application.

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



Trusted Application

Identification

URL:

Id: C0A800EE0116C78D3304

Access Permissions

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).

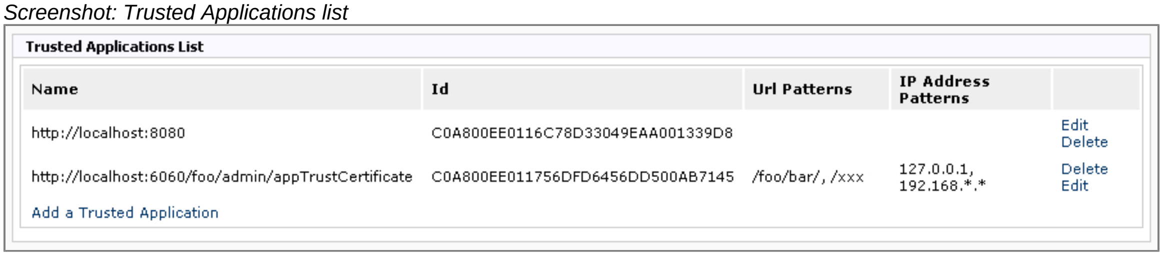
> ⚠ A shorter time out is more secure, but if set too short, it may cause valid requests to be rejected on slower networks.

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.

Customising the Welcome Message

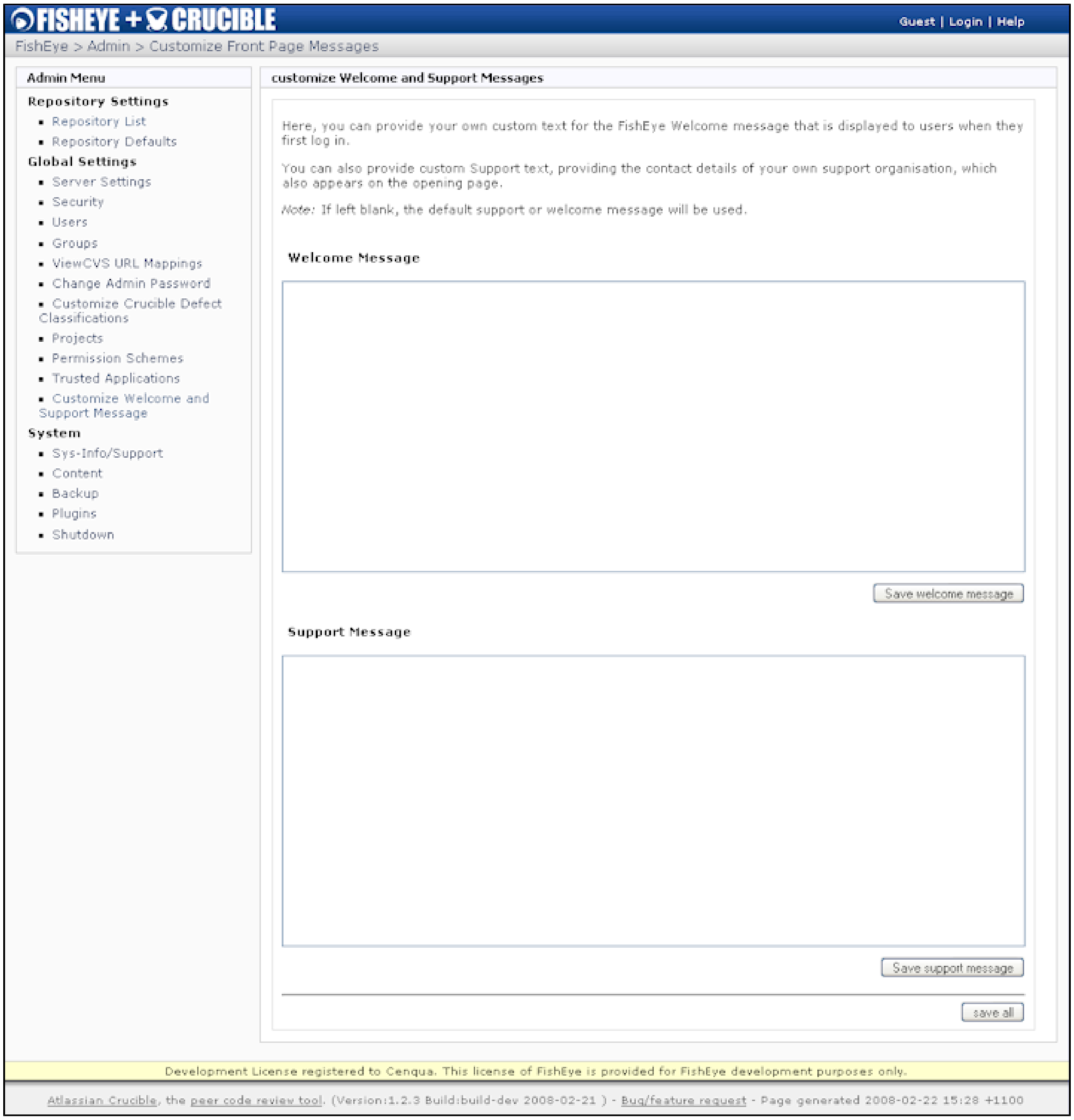
To customise the welcome message, access the administration page, and click '**Customize Welcome & Support Message**' under '**Global Settings**' on the left navigation bar.

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



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>
```

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.

> ℹ 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*

Software Update Notifications

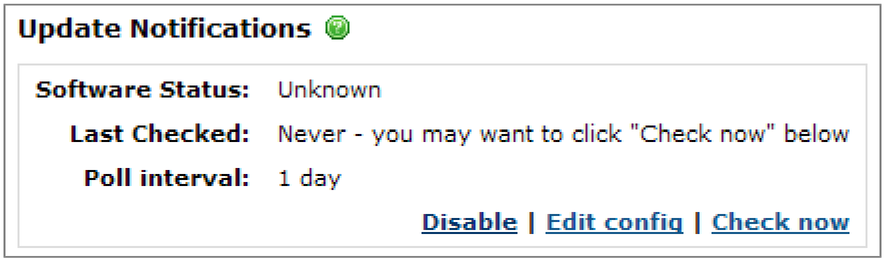
FishEye (and Crucible) can detect and notify you of new versions of the program. In the Admin screen, you can set a background thread to poll Atlassian's servers and report when a new version of FishEye or Crucible has been released.

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



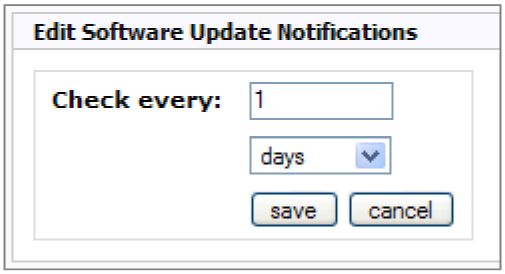
Update Notifications

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

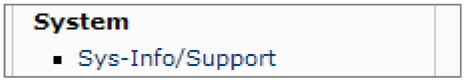
days

Contacting Support directly via FishEye

You can contact Atlassian support directly from the FishEye Admin interface.

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

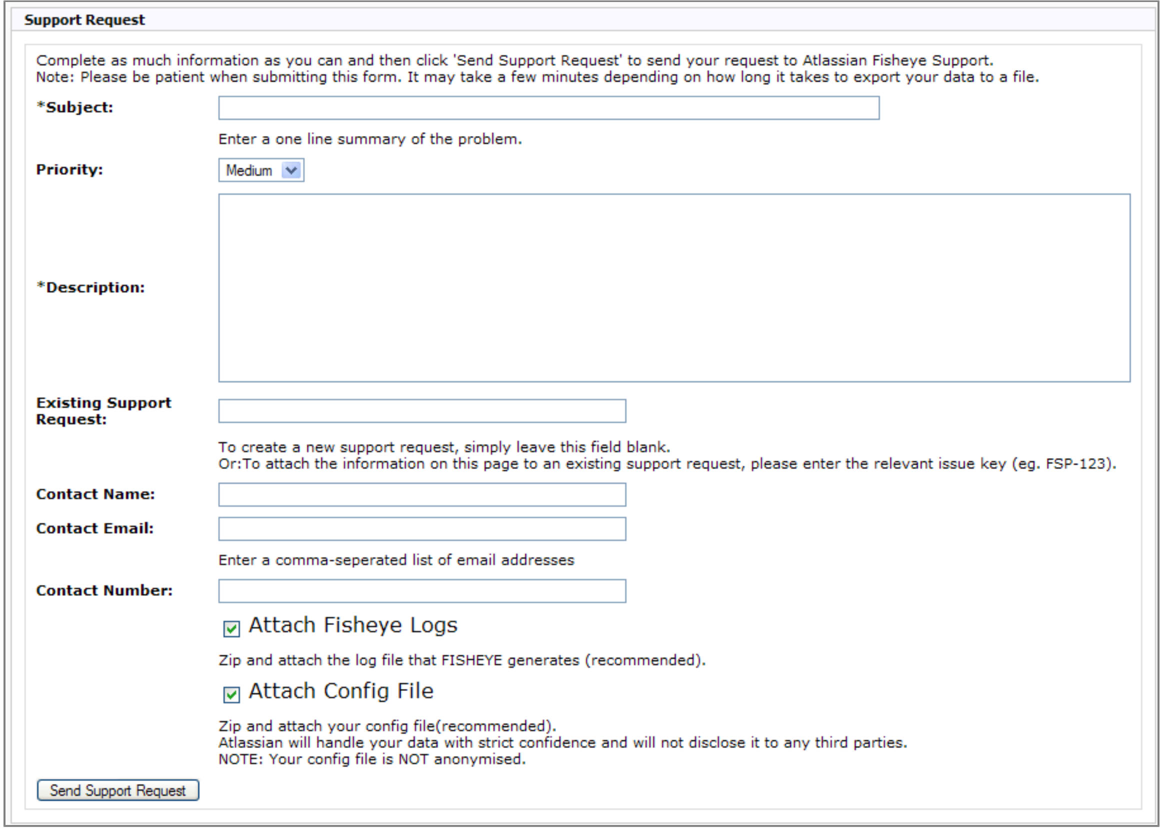
Screenshot: The Sysinfo/Support Menu Option



The **System Info/Support** page loads. On this page, you can fill out a web form which will automatically send an email to Atlassian Support, attaching your FishEye logs and configuration file (if you wish).

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

Screenshot: The System Info/Support page



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.

JIRA Integration in FishEye

This page contains instructions for setting up JIRA integration in FishEye.

ℹ JIRA is Atlassian's issue tracking product, which can be used to manage projects and associated work.

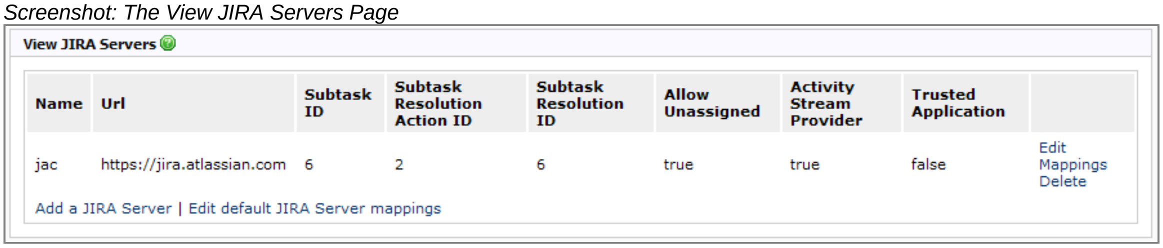
On this page:

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

Opening the Administration Screen for JIRA Integration

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

Screenshot: The View JIRA Servers Page



View JIRA Servers

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

Add a JIRA Server | Edit default JIRA Server mappings

On the View JIRA Servers page, you can carry out the following operations:

General Operations

- Add a new JIRA server.
- Edit default JIRA server mappings.

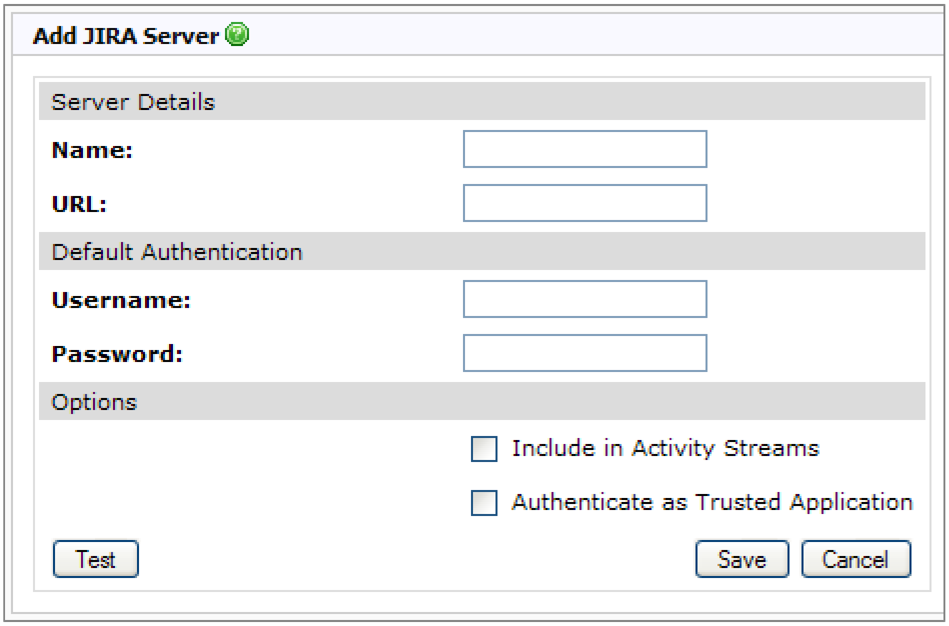
Operations on Existing Servers

- Edit settings for an existing JIRA server.
- Edit mappings for an existing JIRA server.
- Delete an existing JIRA server.

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.



Add JIRA Server

Server Details

Name:

URL:

Default Authentication

Username:

Password:

Options

□ Include in Activity Streams

□ Authenticate as Trusted Application

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
Subtask Type ID	Text Field		No
Subtask Resolution Action ID	Text Field		No
Subtask Resolution ID	Text Field		No
Allow Unassigned	True/False Button		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 FishEye has been configured as a Trusted Application in JIRA.)	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. (Requires a JIRA instance with the streams plugins installed)	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

Operations on Existing Servers

Edit settings for an existing JIRA server.

Edit mappings for an existing JIRA server.

Delete an existing JIRA server.

For more information on setting up JIRA mappings and non-JIRA servers, see Linkers.

FishEye Development Hub

If you're doing custom development with FishEye, you've come to the right place.

FishEye Developer Resources

- Developer Documentation - Get started developing plugins for Atlassian Products.
- FishEye Plugin Types - Learn about the different kinds of plugin technologies.
- Live Code Examples - See the source of real plugins in the wild.
- Plugin Hosting - Atlassian can host your plugin on our servers.
- The Atlassian Developer Blog - Find out more about what our developers are up to.

Documentation

Here you'll find everything you need to code up a storm with FishEye. This includes guides for setting up your environment, building a project and creating a plugin, with real-world examples you can try.

How to Build a FishEye Plugin

How to Build a FishEye Plugin - start here to learn how to set up your development environment, create a plugin template and start coding.

Development Platform for FishEye

FishEye API Javadocs

The FishEye Remote API

FishEye Developer FAQ

FishEye Plugin Types

FishEye plugins come in a variety of flavours, read on to see how the plugin technology interacts with the core of FishEye and what rules can be bent, or possibly *broken* in this world.

Servlet Plugin Modules

- **Servlet Plugin Module**

Spring Component Plugin Modules

- **Spring Component Plugin Module**

Live Code Examples

Below is a list of real-world plugin examples that showcase the various sides of FishEye development. The following items are an excellent resource for the Atlassian developer community. Feel free to investigate these examples, hack them to pieces, or use them as inspiration to really innovate.

FishEye Plugin Examples

- **Developer Report Plugin**
- **FishEye Client for Eclipse**

Plugin Hosting

Atlassian can host your plugin development project. We'll provide a Subversion repository, Confluence space and a JIRA project. Find out more.

The Atlassian Developer Blog

For up-to-date news and opinions from the FishEye, FishEye and other Atlassian development teams.

Atlassian Developer Blog RSS

Agile With A Remote Product Owner
We Are From Mars All agile methodologies stress the need of co-locating development with the customer's representative - the Product Owner - or at least, having them in close proximity -...

Make Your Code Agile: Refactoring
In this post I define and promote refactoring in productivity terms through controlling code complexity.

Help us Integrate Confluence with Alfresco
At Atlassian we're always looking for ways to expand the utility and functionality of our products. Sometimes this means we develop that code ourselves, and sometimes it means our community fills in...

Hamcrest saves your soul - Now with less suffering!
In a previous post I described how Hamcrest can save your soul. After writing that, it was pointed out that you probably don't need to suffer so much boiler-plate to save your...

How to determine the context your macro is being rendered in
For Confluence 2.10 (remember that?), we converted the display of the Jira Issues Macro from using a static HTML table to using a table infused with jQuery goodness. Now we could add...

Also see the Crucible Development Hub.

Developing Fisheye Plugins

Introduction

This page contains the basic information to guide plugin developers in coding and building a plugin for FishEye. There is also Crucible documentation on this topic, if you wish to also access Crucible with your plugin.

FishEye uses the standard Atlassian Plugins framework, so many of the tasks involved in developing a plugin for FishEye are the same as for other Atlassian products.

The differences are:

- The set of plugin types available.
- The API available for plugins to interact with the FishEye application.

Building a FishEye Plugin

The simplest way to build a FishEye plugin is via Maven.

Atlassian provides an Archetype for Fisheye/Crucible plugins.

You can create a Maven 2 project containing a sample Servlet Plugin Module with the following command:

```
mvn org.apache.maven.plugins:maven-archetype-plugin:1.0-alpha-7:create \
  -DarchetypeGroupId=com.atlassian.maven.archetypes \
  -DarchetypeArtifactId=crucible-plugin-archetype \
  -DarchetypeVersion=1-SNAPSHOT \
  -DremoteRepositories=https://maven.atlassian.com/repository/public/ \
-DgroupId=com.foo -DartifactId=foo-crucible-plugin
```

This will create your project in a subdirectory of your current directory named `foo-crucible-plugin`. Change into that directory (`cd foo-crucible-plugin`). You can create the plugin jar with the command `mvn package`, and install it in a running Fisheye or Crucible instance by copying `target/foo-crucible-plugin-1.0-SNAPSHOT.jar` to the `var/plugins/user` directory of your Fisheye/Crucible instance.

FishEye Plugin Module Types

FishEye supports several different kinds of plugin modules, listed below.

FishEye Web Items

Web UI plugin modules allow you to add links, tabs and sections of links to the Fisheye user interface. See FishEye Web Items for more details.

Servlet Plugin Modules

Create a servlet which is deployed to the same web application context as Fisheye/Crucible. See Servlet Plugin Modules for more details.

Spring Component Plugin Modules

Create a plugin that makes use of Spring components. See Spring Component Plugin Modules for more details.

The FishEye API

Your plugin will need to use The FishEye Remote API to retrieve data from FishEye and to perform operations on it, such as retrieving a changeset.

Debugging your plugin

You can start FishEye in debug mode with the environment variable setting:

```
export FISHEYE_OPTS="-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005"
```

This allows you to connect your IDE to the debugger listening on port 5005.

Source Code Links

See the following pages for code from example plugins:

- Developer Report Plugin
- FishEye Client for Eclipse

Further Information

- Servlet Plugin Module
- Spring Component Plugin Module

Plugin Module Types

The plugin functionality described here is **not** available in Fisheye 1.4. It will be available in Fisheye 1.5.

The following types of plugin modules are available to Fisheye/Crucible plugins

- Servlet Plugin Module
- Spring Component Plugin Module

Servlet Plugin Module

Module Descriptor

A Servlet plugin module descriptor looks like this:

```
<servlet name="My Servlet" key="myServlet" class="com.yoydyne.MyServlet"  adminLevel="system">
        <description>Does things with stuff</description>
        <url-pattern>/my-servlet/*</url-pattern>
        <init-param>
            <param-name>foo</param-name>
            <param-value>abc</param-value>
        </init-param>
        ...
</servlet>
```

This makes your servlet available at the URL `http://<your server>/<your context>/plugins/servlet/my-servlet`.

The servlet instance is auto-wired by Spring, so it can have Spring beans injected from the Fisheye/Crucible context.

The `adminLevel="system"` attribute is optional – if supplied, accesing the plugin requires that the user is authenticated as an administrator.

Fisheye Page Decorators

When plugin authors create pages in FishEye and Crucible, they may need to be aware of decorators.

On this page:

- What is a Decorator?
- Decorator Code Example
- Table of Decorator Parameters

What is a Decorator?

A decorator creates standard parts of the page and inserts the content created by a plugin servlet in the appropriate place. For example, the `atl.general` decorator provides the standard header and footer, while the `atl.admin` decorator also provides the left hand column of administration links, with the plugin generated content in a section to its right.

So, a servlet which is rendering a URL from a web item in the `system.admin` location would request the `atl.admin` decorator.

All servlet plugins whose response is rendered as a Fisheye/Crucible page should specify a decorator in the response. Meta tags in the `head` of the HTML page are used to choose a decorator and provide it with any parameters it needs.

Decorator Code Example

The decorator name and parameters are given thus:

```
<head>
...
  <meta name='decorator' content='fisheye.userprofile.tab'/>
  <meta name='profile.tab.key'
content='com.atlassian.crucible.example.plugin.event.crucible-example-event-plugin:test-profile-tab'/>
...
</head>
```

The value given by the `content` parameter of the `profile.tab.key` is the plugin module key of the web-item which creates the tab we want, shown as selected when this page is rendered.

Table of Decorator Parameters

Decorator	Description	Parameters
`atl.general`	Provides standard header and footer.	None
`atl.admin`	Provides left hand column of administration links.	None
`fisheye.userprofile.tab`	Content rendered as a single tab on the user profile page.	`profile.tab.key` The plugin module key of the web item which created the tab.
`crucible.review.tab`	Content rendered as a single tab on the review page.	`review.permId` The permId of the review being displayed. `profile.tab.key` The plugin module key of the web item which created the tab.

In the table above, `crucible.review.tab` only applies when creating plugins for Crucible.

Spring Component Plugin Module

Module Descriptor

To create a Spring component in a plugin configure a module like this:

```
<spring key="componentName" class="com.example.MyComponent">
  ... standard Spring configuration XML ...
</spring>
```

The above module is equivalent to having the following code in a Spring context file:

```
<bean id="componentName" class="com.example.MyComponent">
... standard Spring configuration XML ...
</bean>
```

Note that the spring component modules are created in order of their declaration, and that a component created by an earlier module cannot depend on a component created by a later module.

Documentation for FishEye Development

In this section:

- FishEye's URL Structure

See also Documentation for Crucible Development

FishEye's URL Structure

This page contains information about the FishEye URL structure for plugin developers. Knowing the structure, you will be able to construct hyperlinks for use in plugins or gadgets and find API specifications for your version of Fisheye.

On this page:

- FishEye-Specific URL Structure
    - Browse FishEye Repositories
    - Search FishEye Repositories
    - RSS
- API Help and Resources
    - The Original API
    - The REST API (introduced in version 1.6)
    - Static content URL
    - RDIFF Help

There is also a page about the Crucible structure.

FishEye-Specific URL Structure

Browse FishEye Repositories

This opens a window to browse the specified FishEye repositories.

In the example below, insert the desired constraints in place of "**CONSTRAINT1**" and "**CONSTRAINT2**" (you can have multiple constraints, separated by commas), the desired repository name in place of "**REPNAME**" and the repository path in place of '**PATH**'.

Basic form

```
/browse/~CONSTRAINT1,CONSTRAINT2/REPNAME/PATH
```

Example with typical values

In the example below, the constraint "**br**" stands for the **branch** you want to constrain to, "**CLOV**" is the repository and "**src**" is the path inside the repository.

```
http://example.com/fisheye/browse/~br=clover2/CLOV/src
```

There are four constraint types that can be used: Branch ('**br=**'); Author ('**author=**'); Tag ('**tag=**'); Date + Time ('**date** =YYYYMMDDTHH:MM:SS/repname').

Example using date and time constraint

In this example, from left to right; the date is expressed; a '**T**' is inserted to denote the beginning of the time data; finally, the repository name is specified.

```
http://example.com/fisheye/browse/~date=20090119T23:15:01/CLOV
```

Search FishEye Repositories

This opens a window with search results from the specified query.

In the example below, insert your query parameters in place of '**QUERY**', separated by commas; Enter the name of your repository in place of '**REPNAME**'; insert the columns you want to display in place of '**CSV**'.

Basic form

```
/search/REPNAME/?QUERY&CSV
```

Example with typical values

In the example below, the query is defining a filename ('test') then, three columns are defined in which to return data (in this case, path, author and changeset ID).

```
http://example.com/fisheye/search/CLOV/?filename=test&col=path&col=author&col=csid
```

RSS

This opens an RSS feed defined by the supplied constraints.

In the example below, insert your constraints in place of '**CONSTRAINT1**' and '**CONSTRAINT2**'; also insert the name of your repository in place of '**REPNAME**'; the path inside the repository in place of '**PATH**'; finally, insert the authentication token for this user in place of '**AUTHTOKEN**'.

Basic form

```
/changelog/~rss,CONSTRAINT1,CONSTRAINT2/REPNAME/PATH?FEAUTH=AUTHTOKEN
```

Example with typical values

In the example below, '**br**' is the branch inside the repository; '**feedspan**' is the amount of time to span; '**feedmax**' is the number of items to display.

```
http:
//example.com/fisheye/changelog/~rss,br=1.6,feedspan=2mo,feedmax=10/FE/rss.xml?FEAUTH=username:1324:e79b3
```

Note: You can retrieve an authToken for the current user via the API (go to `/api` in FishEye for more details).

API Help and Resources

The Original API

This opens a window showing the API documentation that ships with the product.

Basic form

```
/api
```

Example with typical values

```
http://example.com/fisheye/api
```

The REST API (introduced in version 1.6)

This opens a window with an XML-based description of all REST commands and their parameters. This page is in WADL format (Web Application Description Language).

Basic form

```
/rest-service/application.wadl
```

Example with typical values

```
http://example.com/fisheye/rest-service/application.wadl
```

Static content URL

For all external assets such as image files or gadgets, use this in the URL to avoid the random string mechanism that normally causes auto-refreshing of these assets.

Basic form

```
/currentstatic/...
```

Example with typical values

```
http://example.com/fisheye/currentstatic/logo.ico
```

ℹ Note that this redirects to the actual address (with a HTTP code of 302), which is randomly generated for each instance, for example `/static/kbn4ib/images/...` for browser caching purposes.

RDIFF Help

This opens a window with the embedded documentation on FishEye's RDIFF feature.

In the example below, insert the name of your repository in place of '**repname**'.

Basic form

```
/rdiff/repname
```

Example with typical values

```
http://example.com/fisheye/rdiff/CLOV
```

Looking for a page on the Crucible URL structure? Click here.

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.

Fisheye Plugins

- Developer Report Plugin — Primarily as a proof of concept, this plugin shows what I've found you can do with FishEye plugins – hopefully this will help and encourage more developers to dive in and see what they can do.
- FishEye Client for Eclipse — Basic Fisheye Integration for Eclipse that includes a repository browser, history view and search capabilities

Developer Report Plugin

> ⚠ This plugin is more of a platform for development and showing what is possible – with some better documentation / examples, I'm sure the reports could be enhanced to provide improved stats.

Name	Developer Report Plugin
Version	1.0
Product Versions	1.5.1
Author(s)	Dan Hardiker
Homepage	http://confluence.atlassian.com/display/CODEGEIST/Developer+Report+Plugin
Price	FREE
License	BSD
Source	http://svn.atlassian.com/fisheye/browse/public/contrib/fisheye/developer-report-plugin/trunk?%40hideDeletedFiles=Y
Download JAR	developer-report-plugin-1.0.jar
Download Source	trunk | 1.0

Description/Features

Primarily as a proof of concept, this plugin shows what I've found you can do with FishEye plugins – hopefully this will help and encourage more developers to dive in and see what they can do.

I am writing up my experiences as a Diary of a FishEye Hacker (and part 2), which includes suggestions to the FishEye developers and 3rd party developers alike.

Here's the general summary:

- Be prepared to be frustrated, very frustrated. There is no documentation or source and the 2 examples won't be much use. **FishEye is so not ready for you to plugin to.**
- If your a little crazy and very persistent, you can plugin your own XWork actions.
- I did not find a clean way to change the classloading for JSPs or Velocity files - so resources should go into somewhere unique in `$FISHEYE_INST/content`
- If you're careful, you can build your own `ContainerManager` / `ComponentManager` (as found in Confluence / JIRA)
- There is no decoration and JSP tags & includes are used heavily - I've not found it practical to try to reuse these components.
- There are no UI hooks outside of the Admin area, so if you're hoping to add a tab, or add a widget, or add a block somewhere from a plugin - think again.
- While the StateAware interface exists for plugins to use, no plugin module respects it.

Moral of this adventure?
If at first you don't succeed, grab a few beers. It helps prevent the *throwing of laptop in anger* effect.

I hope my code serves as a useful example and platform for bigger & better things, and my write up serves as a nudge (or kick up the ass) for the FishEye developers to make the API & plugin subsystem properly usable.

Why not just use Servlets?

Well, as anyone who is used to writing plugin actions in Confluence / JIRA should tell you, writing raw servlets is laborious. WebWork/XWork gives you a lot for free:

1. You get a command pattern implementation.
2. You can build your own actions & results, just like xwork/webwork1 plugins in Confluence/JIRA.
3. Your actions are wired and incoming parameters converted.
4. You get an OGNL stack for use, almost transparently.
5. You get a validation framework at your disposal.
6. You can add your own interceptors and result types in addition to the given ones.
7. You can build you actions in an openly testable way.

In short, you can build your plugins smaller, simpler and quicker, leading to more testable and maintainable code.

Usage

Simply navigate to `/devreport/home.do`, where `/` is the root of your FishEye installation.

Using `/fisheye/` as your context is pretty standard, so you may need to use `http://www.domain.dom/fisheye/devreport/home.do`.

Installation

1. Copy the `developer-report-plugin-xxx.jar` to the `var/plugins` folder
2. Restart FishEye (you might get away with reloading the plugins in the Admin UI, but I wouldn't recommend it)

Uninstallation

Please note that this plugin does some fairly hefty things, such as:

1. It burrows it's way through Spring
2. It tinkers with class loaders
3. It rewrites the XWork/WebWork configuration
4. It hijacks XWork object creation
5. It extracts it's resources into `$FISHEYE_INST/content/devreport`

All of theses will be reverted by simply removing the plugin and restarting FishEye, with the notable exception of:

1. You may wish to delete the `$FISHEYE_INST/content/devreport` folder

TODO List

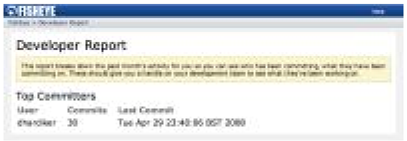
State	Task
✓	Fix up the report (it needs sorting and limiting)
✓	Write some tests (model + action tested)
⚠	Find some Valium

Version History

1.0 - Initial Version

Screenshots

Screenshots

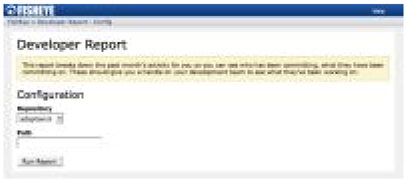


Developer Report

[illegible]

Top Committers

User Commits Last Commit
dhandler 30 Tue Apr 29 23:40:86 0ST 2080



Developer Report

[illegible]

Configuration

Repository

Path

Run Report

Example Report
Setup Form

Other Adaptavist Entries

User Security Management Plugin — An enhancement for the Confluence user management system, to prompt better security practices - including email verification and admin vetting of signups

Custom News — An alternative to Confluence's blog posts macro to aid with customisation

Statistical Analysis Plugin — Confluence has lacked a cluster-ready, enterprise scaleable, remotely accessible statistically gathering and analysis plugin ... not any more!

Developer Report Plugin — Primarily as a proof of concept, this plugin shows what I've found you can do with FishEye plugins – hopefully this will help and encourage more developers to dive in and see what they can do.

Attachment Download Plugin — Adds a servlet so you can download attachments from a page without needing to know the ID.

Synonym Plugin — A search extractor for Confluence to inject synonyms for acronyms, words or phrases into the index to aid with searching

Developer Report Plugin — Primarily as a proof of concept, this plugin shows what I've found you can do with FishEye plugins – hopefully this will help and encourage more developers to dive in and see what they can do.

Ranking Macro — Yet another macro for voting/rating/ranking pages, this one is uniquely different to the others by providing a macro for ranking pages with a 'was this page useful' style approach, tracking only positive answers

Plugin Message Client — A library which when included as an extracted dependancy will allow java communication between the classloaders of the installed plugins

Insert Picture Plugin — A in-place image management widget for Confluence to help with image attachment manipulation

JIRA Visitor Plugin — If you've ever found yourself commenting simultaneously as someone else with the same information, or been faced with the dreaded "workflow has already changed" message, or just thought "I wonder if anyone else is viewing this issue right now" - then this is for you.

Developer Report Plugin — Primarily as a proof of concept, this plugin shows what I've found you can do with FishEye plugins – hopefully this will help and encourage more developers to dive in and see what they can do.

FishEye Client for Eclipse

Name	FishEye client for Eclipse
Version	1.0.0
Product Versions	Fisheye 1.5, Eclipse 3.4 (possibly works in 3.3)
Author(s)	Brock Janiczak
Homepage	
Price	free
License	BSD

JavaDocs	
IssueTracking	
Download JAR	fisheye-1.0.0.zip
Download Source	fisheye-src-1.0.0.zip

Description/Features

Basic Fisheye Integration for Eclipse that includes a repository browser, history view and search capabilities. The client is built on top of a pure OSGI bundle that can be used for other integration projects. The service bundle also contains an extensive object based query model that allows for easy composition of queries and processing of their results.

Usage

Installation

Download and unzip fisheye-1.0.0.zip into an extension location or straight into your Eclipse installation. Sorry, no update site.

Creating a connection to your fisheye repository

1. Open the fisheye view (in the fisheye category)
2. Add a connection to your repository **note:** your server must support remote anonymous connections

Viewing history of a file

1. Double click on a path in the repository view to see the history of the file
2. The history page supports linking with the selection in the repository browser

Searching for changes

1. Open the standard search dialog
2. Switch to the fisheye tab
3. Select a connection
4. select a repository
5. Enter search parameters

TODO

1. Heaps of stuff has been implemented in a non ideal way
2. Increase test coverage (most of the service bundle is tested)
3. Support authenticated servers
4. Icons needed in a lot of places
5. No validation on date inputs
6. Search results should be displayed in a tree using the appropriate grouping (selected on query page)
7. Include history log in search results
8. Link fisheye to a team provider to allow auto discovery of fisheye servers
9. Integrate with Crucible...

Version History

1.0.0 Initial Version

Screenshots

Screenshots

There are no images attached to this page.

The FishEye Remote API

For developers who are interested in accessing the FishEye functionality remotely, this page describes the methods, data types and structures for accessing the FishEye Remote API.

Additional documentation is available as part of your FishEye installation, under `FISHEYE_HOST/api/`, such as in this example:

```
http://localhost:8060/api/
```

This loads a local HTML page, where you will be able to see whether API Access is currently enabled or disabled on your FishEye instance. You will also be able to link to local code examples for REST and XML-RPC from the FishEye folders.

API mechanisms are REST-ful and XML-RPC.

Before you begin using the remote API, you will need to enable it through the FishEye Admin interface. For instructions, see Configuring the FishEye Web Server.

XML-RPC API

The XML-RPC API can be accessed from `FISHEYE_HOST/api/xmlrpc`, such as in this example:

```
http://localhost:8060/api/xmlrpc
```

REST API

The REST API can be accessed from `FISHEYE_HOST/api/rest/`, such as in this example:

```
http://localhost:8060/api/rest/changeset?rep=cvs&csid=BRANCH_2_2%3Amatt%3A20050517064053
```

which returns the details of a single changeset. Note that parameter values must be URL encoded.

REST return values are always enclosed in a `<response>` root element.

Dates are IS0-8601, in the general form

```
YYYY-MM-DDTHH:MM:SS(Z|[+-]HH:MM)
```

The timezone is optional (GMT is used if omitted). The time component is also optional. The seconds component can contain a fractional part.

For XMLRPC, FishEye returns all dates in GMT using

```
YYYYMMDDTHH:MM:SS
```

Note that no timezone is used.

Authentication

FishEye may be configured to require authentication before accessing a repository. Most methods accept an authentication token parameter. To call a method anonymously, use the empty-string for this parameter.

An authentication token can be acquired (and released) using the `login()` and `logout()` methods.

Examples

The following code example files can be found in the API folder under your FishEye instance:

```
FISHEYE_HOME\content\api\
```

Browse to that folder and you will be able to access the files below:

- Python XML-RPC example: `xmlrpc_example.py`
- Python REST example: `rest_example.py`
- Java REST example: `RestClient.java`
- The open source FishEye Plugin for JIRA provides an example of querying using the API.

Methods

> Each of the REST URLs shown below must be supplied with the same set of parameters as the XML-RPC method (although auth is optional).
> Thus the URL to use for login is `api/rest/login?username=jim&password=rover`.

Log in

```
String login(String username, String password)
```

Description
Log in and create an authentication token. Returns the token if log in was successful, or returns an error otherwise.

REST

```
    api/rest/login
```

XML-RPC

```
    String login(String username, String password)
```

Log out

```
boolean logout(String auth)
```

Description
Disables the given auth token. Returns true in all cases.

REST

```
    api/rest/logout
```

XML-RPC

```
    boolean logout(String auth)
```

FishEye Version

```
String fisheyeVersion()
```

Description
Returns the version number of this FishEye instance.

REST

```
    /api/rest/fisheyeVersion
```

XML-RPC

```
    String fisheyeVersion()
```

Example Return Values
"1.3.8", "1.4"

Since
FishEye 1.4 / Crucible 1.2

Crucible Version

```
String crucibleVersion()
```

Description
Returns the Crucible version number if Crucible is installed. This API method will return an empty String if this isn't a Crucible instance.

REST

```
    /api/rest/crucibleVersion
```

XML-RPC

```
    String crucibleVersion()
```

Example Return Values
"1.2", "1.2.1", "" (if not a Crucible instance)

Since
FishEye 1.4 / Crucible 1.2

List Repositories

```
String[] listRepositories(String auth)
```

Description
Returns a list of repository names in this FishEye instance.

REST

```
    api/rest/repositories
```

XML-RPC

```
    String[] listRepositories(String auth)
```

List Paths

```
PathInfo[] listPaths(String auth, String rep, String path)
```

Description
Returns a list of paths immediately under the given path. A path represents either a file or a directory.

REST

```
    api/rest/paths
```

XML-RPC

```
    PathInfo[] getPaths(String auth, String rep, String path)
```

Get Revision

```
Revision getRevision(String auth, String rep, String path, String rev)
```

Description
Returns the details of a particular revision.

REST

```
    api/rest/revision
```

XML-RPC

```
    Revision getRevision(String auth, String rep, String path, String rev)
```

List Tags for Revision

```
String[] listTagsForRevision(String auth, String rep, String path, String rev)
```

Description
Returns the tags associated with particular revision as an array of strings.

REST

```
    api/rest/tags
```

XML-RPC

```
    RevisionTags listTagsForRevision(String auth, String rep, String path, String rev)
```

Path History

```
PathHistory listPathHistory(String auth, String rep, String path)
```

Description
Returns history of a particular path.

REST

```
    api/rest/pathHistory
```

XML-RPC

```
    PathHistory listPathHistory(String auth, String rep, String path)
```

Get Changeset

```
Changeset getChangeset(String auth, String rep, String csid)
```

Description
Gets the details of a particular changeset.

REST

```
    api/rest/changeset
```

XML-RPC

```
    Changeset getChangeset(String auth, String csid)
```

List Changesets

```
Changesets listChangesets(String auth, String rep, String path, Date start=null, Date end=null,
Integer maxReturn=null)
```

Description
Lists changes under a given path, optionally between two dates. Returned structure contains a list of changeset ids, from most-recent to least-recent.

REST

```
    api/rest/changesets
```

XML-RPC

```
    Changesets listChangesets(String auth, String rep, String path)
    Changesets listChangesets(String auth, String rep, String path, Date start)
    Changesets listChangesets(String auth, String rep, String path, Date start, Date end)
    Changesets listChangesets(String auth, String rep, String path, Date start, Date end, Integer
maxReturn)
```

To get changes for the whole repository, use a path of "/"
If the start date is not specified, there is no lower bound.
If the end date is not specified, "now" is used.

The maxReturn clause limits the number of changesets returned by this method. If no limit is specified, FishEye will use its own internal limit (a few thousand). If this limit is exceeded, the return value will be truncated so that it contains the most-recent changesets. The value of this limit is contained in the returned data structure.

EyeQL Query

```
query(String auth, String rep, String query)
```

Description
Execute an EyeQL query. For a "normal" query, returns a list of revision keys that matched to query. If the query contains a "return" clause, then returns a custom Row for each match. The contents of the Row will depend upon the "return" clause.

REST

```
    api/rest/query
```

XML-RPC

```
    RevisionKey[] query(String auth, String rep, String query)
```

or

```
    Row[] query(String auth, String rep, String query)
```

Changeset Bounds

```
ChangesetBounds getChangesetBounds(String auth, String rep, String path=null, Date start=null, Date
end=null)
```

Description
NOT IMPLEMENTED YET. Gets the details of a particular changeset.

REST

```
    api/rest/changesetBounds
```

XML-RPC

```
    ChangesetBounds getChangesetBounds(String auth, String rep)
    ChangesetBounds getChangesetBounds(String auth, String rep, Date start)
    ChangesetBounds getChangesetBounds(String auth, String rep, Date start, Date end)
    ChangesetBounds getChangesetBounds(String auth, String rep, String path)
    ChangesetBounds getChangesetBounds(String auth, String rep, String path, Date start)
    ChangesetBounds getChangesetBounds(String auth, String rep, String path, Date start, Date end)
```

Data Types and Structures

Data types used are the same as defined in XML-RPC.

Some methods return data structures. These map into XML-RPC as expected.

For REST calls, structs are encoded as XML elements of the same name (but all lowercase). Members are encoded as sub-elements, or as attributes as indicated below.

RevisionKey

```
    struct RevisionKey {
      String path; // (REST: attribute)
String rev; // (REST: attribute)
}
```

PathInfo

```
    struct PathInfo {
      String name;    // (REST: attribute)
boolean isFile; // (REST: attribute)
boolean isDir; // (REST: attribute)
boolean isHeadDeleted; // (REST: attribute)
}
```

Revision

```
    struct Revision {
      String path; // (REST: attribute)
String rev; // (REST: attribute)
String author; // (REST: attribute)
Date date; // (REST: attribute)
String state; // one of "changed" "added" or "deleted" (REST: attribute)
int totalLines; // (REST: attribute)
int linesAdded; // (REST: attribute)
int linesRemoved; // (REST: attribute)
String log;
      String csid; // optional (REST: attribute)
String ancestor; // optional (REST: attribute)
}
```

Changeset

```
    struct Changeset {
      String csid; // (REST: attribute)
Date date; // (REST: attribute)
String author; // (REST: attribute)
String branch; // (REST: attribute)
boolean sealed; // (REST: attribute)
String log;
      RevisionKey[] revisions;
    }
```

Changesets

```
    struct Changesets {
      int maxReturn; // (REST: attribute)
String[] csids;
    }
```

Description

A list of Changeset ids, most-recent changeset first. `maxReturn` indicates the maximum number of changesets FishEye is configured to return from this method.

ChangesetBounds

```
    struct ChangesetBounds {
      Changeset first;
      Changeset last;
    }
```

Row

```
    struct Row {
      ...
    }
```

Description

A custom structure, depending on the given EyeQL statement. Each member of Row is typed.

Writing SOAP Clients for Fisheye or Crucible RPC Plugins

This is an example of a SOAP client calling a Fisheye/Crucible SOAP RPC plugin It uses the JAX-WS dispatch approach to calling SOAP, rather than generating stubs from the WSDL.

```
URL wsdlURL = new URL("http://localhost:6060/foo/service/review?wsdl");
String namespace = "http://rpc.spi.crucible.atlassian.com/";
Service service = Service.create(wsdlURL, new QName(namespace, "Review"));
Dispatch<SOAPMessage> disp = service.createDispatch(new QName(namespace, "ReviewPort"),
SOAPMessage.class, Service.Mode.MESSAGE);
MessageFactory mf = MessageFactory.newInstance();
SOAPMessage call = mf.createMessage();
SOAPBody body = call.getSOAPBody();
QName bodyName = new QName(namespace, "getAllReviews", "m");
SOAPBodyElement bodyElement = body.addBodyElement(bodyName);
QName name = new QName("token");
SOAPElement symbol = bodyElement.addChildElement(name);
symbol.addTextNode("blank"); // we are not providing a valid token as we are using Trusted Application
Authentication

/** this section is for Trusted Application Authentication **/
EncryptedCertificate cert = currentApplication.encode("matt");
Map<String, List> headers = new HashMap<String,List>();
headers.put(CurrentApplication.HEADER_TRUSTED_APP_ID, Collections.singletonList(cert.getID()));
headers.put(CurrentApplication.HEADER_TRUSTED_APP_CERT,
Collections.singletonList(cert.getCertificate()));
headers.put(CurrentApplication.HEADER_TRUSTED_APP_SECRET_KEY,
Collections.singletonList(cert.getSecretKey()));
disp.getRequestContext().put(MessageContext.HTTP_REQUEST_HEADERS, headers);
/** end Trusted Application Authentication setup **/

SOAPMessage response = disp.invoke(call);
response.writeTo(System.out);
```

Note that the example above is using Trusted Application Authentication. If you were using username/password authentication you would first call the login method on the `http://localhost:6060/foo/service/auth` endpoint, and pass the token it returned instead of the string "blank".

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 revision?
  - 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?
- Installation & Configuration FAQ
  - Are anonymous users counted towards FishEye's licence limits?
  - Can FishEye be run as a Windows service?
  - How Do I Investigate the FishEye Database Schema?
  - Improve FishEye Scan Performance
    - Setting Up a CVS Mirror with rsync
  - Migrating FishEye Between Servers
- 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?
- Troubleshooting
  - After I commit a change to my CVS repository, it takes a long time before it appears in FishEye.
  - 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
  - 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.
  - URLs with encoded slashes don't work, especially in Author constraints

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

CVS FAQ

FishEye CVS FAQ

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

How does FishEye calculate CVS changesets?

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

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

Example EyeQL Queries

EyeQL

- How do find changes made to a branch after a given revision?
- 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?

For more information on using EyeQL, see the Reference guide.

How do find changes made to a branch after a given revision?

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
```

Installation & Configuration FAQ

FishEye Installation & Configuration FAQ

- Are anonymous users counted towards FishEye's licence limits?
- Can FishEye be run as a Windows service? — 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.
- How Do I Investigate the FishEye Database Schema?
- Improve FishEye Scan Performance — You can increase the speed of your scans using the following options:
    - Setting Up a CVS Mirror with rsync
- Migrating FishEye Between Servers

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.

Can FishEye be run as a Windows service?

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).

How Do I Investigate the FishEye Database Schema?

The purpose of this guide is to walk you through connecting to Crucible's embedded Hypersonic SQL Database using the Database Administration tool DBVisualizer. This will allow you to investigate the FishEye database schema.

On this page:

Read on for step by step instructions on how to Configure DbVisualizer and connect it to FishEye's HSQLDB database.

Prerequisites

1. Download and install the latest copy of DBVisualizer.
2. Create a backup of your database, which you will connect to.
3. Ensure that Crucible is not running or use DBVisualizer to connect to the backup of your database and not the actual production version.

Connection Procedure

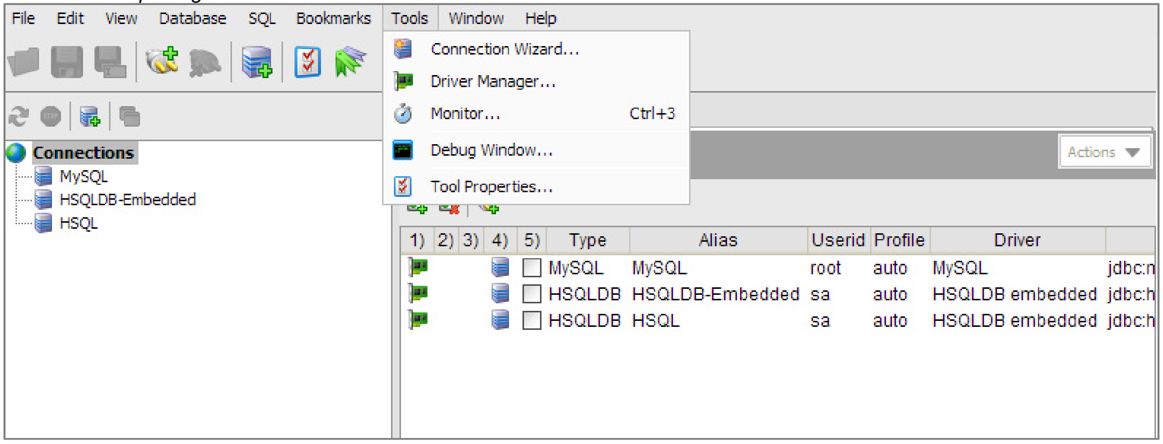
Please ensure that you read and follow the instructions below carefully.

> ⚠ **Ensure you create a backup (Administration > Backup) of your data before attempting any modifications.**

Create the Database Connection

1. Open Connection Wizard.

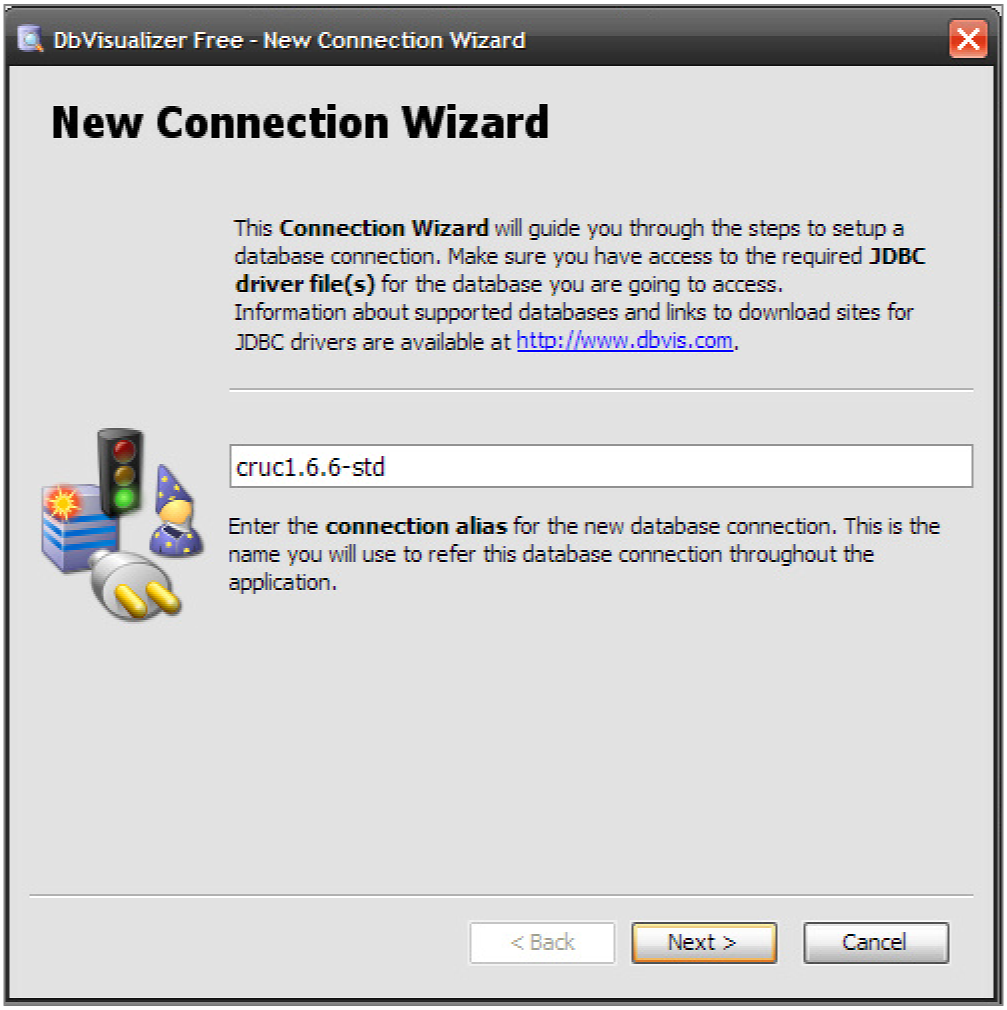
   Screenshot: Opening the Connection Wizard



1)	2)	3)	4)	5)	Type	Alias	Userid	Profile	Driver	
				☐	MySQL	MySQL	root	auto	MySQL	jdbc:n
				☐	HSQLDB	HSQLDB-Embedded	sa	auto	HSQLDB embedded	jdbc:h
				☐	HSQLDB	HSQL	sa	auto	HSQLDB embedded	jdbc:h

2. Enter an identifiable name for the connection. For example: `cruc1.6.6-std`

   Screenshot: Entering a Name for the Connection



New Connection Wizard

This **Connection Wizard** will guide you through the steps to setup a database connection. Make sure you have access to the required **JDBC driver file(s)** for the database you are going to access.
Information about supported databases and links to download sites for JDBC drivers are available at http://www.dbvis.com.

cruc1.6.6-std

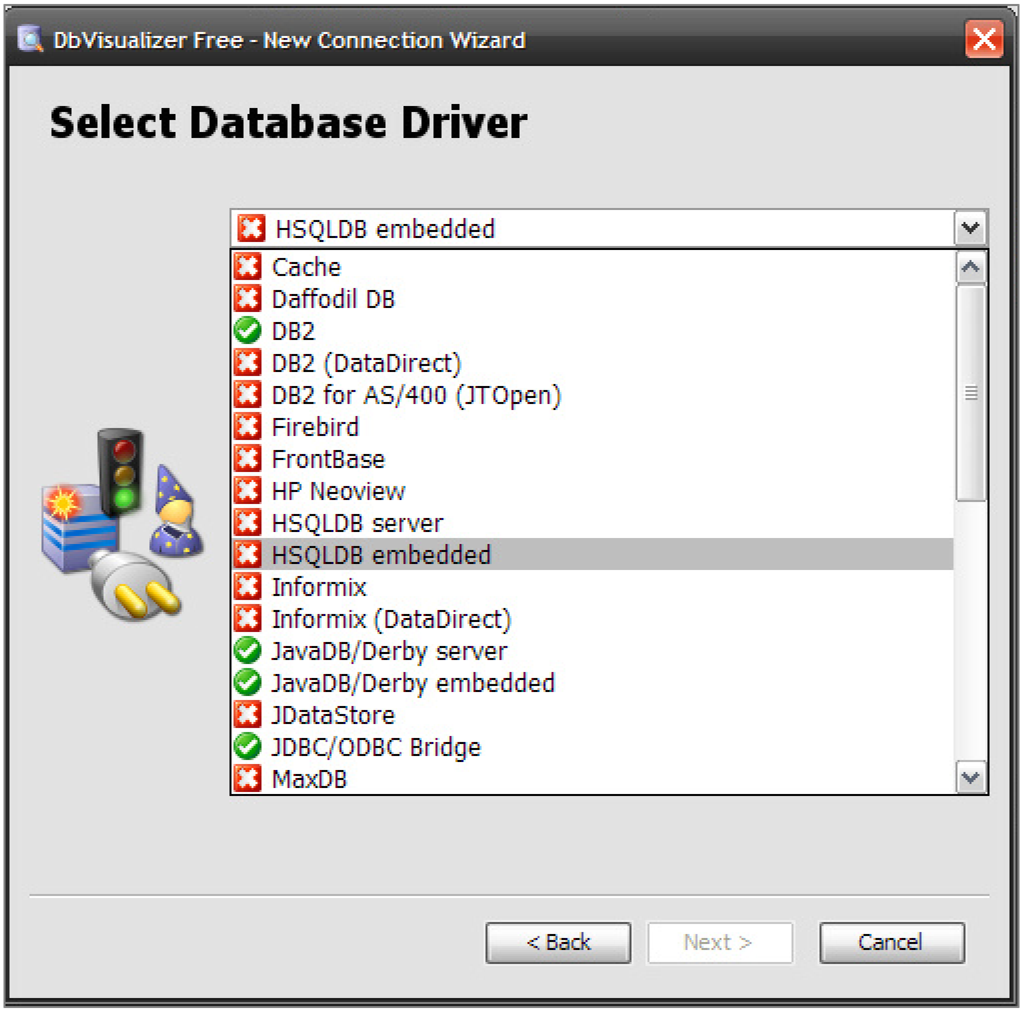
Enter the **connection alias** for the new database connection. This is the name you will use to refer this database connection throughout the application.

3. Click '**Next**'.

Select a JDBC Driver

1. From the drop down list select **HSQLDB Embedded**

   Screenshot: Selecting a Database



2. Click on **Load Driver Files**
3. Browse to FISHEYE_HOME/lib directory where the `hsqldb*.jar` file is located, eg `/Users/pkamal/crucible-1.6.6/lib/hsqldb-1.8.0.10.jar`

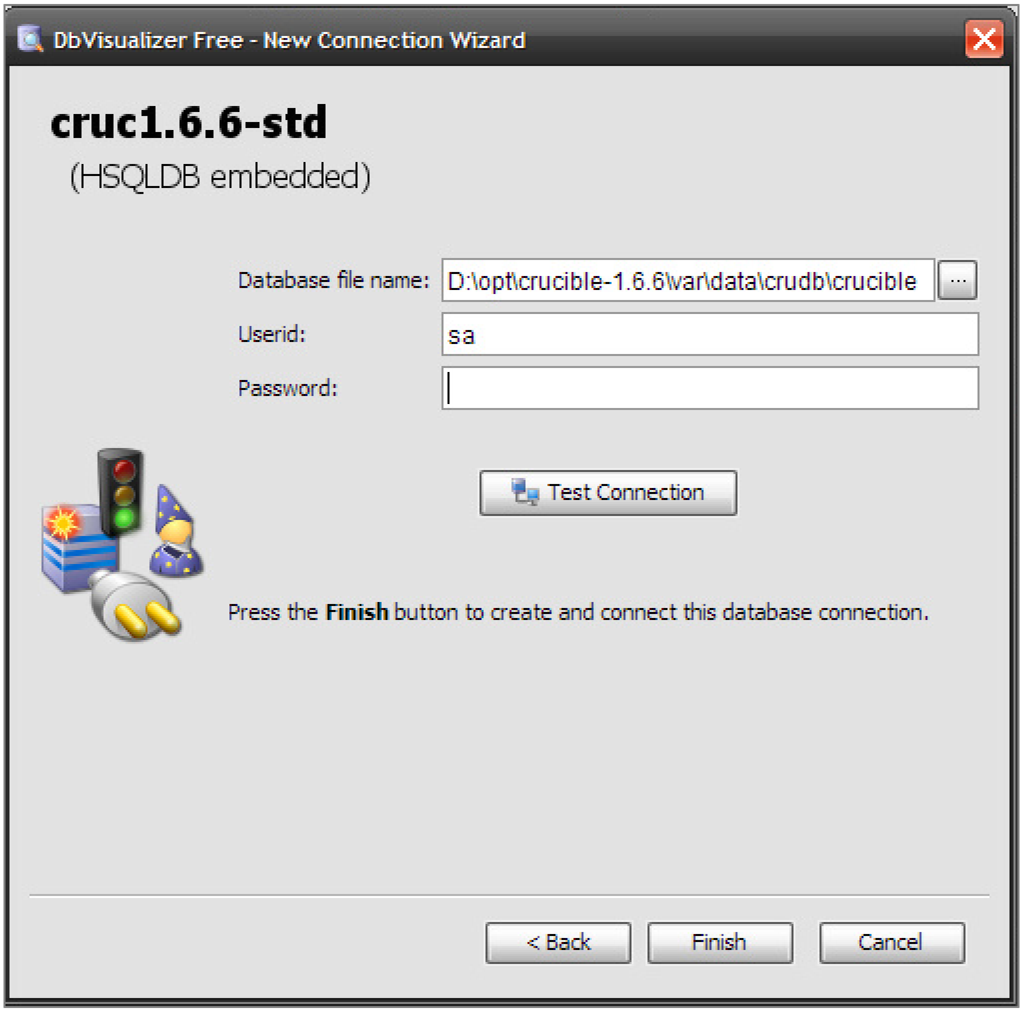
Select the Database Path

1. Browse to your `<Crucible-Home>` directory if you are going to connect to your production database (crucible is shutdown), or Browse to your expanded <<Backup>> directory that was created when you unzipped the backup file you made (if you just want to run some queries against your data).
2. Open the **/var/data/crudb** folder.
3. Select the `crucible.properties` file

Enter the Connection Details

1. Remove the **.properties** from the end of `crucible`

   Screenshot: Entering Connection Details



DbVisualizer Free - New Connection Wizard

cruc1.6.6-std

(HSQLDB embedded)

Database file name: D:\opt\crucible-1.6.6\var\data\crudb\crucible

Userid: sa

Password:

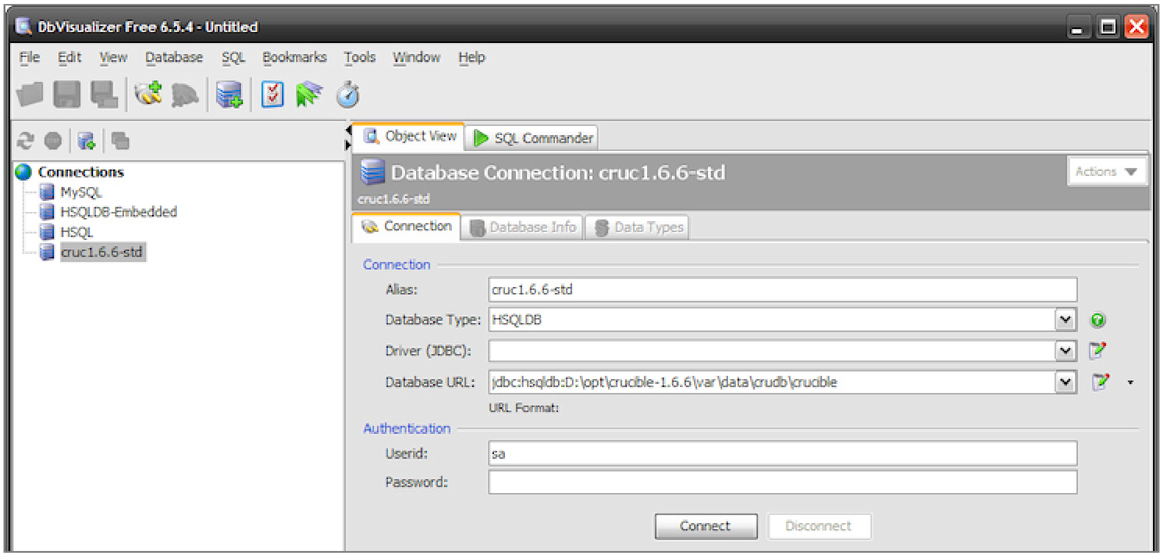
Press the **Finish** button to create and connect this database connection.

2. Type in <u>sa</u> for the username.
3. Leave the password field blank.
4. Click on Test Connection to verify that the details are correct.
5. Click on '**Finish**' to complete the setup.
   Refer to the example screenshot above if you are unsure

Connect to the Embedded Database

1. Select the connection from the list on the left hand side.

   Screenshot: Connecting to the Embedded Database



Connection

Alias: cruc1.6.6-std

Database Type: HSQLDB

Driver (JDBC):

Database URL: jdbc:hsqldb:D:\opt\crucible-1.6.6\var\data\crudb\crucible

URL Format:

Authentication

Userid: sa

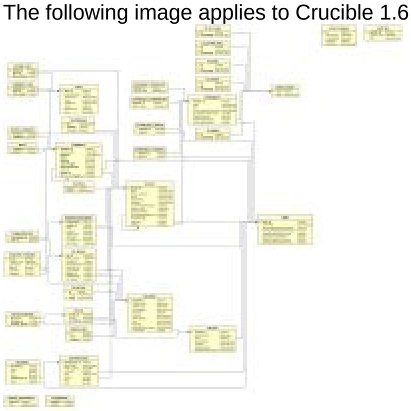
Password:

2. You can now click on '**Connect**' to connect to the embedded database.

Viewing the Tables and their References

1. Click on tables on the left and references in the middle to see an image that displays all tables and shows references.

2. The following image applies to Crucible 1.6.6 tables. Please note that the tables will be different in different versions of the product.



Improve FishEye Scan Performance

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.

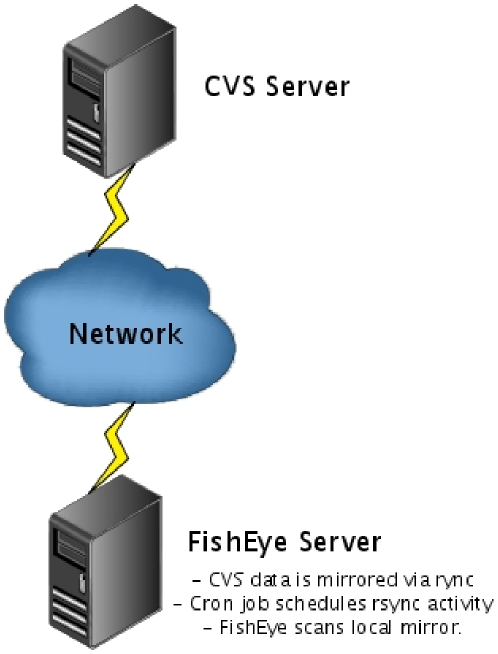
Setting Up a CVS Mirror with rsync

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.

ℹ 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 rync
- 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.

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.

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
- 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.

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.

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.
- 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? — 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
- 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.
- 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.

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

Out Of Memory Errors

There are a number of different memory errors that the JVM will throw. The most common are listed as follows.

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=128m"
```

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=128m` to FISHEYE_OPTS, in addition to any argument you use to set the heap size.

```
FISHEYE_OPTS="-Xms128m -Xmx512m -XX:MaxPermSize=128m"
```

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:

```
-Xss512k
```

Please refer to this guide as a reference for JVM tuning.

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.

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.

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
```

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 Installation & Upgrade Guide

- FishEye Installation Guide
  - System Requirements
    - Setting JAVA_HOME
    - Setting JVM System Properties
  - Installing FishEye
  - Configuring FishEye
  - Best Practices for FishEye Configuration
- FishEye Upgrade Guide

FishEye Installation Guide

This guide describes the advanced installation options that can be used when installing Fisheye. For a quick install see the Quick Start Guide.

✓ **Knowledge Base**

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

- System Requirements
- Installing FishEye
- Configuring FishEye
- Best Practices for FishEye Configuration

System Requirements

On this page:

- Java Environment and Operating System
- Platform Hardware Requirements
- Version Control System
- Web Browser
- Deployment
- Single Sign On with Atlassian Crowd

Java Environment and Operating System

Java Runtime	A JDK or JRE version **1.5** or greater. (Solaris requires **1.5.0_15** as a minimum) ⚠ There is a known issue with FishEye 1.6.x and the JDK/JRE versions 1.6.0 through 1.6.0u3(update 3). FishEye requires version 2.1 of a component called JAXB, but these versions of the JDK/JRE include JAXB2.0. To avoid this issue, upgrade your JDK/JRE to version 1.6.u4(update 4) or later.

⚠ 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.

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.

Note: There appeared to be a problem with some releases of the JRockit JVM that causes corrupted caches in FishEye. If you use JRockit, we recommend you use the latest JRockit 6 JVM. This problem has been confirmed on

- JRockit 5.0 JVM (R25.0.0-75)
- JRockit 5.0 JVM (R27.3.0) |

Operating System	FishEye is a pure Java application and should run on any platform provided the above requirements are satisfied.

Platform 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.
Disk space	In the worst case scenario, your budget for free disk space should meet or exceed three times the size of your repository data. For example, with 80GB of repository data, you would ideally have 3 x 80GB, hence 240GB of free disk space dedicated to FishEye. However this will not usually be necessary, especially if you have a number of sizeable binary files inside your repository (which increase its size but have a relatively small impact on the index). ⚠ Fisheye 1.6.* requires far more disk space. Please refer to FE-1094

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

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

Version Control System

At this time, FishEye supports the following source code management (SCM) systems:

- Subversion
- Perforce
- CVS (and CVS-NT)

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 (such as ClearCase) is planned for future releases. Let us know what SCM system you would like to see supported by logging and/or voting for a JIRA issue.

Web Browser

Fisheye has been tested with Firefox 3, Internet Explorer 7, and Safari 4.
IE6 is NOT supported. Fisheye should work with most modern browsers.

> ✓ **Font size**
> (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.

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 1.3 client library, and therefore is intended to operate with Crowd 1.3 or later.

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.

ℹ 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.

ℹ 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).

ℹ Users integrating FishEye with Single Sign On applications may require an adjustment to the JVM properties, specifically the HTTP header size. This may also be useful in other circumstances where FishEye's default HTTP header is too small (at 4096 bytes).

Installing FishEye

This guide describes the advanced FishEye installation options. For a quick install, see the Quick Start Guide.

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 System Requirements.
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 System Requirements, 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/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 temporary data is found under `$FISHEYE_INST/var/`
`$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.

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.

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

Want a hands-on tour of the best FishEye features? See the FishEye Evaluator Resources 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.

See Also

- Improve FishEye Scan Performance
- Tuning FishEye

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

- 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.

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**.

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 Release Notes

⚠ FishEye 2.0 has now been released. Read the Release Notes.

FishEye Release Notes and Changelogs

- FishEye Release Summary
- FishEye 2.0 Release 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 Release Summary

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.

FishEye 2.0 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.

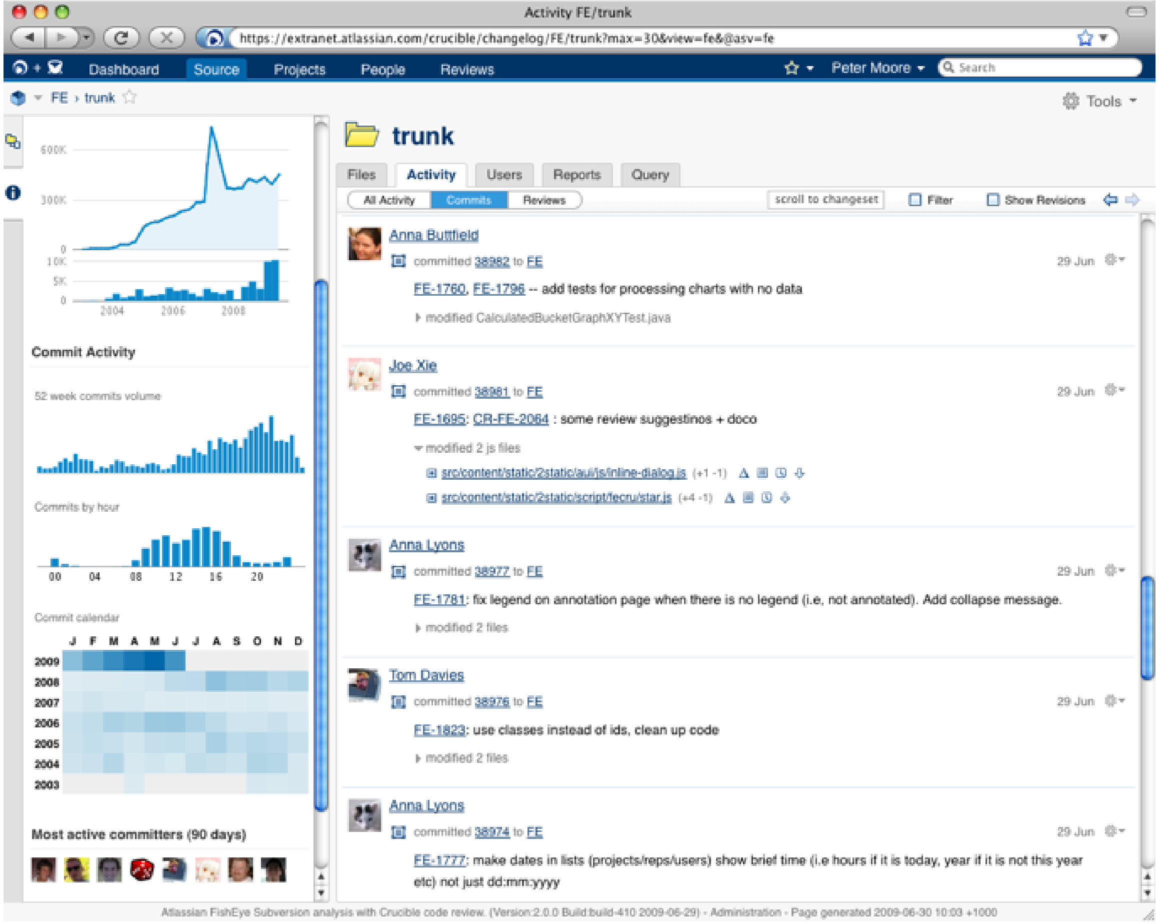
Highlights of FishEye 2.0

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



trunk

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

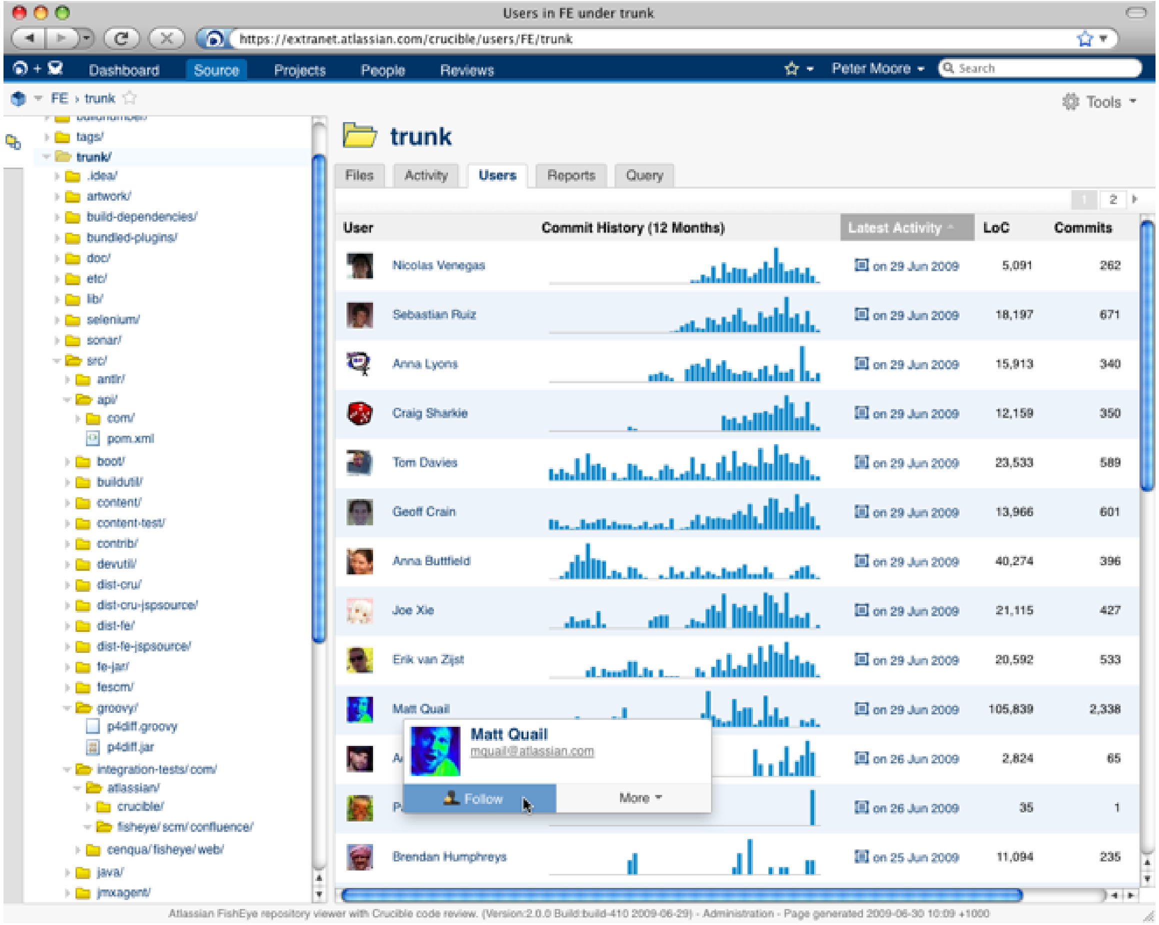
Atlassian FishEye Subversion analysis with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 10:03 +1000

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

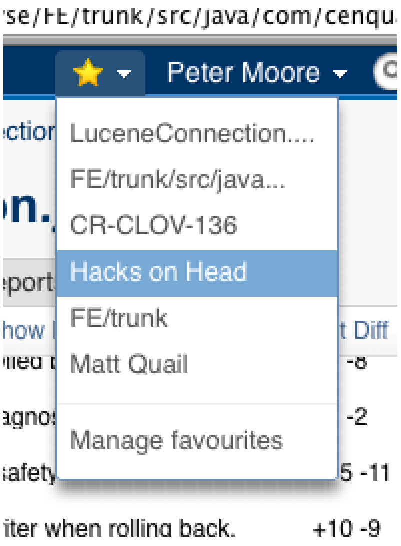
Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 10:09 +1000

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

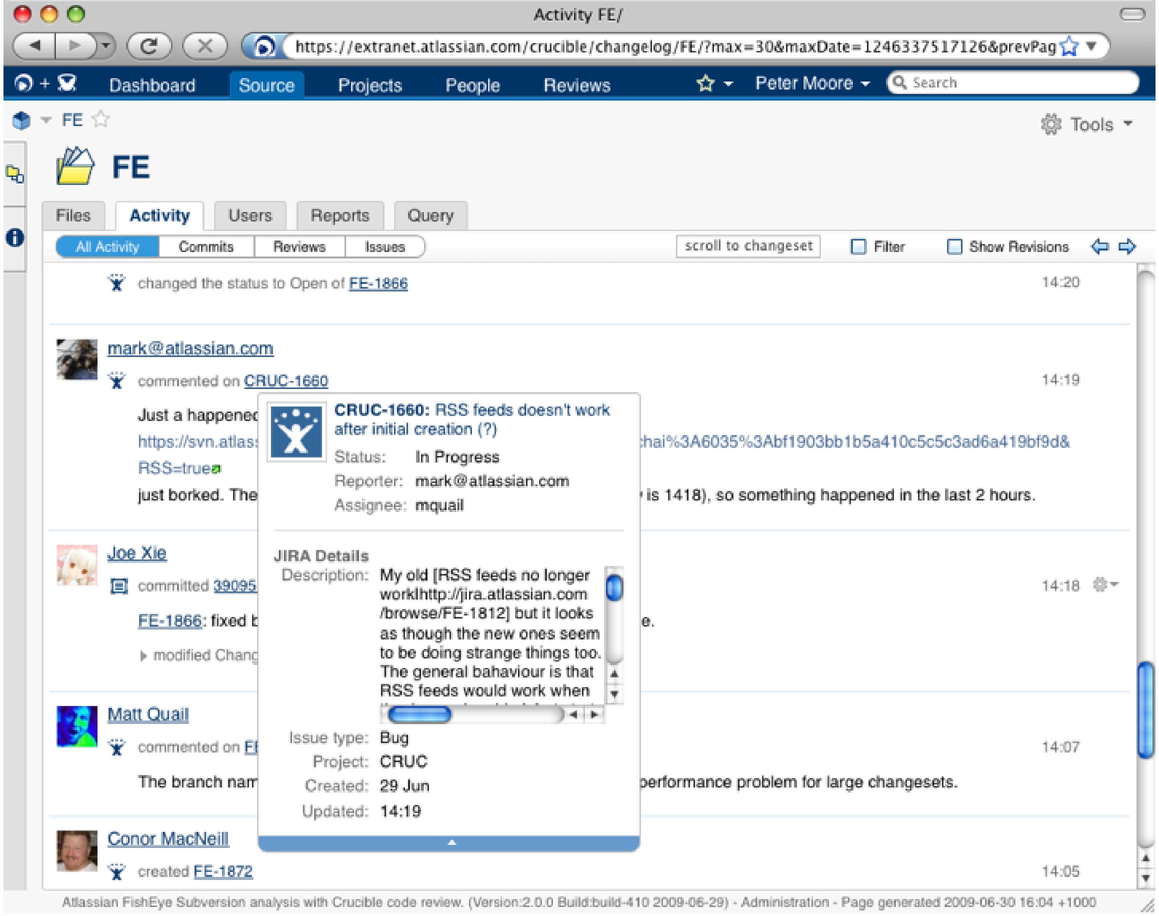


4

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

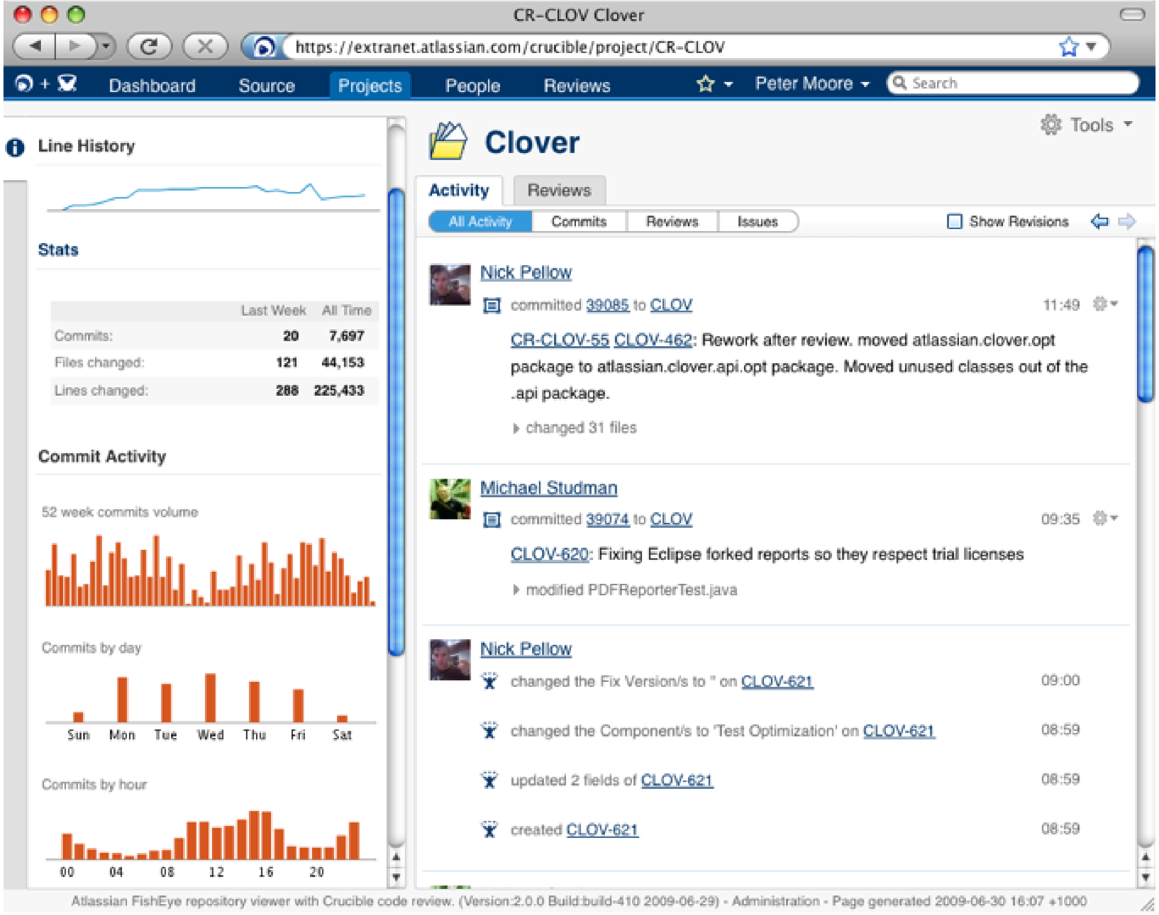
Atlassian FishEye Subversion analysis with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 16:04 +1000

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

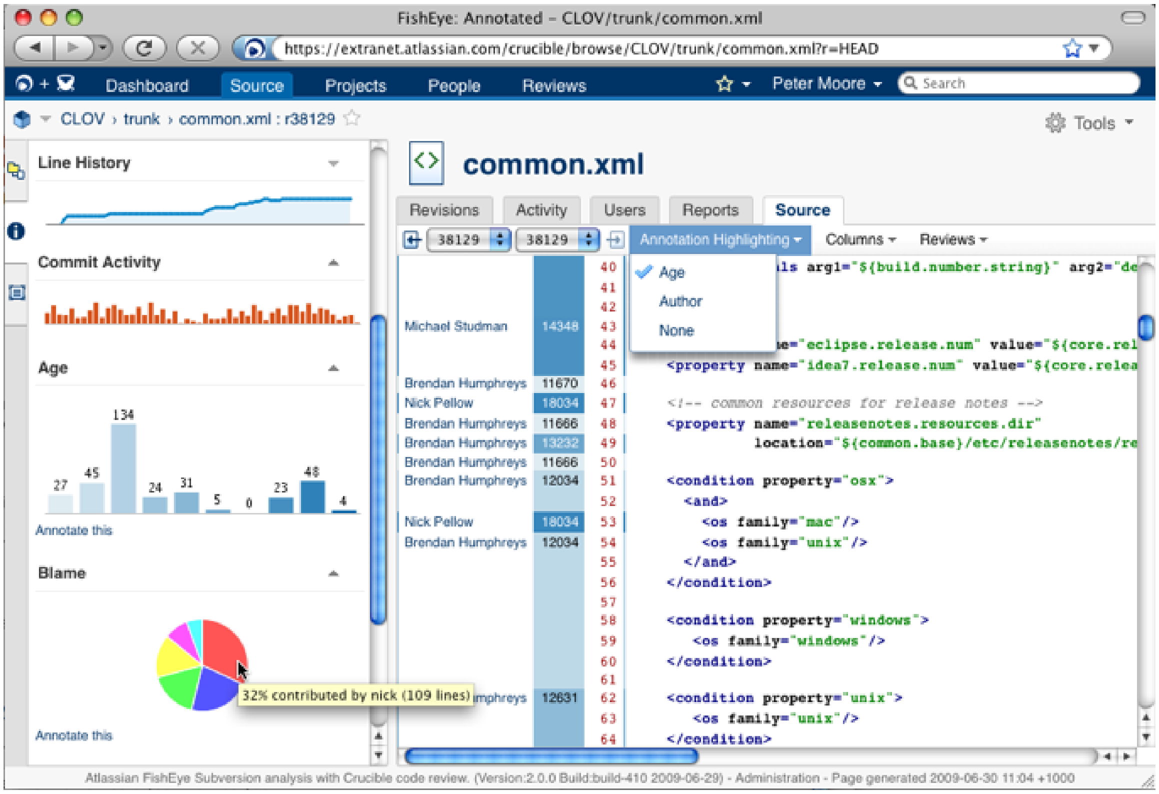
Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 16:07 +1000

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



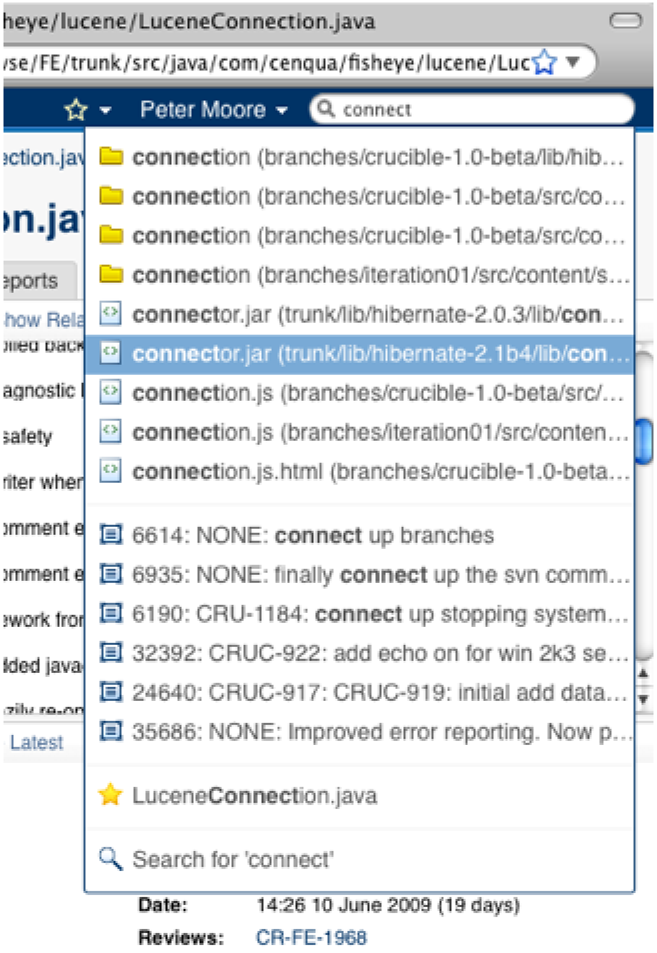
```
                 ls arg1="${build.number.string}" arg2="de



                 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>
```

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…

- LuceneConnection.java

- Search for 'connect'

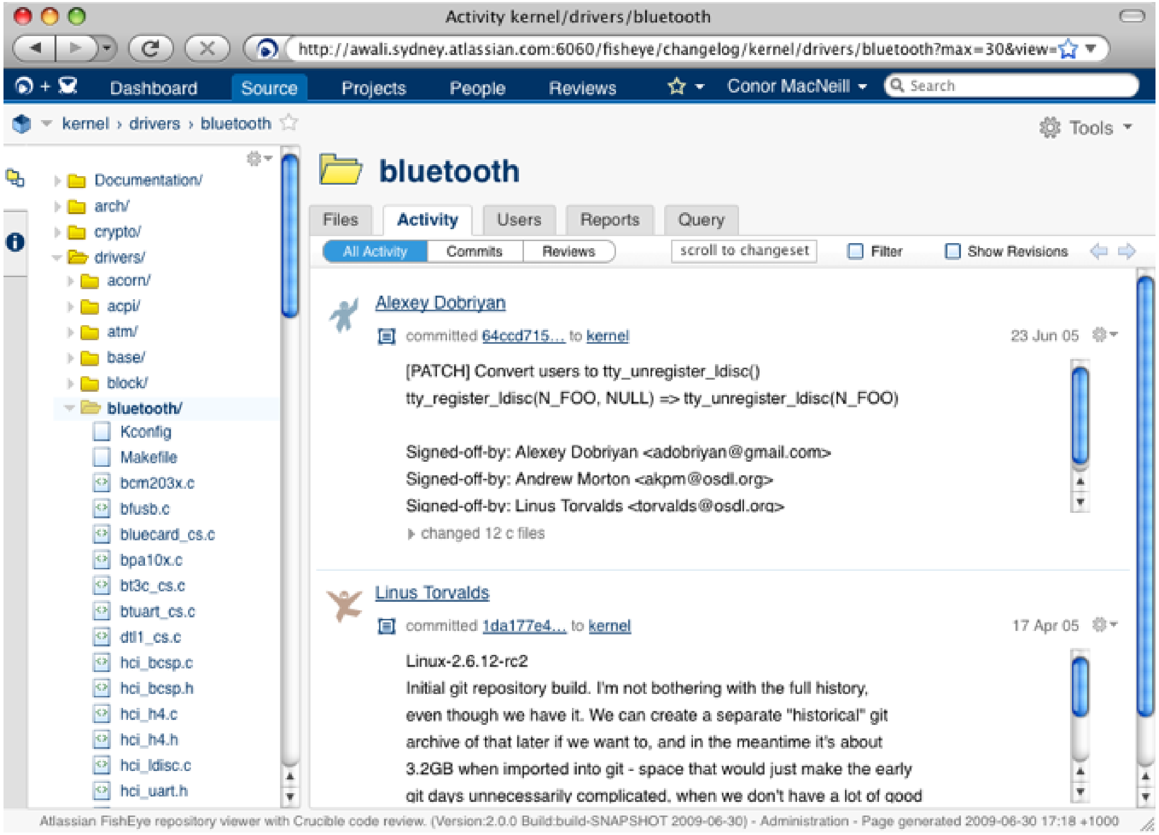
Date: 14:26 10 June 2009 (19 days)
Reviews: CR-FE-1968

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 Beta3 to 2.0
- From 2.0 Beta2 to 2.0 Beta3
- From 1.6.6 to 2.0 Beta
-

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-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-1311	use a better UI than ^ and V, see craig/pete		Closed
FE-838	FE: No hyperlink for copies expander		Closed
FE-1312	change the comparator BY_RECENT_ACTIVITY to sort nulls last -- remove have NoActivityItem	❗	Closed
FE-1038	svn-connection getting wrong values out of the repository configuration	⇑	Closed
FE-526	SVN scan failes when repository contains files whose names contain non-standard characters (escaped) and symbolic links are versioned	⇑	Closed
FE-459	Directory widget and search are now much slower to load	⇑	Closed
FE-1865	removing a favourite and then clicking on it again causes an error	↑	Closed
FE-1854	Activity stream on project page doesn't filter by mapped JIRA projects	↑	Closed
FE-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-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-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-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-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-1310	Rework from CR-FE-1441: FE-1236: sortable and paged list of users at /users and committers	⇧	Resolved
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-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-1285	Integrate new ui for file history page	⇧	Resolved
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-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-1138	Review design of CommitterUserMapping.hmb.xml	⇧	Closed
FE-1136	Add the ability to return unique results in EyeQL	⇧	Closed
FE-962	allow execution contexts to be added to quartz jobs / triggers	⇧	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-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-758	Upgrade to Atlassian-Plugins 2.1	⇧	Closed
FE-682	the "commit history" histogram should optionally show changeset counts, not just revision counts	⇧	Closed
FE-654	Setting show-changelog-calendar to true will cause the changelog to take forever to load.	⇧	Closed
FE-553	Refactor file history page.	⇧	Closed
FE-506	Support mail servers using SSL or TLS	⇧	Closed
FE-466	extend cru eyeql so you can select "included in review"	⇧	Closed
FE-461	Abstraction layer for projects - differentiate projects from repositories	⇧	Closed
FE-351	ability map mulitple authors per repository	⇧	Closed
FE-337	Add support for GIT scm	⇧	Closed
FE-327	diff screen needs quick navigation links to prev/next diff	⇧	Closed
FE-134	Make link to "raw" mode more prominent in annotated view	⇧	Closed
FE-110	Timeout occuring during date based constraint	⇧	Closed
FE-89	Support Start Revision for Perforce	⇧	Closed
FE-80	Dashboard: Home page improvements	⇧	Closed
FE-50	add to 'Search' page in new FishEye Feature Tour, after implemented in Crucible	⇧	Closed
FE-46	Ability to request a repository update on demand	⇧	Closed

FE-41	Switch to Atlassian Icons	↑	Closed
FE-1383	RSS improvements	↓	Resolved

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-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-1311	use a better UI than ^ and V, see craig/pete		Closed
FE-838	FE: No hyperlink for copies expander		Closed
FE-1312	change the comparator BY_RECENT_ACTIVITY to sort nulls last -- remove have NoActivityItem	❗	Closed
FE-1038	svn-connection getting wrong values out of the repository configuration	⇑	Closed
FE-526	SVN scan failes when repository contains files whose names contain non-standard characters (escaped) and symbolic links are versioned	⇑	Closed
FE-459	Directory widget and search are now much slower to load	⇑	Closed
FE-1865	removing a favourite and then clicking on it again causes an error	↑	Closed
FE-1854	Activity stream on project page doesn't filter by mapped JIRA projects	↑	Closed
FE-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-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-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-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-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-1310	Rework from CR-FE-1441: FE-1236: sortable and paged list of users at /users and committers	⇧	Resolved
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-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-1285	Integrate new ui for file history page	⇧	Resolved
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-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-1138	Review design of CommitterUserMapping.hmb.xml	⇧	Closed
FE-1136	Add the ability to return unique results in EyeQL	⇧	Closed
FE-962	allow execution contexts to be added to quartz jobs / triggers	⇧	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-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-758	Upgrade to Atlassian-Plugins 2.1	⇧	Closed
FE-682	the "commit history" histogram should optionally show changeset counts, not just revision counts	⇧	Closed
FE-654	Setting show-changelog-calendar to true will cause the changelog to take forever to load.	⇧	Closed
FE-553	Refactor file history page.	⇧	Closed
FE-506	Support mail servers using SSL or TLS	⇧	Closed
FE-466	extend cru eyeql so you can select "included in review"	⇧	Closed
FE-461	Abstraction layer for projects - differentiate projects from repositories	⇧	Closed
FE-351	ability map mulitple authors per repository	⇧	Closed

FE-337	Add support for GIT scm	⬆	Closed
FE-327	diff screen needs quick navigation links to prev/next diff	⬆	Closed
FE-134	Make link to "raw" mode more prominent in annotated view	⬆	Closed
FE-110	Timeout occuring during date based constraint	⬆	Closed
FE-89	Support Start Revision for Perforce	⬆	Closed
FE-80	Dashboard: Home page improvements	⬆	Closed
FE-50	add to 'Search' page in new FishEye Feature Tour, after implemented in Crucible	⬆	Closed
FE-46	Ability to request a repository update on demand	⬆	Closed
FE-41	Switch to Atlassian Icons	⬆	Closed
FE-1383	RSS improvements	⬇	Resolved

From 1.6.6 to 2.0 Beta

Full list of issues in this release:

JIRA Issues (200 issues)

Key	Summary	Priority	Status
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-1311	use a better UI than ^ and V, see craig/pete		Closed
FE-838	FE: No hyperlink for copies expander		Closed
FE-1312	change the comparator BY_RECENT_ACTIVITY to sort nulls last -- remove have NoActivityItem	❗	Closed
FE-1038	svn-connection getting wrong values out of the repository configuration	⇪	Closed
FE-526	SVN scan failes when repository contains files whose names contain non-standard characters (escaped) and symbolic links are versioned	⇪	Closed
FE-459	Directory widget and search are now much slower to load	⇪	Closed
FE-1865	removing a favourite and then clicking on it again causes an error	⬆	Closed
FE-1854	Activity stream on project page doesn't filter by mapped JIRA projects	⬆	Closed

FE-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-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-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-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-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-1310	Rework from CR-FE-1441: FE-1236: sortable and paged list of users at /users and committers	⇧	Resolved
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-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-1285	Integrate new ui for file history page	⇧	Resolved
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-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-1138	Review design of CommitterUserMapping.hmb.xml	⇧	Closed
FE-1136	Add the ability to return unique results in EyeQL	⇧	Closed
FE-962	allow execution contexts to be added to quartz jobs / triggers	⇧	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-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-758	Upgrade to Atlassian-Plugins 2.1	⇧	Closed

FE-682	the "commit history" histogram should optionally show changeset counts, not just revision counts	⇧	Closed
FE-654	Setting show-changelog-calendar to true will cause the changelog to take forever to load.	⇧	Closed
FE-553	Refactor file history page.	⇧	Closed
FE-506	Support mail servers using SSL or TLS	⇧	Closed
FE-466	extend cru eyeql so you can select "included in review"	⇧	Closed
FE-461	Abstraction layer for projects - differentiate projects from repositories	⇧	Closed
FE-351	ability map mulitple authors per repository	⇧	Closed
FE-337	Add support for GIT scm	⇧	Closed
FE-327	diff screen needs quick navigation links to prev/next diff	⇧	Closed
FE-134	Make link to "raw" mode more prominent in annotated view	⇧	Closed
FE-110	Timeout occuring during date based constraint	⇧	Closed
FE-89	Support Start Revision for Perforce	⇧	Closed
FE-80	Dashboard: Home page improvements	⇧	Closed
FE-50	add to 'Search' page in new FishEye Feature Tour, after implemented in Crucible	⇧	Closed
FE-46	Ability to request a repository update on demand	⇧	Closed
FE-41	Switch to Atlassian Icons	⇧	Closed
FE-1383	RSS improvements	⇩	Resolved

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.

ℹ 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.

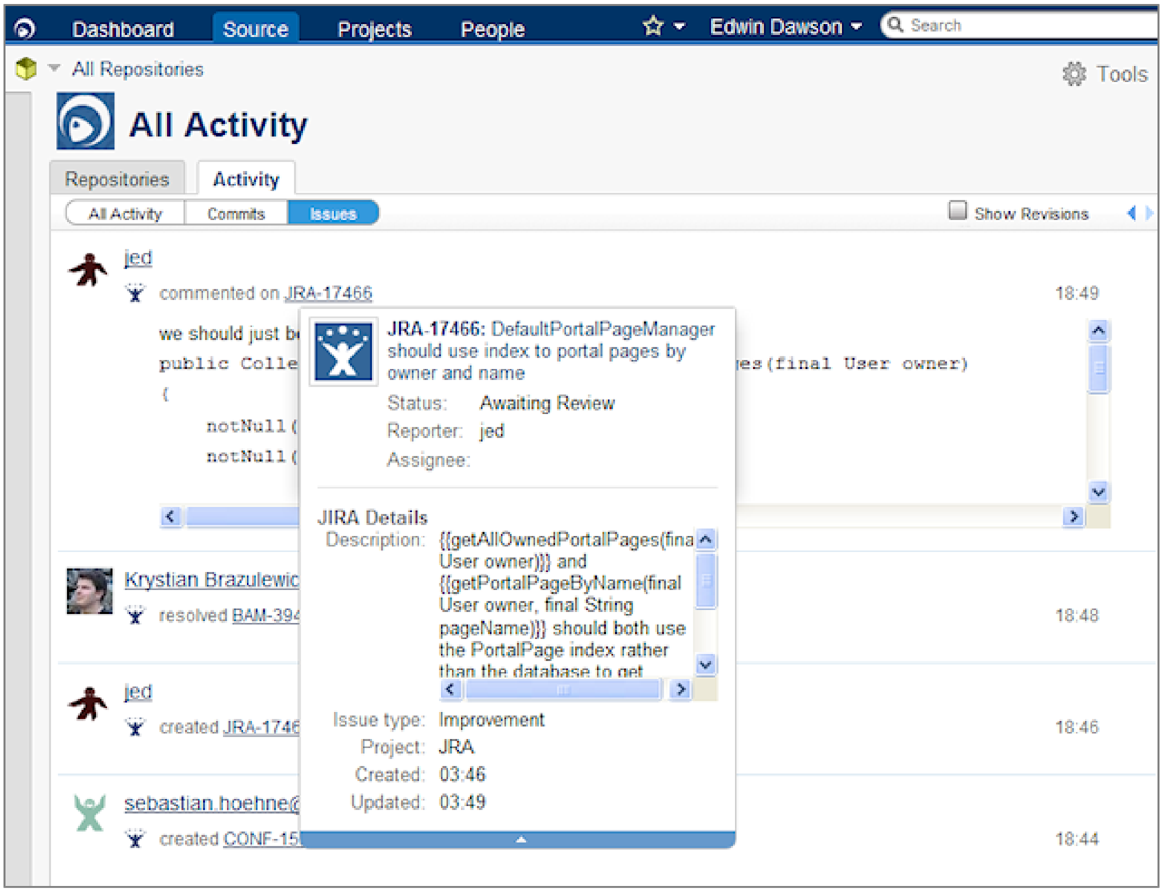
Highlights of FishEye 2.0 Beta

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(
```

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

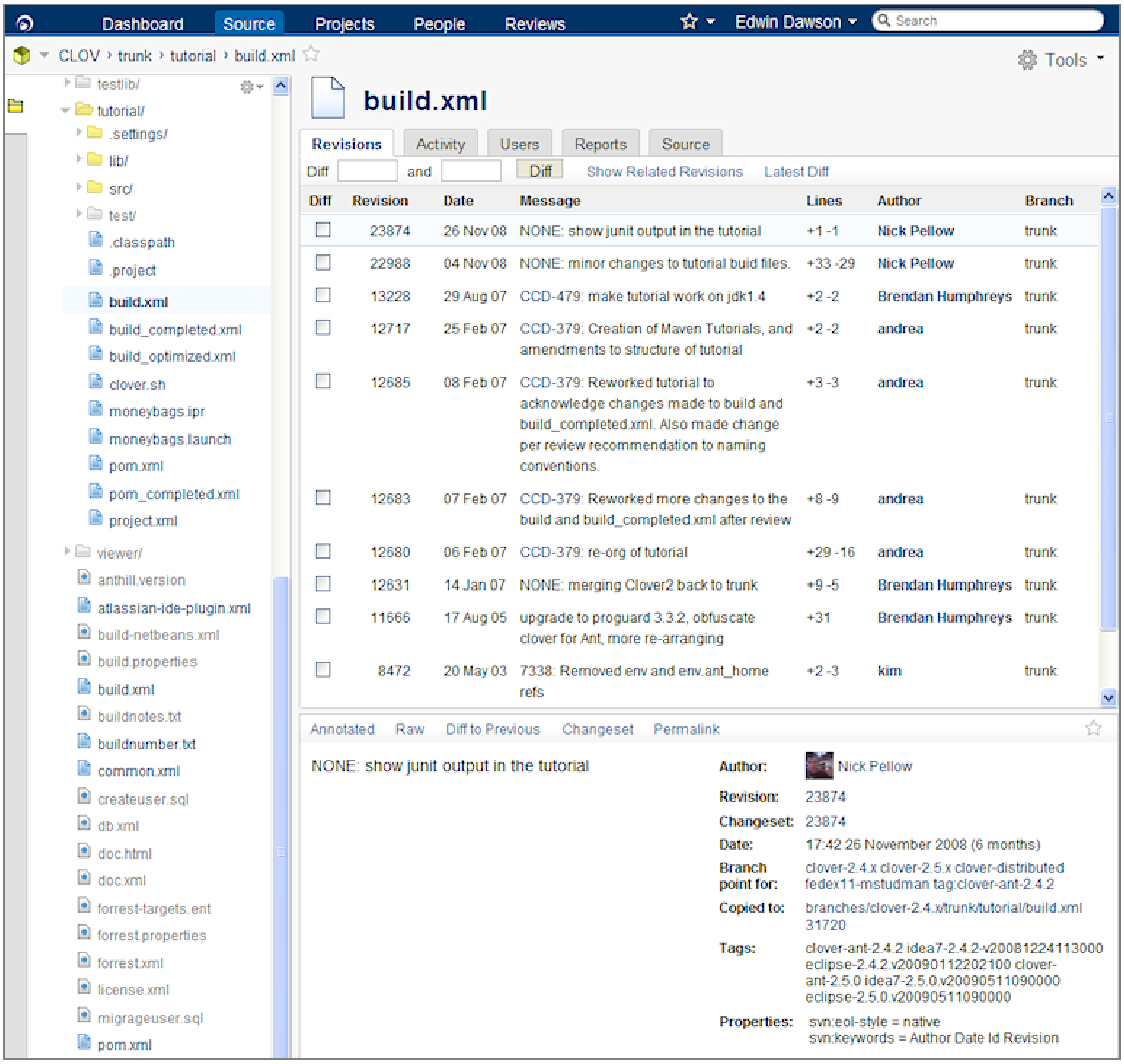
sebastian.hoehne@
created CONF-15 18:44

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



build.xml

Diff	Revision	Date	Message	Lines	Author	Branch
□	23874	26 Nov 08	NONE: show junit output in the tutorial	+1 -1	Nick Pellow	trunk
□	22988	04 Nov 08	NONE: minor changes to tutorial 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

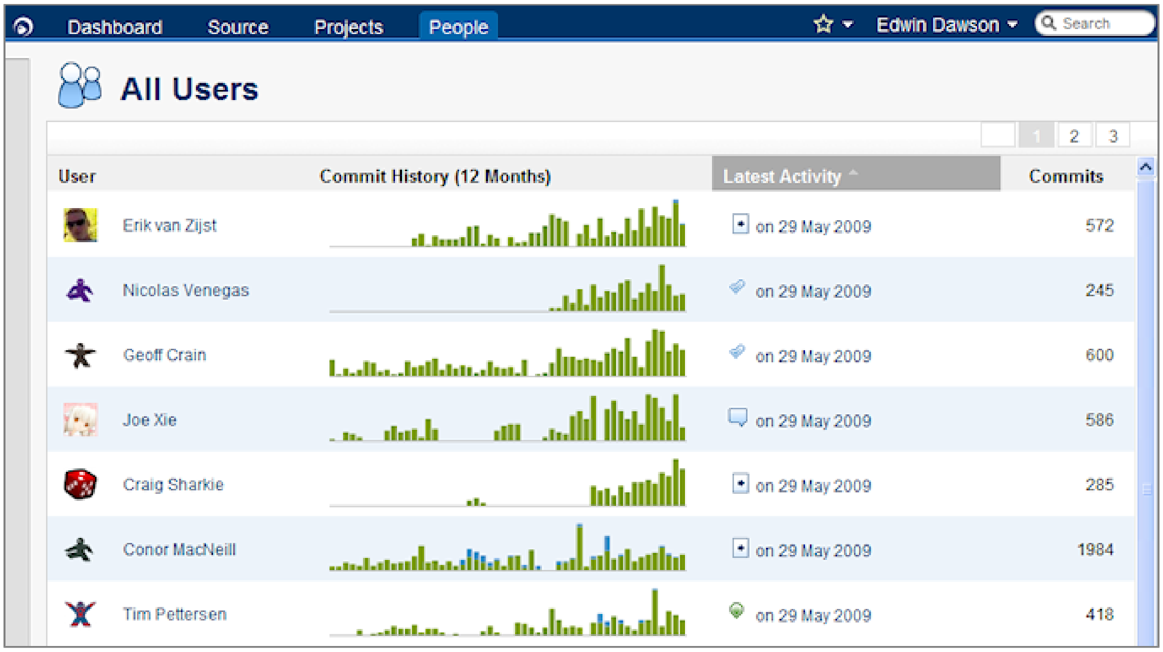
Author:	Nick Pellow
Revision:	23874
Changeset:	23874
Date:	17:42 26 November 2008 (6 months)
Branch point for:	clover-2.4.x clover-2.5.x clover-distributed fedex11-mstudman tag:clover-ant-2.4.2
Copied to:	branches/clover-2.4.x/trunk/tutorial/build.xml 31720
Tags:	clover-ant-2.4.2 idea7-2.4.2-v20081224113000 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 FishEye's internal data in a MySQL or PostgreSQL database, as an alternative to the built-in HSQLDB.
- 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.

ℹ JIRA is Atlassian's issue tracking product, which can be used to manage projects and associated work.

⚠ Before you begin: Ensure that you configure your JIRA instance to enable sub-tasks, enable unassigned issues and allow Remote API access . The instructions on this page have been tested with JIRA 3.13.4.

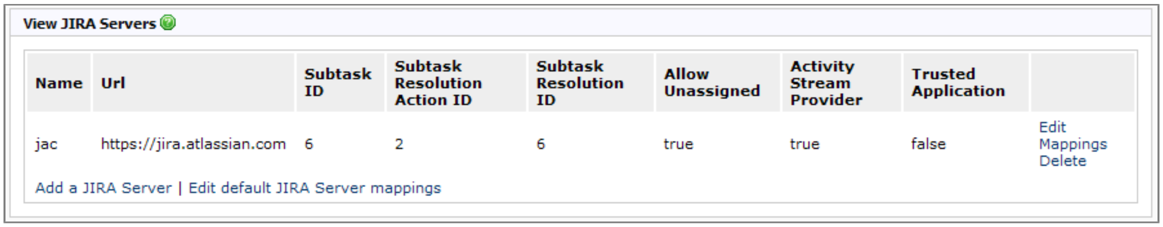
On this page:

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

Opening the Administration Screen for JIRA Integration

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

Screenshot: The View JIRA Servers Page



View JIRA Servers

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

Add a JIRA Server | Edit default JIRA Server mappings

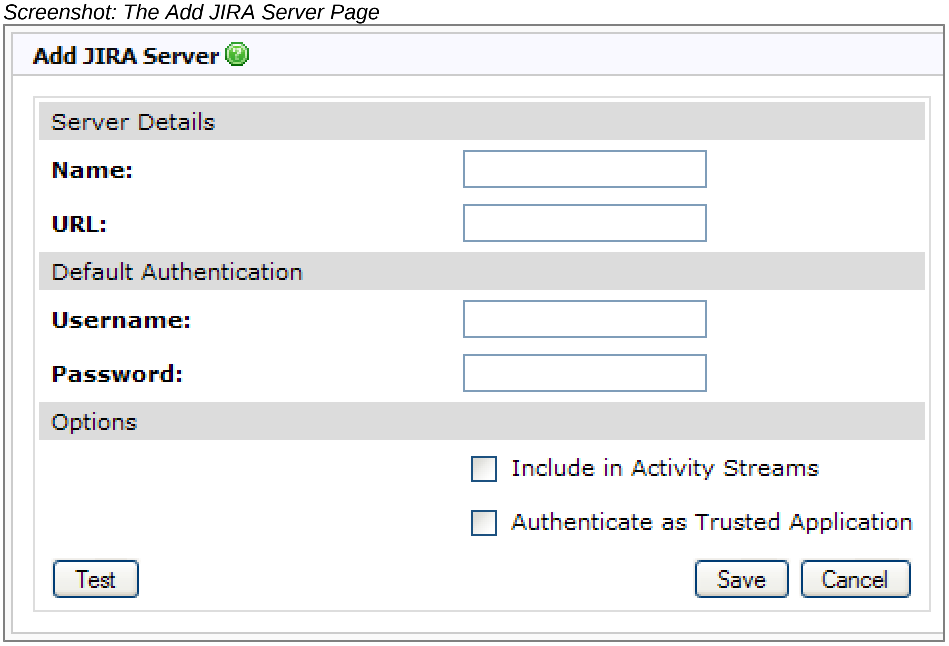
On the View JIRA Servers page, you can carry out a number of operations as listed on this page.

Adding a New JIRA Server

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

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

Screenshot: The Add JIRA Server Page



Add JIRA Server

Server Details

Name:

URL:

Default Authentication

Username:

Password:

Options

☐ Include in Activity Streams

☐ Authenticate as Trusted Application

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

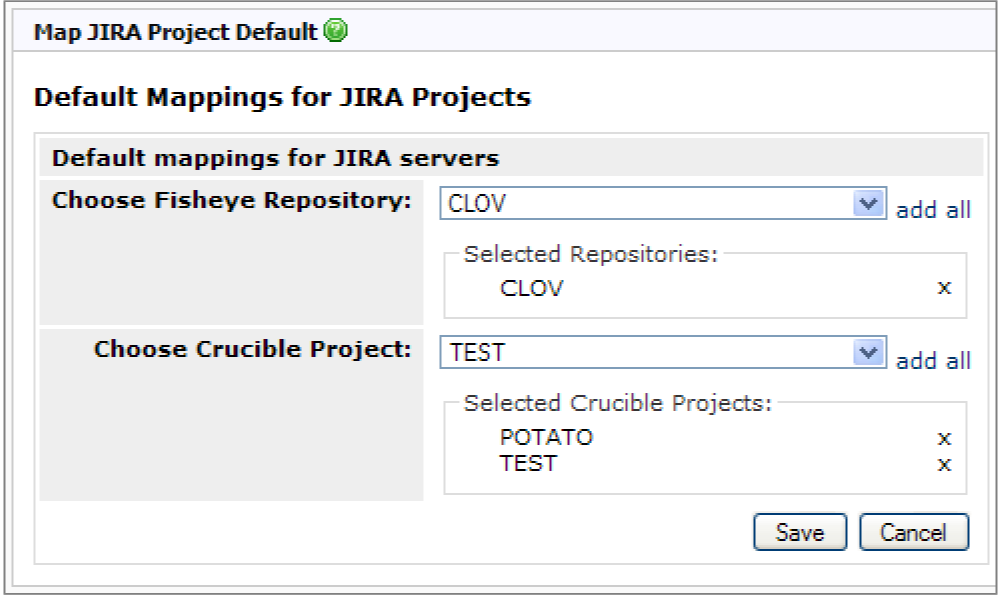
Once you've filled out the necessary fields, click '**Test**' to ensure that your details are correct. If you have a positive message return from the test, click '**Save**'.

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.

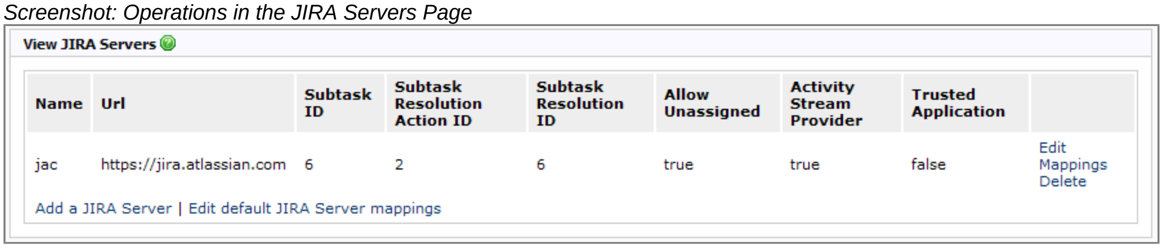
Once you've finished, click '**Save**'.

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

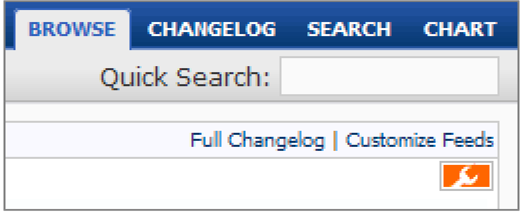
FishEye Search Enhancements

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

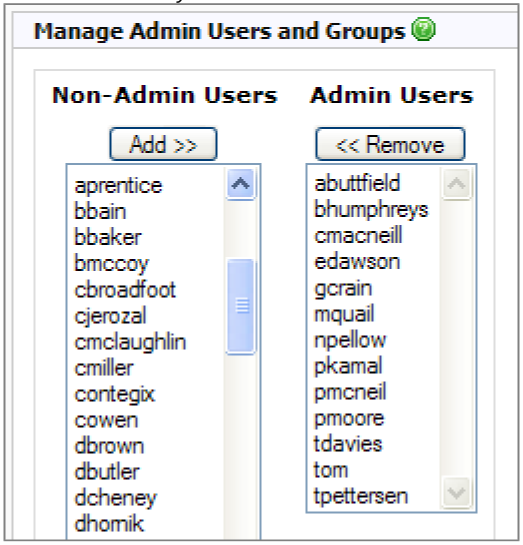
All of the improvements made to Quick Search and General Search are available when using Advanced Search and EyeQL.

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

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

Admin Users

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

3

Remote API Improvements

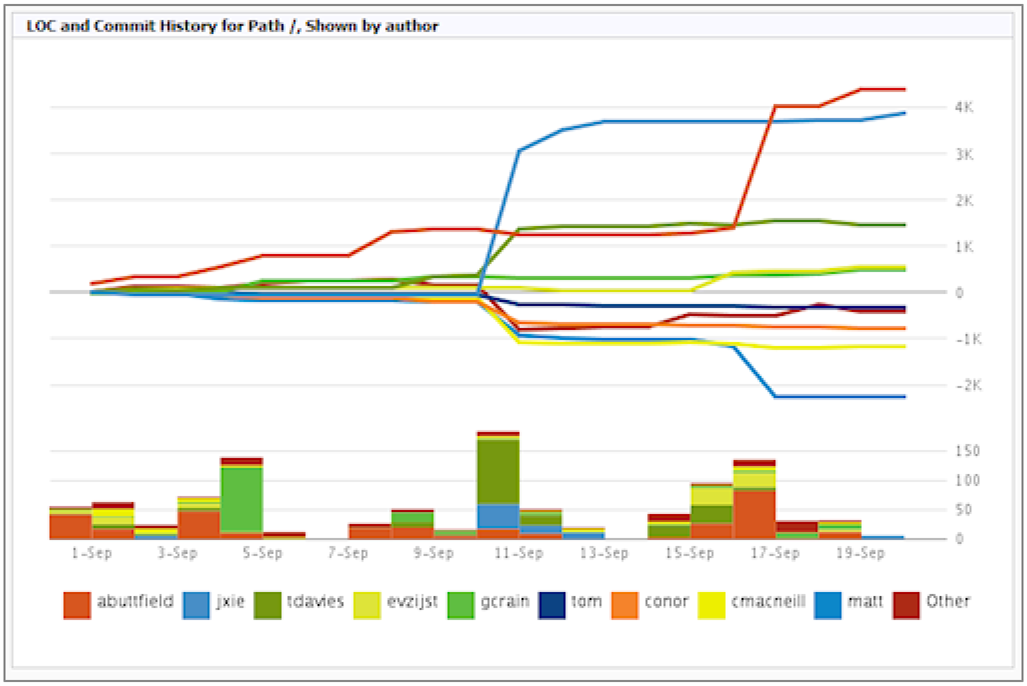
Enhancing the Remote API, the FishEye team have incorporated a **maxreturns** option, which lets you control the quantity of returned results. Also the range of functions is extended with the new **ancestor return** clause and **history search** items. Finally, Perforce data is now exposed in the remote 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 (69 issues)

Key	Summary	Priority	Status
FE-638	webwork 2.2.6 is not setting svnsymbolic in editrepository	❗	Closed
FE-434	show match-in-context in quicksearch (hit highlighting)	⬆	Closed
FE-437	improve performance of filename searches in quicksearch	⬆	Closed
FE-438	Do content searches in quicksearch	⬆	Closed
FE-467	Calculating the correct version for diffs doesn't work for perforce	⬆	Closed
FE-288	Repositories still occasionally get stuck in Stopping state	⬆	Closed
FE-82	Re-index request: show message "Could not stop repository within 20 seconds. Re-index aborted."	⬆	Closed
FE-650	Documentation: New 'Advanced' mode hides attributes in Add Repository screen	⬆	Closed

FE-536	"List Repositories" method in the remote API documentation	⇧	Closed
FE-584	Create the ability for customers to create support cases via Fisheye	⇧	Closed
FE-605	added files appear as empty diffs	⇧	Closed
FE-550	improve catch-all svn symbolic regex	⇧	Closed
FE-600	TODO appearing in UI for comments	⇧	Closed
FE-503	com.cenqua.fisheye.svn.SvnCache is throwing NPEs	⇧	Closed
FE-435	Quick-search redo UI requirements	⇧	Closed
FE-393	Use a single regularexpression to catch all tag/branch/trunk patterns	⇧	Closed
FE-566	diff-to-previous on annotate page 404s	⇧	Closed
FE-617	for new svn repositories, defaut for t/b/t should be None	⇧	Closed
FE-521	CLONE -StackOverflow	⇧	Closed
FE-378	review multithreading of RevCacheReader	⇧	Closed
FE-296	Get Id button does not work in Trusted Application screen under IE	⇧	Closed
FE-483	Ensure all user preferences are in user profile	⇧	Closed
FE-554	Add LIMIT clause to EyeQL documentation	⇧	Closed
FE-658	Chart constraint dropped on second level subdir	⇧	Closed
FE-578	"Search just <repo:parh>" breadcrumb links are borked - escapes parameter separators	⇧	Closed
FE-674	RSS Feed Title is missing space	⇧	Closed
FE-675	RSS Feed Entries have almost no information in title	⇧	Closed
FE-668	"Edit repository details" throws an NPE when a p4 repo has invalid info	⇧	Closed
FE-630	Bundle SAL 1.1 in FishEye	⇧	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-389	only ask group to do group-membership tests for crowd users	⇧	Closed
FE-338	Please add ability to specify initial revision from which to begin initial scan	⇧	Closed
FE-494	Add a link to the changeset on the annotation page	⇧	Closed

FE-549	Search tokenizes on underscores	⇧	✓ Closed
FE-436	Improved Quick Search	⇧	✓ Closed
FE-515	Allow P4 label scanning to be skipped	⇧	✓ Closed
FE-541	Allow limiting of number of results returned by remote API	⇧	✓ Closed
FE-505	Retrieve Password	⇧	✓ Closed
FE-504	command line reindex doens't work when loopback is not 127.0.0.1	⇧	✓ Closed
FE-500	Expose Changeset "Fixes Perforce Jobs" data in EyeQL (and REST api)	⇧	✓ Closed
FE-555	Documentation: Add maxReturn parameter to remote API	⇧	✓ Closed
FE-552	Have an error page rather than 403 page when SVN permission denied	⇧	✓ Closed
FE-524	upgrade to trusted apps 1.0, remove seraph dep	⇧	✓ Closed
FE-301	IndexOutOfBoundsException when opening annotated view	⇧	✓ Closed
FE-562	resolve springsource/log4j versioning problem	⇧	✓ Closed
FE-559	Add 'ancestor' return clause to EyeQL	⇧	✓ Closed
FE-532	upgrade to webwork 2.2.7 (fixes security problem)	⇧	✓ Closed
FE-540	"Data Types and Structures" information in the Remote API page	⇧	✓ Closed
FE-352	Add a shortcut in browse for making a diff for a given revision and the one before it	⇧	✓ Closed
FE-479	next and previous links on diff and annotation pages	⇧	✓ Closed
FE-511	NPE when configuration file was not found	⇧	✓ Closed
FE-685	Automatical user management from Crowd	⇩	✓ Closed
FE-649	Cannot specify starting revision when creating perforce repo. The option only appears in the edit screen	⇩	✓ Closed
FE-607	Create documentation for new feature, create support issues via fisheye Admin > Sysinfo screen	⇩	✓ Closed
FE-495	Self Signup layout borked	⇩	✓ Closed
FE-464	upgrade to latest spring (at least 2.5.4)	⇩	✓ Closed
FE-651	Weight quicksearch results by date	⇩	✓ Closed
FE-687	RSS Feed entry items truncate differently to 1.5.x	⇩	✓ Closed

FE-513	Redirected to javax.servlet.ServletException 500 error page when accessing to a disabled/stopped repository	⬇	✓ Closed
FE-491	Hide optional fields in repo setup	⬇	✓ Closed
FE-428	Check for updates option	⬇	✓ Closed
FE-636	Make update polling configurable in admin section	⬇	✓ Closed
FE-319	Can't access /admin/ when logged in via Crowd	⬇	✓ Closed
FE-643	"Request Garbage Collection" link on SysInfo/Support page redirects badly	⬇	✓ Closed
FE-596	CSS syntax highlighting omissions	⬇	✓ Closed
FE-482	SVN repositories default to UTF-8	⬇	✓ Closed
FE-678	Link to file view in quicksearch results	↓	✓ Closed
FE-529	Add an "Add Repository" link to the top of the repository list page in admin screen	↓	✓ Closed
FE-487	Remove the colon trailing repository names on the Fisheye home page	↓	✓ Closed

FishEye 1.6 Changelog

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-1481	Unable to open connection to Subverion version 1.6.1	!	✓ Closed
FE-1759	Handle log message rewriting	⬆	✓ 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-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-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-250	What happens when SVN operation times out?	↑	Closed
FE-194	NPE when restarting repository	↑	Closed
FE-127	Sort repository lists on screens alphabetically	↑	Closed
FE-63	Document custom content capability	↑	Closed
FE-1141	Print Stack Trace after exception in fisheye command line	↓	Closed
FE-1046	WrappedRequest does not delegate to super for all get* methods when the private params map is not null	↓	Closed
FE-508	A problem occurred starting your repository: Repository is not stopped (com.cenqua.fisheye.rep.RepositoryHandle$StateException)	↓	Closed
FE-343	Option for lines of code in all /trunk subdirectories	↓	Closed
FE-191	FishEye update method: SVN hook script	↓	Closed
FE-583	Chart the change in lines of code over a period	▾	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-830	Need to update svnkit that resolves an NPE to do with NTLM authentication.	❗	Closed
FE-624	Indexing changesets which contain a large number of changes (e.g. brach copy, etc) can cause out of memory errors	❗	Closed
FE-878	Changing display name doesn't work	⇑	Closed
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-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-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-827	Reading p4 annotate fails with "NumberFormatException" when customers annotate ouput does not being with <changeset>:	⇧	Closed
FE-754	Getting "Not Inserting Revision blah because a parent a parent directory is missing	⇧	Closed
FE-852	Diffs link shows up in HTML changeset emails for binary files	⇩	Closed
FE-807	Update the jsvn scripts to point to the new 1.2.1r5011.atlassian.jar version of svnkit	⇩	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-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-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
FE-795	Please don't show: "You have 29 days before your maintenance expires ..."	⇩	Closed
FE-361	Sexy chart type switching	⇩	Closed
FE-241	Same time commits wrongly ordered in changelog	⇩	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-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-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-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-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-707	Strange svn blame issue	↑	✓ 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-305	stdcxx indexing problems(s)	↑	✓ Closed
FE-267	Document setting up a CVS mirror with rsync	↑	✓ Closed
FE-723	quicksearch jump to exact file match	↓	✓ Closed
FE-708	highlighting in keywords should be done manually	↓	✓ Closed
FE-706	Misc quicksearch improvements	↓	✓ 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-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

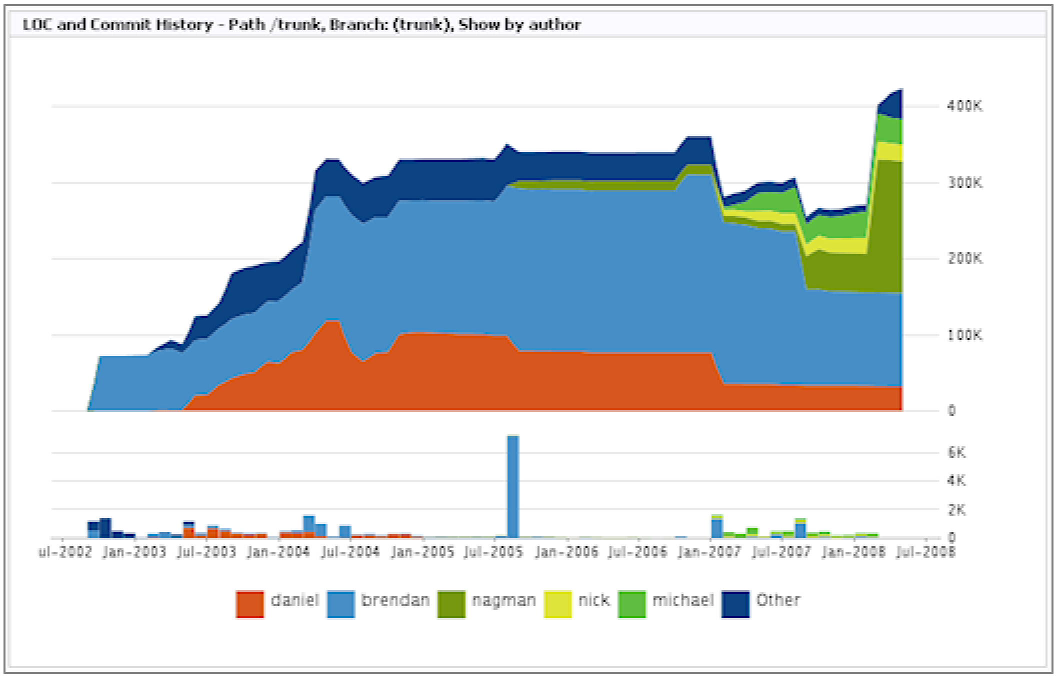
1

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

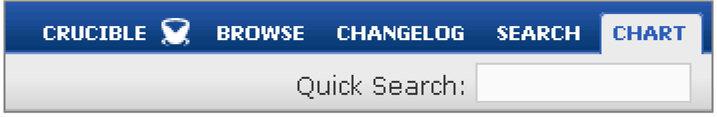


2

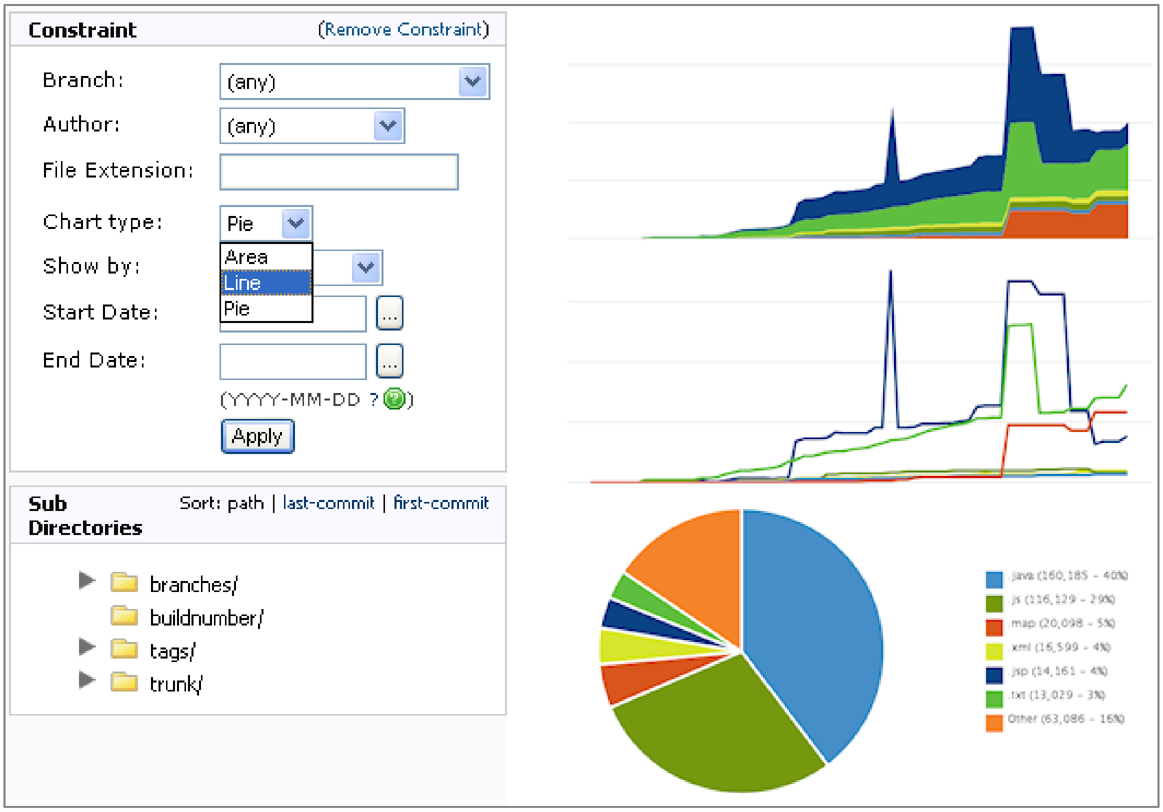
Charting improvements

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 ?)

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.

Numerous improvements and bug-fixes

JIRA Issues (54 issues)

Key	Summary	Priority	Status
FE-394	Can't open page on Multithreading without giving credentials	⇧	✓ Closed
FE-326	FishEye always sorts with oldest first	⇧	✓ 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-331	User credentials are case-sensitive	⇧	✓ Closed

FE-325	Add syntax highlighting for ActionScript (.as) files	⇧	Closed
FE-323	Manually request incremental scan from commandline	⇧	Closed
FE-320	Showing files as directories in tags directories	⇧	Closed
FE-317	Improve email notification handling for commit comments with newlines	⇧	Closed
FE-303	fix single-sign-off problem with Crowd	⇧	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-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-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-335	Default Certificate Timeout value for Trusted Applications should not be 0	⇩	✓ Closed
FE-333	WARN - error parsing file with regexp	⇩	✓ Closed
FE-321	FishEye occasionally does not get the author and/or comment for change sets	⇩	✓ 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-302	Upgrade crowd support to 1.3	⇩	✓ Closed
FE-210	doc: tag names in FE, eyeql help	⇩	✓ Closed
FE-5	[P4] need to consider "utf8" etc file types	⇩	✓ Closed
FE-336	Random order of operations on Trusted Applications	▾	✓ Closed
FE-170	capitalisation on 'User Profile' page is a little inconsistent	▾	✓ Closed

FishEye 1.5 Changelog

On this page:

- From 1.5.3 to 1.5.4
- From 1.5.2 to 1.5.3
- From 1.5.1 to 1.5.2
- From 1.5.0 to 1.5.1

From 1.5.3 to 1.5.4

1 August 2008

This release contains minor improvements and bug fixes.

Errors were reported by the JIRA trusted connection.

- APP_UNKNOWN; Unknown Application: {0}; ["confluence:4557196"]

JIRA Issues (22 issues)

Key	Summary	Priority	Status
FE-558	Stop Fisheye repository does not work	⇧	Closed
FE-662	Customization of GUI	⇧	Closed
FE-593	Certain Input Files causes excessive memory usage in Syntax Highlighting	⇧	Closed
FE-588	Directories that are deleted then replaced by a symbolic link will cause a "is not a directory in filesystem" when svn info command is called.	⇧	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-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-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-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
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-306	To handle SVN operation hang during FishEye scanning	↓	✓ 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-223	Problem getting diff information for rev2232	↑	✓ Closed
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-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-383	Linecount graph calculation performance improvements	⇩	✓ 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-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-372	charting title is wrong	⇧	✓ 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
FE-415	upgrade svnkit to 1.1.7	⇩	✓ Closed
FE-369	clicking (eg) Changelog from Chart tab loses directory constraint	⇩	✓ Closed
FE-360	Small Line History Chart Tweaks	⇩	✓ Closed
FE-376	Old charting code can probably be removed now	▽	✓ 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.

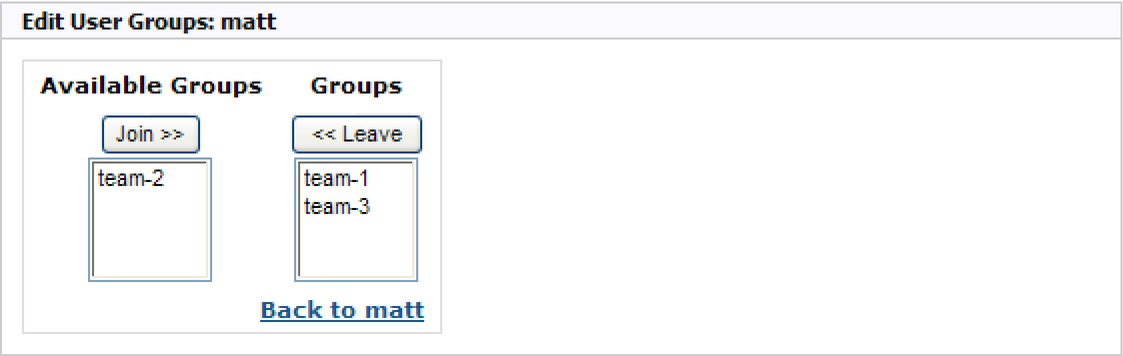
Highlights of FishEye 1.4

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

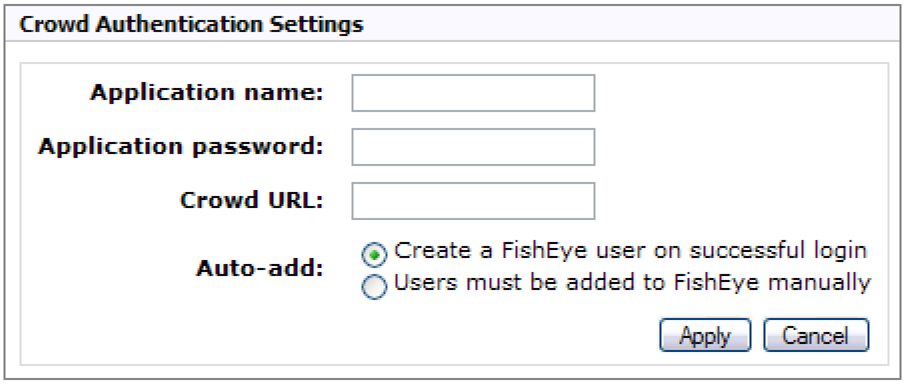
Available Groups: team-2

Groups: team-1, team-3

2

Crowd/SSO support

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



Crowd Authentication Settings

Application name:

Application password:

Crowd URL:

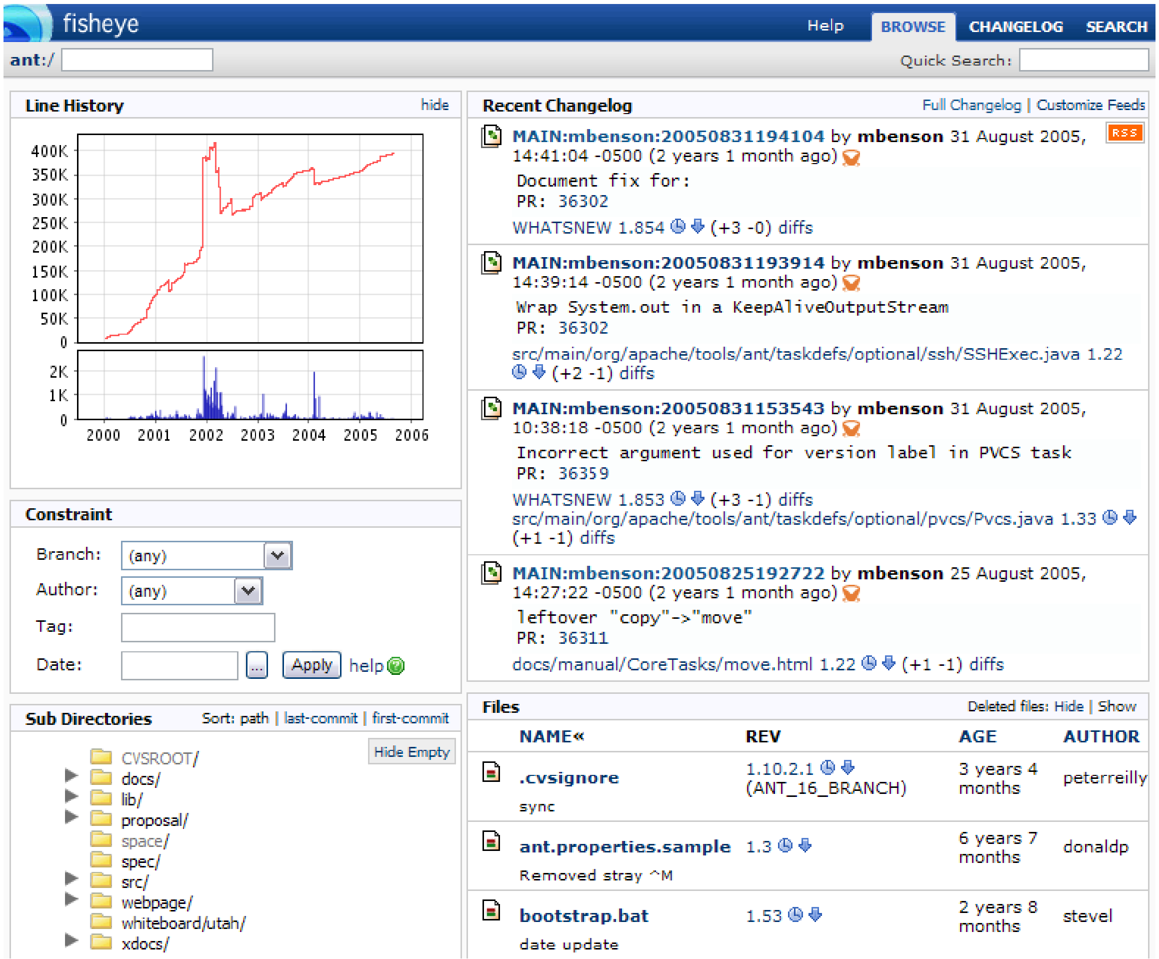
Auto-add: ● Create a FishEye user on successful login
○ Users must be added to FishEye manually

3

Crucible integration

Closer integration between FishEye 1.4 and Crucible 1.2:

- Links to existing Crucible reviews on the FishEye screens. So you can see which files/changesets have been reviewed.
- Search for Crucible data via EyeQL. For example, you can search for files that have not yet been reviewed.



Recent Changelog

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

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/

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-147	Subversion revision indexing fails	⇪	✓ Closed
FE-132	Report FishEye and/or API version via the API	⇪	✓ Closed
FE-163	DbException: Problem getting diff information for rev1	⇧	✓ Closed
FE-149	eyeql textbox too big in safari3.0.4	⇧	✓ Closed
FE-148	Ability to delete user groups	⇧	✓ 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-128	Make online help link to CAC documentation	⇧	✓ Closed
FE-120	NPE when using ajp for authentication	⇧	✓ Closed
FE-119	constraint in email watches can become corrupted by url escaping	⇧	✓ Closed
FE-114	Regex syntax highlighting StackOverflowError	⇧	✓ Closed
FE-111	FishEye should ignore FISHEYE_HOME variable	⇧	✓ Closed
FE-103	ability to rename users	⇧	✓ 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-81	Make email optional for self-registered users	⇧	✓ Closed
FE-73	Fisheye javadoc needs to be uploaded to docs.a.c	⇧	✓ 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-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
FE-156	Bug/feature request link at bottom of screens is wrong	⇩	✓ Closed
FE-133	Cascading documentation links within FishEye	⇩	✓ 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-118	per-rep linkers don't necessarily trump default linkers when they match the same string	⇩	✓ Closed
FE-108	Syntax Highlighting is wrong (keywords match within identifiers)	⇩	✓ Closed
FE-100	allow usernames to contain the @ character	⇩	✓ Closed
FE-99	Linker Update requires a restarty	⇩	✓ Closed
FE-86	Default linkers don't warn that restart is required	⇩	✓ Closed
FE-72	Document "File History View Mode"	⇩	✓ Closed
FE-71	Incorrect spelling 'Seach' on Simple Search screen	⇩	✓ Closed
FE-57	Add 'reindex repository' command to FishEyeCtl interface	⇩	✓ Closed

FishEye 1.4 Changelog

On this page:

- From 1.4.2 to 1.4.3
- From 1.4.1 to 1.4.2
- From 1.4 to 1.4.1

From 1.4.2 to 1.4.3

7 February 2008

This release contains bug fixes.

JIRA Issues (35 issues)			
Key	**Summary**	**Priority**	**Status**
FE-309	Trusted Application not stored properly in configuration file	❗	Closed
FE-284	login error and logout pages return blank page	❗	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-304	Improve documentation on recommended hardware and software (JVM) settings for FishEye	⇧	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-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-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-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
FE-272	Linker regex application is case-sensitive	⇩	✓ Closed
FE-208	Linkers setting in Repository Defaults ignored	⇩	✓ Closed
FE-28	Move ' Improve FishEye Scan Performance' guide to FishEye permanent documentation space	⇩	✓ 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 2.0 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).