Space Details

Key:	FISHEYE
Name:	FishEye 1.3
Description:	Latest documentation for FishEye, your view into your source code repository
Creator (Creation Date):	(Sep 06, 2007)
Last Modifier (Mod. Date):	smaddox (Oct 05, 2007)

Available Pages

- TreeNavigation
- __newreleaseFishEye
- FishEye Documentation Home🏠
  - About FishEye
  - FishEye Administrator's Guide
    - 1. Managing your Repositories
      - 1.1 Adding a Repository
      - 1.2 Configuring Repository Details
      - 1.3 Repository Options
        - Allow (Process)
        - Hidden Directories
        - Indexer
        - Linkers
        - Properties
        - Tarball Settings
        - Updater
        - Watches
      - 1.4 CVS
      - 1.5 Perforce
      - 1.6 Subversion
        - SVN fisheye.access
        - SVN Tag & Branch Structure
    - 2. Setting up a Repository Client
      - Perforce Client
      - Subversion Client
    - 3. Configuring ViewVC Compatibility
    - 4. Setting up your Web Server
      - Configuring the FishEye Web Server
      - Integrating with Other Web Servers
    - 5. Configuring SMTP
    - 6. Setting up your Users and Security
      - Configuring Anonymous Access
      - Configuring Built-In Authentication
      - Configuring External Authentication Sources
        - AJPv13 Authentication

            - Custom Authentication
            - Host-Based Authentication
            - LDAP Authentication
    - 7. Backing Up and Restoring Config Data
    - 8. Advanced Administration Options
        - Command-Line Options
        - Customising FishEye's Look & Feel
        - Environment Variables
            - Setting JAVA_HOME
        - Tuning FishEye
- FishEye Installation & Upgrade Guide
    - FishEye Installation Guide
        - 1. System Requirements
        - 2. Installing FishEye
        - 3. Configuring FishEye
    - FishEye Release Notes
        - FishEye 1.3 Release Notes
            - FishEye 1.3 Changelog
            - FishEye 1.3 Upgrade Guide
    - FishEye Upgrade Guide
- FishEye Knowledge Base
    - CVS FAQ
        - How does FishEye calculate CVS changesets?
    - Installation & Configuration FAQ
        - Can FishEye be run as a Windows service?
    - 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?
        - 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.
        - 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.
        - 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
- FishEye User's Guide
    - 1. Fisheye Quick Start Guide
    - 2. Using the FishEye Screens

    - Browsing through a Repository
    - Searching the Repository
    - Viewing a File History
    - Viewing the Changelog
- 3. Changing your User Profile
- 4. EyeQL Reference Guide
- 5. Antglob Reference Guide
- 6. Date Expressions Reference Guide

TreeNavigation

This page last changed on Sep 18, 2007 by smaddox.

Index

__newreleaseFishEye

This page last changed on Sep 21, 2007 by smaddox.

⚠ FishEye 1.3.6 has now been released. See the bug fixes and improvements in the changelog.

FishEye Documentation Home

This page last changed on Oct 04, 2007 by smaddox.

Getting Started with FishEye 1.3	Using/Administering FishEye 1.3
About FishEye Feature Tour Guided Tour Download FishEye Installation Guide Release Notes | Upgrade Guide	Quick Start Guide Administrator's Guide User's Guide EyeQL Reference Guide Support | Forums | FAQ

Previous Versions	PDF Format
Please see older versions of FishEye	You can download the FishEye documentation in PDF format.

Error rendering macro: java.lang.NullPointerException

About FishEye

This page last changed on Oct 04, 2007 by pmoore.

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 or the Online Forums.

System Requirements

See the FishEye Installation Guide.

What's New in FishEye 1.3?

See the FishEye Release Notes.

FishEye Administrator's Guide

This page last changed on Oct 04, 2007 by smaddox.

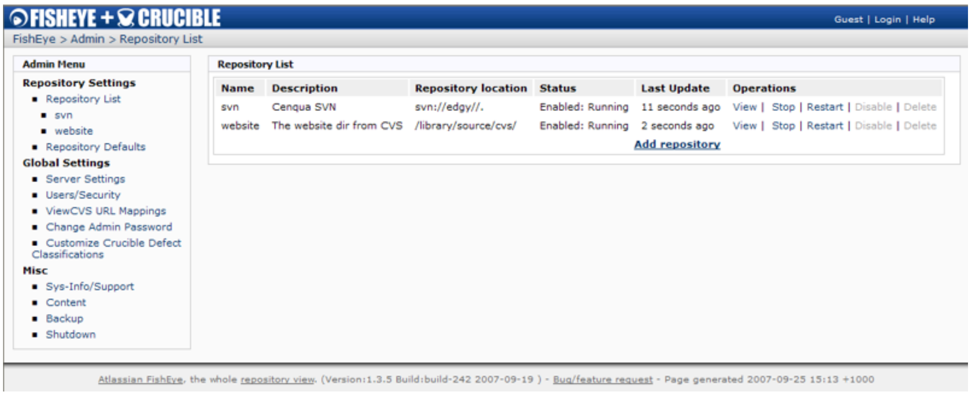
Once you have installed and configured FishEye, you can access the Administration pages at http://HOSTNAME:8080/admin/.

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



FishEye > Admin > Repository List

Admin Menu

Repository Settings

- Repository List
  - svn
  - website
- Repository Defaults

Global Settings

- Server Settings
- Users/Security
- ViewCVS URL Mappings
- Change Admin Password
- Customize Crucible Defect Classifications

Misc

- Sys-Info/Support
- Content
- Backup
- Shutdown

Repository List

Name	Description	Repository location	Status	Last Update	Operations
svn	Cenqua SVN	svn://edgy//.	Enabled: Running	11 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

Add repository

Atlassian FishEye, the whole repository view. (Version:1.3.5 Build:build-242 2007-09-19) - Bug/feature request - Page generated 2007-09-25 15:13 +1000

Information in this Administrator's Guide:

- 1. Managing your Repositories
- 2. Setting up a Repository Client
- 3. Configuring ViewVC Compatibility
- 4. Setting up your Web Server
- 5. Configuring SMTP
- 6. Setting up your Users and Security
- 7. Backing Up and Restoring Config Data
- 8. Advanced Administration Options

1. Managing your Repositories

This page last changed on Sep 18, 2007 by smaddox.

- 1.1 Adding a Repository
- 1.2 Configuring Repository Details
- 1.3 Repository Options
- 1.4 CVS
- 1.5 Perforce
- 1.6 Subversion

1.1 Adding a Repository

This page last changed on Sep 26, 2007 by smaddox.

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.

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.

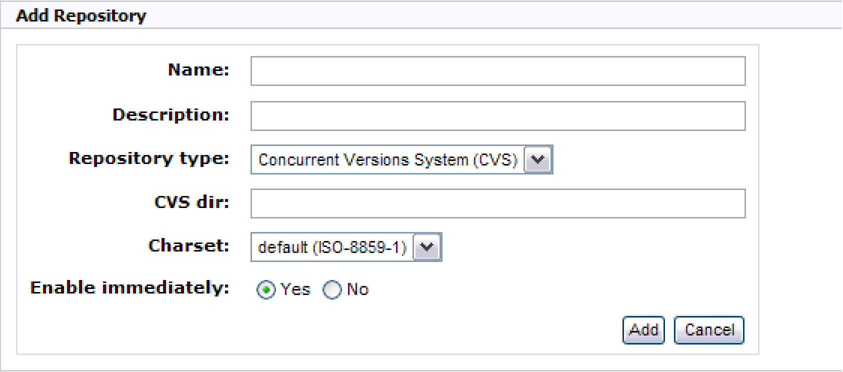
To add a repository,

1. From the 'Admin Menu', click 'Repository List' then '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
- Subversion (SVN)
- Perforce

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

1.2 Configuring Repository Details

This page last changed on Sep 26, 2007 by smaddox.

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
- Subversion (SVN)
- Perforce

1.3 Repository Options

This page last changed on Oct 02, 2007 by smaddox.

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'.

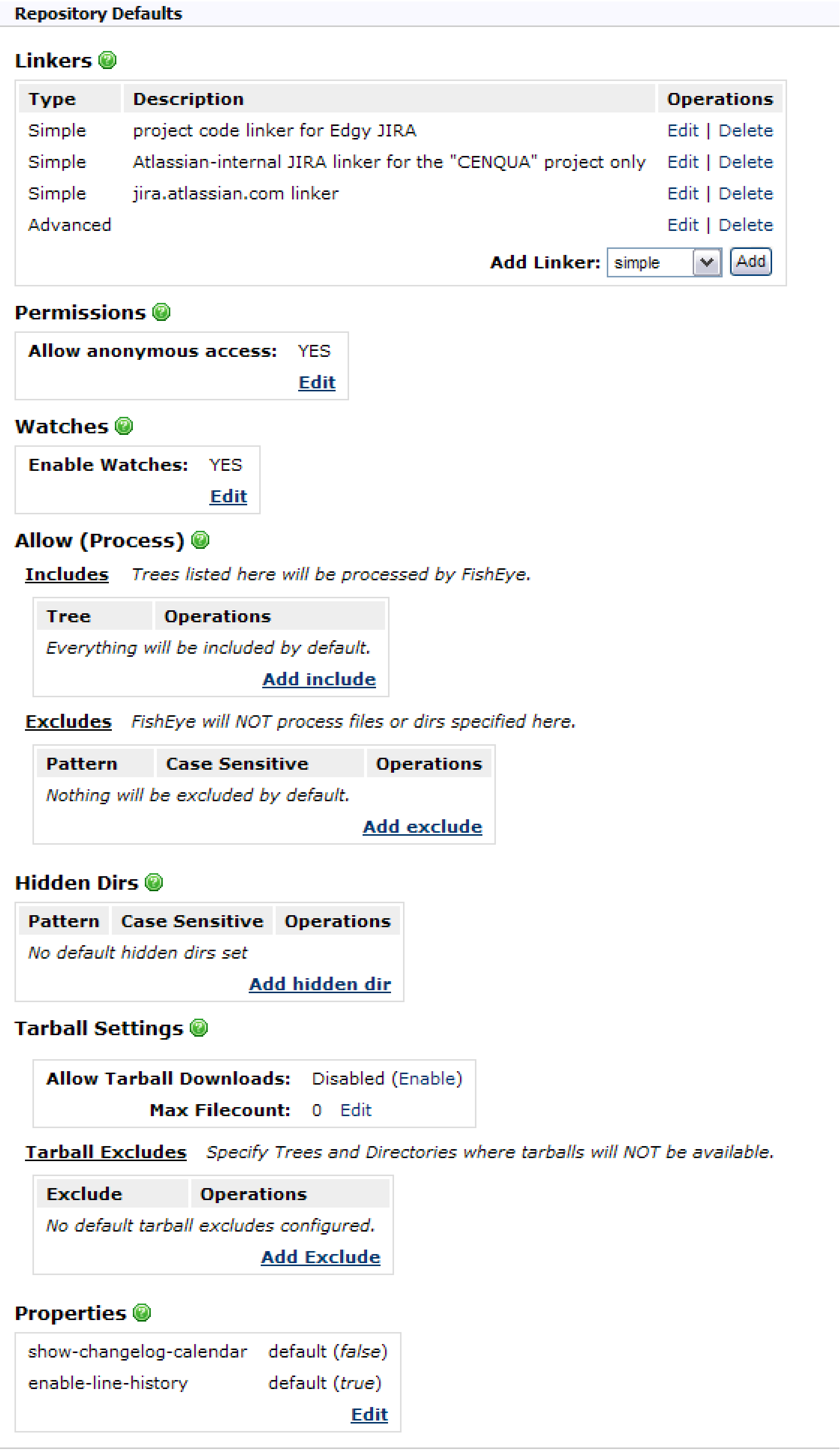
⚠ Some changes will require the repository to be restarted, while others will require the repository to be re-indexed. FishEye will advise you if this is the case when you make the change. You can restart a repository from the 'Repository List'.

Repository options:

- Allow (Process)
- Hidden Directories
- Indexer
- Linkers
- Properties
- 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*)

Edit

Allow (Process)

This page last changed on Oct 02, 2007 by smaddox.

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.

The 'Includes' subsection defines what subtrees of your repository FishEye will access. It defaults to 'everything'. If you specify some 'include' directories, then FishEye will process only those directories (and all their subdirectories).

The 'Excludes' subsection allows you to specifically exclude files and directories that may have been included. FishEye will not process these files/directories. Each exclude is an Antglob. Examples:

- `/CVSROOT/**` or just `/CVSROOT/` excludes `/CVSROOT` and all its children.
- `*.OBJ` excludes any OBJ files.

> ⚠ Changes to Includes and Excludes do not take effect until a full re-slurp of your repository is performed.

Hidden Directories

This page last changed on Oct 02, 2007 by smaddox.

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

This page last changed on Oct 02, 2007 by smaddox.

The 'Indexer' repository option allows an administrator to manually trigger the following actions:

Full Scan	CVS only. Scan the whole repository for any changes since the last scan.
Re-index Repository	Delete the current cache and re-index the repository from the beginning.
Rescan Revision Properties	SVN only. In Subversion it is possible to allow non-versioned properties to be updated by committers. For example, the check-in comment. 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.

Linkers

This page last changed on Oct 02, 2007 by smaddox.

FishEye can detect special substrings in commit messages, and hyperlink those substrings to other systems. This is particularly useful if you use an issue tracking system, and put the issue identifiers into your commit messages. The 'Linkers' repository option allows you to define the substrings and their related URLs.

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

Here are some examples of simple linkers.

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

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

Properties

This page last changed on Oct 02, 2007 by smaddox.

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.	

Tarball Settings

This page last changed on Oct 02, 2007 by smaddox.

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

This page last changed on Oct 02, 2007 by smaddox.

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

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.

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`.

Updater (SVN)

Poll Period	How often FishEye will check to see if there have been any new commits into the SVN 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.

Watches

This page last changed on Oct 02, 2007 by smaddox.

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.

1.4 CVS

This page last changed on Sep 26, 2007 by smaddox.

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:

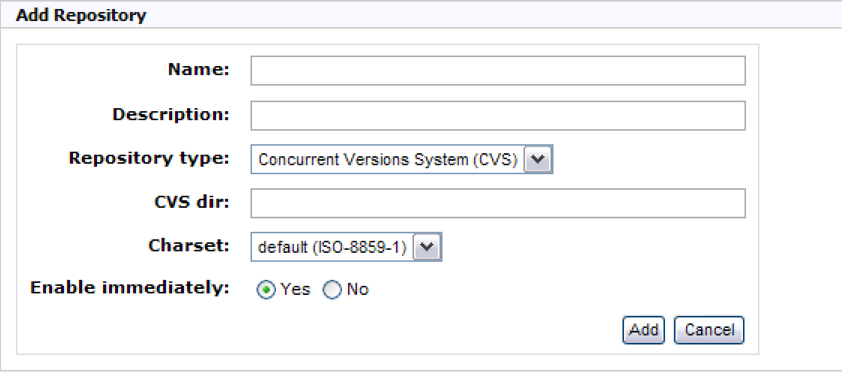
- Updater.
- Indexer.

⚠ To add a CVS repository, FishEye must have file system access to the repository.

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

1.5 Perforce

This page last changed on Oct 02, 2007 by smaddox.

When adding or managing a Perforce repository, you can:

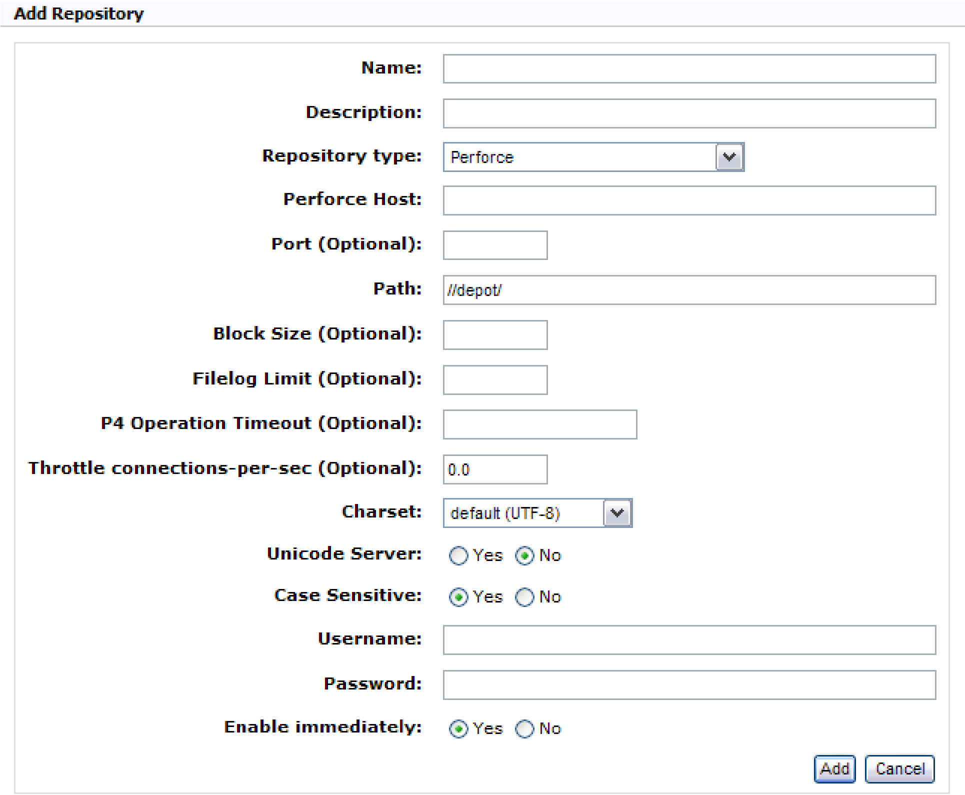
- Define repository details, as described below.
- Set FishEye's repository options.
- Set up a Perforce client.

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.
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.
Username/Password	The credentials to use if your repository requires authentication.
Enable immediately	Controls whether FishEye will immediately enable this repository, which starts the initial scan. If you wish to do some further configuration before the scan starts, then select 'No'. You can enable a repository later from the Repository List.

Screenshot: Adding a Perforce Repository



Add Repository

Name:	
Description:	
Repository type:	Perforce
Perforce Host:	
Port (Optional):	
Path:	//depot/
Block Size (Optional):	
Filelog Limit (Optional):	
P4 Operation Timeout (Optional):	
Throttle connections-per-sec (Optional):	0.0
Charset:	default (UTF-8)
Unicode Server:	○ Yes ● No
Case Sensitive:	● Yes ○ No
Username:	
Password:	
Enable immediately:	● Yes ○ No

1.6 Subversion

This page last changed on Oct 04, 2007 by smaddox.

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.

⛔ 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 slurp time and impact runtime performance. Please refer to the instructions on tag and branch configuration.

✓ Knowledge Base

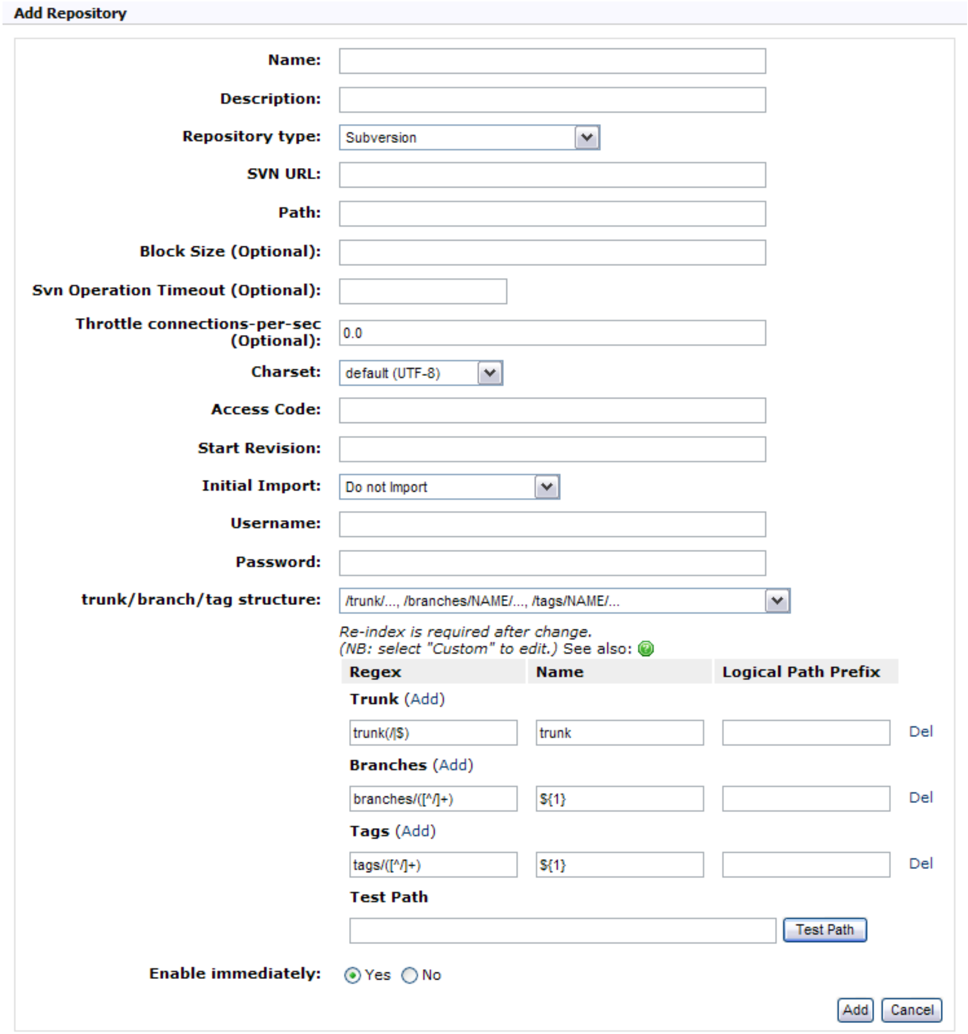
You may find some useful information in the Knowledge Base too.

SVN Repository Details

Name	A name for this repository. The name may contain alphanumeric, underscore, '-' or '.' characters. Use 'svn' if you can't think of a better name.
Description	A short description of this repository.
SVN URL	The Subversion URL to your repository, such has `svn://svn.foo.com/` or `file:///var/svn`
Path	The sub-tree within your repository 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, 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.

Screenshot: Adding a SVN Repository



Add Repository

Name:	
Description:	
Repository type:	Subversion
SVN URL:	
Path:	
Block Size (Optional):	
Svn Operation Timeout (Optional):	
Throttle connections-per-sec (Optional):	0.0
Charset:	default (UTF-8)
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

This page last changed on Sep 25, 2007 by smaddox.

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 does not exist or does not match with that configured in FishEye, FishEye will immediately disconnect from the repository.

> ⚠ By default, FishEye will have access to your repository.

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

This page last changed on Oct 02, 2007 by smaddox.

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.

For more information on tag/branch layout, see Repository Layout in the Subversion documentation.

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.

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.

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.

2. Setting up a Repository Client

This page last changed on Sep 18, 2007 by smaddox.

- Perforce Client
- Subversion Client

Perforce Client

This page last changed on Oct 02, 2007 by smaddox.

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'.

> ℹ Files wrongly 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

This page last changed on Oct 02, 2007 by smaddox.

FishEye can communicate with any server running Subversion 1.1 or later, but it needs to use a Subversion client to do so. You must configure FishEye to use one of the two clients specified below, either the native or the SVNKit client.

If you do not have all the necessary components (see below) of the native client installed, you may find it easier to use the 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.

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 (with one warning - see below), or set the FISHEYE_LIBRARY_PATH environment variable before starting FishEye.

Pre-compiled native clients are available from the Subversion site.

Native Client Configuration

You can configure your Subversion client in the Server Settings section of the FishEye Administration screens, or by editing the `<svn-config>` section of your `config.xml`. If you change these settings, you need to restart FishEye.

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. Note: 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.

SVNKit Client

If you have difficulty acquiring a native Subversion client which contains the JavaHL bindings, you can try to use SVNKit, which is a 100% Java Subversion library.

Note: Prior to SVNKit version 1.1.0, SVNKit was called JavaSVN. We recommend using the 1.1.0 version or later, as it is much improved over earlier releases.

To use SVNKit:

- Disable the native client, by clearing the 'JAR' and 'Dynamic Library' fields described above (or remove the `<svn-config>` element from your `config.xml` file).
- Download SVNKit from the above URL.
- Unzip the SVNKit download, and copy all the `.jar` files to `$FISHEYE_INST/lib`.

⚠ 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 below in FishEye's log, you will need to either swap to native JavaHL layer or upgrade your subversion server to 1.3 or later.

Example exceptions:

- SEVERE: assert #B
- Checksum mismatchwhile reading representation:

3. Configuring ViewVC Compatibility

This page last changed on Oct 03, 2007 by smaddox.

FishEye contains a URL-compatibility mode with the ViewVC (formerly known as ViewCVS) and CVSWeb tools. For example, a ViewVC URL of the form

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

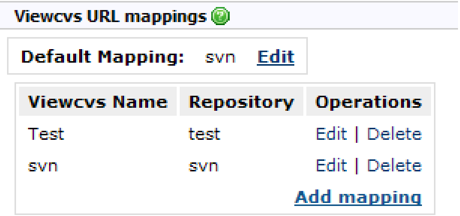
can be viewed in FishEye at

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

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

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

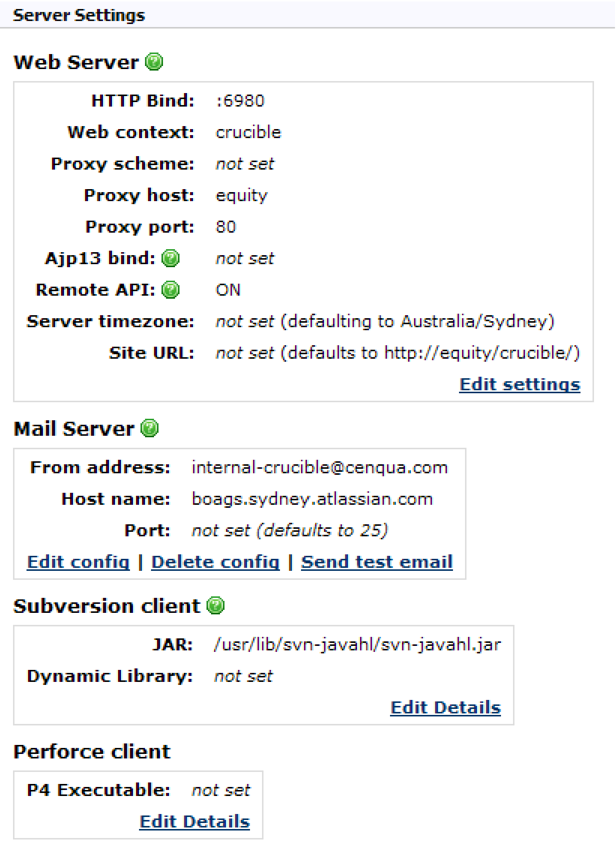
4. Setting up your Web Server

This page last changed on Oct 03, 2007 by smaddox.

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

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

Screenshot: Configuring Server Settings



Server Settings

Web Server

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

Edit settings

Mail Server

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

Edit config | Delete config | Send test email

Subversion client

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

Edit Details

Perforce client

P4 Executable: *not set*

Edit Details

Configuring the FishEye Web Server

This page last changed on Oct 03, 2007 by smaddox.

⚠ You need to restart FishEye for any changes to these settings to take effect.

HTTP Bind	The address the FishEye webserver will bind to. Can be just a port number, or an address and port number. If no host is specified, then FishEye will bind to all available interfaces. Examples are: `:8080`, `hostname:8080`, `10.0.0.11:80`. At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.
Web context	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. For example, if you specify a web context of 'fisheye' then FishEye will be accessible from `http://HOST:PORT/fisheye/` instead of `http://HOST:PORT/`.
Proxy scheme	Can be set if you are forwarding through to FishEye from another webserver.
Proxy host	Can be set if you are forwarding through to FishEye from another webserver.
Proxy port	Can be set if you are forwarding through to FishEye from another webserver.
AJP13 Bind	The bind address for ajpv13. If no host is specified, then FishEye will bind to all available interfaces. Examples are: `:8009`, `hostname:8009`, `10.0.0.11:8009`. At least one of 'AJP13 Bind' or 'HTTP Bind' must be set.
Remote API	Enables/disables FishEye's Remote API. Clicking on the help link will take you to the API doc.
Server timezone	The timezone to use within FishEye. This timezone is used when displaying dates and parsing EyeQL date expressions. If blank, then the timezone of the underlying host is used.
Site URL	The base URL for this FishEye instance. If not specified, FishEye will attempt to determine this value.

See also Subversion Client settings.

Integrating with Other Web Servers

This page last changed on Sep 21, 2007 by smaddox.

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:8080`, then you can set FishEye's Proxy host and Proxy port parameters to `WEBHOST` and `80`.

If the primary web server is using SSL, then you should set Proxy scheme to `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`.

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:8080/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:8080/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.

5. Configuring SMTP

This page last changed on Oct 03, 2007 by smaddox.

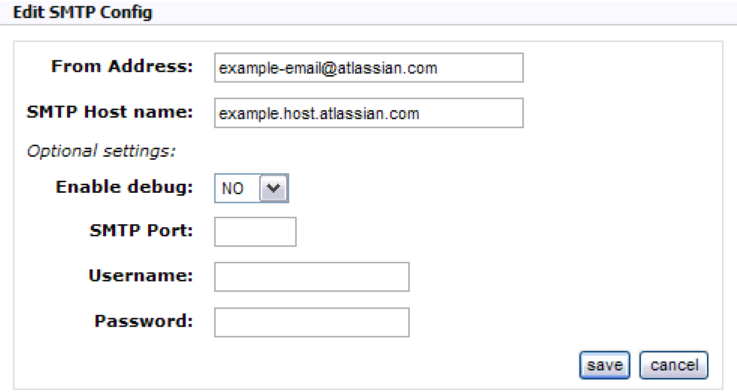
To configure SMTP settings, click 'Server Settings' on the 'Admin Menu'.

You can enter the following parameters:

From Address	The from email address used when FishEye sends an email, e.g. 'fisheye-noreply@example.com'
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.
Username & Password	Optional. Username and password for authenticated SMTP access.

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

Screenshot: Configuring SMTP



Edit SMTP Config

From Address: example-email@atlassian.com

SMTP Host name: example.host.atlassian.com

Optional settings:

Enable debug: NO

SMTP Port:

Username:

Password:

6. Setting up your Users and Security

This page last changed on Oct 03, 2007 by smaddox.

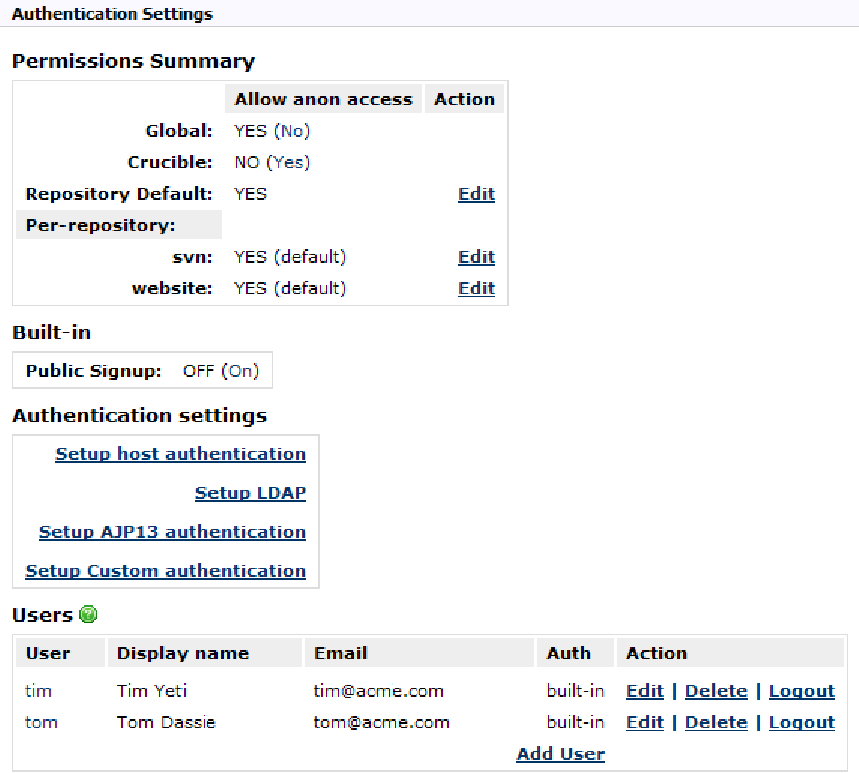
You can implement access control using a set of users built into FishEye, or you can have FishEye look in an external authentication source for users, passwords and permissions.

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

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	Action
Global:	YES (No)	
Crucible:	NO (Yes)	
Repository Default:	YES	Edit
Per-repository:		
svn:	YES (default)	Edit
website:	YES (default)	Edit

Built-in

Public Signup:	OFF (On)

Authentication settings

Setup host authentication

Setup LDAP

Setup AJP13 authentication

Setup Custom authentication

Users ⊕

User	Display name	Email	Auth	Action
tim	Tim Yeti	tim@acme.com	built-in	Edit | Delete | Logout
tom	Tom Dassie	tom@acme.com	built-in	Edit | Delete | Logout
			Add User	

Configuring Anonymous Access

This page last changed on Sep 25, 2007 by smaddox.

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

- Global.
- Repository default.
- Per repository.

From the 'Admin Menu', choose one of the following options to change the anonymous access settings:

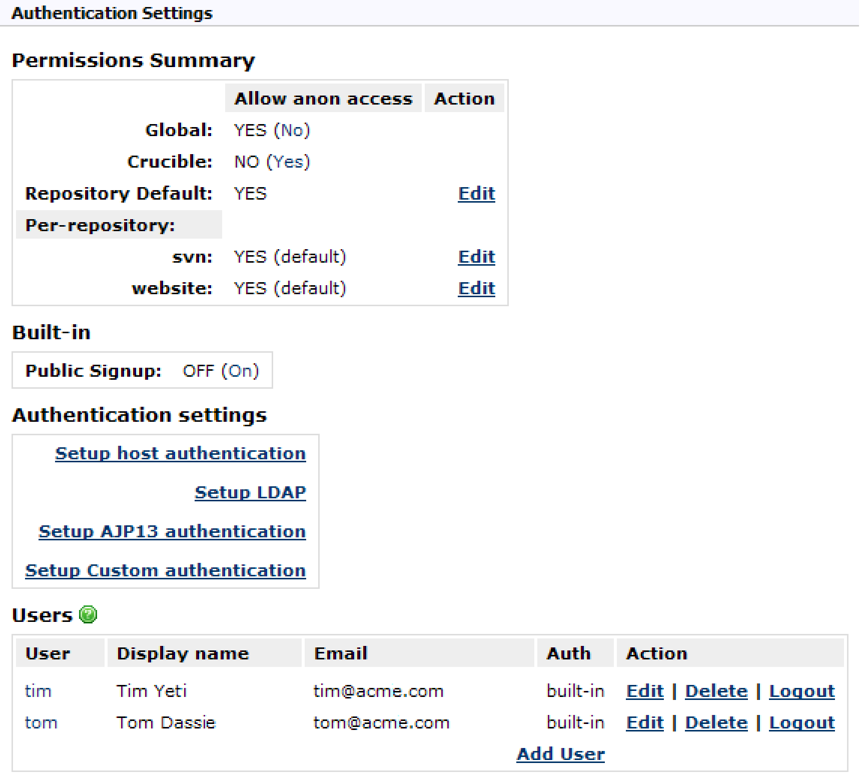
- 'Users/Security' – Allows you to change anonymous access at all levels.
- 'Repository Defaults' – Allows you to change the default setting for repositories.
- A repository name – Allows you to change the setting for the specific repository.

Configuring Built-In Authentication

This page last changed on Oct 03, 2007 by smaddox.

You can implement access control via a user list built into FishEye.

To change this setting and to manage your FishEye users, click 'Users/Security' on the 'Admin Menu'.
Screenshot: Authentication Settings



Authentication Settings

Permissions Summary

	Allow anon access	Action
Global:	YES (No)	
Crucible:	NO (Yes)	
Repository Default:	YES	Edit
Per-repository:		
svn:	YES (default)	Edit
website:	YES (default)	Edit

Built-in

Public Signup:	OFF (On)

Authentication settings

Setup host authentication
Setup LDAP
Setup AJP13 authentication
Setup Custom authentication

Users

User	Display name	Email	Auth	Action
tim	Tim Yeti	tim@acme.com	built-in	Edit | Delete | Logout
tom	Tom Dassie	tom@acme.com	built-in	Edit | Delete | Logout

Configuring External Authentication Sources

This page last changed on Oct 03, 2007 by smaddox.

Although FishEye always maintains a list of users internally, you can have FishEye authenticate and authorise users against an external authentication source.

To set an external authentication source, click 'Users/Security' on the 'Admin Menu'.

Only one authentication can be set at one time. However, each repository can have the option as to whether to use the authentication or not.

To change authentications you will need to remove the settings that are already configured. Just click the 'Remove' link. You will then be presented with the option to add a different authentication.

FishEye currently supports:

- LDAP authentication.
- Host-based authentication. This is implemented using PAM on Linux/Solaris/OS-X, and Local/Domain Accounts on Windows.
- AJPv13 authentication.
- Custom authentication.

AJPv13 Authentication

This page last changed on Sep 25, 2007 by smaddox.

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
```

Custom Authentication

This page last changed on Oct 04, 2007 by smaddox.

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 attached zip archive of the JavaDoc.

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.ExampleFishEyeAuthenti`
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.

Host-Based Authentication

This page last changed on Oct 04, 2007 by smaddox.

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. For example, `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 last changed on Oct 03, 2007 by smaddox.

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.

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:

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=${USERNAME})(dn=${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`

7. Backing Up and Restoring Config Data

This page last changed on Oct 03, 2007 by smaddox.

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 following files will be backed up:

- `config.xml`
- `fisheye.license`
- `var/data/data0.bin`

⚠ No repository cache data will be backed up.

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.

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
```

8. Advanced Administration Options

This page last changed on Sep 18, 2007 by smaddox.

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

Command-Line Options

This page last changed on Sep 21, 2007 by smaddox.

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.

Options:

`--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 ause the page to take a long time to render. This option disables the calculation that determines emptiness.
`--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.

start

This 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`.

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 ...]`*

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`.

backup

Usage:
`fisheyectl backup` *`[filename]`*

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`.

Customising FishEye's Look & Feel

This page last changed on Oct 04, 2007 by smaddox.

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.

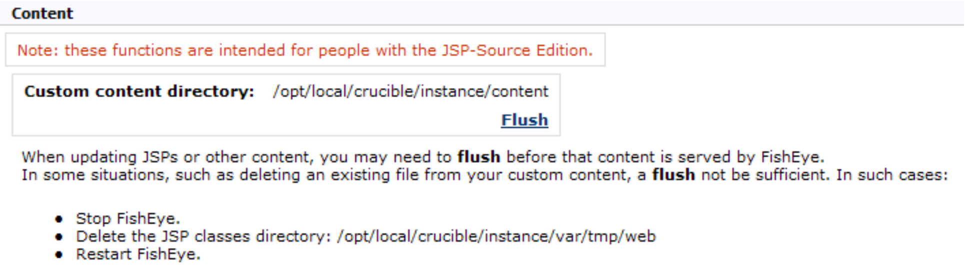
If you have a commercial license assigned to your account, you will see a source download option on your download page.

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

Flush

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

This page last changed on Oct 03, 2007 by smaddox.

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. With a Sun JVM, for example, you would use:

```
FISHEYE_OPTS=-Xmx256m
```

This would give FishEye a 256 MByte heap. See Tuning FishEye for more information.

It is possible to put other JVM options into the FISHEYE_OPTS environment variable. For example, 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.

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, 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_HOME

FISHEYE_HOME is the location of the FishEye application. By default FishEye will set this to the directory above the `fisheyectl` script.

FISHEYE_INST

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

Setting JAVA_HOME

This page last changed on Sep 20, 2007 by smaddox.

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 'nix based systems

There are many ways you can do it on 'nix 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.

Tuning FishEye

This page last changed on Oct 03, 2007 by smaddox.

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.

If you do run into 'Out of Memory' errors, you will need to increase the heap size and restart FishEye.

For Subversion repositories, it is also possible to reduce FishEye's memory footprint by reducing the BlockSize parameter.

FishEye Installation & Upgrade Guide

This page last changed on Sep 11, 2007 by smaddox.

- FishEye Installation Guide
  - 1. System Requirements
  - 2. Installing FishEye
  - 3. Configuring FishEye
- FishEye Release Notes
  - FishEye 1.3 Release Notes
- FishEye Upgrade Guide

FishEye Installation Guide

This page last changed on Oct 04, 2007 by smaddox.

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.

- 1. System Requirements
- 2. Installing FishEye
- 3. Configuring FishEye

1. System Requirements

This page last changed on Oct 04, 2007 by smaddox.

Java Runtime	A JDK or JRE version 1.5 or greater. You can download a Java Runtime for Windows/ Linux/Solaris. On MacOSX the JVM is available as part of the OS install here. Note: There appeared to be a problem with the first release of the JRockit 5.0 JVM (R25.0.0-75) when using FishEye. If you use the JRockit JVM we recommend you use a later release.
CVS	If you are using CVS, FishEye needs read-access your CVS repository via the file system. It does not support protocols such as `pserver` at the moment.
Subversion (server)	FishEye can communicate with any repository running Subversion 1.1 or later.
Subversion (client)	FishEye uses the native Subversion client installed on your system. It must be version 1.2 or later and include the 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 2006.2 or later.
Operating System	FishEye is a pure Java application and should run on any platform provided the above requirements are satisfied.

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.

SCM Support

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

- CVS (and CVS-NT
- Subversion
- Perforce

Support for other version control systems (such as ClearCase) will be added to FishEye in the future. Let us know what SCM you would like to see supported by logging and/or voting for a JIRA issue.

2. Installing FishEye

This page last changed on Oct 05, 2007 by mquail.

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

FishEye Prerequisites

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_HOMEenvironment 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.

> ⚠ Read-only access for FishEye
>
> We recommend you run FishEye as a user that has only read access to your repository.

FishEye Layout

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/fisheye.license	FishEye license.
/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.

> ⚠ A separate FISHEYE_INST location is recommended for production installations of FishEye.

When the FISHEYE_INST environment variable is set, FishEye's directory layout becomes:

$FISHEYE_INST/config.xml	
$FISHEYE_INST/fisheye.license	
$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_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.

3. Configuring FishEye

This page last changed on Oct 04, 2007 by smaddox.

Initial Configuration

FishEye runs its own HTTP web server, and additionally listens on a socket for administration/shutdown commands. These default to `:8080` and `127.0.0.1:8079` 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/
  $ ./run.sh
  ```

Once started, FishEye will run its own HTTP web server, on port 8080 by default.

You can access FishEye immediately by going to `http://HOSTNAME:8080/` 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:8080/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.

FishEye Release Notes

This page last changed on Sep 21, 2007 by smaddox.

⚠ FishEye 1.3.6 has now been released. See the bug fixes and improvements in the 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.

- 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 Release Notes

- FishEye 1.3 Release Notes

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

FishEye 1.3 Release Notes

This page last changed on Sep 21, 2007 by smaddox.

⚠ FishEye 1.3.6 has now been released. See the bug fixes and improvements in the changelog.

FishEye 1.3 contains many bug fixes and improvements, and adds support for Perforce.

Upgrading FishEye

You can now download FishEye from here. Information on installing FishEye can be found here. If upgrading from a previous version, please follow the Upgrade Guide.

Highlights of FishEye 1.3

- Support for the Perforce version control system.
- SVN properties are now shown.
- Quicksearch now searches for changeset ids.
- New 'mixed' chart on annotation pages, showing author-over-time breakdown.
- Side by side diffs (1.3.1)
- Refer to the changelog for more details.

FishEye 1.3 Changelog

This page last changed on Sep 21, 2007 by smaddox.

On this page:

- From 1.3.5 to 1.3.6
- From 1.3.4 to 1.3.5
- From 1.3.3 to 1.3.4
- From 1.3.2 to 1.3.3
- From 1.3.1 to 1.3.2
- From 1.3 to 1.3.1
- From 1.3beta9 to 1.3
- From 1.3beta8 to 1.3beta9
- From 1.2.5 to 1.3beta8

From 1.3.5 to 1.3.6

This is a small bug-fix release. It addresses a stack-overflow problem for some configurations.

From 1.3.4 to 1.3.5

Note: Upgrading to this version will force a complete re-index of P4 repositories.

Improvements

- User-friendly UI-based license entry and maintenance.
- Syntax highlight files when displaying a diff.
- Add Ruby and Coldfusion syntax highlighting. Look for custom syntax files in `FISHEYE_INST/syntax`.
- EyeQL enhancement: a new clause to match on changeset ids `csid = "1234"`.
- EyeQL enhancement: more return types such as `isAdded` and `isBinary`.

Fixes

- Many performance improvements when scanning Perforce repositories.
- Changelog missing some changesets in CVS when there is a path constraint.
- Handle svn diff output in languages other than English.
- Fix problem with long directory names wrapping in the directory tree.

From 1.3.3 to 1.3.4

- Fix compatibility between Cenqua and Atlassian licenses.
- [SVN] Convert slash / characters to a hyphen - in tag/branch names.
- [SVN] Better handling of precedence of tag/branch/trunk symbolic rules.
- [SVN] Add a manual test field in Admin screens to test symbolic rule setup.
- [P4] Add ability to configure Perforce updater in Admin screens.
- [P4] Some improvements when scanning Perforce repositories.

From 1.3.2 to 1.3.3

This build allows FishEye to be used with Atlassian licenses.

From 1.3.1 to 1.3.2

- Fix potential XSS vulnerability in quick-search page.
- Fix problem sending watch emails where the commit message contains a tab character.
- [SVN] Add support for requesting a rescan between given revisions.
- [SVN] Improve scan performance, and better handle add operations from outside FishEye's view of the repository.
- [SVN] Improve scan performance by not fetchings diffs for binary files.
- [SVN] Timeout settings now configurable via Admin screens.
- [SVN] Display SVN properties at the directory level.

- [P4]] Address problem fetching labels for files.
- Fix Javascript problem in IE when logging into the Admin screens.

From 1.3 to 1.3.1

- The truncate diff setting should now work in Internet Explorer.
- Fix issue with duplicate paths in tarball generation.
- [P4] An empty <p4-config> element would cause an error.
- Unknown repos now return a 404 status rather than 500.
- [SVN] Handle empty content files when using SvnKit.
- [CVS] Allow $ in author names.
- FishEye now uses the tabwidth setting in each user's profile.
- [SVN] Fix issue where FishEye incorrectly states that no username was supplied.
- Fix IE7 directory spacing problem.
- Implement side-by-side diffs.

From 1.3beta9 to 1.3

- Various improvements when scanning Perforce repositories.
- [SVN] Fix for problem with diff hyperlinks to re-added files.
- Fix problem where some paths were not correctly html-escaped.
- Fix 'NoSuchFieldError deferredExpression' problem on some platforms (due to a 3rd-party library being included twice).
- Ensure LDAP connections are closed in all situations.

From 1.3beta8 to 1.3beta9

> ⚠ Upgrading to 1.3beta9 will force a complete re-index of CVS repositories.

- Upgrade JVM requirement to 1.5+.
- Upgrade embedded HTTP engine (Jetty). This fixes some bugs and improves performance under load.
- Fix a performance problem (esp. under load). "Recent Changes" pages should return much faster now.
- Fix a very slow memory leak when FishEye is under load (for example, when it is being crawled by a web spider).
- Fix a problem where daily-emails would break after a backup was performed.
- [CVS] Fix an error introduced when FishEye builds its repository cache. This requires a full re-scan of CVS repositories.
- [CVS] Fix a problem where FishEye could not parse in RCS files author names that were only numerical digits.
- [CVS] Fix bug when creating tar/zip files from a branch constraint.
- [SVN] FishEye will now timeout long running SVN connections that have blocked.
- [SVN] Fix problem where FishEye was not storing SVN properties correctly.
- [SVN] Fix a bug when entering a revision beyond the current last revision in quick search.

From 1.2.5 to 1.3beta8

- [SVN] When importing a repository from a given start revision, you can now nominate if it should import the state of the repository at that revision, or just import changes made after that revision.
- [CVS] Fix a bug where FishEye would send out watch emails for historical changesets after a re-index.
- Performance improvements to changeset page when one of the files in the changeset has a very large history.
- [SVN] Some changes that improve the speed of the initial-scan for some SVN repositories.
- Fix a bug when FishEye generates RSS feed urls constrained by author, when the author has an "@" in their name.
- [SVN] Fix a bug when a tag is deleted (as part of a move).

FishEye 1.3 Upgrade Guide

This page last changed on Sep 20, 2007 by smaddox.

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

- Please read the Release Notes and Upgrade Guides for the version you are upgrading to, as well as any versions you are skipping.
- Follow the instructions on upgrading FishEye.

FishEye Upgrade Guide

This page last changed on Oct 04, 2007 by smaddox.

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, Changelog and Upgrade Guides for the version you are upgrading to, as well as any versions you are skipping.
- If you are upgrading version 0.x to 1.0, be aware of the following important changes that occurred between 0.10 and 1.0RC1:
  - The FishEye scripts (fisheyectl, start, stop, etc) have been moved from `/FISHEYE_HOME/` to `/FISHEYE_HOME/bin/`.
  - You can now split part of your FishEye installation into an 'instance' directory `FISHEYE_INST`. This makes upgrades much easier.

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. Extract the new FishEye version to a directory, leaving your FISHEYE_INST environment variable set to its existing location.
2. Start FishEye from the new installation.
3. 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. Copy your `fisheye.license` to `/NEW_FISHEYE/`.
7. Follow any version-specific instructions found in the Release Notes.

FishEye Knowledge Base

This page last changed on Sep 17, 2007 by smaddox.

FishEye Knowledge Base

Answers to commonly raised questions about configuring and using FishEye.

- CVS FAQ
  - How does FishEye calculate CVS changesets?
- Installation & Configuration FAQ
  - Can FishEye be run as a Windows service?
- 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?
  - 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.
  - 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.
  - 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

This page last changed on Sep 17, 2007 by smaddox.

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?

This page last changed on Sep 17, 2007 by smaddox.

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.

Installation & Configuration FAQ

This page last changed on Sep 17, 2007 by smaddox.

FishEye Installation & Configuration FAQ

- 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 on this FishEye forum thread.

Can FishEye be run as a Windows service?

This page last changed on Sep 17, 2007 by smaddox.

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 on this FishEye forum thread.

To install on Windows:

1. Unzip the wrapper zip file into your `FISHEYE_HOME` directory.
2. Run `Fisheye-Install-NTService.bat`, found in `FISHEYE_HOME/wrapper/bin`.
3. Start the Fisheye service under the Windows Control Panel.

Subversion FAQ

This page last changed on Sep 17, 2007 by smaddox.

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.
- 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'

This page last changed on Sep 17, 2007 by smaddox.

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.

This page last changed on Sep 17, 2007 by smaddox.

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?

This page last changed on Sep 17, 2007 by smaddox.

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.

Why do I need to describe the branch and tag structure for Subversion repositories?

This page last changed on Sep 25, 2007 by smaddox.

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 slurp time and may reduce runtime performance.

Troubleshooting

This page last changed on Sep 17, 2007 by smaddox.

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.
- 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.
- 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.

This page last changed on Sep 20, 2007 by smaddox.

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.

I have installed FishEye, and the inital scan is taking a long time. Is this normal?

This page last changed on Sep 17, 2007 by smaddox.

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.

I have installed FishEye, but there is no data in the Changelog.

This page last changed on Sep 17, 2007 by smaddox.

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.

Message 'org.tigris.subversion.javahl.ClientException svn Java heap space'

This page last changed on Sep 20, 2007 by smaddox.

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.

This page last changed on Sep 17, 2007 by smaddox.

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

This page last changed on Sep 25, 2007 by mquail.

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 User's Guide

This page last changed on Sep 11, 2007 by smaddox.

- 1. Fisheye Quick Start Guide
- 2. Using the FishEye Screens
  - Browsing through a Repository
  - Searching the Repository
  - Viewing a File History
  - Viewing the Changelog
- 3. Changing your User Profile
- 4. EyeQL Reference Guide
- 5. Antglob Reference Guide
- 6. Date Expressions Reference Guide

1. Fisheye Quick Start Guide

This page last changed on Oct 04, 2007 by smaddox.

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

⚠ Read-only access for FishEye

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

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 8080. You can access FishEye immediately by going to `http://HOSTNAME:8080/` in a browser.

⚠ Default ports

By default, FishEye will listen on port 8080 for HTTP requests. It also listens on 127.0.0.1:8079 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.

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:8080/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:8080/`.
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
```

2. Using the FishEye Screens

This page last changed on Oct 04, 2007 by smaddox.

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.

When you click the 'FishEye icon' in the top left hand corner, you will see the list of FishEye repositories. Underneath the FishEye icon is the directory that is currently being browsed.

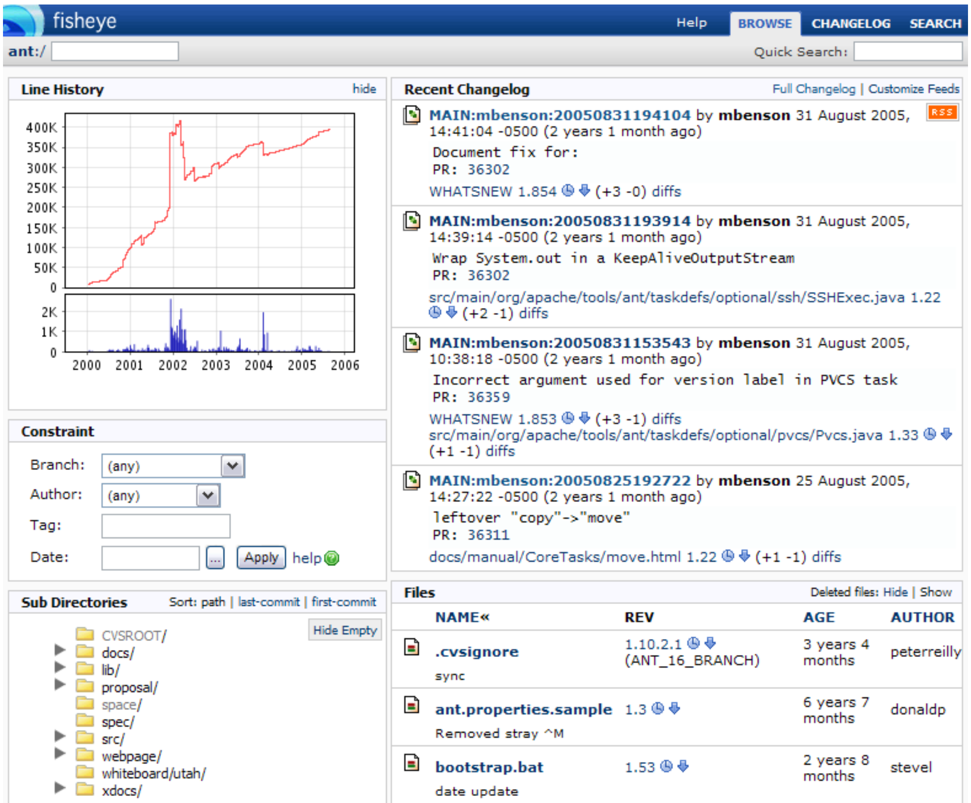
The user currently logged in will be shown at the top right hand corner, or it will say 'Guest' if nobody is logged in. If a user is logged in then they can change FishEye settings such as passwords, notifications and display settings by clicking the 'Profile' link.

Tabs

The tabs at the top right hand corner provide different views into the repository:

- Browse
- Changelog
- Search

Screenshot: Using FishEye



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

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

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

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

Browsing through a Repository

This page last changed on Oct 04, 2007 by smaddox.

The Browse view is the first screen you see when you sign into FishEye. Click the Browse tab in the top right hand corner to get to this screen.

The Browse view lets you explore the revisions, files and directories in your repository. This screen helps you quickly navigate to the file you are looking for, as well as presenting useful information about the directory you are looking at.

Recent Changelog Pane

The top of the right-hand column shows the most recent changesets for this directory subtree. The RSS icon allows you to subscribe to an RSS feed of the recent changes in this subtree.

Files Pane

The list of files is shown in the 'Files' pane below the Recent Changelog pane on the right-hand side. You can sort the Files pane by name, age or author. Click a file name to view the file history.

Line History Graph Pane

The 'Line History' pane is at the top of the left-hand column. 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.

Constraint pane

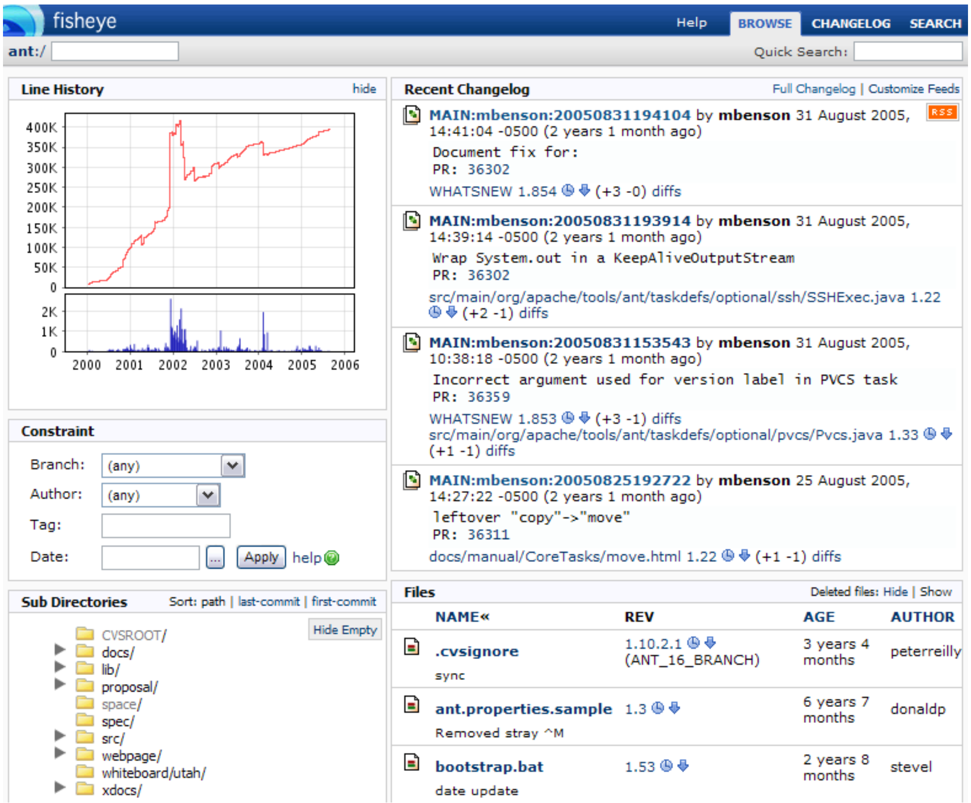
The 'Constraint' pane is on the left, under the 'Line History' pane. You can specify a constraint that controls the information that is shown in the Browse view.

Branch	Will show files and recent changes on only that branch.
Author	Displays the most recent revision in each file that was checked in by the given author. Shows only recent changes checked in by the given author.
Tag	Shows only files/revisions that are tagged with the given tag.
Date	Shows only revisions and changesets that were created on or before that date. Must be of the form YYYY-MM-DD, YYYY-MM or YYYY (you can use / instead of -).

Sub Directories Pane

The 'Sub Directories' pane is on the left, under the 'Constraint' pane. It shows a list of the different folders under the repository. It is sorted in alphabetical order by default but can be sorted by 'last-commit'.

Screenshot: Browse View



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: ... Apply help

Sub Directories

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

Files

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

Searching the Repository

This page last changed on Oct 05, 2007 by mquail.

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.

You can search for:

- Authors
- Branch names
- Commit comments
- Changeset ids
- Filenames/paths - Antglobs can be used

Simple Search

To access the simple search screen, click the 'Search' tab in the top right-hand corner of the FishEye screens.

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 - non-binary files, less than 5mb and on the trunk/head
- File names/paths - Antglobs can be used
- Authors
- Branch names
- Tag names
- Revision dates before and after

Results can be grouped by:

- Changeset
- Revision
- File
- Directory

The results are shown in a standard HTML view. You can choose to show the results in a tabular format by using the '*Tabular/CSV"' checkboxes, and you can save the results to a CSV file.

The following fields are shown:

- Path
- Revision
- Author
- Date
- Comment
- Changeset
- Total lines
- Total lines added
- Total lines removed
- Tags

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.

To do an advanced search, click the 'Switch to Advanced Search' link found on the top left-hand side 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.

Viewing a File History

This page last changed on Oct 04, 2007 by smaddox.

To view a file history, click the file name in the 'Files' pane on the Browse screen. The 'File History View' shows the different revisions of a file.

Branch History

The 'History' page at top right shows a diagram of the branches and revisions of the selected file. Click on the diagram to focus the window to that branch/revision.

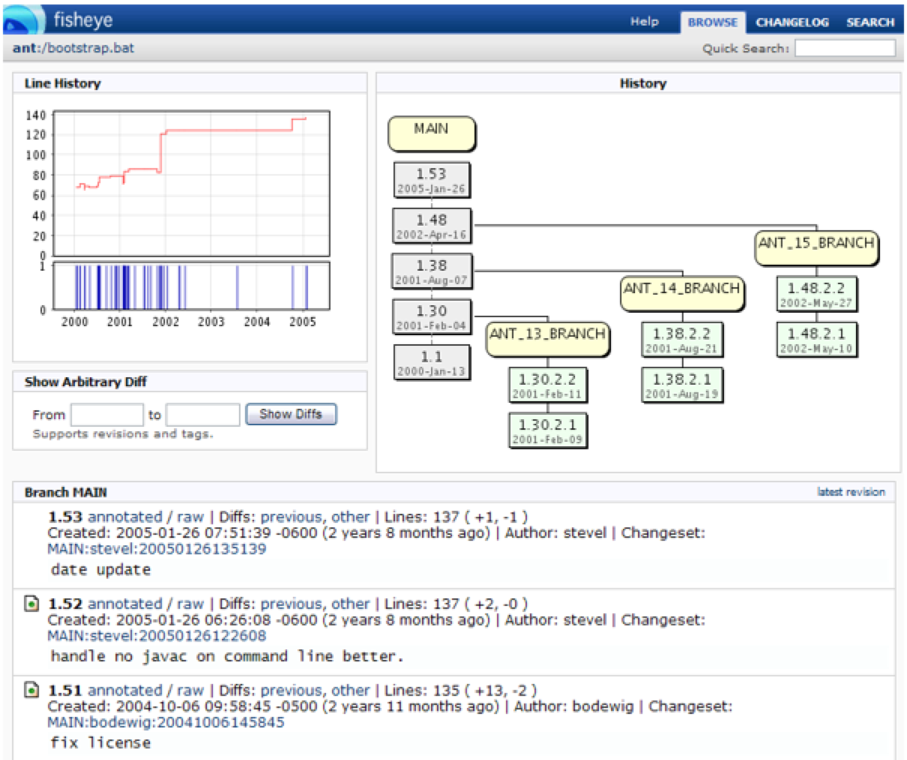
Line History Graph

The 'Line History' tab at top left contains a graph showing the total line-count over time for the selected file. If you have a branch-constraint specified, then the line-count history of that branch is also shown.

Arbitrary Diffs

The 'Show Arbitrary Diff' pane allows you to request a diff between any two revisions of the selected file. You can use revision numbers or tag names.

Screenshot: File History



1.53 annotated / raw | Diffs: previous, other | Lines: 137 (+1, -1)
Created: 2005-01-26 07:51:39 -0600 (2 years 8 months ago) | Author: stevel | Changeset:
MAIN:stevel:20050126135139

```
date update
```

1.52 annotated / raw | Diffs: previous, other | Lines: 137 (+2, -0)
Created: 2005-01-26 06:26:08 -0600 (2 years 8 months ago) | Author: stevel | Changeset:
MAIN:stevel:20050126122608

```
handle no javac on command line better.
```

1.51 annotated / raw | Diffs: previous, other | Lines: 135 (+13, -2)
Created: 2004-10-06 09:58:45 -0500 (2 years 11 months ago) | Author: bodewig | Changeset:
MAIN:bodewig:20041006145845

```
fix license
```

Viewing the Changelog

This page last changed on Oct 04, 2007 by smaddox.

The 'Changelog' tab allows you to browse the changes made to your repository chronologically. It provides a calendar in the left-hand column to allow you to navigate to any time in the history of your repository. You can also drill down into a subdirectory using the directory tree in the left-hand column.

The changesets are shown in the right-hand column. They are ordered with the most recent first.

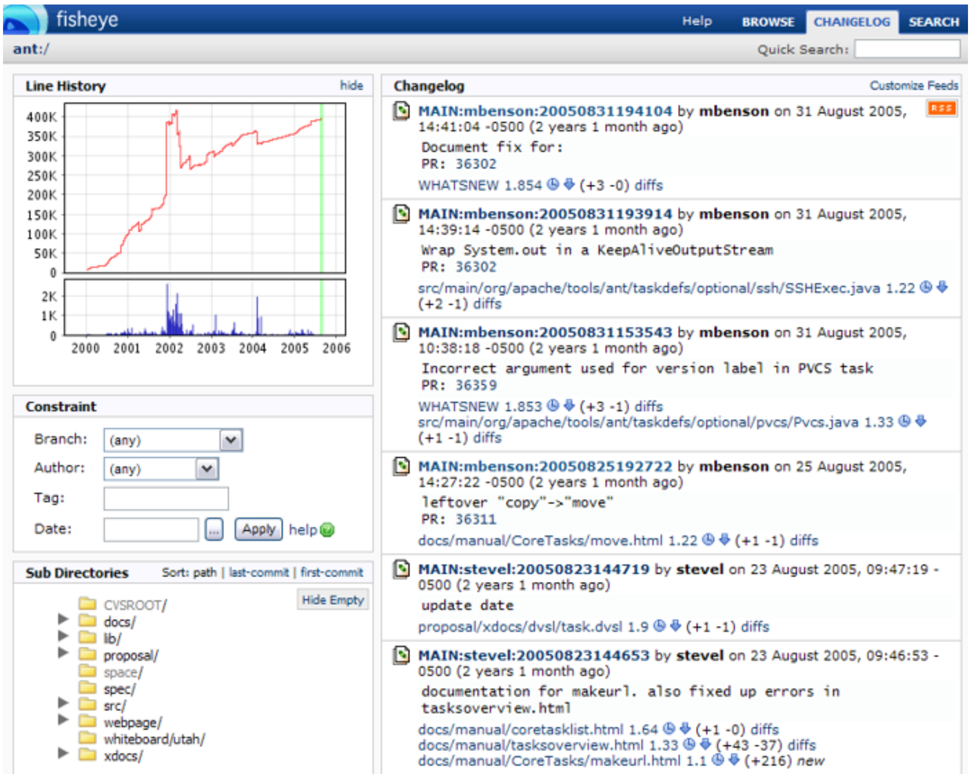
You can move forward/backward in time using the 'earlier/later' controls at the top and bottom of the right-hand column. A timeline on the calendar highlights the range of changesets displayed on this page. Click on the calendar to navigate to the changesets relevant to a specific period of time.

Constraint Mode

You can specify a 'Constraint' to control the information that is shown in the Changelog.

Branch	Will show changesets on only that branch.
Author	Shows only changesets checked in by the given author.
Tag	Shows only changesets that contain revisions tagged with the given tag.
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: Viewing the Changelog



Branch: (any)

Author: (any)

MAIN:mbenson:20050831194104 by **mbenson** on 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** on 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** on 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** on 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

MAIN:stevel:20050823144719 by **stevel** on 23 August 2005, 09:47:19 -0500 (2 years 1 month ago)

```
update date
```

proposal/xdocs/dvsl/task.dvsl 1.9 (+1 -1) diffs

MAIN:stevel:20050823144653 by **stevel** on 23 August 2005, 09:46:53 -0500 (2 years 1 month ago)

```
documentation for makeurl. also fixed up errors in
tasksoverview.html
```

docs/manual/coretasklist.html 1.64 (+1 -0) diffs
docs/manual/tasksoverview.html 1.33 (+43 -37) diffs
docs/manual/CoreTasks/makeurl.html 1.1 (+216) *new*

3. Changing your User Profile

This page last changed on Oct 04, 2007 by smaddox.

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

To view your own user profile, log into FishEye, choose a repository and click the 'Profile' link at the top right-hand corner of the page.

⚠ Always click 'Save' after making any changes.

Below is a description of each tab and its contents.

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.
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. Implemented in 1.3.1.
Line wrapping	Default is 'None' i.e. long lines will never word-wrap. 'Soft' is when long lines will word-wrap. Implemented in 1.3.1.

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. Implemented in 1.3.1.

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.

Change Password Tab

Option to be able to change your password if required.

> The passwords are case sensitive.

Author Mapping Tab

This functionality is currently not used in FishEye.

Watches Tab

> Add a 'watch' on the Browse 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 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.

4. EyeQL Reference Guide

This page last changed on Oct 05, 2007 by mquail.

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

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)? (whereclauses)? (order by date)? (group by (file|dir|directory|changeset))? (returnreturn-clauses)?
clauses:	clause ((or|and|,) clause)* Notes: and binds more tightly than or. ',' (comma) means 'and'.
clause:	(clauses) notclause path (not)? like word Notes: word is an Antglob. date in ((|[) dateExp, dateExp ()|]) Notes: The edges are inclusive if [or] is used. exclusive if (or) is used. datedateopdateExp Notes: dateop can be <, >, <=, >=, =, == or != . author =word author in (word-list) comment matchesword 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 matchesword Notes: Does a full-text search. At this time searches are restricted to HEAD revisions. (modified|added|deleted)? on branchword 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. taggedop? 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 tagstag-range after tagword before tagword is head (onword)? 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.
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)`.
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) (asword)? The attribute to return, optionally followed by a name to use for the column.

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
```

5. Antglob Reference Guide

This page last changed on Oct 04, 2007 by smaddox.

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/`

6. Date Expressions Reference Guide

This page last changed on Sep 12, 2007 by smaddox.

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