

# Jetty - Eclipse Project Creation Review

## Introduction

The Jetty Project is a proposed open source project under the Eclipse Runtime Project.

## Creation Review

- The projection creation review has been scheduled for **March 11, 2009**
- The communication channel for the review is the [newsgroup](#). Use the newsgroup to post any comments or to request the review to be held on a conference call.

## Background

Jetty is a Java-based web server and implementation of the Servlet Specification and is currently hosted at The Codehaus under the Apache 2.0 license. It was initially implemented in 1995 as MBServlet, and has gone through 6 major releases since then. It is now one of the premier implementations of the Servlet standard as found by the monthly survey found at [Netcraft.com](#). While that survey would not count Jetty instances behind firewalls, Apache HTTPD facades, load balancers, or instances with identifiers turned off, it still finds nearly a quarter million internet domains served directly by versions of the Jetty server. Since the version 3.3 release of the Eclipse IDE, Jetty has been employed to serve its help system. It is a mature project and technology, with numerous committers, both to its core, and to its add-on efforts. Furthermore, numerous commercial products and other open source projects use Jetty as an integral component. [The Jetty Powered Page](#) shows many examples of commercial use.

## Scope

The scope of the Jetty project is an extensible framework for, and exemplary implementation of, an embeddable HTTP server and servlet container.

## Objectives

The key objectives of the Jetty project are to:

- Maintain an embeddable release of a HTTP server and servlet container.
- Participate in the formation and implementation of future versions of relevant specifications.
- Foster better integration with other open source projects, such as Eclipse Equinox, and various tooling efforts.

- Provide a platform for other projects to create highly scalable solutions in their particular domains.

## Initial Contributions

The initial code will be the current jetty trunk as located in svn at The Codehaus. A bundle has already been delivered to begin the IP verification process.

## Description

Jetty is an open-source, standards-based, full-featured web server implemented entirely in Java. It is released under the Apache 2.0 license and is therefore free for commercial use and distribution. Once established as an Eclipse project, it will be dual-licensed, maintaining its Apache 2.0 license, and adding the Eclipse Public License. Beyond the license addition, which would not require any changes from the current large number of users or consuming open source projects or commercial products with use of Jetty, the move is seen as having numerous benefits for the projects and community...

- The transition would be a great time to clean up the project's packaging as it moves to an org.eclipse name.
- The project will be focused on providing components and less on providing a bundled application server stack. This will allow great focus of development energies and further bring out Jetty's embedding strengths.
- The Eclipse IP policies will supplement Jetty's existing processes.
- Jetty could discuss handling the maintenance and updating the OSGI HTTP service for Equinox as a module. The project is also interested in being involved in an effort to modernize the standard for OSGI HTTP service. At this time, the project does not envision revising its internal structure to use OSGi, but to stay with POJO that can be configured with IOC/DI frameworks, which will work well for OSGi.
- The being a part of Eclipse's release trains would simplify spreading current versions of Jetty for use in relevant projects within the Foundation that may be interested.

As history, Jetty was first created in 1995, and has benefited from input from a vast user community and consistent and focused development by a stable core of lead developers. There are many examples of Jetty in action on the [Jetty Powered Page](#) however, as Jetty aims to be as unobtrusive as possible, countless websites and products are based around Jetty, but Jetty is invisible!

Jetty can be used as:

- a stand-alone traditional web server for static and dynamic content
- a dynamic content server behind a dedicated HTTP server such as Apache using mod\_proxy or mod\_jk
- an embedded component within a java application
- as a component to build an application server or Java EE server

This flexibility means that Jetty can be encountered in a number of different contexts:

- shipped with products to provide out-of-the-box usability eg Tapestry, Grails, JRuby
- distributed on CDs with books to make examples and exercises ready-to-run

- incorporated into applications as a HTTP transport eg JXTA, Cisco SESM
- integrated as a web container in JavaEE app servers eg Jonas, Geronimo, JBoss, Sybase
- included as a component of an application eg Eclipse IDE, Hadoop

Some of the defining features of Jetty are:

- **Simplicity**
  - Jetty is built by an assembly of simple components.
  - Where possible features are added by aggregation of components rather than creation of complex deep/optional APIs
  - Assembly and configuration can be done by the java API, the jetty XML configuration file, which is an IOC style mapping of XML to POJO APIs, other IOC/component frameworks such as Spring and Plexus, OSGi activators, and other XML to POJO mappings such as XBeans
- **Embeddability**
  - Jetty is designed to be a good component. This means that it can easily be embedded in an application without forcing the application to adapt to it's
    - Configuration files or formats
    - File system layout
    - Classloading hierarchy
    - Usage of dependency injection and inversion of control patterns
- **Pluggability**
  - Jetty is architected for pluggability. The API allows different implementations of all of the principal Jetty components to be selected. At least one, but sometimes more, implementations of a component are always provided. However if these do not meet your needs, you are free to code your own using the interfaces and abstract classes as a basis. This means that Jetty can be easily customized to a particular application environment. This is particularly useful when Jetty is acting as the web container in a JavaEE server, as Jetty's pluggability ensures a tight integration with a host container.

## Organization

### Initial Committers

Jetty has a long history, and a diverse base of existing committers. There are a number of CLA's on file for emeritus committers to jetty that may or may not become active and rejoin the jetty development efforts at eclipse and will be managed on a case by case basis though normal eclipse channels.

The initial committers at Eclipse are proposed to be:

- Greg Wilkins (Webtide): Project Lead
  - project despot since inception, primary code developer and creator of jetty, committer for many other open source projects like dojo, cometd, active mq
- Jan Bartel (Webtide)

- second largest code contributor to jetty , creator of the jetty maven plugin, jaas support, other integrations, committer for many other open source projects
- Athena Yao (Webtide)
  - jetty developer, has worked extensively with message reliability for cometd and jetty and issue resolution
- Jesse McConnell (Webtide)
  - jetty developer, ldap jaas support, build and release work, ssl async performance, jetty async client, maven integration and issue resolution, committer for many other open source projects
- David Yu (Webtide)
  - jetty developer, integrations with third party software, performance enhancements, cometd integration contributions, issue resolution
- Simone Bordet (Webtide)
  - jetty developer, integrations with terracotta and apache activemq, message reliability for cometd, issue resolution, committer for many other open source projects
- David Jencks (IBM)
  - jetty developer, jaspi spec implementation for jetty, security refactoring for jetty trunk, geronimo spec lead and major contributor to geronimo

## Code Contribution

The code being contributed is currently located in svn at The Codehaus under the Jetty trunk. The source has been prepared to be pulled over into eclipse svn minus one directory labeled codehaus-modules-trunk and that code will remain the The Codehaus. The IP review process has already started and initial meetings have already happened.

- <http://svn.codehaus.org/jetty/jetty/trunk>

## Interested Parties

The following projects have expressed interest or are already working with Jetty in its current form.

- nexB: Philippe Ombredanne
- IBM: Simon Kaegi
- Sonatype: Jason van Zyl
- many many more

## Mentors

- Jeff McAffer, EclipseSource
- Bjorn Freeman-Benson, Eclipse Foundation

## Developer Community

There is a base of active core committers, as well as people who contribute to ancillary modules and add-ons. Furthermore numerous systems management, development, and other tools currently support Jetty. Our goal is to continue to work with everyone currently involved, and with greater Eclipse Foundation involvement, expand further.

## User Community

Jetty has a large and active user community that interacts with developers directly through email, through the usage of developer and user mailing lists at The Codehaus, and in person at various conferences.

## Tentative Plan

- March 11, Project Creation Review
- March 12, source imported into Eclipse SVN
- March 23-26, 2009. EclipseCon
- April 1, IP validation process complete
- April 15, Initial Milestone Releases
- June 2, 2009. JavaOne debut of Jetty v7

## Terminology

"Jetty" is: breakwater: a protective structure of stone or concrete; extends from shore into the water to prevent a beach from washing away ([source](#))