Version Control With Subversion

Apache Subversion

Apache Subversion (often abbreviated SVN, after its command name svn) is a version control system. The software is distributed as open source software

Apache Subversion (often abbreviated SVN, after its command name svn) is a version control system. The software is distributed as open source software, under the Apache License. Software developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation. Subversion is generally used to aid in the software lifecycle process, including code commits, binary releases, fault reporting and retraction, patch releases, and developer updates (including documentation). Subversion is an important tool in software development lifecycle tasks such as software versioning. It provides an

easy command line or application-based way to commit, branch, revert, and release all sorts of files (not only source code), though it is most used by the software development community. Unlike Git, which it predates, SVN repositories are more often maintained on privately maintained servers. Subversion's goal is to be a mostly compatible successor to the previously more widely used Concurrent Versions System (CVS).

The open source community has used Subversion widely: for example, in projects such as Apache Software Foundation, FreeBSD, SourceForge, and...

Repository (version control)

Depending on whether the version control system in use is distributed, like Git or Mercurial, or centralized, like Subversion, CVS, or Perforce, the whole

In version control systems, a repository is a data structure that stores either metadata or metadata and files for a set of files or directory structure. Depending on whether the version control system in use is distributed, like Git or Mercurial, or centralized, like Subversion, CVS, or Perforce, the whole set of information in the

repository may be duplicated on every user's system or may be maintained on a single server. Some of the metadata that a repository contains includes, among other things, a historical record of changes in the repository, a set of commit objects, and a set of references to commit objects, called heads.

The main purpose of a repository is to store a set of files, as well as the history of changes made to those files. Exactly how each version control system handles storing those changes, however, differs greatly. For instance, Subversion in the past relied on a database instance but has since moved to storing its changes directly on the filesystem. These differences in storage techniques have generally led to diverse uses of version control by different groups, depending on their needs.

== Overview ==

Version control

a dominant successor was Subversion, followed by the rise of distributed version control tools such as Git. Revision control manages changes to a set

Version Control With Subversion

Version control (also known as revision control, source control, and source code management) is the software engineering practice of controlling, organizing, and tracking different versions in history of computer files – primarily source code text files, but generally any type of file.

Version control is a component of software configuration management.

A version control system is a software tool that automates version control. Alternatively, version control is embedded as a feature of some systems such as word processors, spreadsheets, collaborative web docs, and content management systems, such as Wikipedia's page history.

Version control includes options to view old versions and to revert a file to a previous version.

== Overview ==

As teams develop software, it is common to deploy multiple versions of the same software, and for different developers to work on one or more different versions

simultaneously. Bugs or features of the software are often only present in certain versions (because of the fixing of some problems and the introduction of others as the program develops). Therefore, for the purposes of locating and fixing bugs, it is vitally important to be able to retrieve...

Concurrent Versions System

Versions System

News: Stable CVS Version 1.11.23 Released! [Savannah]". savannah.nongnu.org. Ben Collins-Sussman, Version Control with Subversion For - Concurrent Versions System (CVS, or Concurrent Versioning System) is a version control system originally developed by Dick Grune in July 1986. It builds on top of an older version control system called Revision Control System (RCS), adding support for repository-level change tracking, and a client-server model. CVS servers normally run on Unix systems, while clients can be on any operating system.

== Design ==

CVS operates as a front end to Revision Control System (RCS), an older version control system that manages individual files but not whole projects. It expands upon RCS by adding support for repository-level change tracking, and a client-server model. Files are tracked using the same history format as in RCS, with a hidden directory containing a corresponding history file for each file in the repository.

CVS uses delta compression for efficient storage of different versions of the same file. This works well with large text files with few changes from one version to the next. This is usually the case for source code files. On the other hand, when CVS is told to store a file as binary, it will keep each individual version on the server. This is typically used for non-text files such...

Comparison of version-control software

definition were to be loosened enough to encompass Subversion's approach as every version control system supports branching and would therefore support

The following tables describe attributes of notable version control and software configuration management (SCM) systems that can be used to compare and contrast the various systems.

For SCM software not suitable for source code, see Comparison of open-source configuration management software.

== General information ==

The following table contains relatively general attributes of version-control software systems, including:

List of software that uses Subversion

software that uses Subversion, a revision control system used in software development. SubversionEdge, a web-based front-end for Subversion. TeamForge, distributed

The following is a list of software that uses Subversion, a revision control system

used in software development.

SubversionEdge, a web-based front-end for Subversion.

TeamForge, distributed agile application lifecycle management software.

TortoiseSVN, an extension for Microsoft Explorer.

SnailSVN, a Mac OS X GUI client with Finder integration.

VisualSVN Server, a commercial and proprietary Subversion server package for Windows operating system.

Mindquarry, open-source collaborative software geared towards small and medium-sized workgroups.

Polarion ALM for Subversion - web based application lifecycle management platform.

RapidSVN, a visual subversion client.

SharpForge, a project management and bug-tracking web application.

SVK, a distributed revision control system.

svnX, an open-source GUI client for Mac OS X.

Versions, a Mac OS X GUI client.

Cornerstone, a Mac OS X GUI client.

RabbitVCS, an extension for GNOME's Nautilus file manager and gedit text editor.

Agilo for Trac, a web-based Scrum tool.

SVN Repo Browser Pro, an iPhone and iPad client.

BugBranch, an iPhone and iPad client.

Subdivision, a commercial GUI tool for managing Subversion repositories.

Subversion (disambiguation)

refer to: Subversion and containment, a concept in literary studies Apache Subversion, a software versioning and revision control system "Subversion" (song)

Subversion is an attempt to overthrow structures of authority.

Subversion may also refer to:

Subversion and containment, a concept in literary studies

Apache Subversion, a software versioning and revision control system

"Subversion" (song), a 1996 gothic metal song

Subversion (video game), a postponed computer game from Introversion Software

Version Control With Subversion

Subversion (film), an upcoming action film starring Chris Hemsworth

List of version-control software

formerly Borland; centralized control of digital assets and activities Subversion (SVN) [open, client-server] – versioning control system inspired by CVS Surround

This is a list of notable version control software systems.

== Common attributes ==

Openness, whether the software is open source or proprietary

Repository model, how working and shared source code is handled

Shared, all developers use the same file system

Client–server, users access a master repository server via a client; typically, a client machine holds only a working copy of a project tree; changes in one working

copy are committed to the master repository before becoming available to other users

Distributed, repositories act as peers; typically each user has a local repository clone with complete version history in addition to their working files

Versioning file system

August 2011. Retrieved 28 April 2012. Version Control with Subversion: Next Generation Open Source Version Control "About Apple File System". Apple Developer

A versioning file system is any computer file system which allows a computer file to exist in several versions at the same time. Thus it is a form of revision control. Most common versioning file systems keep a number of old copies of the file. Some limit the number of changes per minute or per hour to avoid storing large numbers of trivial changes. Others instead take periodic snapshots whose contents can be accessed using methods similar as those for normal file access.

== Similar technologies ==

Mercurial

taken steps to ease the transition for users of other version control systems, particularly Subversion. Mercurial is primarily a command-line driven program

Mercurial is a distributed revision control tool for software developers. It is supported on Microsoft Windows, Linux, and other Unix-like systems, such as FreeBSD and macOS.

Mercurial's major design goals include high performance and scalability, decentralization, fully distributed collaborative development, robust handling of both plain text and binary files, and advanced branching and merging capabilities, while remaining conceptually simple. It includes an integrated web-interface. Mercurial has also taken steps to ease the transition for users of other version control systems, particularly Subversion. Mercurial is primarily a command-line driven program, but graphical user interface extensions are available, e.g. TortoiseHg, and several IDEs offer support for version control with Mercurial. All of

Mercurial's operations are invoked as arguments to its driver program hg (a reference to Hg – the chemical symbol of the element mercury).

Olivia Mackall originated Mercurial and served as its lead developer until late 2016. Mercurial is released as free software under the GPL-2.0-or-later license. It is mainly implemented using the Python programming language, but includes a binary diff...