V Skills

Skills for a secure future

Certified CVS Professional
Sample Material

V-Skills Certifications

A Government of India
&
Government of NCT Delhi Initiative

V-Skills

Skills for a secure future

1. VERSIONING BASICS

CVS is a version tracking system. It maintains records of files throughout their development, allows retrieval of any stored version of a file, and supports production of multiple versions. CVS enables multiple developers to work simultaneously on a single file without loss of data. Each developer works on her own copy of a file, and all changes are later merged into a single master copy. CVS can be integrated with bug-tracking and feature-tracking systems, and it provides features that can assist a project manager by tracking changes to a project over time.

CVS can be used in many environments for many purposes. It is used for maintaining configuration files, mail aliases, source code, FAQ files, art, music, articles, essays, and books. Some system administrators keep everything in the /etc directory under CVS in order to track system configuration changes over time. CVS is also used to store and automatically publish content to web sites and FTP servers.

CVS follows the Unix ethos of small programs doing what they do well. The RCS (Revision Control System) program handles revision control of single files, so CVS uses RCS to store file data. CVS adds features to RCS, most notably the abilities to work over collections of files, and to work out of a repository that may be local or remote.

1.1. What Is a Versioning System?

Version control is the process of recording and retrieving changes in a project.[1] A version control system can enable you to retrieve an old version to fix bugs or update features, branch development to allow the project to progress along multiple tracks simultaneously, and generate reports that show the changes between any two arbitrary stages of a project.

Most version control systems store notes with each change, and many provide tools that allow a project leader to analyze the changes. Most also include the ability to retrieve differences between arbitrary versions of files, which makes it easier to create patches or locate bugs.

The benefits of a version control system such as CVS include:

- ✓ Any stored revision of a file can be retrieved to be viewed or changed.
- ✓ The differences between any two revisions can be displayed.
- ✓ Patches can be created automatically.
- ✓ Multiple developers can work simultaneously on the same project or file without loss of data.
- ✓ The project can be branched to allow simultaneous development along varied tracks. These branches can be merged back into the main line of development.
- ✓ Distributed development is supported across large or small networks. (CVS offers a variety of authentication mechanisms.)

Using version control for a project requires some extra work on an ongoing basis. In addition, previous versions of files, or records of changes to the various files in a project, occupy disk space that you might otherwise use for something else. However, the features

that a good version control system makes available are well worth the investment of time and disk space. For example, without version control, project backups typically are timestamped copies of an entire project, hopefully stored together in a logical fashion. Version control provides organized storage and retrieval of the complete record of project changes, rather than whichever copies someone gets a chance to make.

Version control systems store files as the files are created and updated, in a way that allows any saved version of a file, or related versions of a set of files, to be retrieved at any given time. Many version control systems, including CVS, encourage a project's files to be stored together, which means that backups are easy to produce.

The ability to recover any earlier version of a given file allows you to roll back to support feature requests for previous releases and is critical when you create a bugfix branch of the project. CVS (and some other version control systems) allows simultaneous storage of the bugfix branch of a project and its main trunk code.

Many version control systems, including CVS, can display the differences between versions in a computer-readable format. The format CVS uses produces a file that allows the Unix patch program to automatically convert one version of a file (or set of files) to another.
Version control often allows multiple projects to use the same files, which helps divide a larger project among smaller teams. You can give each team a version-controlled copy of the section they're working on, as well as the latest stable version of the files they need to use to test their section.

Sometimes, two or more developers may make changes to the same file. Those changes may be in different parts of the file, and they may be made in such a way as not to conflict with each other. Other times, two or more developers may make conflicting changes to the same portion of a file. In such a case, CVS does its best to merge conflicting changes, but it only knows which lines of a file have been changed. It doesn't know what the changes mean. CVS provides tools to display changes between arbitrary revisions, which can help locate and resolve problems.

Version control is not a substitute for team members communicating with each other. File updates should be passed through CVS, but the meaning of the changes must be passed to other team members by actually discussing them. If one developer needs to change the arguments to a function or the chapter numbering of a book, that must somehow be communicated to the other developers.

Versioning systems are most commonly used for programming, but they are also useful for writing (I used CVS to write this book), system administration (configuration files), and anything else that requires files that change, where you might want to retrieve older versions of those files, or for situations where several people may be working on the same files. One family I know uses CVS to store their shopping list, to keep them from overwriting each other's entries.

1.2. CVS in the Field

CVS records file changes during a project's development. Project files are added to the repository as they are created, and developers check out a personal sandbox — a personal copy of the project's files — to work from. Each developer works in her own sandbox and regularly commits her changes to the repository. Developers also update the contents of their sandboxes regularly to ensure that changes to the repository are reflected in each sandbox.

The term project can take on many different meanings. The stereotypical CVS project is a programming project in which files contain source code for the various programs written as part of the project. But that's a narrow view of what a CVS project can be. CVS can be used in many other settings as well, as the next few sections demonstrate.

System Administration

CVS can store configuration files, mail aliases, domain records, and other files for which changes should be tracked. Import the files (or all of /etc) into a repository and require administrators to check them out into a sandbox to make changes. Commit the files back to the repository and export the changes to the server. If the changes fail, rolling back to the previous state is easy.

Multiple servers with varied but similar configurations can be maintained using different branches of the same files. Changes to any given branch can be merged into other branches selectively.

Every change made through CVS is recorded in a file history, along with the username of the person making the change, the date the change was made, and any notes recorded with the change. All this information can help, for example, when trying to spot which change to which configuration file broke the mail server.

Both the CVS server and the client run on all Unix and Linux operating systems. Third-party graphical clients are available for Unix, Linux, Windows, and Macintosh systems, and for the Java runtime environment. The CVSNT CVS server is available for Windows NT or later. This makes CVS particularly useful for cross-platform environments.

Software Development

Program development is perhaps the most common use for version control systems. After the initial release of a program, two versions usually need to be maintained: the new version that will eventually be the next release and the bugfix version for the current release. CVS allows you to split the development into two or more parts, called a trunk and a branch. Typically, the branch is used for bug fixes, while the main trunk is used for new feature development. Both versions of the program, the bugfix branch and the main trunk, are stored in the same repository. This allows the changes from the bugfix branch to ultimately be merged into the main trunk, ensuring that all bugfixes get rolled into the next release of the program.

A CVS repository can be hosted on the machine that most developers will be using, or you can host the repository on a machine that developers access via a local or wide-area network. A repository can be accessed simultaneously by multiple computers. If you need to authenticate your CVS users, there are a variety of authentication mechanisms available.

When multiple developers are trying to work on the same project, it's likely that two or more developers will eventually want to work with the same file at the same time. Without version control, this would lead to problems, as developers would soon find themselves overwriting one another's changes. One way that some version control systems prevent such conflicts is to enable developers to lock whatever files they are working on, so that no one else can make changes at the same time.

Instead of locking files to prevent conflicts, CVS simply allows multiple developers to work on the same file. CVS's file-merging feature then allows you to merge all changes into one file. File merging aids development across remote time zones, as developers can work on different sections of the same file, regardless of the lock status. In fact, there is no lock status, because there is no locking. With a file-locking system, a developer may have to email someone and then wait until that person wakes up, reads his email, and unlocks the needed file. The CVS approach prevents one developer from blocking another, thus increasing productivity. However, if your project team needs file locking, you can use the cvs watch command to emulate it.

The cvs diff command displays the differences between any two revisions of a file (or set of files) in the repository. A variation of the command creates a Unix standard patch file to update one revision to another, which is useful when sending patches or updates to customers.

CVS can be configured to record commit messages in bug-tracking systems. Chapter 7 explains how to use the administrative files to provide message templates and run scripts automatically during commits. This does not necessarily record the stage of each change (completed, tested, etc.). Unless you rigorously enforce a requirement to write meaningful commit messages, you should maintain a separate change log.

Store your build and installation scripts in CVS to maintain a record of changes and to ensure that such scripts are kept with the project files. Releases should always be built from a freshly checked-out sandbox and tagged with a human-friendly name.

CVS does not include build or installation tools, though cvs export should be part of the installation process for your project. I like to use make for build and installation scripts.

Content-Controlled Publishing

Many people use CVS to maintain web sites and other file servers, and they use scripts to automatically publish updates to those servers. Some people use scripts to distribute and apply patch files on remote machines, saving bandwidth by distributing only the changes. A variety of such scripts are available at http://www.cvshome.org, and some are discussed in

Other Uses for CVS

CVS is also useful for managing any other type of file, from book chapters to blueprints, music to art, mailing lists to shopping lists. The features that make it useful to programmers are also useful to anyone who produces something that can be stored as a computer file.

Computer scientists define version control, source control, and change management as different but overlapping tasks. The term version control is the best fit for the aspects of the field that apply to CVS.