CORE.GOV Versioning Software: Subversion and CVS

With CollabNet Enterprise Edition 3.0, Project Owners on CORE.GOV will now be choosing which versioning software to use—Subversion or CVS. Subversion is a new file control and versioning system with a similar architecture to the more established CVS (i.e., client/server, with a centralized file repository and that supports multiple project branches and versions). The goal of this brief document is to help Project Owners decide which versioning software to use.

The chart below compares and contrasts the two programs. A link at the bottom of every CORE.GOV page—"Developer Tools"—takes you to the latest versions of clients for both Subversion and CVS.

Feature	Subversion (SVN)	Concurrent Version Systems (CVS)
	Newest open-source version control system. Integrated into CollabNet Enterprise Edition 3.0; each standard development project gets a SVN repository. Good for *ad hoc* examination of repository. Performs operations via command line interface. Intended to replace CVS; easy adoption by CVS users; easy to use for the simple case (intuitive). Existing CVS repositories can be converted to Subversion by CollabNet support.	CVS is very popular; easy to use for the simple case (intuitive).
Firewalls	Subversion uses https as a transport. This means that any firewall with port 443 open for https traffic, which is the vast majority, can use SVN without administrator intervention.	To use CVS, an additional port must be open in the firewall.
Network	Subversion was built for a network (distributed development). Subversion has demonstrated capability over WANs. Subversion has delta transfers (which update only the changes, minimizing the bandwidth used) and offline operations.	CVS was not originally built for a network. CVS does not have bi-directional delta transfers or offline operations.

Feature	Subversion (SVN)	Concurrent Version Systems (CVS)
Directories	Subversion has directory versioning, which allows users to delete, copy, and rename directories.	CVS does not have directory versioning, only file versioning.
Binary files	Subversion's binary file handling ("binary differencing") tracks changes made to non-text files such as graphics and compiled executables.	CVS does not have binary differencing and requires the entire binary file to be stored for each revision.
Metadata	Subversion attaches "properties" (arbitrary name/value pairs, such as "Due Date"/"Aug. 31" or "Importance"/"High") to files, directories, or revisions.	CVS allows check-in of comments and tags only.
Branching and tagging	With Subversion, branching (to create new project branches) and tagging (to create new versions of a branch) are constant-time operations. Version diffs and switching branches are also constant-time operations.	The time required for branching and tagging grows as the number of files involved increases. Version diffs and switching branches require more time as the number of files and the number of revisions since the branch point increase.
Atomic commits	Subversion has atomic commits (file commits are rejected if errors occur in a commit of any number of files or directories, preventing versions from getting out of sync if projects are being committed by multiple users simultaneously).	CVS provides atomic commits for individual files only, not for entire multi-file commits.
Clients	Under development for Subversion: TortoiseSVN for Windows Explorer AnkhSVN, PushOK for Microsoft Visual Studio .NET Windows IDE integrations using SCC Java clients (IDEA integrations) Eclipse clients	Currently available for CVS: dozens of clients for all major and many minor platforms, including Java and Eclipse clients and TortoiseCVS.
Graphic User Interfaces (GUIs)	Under development for Subversion: Windows, Mac, Unix.	Currently available and stable for CVS: Windows, Mac, Unix.
Space	Subversion stores two copies of the repository on the client machine, requiring significant disk space.	CVS stores only those files that are checked out onto the client machine.
Edit/Watch	Not currently available. A reserved checkout/locking feature is planned for Subversion 1.2.	CVS provides a watch feature that automatically notifies users when files they have selected to watch are edited.
Renaming/ Moving	Files in Subversion repositories can be renamed or moved without affecting their version history.	In CVS, renaming or moving a file causes its version history to be lost.
Back-end	Subversion uses BerkeleyDB database back-end. Failed transactions can cause the repository to be unusable until reset in some circumstances.	CVS uses RCS flat-file back-end. There is no database to become "wedged".