

# Revision Control Systems Compared

## Comparison charts

There is an in-depth [Version Control Systems Comparison](#) available that covers many important features you might be looking for.

## Popular Systems and their Pros/Cons

This list is by no means complete. For more in-depth information, refer to the link above.

### Visual SourceSafe

SourceSafe is Microsoft's own source control tool. If you do all of your development in Visual Studio, this is the easiest (but not necessarily best) tool for you.

Pros:

- Excellent integration with Microsoft Visual Studio (Ease of Deployment)
- Many microsoft developers already have access to it
- Relatively easy to use

Cons:

- Too simple for many users
- Commits are not atomic
- No support for changesets

- Not portable
- Proprietary and non-free

# CVS

[CVS](#) is a very widely-used system. You should only use CVS if your environment has limitations that keep you from using SubVersion.

Pros:

- Portable
- Excellent tool support
- Open-source / free
- More features than SourceSafe

Cons:

- Dated – lacking many modern features
- No atomic commits

# Subversion

[Subversion](#) is a newer open-source tool designed to replace CVS. It is superior to CVS in almost every way and is the best all-around open-source tool of its kind. There is an article in this knowledge base with more information about Subversion.

Pros:

- All the pros of CVS plus
- Atomic commits
- High performance
- Vast array of configuration options
- Handles binary files efficiently as well

Cons:

- Can be complex to setup depending on the configuration chosen
- No good Visual Studio integration yet

# Perforce

[Perforce](#) is an easy to deploy and very feature-packed proprietary solution. It is used by Google. A free version is available, but it only supports a maximum of 2 users.

Pros:

- Excellent feature-set
- Atomic commits
- Very good IDE integration (including Visual Studio)
- High performance
- Handles binary files efficiently as well

Cons:

- Expensive

# Others

One of the tools listed above will most likely be right for you, but they are not the only tools out there. There is no single best choice for a source control system. It is important to use the tool that is best for you. BitKeeper was used to manage the Linux kernel, Git (designed and developed by Linus Torvalds) is currently used to manage the Linux kernel, and ClearCase by Rational is also popular in many large companies.

# Distributed Version Control Systems

- [http://en.wikipedia.org/wiki/Distributed\\_Version\\_Control\\_System](http://en.wikipedia.org/wiki/Distributed_Version_Control_System)
- [Why distributed version control](#)
- [Choosing a Distributed Version Control System](#)

*Thanks to Jose Hales-Garcia for his post on this on [OSXForum](#) .*

# Future Expansion

Someone with more information about Visual Studio Team System should add something about

that.

---

Revision #7

Created Tue, May 30, 2006 2:59 PM by Nazarian, Sam

Updated Wed, Mar 17, 2010 4:21 PM by Jamison, Jina