

The Subversion Command Line Client: `svn`

To use the command line client, you type `svn`, the subcommand you wish to use [\[58\]](#), and any options or targets that you wish to operate on—there is no specific order that the subcommand and the options must appear in. For example, all of the following are valid ways to use `svn status`:

```
$ svn -v status
$ svn status -v
$ svn status -v myfile
```

You can find many more examples of how to use most client commands in [Chapter 2, Basic Usage](#) and commands for managing properties in [the section called “Properties”](#).

`svn` Options

While Subversion has different options for its subcommands, all options are global—that is, each option is guaranteed to mean the same thing regardless of the subcommand you use it with. For example, `--verbose (-v)` always means “verbose output”, regardless of the subcommand you use it with.

`--auto-props`

Enables auto-props, overriding the `enable-auto-props` directive in the *config* file.

`--change (-c) ARG`

Used as a means to refer to a specific “change” (aka a revision), this option is syntactic sugar for “`-r ARG-1:ARG`”.

`--config-dir DIR`

Instructs Subversion to read configuration information from the specified directory instead of the default location (*.subversion* in the user's home directory).

`--diff-cmd CMD`

Specifies an external program to use to show differences between files. When `svn diff` is invoked without this option, it uses Subversion's internal diff engine, which provides unified diffs by default. If you want to use an external diff program, use `--diff-cmd`. You can pass options to the diff program with the `--extensions` option (more on that later in this section).

`--diff3-cmd CMD`

Specifies an external program to use to merge files.

`--dry-run`

Goes through all the motions of running a command, but makes no actual changes—either on disk or in the repository.

`--editor-cmd CMD`

Specifies an external program to use to edit a log message or a property value. See the `editor-cmd` section in [the section called “Config”](#) for ways to specify a default editor.

`--encoding ENC`

Tells Subversion that your commit message is encoded in the charset provided. The default is your operating system's native locale, and you should specify the encoding if your commit message is in any other encoding.

`--extensions (-x) ARGS`

Specifies an argument or arguments that Subversion should pass to an external diff command. This option is valid only when used with the `svn diff` or `svn merge` commands, with the `--diff-cmd` option. If you wish to pass multiple arguments, you must enclose all of them in quotes (for example, `svn diff --diff-cmd /usr/bin/diff -x "-b -E"`).

`--file (-F) FILENAME`

Uses the contents of the named file for the specified subcommand, though different subcommands do different things with this content. For example, `svn commit` uses the content as a commit log, whereas `svn propset` uses it as a property value.

`--force`

Forces a particular command or operation to run. There are some operations that Subversion will prevent you from doing in normal usage, but you can pass the force option to tell Subversion “I know what I'm doing as well as the possible repercussions of doing it, so let me at 'em”. This option is the programmatic equivalent of doing your own electrical work with the power on—if you don't know what you're doing, you're likely to get a nasty shock.

`--force-log`

Forces a suspicious parameter passed to the `--message (-m)` or `--file (-F)` options to be accepted as valid. By default, Subversion will produce an error if parameters to these options look like they might instead be targets of the subcommand. For example, if you pass a versioned file's path to the `--file (-F)` option, Subversion will assume you've made a mistake, that the path was instead intended as the target of the operation, and that you simply failed to provide some other—unversioned—file as the source of your log message. To assert your intent and override these types of errors, pass the `--force-log` option to subcommands that accept log messages.

`--help (-h or -?)`

If used with one or more subcommands, shows the built-in help text for each subcommand. If used alone, it displays the general client help text.

`--ignore-ancestry`

Tells Subversion to ignore ancestry when calculating differences (rely on path contents alone).

`--ignore-externals`

Tells Subversion to ignore external definitions and the external working copies managed by them.

`--incremental`

Prints output in a format suitable for concatenation.

`--limit NUM`

Show only the first *NUM* log messages.

`--message (-m) MESSAGE`

Indicates that you will specify a either a log message or a lock comment on the command line, following this option. For example:

```
$ svn commit -m "They don't make Sunday."
```

`--new ARG`

Uses *ARG* as the newer target (for use with `svn diff`).

`--no-auth-cache`

Prevents caching of authentication information (e.g. username and password) in the Subversion administrative directories.

`--no-auto-props`

Disables auto-props, overriding the `enable-auto-props` directive in the *config* file.

`--no-diff-added`

Prevents Subversion from printing differences for added files. The default behavior when you add a file is for `svn diff` to print the same differences that you would see if you had added the entire contents of an existing (empty) file.

`--no-diff-deleted`

Prevents Subversion from printing differences for deleted files. The default behavior when you remove a file is for `svn diff` to print the same differences that you would see if you had left the file but removed all the content.

`--no-ignore`

Shows files in the status listing that would normally be omitted since they match a pattern in the `global-ignores` configuration option or the `svn:ignore` property. See [the section called “Config”](#) and [the section called “Ignoring Unversioned Items”](#) for more information.

`--no-unlock`

Don't automatically unlock files (the default commit behavior is to unlock all files listed as part of the commit). See [the section called “Locking”](#) for more information.

--non-interactive

In the case of an authentication failure, or insufficient credentials, prevents prompting for credentials (e.g. username or password). This is useful if you're running Subversion inside of an automated script and it's more appropriate to have Subversion fail than to prompt for more information.

--non-recursive (-N)

Stops a subcommand from recursing into subdirectories. Most subcommands recurse by default, but some subcommands—usually those that have the potential to remove or undo your local modifications—do not.

--notice-ancestry

Pay attention to ancestry when calculating differences.

--old *ARG*

Uses *ARG* as the older target (for use with `svn diff`).

--password *PASS*

Indicates that you are providing your password for authentication on the command line—otherwise, if it is needed, Subversion will prompt you for it.

--quiet (-q)

Requests that the client print only essential information while performing an operation.

--recursive (-R)

Makes a subcommand recurse into subdirectories. Most subcommands recurse by default.

--relocate *FROM TO [PATH...]*

Used with the `svn switch` subcommand, changes the location of the repository that your working copy references. This is useful if the location of your repository changes and you have an existing working copy that you'd like to continue to use. See `svn switch` for an example.

--revision (-r) *REV*

Indicates that you're going to supply a revision (or range of revisions) for a particular operation. You can provide revision numbers, revision keywords or dates (in curly braces), as arguments to the revision option. If you wish to provide a range of revisions, you can provide two revisions separated by a colon. For example:

```
$ svn log -r 1729
$ svn log -r 1729:HEAD
$ svn log -r 1729:1744
$ svn log -r {2001-12-04}:{2002-02-17}
$ svn log -r 1729:{2002-02-17}
```

See [the section called “Revision Keywords”](#) for more information.

`--revprop`

Operates on a revision property instead of a property specific to a file or directory. This option requires that you also pass a revision with the `--revision (-r)` option.

`--show-updates (-u)`

Causes the client to display information about which files in your working copy are out-of-date. This doesn't actually update any of your files—it just shows you which files will be updated if you run `svn update`.

`--stop-on-copy`

Causes a Subversion subcommand which is traversing the history of a versioned resource to stop harvesting that historical information when a copy—that is, a location in history where that resource was copied from another location in the repository—is encountered.

`--strict`

Causes Subversion to use strict semantics, a notion which is rather vague unless talking about specific subcommands (namely, `svn propget`).

`--targets FILENAME`

Tells Subversion to get the list of files that you wish to operate on from the filename you provide instead of listing all the files on the command line.

`--username NAME`

Indicates that you are providing your username for authentication on the command line—otherwise, if it is needed, Subversion will prompt you for it.

`--verbose (-v)`

Requests that the client print out as much information as it can while running any subcommand. This may result in Subversion printing out additional fields, detailed information about every file, or additional information regarding its actions.

`--version`

Prints the client version info. This information not only includes the version number of the client, but also a listing of all repository access modules that the client can use to access a Subversion repository. With `--quiet (-q)` it prints only the version number in a compact form.

`--xml`

Prints output in XML format.

svn **Subcommands**

Here are the various subcommands:

[58] Yes, yes, you don't need a subcommand to use the `--version` option, but we'll get to that in just a minute.

[Prev](#)Chapter 9. Subversion Complete
Reference[Up](#)[Home](#)[Next](#)

svn add