

**NAME**

**svn propset** — Set the value of a property on files, dirs, or revisions

**SYNOPSIS**

```
svn propset (pset, ps) [-FqR] [--changelist CHG] [--cl] [--config-dir DIR]
  [--config-option OPT] [--depth DEP] [--encoding ENC] [--file]
  [--force] [--no-auth-cache] [--non-interactive]
  [--password PWD] [--quiet] [--recursive]
  [{-r | --revision} REV] [--revprop] [--targets]
  [--trust-server-cert] [--username USR]
  PROPNAME PROPVAL PATH ...

svn propset PROPNAME --revprop -r REV PROPVAL [TARGET]
```

**DESCRIPTION**

The first form changes a versioned file or directory property in a working copy.

The second form changes an unversioned property on a repository revision. *TARGET* just determines which repository to access.

The value of the property may be provided with the **--file** option instead of *PROPVAL*.

**OPTIONS**

**--changelist --cl CHG** operate only on members of changelist *CHG*

**--depth DEP** limit operation by depth *ARG*, one of:  
empty  
files  
immediates  
infinity

**--encoding ENC** treat value as being in charset encoding *ENC*

**-F --file FILL** read property value from file *FILL*

**--force** force operation to run

**-q --quiet** print nothing, or only summary information

**-R --recursive** descend recursively, same as **--depth=infinity**

**-r --revision REV** the revision argument *REV* can be one of:  
*NUMBER* revision number  
'{ DATE }' revision at start of the date  
HEAD latest in repository  
BASE base rev of item's working copy  
COMMITTED last commit at or before *BASE*  
PREV revision just before *COMMITTED*

Some commands also take an *ARG1 : ARG2* range.

**--revprop** operate on a revision property (use with *-r*)

**--targets** pass contents of file *ARG* as additional arguments

**GLOBAL OPTIONS**

**--config-dir DIR** read user configuration files from directory *DIR*

**--config-option OPT** set user configuration option *OPT* in the format:  
FILE:SECTION:OPTION=[VALUE]  
For example:  
servers:global:http-library=serf

<b>--no-auth-cache</b>	do not cache authentication tokens
<b>--non-interactive</b>	do no interactive prompting
<b>--password</b> <i>PWD</i>	specify a password <i>PWD</i>
<b>--trust-server-cert</b>	accept unknown SSL server certificates without prompting (but only with <b>--non-interactive</b> )
<b>--username</b> <i>USR</i>	specify a username <i>USR</i>

## NOTES

Subversion recognizes the following special versioned properties:

svn:ignore	A newline separated list of file glob patterns to ignore.
svn:keywords	Keywords to be expanded. Valid keywords are: 'URL', HeadURL The URL for the head version of the object. 'Author', 'LastChangedBy' The last person to modify the file. 'Date', 'LastChangedDate' The date/time the object was last modified. 'Rev', 'Revision', 'LastChangedRevision' The last revision the object changed. 'Id'      A compressed summary of the previous 4 keywords. 'Header' Similar to Id but includes the full URL.
svn:executable	make the file executable.
svn:eol-style	One of native, 'LF', 'CR', or CRLF.
svn:mime-type	The mimetype of the file. Used to determine whether to merge the file, and how to serve it from Apache. A mimetype beginning with <code>text/</code> (or an absent mimetype) is treated as text. Anything else is treated as binary.
svn:externals	A newline-separated list of module specifiers, each of which consists of a relative directory path, optional revision flags and an URL. The ordering of the three elements implements different behavior.  Subversion 1.4 and earlier support the following formats and the URLs cannot have peg revisions: foo          http://example.com/repos/zig foo/bar -r 1234 http://example.com/repos/zag  Subversion 1.5 and greater support the above formats and the following formats where the URLs may have peg revisions: http://example.com/repos/zig foo -r 1234 http://example.com/repos/zig foo/bar  Relative URLs are supported in Subversion 1.5 and greater for all above formats. They are indicated by starting the URL with one of the following strings ../      to the parent directory of the extracted external project '^/'     to the repository root '//'     to the scheme '/'      to the server root  The ambiguous format <code>relative_path relative_path</code> is interpreted as <code>relative_url relative_path</code> with peg revision support.

`svn:needs-lock` Indicates that the file should be locked before it is modified. Makes the working copy file read-only when it is not locked.

The `svn:keywords`, `svn:executable`, `svn:eol-style`, `svn:mime-type` and `svn:needs-lock` properties cannot be set on a directory. A non-recursive attempt will fail, and a recursive attempt will set the property only on the file children of the directory.