

NAME

svn propget — Print the value of a property on files, dirs, or revisions

SYNOPSIS

```
svn propget (pget, pg) [ -Rv ] [ --changelist CHG ] [ --cl ] [ --config-dir DIR ]
    [ --config-option OPT ] [ --depth DEP ] [ --no-auth-cache ]
    [ --non-interactive ] [ --password PWD ] [ --recursive ]
    [ { -r | --revision } REV ] [ --revprop ] [ --strict ]
    [ --trust-server-cert ] [ --username USR ] [ --verbose ] [ --xml ]
    PROPNAME [ PATH ... ]

svn propget PROPNAME --revprop -r REV [ TARGET ]
```

DESCRIPTION

The first form prints versioned properties. If specified, *REV* determines in which revision the target is first looked up.

The second form prints unversioned remote properties in the repository. *TARGET*, a working copy directory name, just determines which repository to access.

By default, this subcommand adds an extra newline to the end of the property values so that the output looks pretty. Also, whenever there are multiple paths involved, each property value is prefixed with the path with which it is associated. Use the **--strict** option to disable these beautifications (useful when redirecting a binary property value to a file, but available only if you supply a single *TARGET* to a non-recursive **propget** operation).

OPTIONS

--changelist	--cl <i>CHG</i>	operate only on members of changelist <i>CHG</i>
--depth <i>DEP</i>		limit operation by depth <i>ARG</i> , one of: empty files immediates infinity
-R	--recursive	descend recursively, same as --depth=infinity
-r	--revision <i>REV</i>	the revision argument <i>REV</i> can be one of: <i>NUMBER</i> revision number '{' <i>DATE</i> '}' 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 <i>ARG1</i> : <i>ARG2</i> range.
--revprop		operate on a revision property (use with -r)
--strict		use strict semantics
-v	--verbose	print extra information
--xml		output in XML

GLOBAL OPTIONS

--config-dir <i>DIR</i>	read user configuration files from directory <i>DIR</i>
--config-option <i>OPT</i>	set user configuration option <i>OPT</i> in the format: FILE:SECTION:OPTION=[VALUE]

For example:

```

servers:global:http-library=serf
--no-auth-cache      do not cache authentication tokens
--non-interactive     do no interactive prompting
--password PWD      specify a password PWD
--trust-server-cert  accept unknown SSL server certificates without prompting (but only with
                    --non-interactive)
--username USR      specify a username USR

```

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 `relative_path relative_path` is interpreted as `relative_url relative_path` 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.