NAME

sccs-get, get – retrieve a version of an SCCS file

SYNOPSIS

/usr/ccs/bin/get [**–begkmnpst**] [**–l** [**p**]] [**–a***sequence*] [**–c***date-time*] [**–G***g-file*] [**–i***sid-list*] [**–r***sid*] [**–x***sid-list*] *s.filename* . . .

DESCRIPTION

get retrieves a working copy from the SCCS history file, according to the specified options.

For each *s.filename* argument, **get** displays the SCCS delta ID (SID) and number of lines retrieved.

If a directory name is used in place of the *s.filename* argument, the **get** command applies to all **s.**files in that directory. Unreadable **s.**files produce an error; processing continues with the next file (if any). The use of '–' as the *s.filename* argument indicates that the names of files are to be read from the standard input, one **s.**file per line.

The retrieved file normally has the same filename base as the **s.**file, less the prefix, and is referred to as the **g-**file.

For each file processed, **get** responds (on the standard output) with the SID being accessed, and with the number of lines retrieved from the **s.**file.

OPTIONS

–b Create a new branch. Used with the **–e** option to indicate that the new delta should have an SID in a new branch. Instead of incrementing the level for version to be checked in, **get** indicates in the **p.**file that the delta to be checked in should either initialize a new branch and sequence (if there is no existing branch at the current level), or increment the branch component of the SID. If the **b** flag is not set in the **s.**file, this option is ignored.

–e Retrieve a version for editing. With this option, **get** places a lock on the **s.**file, so that no one else can check in changes to the version you have checked out. If the **j** flag is set in the **s.**file, the lock is advisory: **get** issues a warning message. Concurrent use of '**get –e**' for different SIDs is allowed, however, **get** will not check out a version of the file if a writable version is present in the directory. All SCCS file protections stored in the **s.**file, including the release ceiling, floor, and authorized user list, are honored by '**get –e**'.

–g Get the SCCS version ID, without retrieving the version itself. Used to verify the existence of a particular SID.

–k Suppress expansion of ID keywords. **–k** is implied by the **–e**.

–m Precede each retrieved line with the SID of the delta in which it was added to the file. The SID is separated from the line with a TAB.

–n Precede each line with the **%M%** ID keyword and a TAB. When both the **–m** and **–n** options are used, the ID keyword precedes the SID, and the line of text.

–p Write the text of the retrieved version to the standard output. All messages that normally go to the standard output are written to the standard error instead.

–s Suppress all output normally written on the standard output. However, fatal error messages (which always go to the standard error) remain unaffected.

–t Retrieve the most recently created (top) delta in a given release (for example: **–r1**).

–l [p] Retrieve a summary of the delta table (version log) and write it to a listing file, with the '**l.**' prefix (called '**l.**file'). When –**lp** is used, write the summary onto the standard output.

–a *sequence*
Retrieve the version corresponding to the indicated delta sequence number. This option is used primarily by the SCCS **comb** command (see **sccs-comb**(1)); for users, –**r** is an easier way to specify a version. –**a** supersedes –**r** when both are used.

–c*date-time*
Retrieve the latest version checked in prior to the date and time indicated by the *date-time* argument. *date-time* takes the form: *yy*[*mm*[*dd*[*hh*[*mm*[*ss*]]]]]. Units omitted from the indicated date and time default to their maximum possible values; that is –**c7502** is equivalent to –**c750228235959**. Any number of non-numeric characters may separate the various 2 digit components. If white-space characters occur, the *date-time* specification must be quoted.

–G*newname*
Use *newname* as the name of the retrieved version.

–i*sid-list*
Specify a list of deltas to include in the retrieved version. The included deltas are noted in the standard output message. *sid-list* is a comma-separated list of SIDs. To specify a range of deltas, use a '–' separator instead of a comma, between two SIDs in the list.

–r*sid* Retrieve the version corresponding to the indicated SID (delta).

The SID for a given delta is a number, in Dewey decimal format, composed of two or four fields: the *release* and *level* fields, and for branch deltas, the *branch* and *sequence* fields. For instance, if **1.2** is the SID, **1** is the release, and **2** is the level number. If **1.2.3.4** is the SID, **3** is the branch and **4** is the sequence number.

You need not specify the entire SID to retrieve a version with **get**. When you omit –**r** altogether, or when you omit both release and level, **get** normally retrieves the highest release and level. If the **d** flag is set to an SID in the **s.**file and you omit the SID, **get** retrieves the default version indicated by that flag.

When you specify a release but omit the level, **get** retrieves the highest level in that release. If that release does not exist, **get** retrieves highest level from the next-highest existing release.

Similarly with branches, if you specify a release, level and branch, **get** retrieves the highest sequence in that branch.

–x*sid-list*
Exclude the indicated deltas from the retrieved version. The excluded deltas are noted in the standard output message. *sid-list* is a comma-separated list of SIDs. To specify a range of deltas, use a '**-**' separator instead of a comma, between two SIDs in the list.

USAGE

ID Keywords

In the absence of –**e** or –**k**, **get** expands the following ID keywords by replacing them with the indicated values in the text of the retrieved source.

Keyword	*Value*
%A%	Shorthand notation for an ID line with data for **what**(1): **%Z%%Y% %M% %I%%Z%**
%B%	SID branch component
%C%	Current line number. Intended for identifying messages output by the program such as ''*this shouldn't have happened*'' type errors. It is *not* intended to be used on every line to provide sequence numbers.
%D%	Current date: *yy***/***mm***/***dd*
%E%	Date newest applied delta was created: *yy***/***mm***/***dd*
%F%	SCCS **s.**file name
%G%	Date newest applied delta was created: *mm***/***dd***/***yy*
%H%	Current date: *mm***/***dd***/***yy*
%I%	SID of the retrieved version: **%R%.%L%.%B%.%S%**
%L%	SID level component
%M%	Module name: either the value of the **m** flag in the **s.**file (see **sccs-admin**(1)), or the name of the **s.**file less the prefix
%P%	Fully qualified **s.**file name
%Q%	Value of the **q** flag in the **s.**file
%R%	SID Release component
%S%	SID Sequence component
%T%	Current time: *hh***:***mm***:***ss*
%U%	Time the newest applied delta was created: *hh***:***mm***:***ss*
%W%	Shorthand notation for an ID line with data for **what**: **%Z%%M% %I%**
%Y%	Module type: value of the **t** flag in the **s.**file
%Z%	4-character string: '**@(#)**', recognized by **what**.

ID String

The table below explains how the SCCS identification string is determined for retrieving and creating deltas.

Determination of SCCS Identification String				
SID* Specified	−**b** Option Used†	Other Conditions	SID Retrieved	SID of Delta to be Created
none‡	no	R defaults to mR	mR.mL	mR.(mL+1)
none‡	yes	R defaults to mR	mR.mL	mR.mL.(mB+1).1
R	no	R > mR	mR.mL	R.1***
R	no	R = mR	mR.mL	mR.(mL+1)
R	yes	R > mR	mR.mL	mR.mL.(mB+1).1
R	yes	R = mR	mR.mL	mR.mL.(mB+1).1
R	–	R < mR and R does *not* exist	hR.mL**	hR.mL.(mB+1).1
R	–	Trunk succ.# in release > R and R exists	R.mL	R.mL.(mB+1).1
R.L	no	No trunk succ.	R.L	R.(L+1)
R.L	yes	No trunk succ.	R.L	R.L.(mB+1).1
R.L	–	Trunk succ. in release ≥ R	R.L	R.L.(mB+1).1
R.L.B	no	No branch succ.	R.L.B.mS	R.L.B.(mS+1)
R.L.B	yes	No branch succ.	R.L.B.mS	R.L.(mB+1).1
R.L.B.S	no	No branch succ.	R.L.B.S	R.L.B.(S+1)
R.L.B.S	yes	No branch succ.	R.L.B.S	R.L.(mB+1).1
R.L.B.S	–	Branch succ.	R.L.B.S	R.L.(mB+1).1

* ‘R’, ‘L’, ‘B’, and ‘S’ are the ‘release’, ‘level’, ‘branch’, and ‘sequence’ components of the SID, respectively; ‘m’ means ‘maximum’. Thus, for example, ‘R.mL’ means ‘the maximum level number within release R’; ‘R.L.(mB+1).1’ means ‘the first sequence number on the *new* branch (that is, maximum branch number plus one) of level L within release R’. Note: if the SID specified is of the form ‘R.L’, ‘R.L.B’, or ‘R.L.B.S’, each of the specified components *must* exist.

** ‘hR’ is the highest *existing* release that is lower than the specified, *nonexistent*, release R.

*** Forces creation of the *first* delta in a *new* release.

\# Successor.

† The −**b** option is effective only if the **b** flag is present in the file. An entry of ‘–’ means ‘irrelevant’.

‡ This case applies if the **d** (default SID) flag is *not* present in the file. If the **d** flag *is* present in the file, the SID obtained from the **d** flag is interpreted as if it had been specified on the command line. Thus, one of the other cases in this table applies.

FILES

''g-file''	version retrieved by **get**
l.*file*	file containing extracted delta table info
p.*file*	permissions (lock) file
z.*file*	temporary copy of **s.***file*

SEE ALSO

sccs(1), **sccs-admin**(1), **sccs-delta**(1), **sccs-help**(1), **sccs-prs**(1), **sccs-prt**(1), **sccs-sact**(1), **sccs-unget**(1), **what**(1), **sccsfile**(4)

Programming Utilities Guide

DIAGNOSTICS

Use the SCCS **help** command for explanations (see **sccs-help**(1)).

BUGS

If the effective user has write permission (either explicitly or implicitly) in the directory containing the SCCS files, but the real user does not, only one file may be named when using –**e**.