NAME

get − get generation from SCCS file

SYNOPSIS

get [**—r**rel[**.**lev[**.**br[**.**seq]]]] [**—c**cutoff] [**—i**incl-list] [**—x**excl-list] [**—a**serial] [**—k**] [**—e**] [**—l**[**p**]] [**—p**] [**—m**] [**—n**] [**—s**] [**—b**] [**—g**] [**—t**] name ...

DESCRIPTION

Get generates an ASCII text file from each named SCCS file according to the specifications given by its keyletter arguments, which begin with ‘‘−’’. The arguments may be specified in any order, but all keyletter arguments apply to all named SCCS files. If a directory is named, *get* behaves as though each file in the directory were specified as a named file, except that non-SCCS files (last component of the pathname does not begin with ‘‘s**.**’’), and unreadable files are silently ignored. If a name of ‘‘—’’ is given, the standard input is read; each line of the standard input is taken to be the name of an SCCS file to be processed. Again, non-SCCS files, and unreadable files are silently ignored.

The generated text is normally written into a file called the *g-file.* See **FILES**, below, for an explanation of how the name of this file is determined.

The keyletter arguments are as follows. Each is explained as though only one named file is to be processed, but the effects of any keyletter argument apply independently to each named file.

—r The SCCS identification string (SID) of the change level to be generated. If the entire argument is omitted, the meaning is the same as if the default SID were specified (see *admin*(I)). If there is no default SID in the SCCS file the highest release which has deltas is used. If only the release is specified, the level defaults to the highest level in that release. A release and level completely identifies a specific change level. If a branch is also specified and the sequence is omitted, the sequence defaults to the highest sequence in the branch. A release, level, branch, and sequence also completely identifies a specific change level. (All deltas are identified either by a 2 component SID—release and level, or by a 4 component SID—release, level, branch, and sequence. SID’s with 4 components identify deltas which have heretofore been called ‘‘non-propagating’’.)

—c Cutoff date-time, in the form YY[MM[DD[HH[MM[SS]]]]]. No delta which was created after the specified cutoff date-time will be applied. Units omitted from the date-time default to their maximum possible values; that is, ‘‘**—c**7502’’ is equivalent to ‘‘**—c**750228235959’’. Any number of non-numeric characters may separate the various 2 digit pieces of the cutoff date-time. This feature allows one to specify a cutoff date in the form: ‘‘"**—c**77/2/2 9:22:25"’’. Note that this implies that one may use the %E% and %U% identification keywords (see below) for nested *gets* within, say the input to a *send*(I) command:

```
˜!get "—c%E% %U%" s.file
```

—i This argument is used to specify a list of deltas to be included (forced to be applied). The list has the following syntax:

```
<list>      ::=<range>
              <list> , <range>
<range>     ::=<delta>
              <delta> − <delta>
<delta>     ::=<rel>
              <rel> . <lev>
              <rel> . <lev> . <br>
```

<rel> **.** <lev> **.** <br> **.** <seq>

If a level is omitted from a delta specification the highest level of the specified release is assumed. If a branch is specified, but the sequence is omitted the highest sequence of the specified branch is assumed.

—x This argument is similar to **i** except that it is followed by a list of deltas to be excluded (forced to not be applied).

—a The serial number of the change level to be generated (see *sccsfile*(V)). This keyletter is used by the *comb*(I) command; it is not a generally useful keyletter, and most users will probably never use it. If both the **r** and **a** keyletters are specified, the **a** keyletter is used. Care should be taken when using the **a** keyletter in conjunction with the **e** keyletter, as the SID of the delta to be created may not be what one expects. The**r** k eyletter can be used with the **a** and **e** keyletters to control the naming of the SID of the delta to be created.

—k This argument suppresses replacement of identification keywords (see below) by specific values. The **k** argument is implied by the **i**, **x** or **e** arguments.

—e This argument indicates that this *get* is for the purpose of making a delta with a later execution of *delta.* It causes creation, or updating of a *p-file* (see **FILES**). Another*g et* with an **e** argument, if at the same delta or for the same new SID, may not be executed until the delta is made. If the *g-file* generated by a *get* with an **e** argument is ruined, a new one may be obtained by executing another *get* with a **k** argument instead of an **e**. Note that although the**c** ar gument may be used in combination with **e**, *delta* will not use it when regenerating the *g-file* for the purpose of determining what changed. When the **e** argument is supplied the protection restrictions determined by the ceiling, the floor, and the list of users authorized to make deltas are enforced.

—l This argument causes a delta summary to be written into an *l-file* (see **FILES**). If**—lp** is used then an *l-file* is not created; the delta summary is written on the standard output instead. The *reform*(I) command can be used to truncate lines of the *l-file.*

—p This argument causes the generated text to be written to the standard output instead of to a *g-file.* All output which normally goes to the standard output goes to file descriptor 2 instead, unless the **s** argument is supplied, in which case it disappears.

—s This argument suppresses all output normally written on the standard output. However, fatal error messages (which always go to file descriptor 2) remain unaffected.

—m This argument causes each generated text line to be preceded by the SID of the delta which inserted that text line. The format is: SID, followed by a horizontal tab, followed by the text line.

—n This argument causes each generated text line to be preceded with the %M% identification keyword (see below). The format is: %M% identification keyword, followed by a horizontal tab, followed by the text line. When both the **m** and **n** arguments are supplied the format is: %M% identification keyword, followed by a horizontal tab, followed by the **m** argument format.

—b This argument is used with the **e** argument to indicate that the new delta should have an SID in a new branch. This argument is allowed only if the **b** flag exists in the file; see *admin*(I).

—g The **g** argument suppresses the actual getting of source. It is primarily used to generate an *l-file*, or to verify the existence of a particular SID.

—t The **t** argument is used to access the most recent (‘‘top’’) delta in a given release (i.e., when no **r** argument is supplied, or an argument of the form **r**rel is supplied).

For each file processed, *get* responds (on the standard output) with the SID being accessed and with the number of lines generated. If there is more than one named file or if a directory or standard input is named, each file name is printed (preceded by a newline) before it is processed. If the **i** argument is supplied included deltas are listed following the notation ‘‘Included’’; if the **x** argument is supplied excluded deltas are listed following the notation ‘‘Excluded’’.

Identifying information is inserted into the generated text by replacing *identification keywords* by appropriate values, wherever they occur. The following keywords are available:

Keyword	*Value*
%M%	Module name; either the value of the **m** flag in the file (see *admin*(I)), or the *g-file* name—see **FILES**.
%I%	SCCS identification string (SID)—**%R%.%L%.%B%.%S%.**
%R%	Release.
%L%	Level.
%B%	Branch.
%S%	Sequence.
%D%	Current date (YY/MM/DD).
%H%	Current date (MM/DD/YY).
%E%	Date of newest applied delta (YY/MM/DD).
%G%	Date of newest applied delta (MM/DD/YY).
%T%	Current time (HH:MM:SS).
%U%	Time of newest applied delta (HH:MM:SS).
%Y%	The value of the **t** flag in the file (see *admin*(I)).
%F%	File name.
%C%	Current line number. This keyword is 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.
%Z%	The 4 characters @(#) (used to construct strings recognizable by *what*(I)).
%W%	A shorthand notation for constructing *what*(I) strings for UNIX program files. **%W%** = **%Z%%M%**<horizontal-tab>**%I%**
%A%	Another shorthand notation for constructing *what*(I) strings for non-UNIX program files. **%A%** = **%Z%%Y% %M% %I%%Z%**

FILES

Several auxiliary files may be created by *get.* These files are known generically as the *g-file*, *l-file*, *p-file*, and *z-file.* The letter before the hyphen is called the tag. An auxiliary file name is formed from the SCCS file name: the last component of all SCCS file names must be of the form ‘‘**s.***modulename*’’, the auxiliary files are named by replacing the leading ‘‘s’’ with the tag. The *g-file* is an exception to this scheme: the *g-file* is named by removing the ‘‘s.’’. For example, if the SCCS file name is ‘‘s.xyz.c’’, the auxiliary file names would be ‘‘xyz.c’’, ‘‘l.xyz.c’’, ‘‘p.xyz.c’’, and ‘‘z.xyz.c’’, respectively.

The *g-file*, which contains the generated text, is created in the current directory (unless the **p** argument is supplied, or zero lines of text were generated). It is owned by the real user. If the**k** ar gument is supplied or implied its mode is 644; otherwise its mode is 444. Only the real user need have write permission in the current directory.

The *l-file* is also created (unless a **p** follows the **—l**) in the current directory, if the **l** argument is supplied; its mode is 444 and it is owned by the real user. Only the real user need have write permission in the current directory. The*l-file* contains a table sho wing which deltas were applied. The following is printed for each delta in the SCCS file:

a) Blank if the delta was applied; ‘‘*’’ otherwise.
b) Blank if the delta was applied or wasn’t applied and ignored; ‘‘*’’ if the delta wasn’t applied and wasn’t ignored.
c) A code indicating a ‘‘special’’ reason why the delta was or was not applied:
‘‘I’’: Included.
‘‘X’’: Excluded.
‘‘C’’: Cut off (by a **c** argument).
d) Blank.
e) SCCS identification string (SID).
f) Tab character.
g) Date and time (in the form YY/MM/DD HH:MM:SS) of creation.
h) Blank.
i) Creator.

The comments and MR data follow on subsequent lines, indented one horizontal tab character. A blank line terminates each entry.

The *p-file* is used to pass information resulting from a *get* with an **e** argument along to *delta.* Its contents are used to prevent a subsequent execution of *get* with an **e** argument until *delta* is executed (subject to the conditions described above under the **e** keyletter description). The *p-file* is created in the directory containing the SCCS file (which might, of course, also be the current directory), and the effective user must have write permission in that directory. Its mode is 644 and it is owned by the effective user. The format of the*p-file* is: the gotten SID, followed by a blank, followed by the SID this delta will have when it is made, followed by a blank, followed by the login name of the real user, followed by a blank, followed by the date-time of the get (*not* the cutoff date-time), followed by a blank and the **—i** keyletter argument if it was present, followed by a blank and the **—x** keyletter argument if it was present, followed by a newline. There can be an arbitrary number of lines in the *p-file* at any time; no two lines can have the same gotten SID or the same new SID.

The *z-file* is created in the directory containing the SCCS file for the duration of updating the *p-file.* The same protection restrictions as those for the *p-file* apply for the *z-file.* The *z-file* is created mode 444. It serves as a *lock-out* mechanism against simultaneous updates. Its contents are (in binary; 2 bytes) the process ID of the command (i.e., *get*) that created it.

SEE ALSO

admin(I), delta(I), prt(I), what(I), help(I), sccsfile(V),
SCCS/PWB User’s Manual by L. E. Bonanni and A. L. Glasser.

DIAGNOSTICS

Use *help*(I) for explanations.

BUGS

If the effective user has write permission (either explicitly or implicitly) in the directory containing the SCCS files, but the real user doesn’t, then only one file may be named when the **e** argument is supplied.