NAME

sccsput – SCCS check-in script

SYNOPSIS

sccsput *[options] [file-specifications]*

DESCRIPTION

Sccsput is a simple, easy to use interface to *sccs* (**s**ource **c**ode **c**ontrol **s**ystem). For each file specified as input, it checks differences against the previously archived version and prompts you for change history comments.

Sccsput uses the *sccs* utilities **admin** and **delta** to maintain versions of a given source file in a dependent directory named "SCCS". It is more than an integration of the **admin** and **delta** utilities, however:

- It checks to ensure that each file is indeed a text file (so that you do not accidentally archive ".o" files, for example).
- If you give **sccsput** a directory name, it will recur, checking-in files in the directory.
- For each file which has a corresponding "s." file, **sccsput** compares the two (using **diff**) and pipes the result through the pager.
- An option is provided so that you may direct **sccsput** to perform the differencing without checking the file into sccs.
- The "s." file is post-processed by **sccsput** so that the check-in date matches the file's modification date.

The last point is the fundamental advantage offered by **sccsput**. The ordinary *sccs* methodology uses the current date as the check-in date. This works well only for large projects in which a central project administrator is responsible for controlling the versions of source files. It does not work well for small projects, for which *sccs*'s primary advantage is its compact storage of multiple versions of a file.

By using the file's modification date as a reference, you can more easily back up to a meaningful version – by date, rather than version number. (By working exclusively in terms of modification date, you lose the ability to specify sccs release numbers – given the complexity of *sccs*'s interface for release and version numbers, this is probably not such a great loss).

Sccsput integrates all of the functions used in the *sccs* check-in process into one utility program.

OPTIONS

Some of the options which you may specify to **sccsput** are passed through to the underlying utilities. Others represent extensions:

-b is passed to **diff**, and directs it to ignore trailing blanks on a line, and to treat repeated blanks as a single blank.

-c directs **sccsput** to use **cat** rather than the **PAGER** (usually **more**) to display differences. This is most useful in an Apollo pad, since the **more** program would otherwise switch to VT100 emulator mode.

-f forces a check-in, ignoring the output of the **file** utility, which identifies text files.

-l *file* causes **sccsput** to generate a log-file of the files which are processed, and all differences which are encountered. The log-file is inherited in recursion to lower directory levels (i.e., it is written to the same place).

-n instructs **sccsput** to test for differences, but not to check the files into *sccs*.

-s suppresses some of the messages generated by the *sccs* **delta** utility describing the number of lines changed, etc.

OPERATIONS

The **sccsput** utility is designed for use in small development projects. The methodology for this tool follows:

- Develop source files "normally". Each file should contain sccs keywords (see *get (1)*) so that you will be able to distinguish checked-out files. The *sccs* keywords should appear at the top of your source file, for consistency. In C language programs, the convention is to make a string which will permit the **what** utility to show the versions of the modules which make up a program:

```
#ifndef lint
static char   sccs_id[] = "@(#)sccsput.doc1.1 88/05/05
08:07:16";
#endif
```

- Periodically archive (with **sccsput**) those versions of files which you wish to keep (You should never have programs which have new features which you wish to keep, while there are defects in other parts of the program – that would be an unsound approach to development!).
- When you reach the point of releasing the program, ensure that all source files have been checked-in. The directory editor (**ded**) is useful for reviewing the check-in dates.
- Copy the directory containing your program to the release directory. Purge all files, except those which are stored in the *sccs* subdirectories. Use **sccsget** to extract the files (the unadorned **get** utility will work, of course, but it retains the file modification dates).
- Ensure that all files have been checked-in and released. You may use **diff** to compare the directories – the only differences should be the substituted *sccs* keywords.
- Build the released version of your program. All files should be present. No embedded path names should refer to your development copy. To ensure good isolation, you may change the permissions on your development directory temporarily.

Sccsput checks your source file out after the check-in, automatically. This is done to facilitate development. A check-in simply adds the latest changes to a file onto the archive.

When checking files into *sccs*, it is a good idea to make a test run (using the "**-n**" option) so that you can inspect the differences. For example, you may have forgotten to remove (or bypass) debugging stubs. Or, you may have been editing a checked-out file (with the *sccs* keywords substituted). **Sccsput** would archive this anyway. If you forget, and wish to kill the check-in, wait until the "comments?" prompt is issued by the **delta** utility. At this point you may kill **sccsput** without having to clean up temporary files.

If you do not have write-permission on the "SCCS" directory, but wish to review changes, use the "**-n**" option. The intermediate files are written in the **/tmp** directory.

ENVIRONMENT

Sccsput is a Bourne shell script. On Apollo DOMAIN/IX, it uses System 5 features including *dirname (1)* and *getopt (1)*.

Environment variables imported by **sccsput** include:

Sccsput also uses the following environment variables:

NOTE Provides a default value for the delta comments. Normally you should provide case-by-case comments for each file. This variable is provided so that other programs can invoke **sccsput**. If the **NOTE** variable is defined (i.e., non-null) it is used; you will not be prompted for comments.

PAGER

identifies the program to use in displaying differences between the file which is being checked in, and the previously archived version. There may be a lot of differences – more than can be shown on one screen.

SCCS_DIR

specifies the directory into which the *sccs* "s." files are stored. If no specified, **sccsput** assumes "SCCS".

FILES

Sccsput uses the following files

sccsput
: the Bourne shell script

putdelta
: A utility which invokes **admin** or **delta** as required, and modifies the sccs "s." file after check-in so that the check-in date matches the file's modification date.

ANTICIPATED CHANGES

Make **sccsput** clean up temporary files if it is interrupted.

Provide a mechanism for inserting dummy version numbers so that **sccsput** can bump the release number (for genuine major releases). Currently, the SID's are restricted to 1.1, 1.2, 1.3, etc.

AUTHORS

Thomas Dickey (Software Productivity Consortium).

SEE ALSO

putdelta, sccsget, ded, admin (1), delta (1), diff (1), get (1), rmdel (1), what (1)