NAME

putdelta – SCCS check-in utility

SYNOPSIS

putdelta *[options] [file-specifications]*

DESCRIPTION

Putdelta is a simple, easy to use interface to *sccs* (**s**ource **c**ode **c**ontrol **s**ystem). It checks into the *sccs* archive all files specified.

Putdelta 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:

- If the *sccs* archive directory does not already exist, **putdelta** creates it.
- If the file has not been locked, **putdelta** creates a lock (i.e., a "p." file). If a "p." file already exists, **putdelta** will attempt to use the lock However, if the lock belongs to another user, **putdelta** will not check-in the file.
- **Putdelta** retains the executable-mode of the file which is checked-in That is, if it is a shell-script (and has a mode such as 755), the mode of *sccs* "s." file is set to 555 The **getdelta** utility works in the opposite fashion, setting the extracted file's mode to correspond to that of the "s." file.
- **Putdelta** sets the "-n" option of delta, so that your file is not deleted after check-in.
- The "s." file is post-processed by **putdelta** so that the check-in date matches the file's modification date.

The last point is the fundamental advantage offered by **putdelta**. 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.

Putdelta integrates the essential functions used in the *sccs* check-in process into one utility program.

OPTIONS

Some options which you may specify to **putdelta** are passed through to the underlying **admin** and **delta** utilities:

–s suppresses some of the messages generated by the *sccs* **admin** and **delta** utilities describing the number of lines changed, etc.

–y*text* specifies the change-history text. You will be prompted for this if you do not supply it, in versions after the first, by the **delta** utility.

Other options are handled directly by **putdelta**:

–b (binary) sets flags to ensure SCCS treats the file as binary

–f (force) creates a new delta even if no lock was made.

–k (keys) on the initial check-in, causes **admin** to require keywords.

–n (no-op) shows actions, but does not perform them.

–r*SID* specify SCCS-sid in case of multiple pending deltas

OPERATIONS

Putdelta is designed to operate with the standard **sccs** utilities with a minimum of fuss and bother It retains your working file after check-in since that facilitates development In effect, it simply appends the current version of each file which you specify onto the archive.

ENVIRONMENT

Environment variables imported by **putdelta** include:

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

TZ is the prevailing timezone, which **putdelta** uses to set the check-in date in the "s." file **Putdelta** sets the timezone to **EST5EDT**, in part to avoid conflict with **get**'s date checking.

FILES

Putdelta is a single binary file, written in C It runs in a POSIX environment.

ANTICIPATED CHANGES

Provide a mechanism for inserting dummy version numbers so that **putdelta** 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

sccsput, sccsget, getdelta, admin(1), delta(1)