

**NAME**

padm — program administration system

**SYNOPSIS**

padm cmd [ options ] [ files ]

**DESCRIPTION**

**Padm** is a collection of shell programs which will assist a project in using the Source Code Control System (SCCS). Virtually all of the commands begin with the letter 'g' and many of them are just 'g' prepended to the appropriate SCCS command.

The Padm allows a group of people to define a directory which subtends all SCCS source directories of interest. The shell variable `SCCSOURCE` is then set to this directory and *exported*. The user then defines `SUBSYSTEMS` which are of interest. Specifically, the shell variable `SUBSYSTEMS` is set to the directories relative to `SCCSOURCE` which are of interest.

As an example assume a project's SCCS files are all under the directory `/usr/proect`. If one wants to work on a specific subsystem in "project" one must do the following:

```
SCCSOURCE=/usr/project
SUBSYSTEMS="dispatcher watchdog"
export SCCSOURCE SUBSYSTEMS
```

This would imply the directories `/usr/project/dispatcher` and `/usr/project/watchdog` exist and contain SCCS files or directories. Assuming the directories contain SCCS files one may *gget* or *gdelta* any file in either of the two directories. One need only refer to a file by its non-SCCS filename. Thus if in `/usr/project/dispatcher` there is a file called `s.startup.c` he can get for the purpose of editing `s.startup.c` by doing the following:

```
gget -e startup.c
```

In addition if one wanted a particular version one could say:

```
gget -r3.4 startup.c
```

which would retrieve SCCS version 3.4 of `s.startup.c` in `/usr/project/dispatcher`. (This time no edit is implied.) In addition if one wanted to find out what versions existed one could type:

```
gprt startup.c
```

Likewise, when it comes time to put the source code back into SCCS one types:

```
gdelta startup.c
```

The `delta(1S)` command will respond with normal run of the mill delta stuff ("comments?" and "MR'S?" if the flag is set) and perform the delta.

The current version of the Padm allows the user to get a file no matter what directory one is in. This might present a problem if more than one directory is specified in `SSUBSYSTEMS`. For instance, if both `/usr/project/dispatcher` and `/usr/project/watchdog` have a 'Makefile' and the user's environment is:

```
SCCSOURCE=/usr/project
SUBSYSTEMS="dispatcher watchdog"
```

and the user types:

```
gget -e Makefile
```

the user will get the 'Makefile' from the first directory listed in \$SUBSYSTEMS. To get the 'Makefile' from watchdog the user can type:

```
gget watchdog/Makefile
```

However, when one *delta's* the file 'Makefile' if one does not specify watchdog/Makefile one will get an error (unless one is also editing the dispatcher/Makefile). Hence, under certain circumstances one could *delta* a 'Makefile' into the wrong spot. This problem could be avoided by allowing only one directory entry in \$SUBSYSTEMS. However, at this time, we have had not had this problem so no attempt has been made to solve it.

When a subsystem is "finished" (?) it is assumed that the programmer wants to *mark* it in some way such that two months later when one has fixed a bug one can know what it was that went to the field. For this purpose, the *gmark*(1S) command exists. One can type:

```
gmark subsys
```

and the latest release of every SCCS file in subsys is remembered in an SCCS file originated and maintained completely by *gmark*(1S). Thus, if one were working on the shell in /usr/project/sh and wanted to mark the shell one would type:

```
gmark sh
```

The mark file would contain the name and version of each SCCS file in /usr/project/sh. To see this one could type:

```
gmark -L sh
```

which gives a partially formatted list of files with their versions. Note, the markfile is maintained in SCCS format and the only interface to it is through the *gmark*(1S) command. When making a new markfile the *gmark* command *get's* for the purpose of editing the markfile; notes the most recent versions ( with a 'get -g .' ) of each SCCS file in the named directory (excluding the markfile itself); and *delta's* this new information into the markfile. The '-L' command just does a 'get -p' on the markfile.

In addition, when one *gmark's* a directory, all subtending directories are marked. Also, with each mark the user must supply a comment which goes into each new delta'ed markfile.

For further, information on the various commands one is urged to read the individual manual pages for the following commands: *gget*(1S), *gdelta*(1S), *gadd*(1S), *gmark*(1S), *gpri*(1S) and *gadmin*(1S). Also note the convention of upper case letters for options meant specifically for the padm commands. This is an attempt to avoid collisions with SCCS options which at present are all lower case letters.