Bell Telephone Laboratories, Incorporated   - 1 -   PA-1C600-01
PROGRAM APPLICATION INSTRUCTION   Section 7 (V)
Issue 1, 1 October 1977
AT&TCo SPCS

SCCSFILE(V)   **SCCSFILE(V)**

NAME

         sccsfile — format of SCCS file

DESCRIPTION

         Each record in an SCCS file is preceded by a one byte count, followed by that many bytes of data, followed by a pad byte if the count is even. The first byte of the file is also a pad byte. The count is stored offset from −128; i.e. a count of zero is stored as −128 and a count of 255 is stored as 127. The first byte of the file is null, but the other pad bytes contain garbage.

         An SCCS file consists of four logical parts: the *header* contains assorted descriptive and control information about the file, the *release table* contains a count of the number of deltas in each release, the *delta table* contains information about each delta and the *body* contains the actual text records intermixed with control records. Each part is described below in terms of C structures.

         Header. The header contains the following 226-byte structure:

```
struct Header {
        int  Hmagicno;
        char Htype[10];
        char Hpers[14];
        char Hdesc[100];
        int  Hfloor;
        int  Hceil;
        int  Hsw[5];
        int  Hrdef;
        char Hulist[32];
        char Hexpand[50];
        int  Hash;
};
```

         The header fields have the following meanings. All "char" arrays except for *Hulist* must be null-terminated.

Hmagicno	This identifies SCCS files; it contains a binary 7.
Htype	This is used to document the contents of the file.
Hpers	This field is documentation only.
Hdesc	This field is documentation only.
Hfloor	The lowest release in which deltas may be added.
Hceil	The highest release in which deltas may be added. A zero is equivalent to 9999.
Hsw	Release locks. A non-zero value in one of the 5 elements locks the release whose number is that value.
Hrdef	The default release. A zero means that the default is determined from the release table in the manner described in *admin*
Hulist	A bit map on the user IDs 0 to 255; a 1 means the corresponding user can add deltas. All zeros is equivalent to all ones. To form the map, bytes are taken left-to-right within the 32-byte field and bits are taken right-to-left within each byte.

SCCSFILE(V)   **SCCSFILE(V)**

Hexpand	Room for expansion; must contain all zeros for now.
Hash	A hash count of the entire file. The hash count is determined by zeroing this field, appending a null byte to the file (SCCS files are always of odd length), and summing the words of the file using integer arithmetic. The result is then stored into this field.

Release table. Each release table record contains the following 4-byte structure:

```
struct Reltab {
        int  Rrel;
        int  Rlevs;
};
```

There is one release table record for each release that has deltas, which gives the number of deltas in that release. If there are no deltas in a release that release need not appear in the release table at all, even if later releases do. The release records are kept in order, with the latest release first. The release table is terminated with an entry for release 0.

Delta table. Each delta table record contains the following structure. All "char" arrays must be null terminated.

```
struct Deltab {
        int  Drel;
        int  Dlev;
        char Dtype;
        char Dfill;;
        int  Ddatetime[2];
        char Dpgmr[8];
        char Dhist[100];
};
```

The length of a delta table record varies from 19 to 118 bytes, depending on the length of *Dhist*. The fields are as follows:

Drel	Release number.
Dlev	Level number.
Dtype	Type of delta: "D" if normal, "U" if unacknowledged non-propagating, "P" if acknowledged non-propagating, "R" if removed, "H" if changed history, and "C" if combined.
Dfill	Blank.
Ddatetime	The time, in the form returned by *time*
Dpgmr	The login name corresponding to the real user ID at the time the delta was created.
Dhist	History information associated with the delta.

Body. The body consists of text records and control records. Text records are stored without the newline character and are not null-terminated. Control records have the following structure:

```
struct Control {
        int  Crel;
        int  Clev;
        char Cctl;
```

Bell Telephone Laboratories, Incorporated   - 3 -   PA-1C600-01
PROGRAM APPLICATION INSTRUCTION   Section 7 (V)
Issue 1, 1 October 1977
AT&TCo SPCS

SCCSFILE(V)   **SCCSFILE(V)**

```
};
```

Control records are 5 bytes long. There are three kinds of control records: *insert delete* and *end* represented by a *Cctl* of -11, -12 and -13 respectively. When text records are inserted, they are bracketed by insert and end controls for the release and level of the delta of which they are a part. Deleted text records are bracketed with delete and end controls. The body (and hence the SCCS file) is terminated by an end control for release 0, level 0.

SEE ALSO

prt(I), admin(I), get(I), delta(I),
The Source Code Control System TM75-9152-3. *SCCS/PWB User's Manual,* TM75-9144-1.