Published: 11/17/67

Identification

Obtain information about entry in File System hierarchy
entry_status
Charles Garman

Purpose

The procedure entry_status obtains for its caller the information stored in the file system hierarchy for an entry in a directory. It calls the Basic File System primitive status, and then extracts the information from the returned structure. Depending on which of several entries was called, it assigns to the various arguments of its calling sequence the corresponding data from the structure, in (nominally) logical groupings: dates, lengths, miscellaneous switches and values, etc. For information on the meaning of the possible returned values, see the appendix to this MSPM section which translates the variables named below into a short verbal description and gives references for further information.

Usage

```
dcl (dir, ent) char(N),

     /*N is defined by user */

     chase fixed bin (1),

     scode fixed bin (17);  /* First four
     arguments are common to all calls */

dcl /* declarations for return arguments; see
     Appendix for explanation of return arguments */

     uid bit (70),

     (dtu, dtem, dtd, dtsm, rd) fixed bin (71),

     (ml, cl) fixed bin (9),

     bc fixed bin (24),

     (dirsw, nomore) fixed bin (1),

     (etype, optsw, consistsw, usage) fixed bin (2),

     mode char (5),

     (nnames, usagect) fixed bin (17);
```

```
call entry_status$type (dir, ent, chase, scode, etype);

call entry_status$dates (dir, ent, chase, scode, uid,
     dtu, dtem, dtd, dtsm, rd);

call entry_status$lengths (dir, ent, chase, scode, ml,
     cl, bc);

call entry_status$switches (dir, ent, chase, scode,
     dirsw, mode, optsw, consistsw, usage, usagect,
     nomore, nnames);
```

The first four arguments are the same for all calls: the directory pathname, dir, and the entry within the directory, ent, define the hierarchy entry about which the information is desired; dir, ent, and chase are the same as the corresponding arguments to the Directory Control primitive status (BG.8.02). Briefly, if chase is 0, the call to status searches only the directory dir, and returns information for a branch or a link as defined by ent. If chase is non-zero then if ent is a link, status searches until: 1) it finds the effective branch to which ent points; 2) it appears that the link is a loop (a depth of 10, currently); or 3) a directory appears which it cannot search.

Scode is set to 0 if all the desired return parameters were filled in; it is set to 1 if any or all the parameters were not relevant to the type of the entry (i.e., the entry was a link, in which case dtsm would not be relevant, for example). If scode is 1 then all non-relevant arguments will be set to 0 or "0"b as appropriate (or " " for mode). If an error occurred in the call to status, the value of scode is twice the file system error code returned by status; and none of the return arguments are filled in, except for the $type entry (see next paragraph).

Except for the $type entry, the names of the return arguments correspond to the names of the second-level structure items in the branches array (BG.8.02) and the precisions of the values are the same as the size in bits of their like-named brethren in the structure. See appendix for a list of the return arguments. etype in the $type call has value 0 if the entry is a link, 1 if it is a nondirectory branch, 2 if it is a directory branch, and 3 if there was an error code returned by status. For the mode arguments of $switches, the 5-bit value is converted to the equivalent

symbolic character string, with the letters occurring left-justified in the string (with trailing blanks if any bits in the mode switches were off).

Dir and ent may be varying or non-varying strings; cv_string (BY.10.02) is called to "convert" each to non-varying before the call to status.

Implementation

At each entry point, an index is set to a value corresponding to the entry. Control is then passed to a common section, which calls the Directory Control primitive status with dir, ent and chase. If no errors occurred (i.e., the entry was found), control passes to a section of code corresponding to the invoked entry, which extracts the desired data from one of the status structures (see declarations in BG.8.02), places it in the return arguments, and returns.

Appendix:

Abbreviations for returned items

uid	unique identifier for entry
dtu	date/time entry referenced
dtem	date/time entry (in directory) modified
dtd	date/time dumped (backup storage)
dtsm	date/time segment modified
rd	retention date
ml	maximum length, in blocks of 1024 words
cl	current length, in blocks of 1024 words
bc	bit-count of data in segment pointed to by branch; may be set by user via set_count, BY.2.07
dirsw	value=1 if branch is a directory
mode	reference attributes "trewa" (see BX.8.00)
optsw	copy and relate option switches (see BD.3)
consistsw	"consistency switch" (BH)
usage	current status of segment-usage lock (none, read, read-write, data-share) respectively
usagect	number of users (i.e., processes) currently accessing segment with lock status usage, above
nomore	set to 1 if no more processes are to access the segment with status usage, because a process wishes to change the lock mode
nnames	number of names in the entry ($\geq$1).