DR:MOWN

Access Multiple Owned Objects in a Directory

Input parameters:

IP1	C:	A directory (with OB.UACC, OB.IMPL)
IP2	C:	A capability list
IP3	D:	A starting index $\underline{i}$, $\underline{i} \geq 1$
IP4	D:	A count $\underline{k}$, $\underline{k} \geq 1$

Return parameters:

RDAT(0)	A count $\underline{m}$ of objects actually placed in the c-list, $\underline{m} \geq 0$
RDAT(1)	A count $\underline{n}$ of owned objects remaining in the directory, $\underline{n} \geq 0$

Action:

Capabilities for objects appearing in ownership entries of a given directory (IP1) are copied into a given c-list (IP2). The index $\underline{i}$ (IP3) controls where the copying process begins -- $\underline{i} = 1$ means the first directory ownership entry, $\underline{i} = 2$ means the second, etc. Slots in the c-list are always filled starting with the first. The number $\underline{m}$ of capabilities moved (returned as RDAT(0)) is equal to the minimum of the following three quantities:

(i) the parameter $\underline{k}$ (IP4);
(ii) the length of the c-list;
(iii) $(\underline{p}-\underline{i}+1)$, where $\underline{p}$ is the number of ownership entries in the directory, or zero if $\underline{p} < \underline{i}$.

Returned as RDAT(1) is the quantity $\underline{n} = \underline{p}-\underline{m}$ = the number of ownership entries in the directory after the ones accessed. Note that this action never freturns.

Errors:

No such (directory, c-list); non-positive (starting index, count).

PMcJ
27 Oct 71

DR:GMDA

Get Multiple Disk Adresses from a Directory File

Input parameters:

IP1 C: A disk file, in the format of a directory
IP2 C: An ecs file, to be filled with (disk address, unique name) words
IP3 D: A starting index $\underline{i}$, $\underline{i} \geqslant 1$
IP4 D: A count $\underline{k}$, $\underline{k} \geqslant 1$

Return parameters:

RDAT(0) A count $\underline{m}$ of disk addresses actually returned, $\underline{m} \geqslant 0$
RDAT(1) A count $\underline{n}$ of disk addresses following those returned, $\underline{n} \geqslant 0$

Action:

The file IP1 is assumed to represent a directory -- presumably one with no-longer-valid disk addresses. This action finds (disk address, unique name) words in the directory file and copies them into the ecs file IP2. Copying begins with the $\underline{i}$-th disk address encountered, while the ecs file is always filled up starting at its file address zero. The number of disk-address words copied is $\underline{m}$, the minimum of:

(i) the count $\underline{k}$ (IP4);
(ii) the length of the ecs file;
(iii) $(\underline{p}-\underline{i}+1)$, where $\underline{p}$ is the total number of disk addresses contained in the directory file, or zero if $\underline{i} > \underline{p}$.

~~Missing blocks in the ecs file will be created as needed.~~ Returned parameters include $\underline{m}$ and $\underline{n} = \underline{p}-\underline{m}$ = the number of disk addresses in the directory file after the ones accessed. The action never freturns.

Errors:

No such (disk file, ecs file); incorrect format for directory file; non-positive (starting index, count).

PMcJ
27 Oct 1971

DR:PMDA

Put Multiple Disk Addresses into a Directory File

Input parameters:

IP1 C: A disk file, in the format of a directory
IP2 C: An ecs file, containing (disk address, unique name) words
IP3 D: A starting index $\underline{i}$, $\underline{i} \geq 1$
IP4 D: A count $\underline{k}$, $\underline{k} \geq 1$

Return parameters:

RDAT(0) A count $\underline{m}$ of disk addresses actually replaced, $\underline{m} \geq 0$
RDAT(1) A count $\underline{n}$ of disk addresses yet to be replaced, $\underline{n} \geq 0$

Action, errors:

Same as DR:GMDA (Get Multiple Disk Address), except the disk-address words are copied *from* the ecs file *to* the directory file, rather than the other way around.

PMcJ
27 Oct 1971