Identification

link
R. J. Feiertag, S. Rosenbaum, W. H. Southworth, P. Smith,
E. Bjorkman

Purpose

This command creates links in a directory.

Usage

At command level:

        link path1   -path2-

As a subroutine:

        call link_(path1, path2, code);

    or   call link_(path1, code);

    dcl   path1 char(*),   /*path of entry to which
                                        link is to point*/

          path2 char(*),   /*path pointing to link
                                        entry to be created*/

          code fixed bin(17);   /*command system error
                                        code*/

A link which points to path1 is created at the entry effectively
pointed to by path2. The entry effectively pointed to
by path2 must not exist and the entry specified by path1
may or may not exist.

The append mode is necessary in the directory of the entry
effectively pointed to by path2. If path2 is not specified
then path2 is taken as the working directory and the entry
name specified by path1. The equal convention described
in BX.8.00 may be used.

Examples

1)   link george >system_library>fred

> A link with name "fred" is created in the directory ">system_library" which points to the entry "george" in the working directory.

2)   link >user_dir_dir>fruit>apple

> A link with name "apple" is created in the working directory which points to the entry "apple" in the directory ">user_dir_dir>fruit".

3)   link ([files >user_dir_dir>fruit>**])

> For each entry in the directory ">user_dir_dir>fruit" a link with the same name as that entry is made in the working directory.

4)   If there exists a link "george" in the working directory pointing to a non-existant entry "fred" in the working directory then the command:

> link isaac george

will create a link entry "fred" in the working directory that points to an entry "isaac" in the working directory. Note that "george" now effectively points to "isaac".

Implementation

setpath breaks the path names into directories and entry names. equalcomp (BY.2.06) is called to interpret any equal signs if a second path name is given. If there is only one path name then the second path name is formed from the working directory, by a call to wdir (BY.17.01), and the first entry name. This second path name is chased by a call to ufo$chase (BY.2.01) to find the nonexistant entry pointed to, and the link is made in the entry by a call to ufo$append1 (BY.2.01). Errors are reported by a call to command_error.