Identification

chasepath
R. J. Feiertag, S. Rosenbaum

Purpose

This command retrieves the path name of the entry effectively pointed to by a specified entry.

Usage

At command level:

chasepath path

As a subroutine:

call chasepath_(path, path2, code);

dcl path char(*), /*path entry to be chased*/

path2 char(511) var, /*path of entry effectively pointed to, returned by chasepath*/

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

The path of the branch effectively pointed to by path is returned as a complete path name (i.e., if branch is a directory the returned path name will end with ">"). Read mode is necessary in the directory of the branch effectively pointed to.

Examples

1) a) If "fred" is a directory branch in the working directory and the working directory is ">user_dir_dir>user" then the command:

chasepath fred

will produce:

>user_dir_dir>user>fred>
b) If "fred" is a link pointing to the non-directory branch ">system_library>george" then the command:

    chasepath fred

will produce:

    >system_library>george

2) If the link entry "apples" in the directory ">old_dir" points to ">system_library>fruit" then the command:

    link [chasepath >old_dir>apples]

will create a link entry with name "fruit" in the working directory which points to ">system_library>fruit".

Implementation

setpath breaks the path name into a directory and entry name and ufo$chase (BY.2.01) chases the entry. The resulting path name is returned. Errors are reported by a call to command_error.