Tech Info Library

HyperCard: Global Paths & Avoiding Lost Pathnames

Revised: 6/24/90
Security: Everyone

HyperCard: Global Paths & Avoiding Lost Pathnames

===

Article Created: 24 April 1989
Article Last Reviewed: 14 July 1992
Article Last Updated: 14 July 1992

TOPIC ---

Whenever I move various applications, stacks, and folders comprising my HyperCard projects to another hard disk, I am inundated with "Where is..." dialog boxes as my scripts try to re-find the correct paths to jump between stacks.

DISCUSSION --

Some HyperCard developers have asked if there is any way to provide a global path command or to assign physical or logical names that initiate at startup. This becomes important when a program is moved to another machine (with different names). This kind of porting can make all the pathnames incorrect with the result, as above, that the program constantly asks for help in finding files.

Many users try to avoid this problem by beginning the path names with the actual volume name. This is not necessary.

An easier solution is to load the folders in question into the same folder that contains the HyperCard application itself, then place a colon at the beginning of the path name, like this:

```
:HyperCard Ideas:Idea Stacks:
```

The first colon in this statement tells HyperCard to look for the "HyperCard Ideas" folder within the same folder that contains HyperCard itself. HyperCard will then continue on to the "Ideas Stack" folder for the chosen stack.

The problem can also be handled by getting the Long Name of the currently executing stack. This name returns the path to the stack. As long as the other stacks or accessed files are kept in the same folder, the pathname can be manipulated to open those files. You can even place the stacks or files in subfolders, as long as the folder names are placed in the path.

An example:

```
Function PathWay
  put the long name of this stack into NewName
  delete char 1 to 7 of NewName
  repeat
    if offset(":",NewName) is 0 then
      exit repeat
    else
      put char 1 to offset(":",NewName) of NewName after NewPath
      delete char 1 to offset(":",NewName) of NewName
    end if
  end repeat
  return NewPath
end PathWay

put PathWay () into the message
```

This piece of code returns the current path to the stack. By appending the name of a stack or file onto the end of the returned path, your stacks can open any stacks or files in the same folder. By placing a folder name and the name of the stack after the returned path, you can open the stacks within a subfolder.

Tech Info Library Article Number:3868