MoveFolder

MoveFolder [***flags***][***srcFolderStr***] [**as** ***destFolderStr***]

The MoveFolder operation moves or renames a folder on disk. A folder is renamed by "moving" it into the same folder it is already in, but with a different name.

Warning: *The* `MoveFolder` *command can destroy data* by overwriting another folder and its contents!

If you overwrite an existing folder on disk, MoveFolder will do so only if permission is granted by the user. The default behavior is to display a dialog asking for permission. The user can alter this behavior via the Miscellaneous Settings dialog's Misc category.

If permission is denied, the folder will not be moved and V_Flag will return 1088 (Command is disabled) or 1275 (You denied permission to overwrite a folder). Command execution will cease unless the /Z flag is specified.

Parameters

srcFolderStr can be a full path to the folder to be moved or renamed (in which case /P is not needed), a partial path relative to the folder associated with *pathName*, or the name of a folder within the folder associated with *pathName*.

If the location of the source folder cannot be determined from *srcFolderStr* and *pathName*, it displays a Select Folder dialog allowing you to specify the source.

If /P=*pathName* is given, but *srcFolderStr* is not, then the folder associated with *pathName* is moved or renamed.

destFolderStr specifies the final location of the folder or, if /D is used, the parent of the final location of the folder.

destFolderStr can be a full path to the output (destination) folder (in which case /P is not needed), or a partial path relative to the folder associated with *pathName*.

If the location of the destination folder cannot be determined from *destFolderStr* and *pathName*, it displays a Save Folder dialog allowing you to specify the destination.

If you use a full or partial path for either file, see **Path Separators** on page III-451 for details on forming the path.

Flags

/D — Interprets *destFolderStr* as the name of (or path to) an existing folder (or "directory") to move the source folder into. Without /D, it interprets *destFolderStr* as the name of (or path to) the moved folder.

If *destFolderStr* is not a full path to a folder, it is relative to the source folder.

/I [=*i*] — Specifies the level of interactivity with the user.

- /I=0: Interactive only if *srcFolderStr* or *destFolderStr* is not specified or if the source folder is missing. (Same as if /I was not specified.)
- /I=1: Interactive even if *srcFolderStr* is specified and the source folder exists.
- /I=2: Interactive even if *destFolderStr* is specified.
- /I=3: Interactive even if *srcFolderStr* is specified and the source folder exists. Same as /I only.

/M=*messageStr* — Specifies the prompt message in the Open File dialog. If /S is not used, then *messageStr* will be used for both Open File and for Save File dialogs.

/O — Overwrite existing destination folder, if any. This deletes the existing destination folder. When /O is specified, the source folder can't be moved into an existing folder without specifying the name of the moved folder in *destFolderStr*.

/P=*pathName* — Specifies the folder for relative paths in *srcFolderStr* and *destFolderStr*. *pathName* is the name of an existing symbolic path.

If *srcFolderStr* is omitted, the folder associated with *pathName* is moved. If *destFolderStr* is omitted, the source folder is moved into the folder associated with *pathName*.

Using /P means that *srcFolderStr* (if specified) and *destFolderStr* must be either simple folder names or paths relative to the folder specified by *pathName*.

/S=*saveMessageStr* Specifies the prompt message in the Save File dialog.

/Z[=z] Prevents procedure execution from aborting if it attempts to move a folder that does not exist. Use /Z if you want to handle this case in your procedures rather than having execution abort.

- /Z=0: Same as no /Z.
- /Z=1: Moves a folder only if it exists. /Z alone is equivalent to /Z=1.
- /Z=2: Moves a folder if it exists or displays a dialog if it does not exist.

Variables

The MoveFolder operation returns information in the following variables:

`V_flag` Set to zero if the file was moved, to -1 if the user cancelled either the Open File or Save File dialogs, and to some nonzero value if an error occurred, such as the specified file does not exist.

`S_fileName` Stores the full path to the folder that was moved, with a trailing colon. If an error occurred or if the user cancelled, it is set to an empty string.

`S_path` Stores the full path of the moved folder, with a trailing colon. If an error occurred or if the user cancelled, it is set to an empty string.

Details

You can use only /P=*pathName* (omitting *srcFolderStr*) to specify the source folder to be moved.

A folder path should not end with single Path Separators. For example:

```
MoveFolder "Macintosh HD:folder" as "Macintosh HD:Renamed Folder:"
MoveFolder "Macintosh HD:folder:" as "Macintosh HD:Renamed Folder"
MoveFolder "Macintosh HD:folder:" as "Macintosh HD:Renamed Folder:"
```

will do weird, unexpected things (and probably damaging things when /O is also used). Instead, use:

```
MoveFolder "Macintosh HD:folder" as "Macintosh HD:Renamed Folder"
```

Beware of PathInfo and other command which return paths with an ending path separator. (They can be removed with the **RemoveEnding** function.)

A folder may not be moved into one of its own subfolders.

Conversely, the command:

```
MoveFolder/O/P=myPath "afolder"
```

which attempts to overwrite the folder associated with myPath with a folder that is inside it (namely "afolder") is not allowed. Instead, use:

```
MoveFolder/O/P=myPath "::afolder"
```

On Windows, renaming or moving a folder never updates the value of any Igor Symbolic Paths that point to a moved folder:

```
// Create a folder
NewPath/O/C myPath "C:\\My Data\\My Work"

// Move the folder
MoveFolder/P=myPath as "C:\\My Data\\Moved"

// Display the path's value
PathInfo myPath               // (or use the Path Status dialog)
Print S_Path
• C:My Data:My Work
```

You can use PathInfo to determine if a folder referred to by an Igor symbolic path exists and where it is on the disk. Use NewPath/O to reset the path's value.

On the Macintosh, however, renaming or moving a folder on the same volume does alter the value of symbolic path. This is because MoveFolder uses a Mac OS alias to keep track of the folder. A folder renamed or moved on the same volume retains the original "volume refnum" and "directory ID" stored in the alias mechanism, so that the alias (and hence Igor's symbolic path) remains pointing to the moved folder. After moving the folder, using the unchanged volume refnum and directory ID (in PathInfo or when you use /P=pathName) returns the updated path.