Examples

```
MoveDataFolder root:DF0, root:archive      // Move DF0 into archive
```

See Also

See the **DuplicateDataFolder** operation. Chapter II-8, **Data Folders**.

MoveFile

`MoveFile` `[`***`flags`***`][`***`srcFileStr`***`] [`**`as`** ***`destFileOrFolderStr`***`]`

The MoveFile operation moves or renames a file on disk. A file is renamed by “moving” it to the same folder it is already in using a different name.

Parameters

srcFileStr can be a full path to the file 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 file in the folder associated with *pathName*.

If Igor can not determine the location of the file from *srcFileStr* and *pathName*, it displays an Open File dialog allowing you to specify the source file.

destFileOrFolderStr is interpreted as the name of (or path to) an existing folder when /D is specified, otherwise it is interpreted as the name of (or path to) a possibly existing file.

If *destFileOrFolderStr* is a partial path, it is relative to the folder associated with *pathName*.

If /D is specified, the source file is moved inside the folder using the source file’s name.

If Igor can not determine the location of the destination file from *pathName*, *srcFileStr*, and *destFileOrFolderStr*, it displays a Save File dialog allowing you to specify the destination file (and folder).

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

Folder paths should not end with single Path Separators. See the **Details** section for **MoveFolder**.

Flags

/D — Interprets *destFileOrFolderStr* as the name of (or path to) an existing folder (or directory). Without /D, *destFileOrFolderStr* is the name of (or path to) a file.

If *destFileOrFolderStr* is not a full path to a folder, it is relative to the folder associated with *pathName*.

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

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

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

/O — Overwrite existing destination file, if any. Without /O, the user is asked if replacing the existing file is to be allowed.

/P=*pathName* — Specifies the folder to look in for the source file, and the folder into which the file is copied. *pathName* is the name of an existing symbolic path.

Using /P means that both *srcFileStr* and *destFileOrFolderStr* must be either simple file or 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 file 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 file only if it exists. /Z alone is equivalent to /Z=1.
/Z=2:	Moves a file if it exists or displays a dialog if it does not exist.

Variables

The MoveFile 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 where the file was moved from. If an error occurred or if the user cancelled, it is set to an empty string.

`S_path` Stores the full path where the file was moved to. If an error occurred or if the user cancelled, it is set to an empty string.

Examples

Rename a file, using full paths:

```
MoveFile "HD:folder:aFile.txt" as "HD:folder:bFile.txt"
```

Rename a file, using a symbolic path:

```
MoveFile/P=myPath "aFile.txt" as "bFile.txt"
```

Move a file into a subfolder (the subfolder must exist):

```
MoveFile/D "Macintosh HD:folder:aFile.txt" as ":subfolder"
```

Move a file into an unrelated folder (the subfolder must exist):

```
MoveFile/D "Macintosh HD:folder:afile.txt" as "Server:archive"
```

Move a file from one folder to another and rename it:

```
MoveFile "Macintosh HD:folder:afile.txt" as "Server:archive:destFile.txt"
```

Move user-selected file into a particular folder:

```
MoveFile/D as "C:My Data:Selected Files Folder"
```

Move user-selected file in any folder as `bFile.txt` in same folder:

```
MoveFile as "bFile.txt"
```

Move user-selected file in any folder as `bFile.txt` in any folder:

```
MoveFile/I=2 as "bFile.txt"
```

See Also

The **Open**, **MoveFolder**, **CopyFolder**, **NewPath**, and **CreateAliasShortcut** operations. The **IndexedFile** function. **Symbolic Paths** on page II-22.