#69: Setting ioFDirIndex in PBGetCatInfo Calls

See also: The File Manager
Technical Note #24—Available Volumes and Files
Technical Note #67—Finding the Blessed Folder

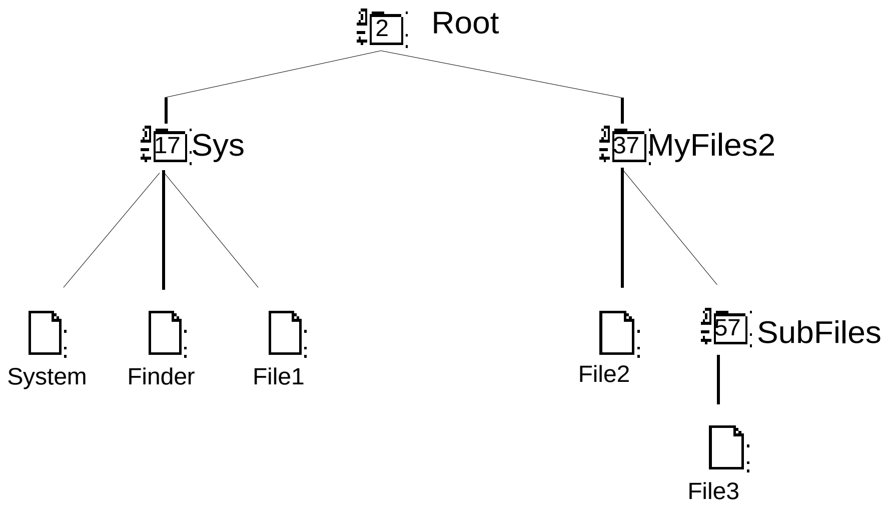
Written by: Jim Friedlander February 15, 1986
Updated: March 1, 1988

This technical note describes how to set `ioFDirIndex` for `PBGetCatInfo`.

The File Manager chapter of *Inside Macintosh* volume IV is not very specific in describing how to use `ioFDirIndex` when calling `PBGetCatInfo`. It correctly says that `ioFDirIndex` should be positive if you are making indexed calls to `PBGetCatInfo` (analogous to making indexed calls to `PBGetVInfo` as described in Technical Note #24). However, the statement “If `ioFDirIndex` is negative or 0, the File Manager returns information about the file having the name in `ioNamePtr`...” is not specific enough.

If `ioFDirIndex` is 0, you will get information about files or directories, depending on what is specified by `ioNamePtr^`.

If `ioFDirIndex` is –1, you will get information about directories only. The name in `ioNamePtr^` is ignored. For example, given the following tree structure (with sample `DirIDs` for the directories):



Calling `PBGetCatInfo`

We will now make calls to `PBGetCatInfo` of the form:

```
    err:= PBGetCatInfo(@myCInfoPBRec,FALSE);
```

Note: We will assume that we just have a `WDRefnum` and a file name—the information that `SFGetFile` returns.

Setting up the parameter block

We will use the following fields in the parameter block. Before the call, `ioCompletion` will always be set to `NIL`, `ioNamePtr` will always point at a `str255`, `ioVRefNum` will always contain a `WDRefNum` that references the directory 'SubFiles', and offset 48 (`dirID`/`flNum`) will always contain a zero:

Offset in parameter block	Variable name(s)
12	`ioCompletion`
18	`ioNamePtr`
22	`ioVRefNum`
28	`ioFDirIndex`
48	`ioDirID`/`ioFLNum`/`ioDrDirID`
100	`ioDrParID`/`ioFlParID`

Sample calls to `PBGetCatInfo`

The first example will call `PBGetCatInfo` for the file 'File3'—we will get information about the file (`ioFDirIndex` = 0):

Before the call		**After the call**	
`ioNamePtr^`:	'File3'	`ioNamePtr^`:	'File3'
`ioFDirIndex`:	**0**	Offset 48(`ioFLNum`):	a file number
		Offset 100(`parID`):	57

Now we will get information about the directory that is specified by the `iovRefNum` (`ioFDirIndex` = –1). Notice that `ioNamePtr^` is ignored:

Before the call		**After the call**	
`ioNamePtr^`:	ignored	`ioNamePtr^`:	'SubFiles'
`ioFDirIndex`:	**–1**	Offset 48(`dirID`):	57
		Offset 100(`parID`):	37

Notice that, since `ioNamePtr^` is ignored, Offset 48 contains the `dirID` of the directory specified by the `iovRefNum` that we passed in and that Offset 100 contains the parent ID of that directory.
Notice that if we try to get information about the directory SubFiles by calling `PBGetCatInfo` with `ioFDirIndex` set to 0, we will get an error –43 (File not found error) back because there is neither a file nor a directory with the name 'SubFiles' in the directory that `ioVRefNum` refers to.

If you specify a **full** pathname in `ioNamePtr^`, then the call returns information about that path, whether it is a directory or a file. The `ioVRefNum` is ignored:

Before the call		**After the call**	
`ioNamePtr^`:	'Root:Sys'	`ioNamePtr^`:	'Root:Sys'
`ioFDirIndex`:	**0**	Offset 48 (`dirID`):	17
`ioVRefNum`:	refers to 'SubFiles'	Offset 100 (`parID`):	2

Or, if the full pathname specifies a file, the `iovRefNum` is overridden:

Before the call		**After the call**	
`ioNamePtr^`:	'Root:Sys:Finder'	`ioNamePtr^`:	'Root:Sys:Finder'
`ioFDirIndex`:	**0**	Offset 48 (`flNum`):	fileNumber
`ioVRefNum`:	refers to 'SubFiles'	Offset 100 (`parID`):	17

Or, given an `ioVRefNum` that refers to MyFiles2 and a partial pathname in `ioNamePtr`^, we'll get information about the directory 'SubFiles':

Before the call		**After the call**	
`ioNamePtr^`:	'SubFiles'	`ioNamePtr^`:	'SubFiles'
`ioFDirIndex`:	**0**	Offset 48 (`dirID`):	57
`ioVRefNum`:	refers to 'MyFiles2'	Offset 100 (`parID`):	37

PBGetCatInfo and The Poor Man's Search Path (PMSP)

If no `ioDirID` is specified (`ioDirID` is set to zero), calls to `PBGetCatInfo` will return information about a file in the specified directory, but, if no such file is found, will continue searching down the Poor Man's Search Path. **Note:** the PMSP is not used if `ioFDirIndex` is non-zero (either –1 or >0). The default PMSP includes the directory specified by `ioVRefNum` (or, if `ioVRefNum` is 0, the default directory) and the directory that contains the System File and the Finder—the blessed folder. So for example:

Before the call		**After the call**	
`ioNamePtr^`:	'System'	`ioNamePtr^`:	'System'
`ioFDirIndex`:	**0**	Offset 48 (`ioFLNum`):	a file number
		Offset 100 (`parID`):	17

You must be careful when using `PBGetCatInfo` in this way to make sure that the file you're getting information about is in the directory that you think it is, and not in a directory further down the Poor Man's Search Path. Of course, this does not present a problem if you are using the `fName` and the `vRefNum` that `SFGetFile` returns.
If you want to specifically look at a file in the blessed folder, please use the technique described in technical note #67 to get the `dirID` of the 'blessed folder' and then use that `dirID` as input in the `ioDirID` field of the parameter block (offset 48).

Summary (`DirID` = 0 in all the following):

If `ioFDirIndex` is set to 0:

1) Information will be returned about files.

2) Information will be returned about directories as follows:

A) If a partial pathname is specified by `ioNamePtr^` then the volume and directory will be taken from `ioVRefNum`.

B) If a full pathname is specified by `ioNamePtr^`. In this case, `ioVRefNum` is ignored.

If `ioFDirIndex` is set to –1:

1) Only information about directories will be returned.

2) The name pointed to by `ioNamePtr` is ignored.

3) If `DirID` and `ioVRefNum` are 0, you'll get information about the default directory.