

November, 2002

Advisor Answers

Deleting a hidden file

VFP 7/6/5/3

Q: How can I find and delete a hidden file using VFP code?

-Alison Michael (via CompuServe.COM)

A: Working with hidden files in VFP is interesting. Some commands and functions can see them, while others cannot. For example, MODIFY FILE and MODIFY COMMAND are unable to open the file. If you specify the file name in the command, you get error 202 ("Invalid path or file name"). However, if you issue MODIFY FILE ? or MODIFY COMMAND ?, your Explorer settings determine whether you see the file in the dialog. If you choose a hidden file, you get error 1705 ("File access is denied").

VFP has mixed behavior with tables that are hidden, too. USE happily opens the table and SET ORDER sees all the tags and lets you set any order. However, attempting to add a hidden table to a database with ADD TABLE results in error 1 ("File does not exist"). On the other hand, once a table is in a database, you can remove it with REMOVE TABLE, whether or not it's hidden.

Finding a hidden file isn't too hard. VFP's FILE() function can't see hidden files; it returns .F. when passed the name of a hidden file. However, ADIR(), which fills an array with information about files in a directory, has the ability to see hidden files. The optional third parameter lets you specify the file types to include. In this case, pass "H" to indicate hidden files. Assuming the file name and path you're interested in is stored in cFileName, this code determines whether the file exists:

```
nFileCount = ADIR(aFileList, cFileName, "H")
IF nFileCount = 1
    * the file exists
ENDIF
```

If you're interested in hidden directories as well, pass "HD". However, the attributes are combined with "or," not "and," so you'll get a list of all files and directories that match the filespec you pass, whether hidden or not. You need to look at the attributes in the fifth column of

the array to distinguish directories from files, and hidden items from visible ones.

Once you're sure there's a hidden file to delete, you have to step outside VFP to do it. FoxPro's two commands for deleting files, DELETE FILE and ERASE, are both unable to delete hidden files, generating a "File does not exist." message in the status bar. Be aware that these two commands do not raise an error if the specified file doesn't exist; they just don't do anything. On the other hand, the DELETE DATABASE command fires error 1 ("File does not exist") when you attempt to delete a hidden database.

So, if you can't use the native commands, what can you do? You have several choices. One possibility is to change the attributes of the file, so it's no longer hidden, then delete it. You can do that by using VFP's RUN command to run the DOS ATTRIB command, like this:

```
RUN /n ATTRIB -h &cFileName
```

The downside to this approach is that your users will see the DOS window flash.

Another alternative is to use the Windows Scripting Host. That approach allows you to check for the file, and to delete it directly. It also provides the ability to change file attributes. Here's code that determines whether the file exists and, if so, deletes it:

```
oWSH = CreateObject("Scripting.FileSystemObject")
IF oWSH.FileExists(cFileName)
    oWSH.DeleteFile(cFileName)
ENDIF
```

The only problem with this version is that some users may not have the Windows Scripting Host available. Because certain kinds of viruses exploit its abilities, some network administrators remove the WSH.

Fortunately, there's another approach. You can go directly to the Windows API and delete the file from there. The DeleteFile API function has the ability to delete hidden files. Here's the declaration:

```
DECLARE LONG DeleteFile IN Win32API STRING
```

The function returns 0 if there's a problem, so you can use it like this:

```
nResult = DeleteFile( cFileName )
IF nResult = 0
    * Deal with the problem
ENDIF
```

It's also worth noting that the API has functions that let you examine and change the attributes of a file. The functions are `GetFileAttributes` and `SetFileAttributes`, respectively. For more information on these functions, check out Christof's answer in the May, 1999 ADVISOR Answers column. You can also manipulate the attributes of a file using the Windows Scripting Host. The File object has an `Attributes` property. For details, see Pamela's answer in the May, 2001 column.

The bottom line is that, even when VFP can't do something natively, there are plenty of options for getting it done.

-Tamar