

May, 2000

## Advisor Answers

### Handling Long File Names

Q: I'm trying to read a text file from a user folder and append its content to a table. I'm using the GETFILE() function to read the file name. I encounter a problem when the source text file is located in a folder that has a space in its name (such as C:\PROGRAM FILES). However, if the folder the user selects consists of a long folder name (like C:\PROGRAMFILES), I don't encounter the problem.

Can you help me how to solve this problem?

-Maria Loretta C. Escobal (via Advisor.COM)

A: Personally, I think the decision to allow spaces in paths and file names was a poor one, but Microsoft didn't ask me, and since Microsoft itself uses spaces in some of the standard paths in Windows (like C:\PROGRAM FILES), our code must be able to deal with it.

A number of FoxPro commands require special handling when dealing with paths that include a space. Since you never know whether the path to a file will include a space, your best bet is to write all your code to assume that every path might include spaces.

When you refer to a file or path directly, the secret is to put quotes around the path, like this:

```
USE "C:\My Path With Spaces\MyTable"
```

In your case, however, you probably have code like:

```
cTextFile = GETFILE("TXT","File to append","Choose")
USE MyTable IN 0
APPEND FROM &cTextFile TYPE DELIMITED
```

In this situation, consider using a name expression rather than macro substitution by putting parentheses around the variable. So, change the last line to:

```
APPEND FROM (cTextFile) TYPE DELIMITED
```

This ensures that the APPEND FROM can handle whatever path and file name the user chooses.

On a related note, if you have tables and views in a VFP database that have spaces in their names, and open them without specifying an alias, FoxPro substitutes an underscore for each space. You can use `CursorGetProp()` to determine the `SourceName` property of the cursor to find out the original name of the view or table.

Here's an example, using the `TasTrade` database that comes with Visual FoxPro:

```
OPEN DATABASE _SAMPLES + "TasTrade\Data\TasTrade"
* Use quotes to handle the embedded space
USE "Category Listing"
* VFP substitutes an underscore
? ALIAS()   && Category_Listing
* But the SourceName property tells us the truth
? CursorGetProp("SourceName") && Category Listing
```

Why does this matter? Because you need the actual name of the table or view in functions like `DBGETPROP()` to inquire about the table or view and for other generic code. (Actually, even if you assign an alias when you open the table or view, you need the `SourceName` when writing generic code.)

While my overall inclination is to avoid spaces in directory and file names I create, I've learned that I can't prevent others from creating them on my machine, nor can I tell the people who use the applications I write not to use them. So the next best thing is to make sure that my applications can deal with them.

-Tamar