

November, 1998

Advisor Answers

Visual FoxPro 3.0, 5.0 and 6.0 and FoxPro 2.x

Q: In Visual FoxPro, is there a way for our phone operators to search for a name by means of phonetic sounds and have it respond and display the name on the screen?

When our office operators receive an inquiry about phone numbers or whereabouts of an individual, they type in the last name. A form displays the information once it is found. We want our operators to search names by sound. This is very useful, especially if the name is not too common or is unusual sounding.

Is there a function in VFP that will handle this function?

–Ben Alfon (via Advisor.Com)

A: As someone with both first and last names that are always misspelled or mispronounced, I'm sympathetic to your situation. In fact, the problem isn't just unusual or uncommon names; many names have multiple spellings (like "Roche" and "Roach"), or are so similar that it's hard to distinguish them on the phone (like "Cohen" and "Cohn").

The answer to your question comes in two parts. First, the FoxPro SOUNDEX() function was intended to do what you want. It takes a character string and returns a four-character string beginning with the same letter as the original string and followed by three digits. This result characterizes the original string by its sound. (It's based on a standard system for this, also known as "Soundex.") For example:

```
SOUNDEX("Smith")
```

returns:

```
S530
```

as does:

```
SOUNDEX("Schmidt")
```

However, the algorithm SOUNDEX() uses often isn't smart enough. The biggest problem, of course, is that the result always uses the first letter, but that's not the only issue. Consider this set of homonyms: "right," "rite," "write," and "wright." While I'd expect different results for the words beginning with "R" than for those beginning with "W," I'd also expect that each pair would return similar results. But look at what actually comes back:

Word	Soundex(word)
Right	R230
Rite	R300
write	W630
wright	W623

On the other hand, when all you change is the first letter, the numeric part seems to stay the same. For example:

```
SOUNDEX("night")
```

returns

```
N230
```

which is the same as SOUNDEX("right") after the letter. However, in my tests, I also noticed that both "bat" and "bite" result in B300, indicating similarity to "rite" after the first letter. In the part of the United States where I live, "bat" and "bite" sound quite different, but it is true that, with some regional accents, they may sound quite similar.

If you can live with these limitations, you can set up a query to pull up the matches with a WHERE clause like this:

```
WHERE SOUNDEX(cLastName) = SOUNDEX(cInputString)
```

You could then display this information in a list or grid and allow the user to choose the relevant record. If your table contains more than a few records, you should create an index tag based on SOUNDEX(cLastName) to optimize the query.

To be honest, though, I wouldn't depend on SOUNDEX() for the kind of work you're talking about. Instead, I would use a third-party library called PhDbase (Korenthal Associates, (212) 242-1790)). PhDbase offers what's called "fuzzy search" capability. In other words, it lets you find near matches. Unlike SOUNDEX(), it doesn't consider the first letter more significant than others. (PhDbase also provides rapid search of text data, such as memo fields.)

PhDbase is far too versatile for me to fully describe it here, so I'll just give you a sense of how it works and what it can do. PhDbase is a library, so before you can use it, you have to SET LIBRARY to the appropriate file. It comes with library files for FoxPro 2.x/DOS, FoxPro 2.x/Windows, VFP 3 and VFP 5/6. (I've used PhDbase with FoxPro 2.x/Mac as well, but I think it's packaged separately.)

To put PhDbase to work, you create a PhDbase index. Once you've done so, you can use it for fuzzy search, passing a string for which you want all exact and near matches. You

have a tremendous amount of control over the process, both in terms of what gets indexed and how a search operates.

I've used PhDBase for several projects over the years. As I mentioned above, it also provides speedy searches of textual data. I took advantage of that aspect for a client who needed a way to search hundreds of pages worth of information (stored in memo fields) to find documents that matched certain criteria.

-Tamar