

June, 2000

Advisor Answers

Creating New Sort Orders

Visual FoxPro 6.0, 5.0 and 3.0

Q: In Visual FoxPro you can specify a sort order with SET COLLATE TO cExpression. But it seems cExpression is a predefined expression. I want to define a new sort order for my own purposes and make new priorities for the characters. How can I set my own sort order in Visual FoxPro tables to one that is not previously defined?

–Mohammad Hamed Arab (via Advisor.COM)

A: As you note, SET COLLATE TO lets you specify the sort order used for indexing character strings and all string comparisons. However, the list of sort orders is fixed and, in fact, the number of options is pretty small, though it covers most Latin based languages.

The normal collating sequence, used by default, is "MACHINE". It's based on the ANSI values of the characters, that is the system's code page. Other collating sequences provided include "SPANISH", which handles the special "ch", "ñ" and "ll" characters in the Spanish character set, and "NORDAN" for Norwegian and Danish.

Be aware that using a collating sequence other than "MACHINE" has optimization consequences. FoxPro is faster with machine collation than with other collation sequences. (In addition, there are some known bugs involving other collation sequences. See the February '98 Ask ADVISOR column.)

But what do you do if you need to sort characters in an order that doesn't correspond to one of the collation sequences provided? There are a couple of FoxPro functions that can help you here, but the best candidate is SYS(15). Although Help says it's provided only for backward-compatibility, its original purpose in the language was to do exactly what you need – provide an index order other than the normal order.

SYS(15) takes a "translation table" and a string and returns the string as processed by the translation table. The translation table is a string of up to 256 characters, that is, one for each of the ANSI characters.

Each character in the original string is replaced by the corresponding character in the translation table to form the return value.

Let's look at an example. First, please note that this code is *not* the right way to deal with accented characters. But it is a good way to demonstrate the workings of SYS(15). The function TransTable (included on this month's Professional Resource CD) creates and returns a string that removes all accents. For every accented character, it substitutes the same character without an accent. For all other characters, it leaves the original character. Here's the code:

```
* TransTable.PRG
* Create a translation table for use with SYS(15).
* Accented characters are mapped into regular characters.
* Note that this is NOT the right way to handle accented
* characters.
* This code assumes that your system code page is 1252,
* the standard code page of Windows for most Western
* languages.
LOCAL cTransTable, nChar
cTransTable = ""
FOR nChar = 1 TO 191
    cTransTable = cTransTable + CHR(nChar)
ENDFOR
* Now accented A's
FOR nChar = 192 TO 198
    cTransTable = cTransTable + "A"
ENDFOR
* Now C with circumflex
cTransTable = cTransTable + "C"
* Now accented E's
FOR nChar = 200 TO 203
    cTransTable = cTransTable + "E"
ENDFOR
* Now accented I's
FOR nChar = 204 TO 207
    cTransTable = cTransTable + "I"
ENDFOR
* Now special D
cTransTable = cTransTable + "D"
* Now N with tilde
cTransTable = cTransTable + "N"
* Now accented O's
FOR nChar = 210 TO 214
    cTransTable = cTransTable + "O"
ENDFOR
* Next is regular character
cTransTable = cTransTable + CHR(215)
* Now another accented O
cTransTable = cTransTable + "O"
* Now accented U's
FOR nChar = 217 TO 220
    cTransTable = cTransTable + "U"
```

```

ENDFOR
* Now accented Y
cTransTable = cTransTable + "Y"
* Next two are regular characters
FOR nChar = 222 TO 223
    cTransTable = cTransTable + CHR(nChar)
ENDFOR
* Now accented a's
FOR nChar = 224 TO 230
    cTransTable = cTransTable + "a"
ENDFOR
* Now c with circumflex
cTransTable = cTransTable + "c"
* Now accented e's
FOR nChar = 232 TO 235
    cTransTable = cTransTable + "e"
ENDFOR
* Now accented i's
FOR nChar = 236 TO 239
    cTransTable = cTransTable + "i"
ENDFOR
* Now unusual d
cTransTable = cTransTable + "d"
* Now n with tilde
cTransTable = cTransTable + "n"
* Now accented o's
FOR nChar = 241 TO 246
    cTransTable = cTransTable + "o"
ENDFOR
* Next is regular character
cTransTable = cTransTable + CHR(247)
* Now another accented o
cTransTable = cTransTable + "o"
* Now accented u's
FOR nChar = 249 TO 252
    cTransTable = cTransTable + "u"
ENDFOR
* Now accented y
cTransTable = cTransTable + "y"
* Next is regular character
cTransTable = cTransTable + CHR(254)
* Now accented y
cTransTable = cTransTable + "y"
RETURN cTransTable

```

We can apply this translation table using SYS(15) to create a string containing no accented characters. For example, using the Products table from the example TasTrade database, you can remove accents from the Product_Name field like this:

```

USE _SAMPLES + "TasTrade\Data\Products"
LIST OFF NEXT 20 ALLTRIM(Product_name) + ;
    " becomes " + SYS(15, TransTable(), Product_Name)

```

To create an index without accents, use this expression as the index key:

```
INDEX ON SYS(15, TransTable(), Product_Name) TAG TransProd
```

The Products table already has an index tag (Product_Na) based on the Product_Name field. Figure 1 shows part of a Browse when order is set to Product_Na. Figure 2 shows a Browse of the same section with order set to the new TransProd tag. Notice the different position of "Pâté chinois" in the two lists.

Product_id	Supplier_id	Category_id	Product_name	English
8	3	2	Northwoods Cranberry Sauce	Northwoo
25	11	3	NuNuCa Nuß-Nougat-Creme	NuNuCa C
77	26	2	Original Frankfurter grüne Soße	Original F
70	7	1	Outback Lager	Outback L
16	7	3	Pavlova	Pavlova P
53	24	6	Perth Pasties	Perth Mec
55	25	6	Pâté chinois	Shepard's
11	5	4	Queso Cabrales	Cabrales
12	5	4	Queso Manchego La Pastora	Mancheg
59	28	4	Raclette Courdavault	Courdave
57	26	5	Ravioli Angelo	Angelo R

Figure 1 Ordinary Index—An index using the "Machine" collation sequence can produce an order that seems surprising. Notice the position of "Pâté chinois".

Product_id	Supplier_id	Category_id	Product_name	English
8	3	2	Northwoods Cranberry Sauce	Northwoo
25	11	3	NuNuCa Nuß-Nougat-Creme	NuNuCa C
77	26	2	Original Frankfurter grüne Soße	Original F
70	7	1	Outback Lager	Outback L
55	25	6	Pâté chinois	Shepard's
16	7	3	Pavlova	Pavlova P
53	24	6	Perth Pasties	Perth Mec
11	5	4	Queso Cabrales	Cabrales
12	5	4	Queso Manchego La Pastora	Mancheg
59	28	4	Raclette Courdavault	Courdave
57	26	5	Ravioli Angelo	Angelo R

Figure 2 Indexing with SYS(15)—Using SYS(15), you can index in any order you want. The index used here strips out all accents. Notice that "Pâté chinois" is now among the "P"s rather than at the end.

Keep in mind that the TransTable() function must be available whenever the table is opened and in use. You can guarantee this by making it a stored procedure, if the table is part of a database. In addition, there's some overhead involved in calling the function to

create the translation table string every time the index is used . In addition, the index is not available for Rushmore and therefore can't be used to query the table. The alternative is to create the string once and ensure that it's always available. Given the scope issues that could arise, I think the cost of using a function is a better choice unless performance becomes an issue. If it does, consider alternatives such as using a public variable or an application property to hold the translation table. The best alternative, if you can use it, is to use macro expansion to include the translation table directly in the index expression. Because of the nature of the translation table strings, that may not be an option in many situations. (It's not for the example here.)

Make sure to test your application extensively with different data sets, as there are many bugs in all versions of Visual FoxPro when SET COLLATE is changed from the default. Keep in mind that every COLLATE sequence except "MACHINE" uses 2 bytes to store a character. This doubles the size of the index file and limits the length of the index expression to 120 characters. Also SET COLLATE not only affects the index order, but any string comparisons performed by Visual FoxPro, implicitly and explicitly. If possible, leave COLLATE set to the default setting throughout the application and create an index in the proper COLLATE sequence only for display purposes, not to find data.

-Tamar