

August, 2004

Advisor Answers

Cleaning up strings

VFP 9/8/7/6

Q: I know I can use the TRIM() functions to remove spaces at the beginning or end of a string, but what about extra spaces in the middle? Also, what if the space at the beginning or end isn't the space character, but tabs or some other non-printing character?

A: You've asked two questions here and I'll tackle them separately. As you note, the various TRIM() functions remove leading and trailing blanks. Use LTRIM() for leading blanks, RTRIM() or TRIM() for trailing blanks, and ALLTRIM() for all leading and trailing blanks.

If you want to eliminate all spaces in a string, the CHRTRAN() function is your best bet. For example:

```
cString = " Leading, trailing and embedded blanks. "  
?CHRTRAN(cString," ","")  
  && Returns " Leading,trailingandembeddedblanks."
```

More likely, what you want within a string is to reduce a series of spaces to a single space. VFP doesn't offer a native function to do this, but the FoxTools library that comes with VFP includes a function called REDUCE(), which trims leading and trailing blanks, but reduces embedded blanks to a single space. Continuing the example above:

```
SET LIBRARY TO HOME() + "FoxTools.FLL"  
?REDUCE(cString)  
  && Returns "Leading, trailing and embedded blanks."
```

REDUCE() is actually more capable than this example indicates. It takes an optional second parameter that specifies what character is to be reduced. For example:

```
cSlashes = "///Lots/of///embedded//slashes//"  
?REDUCE(cSlashes, "/")  
  && Returns " Lots of embedded slashes"
```

Note that the leading slashes are reduced to a single space, but trailing slashes are removed. I'm not sure why the difference. Also, be aware that you can pass multiple characters for the second parameter. They're handled as a group, so that any sequence of occurrences of the listed characters is reduced to a single space. For example:

```
cString = "Slashes///\\\\//and \\ backslashes"
?REDUCE(cString,"\/ ")
  && Returns "Slashes and backslashes"
```

Note that once you specify the second parameter, spaces aren't considered in processing. If you want to reduce spaces as well, you need to add a space to the processing list.

In VFP 8 and earlier, REDUCE() also provides one answer to your second question, how to trim nonprinting characters like tabs and returns from a string. Simply include the appropriate characters, using the CHR() function to specify them, as part of the second parameter. For example:

```
cString = "My line " + CHR(9) + CHR(13) + CHR(10) + CHR(9)
?REDUCE(cString, " " + CHR(9) + CHR(13) + CHR(10))
  && Returns "My line"
```

The downside to this approach is that it removes nonprinting characters embedded in the string, as well. To avoid that, you have to use a loop. Here's a function (TrimAny.PRG on this month's PRD) that removes all specified characters from the beginning and end of a string. This is a barebones version; you'll want to add parameter checking and error handling for production.

```
LPARAMETERS cString, cRemove

LOCAL nPos, cOutString

cOutString = cString

* Remove trailing characters
nPos = LEN(cOutString)
DO WHILE nPos >= 1 AND SUBSTR(cOutString, nPos, 1) $ cRemove
  nPos = nPos - 1
ENDDO

cOutString = LEFT(cOutString, nPos)

* Remove leading characters
nPos = 1
DO WHILE nPos <= LEN(cOutString) LEFT(cOutString, nPos) $ cRemove
  nPos = nPos + 1
ENDDO

cOutString = SUBSTR(cOutString, nPos)

RETURN cOutString
```

To use the function, call it as you would REDUCE(), passing the string as the first parameter and the list of characters to be removed as the second:

```
cString = "My line " + CHR(9) + CHR(13) + CHR(10) + CHR(9)
?TrimAny(cString, " " + CHR(9) + CHR(13) + CHR(10))
  && Returns "My line"
```

VFP 9 offers an easier solution. The TRIM() functions have been enhanced to let you specify the characters to be removed. The new syntax is:

```
cResult = xxxTRIM( cString [, nFlags
                  [, cParseChar [, cParseChar [, ... ]]])
```

Substitute any of the TRIM() function for xxxTRIM(). For example,

```
cString = "My line " + CHR(9) + CHR(13) + CHR(10) + CHR(9)
?TRIM(cString, 1, " ", CHR(9), CHR(13), CHR(10))
  && Returns "My line"
```

The nFlags parameter currently represents a single flag that controls case-sensitivity. Pass 1 (as in the example) to make the consideration of the characters to be removed case-insensitive.

If you've never explored the FoxTools library, it's worth taking a look at. While many of the functions there have migrated to VFP itself over the years, there are still some useful items there. You'll find both the library and a Help file for it in the VFP home directory.

-Tamar