

November, 2003

## Advisor Answers

### Using SYS(2015) for an alias

VFP 8/7/6

Q: I have an application that's been running without problems for years. One morning recently, a CREATE TABLE command suddenly changed its behavior. The CREATE TABLE code looks like this:

```
cTable = SUBSTR(SYS(2015),3)
CREATE TABLE (cTable) (<field list>)
```

Previously, the new table would have the same alias as the table name. Now it's getting an alias of A or B, or some other letter. What's going on here?

A: The problem here isn't actually with CREATE TABLE, but with your use of SYS(2015). That function is one of two in VFP designed to return a (sort of) random character string. The other is SYS(3) (which I wrote about in this column way back in June, 1994).

SYS(3) returns an 8-character string of digits. It's handy when you need a filename. However, it's based on the system clock and won't necessarily return a unique value each time you call it.

SYS(2015) returns a 10-character string that always begins with an underscore ("\_"). It was added to the language in FoxPro 2.0 to provide names for procedures generated when forms designed in the Screen Builder were generated into PRG files. Although it's also based on the system clock, when you call it several times in succession, it always returns a unique value.

To confirm the behavior of the two functions, I wrote a short test program that calls each 10,000 times in a tight loop. (All that happens inside the loop is that the value generated is added to a cursor.) In multiple runs, SYS(2015) always produced 10,000 different values. However, SYS(3) never did so. The number of values depended on the machine. On an 800-mhz Pentium 3, the number of unique values was between 1100 and 1200, with no value appearing more than 10 times. (I tested with VFP 8, VFP 7 SP1, and VFP 6 SP5.) On a much slower machine (Pentium II/200mhz), I saw between 650 and 750 unique values for SYS(3), with no value appearing more than 16 times. (On that machine, I tested with VFP 8 and VFP 7 SP1.) My test program is

included on this month's Professional Resource CD as TestUniqueValues.PRG.

Clearly, if you really need a random string, SYS(2015) is a better choice. Because that's the case, many people have used it for random filenames since the time when filenames were limited to 8 characters plus an extension. SYS(2015) returns 10 characters and an examination of the return values makes it clear that it's the last characters that change from one call to the next. So, many Fox programmers are in the habit of using SUBSTR(SYS(2015),3) to get random filenames.

In fact, this practice is so common that the Help for SYS(3) includes this sentence:

"SYS(3) may return a non-unique name when issued successively (sic) on a fast computer. Use SUBSTR(SYS(2015), 3, 10) to create unique, legal eight character file name."

However, beginning in VFP 7, there's an additional warning (which really should be in the SYS(2015) topic, as well):

Avoid using this string in CREATE TABLE, CREATE CURSOR, SELECT INTO CURSOR, and other similar commands. Using this string generates an error because a valid FoxPro cursor or alias name cannot begin with a number.

Many people failed to heed this warning, and often it's not a problem. However, sometime in mid-July, that changed and SYS(2015) began returning values that start "\_10." This wasn't the first time SYS(2015) returned a digit for the third character, however, though it hadn't done so since late in 2000.

To check out the magnitude of the problem, I wrote a program that changes the system date one day at a time and calls SYS(2015) for each date. I ran it for dates from January 1, 1993 through December 26, 2012 and found that, of those 7300 dates, SYS(2015) had a digit in the third position on 2149 of them. (Since the return value depends on the millisecond, when I ran the program twice, I got different values back for each date. So your results may vary slightly.) My code is included on this month's PRD as Test2015.PRG. Be aware that while the code does reset the system clock to the date and time when the program was started, it will probably leave your system clock several minutes slow (as long as it takes the program to run). In addition, it might interfere with running applications that respond to the current

system time such as Microsoft Outlook or a task scheduler. In fact, I did my testing on a machine dedicated to testing, where I don't receive email or have any scheduled tasks. (The PRD also includes Christof's Struct program, which is needed to run Test2015.PRG. Be sure to put them both in the same directory. For more information on Struct, see the November '99 column.)

Why is this a problem in your scenario, since you're using the value for a table name, not an alias? When you issue CREATE TABLE, the new table is opened and, if possible, it's assigned an alias based on its name. However, you're now creating files whose name isn't valid as an alias, so VFP is using the default alias for the work area. (See Christof's answer in the July issue for more on default aliases.)

Other people are reporting actual crashes in applications that use SUBSTRING(SYS(2015),3) or RIGHT(SYS(2015),8).

How can you and the others solve the problem? That depends on the exact circumstances.

If you need a filename and want to be sure the alias is the same as the filename, you have two choices. The first is to use SYS(2015) in its entirety; after all, the 8.3 convention for filenames is long past. Since SYS(2015) always begins with "\_", you'll never have a problem.

If you need to stick with the 8.3 convention for some reason, another choice is use "\_" + RIGHT(SYS(2015),7). Again, the underscore ensures that the alias is the same as the filename. While the seven rightmost characters of the SYS(2015) will repeat sooner than the eight rightmost characters, in most situations, the files you're creating aren't kept forever, so you're not likely to have a problem.

In both cases, keep in mind, though, that SYS(2015) is unique only for a single instance of VFP, not across instances or across the machines in a network. If you have multiple machines or instances creating files at the same time, consider another approach. One possibility is to use some unique identifier of the computer itself as part of the filename. But for a truly unique string, you need both SYS(0) to identify the computer and \_VFP.ProcessID. An easier approach that guarantees an absolutely unique string is to use the API function CoCreateGuid to retrieve a GUID (globally unique ID). MakeGuid.PRG on this month's Professional Resource CD uses CoCreateGuid:

```
Function MakeGuid
  Local lcGUID, lcText
  lcGUID = Space(16)
```

```
lcText = ""
Declare Integer CoCreateGuid in OLE32.DLL String@
CoCreateGuid(@lcGUID)
For lnChar=1 to 16 step 4
    lcText = m.lcText + Right(Transform( ;
        CtoBin(Substr(m.lcGUID,lnChar,4)), "@0"),8)
Endfor
Return "G"+m.lcText
```

If you're generating an alias and not a filename, then SYS(2015) (or SYS(3)) is working way too hard. Aliases need to be unique only within a single data session. Since you generally know that you're creating them, most often, you can simply use a hard-coded value. If you may need multiple aliases without knowing how many ahead of time, consider using a fixed string suffixed with a few digits.

The same is true if you're creating a cursor (with CREATE CURSOR or SELECT ... INTO CURSOR). The name you give a cursor is actually an alias, not a filename.

This problem is a vivid example of why it's a bad idea to depend on undocumented behavior. Just because RIGHT(SYS(2015),8) often begins with a character, many developers assumed it always will.

-Tamar