September, 1998

Advisor Answers

Visual FoxPro 6.0, 5.0 and 3.0

Q: If I create a local view with more than 23 fields, and make it updateable, when I issue TableUpdate() for the view, I get error 1812–SQL Statement Too Long. Is there any way to make this work?

–Anonymous

A: This has been a problem since the first version of VFP. In VFP 3 and VFP 5, the only solutions involve changing the technique used to update the table. In VFP 6, however, you can modify VFP's behavior to make this work.

The problem occurs in VFP 3 and VFP 5 when a local view has more than 23 fields, and uses a SQL WHERE clause of "Key and updatable fields". The problem can also occur with a SQL WHERE clause of "Key and modified fields" if more than 23 fields change before an update. Apparently, the VFP SQL engine can't handle a SQL UPDATE or DELETE command whose WHERE clause contains more than 23 conditions.

There are a couple of ways to work around the problem, but neither of them is a particularly good choice. The first is obvious – make sure your views include 23 fields or fewer. That's not much help when the data you want to work with has more than 23 fields.

The second option is a little better, but also problematic. That's to set the SQL WHERE clause to "Key fields only". The problem with this approach is that you get no automatic conflict resolution, so you either have to take the chance of overwriting someone else's changes or handle it manually.

Fortunately, VFP 6 resolves this problem entirely. First, the number of fields that can be handled without any intervention has been raised to 40, so if your views involve 40 or fewer fields, you don't need to change anything.

Even better, Microsoft has put control of the situation into our hands. The new SYS(3055) function lets us control the complexity of SQL commands that can be processed.

The syntax is simple. Call SYS(3055) without a second parameter and it tells you the current setting. The default (and minimum) is 320. Pass a second parameter to change the allowable complexity. To permit a particular level of complexity in the WHERE clause, pass 8 times the number of expressions. So, for example, to handle a local view with 100 fields, call:

```
SYS(3055,800)
```

The maximum value you can pass is 2040 or 255*8. Since views are limited to 255 fields or less, any view you can create can be updated.

Even when you pass the second parameter, the value returned is the previous setting, so don't be fooled into thinking your change hasn't taken effect.

SYS(3055) is also supposed to be the cure for some of the other cases that trigger error 1812, as well as those that give error 1308 (Insufficient Stack Space). I haven't been able to test this because I haven't been able to write a query that gives either of those errors in VFP 5.

I hoped that this function would also lift some of the other limits on queries. Unfortunately, SYS(3055) does not let us go beyond 9 JOIN statements in a query. As for the limit of two subqueries per query, this one seems to have disappeared on its own. (In fact, I can put three subqueries in a query in VFP 5.0a. I didn't test versions older than that.)

At this point, you're probably wondering whether you should just issue SYS(3055,2040) when you start up and never worry about the problem again. Maybe. In my tests, I issued TableUpdate() against the same view (in fact, the same record) 1,000 times in a loop. I tried it for values of SYS(3055) running from the minimum for my view up to the maximum. The total time did increase with the value of SYS(3055) (though not in a straight line). However, the overall increase was pretty small, a little more than 3 seconds for the thousand passes on my 64MB machine.

I also checked SYS(1016) to see if more user memory is needed as SYS(3055) increases. That one's easy. No, it's not.

At this point, my instinct is to be cautious and set SYS(3055) as high as I need it, but no higher.

–Tamar