

Re: SQLCeCommandBuilder...IT'S A BUG...

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.compactframework/2007-06/msg000>

- *From:* "John Tamburo" <johntam@xxxxxxxx>
 - *Date:* Fri, 1 Jun 2007 22:19:09 -0500
-

Hi... It's a serious bug in the SQLCeDataAdapter and/or the SQLCeCommandBuilder.

In order to evade this bug, you must update the dataset with the exact same data adapter with which you fill.

This is a show-stopper for anyone who needs to make his or her code functional with both SQLCE and SQL Server 2005. I need to do that, and in order for this to work properly I need to have the data adapters kept isolated from the forms -- they're different types. I created classes to do this and do it well, with near zero duplicate code.

Now, in order to attempt to keep this horrific bug from ruining my program design (there is no sane reason for this requirement -- it's grossly lazy MS programming), I have to fill a NEW dataset with a select that can never return any rows and then copy the data rows from the passed-in data set's data table. BUT! Won't this blow up in my face as soon as I try to update a record? After all the rows in the fill dataset will be marked ADDED and will simply be inserted into the database.

I'm sunk. It's easier to fistfight a Great White Shark than to get MS to even admit a bug, much less fix it.

"Ginny Caughey [MVP]" <ginny.caughey.online@xxxxxxxxxxxxxxxx> wrote in message news:4A4259E5-999A-4B3E-94E6-52366FE7E5E5@xxxxxxxxxxxxxxxx

John,

Start by looking at the actual commands the command builder builds for you. For example, you can use `SqlCeCommandBuilder.GetInsertCommand.CommandText` to see what the insert command is.

--
Ginny

Re: SQLCeCommandBuilder...IT'S A BUG...

"John Tamburo" <johntam@xxxxxxxx> wrote in message
news:eLkE5KHpHHA.4212@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

vs 2005, vb, .net 2.0, using sql 2005 Compact edition, compiling for desktop use.

I retrieve a dataset with 0-1 rows of a table. Table is a single table with an identity column as primary key. Identity column is selected into the dataset. In this case we're retrieving 0 rows, and adding a row. Form populated with databound controls (textboxes only for this test). Bindings work perfectly.

I pass dataset to my object to post up the collected data. Data row 0 of table 0 shows perfectly, just as typed. Primary key (identity is on) is null. Code executes without exceptions, the update call you see below returns 1 for one row.

However, the only non-null column in the whole table is the primary key itself. Table name in the dataset matches the table name in the sql-ce database. I am stumped. The values are in the dataset. the rowstate is 4 (added). I see this when I watch the code, both before and after the update. HELLLP! There is no viable option to hand-code the commands and parameters.

I put in an event handler for rowupdated event of the adapter. the parameters show the correct column mappings but they are all value of system.dnull.

I am stumped and suspect that this is a MAJOR NASTY BUG in the commandbuilder or the dataadapter.

Here is the code:

```
Public Function PostDataSet( _  
ByRef dsResult As System.Data.DataSet, _  
ByRef iRet As Integer, _  
ByRef sErr As String) _  
As Boolean  
Dim dcConn As New SqlCeConnection(sConnect)  
Try  
dcConn.Open()  
Catch ex As Exception  
sErr = "Error in running PostDataSet, DB=" & sActiveDB & ",  
Message=" & ex.Message  
dcConn.Dispose()  
Return False  
End Try  
Dim cbcmd As SqlCeCommand = dcConn.CreateCommand()  
cbcmd.CommandText = "select * from " &  
dsResult.Tables(0).TableName  
Dim cb As New SqlCeCommandBuilder()
```

Re: SQLCeCommandBuilder...IT'S A BUG...

```
Dim daAdapter As New SqlCeDataAdapter(cbcmd)
cb.DataAdapter = daAdapter
cb.SetAllValues = True
With daAdapter
Try
iRet = .Update(dsResult, dsResult.Tables(0).TableName)
Catch ex As Exception
sErr = "Cannot update the " & dsResult.Tables(0).TableName
& " table, error=" & ex.Message
dcConn.Close()
dcConn.Dispose()
Return False
End Try
If iRet < 1 Then
sErr = "Nothing was updated."
dcConn.Close()
dcConn.Dispose()
Return False
End If
End With
dcConn.Close()
dcConn.Dispose()
sErr = ""
Return True
End Function
```