Denver Area Access Users Group

Next Meeting

Date: Thursday, July 19th, 2007

Time: 6:00 PM to 6:30 PM - Networking
6:30 PM to approx. 9:00 PM - Main Meeting

Location: **Microsoft USA - Rocky Mountain District**
7595 Technology Way
Suite 400 (4th Floor)
Denver, CO 80237
(driving directions)

DAAUG Website: www.DAAUG.org

This Month's Presentation

"Access Data Projects for Fun and Profit"

Presenter: Danny J. Lesandrini

Access Data Projects (ADPs) are more than just a mysterious file option in the Microsoft Access menu. Sometimes maligned, often misunderstood, ADPs provide a powerful alternative for creating a secure user interface to SQL Server data.

In addition to the basics, creating ADPs & connecting to data, Danny will show you how easy it is to programmatically load data in forms and controls using ADO recordsets from SQL Server tables, views and even stored procedures. He'll even demonstrate how to completely obfuscate the data entirely from the user, drawing from multiple SQL Server databases at once.

Our presenter, Danny J. Lesandrini, a Microsoft Certified Professional in Access, Visual Basic and SQL Server, has been programming with Microsoft development tools since 1995. He maintains a web site with free code samples at http://amazecreations.com/datafast and replies to all questions and comments sent to datafast@comcast.net.

Short Demonstration

Due to a change in this month's meeting format there will be no short demonstration.

Next Month's Topic (August 16th, *2007*)

To be announced.

If **you** are interested in presenting to the group, please contact Danny Lesandrini at vicepresident@daaug.org.

A Message from the Prez

"Fireworks"

This month's column is going to be short. Patrick used up most of this month's electron allotment on his VBA tip. ☺

I hope that everyone had a safe Fourth of July. Happy birthday USA! I hope you got to see some good fireworks.

Speaking of fireworks, we almost got some from Microsoft. Late on June 25th, they released the free Access 2007 runtime! ☺ The afternoon of June 26th, they pulled it. ☹ Go figure. It's been delayed forever and it was up for less than 24 hours. Arrrg... If they can ever get it released and keep it released, it will mark history. This will be the first version of Access to ever have a "free" runtime.

If there is good news, it's that the Access 2007 Developer Extensions are still available at http://www.microsoft.com/downloads/details.aspx?FamilyId=D96A8358-ECE4-4BEE-A844-F81856DCEB67&displaylang=en. There is information about what is included in the package at http://msdn2.microsoft.com/en-us/office/bb229700.aspx. Watch for a presentation on this package in the not-to-distant future.

Want a good laugh? Check out http://neopoleon.com/home/blogs/neo/archive/2003/09/29/5458.aspx. The title says it all! ☺

See you all next Thursday,

Your President,
M.L. "Sco" Scofield

New and Renewing Members

The Denver Area Access Users Group gives thanks to our new and renewing members for supporting our user group.

Ed Bade
Tom Breay
Phil Ferrero
David Joseph
Danny Lesandrini
Galen Moore
Alan Ruff
Alan Shrater

Last Month's Giveaway Winners

Night Lite - Phil Henke
Access Advisor Subscription - Dean Haze
Total Access Memo - George Mosteller
CD Cases - Duane Hanstein
Access Advisor Backpack - Tom Breay
Access Advisor T Shirt - Blue - Chris Nebinger
Access Advisor T Shirt - Cream -- J K August
Access Advisor T Shirt - White -- Ted Shinn
Visual Studio Mug - Barb Hagerman
Access 2007 Bible - Kevin Bell
SQL Server 2005 Starter Kit - Mark Evans
Beginning Transaction SQL - Alan Shrater
Sybex T Shirt - Dale Tweden
Sybex T Shirt - Patrick Headley
$20 Soft Pro Certificate - Ryan Stolcpart

Access/VBA Tip of the Month

"Quotes within Text Fields Break My VB Code"

I was helping to answer a question on an Access forum today and thought the solution I came up with would make a great Access Tip. The question was essentially this, "How do I insert text that has an apostrophe into a record using SQL Action Queries? When I try to do this I get Access Error 3075." The person asking the question implied that he was using an unbound form to gather the information for the record and then using the form's control values to assemble a SQL INSERT query to add the data to a table.

In my opinion, the correct answer for the question is to use a bound form for data entry. Besides avoiding the issue with quotes you have to write less code and a bound form will out perform an unbound form. I have actually tested the performance of both. However, there are exceptions to every rule and there are times when you cannot use a bound form, such as a database conversion procedure that has no associated form. So, let's move forward with solving the problem of quotes in strings used in action queries.

The problem occurs if you use VBA code to run an action query that updates fields from a VALUE list and the field delimiter you use also appears in the data. For example, the following line of code will cause an error because the apostrophe in the last name, O'Connor is interpreted by the database engine as the closing delimiter.

```
dbCurrent.Execute "INSERT INTO tblAddress (FirstName, LastName)
VALUES 'John', 'O'Connor'"
```

The same issue can occur if you delimit your variables with double-quotes in code and then you have a name like "Jimmy "Dog" James" in a text control that feeds into the action query.

The solution is to change the offending character to another character or character combination that the database engine won't interpret as a field delimiter. So, to fix the problem with the first example we can use one of the following forms:

```
dbCurrent.Execute "INSERT INTO tblAddress (FirstName, LastName)
VALUES 'John', 'O''Connor'"

or

dbCurrent.Execute "INSERT INTO tblAddress (FirstName, LastName)
VALUES 'John', 'O' & Chr(39) & 'Connor'"
```

In the first of these two examples the database engine "knows" to interpret two single-quotes that are adjacent to one another as a single quote. In the second sample, which is the way many developers prefer, the apostrophe is "inserted" into the text string through the Chr() function. Either of these methods work well if you are writing an INSERT query for a single use. But how do you get from the name "O'Connor" in a text control to a value like 'O''Connor' or 'O' & Chr(34) & 'Connor' in code so that it can be used in a dynamic SQL query? The answer is to use a function and the most logical function for this purpose is the Replace() function.

The Replace() function allows you to find a character or a string of characters within a string and replace it with one or more characters. The format is **Replace(*expression*, *find*, *replace*[, *start*[, *count*[, *compare*]]]).** We will focus on the first three parameters. The first parameter is the string that you want to work with. The second parameter is the character or characters to match. The third parameter is the character or characters that you want to use as the replacement. The function returns the modified string. If the match string isn't found, the function returns the

original string that was passed in through the ***expression*** parameter. Replace() can accept constants or variables.

Here is what the sample query would look like if were to get the values from controls on an unbound form and use the Replace() function to modify the single-quote characters.

```
dbCurrent.Execute "INSERT INTO tblAddress (FirstName, LastName)
VALUES '" & Replace(Me.TextFirstName.Value,  Chr(39), Chr(39) &
Chr(39)) & "', '" & Replace(Me.TextLastName.Value, Chr(39),
Chr(39) & Chr(39)) & "'"

Or

dbCurrent.Execute "INSERT INTO tblAddress (FirstName, LastName)
VALUES '" & Replace(Me.TextFirstName.Value,  """", """""" & "',
'" & Replace(Me.TextLastName.Value, """",  """""" & "'"
```

Now, most people would agree that it's more readable to use the first example over the second because it's hard to count all those double-quotes that are strung together. However, I would have to contend that all those Chr(39)s make for a lot of typing and I wouldn't want to do that much typing if I were storing more than just a couple fields, which has been the case on several occasions. So, to take this approach even further, I created a function of my own to help out. Please note that it would have been better for me to use the Replace() function within my function. To make it easier to view the commenting, copy the function into the VBA editor. In order to make the function actually work you will need to change the error handling to work with your own error handling procedures.

```
Public Function AddSpecialQuotes(ByVal strSource As Variant) As
Variant
         'Written by Patrick O. Headley, Linx Consulting, Inc.,
3/20/03.
         'Parses a string for a single double-quote character. If
any are found
         'they are turned into pairs of double-quotes so the
string can be used
         'as a value in a database field or as a criteria
expression, in code.

10       On Error GoTo AddSpecialQuotesError

Variables:
         Dim lngQuotePosition As String
         Dim strSourceCharacter As String

Start:
            'See if there is text to process.
20       If Not IsNull(strSource) Then

               'See if there are even any quotes in the string.
30          If InStr(strSource, Chr$(39)) > 0 Then
```

```
                  'Parse the source string.
40             Do Until Len(strSource) = 0

                     'See if the left most character in the
source string is a quote.
50                If Left(strSource, 1) = Chr$(39) Then

60                   AddSpecialQuotes = AddSpecialQuotes &
Chr$(39) & Chr$(39)
70                Else

                        'The character is a printable character
so copy it straight across to the target string.
80                   AddSpecialQuotes = AddSpecialQuotes &
Left(strSource, 1)
90                End If

                     'Remove the left most character from the
source string.
100               strSource = Right(strSource, Len(strSource) -
1)
110            Loop     'Check the next character
120         Else

                  'There are no quotes in the source string so
copy the entire string to the target.
130            AddSpecialQuotes = strSource
140         End If      'For seeing if there are any quotes in
the string.
150      Else

                'The source value is null so return a null.
160         AddSpecialQuotes = strSource
170      End If

Done:
180      Exit Function

AddSpecialQuotesError:
190      AddSpecialQuotes = ""
200      ErrorHandler "PublicFunctions",
"AddSpecialQuotesError:", Erl
210      Resume Done

End Function
```

Using the function shown above in our sample query the query would look like this:

```
dbCurrent.Execute "INSERT INTO tblAddress (FirstName, LastName)
VALUES """ & AddSpecialQuotes(Me.TextFirstName.Value) & """, """
& AddSpecialQuotes(Me.TextLastName.Value & """;"
```

This example shows my preference for using multiple double-quotes but even using double-quotes in the code the function is easier to read than the samples that precede

it. The next example is a line of real code from a real project that demonstrates the use of AddSpecialQuotes():

```
                        'Open the set of unformatted words and try
to match the parsed out string.
                        'Add special quotes as needed.
250                 rstString.Open "SELECT Word, ExceptOnFirstWord
" & _
                                "FROM luUnformattedWords " & _
                                "WHERE Word = """ &
AddSpecialQuotes(strParsedString) & """", _
                               Application.CodeProject.Connection,
adOpenStatic, adLockReadOnly
```

So, to sum things up, if you are using an Access form to gather data for a record, use a bound form. If that's not an option, use a function like Replace() to handle characters in your data that are considered special characters by the database engine. As shown in the sample, you may also create your own function to handle specific special characters. Whether or not you use Replace() or your own function, doing so will make your code more compact and easier to understand.

Have fun developing your applications,

Patrick Headley
DAAUG Communications Director

The New Denver Microsoft Visio User Group

The new Microsoft Visio user group is taking the summer off. There will be no meetings in July and August. When the meetings resume in September, they will be on the second Wednesday of the month at the Microsoft office. There will be a networking/gathering time from 6:00 PM to 6:30 PM with the meeting starting at 6:30 PM. Please see the Colorado Event Calendar Website for meeting dates.

Colorado Event Calendar Website

To find user group meetings and other programmer related events in the Denver metro area and throughout Colorado, be sure to check out www.MSColoradoEvents.com.

If you know of a user group that is not posting their meetings, please let them know about www.MSColoradoEvents.com and have them contact the administrator at administrator@mscoloradoevents.com.

Denver Visual Studio User Group

As a service to our sister group...

Monday, July 23rd, 2007 (4th Monday each month)
Networking & Refreshments 5:30 p.m., Meeting 6:00 - 9:00 p.m.

Location: Microsoft, Desert Mountain (Denver)

"Simple Design Patterns That Solve Real World Development Issues"

Presenter: Ben Hoelting

Learn the concepts behind some of the most useful design patterns to solve your real world development issues!

For our complete agenda and e-mail, join the Denver Visual Studio Users Group™ at www.DenverVisualStudio.net.

Member to Member

Do you have something that might be of interest to other DAAUG members? Have a service? Something to sell? Looking for something? Then this section is for you. And no, it **doesn't** need to be computer or Microsoft Access related. Send your listings to Communications@daaug.org with "DAAUG: Member to Member" in the subject line.

For Sale: **FMS Total Access Emailer Pro v10.5 (2002/XP)**

This powerful tool allows you to easily send single or bulk email messages directly from Microsoft Access, without having to go through Outlook. Although this version only allows development of blast email messages in Access 10 (2002/XP), the capability can be deployed in Access 9 (2000) applications. For more details, go to http://www.fmsinc.com/products/Emailer/index.asp. Retail value $499. $250 OBO. Contact Ed Bade: erb@opus-tech.com.

Access classes by M.L. "Sco" Scofield

Check out our online schedule at http://www.scobiz.com/ClassSchedule.asp. We offer classes at all levels from beginning to advanced. To find the right class for you, check out our web site, call Sco at 303-757-7768, or send an email to Training@ScoBiz.com.

Be sure to ask for the special DAAUG member discount.

Special! If you are in possession of an expired class certificate from a DAAUG drawing, Scofield Business Services will be offering a limited time exchange for a new certificate. Send an email to Training@ScoBiz.com with "DAAUG Certificate Exchange" in the subject line and we will contact you with the details.

Linx Consulting, Inc. would like to help your organization

Linx Consulting, Inc. offers database application development for small to large sized organizations. For more information on how we may serve you, please call Patrick Headley at (303) 916-5522 or visit our Website at www.linxco-inc.com. The entire site contains examples of what we can do for your database application or Website. Each click on the sample link displays a different sample. Please take a look.

My experience with Over-clocking

A couple years back I learned about over-clocking, which is where you take a system board, processor and memory and increase the clock speeds to gain more of a benefit from your purchase. Many video cards also have over-clocking capabilities. I like to build computer systems and experiment with electronics so over-clocking was something that I found interesting and figured that as long as I didn't take it to an extreme it may be worth the time.

I had an older machine with some basic over-clocking features but it didn't take well to over-clocking at all. So, when I purchased parts for my latest computer system, about a year ago, I specifically researched system boards for one that was known to have an abundance of over-clocking settings. The primary reason for the system purchase was to build a machine that could run multiple virtual PC machines and any other software that I threw at it simultaneously, without seeing a major performance hit. So over-clocking was secondary but I figured I'm spending the money so I may as well get a system that could be over-clocked.

I have a friend who likes to make his vehicles more powerful. He likes to repeat a quote by someone in the auto racing industry that goes "There is no replacement for cubic displacement". For those of you who are unfamiliar with the cubic displacement of an engine it basically means how big the engine is; how much air/fuel mixture can be drawn into the cylinders. In my experience this saying also applies to computers in a metaphorical way. While I was able to get about a 10% gain in both processor and memory speed with no significant increase in operating temperature it came with a price. The price was system stability and time wasted while rebooting and adjusting the settings. Even with the smallest increases in speed, in either the processor or memory, over the rated values the computer was unstable. Certain settings were better than others but no setting was completely stable. And

this wasn't just with games. In fact, I rarely use the system for games. My tests were with business software.

After much experimentation I've come to the conclusion that the settings a system board manufacturer gives you are useful for matching the system board with the processor and memory specifications; as prescribed by the manufacturers of those componenents but it's unlikely that you will be able to make your machine perform like a faster machine through manipulation of these settings. Maybe your experience has been different than mine. If so, send an email to communications@daaug.org and tell us how your over-clocking experience went. We will include your story in the Member-to-Member section of a future DAAUG newsletter.

Patrick Headley.

A Message from the Editor

"Presentations and Presenters"

We are always looking for speakers and topic ideas for our monthly meetings. To volunteer to be a speaker or to suggest a topic, see our Vice President, Danny Lesandrini (vicepresident@daaug.org) or President, Sco Scofield (president@daaug.org). You may also email one of them with "**DAAUG Speaker**" or "**DAAUG Topic**" in the subject line of your message.

This is your group and we need your support. You don't have to be a professional speaker to give a presentation. Your presentation can be about anything that's Microsoft Access related.

We are always looking for articles for our monthly newsletter. Send your articles to Communications@daaug.org with "DAAUG Article" in the subject line.

Please visit www.DAAUG.org for the latest news, consultant lists, officer email addresses and more.