

Re: Jeff C

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.formscoding/2004-03/2687.html>

From: Jeff Conrad (jeffc_at_ernstbrothers.com)

Date: 03/21/04

Date: Sat, 20 Mar 2004 19:12:14 -0800

Hi,

Was this post for me?????

> *It has been almost a year. How have you been doing? Hope
> all is well.*

I'm fine, thanks for asking!
Can't complain, family is doing well.
Finally getting some nice warmer weather....I digress...

> *I have a question.*

I have an answer; let's do lunch.

> *Until recently my project has work very well. But (and
> that always sneaks in) I made my application an FE/BE.
> Now I get an error 3219 and not sure how
> to fix it. See code below:*

Well shoot, there's the problem: you "fixed" something! ;-)

This sounds like maybe something I may have helped on in the past.

Looking at the code you no doubt are password protecting a single form using code from the following KB articles:

ACC97: Create a Password Protected Form or Report

<http://support.microsoft.com?id=179371>

ACC2000: How to Create a Password Protected Form or Report

<http://support.microsoft.com?id=209871>

Correct?

I'm further guessing that I must have sent you one of my samples that takes this to the next level by not using the

Re: Jeff C

InputBox as demonstrated in the articles. We probably made a special password form that would mask the password as it is typed and then open or not open the form depending upon a correct password. I have several types of password sample databases so I'm not exactly sure what I sent. Am I still on track?

> *Any suggestions?*

Yep. You actually have many options available to you.

1. I'm assuming you are well aware of the limitations of a "home-grown" security system. Any person with some Access experience could circumvent this in a heartbeat. Implementing User Level Security is the **best** way to properly secure an Access database. For the sake of simplicity I'll assume you really don't need that level of sophistication.

2. As you noticed the code now coughs up a hairball since you split the database. As Peter already mentioned, this code will now fail. The problem stems from the fact that you cannot use the Seek function on a linked table. For more information see the following KB article:

ACC2000: Cannot Perform OpenTable Method on Linked/Attached Table

<http://support.microsoft.com?id=208379>

So the EASIEST solution is to just move the tblPassword back into the front end and problem solved!

3. OK, maybe you want to have the tblPassword reside in the back end along with the other tables. It is possible to make this work, but it will require more work. Before we start, make a BACK-UP copy (or several) of your database before beginning.

– We don't need to change anything on frmPassword (I think). I'm going off a lot of assumptions since I can't see your database.

– Open the form you want to protect and go to the code window.

– Way up at the top in the Declarations area add this one line of code:

```
Private NoOpen As Boolean
```

So the first three lines of your code should now look like this:

```
Option Compare Database
Option Explicit
```

```
Private NoOpen As Boolean
```

– I used the following KB article for a new procedure:

ACC2000: How to Use the Seek Method on Linked Tables
<http://support.microsoft.com?id=210266>

– Using that as a guide I made the following changes to the Form_Open code and added a new procedure. Change your Form_Open code to this:

```
Private Sub Form_Open(Cancel As Integer)
On Error GoTo ErrorPoint
```

```
Dim Hold As Variant
```

```
    'Check to see if the user is passing in the Password.
    DoCmd.OpenForm "frmPassword", acNormal, , , acDialog
    Hold = MyPassword
    SeekAttachedTable "tblPassword", "KeyCode", _
    KeyCode(CStr(Hold))
```

```
    If NoOpen = False Then
        Cancel = True
    End If
```

```
ExitPoint:
Exit Sub
```

```
ErrorPoint:
MsgBox "The following error has occurred:" _
& vbNewLine & "Error Number: " & Err.Number _
& vbNewLine & "Error Description: " & Err.Description _
, vbExclamation, "Unexpected Error"
Resume ExitPoint
```

```
End Sub
```

– Now we need to add a new procedure so go down to the bottom of the code window and copy/paste this code in:

```
Private Sub SeekAttachedTable(Tablename, Indexname, _
SearchValue)
On Error GoTo ErrorPoint
```

```
Dim db As DAO.Database
Dim t As DAO.TableDef
Dim rs As DAO.Recordset
Dim dbpath As String
Dim SourceTable As String
```

```
Set db = DBEngine(0)(0)
dbpath = Mid(db(TableName).Connect, InStr(1, _
db(TableName).Connect, "=") + 1)
If dbpath = "" Then GoTo ExitPoint
```

```
SourceTable = db(TableName).SourceTableName
```

```
Set db = DBEngine(0).OpenDatabase(dbpath)
Set rs = db.OpenRecordset(SourceTable, DB_OPEN_TABLE)
rs.Index = Indexname
rs.Seek "=", SearchValue
```

```
If Not rs.NoMatch Then
    NoOpen = True
Else
    MsgBox "The password you have entered " _
    & "is incorrect." & vbNewLine & _
    "Please try again.", vbExclamation, _
    "Incorrect Password"
End If
```

```
ExitPoint:
    rs.Close
    db.Close
    Set rs = Nothing
    Set db = Nothing
    Exit Sub
```

```
ErrorPoint:
    MsgBox "The following error has occurred:" _
    & vbNewLine & "Error Number: " & Err.Number _
    & vbNewLine & "Error Description: " & Err.Description _
    , vbExclamation, "Unexpected Error"
    Resume ExitPoint
```

```
End Sub
```

– Now compile the code to make sure there are no errors.

– Close and save the form and then test. When you try and open this form, the frmPassword should pop up on screen first before you even see the regular form. You enter the password in the text box on that form and hit the OK button. The frmPassword will then close. If you have successfully entered a correct password the regular form

microsoft.public.access.formscoding: Re: Jeff C

should now open. If an incorrect password is entered you will see the message box and the form will not open. Works in my test.

> *Thank you for your help.*

You're very welcome (assuming this was even for me!)
Come back any time!

Jeff Conrad
Bend, Oregon