

Re: access 2003

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.conversion/2006-08/msg00020.html>

- *From:* doglover <doglover@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 27 Jul 2006 19:50:02 -0700
-

Geoff:

Just completed reading your e-mail and since it is late here I am going to defer my response until Saturday.

I did want to share this information with you. I disassembled the form piece by piece and ended up with a form that had:
form header
detail
properties for the form set to continuous
form query source (same as before less parameters)

I removed the parameters from the form query source. Removed both combo boxes from the form header, events, code, etc and ran the form query source and it worked fine. I opened the form in form view and it produced records in the detail section just like a continuous form should. Obviously, it did not synchronize combo boxes to the detail section or the parameter form query source because I had removed these items.

I placed the combo boxes back on the form header and the form stopped working like a continuous form even though the properties of the form indicated continuous.

I also believe there is a problem with the parameter in at least one of the forms queries and the SQL because syntax of the SQL will change randomly. What I mean is the form works and still produces the same problem we have been discussing but an additional error message shows up indicating that a syntax problem exist. If you look at the design view of the query things look ok but if you flip over to SQL view you notice that additional brackets have been placed around "forms" and "ChooseCust" in the PARAMETERS LINE OF THE SQL VIEW. Remove the extra brackets and the query runs with no error message. Specifically, the query with the parameters that start off "LIKE nz" which is the parameter placed in the query row source of the combo box ChooseJob.

I realize this has been incredibly time consuming for you and I really do appreciate your help. I decided yesterday to go on line with Microsoft to try and help the situation. My code was confirmed to be valid and they

Re: access 2003

really do not understand what is creating this problem.

The information about the detail section I mentioned earlier was made available to Microsoft today and their response will be interesting. I will share with you their results because you have more than gracious with your time and knowledge.

Regards,
doglover

"doglover" wrote:

Sorry the reply was so long I was trying to explain how the code actually appears. I believe you are correct the problem is behind the queries (most likely the criteria) in the combo boxes.

When I said predefined I really meant functions, operators, etc.

Will post when I get more feed back form Microsoft.

Doglover
"Geoff" wrote:

This is a very long reply!
I could not finish it yesterday.
I have tried to comprehensively answer your questions.

I would focus on the queries behind the combo boxes. Examine in detail why they list invalid customers or jobs or both. I think they are passing invalid criteria to the query behind the form.

I'll be working flat out next week, so if you have any further questions, post back soon.

Sorry, this hasn't been simple.
Geoff

1.)
The code behind the After_Update event on ChooseCust
combo box:
Private Sub ChooseCust_AfterUpdate()
On Error Resume Next
Me!ChooseJob = Null
Me!ChooseJob.Requery

Re: access 2003

Re: access 2003

End Sub

Tell me exactly where you want the code placed in the
ChooseJob combo box
and the ChooseCust combo box.

I just want to make sure that I am following your instructions
correctly.

Apologies – the code is already there; ie:

```
Me!ChooseJob = Null  
Me!ChooseJob.Requery
```

One of your first posts said so. I missed it!

2.) Could the code be missing that tells the expressions
placed in form
frmInvoiceSelect and the associated queries be missing. I
assumed
that an access built in function directs the expressions to fire.
I was
wondering if perhaps a built-in function could be missing or
a predefined
property?
Just a thought.

As you imported the form "frmInvoiceSelect" and the associated queries
from
the Access 97 database, I wouldn't have thought any expressions would be
missing.

So far No missing modules found in the program database or
the database
that stores data.

I intended you search the Access 97 database and I believe you've done that now. That's where the hidden modules might be and where that elusive "zfSortLabel" function might be.

If you find hidden modules in the Access 97 database, then you need to import them into your Access 2003 Program database.

LIBRARY DATABASE

If the Access 97 does not contain any hidden modules, then the "zfSortLabel" function must exist somewhere else. Perhaps the Access 97 database had a reference to a library database. To check this follow these steps:

1. Open the Access 97 database.
2. Open the VBA editor.
3. Open the Tools, References menu.
4. Examine the References dialog.
5. Is anything checked that might be another Access database or even a customised DLL file?

It seems to be me that the expressions in the queries are not communicating with the combo boxes on the form.

I don't think that's right. The tests you've done so far have demonstrated that the queries do communicate with the combo boxes.

Let me know if you agree or disagree with the following summary:

1. Essentially, there are three queries in contention. These are:
 - (a) The query that selects rows in the ChoostCust combo box.
 - (b) The query that selects rows in the ChooseJob combo box.
 - (c) The query that selects records for the form "frmInvoiceSelect".
2. Query (b) relies on the ChooseCust combo box.
3. Query (c) relies on both the ChooseCust and ChooseJob combo boxes.
4. When you select a customer and a job in the two combo boxes, sometimes records will be listed in the Detail Section and sometimes records will not be listed.

Re: access 2003

(a) When records are listed, it is possible to create a PreBilling report and records will be updated correctly.

(b) When records are not listed, this indicates that something is wrong, because the queries should function so that records are always listed in the Detail Section.

Q: Is that right?

If there was a problem with the relationship then the queries would not retrieve the correct data, right?

The test you did earlier proved that the relationship between the combo boxes and the "qryworkorders_uninvoiced" is working. In the test, you selected a customer and job number, got some records listed in the Detail Section, and produced a PreBilling report. Right?

It seems that, sometimes there are records in the Detail Section and sometimes there aren't. Therefore, I think that, when there are no records in the Detail Section, the two combo boxes are pumping invalid criteria into "qryworkorders_uninvoiced" that result in no records.

This makes me think that the two combo boxes are listing customers, or job numbers, or both, that are invalid. In other words, there might be something wrong with queries "qlbCustomerOpenInvoice" and "qlbJobByCustomer_frmInvoice" (which act as the row sources for the combo boxes). The criteria in these queries might be wrong. They need very careful examination.

I would suggest you find a customer and job number that result in no records in the Detail Section and then examine in minute detail why this customer and job number are being listed in the combo boxes. Clearly, they should not be listed in the combo boxes.

```
Private Sub Form_Open(Cancel As Integer)
Me.Filter = "[wrkid]=0"
Me.FilterOn = True
End Sub
```

Re: access 2003

The above code suggests that, when the form opens, a form filter is used to prevent records from showing in the Detail Section. The code in the After_Update event of the ChooseJob combo box (see next two code blocks) removes the form's filter (Me.FilterOn = False) and requeries the form (Me.Requery), so that records are (or should be) shown in the Detail Section for the selected customer and job number:

```
Private Sub ChooseCust_AfterUpdate()  
On Error Resume Next  
Me!ChooseJob = Null  
Me!ChooseJob.Requery  
End Sub
```

```
Private Sub ChooseJob_AfterUpdate()  
On Error Resume Next  
Me.FilterOn = False  
Me.Requery  
End Sub
```

It seems that one problem with the above code is that the user could select a job number when no customer has been selected.

If you want to prevent this from happening then the solution is:

- (1) to disable the ChooseJob combo box when the form opens and
- (2) enable it after a customer has been selected, as follows:

(1) The Form_Open event procedure needs to include an extra line of code as follows:

```
Private Sub Form_Open(Cancel As Integer)  
Me.Filter = "[wrkid]=0"  
Me.FilterOn = True  
Me.ChooseJob.Enabled = False  
End Sub
```

(2) The ChooseCust_AfterUpdate event procedure needs an extra line of code as follows:

```
Private Sub ChooseCust_AfterUpdate()  
On Error Resume Next  
Me.ChooseJob.Enabled = True  
Me.ChooseJob = Null  
Me.ChooseJob.Requery  
Me.Requery  
End Sub
```

2.) Could the code be missing that tells the expressions placed in form frmInvoiceSelect and the associated queries be missing. I assumed that an access built in function directs the expressions to fire. I was wondering if perhaps a built-in function could be missing or a predefined property? Just a thought.

CHECK EVENTS ARE FIRING

Check that you see [Event Procedure] in the appropriate property of the property sheet for each object that has an event. For example, you have code for a Form_Open event. Therefore, check the property sheet for the form and ensure that the OnOpen property has [Event Procedure].

2.) ...
So I changed the view to SQL and the code was:
PARAMETERS
[[forms]]!frmInvoiceSelect![[ChooseCust]] Text (255);

It seems odd that there were two square brackets before and after [[Forms]] and [[ChooseCust]] in the Parameters clause. Perhaps this happened when the database was converted.

I think the square brackets are only absolutely necessary when the parameter contains a space. I'm in the habit of including them anyway.

I am wondering if there is a problem deeper than just the forms but the code source that provides instructions to the expressions, etc.

Re: access 2003

The code source could be library database or a DLL (dynamic link library) file. As mentioned above, you need to open the Access 97 database and check References in the VBA editor to determine if there is another place where code could be stored.

If you recall I indicated that the expression in the on click event is different from version 97 database.

version 2003: =zfSortLabel("tktno",Screen.ActiveForm)
version 97: =zfSortLabel("tktno",[Screen].[ActiveForm])

If you type in the missing brackets on the zfSortLabel, etc for version 2003 and close the screen the replaed bracket go away but no error message ever appears.

I don't think the brackets are needed (see following test).

I created a dummy zfSortLabel function in a standard module. As an experiment, you could do the same. Copy and paste the following code into a standard module:

```
Public Function zfSortLabel(Param1 As Variant, Param2 As Form)
MsgBox "The active form in the active screen is:" _
& vbNewLine & vbNewLine _
& CStr(Param2.Name)
End Function
```

Notice Param2 is defined "As Form", which means the function expects a form object to be passed to it as Param2.

Now, open form "frmInvoiceSelect" and click one of your five labels whose OnClick property is set to =zfSortLabel("sometext", Screen.ActiveForm). The above function should now run and display a message box saying that the active form is "frmInvoiceSelect". If the above function does not run, double-check that the OnClick property for the label contains [Event Procedure].

CONCLUSION:

Re: access 2003

If the above dummy function works, then you have demonstrated that your labels can successfully call the zfSortLabel function. The fact that the labels don't call the real zfSortLabel function is odd. Either the function can't be found, or does nothing, or [Event Procedure] is missing from the OnClick property.

Delete the dummy function (or rem it out).

The database I am working with uses the execution file from a retail version of access 2003 but I noticed under the option, Default File Format "Access 2000" is selected rather than the other option of "Access 2002–2003."

Unfortunately, I'm not an Access 2003 user. However, the above information suggests that Access 2003 is set to create files in Access 2000 format. But this is probably irrelevant. You started by using Access 2003 to create a blank database. This would (presumably) have been in Access 2000 file format. You then imported database objects from the Access 97 version of the database. So, Access 2003 is now working with an Access 2000 file–format database. If Access 2003 isn't complaining, I don't see that as a problem. It should be just like upgrading a Word 97 document to Word 2000 and then