

Re: Passing Form input to stored procedure to populate a report

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.adp.sqlserver/2006-06/msg00145.html>

- *From:* "Marish" <marish01@xxxxxxxx>
 - *Date:* 19 Jun 2006 13:38:40 -0700
-

ZOMG! You answered that in 20 minutes. Makes me feel like the old Taylor song...10 miles behind me and 10,000 more to go. Thanks very much.

AkAlan via AccessMonster.com wrote:

This is how I pass parameters to a function, it's a rather lengthy process and you need to read it through a couple of times to really get the bigger picture but it really works well.

1st, create a function (I will call this fn_INV_Lookups) with the parameters you need. The one I give in the example uses 7 parameters.

It is important to have default values for all the parameters. Don't worry about it for now, I will tell you some things about making the parameter defaults later on.

Next, use this as your recordsource during development.

```
SELECT * FROM  
fn_INV_Lookups(DEFAULT,DEFAULT,DEFAULT,DEFAULT,DEFAULT,DEFAULT,  
DEFAULT)
```

Having the word Default in the function allows you to design the report or form without getting the little green marks in the text fields.

You are going to collect all the parameter values from a command button that opens the form or report and pass them to an array variable, so first lets do that. We will get to passing the parameter values later.

Create a module (I called mine modGlobals).

In it define a global variable , this is where you will pass the parameter values to

Re: Passing Form input to stored procedure to populate a report

```
Public astrParams(10) As String
```

This is an array type variable that will hold 10 string values numbered 0 through 9.

Then create a procedure to parse these parameters (the ones we will pass from the Report/Form Open procedure, we will get to that later)

```
Public Function GetParams(numParams As Integer, astrParams() As String) As String
```

```
' this function pulls a number of strings from astrParams and compiles them into a Parameter list
```

```
' suitable for passing to SQL Server in a sp or function call
```

```
Dim strParams As String
```

```
Dim i As Integer
```

```
strParams = ""
```

```
For i = 0 To (numParams - 1)
```

```
strParams = strParams & astrParams(i) & ","
```

```
Next
```

```
' strip off last comma
```

```
strParams = Left(strParams, Len(strParams) - 1)
```

```
GetParams = strParams
```

```
End Function
```

Ok so now we need to populate the parameter values into the astrParams() variable. This will go in the On Click event of the command button you use to open the form or report. It is rather long since I use 7 parameters but it will give you some ideas on how this can be used. Notice how we reference the astrParams by place holder as we populate it. You need to know the order of parameters the function is expecting them in . One way to know for sure is to simply open the function from the query list. You will be prompted to input the parameters in the order in which you created them and therefore the order it is expecting you to pass them in.

```
On Error GoTo Err_btnReport_Click
```

```
Dim stDocName As String
```

```
Dim strFilter As String
```

```
If chkSite Then
```

```
astrParams(0) = "" & cboSite.Value & ""
```

Re: Passing Form input to stored procedure to populate a report

```
Else
astrParams(0) = "% "
End If

If chkShop Then
astrParams(1) = "" & cboShop.Value & ""
Else
astrParams(1) = "% "
End If

If chkCat Then
astrParams(2) = "" & cboCat.Value & ""
Else
astrParams(2) = "% "
End If

If chkSOS Then
If (cboSOS.Value <> "SP") And (cboSOS.Value <> "BS") Then
astrParams(3) = "" & cboSOS.Value & ""
astrParams(4) = "0"
astrParams(5) = "0"
Else
If cboSOS.Value = "SP" Then
astrParams(3) = "% "
astrParams(4) = "1"
astrParams(5) = "null"
Else
astrParams(3) = "% "
astrParams(4) = "null"
astrParams(5) = "1"
End If
End If
Else
astrParams(3) = "% "
astrParams(4) = "null"
astrParams(5) = "null"
End If

If frmReport.Value = 5 Then
astrParams(6) = "P%"
Else
astrParams(6) = "% "
End If

Select Case frmReport.Value
Case 1
stDocName = "rptINVLlisting"
Case 2
stDocName = "rptINVReconciliationSheets"
Case 3
stDocName = "rptBinLabels"
```

Re: Passing Form input to stored procedure to populate a report

```
Case 4
stDocName = "rptYellowTags"
Case 5
stDocName = "rptBinStatusCards"
End Select
```

```
DoCmd.OpenReport stDocName, acPreview
```

```
End Sub
```

Now for the last task. In the form/reprot On Open event we need to create a way of getting the parameter values from the asrtParams() variable. Here is how we do that.

Create a On_Open event procedure:

```
Private Sub Report_Open(Cancel As Integer)

Dim strFilter As String

strFilter = GetParams(7, astrParams)

Me.RecordSource = "SELECT * FROM fn_INV_Lookups(" & strFilter & ") fn"

End Sub
```

Ok if you are still with me I will give you a break down of the sequence of events.

So when the user clicks the report button...

- 1) The form with the command button(FormA) calling the Form/Report looks at what options the user has selected on the form and fills the parameter array appropriately.
- 2) FormA calls the report (no filter needed)
- 3) The form/report (in On_Open event) processes the global parameter array and creates its own parameter list.
- 4) The form/report changes its record source property to reflect user criteria.
- 5) The form/report opens

More thoughts. Once you are finished designing the report , leave the recordsource empty, as you can see it will have a recordsource once it opens. I found that it will first grab all the default records then refilter in the on open event.

As for function defaults, that can be a little tricky. If you are dealing with a string type, you need to put Like @Param1 in the criteria block, not =@Param1 like with an integer. Then go to the function properties and put in '%' which is a wild character for All.

Re: Passing Form input to stored procedure to populate a report

There are other issues with defaults for Bit types and if you get that far just repost here and I'll help you with that.

Hope this helps.

marish01@xxxxxxxx wrote:

I know this has been asked before but I can't find an answer I understand. I am new to ADP and in many ways wished I had just linked tables in an MDB. Never the less, the power and flexibility for other front ends when all is done has kept me pushing forward. Microsoft provided an informative article 235359 discussing passing a form selection to a second display form but I can't separate the chaff from the wheat. It appears obvious that we have to pass the form value to SQL somehow so it knows what to crunch on, but my research has shown several possibilites and now I'm all flummoxed.

So, I have a table RAMain. I would like to return data in that table that is filtered according to a form (frmCSRSelect) where the user selects a value from the field "CSR" (CSR is a field in RAMain). Then I'll populate a report based on the stored procedure or function or view or whatever. (Try figuring out the difference between those without a peer group or in MS help)

FBGBF (Fairly Bright Guy with Brain Freeze) thanks you in advance.

Geoffrey Clark
IT Manager
ProTeam, Inc.
marish01(delete me)@gmail.com

—
Message posted via AccessMonster.com
<http://www.accessmonster.com/Uwe/Forums.aspx/access-adp-sqlserver/200606/1>