A Matter of Form

Access Forms to make Reporting a Snap (or a Click)

- Session #45
- Technical Track

<table>
<thead>
<tr>
<th>Alan Manifold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems Implementation Manager</td>
</tr>
<tr>
<td>Purdue University Libraries</td>
</tr>
<tr>
<td><a href="mailto:manifold@purdue.edu">manifold@purdue.edu</a></td>
</tr>
</tbody>
</table>

VUGM Endeavor User 2003
You can use Access forms to create an interface to your reports and queries. You can add:

- Buttons to initiate reports and queries
- Combo Boxes, List Boxes and Option Groups to facilitate selecting criteria
Form Examples

Simple control panel

Control panel with options

Record selector
Forms appear to have been designed primarily for data entry and display, not for running reports, so the wizards aren’t very helpful in creating Control Panel type forms.

Go straight to Design View
Filling Your Form

A new, blank form is about as empty as a thing can be. Use tools from the toolbox to fill it.

Label
Option Group
Combo Box
List Box
Button
Event Procedures

Behind every successful button is an Event Procedure. This is an expression, macro or Visual Basic code that is executed when you click.

```
Private Sub Command0_Click()
On Error GoTo Err_Command0_Click
Screen.PreviousControl.SetFocus
DoCmd.FindNext

Exit_Command0_Click:
    Exit Sub

Err_Command0_Click:
    MsgBox Err.Description
    Resume Exit_Command0_Click
End Sub
```
Wizard as Teacher

By trying the various actions in the wizard, you can learn how actions are coded in Visual Basic.
Looking at Code

Get to Visual Basic by clicking the Code icon in the toolbar.
You can start from what the wizard creates and add a bit at a time until it does what you want.

We’ll look at more VB details later.
Choosing Command Button from the toolbar walks you through the Command Button Wizard.

Buttons are good for initiating actions.
Combo Boxes

Combo Boxes let you select items from a list.

There is a Combo Box Wizard that helps you set up the value list.
The “Row Source” property of a Combo Box is a query that you can change. If you click on the ellipsis, it shows you the query and you can alter it as you wish.
List Boxes are similar in concept to Combo Boxes, but I find them more restrictive and usually use Combo Boxes instead. List Boxes have their own wizard.
Option Groups provide groups of mutually exclusive choices. These are great for overall selections that may determine what parameters are necessary.

There is a wizard for Option Groups, too.
There are several clunky ways to pass parameters from forms to queries and reports:

1. Start the query and let it get its own parameters.
2. Make the query get data from the (open) form.
3. Save the form input to a table, and link the table into the query.

```vba
Private Sub Form_Close()
    Dim DB As Database
    Dim PathTBL As TableDef
    Dim PathRST As Recordset
    DeleteTable "htmlpath"
    Set DB = CurrentDb
    Set PathTBL = DB.CreateTableDef("htmlpath")
    PathTBL.Fields.Append PathTBL.CreateField("path", dbText)
    DB.TableDefs.Append PathTBL
    Set Pathrst = PathTBL.OpenRecordset(dbOpenTable)
    PathRST.AddNew
    PathRST![path] = Me!Text1.Value
    PathRST.Update
    PathRST.Close
End Sub
```
A Better Way

Dim DB As Database
Dim PatQDf As QueryDef

Set DB = CurrentDb
Set PatQDf = DB.QueryDefs("Patron info")
PatQDf.Parameters("Barcode") = txtBarcode
PatQDf.Execute

You can call a query from Visual Basic after setting its "Parameters" property.

1. Declare Database and QueryDef type variables with "Dim"
2. Set the Database variable to the predefined value, "CurrentDb"
3. Set the QueryDef variable to the desired query by name from the QueryDefs collection of the Database
4. Set any needed query parameters from fields in the form with the "Parameters" property of the QueryDef
5. Then you can run your query; this one uses the "Execute" method, because it is a Make Table Query
Visual Basic’s object-orientation means that objects (such as databases, queries, combo boxes and buttons) have properties that can be read or set and methods that can be applied.

Object properties:

```vbscript
DB.QueryDefs("PO Minimal")
MinQuery.Parameters("[PO Number?]")
MinRecs.EOF
```

Object methods:

```vbscript
DB.CreateTableDef("PO_ID Table")
PPHRecs.AddNew
MinRecs.Close
```

Combinations:

```vbscript
POIDTbl.Fields.Append
DB.TableDefs.Delete
```
More on Methods

Methods often are part of more complex code

The Append method needs a field, and Createfield needs a name and a type:

```vba
POTbl.Fields.Append POTbl.CreateField("PO_ID", dbText)
```

OpenRecordset needs a mode parameter and produces a value that can be assigned to a Recordset type variable:

```vba
Set BibRST = BibQry.OpenRecordset(dbOpenForwardOnly)
```

Here the parentheses specify one member of the QueryDefs collection, which is then executed:

```vba
DB.QueryDefs("ugrl,resv charge totals").Execute
```

Some methods don’t really apply to an object, so there is a special “DoCmd” object that stands in:

```vba
DoCmd.OpenReport "PO, Single Ship-To", acPreview
```
Private Sub optChooseReport_AfterUpdate()
    If (optChooseReport = 1) Then
        cmbFISCAL_YEAR.Visible = True
        lblFISCAL_YEAR.Visible = True
        txtSTART_DATE.Visible = False
        txtEND_DATE.Visible = False
    ElseIf (optChooseReport = 2) Then
        cmbFISCAL_YEAR.Visible = False
        lblFISCAL_YEAR.Visible = False
        txtSTART_DATE.Visible = True
        txtEND_DATE.Visible = True
    ElseIf (optChooseReport = 3) Then
        cmbFISCAL_YEAR.Visible = True
        lblFISCAL_YEAR.Visible = True
        txtSTART_DATE.Visible = True
        txtEND_DATE.Visible = True
    End If
End Sub
A Small Problem

I have not found a way to pass query parameters directly to a Report that is based on the query. I have had to run a Make Table query, then base the report on the results.

Normal case: report is based directly on query

Special case: report is based on table created by a Make Table query
Here's the Prep

Create a Make Table query. Create a report based on the “made table”. Create a form that accepts parameters and has a “Run” or “Go” button.
Private Sub cmdRunReport_Click()
    Dim DB As Database
    Dim BaseQry As QueryDef

    Set DB = CurrentDb
    DeleteTable DB, "Form Test Table"
    Set BaseQry = DB.QueryDefs("Form Test Make Table Query")
    BaseQry.Parameters("Start Date:") = txtStartDate
    BaseQry.Parameters("End Date:") = txtEndDate
    BaseQry.Execute
    DoCmd.OpenReport "Form Test Report", acPreview
End Sub
The example had parameters from Text Boxes and used a Command Button. But the principles are the same with other Form objects, such as Option Groups and Combo Boxes.
Since forms are designed mostly for data entry and display, the default properties are not usually what you want for a Control Panel.