Handout 6
Visual Basic 6.0
Deere & Company
Handout 6

**Data Environment Designer**

The Data Environment Designer (DED) is available in VB Professional and Enterprise editions to automate the database and front-end design chores. DED’s visual design-time environment enables the creation of ADODB Connection, Command, and Recordset object, helps to write SQL statements, execute stored procedures, and generate data bound controls on the forms.

In the following example, we will use the Data Project, which will automatically invoke the DED.

**Step 1**

Start a new Data Project in Visual Basic 6. Opening a DataProject automatically adds frmDataEnv, DataEnvironment1, and DataReport1 to the Project Explorer. It also creates references to a group of ADO-related type libraries. In our example, we will use the NorthWind database.

**Step 2**

Click on the View Object button in the Project Explorer to show the DataEnvironment window. Rename the DataEnvironment1 to envNwind (you do this by changing the Name property). Remove DataReport1 (we do not need this report writer capability now). You might have noticed that the Toolbox has inherited ADO enabled versions of the Data control (ADODC), DBGrid, DataCombo, and DataList.

**Step 3**

Right click on the Connection1 icon on the envNwind window to open the Provider page of the Data Link Properties sheet. This page gives a list of all the OLE DB providers installed on your system. Choose Microsoft OLE DB Provider for Microsoft Jet 3.51 and click the Next button to move to the Connection Page.

Microsoft provides a number of OLE DB drivers. The SQL Server and Oracle Drivers allow connection to SQL Server and Oracle database servers. ODBC drivers allow for connection to any ODBC capable data source. For DB2 connection, therefore, an ODBC connection established via the CAE tool is used. See Figure 1 for example.
Step 4
On the Connection page, click the browse button to search and select the NorthWind database. Type a username if on is needed and a password if needed. Check the Blank Password checkbox. Click on the Test Connection button to test our database connection. If successful, we are ready to proceed.

Step 5
We will now build a data entry form based on a query command. Click on the Add Command button on the toolbar for the envNwind window. This opens a properties sheet for Command1. Rename the Command1 to Customers (see Figure 2). Also rename the Connection1 to cnnNwind. Select the SQL statement option. Type a simple SELECT statement in the text box as shown.
Step 6
Expand the SQL command item to show the list of columns returned by the query. Open the default form frmDataEnv and line up next to the envNwind window. Drag the command Customers from the envNwind window to the default form window on the right. This will add a set of textbox controls and labels on the frmDataEnv form.

Step 7
To navigate the table, we will add a set of navigation buttons and a Delete button to the form. First we shall set the LockType property to Optimistic in the Advanced page of the Customers Property sheet. See Figure 3. Optimistic allows for multiple or concurrent use of the data tables.
Figure 3 Setting Concurrency Property

Step 8
The following codes are added to the various buttons.

Private Sub cmdDelete_Click()
    With envNwind.rsCustomers
        .Delete
        .MoveNext
    End With
End Sub

Private Sub cmdFirst_Click()
    envNwind.rsCustomers.MoveFirst
End Sub

Private Sub cmdLast_Click()
    envNwind.rsCustomers.MoveLast
End Sub

Private Sub cmdNext_Click()
envNwind.rsCustomers.MoveNext
End Sub

Private Sub cmdPrevious_Click()
    envNwind.rsCustomers.MovePrevious
End Sub

The MoveNext and MovePrevious operations can trigger EOF or BOF status if they reach end of file or beginning of file conditions. We need to trap these conditions. We will add a MoveComplete event procedure for the recordset. Right click the SQL command Customers item and select View Code. The code window or the DED opens. Select MoveComplete procedure for the recordset and type the body as shown here.

Private Sub rsCustomers_MoveComplete _
    (ByVal adReason As ADODB.EventReasonEnum, _
     ByVal pError As ADODB.Error, _
     adStatus As ADODB.EventStatusEnum, _
     ByVal pRecordset As ADODB.Recordset)

With rsCustomers
    If .BOF Then
        .MoveFirst
    End If
    If .EOF Then
        .MoveLast
    End If
End With
End Sub

I have also added an ADD button with the following code.

Private Sub cmdAdd_Click()
    envNwind.rsCustomers.AddNew
End Sub

Note that the Add button behaves slightly different from the other buttons. When Add is pressed you get an empty form where new record values are typed. To record these in the table press any other button. Pressing the Add button a second time gives you another empty form for data entry!
**Hierarchical Data Access**

In the following example, we will enhance our current Data Project so that we can relate Orders and their details to the corresponding Customer who placed these. To generate this hierarchy do the following.

**Step 1**

Open the current Data Project and create a copy of the Project and Form files in a different folder. Name the files different. Open the Data Environment window for envNwind and click on the first command item (the first SQL command item in the window). Select the Property sheet for this command.

![Figure 4 Creating SQL Command](image)

Either use the Query Builder or type in the SQL statement as shown above. Name the command Orders. The Orders SQL command is now part of the Customers command hierarchy.

**Step 2**

Add another command. This time build a SQL query to link Order Details and Product tables to fetch the order details including product name. Name the command OrderDetails. We now need to relate all of these commands together so that a hierarchy of Customer → Orders → OrderDetails can be realized.
Step 3

Open the Orders property sheet and click on the Relation tab (see Figure 5). Check the Relate to a Parent Command Object check box and select the Customers command from the pull down list. DED automatically chooses the correct linking attributes (CustomerID fields from Customers and Orders). Click on the Add button to specify the relation. Repeat the process for OrderDetails command. Here we will link OrderID attributes of parent and child.

![Orders Properties dialog box](image)

**Figure 5 Creating the Join**

Step 4

Right click on Customers command (the top most in the hierarchy) and Hierarchy Info. See what you have! See Figure 6.

The APPEND clauses create the hierarchy and the RELATE … AS defines the relationship. SHAPE is command VB uses to arrange this structure.

Step 5

Now that we have seen the code, let us create the form to display the data. Open the project form where the Customer data bound controls are. Select all of the controls on the form using the mouse and dragging a selection around the controls. When all controls are selected delete these. Make sure all controls except the buttons are removed from the form.
Figure 6 Viewing Hierarchy Command

Drag the Customers command from the envNwind window over to the form window. VB automatically adds text box controls for the fields corresponding to Customers. Also, the MSHFlexGrid control will be added to link up Orders and OrderDetails.

Run the application. See the results. See Figure 7.

The second project is not included among the downloadable files.
Figure 7 Application Layout