DEPARTMENT OF COMMERCE (CA)

VISUAL BASIC (Semester-III)

II M.COM (CA)   Sub Code-18MCC32C

UNIT – V

Tables and Reports-Data grid control, Flex grid control, Data Report - Data Report Designer.

Tables

There are many controls like combos, list box, grid control, which are used to design a form to display the data from a database table. We can access and manipulate the data of tables by using the data access objects, like DAO, ADO etc. The record set object represents a complete set of records from an executed command or from an underlying base table in the database.

Information from a database table is fetched by the DAO and then supplied to various data aware controls like Text Box, DBListBox, etc. Command objects can reference a table in a database, a stored procedure, a database or a SQL statement to modify or retrieve records.

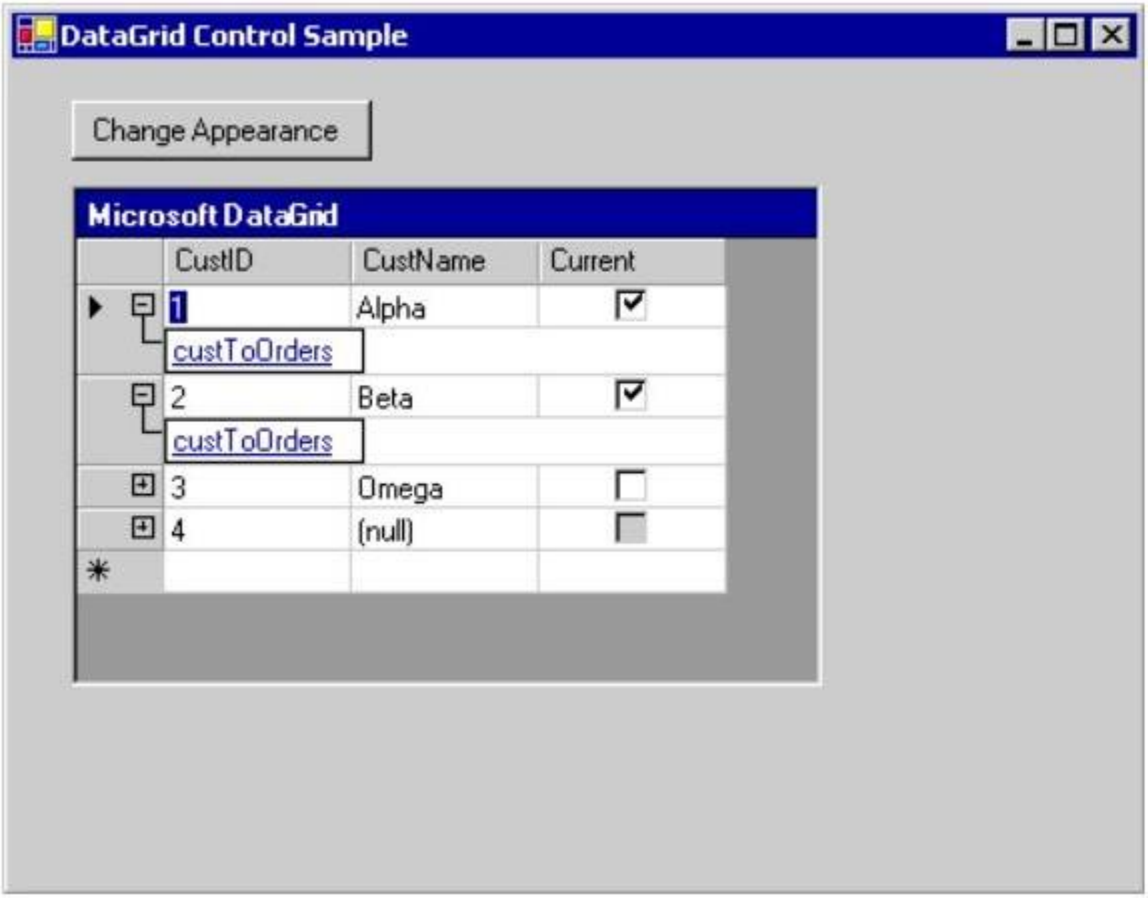
Data Grid Control

The Windows Forms DataGrid control displays data in a series of rows and columns. The simplest case is when the grid is bound to a data source with a single table that contains no relationships. In that case, the data appears in simple rows and columns, as in a spreadsheet. For more information about binding data to other controls, see Data Binding and Windows Forms.

If the DataGrid is bound to data with multiple related tables, and if navigation is enabled on the grid, the grid will display expanders in each row. With an expander, the user can move from a parent table to a child table. Clicking a node displays the child table, and clicking a back button displays the original parent table. In this manner, the grid displays the hierarchical relationships between tables.

The DataGrid can provide a user interface for a dataset, navigation between related tables, and rich formatting and editing capabilities.

The following picture shows a DataGrid bound to data with multiple tables:



	CustID	CustName	Current
▸ ⊟	1	Alpha	☑
	custToOrders		
⊟	2	Beta	☑
	custToOrders		
⊞	3	Omega	□
⊞	4	(null)	□
*			

Flex Grid Control

FlexGrid control (MSFLXGRD.OCX) you can create utilities to display, filter, edit, validate and update your data. For example, such utilities could include:

1. data entry & validation
2. high level reports
3. ported spreadsheet macro applications retaining cell layout & format

Included in VB4(32-bit), VB5 and VB6, are several grid-orientated controls aimed at managing rows and columns of data, however one of the most versatile, hence its name, is the FlexGrid. Most of the other grid objects are specifically designed for data binding, whereas the FlexGrid has many collections of properties, methods and events that lend themselves to several environments in addition to just data-binding.

One of the most powerful applications of the FlexGrid is in managing data from database tables in an unbound state. That is, to populate & edit the FlexGrid and write the contents back to the database in code. This may seem a pointless exercise as objects such Microsoft's Data Bound Grid Control are written specifically for this purpose, except these grids simply display the contents of a recordset, record by record, field by field. Whereas the FlexGid can populate the grid in any manner, examples of which are:

1. separating grouped data with blank rows/columns,
2. adding sub/grand-total rows/columns,
3. changing the colour of the cell background or text in individual or multiple cells,
4. reacting to the state of the data, e.g. highlighting negative values,
5. validating entered data e.g., numeric values, positive values, permitted date ranges etc.

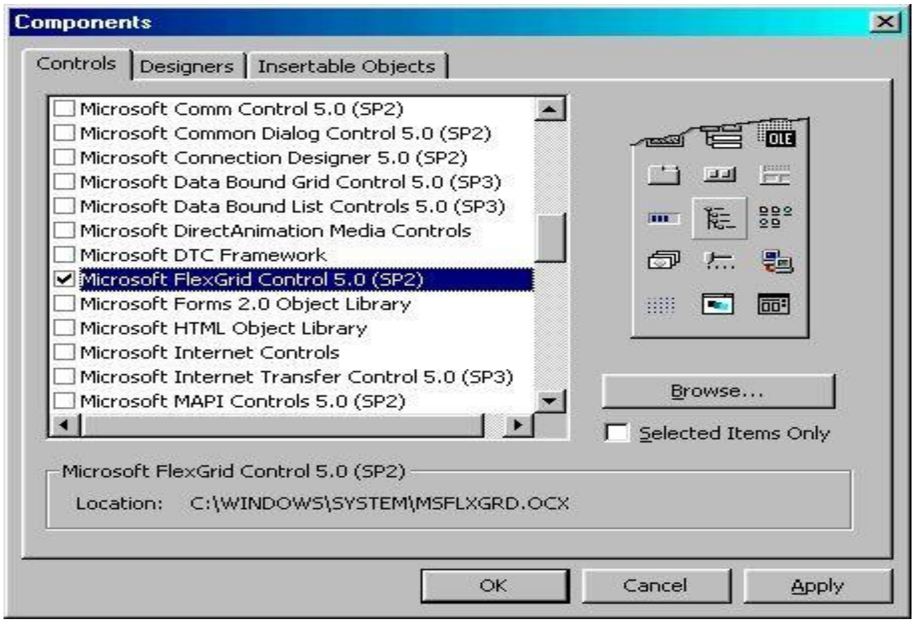
These are just a few of the possibilities available with FlexGrids, we'll take a look at more later on.

One important point is that in addition to the grid controls available as standard to Visual Basic, many third party grid controls are available that boast impressive features such as embedded combo boxes, picture boxes, custom data validation etc. Many of these are excellent controls, however they all come at a price proportional to the size of their feature list.

That said, most of the functionality found in these third party grids can be accessed with FlexGrids with a little extra code which we'll cover in this article and as well as saving money, you'll gain knowledge of creating and customising your grid controls to suit your specific application and you won't have to remember to distribute those third part runtime dll's that custom controls often require.

Using the FlexGrid

To use the FlexGrid in your application, right click the Toolbox or select the Project menu and choose 'Components'. The Components dialog box appears.



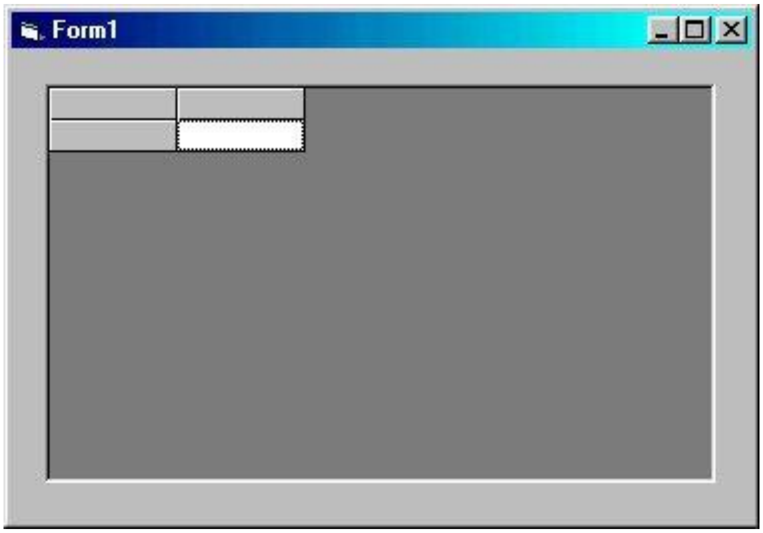
☐ Microsoft Comm Control 5.0 (SP2)
☐ Microsoft Common Dialog Control 5.0 (SP2)
☐ Microsoft Connection Designer 5.0 (SP2)
☐ Microsoft Data Bound Grid Control 5.0 (SP3)
☐ Microsoft Data Bound List Controls 5.0 (SP3)
☐ Microsoft DirectAnimation Media Controls
☐ Microsoft DTC Framework
☑ Microsoft FlexGrid Control 5.0 (SP2)
☐ Microsoft Forms 2.0 Object Library
☐ Microsoft HTML Object Library
☐ Microsoft Internet Controls
☐ Microsoft Internet Transfer Control 5.0 (SP3)
☐ Microsoft MAPI Controls 5.0 (SP2)

☐ Selected Items Only

Microsoft FlexGrid Control 5.0 (SP2)
Location: C:\WINDOWS\SYSTEM\MSFLXGRD.OCX

Components Dialog Box

Select Microsoft FlexGrid Control and copy the control onto your form from the Toolbox. The default FlexGrid will appear on the form.



Default FlexGrid Appearance

There are a few simple formatting tips to improve to appearance of the FlexGrid:

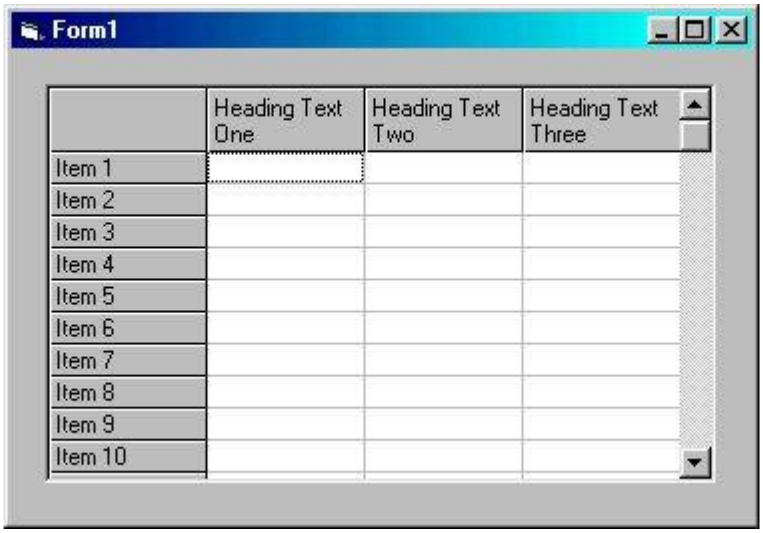
First of all, it's good practice to determine the initial number of grid rows and columns using the Row and Col properties respectively. Then set the number of fixed rows and columns (those that hold row and column headings - raised and in grey) with the FixedRow and FixedCol properties respectively.

To ensure the columns are the correct width to fit inside to FlexGrid, first set a variable to the width of the FlexGrid, allowing for a vertical Scrollbar width. Then use this variable to resize to individual column widths, by dividing it by the number of columns.

The height of individual rows can be changed to accommodate multi-line headings. This is done by simply multiplying the appropriate row height by a scaling factor, in this case two, to double the height.

When increasing the size of a row height, to make sure text uses the extra available space, use the WordWrap property the ensure text continues on the next cell line.

Regarding writing text to the grid there are two main methods. The first one is using the Additem method, this writes text to an entire row in one action and is therefore useful for adding headings. The vbTab constant is used to distinguish adjacent cells. The second method to populate grid cells with text is to directly address individual cells and use the TextMatrix or Text property and set the cell's text.



	Heading Text One	Heading Text Two	Heading Text Three
Item 1			
Item 2			
Item 3			
Item 4			
Item 5			
Item 6			
Item 7			
Item 8			
Item 9			
Item 10			

Initialised FlexGrid Appearance

Data Report

It helps you design a report that displays fields and records from the underlying table or query.
A report is an effective way to present data in a printed format. You can display the information the way you want to see it
You create the report using a graphical object called Data report designer controls. Data report designer controls. Include: a data-bound Textbox control, a function control which displays calculated figures, and Image control for inserting graphics, labels that display captions, and a line and a shape control that graphically organizes the data.
Note: Although the data report designer controls are similar to Visual Basic intrinsic controls, data report designer controls have a limited subset of features. When the Data report designer is added to a project, the designer's controls are placed in the Visual Basic toolbox on a new tab named data report and can be used only in the Microsoft data report.
Creating reports is a main function of any good business application. A system might have useful data, but without a coherent way to present it, the numbers are meaningless. So, Microsoft provides a Data Report designer. Which is used to generate eye catching reports. It is used in conjunction with a data source such as the data environment designer.
The Data Report generates reports using records from the database. To use it.

1. Configure a data source, such as the Microsoft data environment, to access a database.
2. set the Data member property of the data report object to a data member.
3. set the data member property of the Data Report.
4. Right click the designer and click retrieve structure,
5. Add appropriate controls to the appropriate sections.
6. Set the data member and Data field properties for each control.
7. At run time, use the show method to display the data report.

Use the data report object to programmatically change the appearance and behavior of the data report by changing the layout of each section object.

The data report designer also features the ability to export reports using the export report method. This method allows you specify an export format object, from the export formats collection, to use as a template \for the report.

Data Reporter designer

It consists of the following objects:

Data Report Object – Similar to a Visual Basic form, the Data Report object has a visual designer. It is used to create the layout of a report.

Section object – The Data report designer contains sections which helps to configure the report more elegantly. The sections of the Data report designer are explained in the next section.

Data report controls – special controls that only work on the data report designer are included with it. These controls are found in the Visual Basic toolbox, but they are placed on a separate tab named "Data Report'

Report Designer – User the report Designer to create and modify reports. When the report designer window is active, the Visual Basic display report controls the toolbar.

Data report designer features – the data report designer has several features:

Drag and drop functionality for fields – When fields are dragged from the Microsoft data environment designer to the data report designer, Visual Basic automatically creates a text box control on the data report and sets the Data member and Data Field Properties of the dropped field. A command object can also be dragged from the Data environment designer to the data report designer. In that case, for each of the fields contained by the command object, a text box control with be created on the date report; the Data member and Data field property for each text box will be set to the appropriate values

Toolbox controls – The data report designer features its own set of controls. When a Data report designer is added to a project, the controls are automatically created on a new toolbox tab named Data report. Most of the controls are functionally identical to Visual Basic intrinsic controls, and include a label, shape, image, textbox, and line control. The sixth control, the function control, automatically generates one of four kinds of information: sum, average, minimum, of maximum.

Print Report – When a report is generated, it can be printed by clicking the printer icon on the toolbar.

File export – in addition to creating printable reports, it can be exported to text files, by clicking the export icon and specifying text file name.
Section of the Data Report Designer

The Data report designer contains the following sections:

Report Header – Contains the text that appears at the very beginning of a report, such as the report title, author, or database name.
Page Header – Contains information that goes at the top of every page, such as the report's title.
Group Header/Footer – contains a "repeating" section of the data report. Each group header is matched with the group footer. The header and footer pair are associated with a single command object in the data environment designer.
Details- Contains the innermost "repeating" part (the records) of the report. The details section is associated with the lowest level command object in a data environment hierarchy.
Page Footer – Contains the information that goes at the bottom of every page, such as the page number.
Report Footer – Contains the text that appears at the very end of the report, such as summary information, or an address of contact name.
Data report controls – When a new data report designer is added to a project, the following controls are automatically placed in the toolbox tab named data report.

- **Text Box control (Rpt text Box)** – Holds the data that is supplied at runtime.
- **Label control (RptLabel)** – used to identify fields or sections.
- **Image control (RptImage)** – enables to place graphics on
- **Line Control (Rpt Line)**- lets to draw rules on the report to further distinguish
- **Shape Control (Rpt Shape)** – Enables to place rectangles, triangles, or circles 9and ovals) on a report.
- **Function control (Rpt Function)** – A special text box that calculates values as the report is generated.

Printing a Data Report
Printing a data report can be accomplished in one of two ways. The user can click the print button that appears on the data report in print preview mode (using the show method), or you can programmatically enable printing using the print report method. If an error occurs during printing, trap it in the error event.
Choosing to display a print dialog box
When printing a report programmatically, you have two choices: to print by displaying the print dialog box or by printing without displaying the dialog box.
To display the print dialog box.

1. Add a command button to a form.
2. In the button's click event, place the following code: data report1. print report true.

The print dialog box allows the user to select a printer, print to file, select a range of pages to print, and specify the number of copies to print.

Printing without a Dialog Box

In some cases, you may wish to print the report without user intervention. The print report method also gives you the option of selecting a range of pages to print, either all, or a specified range.

To print without displaying the dialog box

1. Add a command button to a from
2. In the button's click event, place the following code:

Data report1. Print report false

Or, to specify a range of pages to print, use the code below:

0 Data report print report false, rpt created with the data report designer.

Print Report Method

At run time, prints the data report created with the data report designer.

Syntax

Object. Print report (Show Dialog, Range, and Page From, Page To)

The print report method syntax has these parts:

Past description object required. An object expression that evaluates to an object in the applies to list. Show dialog Optional. A Boolean expression that determines if the print dialog box is shown. Range optional. Sets an integer that determines if all the pages in the report will be executed or a range of pages, as shown in setting. Page from optional. An integer that sets the page from which to start printing. Page to optional. An integer that sets the page at which to stop printing.

REFERENCE :

1. Programming Microsoft Visual Basic- Francesco Balenda, WP Publications and Distributors.

2. Visual Basic 6-Gary Cronell, TataMcGraw Hill Publishing Company Ltd.

3. Visual Basic 6 – How to Program, H.M.Deitel., P.J .Deital and T.R.Nieto

Prepared by Dr.N.SHANMUGAVADIVU