**VISUAL BASIC 6.0**

**PREPARED BY S.KALAISELVI/R.KANAKARAJ/R.LATHA**

**UNIT – 1**

First step with Microsoft visual basic 6.0 : integrated development environment ; first program in VB: introduction to forms : common properties , methods and events.

**UNIT – 2**

Intrinsic controls: text box controls, labels and frame controls , command button , check box and option button controls , list box and combo box picture and image controls. Drive – list box , dir list box and file box xontrol and other controls, control arrays

**UNIT – 3**

variables and procedures : scope and life time of variables . native data types and aggregate data types – array . VB for application and VB for libraries : control flow , working with numbers , string , date and time

**UNIT – 4**

Database : data access SAGA, visual data base tools, ADO data binding , data environment designer , crash course in SQL . database programming : ADO at work – setting up a connection processing data

**UNIT – 5**

Tables and reports – data grid control , flex grid control , data report designer

**UNIT – I**

**VISUAL BASIC 6.0**

Visual: graphics pixels

Basic: coding

Programming ->VB->standard.Exe->form.

=>to run the Vb program press f5.

=> To save the VB program use following extensions.

\*.Vbp

\*.frm

**MICROSOFT VISUAL BASIC 6.0**

**FIRST STEP IN MS VB 6.0**

IDE- application creating, writing the coding, testing the coding executing the coding.

Creating applications and writing the codings. The code will be tested and produce executable files. That file should be independent.

**RUNNING THE IDE:**

Com components:

Controls:

OLE

ADO

ODBC

**SELECTING THE PROJECT TYPE:**

|  |
| --- |
| Microsoft visual basic 6.0 |
| Recent  Existing  New |
| Wizard.Exe  Active.Exe  Standard .Exe |

**TOOL BOX PROPERTIES**

1. Pointer

Maximizing and minimizing form design

2. Label

Display the text but it cannot be modify

3. Text

Display the text but it can be modify

4. Frame control

Controlling the functions

5. Check box:

Giving the format like true () or false (), yes or no.

6. Combo box

Display the text at a time of clicking file option

We can select more than one option in combo box but it cannot in list box

7. List box

8. ‘H’ and ‘V’ scroll bar

Select the value within a range

9. Timer control

To specify the intervals like seconds, minutes.

10. Directory list box

To select directories c:\

11. Shape controls

12. Image controls

Icons, bitmaps, Meta files

13. OLE controls

Object linked embedded

14. Command Button

To carry out the specify actions

15. Option button or radio button

Displaying some particular things like enable & disable

**IDE WINDOWS:**

1. Project window
2. Form design
3. Tool box
4. Code window
5. Properties window
6. Form layout
7. Immediate window
8. Project window

Gives over view of all modules, that modules will contain in applications,, that module will be displayed in grouped form that is in alphabetical order by clicking on the right mouse icon on the project windows tool bar.

1. Form designer
2. Design applications on user interface
3. Application can contain multiple forms
4. Open any number of forms at a time in both VB 5.0 and 6.0
5. Set of object can be placed on a form
6. Fixed set of controls called intrinsic controls.
7. Code window
8. To write code that determines the behavior of form and object of applications
9. Keep any number of code windows visible at a time.
10. Split code window in two distinct and independent position by dragging
11. Properties window
12. List all the properties of the object that currently selected and arranged in alphabetical order and also group them
13. Change fore ground and back ground colors. If it is not visible press “F4” or by clicking tool box icon on the “ TOOL BAR”
14. Form layout
15. Form will be displayed when the program runs.
16. Immediate windows
17. Temporary suspends while running program

**MENUS:**

1. File
2. View
3. Project
4. Format
5. Debug
6. Run
7. Query
8. Diagram
9. Tools
10. Add-ins
11. Windows
12. Help
13. File

\* load and save a VB project or group of project

\*VB 6.0 can open multiple projects to save the current module

\* To print the entire project or selected project

\* To built the executable files

1. Edit
   1. Editing command include cut copy, paste , find ,replace , undo and redo.
   2. It also include bunch of commands related to Microsoft intelligence
2. View
   1. Database related commands
3. Project
   1. Add modules include forms, standard modules, class modules, user control module , designer module among this standard modules , class modules , user control modules are also used for reference.
4. Format
   1. Align and resize one or more controls on a form
   2. Centre a control
   3. Increase and decrease the distance among the group of controls
5. Debug
   1. Testing an application with in a integrated development environment
   2. Execute the codes step by step
   3. Display the values of variables and expressions
6. Run
   1. Command start the execution of the application
7. Query
   1. It is available only in VB Enterprises
8. Diagram

|  |
| --- |
| New text |
| Add related tables |
| Show relationship labels |
| Modify custom view |
| View page breaks |

1. Tool
   1. Tool menu contains option command and it access only in dialog box by customize IDE.
2. Add –ins
   1. Collection of commands related to external modules
3. Windows
   1. It is applicable only in MDI applications
4. Help

**TOOL BARS:**

* Tool bars include common commands are loading, saving the project, running the program and opening.
* Debug, edit and form editors are visible only after right click on the standard tool bar.

**CUSTOM TOOL BAR**

* Project menu
* Comment block
* Uncomment block
* Command from the edit tool bar
* Book marks sub menu from edit tool bar
* Option command from the tools.

Adding controls to a form launch visual basic standard. Exe project

Design time create forms and other variables

Form design- defined the appearance of the main window of the application

Run time – a limited ability to modify the code

Setting a properties of controls

1. Label controls
2. Border style
3. Text box
4. Caption property
5. Control

Naming controls

* Name property - string of character that identifies the control in code.
* Default assigning name are label1, text1, command1.

|  |  |
| --- | --- |
| **Control/ class** | **Prefix** |
| Command button | Cmd |
| Text box | Txt |
| Picture box | Pic |
| Label | Lbl |
| Option button | Opt |
| Check box | Chk |
| Combo box | Cbo |
| List box | Lst |
| Timer | Tmr |
| Frame | Fra |
| Data | Dat |
| H scroll bar | Hsb |
| V scroll bar | Vsb |
| Drive list box | Drv |
| Dir list box | Dir |
| File list box | Fil |
| Line | Lin |
| Shape | Shp |
| OLE | Ole |
| Form | Frm |

**Moving and resizing controls:**

* Select one or more controls and move them as a single entity using the mouse
* Move one or more controls with arrow keys while pressing the control key
* X and Y axis are determined by the grid control or grid units
* Resize the selecting controls by using the arrow keys , while press the ‘shift’ key
* Center a control or a group of controls on the form horizontally and vertically while using the centre in form sub menu
* Align a group of controls or align a sub menu
* Align or resize the controls by selecting them , pressing “F4” to display the properties window.

**Setting the tab order**

* TAB key is set to be logical order, search order is known as TABORDER sequence.
* Correct tab order sequence assign with tab index property
* Tab index property starts with ‘0’ to the text width controls and ‘1’ to the text height control tab index property

**Adding code**

1. In command button “cmdsum”
2. Dim statement is used to specify data type.
3. Option explicit statement is use to verify default errors.

**Running and debugging the program**

* Clicking the procedure, while program is in design mode that mode is called toggle break event command from the debug menu or press ‘F9’ in keyboard.

**Refining the sample program**

* First time defining is very simple text parameter and text area controls are used to show the result of the competition.
* To increase the application consistency and use ability
* To double click the control to create the “sub” and “end sub” template

**Exit sub statement**

* Msg box statement executed during the normal execution flow.

**Ready , compile and run**

* Vb convert source code into pcode and then integrate it
* P code is sort of intermediate language. it is not executed directly by CPU , its speed is normally slower.
* Conversion from the p code to source code takes only a fraction of time need to delivery a compiled application
* During the program development create and executable program
* Compile programs are often faster then interprets one
* All VB programs depends on number of accessory files
* MSVBVM6.0.DLL is a part of VB
* P code compile executable are often smaller then programs compile to native code
* P code compliatation is often faster than native code compilation
* P code complied programs run at the same speed as interpreted programs within IDE

**Introduction to forms**

Common properties

1. All visible object forms and controls expose this properties which effect the object position and size. These values are always relative to the object container. this screen for a form and aprent form for the control
2. A control can also be contain in another control which is set to be its container
3. Top and left properties are controlled and it is measured by “TUIPS”
4. Left, top , width height properties are forms and also measured by “TUIPS”

**Font color and back color properties**

1. Fore color and back color properties which effect the color of text and color of the back ground
2. Back color property of a label control has no effect
3. Back style property of that label is ‘0’ transparent
4. Command button controls peculiar.

There are two properties

I standard window color is use for system tabs

II custom color it is used for color palatte tab

Vb defines Black , VBblue as numeric and hexadecimal constant use and RGB function to bulid a color value.

Example

Text1.backcolor = VB black

Text1.back color = 16776960

Text1.back color = &HFFFOO

We can set value of its Green, Red and Blue components

Example

Text1.back color = 0,255,255

Quick basic application support “QB COLOR” function

Example

Text1.backcolor = QBC

**Font property**

Text1.font name = name

Text1.font size = 12

Text1.font bold = true

Text1.font underline = true

**Caption and text property**

1. Caption property is a string of character that appears inside a control and that the user cant modify directly
2. Caption property are the label , command button , check box , option button , data and frame controls
3. Text property are text box , list box ,combo box controls
4. Caption property special that include ‘&’ that character associated with ‘hot keys’

Example

Text2.text = text1.text

[or]

Text 2 = text 1

1. The text property also support ‘seltext’, ‘selstart’,’sellength’ and return information about the portion of text that currently selected in the control

**Parent and container property**

* Parent property is only a run time set
* Data will be move from one control to another
* Container property is only run time property which returns a reference to the container of the control , data will be move from one place to another place with assigning of different values

**Enable and visible properties**

Text1.enabled = [chack1.value = Vbchecked]

* Default controls and forms are both visible and enable at run time
* Frame control can be a container for a group of option button controls setting its visible and enable at run time
* Set the enabled property to false for ‘text box’ or ‘combo box control’ that must work in a read only mode.

**The hwnd property**

1. Hwnd property does not appears the properties window because its value available only at run time
2. Hwnd property returns the 32 bit integer values

**Tab stop and tab index properties**

* Controls are to receive the input focus it expose the tab stop property
* Intrinsic controls support text box, option button, check box, command button , OLE, combo box
* If a controls support tab stops property it also support the tab index property which must be the tab order index.
* The tab index property is also support the labels and frame controls. but this two controls does not support tab stop property

**The mouse pointer and mouse icon property**

* This property effect shapes of the mouse cursor.
* Each form and each control can display a different cursor.
* Screen mouse pointer property is set to values different from ‘0’ – VB default
* Screen mouse pointer is ‘0’ and mouse cursor is over the control, Vb checks that controls mouse pointers property. If values different means its set the values
* If screen mouse pointer is ‘0’ and the mouse is over the form surface (or) it’s over a control whose mouse pointer property 0 , Vb uses the values stored informs mouse pointer property

**The TAG property**

Data is relating to the controls.

1. Value property
2. Index property
3. Intrinsic controls
4. Align property
5. Border style property
6. tool tips

* Value property

Value property is commonly to several intrinsic controls namely check box, option button, command button and scroll bar controls as well as too many externals.

* Index property

It is used to bulid control arrays

* Intrinsic controls

It is use to support appearance property and assign in design time and it will be read only at run time

* Align property

It is a non – null value

0 – none

1 – Align top

2 – Align bottom

3 – Align left

4 – Align right

* Border style property

It supports few intrinsic controls are text box, label, frame, picture box, image and ole controls.

* Tool tips

**Drag mode and drag icon property**

Where used to trag control in form, link mode, link topic, link item, link time out are used to communicate through DDE [Dynamic Data Exchanges]

**Command methods**

1. The move method
2. Refresh method
3. Set focus method
4. Zorder method
   * 1. The move method

If a controls support left, top, width, height properties and also change all are single operation

**Syntax**

Move left, top, height, width

Form1.move 0.0, form 1. Width \* 2

ii. Refresh method

It is used for redraw the data automatically

**Syntax**

For n = 1000 to step – 1

label1.caption = str (1)

Label. refresh

iii. Set focus method

Set focus method moves the input focus on the specified controls. It modifies only the default TAB order sequence. Set focus method is that if fails and a raise or run time error if the control is currently visible or disable

**Syntax**

1. If text1.visible and text1.enabled then

Text1.set focus

End if

1. Private sub form\_ load( )

Show

Text1.setfocus

End sub

1. Private sub fom\_load( )

Text1.tabindex = 0

End sub

Iv Z order method:

Affect the visibility of the control and over lapping the controls. It is used in only design time.

Text1.zorder = 1

Text1.zorder

**Common events**

1. The click and double click event
2. Change event
   1. Text box
   2. Combo box
3. Get focus and lost focus events
4. Key press, key down, key up events
5. Mouse down, mouse up, mouse move events
   1. Click and double click event access when the user left click on the control

Click and double click event does not pass arguments to the program

**Syntax**

Private sub form \_click ( )

Dim t as single

Is click = true

T = Times

Do

Do event

If not exit sub

Loop until timer> E+5 or timer < T

End sub

Private sub Dbl\_click ( )

Is click = false

End sub

* 1. Text box and combo box controls raise a change event when the user types editable area of the control.

Scroll bar controls raise the change event when the user click on either or moves the scroll bar.

Change event is also supported by picture box, dirlist box, directory list box.

* 1. Get focus control receives the input focus and lost focus.. input focus leaves and passes to another control.
  2. Form support both get focus and lost focus event a raised only when the form doesnot contain any control.
  3. Key down user uses the keypress to translate ANSIC numeric code.

[ANSIC – American National Standard International Code]

**Syntax:**

Private text1\_keypress [keyascii as integer]

Msgbox”user pressed”+char [key ASCII]

End sub

[ASCII – American Standard Code for Information Interchange]

Key down, key up events receives two parameters. Key code and shift code.

**Syntax:**

Private sub text1\_keydown [key code as integer , shift as integer]

If shift & vbshift mark then

“Shift key pressed”

End if

If shift & vb ctrk mark then

“Ctrl key pressed”

End if

If shift & vb alt mark then

“Alt key pressed”

End if

End sub

* 1. This event fires when the mouse is click realize on a move control respectively. All this 3 controls received same set of parameters.

Mouse down and mouse up event are raised any time a user press or realize a button.

**Syntax:**

Private sub form \_mouse down [button as integer, shift as integer, ‘x’ – as single, ‘y’ – as single]

‘Show mouse state button, shift x, y’

End sub

Private sub form mouse up[button as integer – shift as integer x – as single , y – as single]

‘Show mouse state button, shift x, y’

End sub

Private sub form \_mouse move [button as integer – shift as integer x – as single, y – as single]

‘Show mouse state button, shift x, y ’

End sub

**UNIT – II**

Intrinsic controls: text, combo, list, command, option, check.

MSV BVM 6.0.DLL – it supports – OLE – files

Intrinsic controls it support the version of MSVBVM 6.0 .DLL intrinsic controls does not need any additional ocx files.

It simply files the installation and reduces disk requirements

It can create and display intrinsic controls in external active x – controls.

1. Text box controls:

It can create and display intrinsic control in external active x controls

Multiple lines set multiple properties as true

Alignment property set text box controls to left, right and center align.

Right alignment text box controls are useful when displaying numeric value. It works with single line controls only under MS windows 98, NT4 etc.,

Maximum length property default value ‘0’ , that you can enter in any positive values that

N in focus a limit to the length of the field contents to N character length

Run time properties:

1. Sel start
2. Sel length
3. Sel text
   1. Sel start:
      1. Its sets are returning the position at the blinking caret. blinking char inside text Other name is called caret.
      2. Caret at the beginning of the content of the text box controls sel start ‘0’ and set typics length.
   2. Sel length

It returns number of character in the position of the text. That has been highlighted by the user Are return 0

If there is no highlight text you have align none ‘0’ value to the property to programicalyy

Select text form code.

* 1. Sel text

If set a return the portion of the text that currently selected

To append

Text1.selstart = len [text1.text]

Text1.selstart = string to be added

**Syntax:**

Private sub text1\_getfocus ( )

Text1.selstart = 0

Text1.sel length = 9999

End sub

Trapping keyboard activity:

The text box control support key down, key press, key up standard events.

**Syntax:**

Private sub text1\_key press [key ASCII as integer]

Select case key ASCII

Case is < 32

Case 48 to 57

Case else

Key ASCII = 0

End select

End sub

Private sub text1\_key down [key code as integer, shift a integer]

Select case key code

Case VB key down

Text1.text = cdbl (text1.text) – 1

End select

End sub

**Validation routines for numbers**

While doing project if there is any invalid value just trapic

Instrated of trapping before it get to the text box controls

**Cause validation property and validate event**

When input focus leaves a control VB checks the cause validation property whether it receives are not.

If cancel parameter is set to true. VBcancel the user can take in input focus.

Private sub text required – validate [cancel as Boolean]

If text required .text = “ ” then msg box

“Please enter something VB exclamation cancel = true

End if

End sub

**Unload mode:**

Private sub form \_ query unload (cancel as integer, unload mode as integer)

If unload mode = VB form control name then

Validate control

If err = 380 then

Cancel = true

End if

End if

End sub

Clicking the unload mode parameter is important. If application is mistakenly executive a validate control method when the user click on the cancel button

**Auto tabbing field**

Private sub text1\_change ()

If len (text1.text) = text1.maxlength

Then send keys “{tab}”

End if

End sub

**Formatting text:**

1. Numeric values
2. Fixed numbers
3. Currency values
4. Phone no’s
5. Credit cards
6. Date

Private sub txtphone\_gotfocus ()

Txtphone.text = filterstring (txtphone.text, “0123456789”)

End sub

Private sub txtphone\_lostfocus ()

Txtphone.text = format phone number (textphone.text)

End sub

Numerical values can be formatted with 1000 separators

Eg:

(“##. ##. ##0. ####”)

Fixed number can be formatted in decimal digits

Eg:

(0.1, “.”)

Currency values, symbols can be automatically inserted

Eg

Format $(0, “currency”)

Format (“$0.00”)

Phone numbers formatted can be used in dash (-) \_ to split into groups of digits

Eg

“0123456789”

Credit card numbers it is used in readable with embedded space

Date

(“September, 10, 1999”)

(“MM/DD/YYYY”)

(“DD/MM/YYYY”)

**Multi line textbox controls:**

Multi line text box setting the multiline property to true

Scroll bar property assign the value (2) for vertical and other are (3)

Vertical scroll bar causes the contents, contents of the control that automatically wrap. The line is two lengths.

If multiline is falls means it ignores the scroll bar property

Horizontal and vertical are read only at run time, it can’t alter between a regular and multiline text box.

**Label and frame controls:**

Label controls provide a descriptive caption

Label controls assign ‘&’ in front of the hot keys

Label controls event procedure expose only the subset of events supported by other controls

**Frame controls:**

If we move the frame control all the child controls go with it.

If we make container control disabled (or) invisible all the child controls are become disabled (or) invisible

Command button, check box, option button:

* Unload me
* Set default
* Cancel property

Private sub command 1\_click ( )

‘Save data, the unload the current form’

End sub

**Check box:**

Private sub command1\_click ( )

Frame1.enabled = (check1.value = Vbchecked)

End sub

**Option button controls:**

If opt weekly. Value then

Else if

If opt monthly. Value then

Else if

If opt yearly. Value then

Else if

In option button group of two or more data can be click at any time

In a group of button select the state other are unselected

**Going graphical:**

Private sub command1\_picture = load picture (“c.\vbp\myicon.Ico”)

**List box & combo box**

List box shares many methods, events and properties

List box sets the sorted attributes to true and to create list box automatically sorted in alphabetical order also set vertical, horizontal or single

Private sub form1\_load ( )

List1.additem = “first”

List1.additem = “second”

List1.additem = “third”

End sub

1. Add beginning

List1.additem = “zero”, 0

1. Remove & clear

List1.remove item = 0

List1.clear

1. List index

If list1.listindex = -1 then

Msgbox “no item selected”

Else

Msgbox”user selected & list1.text

End if

1. Select third item

List 1.listitem = 2

Enumerate item

For i = 0 to list1.list count [-1]

Print “#&i&” = ‘&list1.list (I)

Next

Reacting to user action

The item data property

“Add an item to the end of the list

Lstcust.additem customer number

“Remember the matching customer ID

Listcust.itemdata

**Multiple selection list boxes:**

1. Simple
2. Extended
3. Sel count

Simple – select and deselect individual items only by using the space bar (or) the mouse

Extended – shift key to select range of items multi select property change when the program is running

Sel count property written the number of items that are currently selected

**Picture box ad image controls**

1. Auto redraw
2. Draw

Graphical methods = cls, pset, point, lines, circle

Conversion = scale x, scale y

Picture box property support graphical output including auto draw, draw

**Loading images:**

I bit map (BMP)

II Device Independent Bit Maps [DIB]

III Meta Files [MF]

IV Enhanced Meta files [EMF]

V [GIF, JPEG] Compressed Files

VI [ICO & UR] Icon

**Syntax:**

Picture1.picture = load picture [“c.\windows\setup.bmp”]

Paint picture method:

Perform wide variety of graphicals including zooming, scrolling, painting , flipping

Unset (null set)

Nutshell perform pixels by pixels copy from source to designation controls.

Syntax:

Dest Pict box paint picture. Srcpict box .picture, destx, dest y, [dest width],[dest height][src x],[src y],[src width],[src height]

Image controls:

Picture.picture[c:\windows\bmp]

**Syntax:**

1. Image1.stretch = false

Image 2.picture = load picture [“c:\windows\setuo.bmp”]

1. Image1.stretch = true

Image 1.move = 0, 0

Image1.width/2

**Scrollbar controls:**

I hscroll bar

II vscroll bar

Small change & large change maximum and minimum represents the valid range of values.

Small change represents the variation in clicking on the scroll bar. Large change represents variation in values either side of the scroll bar indicator.

**Drive list box, dir list box, & file list box:**

Drive list control combo box that automatically filled with Ur drive letters and volume labels

Dir list box is a special list box that display directory free

File list box control is filled with list of files in that directory [\*.\*]

**Syntax:**

File1.pattern = “\*.txt;\*.doc;\*.setf;”

Also enter multiple specifications using the semicolon as a separators.

Other controls: timer

Timer control visible at run time, its purpose to send a periodic pulse to current applications, there are two properties they are internals and enable.

Interval stands for the no.of.multiseconds between subsequent.

Enable is used for activate and deactivate events.

**Syntax:**

Private sub form load ( )

Timer1.interval = 500

End sub

**Unit – III**

**VARIABLES & PROCEDURES**

**Variables:**

Variable is used to storing values temporarily variable name must begins with an alphabetical letter. It should not exit 255 characters. It must be unique. It should not contain any special character like %, @, #, &! Etc

**Syntax:**

*Dim variable [as type]*

Eg

Dim strname as string

Dim int count as integer

Using option explicit statement

General

Option explicit

**Scope of the variables:**

It determines procedure or object in which part of code.

There are two types of variables procedures or local variables and modules level variables .

Local variables should be declare inside the procedure

**Syntax:**

*Dim int temp as integer*

**Static variables**

**Syntax**

*Static int permanent as integer*

Eg

Function running total ( )

Static accumulate

Accumulate = accumulate + num

Running total = accumulate

End function

Calculate total by adding new values to the previous values stored in the static variable values

**Global variables :**

Global variables using the public key word BAS modules . it read at modifies from any in the current program

**Module level variables**

Public int permanent as integer

Private int temp as integer

Public variable should not be declare with in aprocedure . it is public integer variable int yis declare in both form and modules . one of the project . it can be refferd as form1.inty and module1.inty

**OVERVIEW OF NATIVE DATA TYPES**

**Integer data types**

Range -32,768 to 32,767

16 bits and 2 bytes

**Long data types**

-2,147,483,648 to 2,147,483,647

32 bits and 4 bytes

Integer value in the range – 32,768 to 32,767

Wholed integer value - 2,147,483,648 to 2,147,483,647

**Boolean data types**

It is used for to state true or false. It has only one in single bit

**BYTE DATA TYPE**

**Single data type**

-3.402823E.38 through -1.401298E.85 negative

1.401298E.85 through 3.402823E.38 positive

**Double data types**

-1.79769313486232E.308 through -4.9406564581247E.324 = negative

4.9406564581247E.324 through 1.79769313486232E.308 = positives

**String data types**

Dim varlen str as string

Dim fixed lenstr as string \*40

**Currency data type**

-922,337,203,685,477.5808 through 922,337,203,685,477.5808

**Date data types**

DD/MM/YYYY

MM/DD/YYYY

1st Dec 1999

**Object data type**

Eg : dim frm as form

Dim MDIfrm as MDI form

**Variant data type – it is divided into following types**

* Var str
* Unused
* Value

**Decimal data type**

79,228,162,514,264,337,593,543,950,335

Eg:

Dim V as variant

V= (dec(text1.text))

**Aggregate data types**

1. Private type employee UDT

Name as string

Department ID as string

End sub

1. dim emp as employee UDT

EMP name = “GURU”

EMP department ID = 123

**Arrays**

1. Static array
2. Dynamic array

Static and dynamic arrays:

Dim name (100) as string

Create either static or dynamic array

Static array must include a fixed no of items and the no of items must be known at compile time

To create a static array using a dim statement with the constant argument

**Syntax**

Dim name(100) as string

Dynamic arrays can be recreated while receive data its contents or reset to zero or empty string

If we want to resize an array without losing it contents use the redim command

Example

Redim names (100) as string

**L bound and U bound**

Print L bound (cells, 1)

Print U bound (cells, 2)

Array within user defined data type

**Structure**

*Type my UDT*

*Static array (100) as string*

*Dynamic arr ( ) as long*

*End type*

Array and variants

Redim names ( ) as string, var as variant

Var = name

Print var (1)

Assigning and returning array

Redim a(10,10) as integer

Dim b ( ) as integer

b ( ) = a ( )

if you can perform assignment between array and you can write procedures that return arrays. You can write procedures that retrun arrays. You can assign array only if the same type and only if the target is a dynamic array

**syntax**

*Function in array*

*Redim result (first to last ) as long*

*Dim I as long*

*For i= first to last*

*Result (I) =i (100)*

*Next*

*In array = result*

*End function*

**Byte array**

Dim b ( ) as byte, text as string

Text = “123”

b ( ) = text (or) text =b ( )

**ODD / EVEN NUMBERS**

**Coding**

Dim b ( ) as byte, I as long

b ( ) = text

For (i) = 0 to U bound (b)

If b (i) = 32 then count space = countspace +1

Next

End if

End function

**Inserting and deleting items**

**Inserting**

Sub insert arr item (arr as variant, index as long, new value as variant)

Dim I as long

For I = U Bound (arr) -1 to index

Next

Arr (index) = new value

End sub

**Deleting**

Sub delete arr item (arr as variant, index as long)

Dim I as long

For I = index to UBound (arr) – 1

Arr (i) = arr(i+1)

Next

Arr[U bound(arr)] =empty

End sub

**SORTING**

Sub shell sort any (arr as variant, num Els as long, descending as Boolean)

Dim index as long, index as long, first item as long

Dim distance as long

If varitype (arr) < vbarr then

Exit sub

First item = L Bound (arr)

Do

Distance = distance\*3+1

Loop until distance > num els

Do

Distance = distance/3

For index = distance 4 first item to numEls + first item – 1

Value = arr (index)

Do while (arr (index) 2 – distance) > value x or descending

Arr (index (2)) = arr (index (2) – distance)

Index = index 2 = distance

If index = distance < first Item then

Exit

Do

loop

Arr(index (2)) = value

End sub

**ARRAYS OF ARRAY**

Collection

Create

Dim employee names as collection

Set employee names as new collection

Or

Dim employee names as new collection

Or

Employee names.add “Guru”, “marketing”

Or

Employee names.add “Guru”, “slaes”

Employee names . add value 2 “Guru” , “sales”

**Numeric & string**

Print employee name item (“sales”)

Print employee name item ( )

Print employee name item (“sales”)

Print employee names item ( 1)

**Iterating on collections object**

Dim I as long

For I = 1 to employees names. Count

List 1.additem employee names (i)

Next

For each ….. next loop

Dim I as long

For each I as employee names

List1. Additem i

Next

**Dictionary object**

Dim dict as new scripting

Add

Dict. Add “key” . value

Print dict (“key”) empty

Dict (key 2) new value

Remove

Dict . key (“key”) = “new key”

Exist method

If dict exists(“Guru”) then

If dict print item “Guru” exist

2 methods : items and keys

Dim item values ( ) as variant

Dim item key ( ) as variant

L as long

Item values = dict .items

Item keys = dict.keys

For I = 0 to U Bound (item values . list1.additem item keys(i) &’ = “&”

Item values (i)

Next

**Procedures**

* Sub
* Function
* Property
* Private
* Public

**Get statement – public or friend function**

Function get total ( ) as currency

Get total = currency(txttotal.text)

End function

**Private**

Private sub form\_load( )

End sub

**Active . Exe project**

A class module

Private procedure can be called only from code within the same window

Private sub one ( )

Friend sub two ( )

Code in another module of your application can called friend and public

Private sub three ( )

**Parameter list and return values**

Sub function procedure can accept arguments and function also return a value

Eg:

Integer , Boolean, long , byte etc

Argument value past three procedures where as parameter is a value received by it

**Passing by value or by reference**

By val

By Ref

An argument can be passed by value using the value keyword and reference

Arguments passed by reference can be modify by the called procedures

The modify values can be read back by the caller arguments . that must be modify by the procedures

**Syntax**

*Sub draw point ( by val x as long, by val y as long , by val z as long)*

*If x< 0 then x = 0*

*If y< 0 then y = 0*

*If z< 0 then z = 0*

*End if*

*End sub*

Example

Draw point 10,20 ,40

Draw point x\*z, y\*z, z\*z

Draw point x,y,z

Declaring parameter by value fall the procedure passing a variable are an expression of any type

Declaring by reference3 pass a variable and this type must match

Eg:

1. Draw point x,y , 100

Draw point x,y,z

1. Public sub clear field (frm as form)

……..

…….

End sub

1. Friend sub clear field ( frm as form)

…….

……

End sub

**Optional keyword**

1. Sub print data 1(text as string , optional variant)

If Is missing (color) the color = vbwhite

Forecolor = color

Print text

End if

End sub

1. Sub print data 2(text as string , optional as variant)

If is missing color then

Form1.font transparent = false

Form1.font transparent = false

Color = vbwhite

End if

Form1.forecolor = color

Form1.print text

If is missing (color) then

Form1.font transparent = false

End if

End sub

1. Sub print data3 (text as string, optional color as long = vbwhite)

Form1.forecolro = color

Form1.print text

End sub

End sub

If a procedure receives ‘0’ value , empty string (or) nothing depending on the type of the parameter the data type cant be used with optional key or user defined data types

Function accepts only value the range 0 through 65,535

Function missing value (optional don’t pass this as variant) as variant

Missing value – don’t pass this

End function

**Named argument**

Eg

Err.raise 999 , “ value out of range”

Err.raise number = 999

Description = “value out of range”

**Syntax**

*Sub int(optional name as string .optional dept as integer , optional salary as currency)*

*………………..*

*……………….*

*End sub*

Int name= “guru” , salary=8000

Or

Int name = “ramya” ,dept id = 012

**The parameter keyword**

(name as string ) as double

Function sum(parameter array( ) as variant) as double

Dim I as integer

For I = 0 to UBound (arrays)

Sum = sum+arrays(i)

Next

End function

* Accept any number of arguments using paramarray keyword . there are 3 various types of keyword . they are only one paramarray keyword and it must be at end of the parameter list, only one variant type and no option parameter can proceed the paramarray keyword

**Error handling**

There are 3 types

1. On error resume next statement

Ignore any error that accuse in error function are ignore

1. On error goto label

Any error will cause and jump to the named label

1. On error go to “0” statement

Cancel the effective of any active on error. Resume next on error go to statement

**Exit error routine**

1. Resume statement
2. Resume next statement
3. Resume line statement
4. It executes the retry line of code that cause the error
5. It executes a resume execution in the procedure body at the line immediately after the one that cause the error
6. It is executes resume executable at given line in the procedure body

**Syntax**

*Dim ctrl as control*

*Property (timer don’t)*

*For each ctel in form1.controls*

*Ctrl.visible = false*

*Next*

If any error occurs the execution continuous with the next statement in the procedures which ever next statement

**Unhandled errors**

On error resume next

On error go to line statement

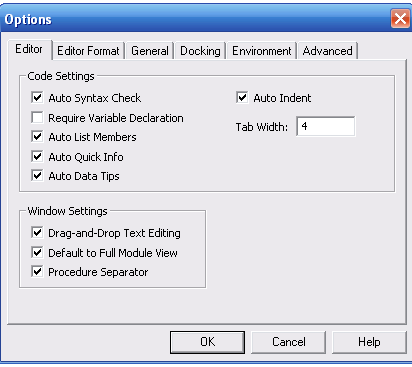
When either one of the statement is currently active the procedures is said to have an active error handles

If the procedures has been called by another procedures VB immediately terminates the current procedure and report the error to the calling procedures

If no procedure on the stack has an active error handler VB has n code to modify the error so it immediately stops the program with an error

All event procedures such as form load, command click are not called

**Error handling inside the VB IDE**



**Break on all errors**

All errors stop execution as soon as they occurs

**Break in class module**

All error in class modules stop execution as soon as they occur before they return to the calling code . support if it is as calling code . it should be specified in public and its located in another process

**Break on unhandled error**

Error stops execution only if there are not handle any where in program

**VISUAL BASIC APPLIATIONS AND VISUAL BASIC LIBRARIES**

Branch statement

Loop statement

Multiple statement

1. If ……. Else…… else if……… end if

If x>0 then y=x

X=0 else y=0

Eg

* 1. If x>0 then

Y=x

X=0

Else

Y=0

End if

* 1. **If ….. elseif …..end if**

If x>0 then

Y=x

Elsif x<0 the

Y=x\*x

Else

X=-1

End if

**Single statement**

1. Without else

If x>0 then y

1. With else

If x> 0 then y= x else y=0

**Multiple condition using AND or ORoperator**

* If x=0 AND y=0 then

If (x or y) = 0 then

* If x< >0 AND y< >0 then

If (x (or)y) < >0 then

**XOR**

If (x<0 and Y>=0) or (x>=0 and y< 0)

If (x XOR y) < 0 then

**Not operator**

If NOT (x = y) then

**Workin gwith numbers**

**1.math operators**

Abs

Sgn

Sqr

Exp

Log

Abs - it returns absolute value of its arguments

Sgn - it returns (-1) or (+1) respectively

Sqr - it returns the square root of numbers

Exp - raise in the base of neutral logarthims

Log - returns the natural logarthims of its arguments

**2.comparision operator**

**< > ,<= ,>=, < >**

**Boolean and bitwise operator**

AND,OR,NOR

**Unit – IV**

**DIFFERENT TYPES OF CONTROLS**

1. [DAC ] – Data Access Object
2. [ADO] – active x data object
3. [RDO] – remote data object
4. [ODBC] – open database connectivity
5. [OLE] – object linking embedded

Data converting word into visual basic

**Student mark statement**

* Create a student mark statement in word
* Copy the mark statement to excel work sheet
* Then put auto sum or total for that mark statement
* Open ms access -> blank statement -> create new blank database
* Saves that database in database 2003
* Go to menu bar -> click external data -> click excel details to msaccess
* Then double click the sheet 1. it will shows all the details in ms access

**Data environment**

Project menu - > data environment - > ms data environment -> instance 1.0 library

Property window

**Custom property**

Data environment – sql empcmd – empno, empname, EMP add, empid - child

**Data view**

View menu – data view

**4 connections**

* Provider
* Connect
* Advance
* All

Cmd object - > light clicking the database or right click – select command object

Project – menu – ADO data environment

**Format**

Form 1 design

In access

Ms Access – file – new – blank database – no primary key – id auto number

Save that table – close table

In visual basic

Go to menu bar – project – components – select ADO or OLE control – apply – ok

Right click on the tool box – project properties – components – select ADO /OLE control – apply – ok

After clicking ok – the new object will shows on the tool bar

Select – Microsoft jet 4.0 versions – built – next – apply – ok

After that selection the option button don’t show to alter that form

**ADO control selection**

Project window – project page – name – apply – ok

In vb

Text1 – click properties windows in datasource select in ADODB controls – and click datafield

Select username or Empid

Text 2 – click properties windows in datasource select in ADODB controls – and click datafield

Select in that password or empname

After that click ADO button . it will shows that process of ADO or details in access wat we done in that

**Crash course in SQL**

DDL – data definition language

DML – data manipulation language

To retrieve and update data stored in data based

Select command

Select \* from table name ;

Select pubid, company name , address from publishers

Select author, 2000 [year 2008 born] as age from authors

**Aggregate functions**

Select count \* as author cnt avg as from authors

Combine multiple conditions using AND , OR

**Boolean operators:**

Select \* from table name where state = “T” AND city = “C”

Comparision operators

<,>,<=,>=,=, like , between & in

Select \* from table name where state = “T” and name like city =”C”

Select \* from titles where [year published] between 1996 AND 1998;

Select name, statefrom publishers where state IN(‘CA’,’A’,’CS’);

**SORTING & GROUPING**

ORDER BY (only sorting numbers (or) fields)

Select \* from publishers order by [company name];

**MULTIPLE SORTING**

Select \* from publishers order by state , city,desc;

**SORTING AND GROUPING**

Select top 5 \* from publishers

Descending order with year

Select top 5 \* from titles order by [year published] desc;

Percentage

Selct top5 percentage \* from publishers

**Grouping by**

Select [year published] ; count \* as titles in year from titles grouping by [year published];

Before 10 years to update

Select top 10[years published] ; count (all) as title in year from titles grouping by [year published] order by [year published ] desc;

**Sub queries:**

To avoid ambiguities when the two tables have fields with the same name

Examples

The table name .field name which retrives a value which you can use to the left of the equal operator

Select \* from titles where pubid=[selct pubid] from publishers where name = “vb”

**Join**

The join operator is used to retrieve data from two tables that are related to each other through a common field . the result of the join is a new table whose rows consist of some all the fields from the first table followed by some all the fields from the second tables

The expression in the class in a join command determines which rows from the second table will match a given row from the first table

The most common form join operator is left join and right join

The left join operator retrieve all the record in the first table

**Syntax**

Select titles.titles .[year published], publishers.name from titles left join publishers on titles .pubid = publishers.pubid

The right join operator retrieves all the records in the second tables

**Union**

Select name , address , city from customers union

Select company name , address, city from suppliers.

Two tables can have different structures provided that the fields return by each selct command of that the same time

**Insert**

Insert into command a new record to table and sets its fields in one operation

**Syntax**

Insert into author [author,[year born]]values [1990]

Insert into authors select \* from new authors

The following copies all the record from a table called new authors into the authors tables

**Update**

The update command modifies the vcalue in one or more records

**Syntax**

Update authors set [ year born – 1961] where author = ‘fransico’

**Delete**

Delete command remove one or more records from a table

* Delete from titles[year published <1950]
* Delete from employees where employees last name = “guru”]

**UNIT – V**

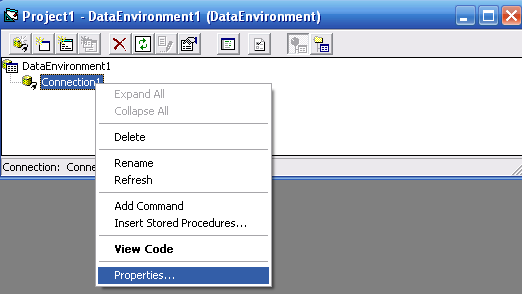
**DATA REPORT DESIGNER**

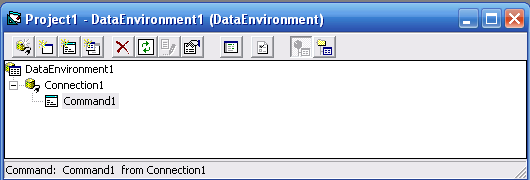
Standard Exe – project – data report – data report environment – 6 controls

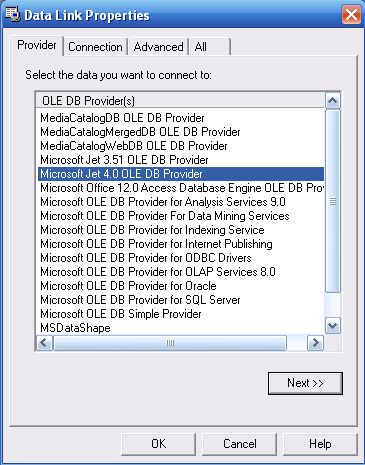
1. rpt text box
2. rpt label box
3. rpt image
4. rpt shape
5. rpt line
6. rpt function

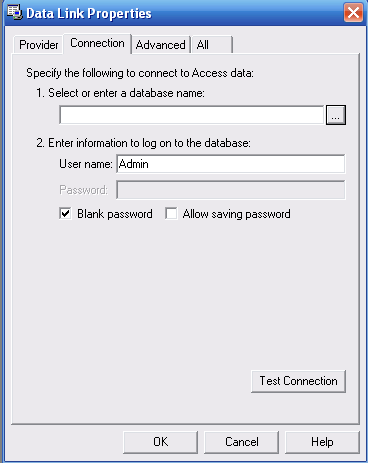
project 1 – data report

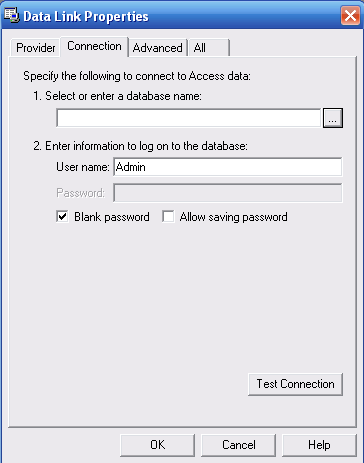
project menu- data environment designer – form window will open in default

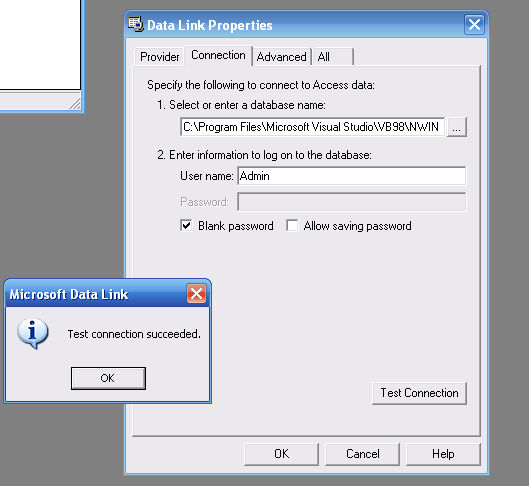


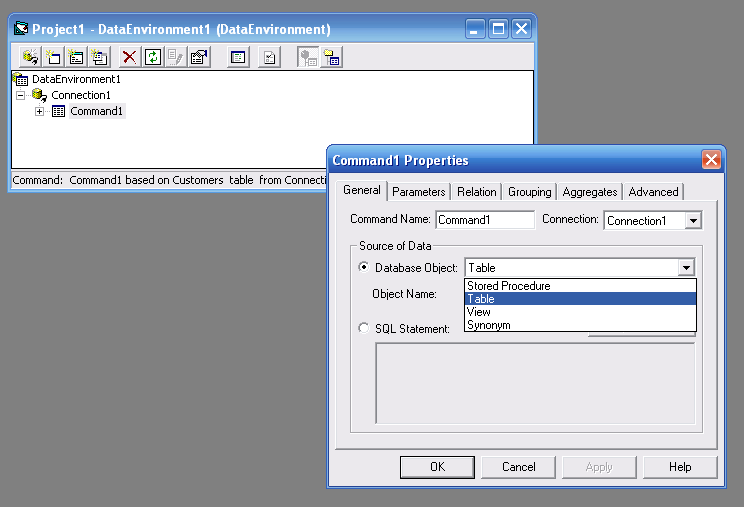


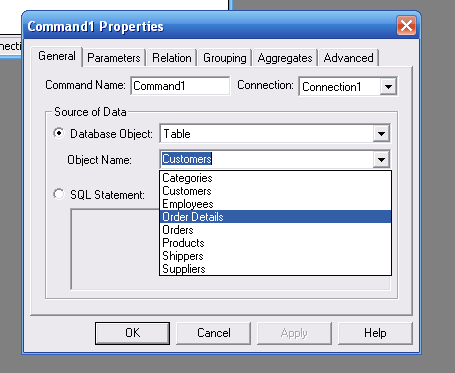


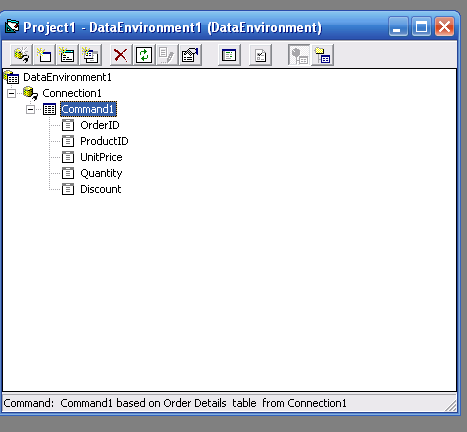


****

****

****

****

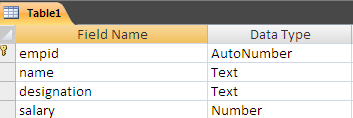
****

**DATA BINDING**

**1.simple binding**

**2.Simple bound binding**

Access empid, name,designation, salary



In vb

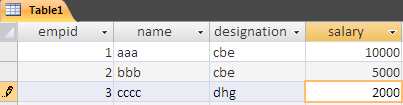
Toolbox – data tab- dropdown box- oleconnection button

**data binding**

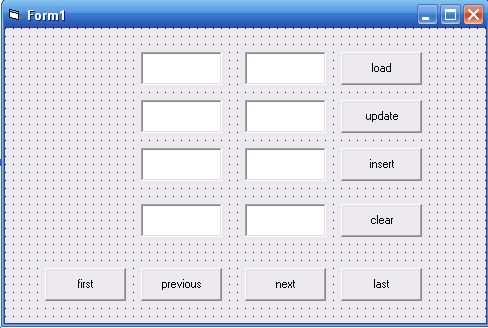
* Data binding is a binding control to add from the database
* Databinding we can bind a control to a particular column in a table from the database
* We can bind the whole table to the datgrid
* Data binding provides simple convenient and powerful to create a read, write, link between the controls on a form and the data in their applications
* Window form support binding data at ADO
* A control can be bound to any collection that support index access to the elements in that collection

**Simple data binding**

* To display one data element from a table in a control simple binding is managed by the simple binding collection of each controls
* Simple bound controls show only one data element at a time
* In data binding property the properties are working with examples
* We will create access and access data from table with form
* To start to open a blank document in ms access and save it

****

* Once you finish creating the table with required columns and enter some values and close it
* In vb new window forms are open and form the tool box add ‘8 cmd buttons’,4 label boxes, 4 text boxes
* Here we will bind data from the table we create in access to text box . if text box will display Empid, and text box , empname etc



* Set the text and name property of all the buttons now the tool box click the data tab and drag ‘oledb connection object ‘ on the form
* Once the connection is object is added to the component tray open its properties window to set the connection string
* Select connection string property in the property window – click on the drop down and select new connection
* When you select new connection it opens the data link properties in that you click provider tab and select the **microsoftjet4.0OLEDBProvider**
* By default it select provider for SQL server and click next
* Next you takes to the connection tab here browse EMP.MDB database on the selection area by the clicking the ellipses  button.

**Codings**

Private sub form1\_load(by value sender as system object , by val sender as system

arguments)

Handles my base load

End sub

Private sub load\_click()

Oledbdataadapter1.fill(dataset11)

End sub

Private sub update\_click()

Try

Me.binding context (datset11,”table1”)

Me.oledbdataadapter1.update (dataset11)

Catch

End try

End sub

Private sub insert\_click( )

Me.binding context(dataset11, “ table 1”).addnew( )

Msg box(“successfully inserted”)

End sub

Private sub clear \_click( )

Text1.text = “ ”

Text2.text = “ ”

Text3.text = “”

Text4.text= “”

End sub

Private sub first\_ click( )

Me.binding context (dataset11, “table1”).position=0

End sub

Private sub next \_ click( )

Me.binidng context (dataset , “table 1”)position = 0

Me.binidng context (dataset , “table 1”)position +1

End sub

Private sub previous \_ click ( )

Me.binidng context (dataset, “table 1”) position = 0

Me.binidng context (dataset, “table 1”) position - 1

End sub

Private sub last\_click ( )

Me.binidng context (dataset, “table 1”) position = 0

Me.binidng context (dataset, “table 1”) count -1

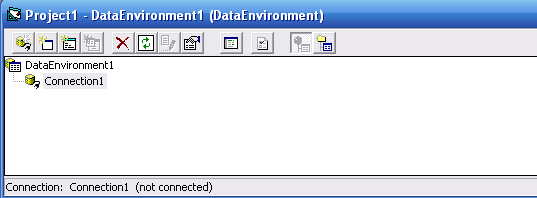
End sub

**DATA REPORT**

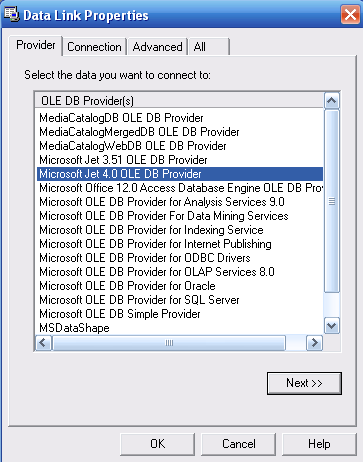
Step 1: adding data report

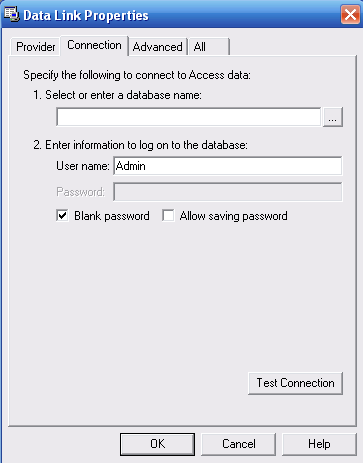
* Vb as standard Exe project from the data report in the drop down mwnu
* You will be presented with a datareport environment
* The data report environment contains 6 controls they are
  + rpt text box
  + rpt label box
  + rpt image
  + rpt shape
  + rpt line
  + rpt function
* title to the page header using the report label as RPT label
* simply drag and draw the Rpt label control on the data report designer window and use the caption property to change the text that should be displayed
* you can also add graphics to the report using the rpt image control

Step : 2 connecting the report to database using data environment designer . click the project menu and then select data environment from the drop down menu . the default data environment will appear



Now to connect to the data base right click connection1 and select Microsoft jet 4.0 OLEDB Database provider

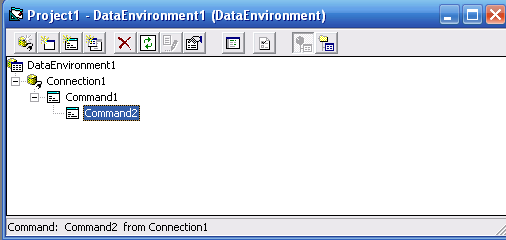




Step: 3 retrieving information from the database in order to use a database in your report. You need to create a query to retrieve the information from the database.

* Right click on my connection to add a command to the data environment

Step: 4 the default command1. Rename it is my command

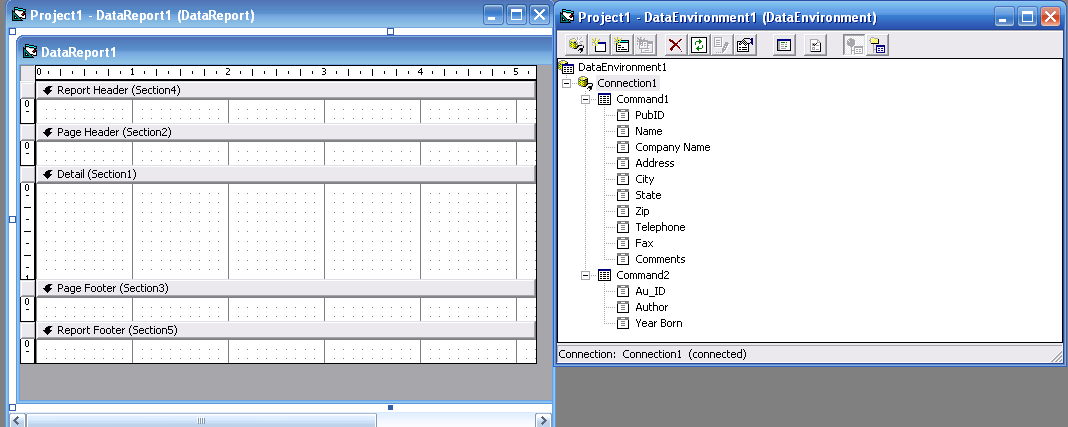


In order to use SQL command right click my command and you can its property dialog box at general tab - sql command and key in the SQL statement

This command is use to select all the field from the database

Now you need to customize of few properties of your data report , so that it can connect to the database

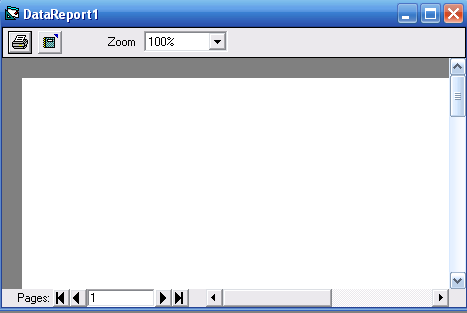
The first property to set its data source . the second property set my data environment next you need to set the data member property to my command . to add data to a report you need to drag in the fields from my command in my data environment into my data report as



Vb automatically draw a Rpt text box along with RPT label control for each field on the report

You can customize the labels as well as the etxt box fom the property of my data report

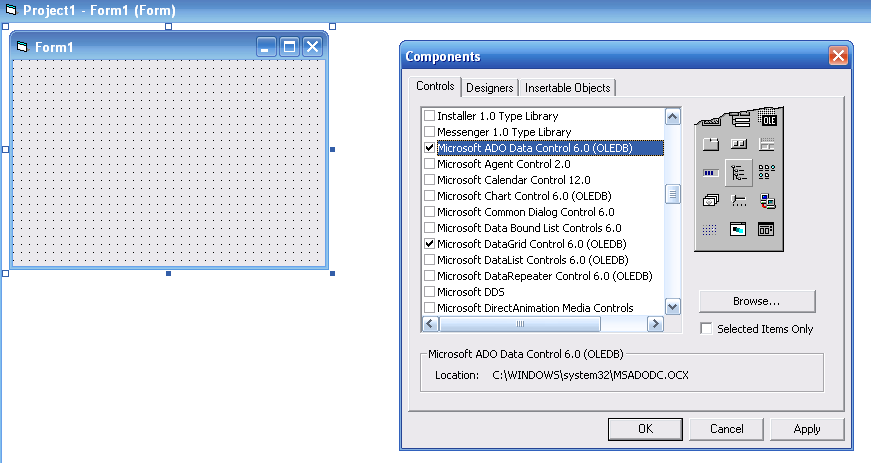
In final step to set my data report as the start up from the project menu

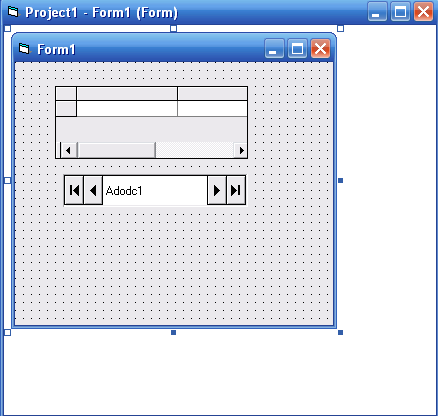


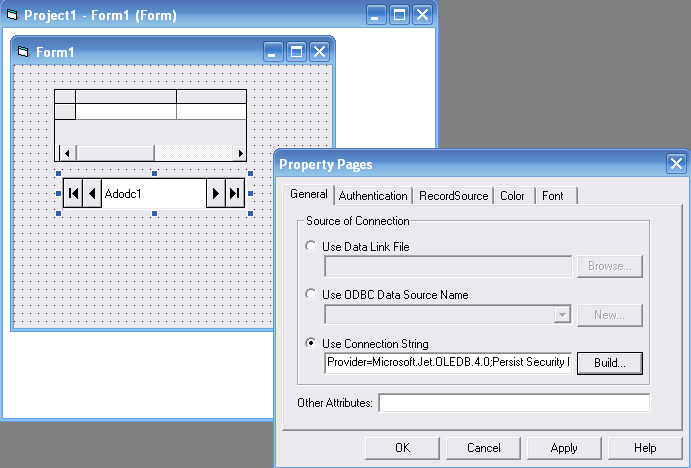
**Data grid control**

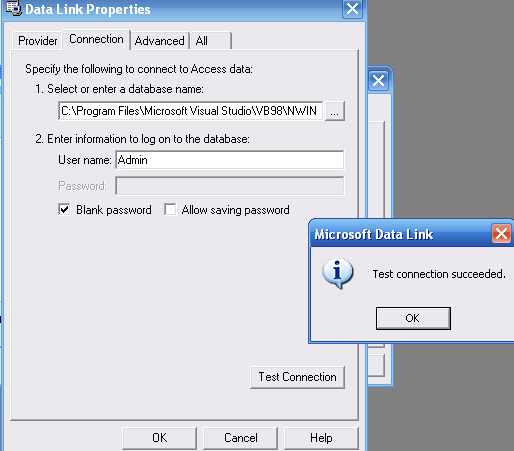
Project – components – msdata grid control – apply – ok

Project – components – msadodc – apply – ok



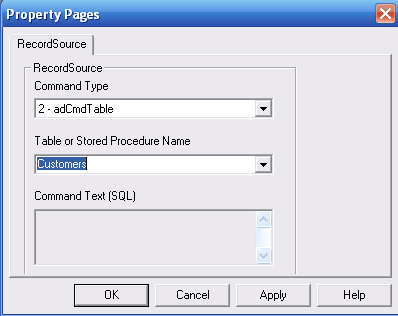






ADO control properties – record source

ADODC property – click – page dialog box – record source – dialog box will be appear on the screen



Set record source – data grid control

To display the entire table of a record set of database . it allows user to view and edit the data

Project – components – msdata grid control 6.0 – apply – ok

And select ADO control able to access the database and click ok to exit dialog box

In tool box data grid control and Ado control will be seen

To drag the dat grid control and ADO data grid control in the form

Drag the selected data from the tool box and paste it into form design

To connect database to the ADO data control right click on the ADO control and select the ADODC properties

The dialog box will be appear and click on the bulid button and next to another property will appear . in this dialog box select database file we have created

Press test connection successful . if connection is successful click ok to return to the ADODC property page dialog box . in that dialog box click record source and select the table name and stored procedure and then click ok

Finally you need to display the data in the data grid control . to accomplise this go to property of the data grid to the record

**EXPLAIN ABOUT SAGA**

* DAO
* RDO
* OLE
* ADO

ADO is a active database object is a programming model which means that it

is not dependent upon any given back and end. Currently the only engine supporting the

ADO model is OLEdatabase

**ADO OBJECT MODEL**

It consist of 60 objects

* Connection – represenst and open connection
* Error – contain details about data access error
* Command – defines a specific command
* Parameters – optional collection of the cmd object for any parameter
* Record set – it represents a set of record of set from a table , cmd object
* Field – it represent a single column of data in a record set
* Property – a collection of values raised by the provider for ADO

**DAO object**

It is an application program interface available with Microsoft Vb that let a programmer request access to ms access database . ADO was Microsoft first object oriented interface with database , ADO object encapsulation access jet function through jet function it analyse access other SQL database

**OLE DATABASE**

It is an MS strategic low level application for access to different data source ole database includes not only the sql capability of ms sponser standard data interface open database connectivity but also includes access to data

As a design form ms components object models

OLEDB is a set of methods for reading and writing data

The objects in OLEDB consist of data source object, a session object, a command object, a record set object.

**RDO ( Remote Data Access Object)**

RDO object and collections provide a frame work for using code to create and manipulate components of remote

ODBC vb system object collection have properties that describes the characterstics of database components and methods that you use to manipulate them

**RDO object model**

Rdo engine

Rdo environment

Rdo connection

Rdo query

Rdo parameter

Rdo column

Rdo prepare statement

Rdo column

Rdo tables

Rdo column

Rdo result set

Rdo parameters

Rdo connection

**RDO Engine**

In the basic object create automatically when access RDO in your application

**RDO Error**

It is used to handle all ODBC Error and message generated by RDO created automatically

**RDO Environment**

A logical set of connection & transaction scope for a particular user name contain both open allocated connection provider and provide the security context for DML operations on tha database RDO environment created automatically

**RDO Connection**

It represents an open connction to a remote data source and specific database on that data source or an allocated as unconnected object

**RDO Table**

It represents the stored definition of a base table or an SQL view

**RDO Result Set:**

It represent the row that result from running a query

**RDO Column**

It represents a column of data with a common data type and a common set of properties

**RDO Query**

An SQL query definition that can includes 0 or more parameter

**RDO Parameter**

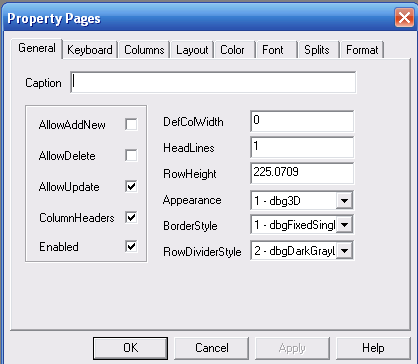
It represent a parameter associated with an RDO query object, query parameter can be input or output or both

**HIERARCHICAL FLEX GRID CONTROL**

Project – components – ms hierarchical flex grid control – apply – ok

1. It can merge contiguous cells in different rows . if they contain the same values .
2. When you assign a hierarchical ADO record set to its data source property , it display multiple bands
3. Each band set data column that come from a different child record set .
4. To create hierarchical flex grid control is to built a hierarchical command object in a data environment designer use the right mouse button to drop it on a form and select the hierarchical flex grid command from the pop up menus
5. Like data grid control the hierarchical flex grid control expose an object model

**General tab**



The general tab assigns a value to the controls rows, and columns properties . it determines the number of rows and columns in the grid.

The appearance will effect the control only in unbound mode

The dimension of the grid depend on the number of record and fields in the sources , fixed rows and fixed columns

The fixed rows and fixed columns properties effect , how many fixed rows and columns display at the left and on the upper border of the grid

**Allow big selection property**

Allow big selection property is true . clicking on a row or column , header select the entire row or column

**High light property**

It effect the appearance of the selected cells and can be any one

0 – flex highlight viewer

1 – flex highlight always

2 – flex highlight with focus

**Focus Rect property**

Determines which kinds of border appears around the current cells

0 – flex focus none

1 – flex focus light

2 – flex focus heavy

**Band display property**

It can change how the bands are displayed in the control and can be either

1. - flex band display horizontal
2. – flex band display vertical

**Fill style property**

In this property all the selected cells will be effected by the assignment

0 – flex fill single to one flex will repeat

**Selection mode property**

Decides you can select any cell

1. – flex selection free
2. – flex selection by row
3. – flex selection by column

**Allow user resizing property**

The user can resize rows or columns or with a mouse

1. – flex resize none
2. – flex resize column
3. - flex resize rows
4. - flex resize both

**Row resizing mode property**

1. – flex resizing individual
2. – flex resizing all

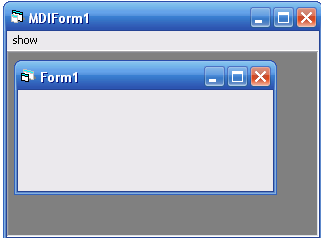
**MULTIPLE DOCUMENT INTERFACE [MDI]**

Save – MDI.vbp, parent.frm, child.frm

Private sub mnnew\_click ( )

Dim new form as new form1

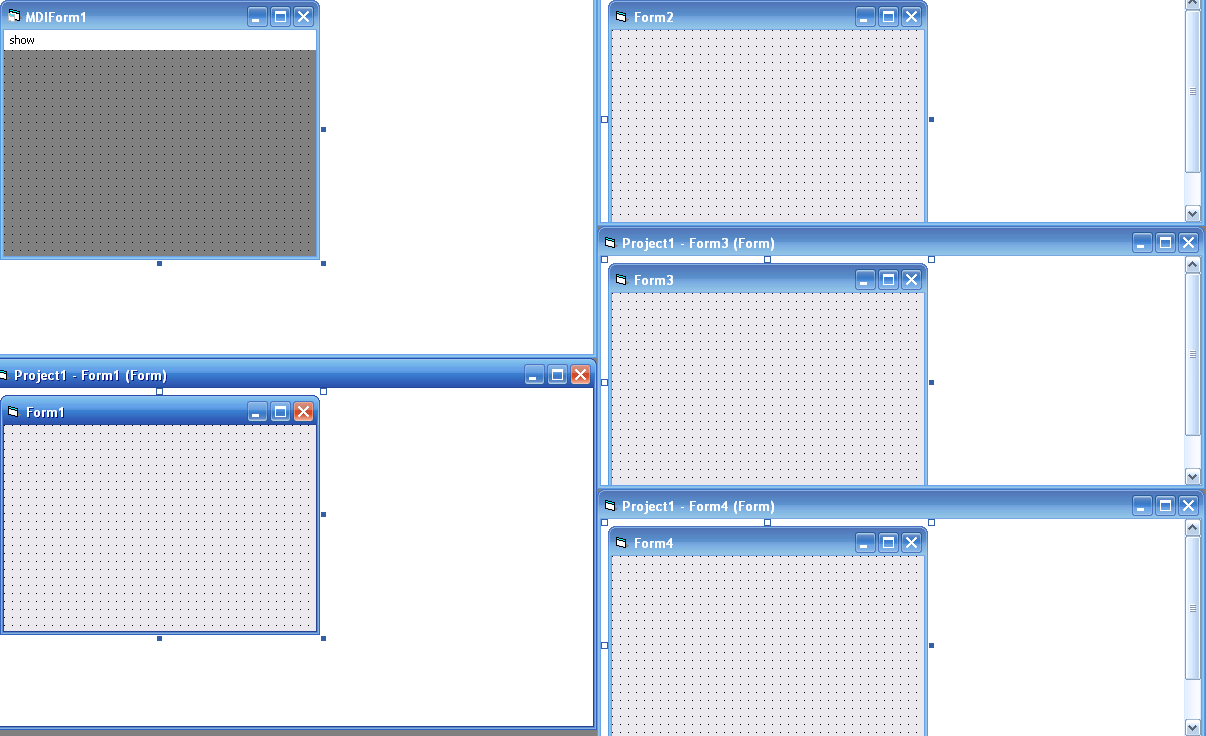
End sub

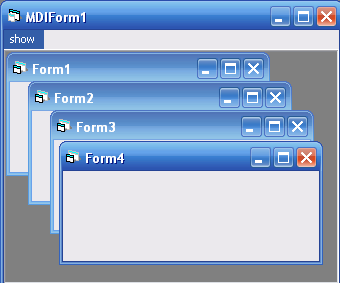


Private sub mnucascade\_click ( )

Form1.windowstate = vbcascade

End sub





MDI stands for multiple document interface used for opening many windows at the same time . child window are contain in a parent window .

VB application has one MDI form which contains all child forms

The child window is displayed within the internal area of MDI forms

**Creating an MDI applications**

MDI can be designed for document in centered applications

It helps to open many similar documents at the same time

We required at least two forms one MDI form and another one child form.

Project – add MDI form

The project is saved as MDI.vbp the form is saved chid.frm . the MDI form is saved as Parent .frm

**Adjusting the text box**

The text box in the child form can be adjust by using resize event

The resize event is fixed whenever the size of the form is changed

Private sub form resize ( )

Me.text1.height = me. Scale height

Me.text1.width = me. Scale width

End sub

Creating a tool bars most window programs contains tool bar, provide quick access to the commonly used operation

Tool bar is also called as ribbon bar or control bar

To add tool bar , select MDi form, picture control in the tool box is double clicked

The picture box control in the MDI form cannot drag to another location in the form

Tool box are placed in picture

**Creating and using controls**

1. **Text box**

A text box controls sometimes called an edit field or edit control display information entered at a design time

**Properties of text box**

* Text can be entered into the text box by assigning the necessary string to the text property of the control
* Multiple lines of the text in a text box controls and to set the multiple line property is true
* If a multiple line text box does not have horizontal scroll bar, text wraps automatically even when the text box is resized
* In order to customize the scroll bar combination on a text box set the scroll bar properties
* Scroll bar will always appear on a text box when its multiple property is set to true and its scroll bar property is set to anything except none or 0
* Text box controls can also act as a designation link in a DDE conversion

**Sel length ( )**

Returns are select or set the no of characters selected

**Sel start ( )**

Returns are sets the starting pointing selected text

**Sel text ( )**

Returns are set the string contains the currently selected text

1. **Label control**

Label control is a graphical control you can use to display text that a user cannot change directly

**Properties of label control**

* We can write code that changes the caption property display by a label control in response to event at run time
* The auto size and word wrap properties should be set if the user wants to label to property display variant length lines or varying no of lines
* A label control can act as a designation in a DDE conversion
* Link topic set the conversion, link mode activate the link . link item is specify an item for the properties have been set
* Use property tot rue if you want to define a character in the caption property of the label as an access key

1. **Command button**

A command button control is use to begin in drapt or end a process

**Properties of command button**

* To display text on a command button control set its caption property
* An event can be activated by clicking a command button
* A button background color can be set by setting the back color property to the desire choice
* The text color of a command button can be set using the fore color property
* The button can be enabled or disabled by setting the enable property to true or false respectively
* The button can be may visible or invisible at runtime using the visible property

1. **Option button control**

Option button provides a set of choice form which user can select only one button by clicking it at run time , assigning the value property to true. In the code like option1.vlaue = true, using the shortcut keys specified in the caption of the label

1. **Using list box and combo box controls**

List box and combo box controls present a set of choice that displayed vertically in a single column

If the no of items exist the value that can be displayed scroll bar with automatically appear on the control

**Adding items to a list**

* **Design time**

To add items to a list at design time click list in the property box and then add the items. Press ctrl+enter after adding each item

* **Run time**

The add item method is used to add item to a list at run time. The add item object

Add item\_ ( ) item, index

The item argument is a string that represents the text to be added to the list

The index argument is an integer that indicates where in the list to add the new item

**Removing item from the list**

* Remove item method is used to remove an item from a list

**Syntax**

Object .removeitem index

* Selecting an item from list box
* List index property sets the index number of the currently selected item
* List count property is always one more than the index number of the last item in the list
* New index property set the index of the last item added to the list

Example

List1.additem [television]

List1.listindex = list1.new index

**Sorting the list**

The sorted property is set to true to enable a list to appear in alpha numeric order and falls to display the list items in the order in which they are added to the list

**Combo box**

Combo box combines the features of text box and list box. This control enables to the user to select either by typing the text on the combo box or by selecting a item from the list

**Simple combo box**

It display an edit area with an attached list box all ways visible immediately below the edit area

**Drop down combo box**

It can edit area with a down arrow button at the right

The list portion stays hidden until the user clicks down button to drop down the list portion

**Drop down list**

At run time the controls like drop down comb box. The user click the down arrow to view the list

The difference between drop down combo and drop down list combo is that edit area in the drop down list is disabled

**Scroll bar controls**

The scroll bar is commonly used control which enables the user to set a value by possessing it at the desire locations

The minimum and maximum property represents the minimum and maximum values

The value property of the scroll bar represent its cursor value which may be any integer between minimum and maximum values

Program:

Private sub hscroll1\_ click( )

Text1.text = Hscroll1.value

End sub

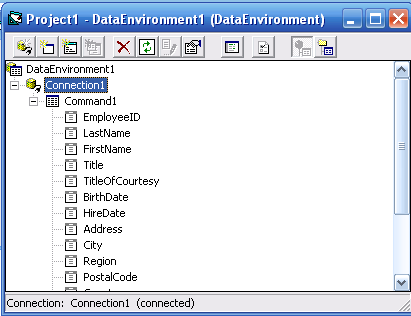
Private sub command1\_click( )

End

End sub

**DATA ENVIRONMENT DESIGNER/ CONNECTION COMMAND OBJECT / IN REPORT**

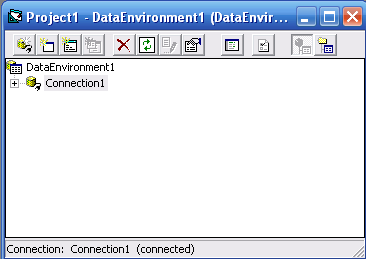
Data environment designer is a powerful sophisticated which provides an interactive design time environment for creating programs at runtime access to data. The connection command can be set at design time.



Accessing data using data environment

Data environment designer provides an essay means to access. Data in visual basic 6.0 project. Data environment designer support all 3 types of data access DAO, RDO and ADO. Data environment must include at least one connection object in it. A connection object represent a connection to a database that is used at the data source for associated commands like, tables & queries.

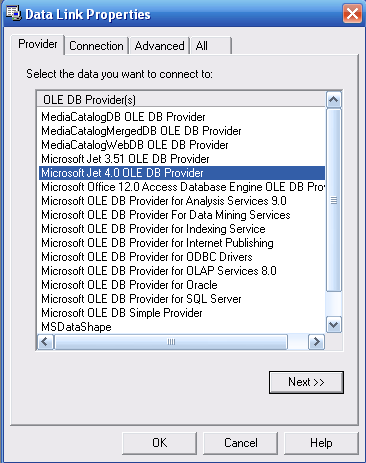
When ever data environment is added to visual basic project. A new connection by the name connection1 is created by default.



Connection object:

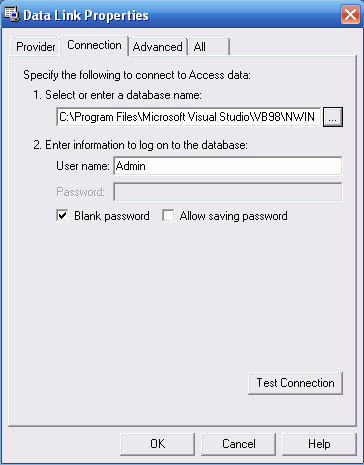
Connection object has 4 tabs

1. **Provider**



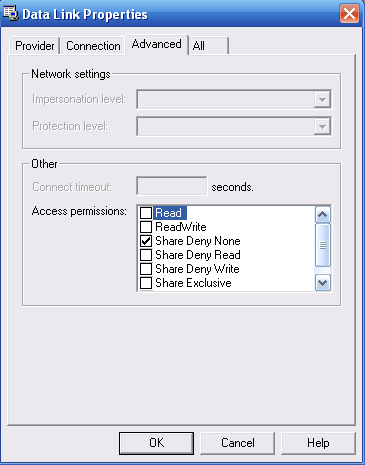
1. **Connection**

Use to select the database provide the user id and password appropriate to the requirements of the user



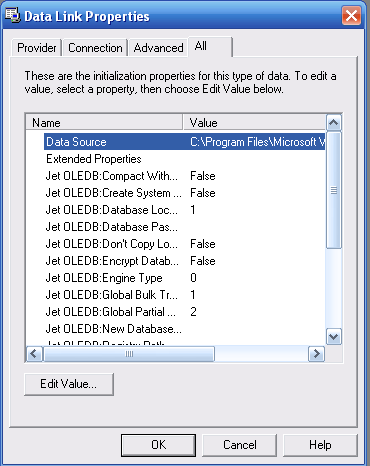
1. **Advanced**

Used to set time out parameter



1. All

Used to view a summary of the connection object settings



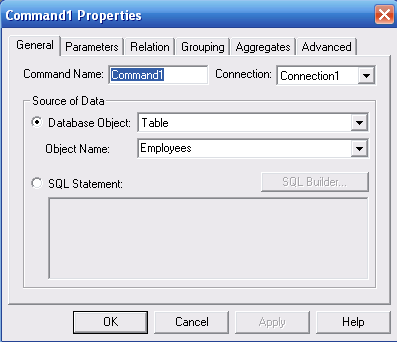
**Command object**

A command object is a standard active x data object

This is added to the connection object by checking add command object from the list of options

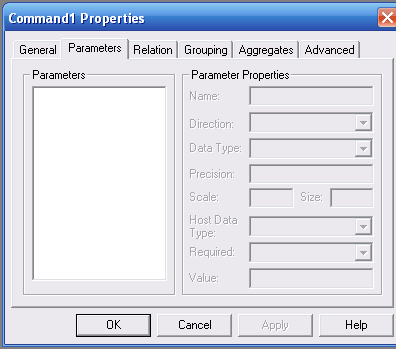
Command object has 5 tabs

1. **General**



This tab is used to rename the object or change the connection object

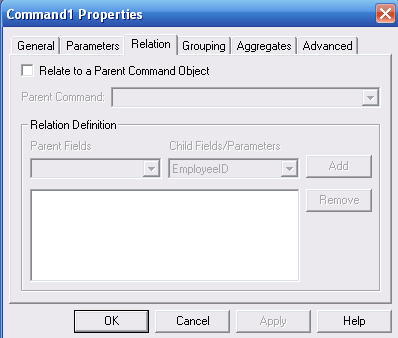
1. **Parameter**



this is used to display all parameters associated with it

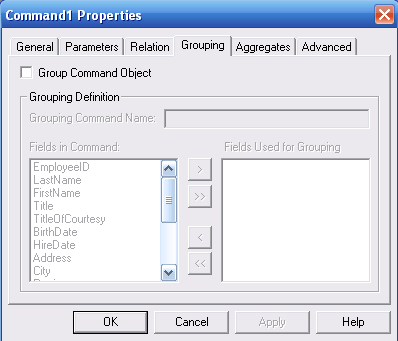
1. **Relational**

This allow the develop to associated command object with each other in hierarchical relationship



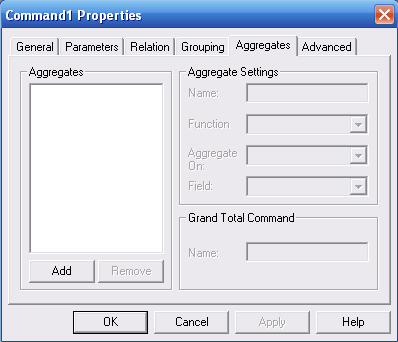
1. **Grouping**

This tab allows the user to specify one or more columns on which the result is to be grouped



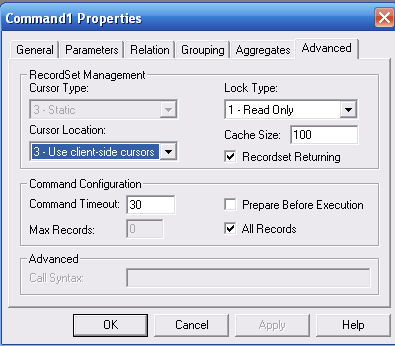
1. **Aggregate**

This tab is to aggregate the data based on one or more several functions type



1. **Advanced**

This tab has different option available depending on the data source



**UNIT – 5**

**PROCESSING DATA**

Opening a record set object

Source string – select command

1. Dim rs as new ADODB recordset

If optsource ( 0) value then

rs.source = ‘authors’

Else if opt source(1) .value then

rs.source = ‘reptq’

Elseif optsource(2).value then

rs.source= ‘select \* from authors where all names like ‘A\*’

End if

1. Dim rs as new ADODB recordset

If optsource ( 0) value then

rs.open “publishers” ‘’ ‘’adcmd table

Else if opt source(1) .value then

rs.open “reptq” ‘’ ‘’adcmd store proc

Elseif optsource(2).value then

rs.open = “select \* from authors” ‘’ ‘’ adcmdtext

End if

**Cursors & concurrency**

**ADODB –** forward only, read only , static and dynamic

Coding

rs.cursor type = adopendynamic

rs.open = “select \* from authors” ‘’ ‘’ adcmdtext

rs.open = “select \* from authors” ‘’ ‘’ adopenkeyset

rs.open = “select \* from optimistic, adcmd text

**Stand alone record set object**

1. Dim rs.as new ADODB record set

rs.field.append “first name”, adchar,40,adid is nullable

cn.open “provider= sqloledb; datasource = p2;

cn.cursor loacation = acluseclient

set.rs = cn.execute(“authors”)

1. Dim rs.as new ADODB record set

rs.field.append “first name”, adchar,40,adid is nullable

rs.field.append “last name”, adchar,40,adid is nullable

rs.field.append “birthdate” addate

rs.open