AboutBox Method

Displays the About box for the control.

Syntax

[*form*!]*PopupMenu*.**AboutBox**

Remarks

This is the same as clicking About in the Properties window.

AppendMenu Method

Returns **True** if the Menu Item is Appended successfully..

Syntax

[*form*!]*PopupMenu*.**AppendMenu** *Caption*[, *itemdata*, *PopupMenu*]

The **AppendMenu** Method syntax has these parts:

Part	Description
Caption	A String that will be displayed as the Menu Item Caption. If **Caption** is Empty ("") a Separator is Inserted.
itemdata	A String that will be associated with the Menu Item and returned in the **Itemdata** Value of the **MenuClick Event**.
PopupMenu	(Boolean) Set to **True** if the Menu Item has a Sub Menu.

Remarks

Use this Method to add Menu Items to the Popup Menu whan used in programmatic mode.
It only makes sense to call this Method in the **GetMenuContent Event**.

DataCol Property

Returns or sets a value that binds the PopupMenu to a Column in the current recordset.

Syntax

[*form*!]*PopupMenu*.**DataCol**[= *value*]

The **DataCol** Property syntax has these parts:

Part	Description
value	A string expression that evaluates to the name of one of the columns in the **Recordset** object specified by a **Data** control's **RecordSource** and **DatabaseName** properties.

Remarks

Bound controls provide access to specific data in your database. The **PopuX** control is bound to one of the **Recorset** object's rows. Bound controls that manage a whole column typically display the value of a specific column in the current **Recordset.** The **DataSource** property of a bound control specifies a valid **Data** control name, and the **DataCol** property specifies a valid column name in the **Recordset** object created by the **Data** control. Together, these properties specify what data appears in the bound control.

When you use a **QueryDef** object or SQL statement that returns the results of an expression, the column name is automatically generated by the Microsoft Jet database engine. For example, when you code an SQL aggregate function or an expression in your SQL query, unless you alias the aggregate fields using an AS clause, the column names are automatically generated. Generally, the expression column name is Expr1 followed by a three-character number starting with 000. The first expression returned would be named Expr1000.

It's recommended that you code your SQL queries to alias expression columns as shown below:

```
Data1.RecordSource = "Select AVG(Sales)    " _
        & " AS AverageSales From SalesTable"
PopupMenu1.DataCol = "AverageSales"
Data1.Refresh
```

Note Make sure the **DataCol** property setting is valid. If you change the setting of a **Data** control's **RecordSource** property and then use Refresh
, the **Recordset** identifies the new object. This may invalidate the **DataCol** settings of the **PopupX** control and produce a trappable error.

Data Type

String

DataSource Property

Sets a value that specifies the Data control through which the **PopupX** control is bound to a database.

Syntax

[*form*!]*PopupMenu*.**DataSource**[= *value*]

Remarks

To bind a control to a field in a database at run time, you must specify a **Data** control in the **DataSource** property at design time using the Properties window or programmatically at run time..

To complete the connection with a column in the Recordset managed by the Data control, you must also provide the name of a column object in the **DataCol** property. Unlike most data aware controls, the **DataSource** property setting is available both at run time and design time.

Data Type

String

Enabled Property

Returns or sets a value that determines whether the **PopupX** control can respond to user-selections.

Syntax

[*form*!]*PopupMenu*.**Enabled** [= *boolean*]

The **Enabled** property syntax has these parts:

Part	Description
boolean	A Boolean expression that specifies whether the PopupX control can respond to user selections.

Settings

The settings for boolean are:

Setting	Description
True	Allows PopupMenus to respond to events.
False	Prevents PopupMenus from responding to events.

GetMenuContent Event
MenuClick Event

GetMenuContent Event

Occurs after the **ShowPopupMenu** Method. is called and before the Popupmenu is displayed.

Syntax

Private Sub *PopupMenu*_**GetMenuContent**(*MenuName* As String, *Caption* As String, *ItemData* As String)

The **DataCol** Property syntax has these parts:

Part	Description
MenuName	PopupMenu Name (String) requesting content.
Caption	Caption of the MenuItem selected by User
Itemdata	ItemData associated to the Selected MenuItem

Remarks

GetMenuContent is fired when **ShowPopupMenu** Method is called.With **ShowDBPopupMenu** Method the menu contents are provided automatically by the **Data** control.
GetMenuContent is also fired for every Submenu the Popup Menu contains, when the user points the mouse pointer to it. See the **AppendMenu** Method for information on creating Submenus.

ItemDataCol Property

Returns or sets a value that binds the PopupMenu to a second Column in the current recordset.

Syntax

[*form*!]*PopupMenu*.**ItemDataCol**[= *value*]

The **ItemataCol** Property syntax has these parts:

Part	Description
value	A string expression that evaluates to the name of one of the columns in the **Recordset** object specified by a **Data** control's **RecordSource** and **DatabaseName** properties.

Remarks

This propery provides the same functionality as the **DataCol** Property, except that the contents of the bound column are not displayed in the Popupmenu. It allows the programmer to store hidden values associated with each Menuitem. See the Databound sample program for an example on how to use this property.

Data Type

String

MenuClick Event

Occurs when a Menuitem is selected by the user.

Syntax

Private Sub *PopupMenu*_**MenuClick**(*MenuName* As String, *Caption* As String, *itemdata* As String)

The **MenuClick** Event syntax has these parts:

Part	Description
MenuName	PopupMenu Name (String)
Caption	Caption of the MenuItem selected by User
Itemdata	ItemData associated to the Selected MenuItem

Remarks

none.

AppendMenu Method
ShowDBPopupMenu Method
ShowPopupMenu Method

PopupX Control Reference

Intoduction

PopupX Control finally allows Microsoft Visual Basic programmers to fully exploit the power of popup menus. Until today VB programmers were restricted to the limited functionality provided by the intrinsic VB menuing system. Menus had to be created at design time,which makes it impossible to have dynamic run-time generated popup menus.

PopupX unleashes the power of dynamically generated popup menus and makes it easy for the programmer to create popup menus and respond to them at run-time.. And because **PopupX** is data aware, it can be bound to any database and provide a convenient representation for data selection.

Features

Simoultaneous bound and programmatic (unbound) modes.
Runtime switching of Datacol, ItemDataCol, and DataSources.
Unlimited number of Menu Items
Automatic or user defined Popup menu positioning
Automatic adjustment to System Font, Font Size and Color.

Overview

The PopupX Control can be used in two modes:

Bound mode:

In Data Bound mode you use the standard **Data control** provided with Visual Basic. to bind the Popupmenu to one or two columns of the database. For more information on the Data control refer to the Visual Basic Help. After setting the Data control's DatabaseName and Recordsource Properties, set the DataSource, DataCol and optionally ItemDataCol Properties of the PopupX control.
The ShowDBPopupMenu Method is used to actally display the Popupmenu. This Method can be called in response to any user action as long as no other menus are being displayed. After the user selects a Menu item, the MenuClick Event is fired providing you with information about the menu item that was selected allowing you to do any action in response.

UnBound mode:

Unbound or Programmatic mode allows you to diplay and number of cascaded Menus and Submenus. After calling the ShowPopupMenu Method, The GetMenuContent Event is fired. That is the time to fill your popup menu with Menu Items and Submenus. The GetMenuContent is fired once for every SubMenu abou to be displayed making it faster by only loading the Items that have been requested by the user. Once a Menu Item is clicked by the user, the MenuClick event is fired identically to the Bound mode operation. It is useless to day that the Unbound mode gives you much more control over what needs to be displayed where.

SQL statement

A complete phrase in SQL that begins with a keyword and completely describes an action to be taken. For example, SELECT * FROM Orders. SQL statements should not be confused with statements.

ShowDBPopupMenu Method

Displays a Popup menu with the contents of the bound column as MenuItems.

Syntax

[*form*!]*PopupMenu*.**ShowDBPopupMenu** *MenuName*, [*xPos*, *yPos]*

The **ShowDBPopupMenu** Method syntax has these parts:

Part	Description
MenuName	A String that will Identify the PopupMenu and passed in the **MenuName** parameter of the **MenuClick** Event**.**
xPos	Optional X coordinate at which the Popupmenu is displayed.in Twips
yPos	Optional Y coordinate at which the Popupmenu is displayed in Twips.

Remarks

Note

If a the required display area for a Popupmenu is larger than the available display area it is not displayed by the Operating System.

ShowPopupMenu Method

Displays a Popup menu after getting the MenuItems in the **GetMenuContent** Event.

Syntax

[*form*!]*PopupMenu*.**ShowPopupMenu** *MenuName*, [*xPos*, *yPos]*

The **ShowPopupMenu** Method syntax has these parts:

Part	Description
MenuName	A String that will Identify the PopupMenu and passed in the **MenuName** parameter of the **MenuClick** Event**.**
xPos	Optional X coordinate at which the Popupmenu is displayed.in Twips
yPos	Optional Y coordinate at which the Popupmenu is displayed in Twips.

Note

If a the required display area for a PopupMenu is larger than the available display area it is not displayed by the Operating System.
Sub menus can have up to 1400 Items even is the display area isn't large enough to display all of the items.

Boolean expression

An expression that evaluates to either True or False.

bound control

A data-aware control that can provide access to a specific field or fields in a database through a **Data** control. A data-aware control is typically bound to a **Data** control through its **DataSource** and **DataField** properties. When a **Data** control moves from one record to the next, all bound controls connected to the **Data** control change to display data from fields in the current record. When users change data in a bound control and then move to a different record, the changes are automatically saved in the database.

design time

The time during which you build an application in the development environment by adding controls, setting control or form properties, and so on. In contrast, during run time, you interact with the application like a user.

twip

A screen-independent unit used to ensure that placement and proportion of screen elements in your screen application are the same on all display systems. A twip is a unit of screen measurement equal to 1/20 of a printer's point. There are approximately 1440 twips to a logical inch or 567 twips to a logical centimeter (the length of a screen item measuring one inch or one centimeter when printed).