The Multiple Document Interface (MDI) was designed to simplify the exchange of information among documents, all under the same roof. With the main application, you can maintain multiple open windows, but not multiple copies of the application. Data exchange is easier when you can view and compare many documents simultaneously.

You almost certainly use Windows applications that can open multiple documents at the same time and allow the user to switch among them with a mouse-click. Multiple Word is a typical example, although most people use it in single document mode. Each document is displayed in its own window, and all document windows have the same behavior. The main Form, or MDI Form, isn't duplicated, but it acts as a container for all the windows, and it is called the parent window. The windows in which the individual documents are displayed are called Child windows.

An MDI application must have at least two Form, the parent Form and one or more child Forms. Each of these Forms has certain properties. There can be many child forms contained within the parent Form, but there can be only one parent Form.

The parent Form may not contain any controls. While the parent Form is open in design mode, the icons on the ToolBox are not displayed, but you can't place any controls on the Form. The parent Form can, and usually has its own menu.

To create an MDI application, follow these steps:

1. Start a new project and then choose Project >>> Add MDI Form to add the parent Form.
2. Set the Form's caption to MDI Window
3. Choose Project >>> Add Form to add a SDI Form.
4. Make this Form as child of MDI Form by setting the MDI Child property of the SDI Form to True. Set the caption property to MDI Child window.

Visual Basic automatically associates this new Form with the parent Form. This child Form can't exist outside the parent Form; in the words, it can only be opened within the parent Form.



**Parent and Child Menus**

MDI Form cannot contain objects other than child Forms, but MDI Forms can have their own menus. However, because most of the operations of the application have meaning only if there is at least one child Form open, there's a peculiarity about the MDI Forms. The MDI Form usually has a menu with two commands to load a new child Form and to quit the application. The child Form can have any number of commands in its menu, according to the application. When the child Form is loaded, the child Form's menu replaces the original menu on the MDI Form

Following example illustrates the above explanation.

\* Open a new Project and name the Form as Menu.frm and save the Project as Menu.vbp

\* Design a menu that has the following structure.

<> MDIMenu Menu caption

* MDIOpen opens a new child Form
* MDIExit terminates the application

\* Then design the following menu for the child Form

<> ChildMenu Menu caption

* Child Open opens a new child Form
* Child Save saves the document in the active child Form
* Child Close Closes the active child Form

At design time double click on MDI Open and add the following code in the click event of the open menu.

Form1.Show

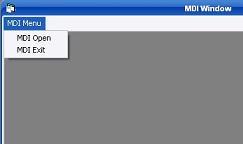
And so double click on MDI Exit and add the following code in the click event

End

Double click on Child Close and enter the following code in the click event

Unload Me

Before run the application in the project properties set MDI Form as the start-up Form. Save and run the application. Following output will be displayed.



And as soon as you click MDI Open you can notice that the main menu of the MDI Form is replaced with the Menu of the Child Form. The reason for this behavior should be obvious. The operation available through the MDI Form are quite different from the operations of the child window. Moreover, each child Form shouldn't have it's own menu.