ANNALS of the ORADEA UNIVERSITY.

Fascicle of Management and Technological Engineering, Volume VII (XVII), 2008

MDI APPLICATIONS FOR FINANCIAL FUNCTIONS

Emil COSMA,
"Ovidius" University of Constanța,Faculty of Economic Sciences,
e-mail: ecosma@univ-ovidius.ro

Keywords: Parent Form, Child Form, Multiple Document Interface, Financial Functions

Abstract: Within the **Visual Basic 2008** design environment, a *MDI* application contains a *MDI* form, also known as *parent form*. This could also contain many types of *MDI child forms*. All child forms must enter the *client zone* of the *MDI* parent form, a zone that is consisted of the form's space and does not include the title bar, the status bar, the panels and the parent form's border. The users may need to use a *MDI* application when they work with many forms or documents that have the same visual characteristics.

1. INTRODUCTION

One may distinguish two types of graphical user interface for the *Windows* applications:

- *Single* Document Interface (SDI);
- *Multiple* Document Interface (*MDI*).

Microsoft Excel is a *MDI* application (the documents that contain spreadsheets are the child forms). On the other hand, *Microsoft Word* is a *partial MDI* application: *Word* documents represent child windows, but sometimes, the *Word* application reacts more as a *SDI* application, rather that a *MDI* one due to the fact that every document is opened within a separate window program (and not all within a single window). Depending on how *Visual Studio 2005* is configured, it can be mainly a *MDI* application. Many child windows might be opened and become visible (the code editor, the *forms* design environment etc.) if one selects the *Multiple Documents* option (*MDI* environment) from within the menu *(Tools, Options, Environment, General).*

2. MDI PARENT FORM

The parent *form* within a *MDI* acts as a container for a *MDI* application. Usually, one defines a single parent form and many child forms (it is not necessary for the child forms instances to be based on the same *Form* class).

Example : The parent form creation (within Visual Basic 2005).

- One should open an application form.
- **Name**: *ParentForm.*
- **Text:** *Financial Functions.*
- **IsMdiContainer:** *True.* After this property has changed from *False* to *True*, the form's background aspect modifies itself in order to reflect its new role - the role of container for a *MDI* application (fig. 1):

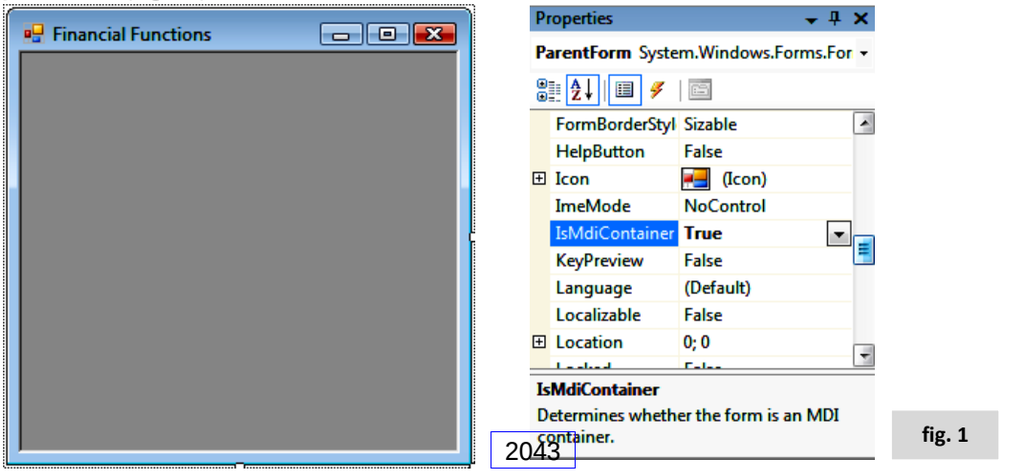


fig. 1

3. MENUS

Main menus.

Internal, the zero level menus (main menus) consist of the *MenuItems* collection of the *MainMenu* control. The menu items create another *MenuItems* collection, each of it being attached to the menu it belongs to. Regarding the visual aspect, the zero level menus are also *MenuItem* type objects, just the same as every element included within a menu or a submenu situated within a lower menu hierarchy.

One may further exemplify the way in which a MDI gets constructed for performing some financial functions. Those financial functions that one should consider are **predefined Basic functions** *(FV, PV, NPer, Pmt, Ipmt)*:

FV*(rate,nper,pmt,pv,type)*	The subsequent value of an investment or a loan.
PV*(rate,nper,pmt,fv,type)*	The current (up to date) value of an investment
NPer*(rate,pmt,pv,fv,type)*	The payment periods' number for an investment or a loan.
PMT*(rate,nper,pv,fv,type)*	The periodical rate for an investment or a loan.
IPMT*(rate,per,nper,pv,fv,type)*	The loan's interest for a certain period.

- For the already opened form within the design environment, one may select the *MenuStrip* control from the *Toolbox*. This will be added to the form, having the name of **MnuFunction** (fig. 2):

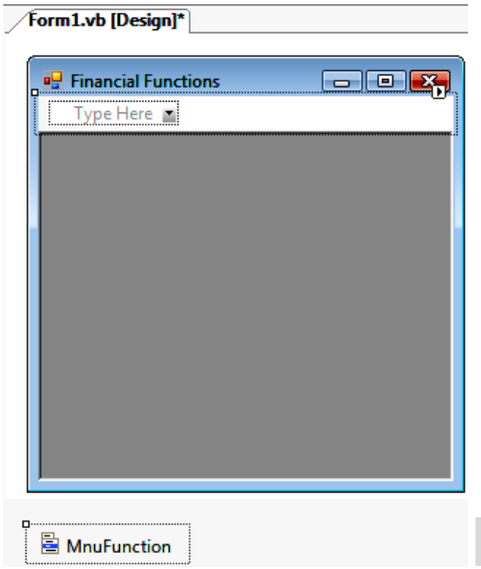


fig. 2

- One may type the menu's text value after selecting the *Type Here* box within the form, which stands also for the menu's title (here *Function*). After introducing the text related to the first menu, a new *Type Here* box will appear for adding either an element from submenu, or another zero level meniu (fig. 3):

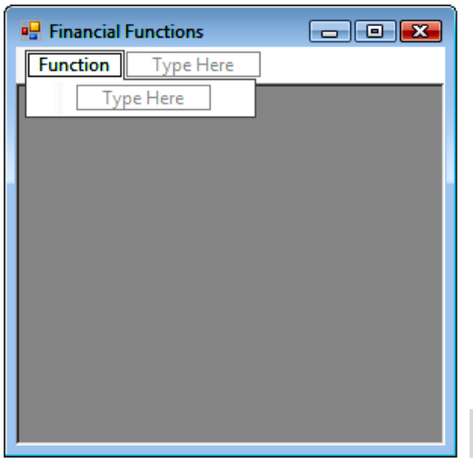


fig. 3

- One should introduce successively the *Windows* option within the zero level meniu (fig. 4).

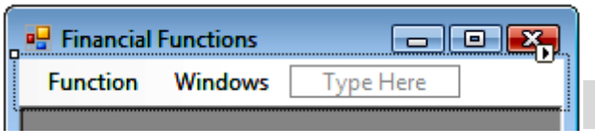
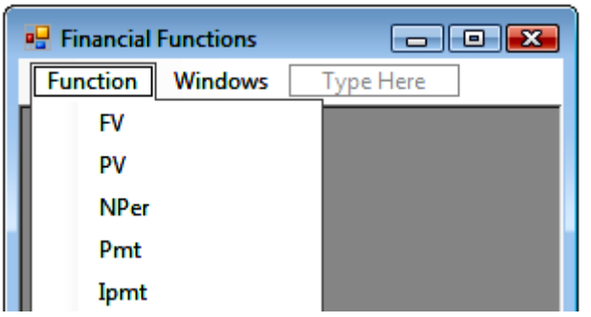
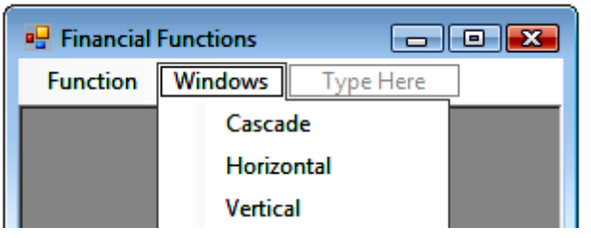


fig. 4

- The submenus have to be introduced (fig. 5):



FV Name : mnuFVFunction
PV Name : mnuPVFunction
Nper Name : mnuNperFunction
Pmt Name : mnuPmtFunction
Ipmt Name : mnuIpmtFunction



Cascade Name : mnuCascadeName
Horizontal Name : mnuHorizontalName
Vertical Name : mnuVerticalName

fig. 5

4. MDI CHILD FORM

Generally, every *Windows* form may serve as a base for a *MDI* child form. In order to convert a certain instance of a *form* class into a *MDI* child, the **MdiParent** property of the respective form specifies the *MDI* parent form.

Example: The creatioon of a child form (fig. 6).

- One may include a new application type form within the previous project (the *Add New Item* button).
- **Name** Property:*ChildForm.*
- One should add 6 *TextBox*es within the *ChildForm* – in order to introduce the arguments needed for the financial functions; a *Label* – for showing the results; and a button in order to launch the calculation:

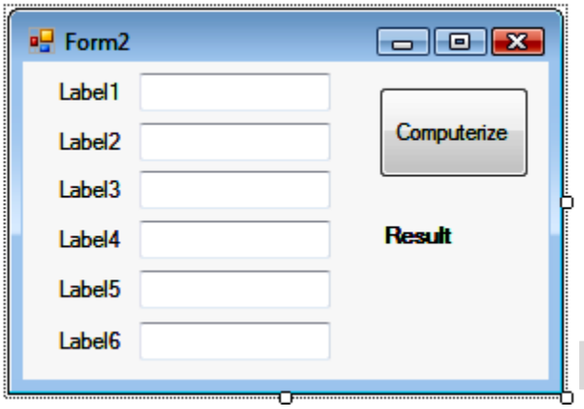


fig. 6

Controls			
Control	***Type***	***Propertier***	***Propertier Value***
Form2	Form	**Name**	*ChildForm*
Label1	Label	**Name**	*lbl1*
Label2	Label	**Name**	*lbl2*
Label3	Label	**Name**	*lbl3*
Label4	Label	**Name**	*lbl4*
Label5	Label	**Name**	*lbl5*
Label6	Label	**Name**	*lbl6*
		Visible	*False*
Label7	Label	**Name**	*lblResult*
		Text	*Result*
textBox1	TextBox	**Name**	*txt1*
textBox2	TextBox	**Name**	*txt2*
textBox3	TextBox	**Name**	*txt3*
textBox4	TextBox	**Name**	*txt4*
textBox5	TextBox	**Name**	*txt5*
textBox6	TextBox	**Name**	*txt6*

		Visible	*False*
btnComput	Buton	**Text**	*Computerize*

Pay attention! In **ParentForm**, one must obligatorily activate an instructions sequence through which one creates an instance of the new form and gives the **MdiParent** property (as a consequence, when running the project, the child form appears in the client zone of the parent form):

```
...
    Dim Child As New ChildForm
        Child.MdiParent = Me
        Child.Show()
...
```

The necessary code sequences (methods) must be associated with the two forms described previously. These code sequences will run when selecting the menu options.

Code sequences associated with the parent form:

Methods

Control	*Code*

Form1	`Public Class ParentForm`
	`  Private Sub TheChild(ByVal Title As String)`
	`  Dim Child As New ChildForm`
	`  Child.MdiParent = Me`
	`  Child.Show()`
	`  Child.Text = Title`
	`  Select Case Title`
	`  Case "FV"`
	`  Child.lbl1.Text = "Rate"`
	`  Child.lbl2.Text = "Nper"`
	`  Child.lbl3.Text = "Pmt"`
	`  Child.lbl4.Text = "Pv"`
	`  Child.lbl5.Text = "Type"`
	`  Case "PV"`
	`  Child.lbl1.Text = "Rate"`
	`  Child.lbl2.Text = "Nper"`
	`  Child.lbl3.Text = "Pmt"`
	`  Child.lbl4.Text = "Fv"`
	`  Child.lbl5.Text = "Type"`
	`  Case "NPer"`
	`  Child.lbl1.Text = "Rate"`
	`  Child.lbl2.Text = "Pmt"`
	`  Child.lbl3.Text = "Pv"`
	`  Child.lbl4.Text = "Pv"`
	`  Child.lbl5.Text = "Type"`
	`  Case "Pmt"`
	`  Child.lbl1.Text = "Rate"`
	`  Child.lbl2.Text = "Nper"`
	`  Child.lbl3.Text = "Pv"`
	`  Child.lbl4.Text = "Fv"`

```
                Child.lbl5.Text = "Type"
            Case "Ipmt"
                Child.lbl1.Text = "Rate"
                Child.lbl2.Text = "Per"
                Child.lbl3.Text = "Nper"
                Child.lbl4.Text = "Pv"
                Child.lbl5.Text = "Fv"
                Child.lbl6.Text = "Type"
                Child.lbl6.Visible = True
                Child.txt6.Visible = True
        End Select
        Child.Parent = Me
        Child.Show()
    End Sub
```

Function

```
Private Sub mnuFVFunction_Click(ByVal sender As …
        TheChild("FV")
End Sub

Private Sub mnuPVFunction_Click(ByVal sender As …
        TheChild("PV")
End Sub

Private Sub mnuNperFunction_Click(ByVal sender As …
        TheChild("Nper")
End Sub

Private Sub mnuPmtFunction_Click(ByVal sender As …
        TheChild("Pmt")
End Sub

Private Sub mnuIpmtFunction_Click(ByVal sender As …
        TheChild("Ipmt")
End Sub
```

Windows

```
Private Sub mnuCascadeName_Click(ByVal sender As …
        Me.LayoutMdi(MdiLayout.Cascade)
End Sub

Private Sub mnuHorizontalName_Click(ByVal sender As …
        Me.LayoutMdi(MdiLayout.TileHorizontal)
End Sub

Private Sub mnuVerticalName_Click(ByVal sender As …
        Me.LayoutMdi(MdiLayout.TileVertical)
End Sub
```

```
End Class
```

Code sequences associated with the child form:

Methods

Control	*Code*
Form2	`Public Class ChildForm` `  Public TheParent As ParentForm`

Computerize	`Private Sub btnComput_Click(ByVal sender As …`
	`  Dim A1, A2, A3, A4, A5, A6 As Double`
	`  A1 = Val(txt1.Text)`
	`  A2 = Val(txt2.Text)`
	`  A3 = Val(txt3.Text)`
	`  A4 = Val(txt4.Text)`
	`  A5 = Val(txt5.Text)`
	`  A6 = Val(txt6.Text)`
	`  Dim selection As String`
	`  selection = Me.Text`
	`  Select Case selection`
	`  Case "FV"`
	`  lblResult.Text = FV(A1, A2, A3, A4, A5)`
	`  Case "PV"`
	`  lblResult.Text = PV(A1, A2, A3, A4, A5)`
	`  Case "NPer"`
	`  lblResult.Text = NPer(A1, A2, A3, A4, A5)`
	`  Case "Pmt"`
	`  lblResult.Text = Pmt(A1, A2, A3, A4, A5)`
	`  Case "Ipmt"`
	`  lblResult.Text = _`
	`  IPmt(A1, A2, A3, A4, A5, A6)`
	`  End Select`
	`End Sub`

	`End Class`

The result of the application (fig. 7):

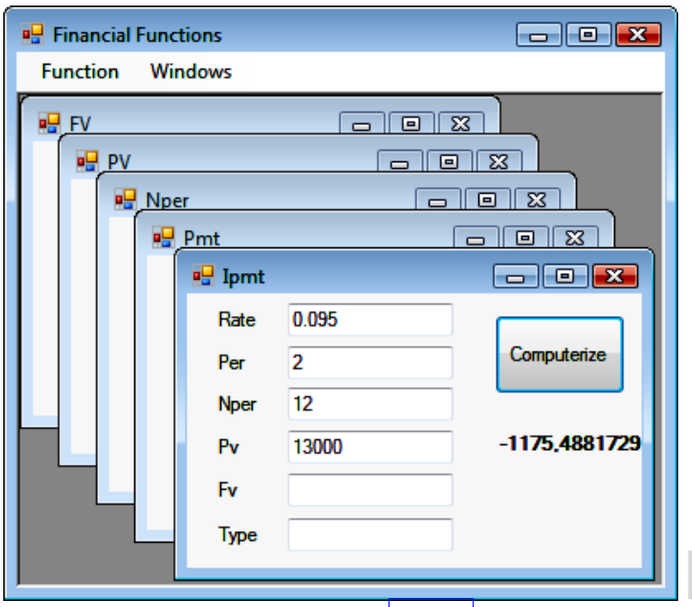


fig. 7

REFERENCES

[1] Emil Cosma (2007), *VISUAL BASIC …VBA 2007 … Studio 2005,* Ed. MATRIX ROM, Bucureşti
[2] Harold Davis (2004), *VISUAL BASIC pentru Windows*, Ed. Corint, Bucureşti
[3] McFedries (2006), *VBA – Ghid pentru începători,* Ed. TEORA, Bucureşti
[4] Parsons Andrew (®2005), *Wrox's Visual Basic, Express Edition,* www.wrox.com
[5] Walnum Clayton (2003) , *VISUAL BASIC.NET,* Ed. B.I.C. ALL, Bucureşti