

Streamlining Access Form Module Code in Standalone Class Module.

Ramachandran (A.P.R.) Pillai

Microsoft Access Developer Since 1996.

**Worked as Access & Excel Developer,
Internal Audit Department.
Saud Bahwan Automotive LLC,
Sultanate of Oman, from 1996 to 2010**

**Blog: Advanced Access Programming Tips and Tricks
from Sept. 2006**

LEARN MS-ACCESS TIPS AND TRICKS

<https://www.msaccesstips.com>

<https://www.msaccesstips.com/2023/04/reusing-form-module-vba-code-for-new.html>



Highlighting Text, Combo, ListBoxes on Form

2

Form1 Module

```
Option Compare Database  
Option Explicit
```

```
Private Clr As New myClass2_Init
```

```
Private Sub Form_Load()  
    Set Clr.o_Fm = Me  
End Sub
```

Form1

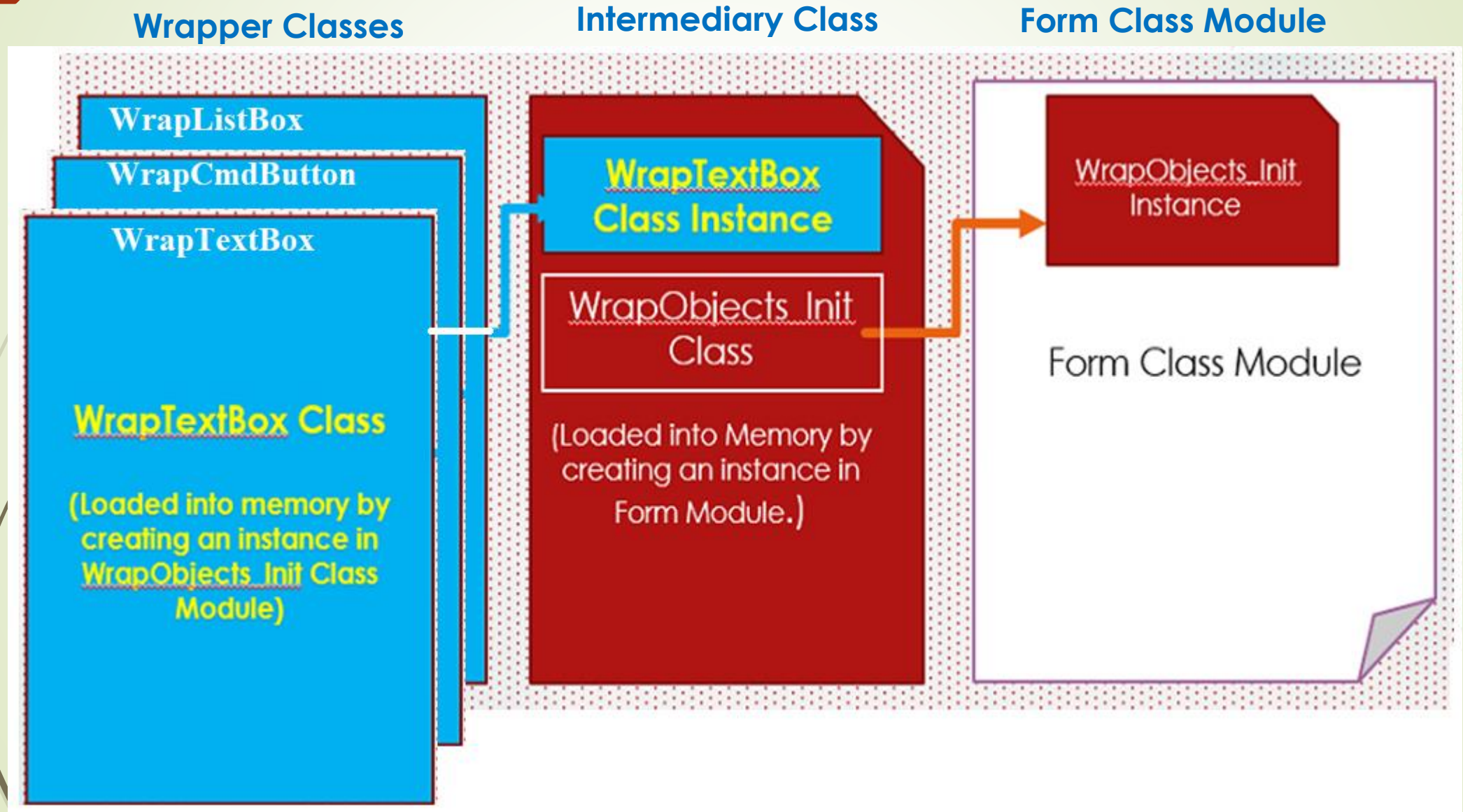
TextBox_highlight_Normal

Event Subroutines in Form Module.

SID	<input type="text"/>
Description:	<input type="text"/>
Quantity(1-10):	<input type="text"/>
Unit Price:	<input type="text"/>
Tax 18%:	<input type="text"/>
Total Price:	<input type="text"/>
Discount Amt.:	<input type="text"/>
Net Payable:	<input type="text"/>

All Class Modules and Instances are Loaded into Memory When the Form is open.

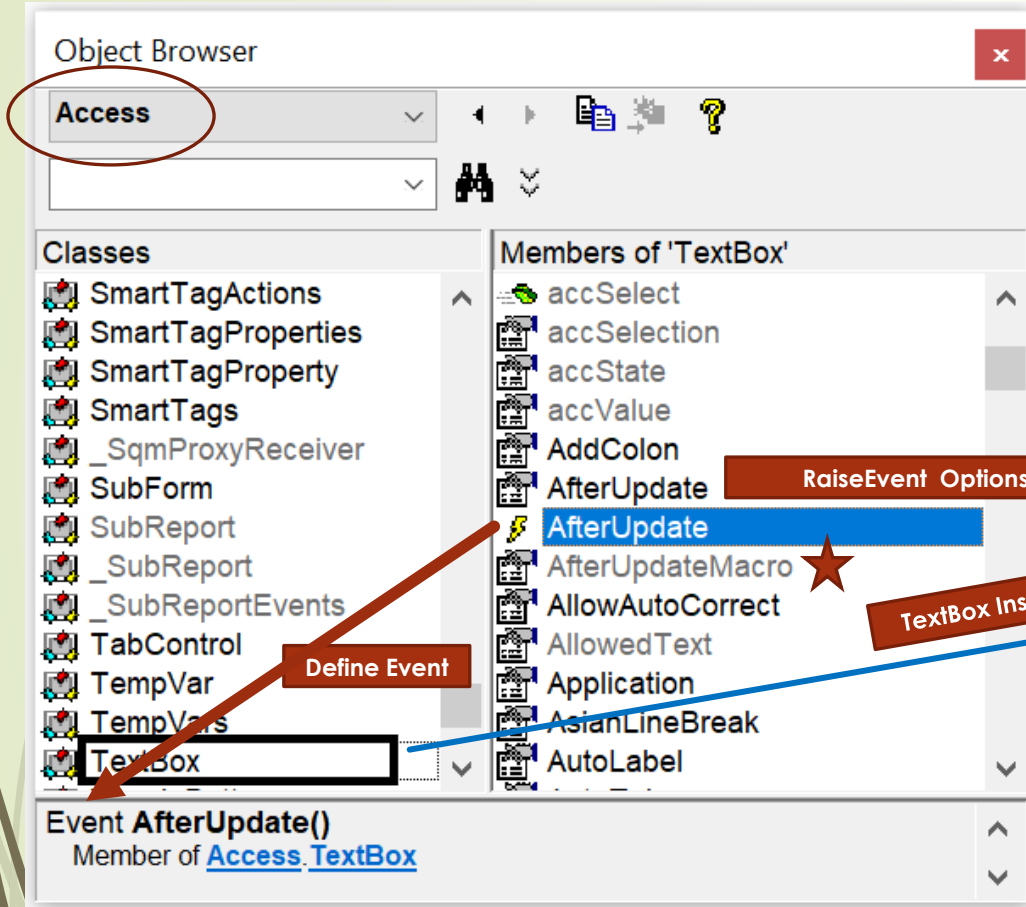
3



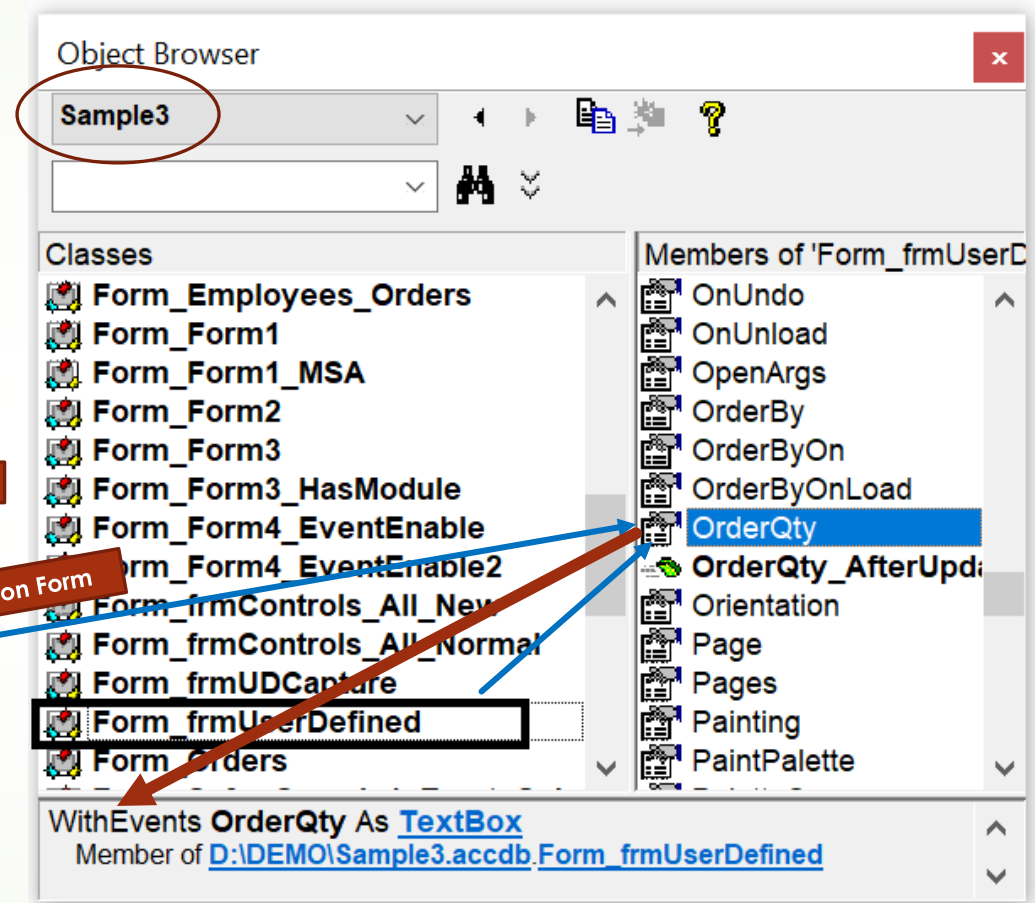
HOW ACCESS OBJECTS ARE BUILT? – OBJECT BROWSER VIEW

4*3

TextBox: Event, RaiseEvent



TextBox Instance Defined with: WithEvents



About: Access Class Objects & Events

Object Event Declarations: **Event**, **RaiseEvent**, and **WithEvents**

Event – The Exit Event of TextBox Class.

Example: **Event** Exit(Cancel As Integer)

RaiseEvent – Calls the Exit Event Subroutine.

Example: **RaiseEvent** Exit(False)

Property **On Exit As String** – Fires **RaiseEvent** when Property Value =  **[Event Procedure]**

Inbuilt **On Exit Property**
accepted Values.

Nothing to do with
WithEvents

Macro Name
=Function()

WithEvents – Module-level Class **Object Instance** declaration to capture the RaiseEvent when Fired.

Example: Private **WithEvents** Text0 As Access.TextBox

Event Sub-routine Code in the **Parent Class Module**, where the object Instance is created.

Example: **Sub** Text0_Exit(Cancel As Integer)
 'Code
 End Sub

} Class Module
required.

The Wrapper Class Template for Streamlining TextBox Events

6

```
Option Compare Database
```

```
Option Explicit
```

```
Private frm As Access.Form
```

```
Private WithEvents Txt As Access.TextBox
```

```
Public Property Get m_frm() As Access.Form
```

```
    Set m_frm = frm
```

```
End Property
```

```
Public Property Set m_frm(ByRef mNewFrm As Access.Form)
```

```
    Set frm = mNewForm
```

```
End Property
```

```
Public Property Get m_txt() As Access.TextBox
```

```
    Set m_txt = Txt
```

```
End Property
```

```
Public Property Set m_txt(ByRef mNewTxt As Access.TextBox)
```

```
    Set Txt = mNewTxt
```

```
End Property
```

```
Sub txt_BeforeUpdate()
```

```
    Global Area Code
```

```
    Select Case txt.Name
```

```
        Case "Text0"
```

```
            'Code
```

```
        Case "Text2"
```

```
            'Code
```

```
        Case "Text4","Text6","Text8"
```

```
            'Common Code
```

```
        Case "Text8"
```

```
            'Code
```

```
    End Select
```

```
End Sub
```



Event, RaiseEvent, WithEvents

7*3

Rule-1

1. Event
2. RaiseEvent
3. WithEvents

User Defined Event in
Form1 Class Module.

Opens First

Form1

Text0
Hello World
Textbox

```
Public Event Message(Byval Txt As String)
```

```
Private Sub Text0_Change()
```

```
    RaiseEvent Message(Me.Text0.Text)
```

```
End Sub
```

Rule-2

The Fired Event is captured by the **Form_Form1** Instance. The Event Subroutine must be written on the Parent Module of the Object Instance.

Form2

Label0: Hello World

```
Private WithEvents frm As Form_Form1
```

```
Private Sub Form_Load()
```

```
    Set frm = Forms("Form1")
```

```
End Sub
```

```
Private Sub frm_Message(ByVal txtmsg As String)
```

```
    Me.Label0.Caption = txtmsg
```

```
End Sub
```