

```

Rem =====
Rem V_Mouse Listener_OOoBasic Program (Copyright:Shuichi Sunaga)
Rem (2004/09/24)Added cbVeriLN(2006/04/25:56)(2009/01/28:101)
Rem Classified MouseEvent into Sheet/Dialog/Doc related actions(2011/07/05:151)
Rem Rearrangement on the discarding of AcctDialog(2012/01/01:136)
Rem Translation into English(2012/08/16:98)
Rem -----
Rem Sub RegisterMouseHdlr()
Rem Sub UnRegisterxMouseHdlr()
Rem Sub Ms_mousePressed(oEvent) As Boolean
Rem Sub Ms_mouseReleased (oEvent) As Boolean
Rem Sub Ms_disposing() 'indispensable
Rem Sub MouseListener_disposing() 'Indispensable.
Rem Sub MouseListener_mousePressed(oEvent)
Rem Sub MouseListener_mouseReleased(oEvent)
Rem Sub MouseListener_mouseEntered(oEvent)
Rem Sub MouseListener_mouseExited(oEvent)
Rem Sub RegisterDocListener()
Rem Sub UnRegisterDocListener()
Rem Sub DL_notifyEvent(oEvent)oListen As Object
Rem Sub DL_disposing(oEvent)
Rem =====
Option Explicit
Rem =====(Actions on Sheet related monseEvents)=====
Sub RegisterMouseHdlr()
    If IsNull(xMouseHdlr) Then
        xMouseHdlr = CreateUnoListener("Ms_", "com.sun.star.awt.XMouseClickedHandler")
        ThisComponent.CurrentController.addMouseClickedHandler(xMouseHdlr)
    End If
End Sub
Sub UnRegisterxMouseHdlr()
    ThisComponent.getCurrentController.removeMouseClickedHandler (xMouseHdlr)
    xMouseHdlr = oNULL
End Sub
Sub Ms_mousePressed(oEvent) As Boolean
    Dim oSelection As Object
    Dim addr, addrCln%, currCol%, z%, tMonth% 'addr.Sheet addr.Row
    If oEvent.ClickCount >= 2 Then
        Set aController = ThisComponent.CurrentController
        Set oSheets = ThisComponent.Sheets
        oSelection = aController().Selection()
        If oSelection.SupportsService("com.sun.star.sheet.SheetCell") Then
            Set aSheet = aController.ActiveSheet
            aSName$ = aSheet.getName()
            addr = oSelection.CellAddress
            addrCln% = addr.Column+1
            z% = iWS(aSName$)
            If z% < 5 then
                Col_ActCD% = Col_InArray(aSName$, "ActCD")
                If addrCln% = Col_ActCD% Then
                    ShowAcctName()
                Else
                    If IsAWDIgOpen() Then
                        oAWDIg.getControl("g_sVL").text = "Stop"
                    end if
                End If
            ElseIf (z%-10)*(z%-12)=0 then
                DirectAcctAnlys()
                Ms_mousePressed = True
                Exit Sub
            End If
        End If
    Endif
    Ms_mousePressed = False
End Sub
Rem In the above example, return value is true only when a cell is double clicked
Rem This is intended to suppress edit mode and allow view action.
Sub Ms_mouseReleased (oEvent) As Boolean
    Ms_mouseReleased = False
End Sub

```

```

1 Rem =====
2 Rem V_Mouse Listener_Excel VBA Program (Copyright:Shuichi Sunaga)
3 Rem EXCEL VBA does not require Mouse Listener to capture MouseEvents.
4 Rem The procedure name attached to Forms functions as MouseLisr
5 Rem Do not know how to trap worksheet related MouseEvents.
6 Rem Translation into English(2012/08/16:98)
7 Rem -----
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

```

```

Sub Ms_disposing() ' cannot omit
End Sub
Rem =====(Actions on Dialog related monseEvents)=====
Rem   Used in Matching AR/AP. No registration of MouseClickHandler
Rem =====
Sub MouseListener_mouseEntered(oEvent)
End Sub
Sub MouseListener_mouseExited(oEvent)
End Sub
Sub MouseListener_disposing() 'Cannot omit this proc.
End Sub
Sub MouseListener_mousePressed(oEvent)
End Sub
Sub MouseListener_mouseReleased(oEvent)
    Dim pSName$, sESMname$
    If oEvent.ClickCount > 1 then
        pSName$ = aController.ActiveSheet.getName()
        sESMname$ = oEvent.Source.Model.Name
        Select Case sESMName$
            Case "ListBox1"
                SelectByItem()
            Case "ListBox2"
                RemoveByItem()
            Case "CurYMD"
                aController.ActiveSheet =WkBK
        End Select
    End If
End Sub
<!-- End of Page -->
Rem =====(Actions on Document related monseEvents)=====
Sub RegisterDocListener()
    If IsNull(oDocListener) Then
        oDocListener = CreateUnoListener("DL_", "com.sun.star.document.XEventListener")
        ThisComponent.com_sun_star_document_XEventBroadcaster_addEventListener(oDocListener)
    End If
End Sub
Sub UnRegisterDocListener()
    If Not IsNull(oDocListener) Then
        ThisComponent.com_sun_star_document_XEventBroadcaster_removeEventListener(oDocListener)
    End If
    Dim oNull As Object
    oDocListener = oNull
End Sub
Sub DL_notifyEvent(oEvent)oListen As Object)
    Select Case oEvent.EventName
        Rem Case "OnUnload"
        Rem UnRegisterDocListener()
        Rem msgbox "OnUnload"
        Case "OnPrepareUnload"
            UnRegisterDocListener()
            MsgTxt$ = "File to be closed."
            msgbox(MsgTxt$)
        Case "OnPrint"
            PrintWarning()
        Case "OnSaveDone"
            MsgTxt$ = "File saved."
            msgbox(MsgTxt$)
    End Select
End Sub
Sub DL_disposing(oEvent)
    Rem Do Nothing
End Sub

```

71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

<!-- End of Page -->