

```

Rem =====
Rem V_Functions used in common OoBasic Program (Copyright:Shuichi Sunaga)
Rem (2005/10/31:404)(2009/01/28:507)(2011/08/31:390)(2012/01/30|384)
Rem Translation in English(2012/08/16|385)
Rem ===== Get Cell content·Write into Cell·Move cursor from cel to cell =====
Rem Function SelectionRow&()
Rem Function SelectionColumn%()
Rem Function SelectionAddress$()
Rem Function GetCellName$(ByVal iRow&, ByVal iCol%)
Rem Function CellValue(oWS as object, ByVal iCol%, ByVal iRow&)
Rem Function CellString(oWS as object, ByVal iCol%, ByVal iRow&)
Rem Function CellFormula(oWS as object, ByVal iRow&, ByVal iCol%)
Rem Sub SelectCell(ByVal eRow&, ByVal eCol%)
Rem Sub SelectCellFS(ByVal oWS As Object, ByVal eRow&, ByVal eCol%)
Rem Sub ClearCell(ByVal iRow&, ByVal iCol%)
Rem Sub SetCellValue(oWS as object, ByVal iRow&, ByVal iCol%, vValue$)
Rem Sub SetCellString(oWS as object, ByVal iRow&, ByVal iCol%, vValue$)
Rem Sub SetCellFormula(oWS as object, ByVal iRow&, ByVal iCol%, vValue$)
Rem ===== get CellRange contents·clear CellRange contents =====
Rem Sub ClearRange(ByVal oWS as object, iRow0%, iCol0%, iRow1%, iCol1%)
Rem Function RangeSum&(ByVal oWS As Object, StrCol%, StrRow&, EndCol%, EndRow&)
Rem ===== get Sheet/Doc attributes =====
Rem Function GetSheetNdx$(ByVal pSName$)
Rem Function IsWSFound(ByVal pSName$) as Boolean
Rem ===== get Row/Column number =====
Rem Function getFSRow%(oWS As Object, FindWord$, fCol%)
Rem Function getBKRow%(fLine$, fBKCd$)
Rem Function CtxInsRow&(iStrtRow%, FindNo%)
Rem Function Col_InFR%(ByVal pSName$, sWD$)
Rem Function GetLastUsedRow(oWS as Object) as Integer
Rem Function GetLastUsedCol(ByVal oWS as Object) as Integer
Rem ==== Other convenient functions without relation to cursor position ====
Rem Function CAlpha(ByVal iCol%) as string
Rem Function myIsEmpty(ByVal iCol%, ByVal iRow&) as Boolean
Rem Function myVal(ByVal v$) as Long
Rem Function myDVal(ByVal v$) as Double
Rem Function myMax(ByVal a%, ByVal b%)
Rem Function myMin(ByVal a%, ByVal b%)
Rem Function myIIF(ByVal a as boolean, ByVal b, c)
Rem Function mySigma(ByVal v) as integer
Rem Function IsAWDIgOpen as Boolean
Rem Function IndexInACT%(ByVal SearchString$)
Rem Function IndexInArray%(ByVal SearchString$, LocList())
Rem Function Acct2Txcd$(ByVal SearchString$)
Rem F:get1stLetter$(ByVal fName$)
Rem Sub WriteDbg(obj, what$)
Rem =====
Option Explicit
Rem ===== Get Cell content·Write into Cell·Move cursor from cel to cell =====
Function SelectionRow&()
    SelectionRow& = aController.Selection.RangeAddress.StartRow + 1
End Function
Function SelectionColumn%()
    SelectionColumn% = aController.Selection.RangeAddress.StartColumn + 1
End Function
Function SelectionAddress$()
    SelectionAddress$ = GetCellName$(SelectionRow&, SelectionColumn%)
End Function
Function GetCellName$(ByVal iRow&, ByVal iCol%)
    GetCellName$ = CAlpha(iCol%) & "$" & CStr(iRow&)
End Function
Function CellValue(ByVal oWS as object, ByVal iRow&, ByVal iCol%)
    If iCol%>=1 and iRow&>=1 then
        CellValue = oWS.GetCellbyPosition(iCol% - 1, iRow& - 1).Value
    End if
End Function
Function CellString(ByVal oWS as object, ByVal iRow&, ByVal iCol%)

```

```

1 Rem =====
2 Rem V_Functions used in common Excel VBA Program (Copyright:Shuichi Sunaga)
3 Rem (2011/08/13:350)(2012/01/31:328)
4 Rem Translation in English(2012/08/16|329)
5 Rem ===== Get Cell content·Write into Cell·Move cursor from cel to cell =====
6 Rem F:SelectionRow&()
7 Rem F:SelectionColumn%()
8 Rem F:SelectionAddress$()
9 Rem Function GetCellName$(ByVal iRow&, ByVal iCol%)
10 Rem Function CellValue(ByVal oWS as Worksheet, ByVal iRow&, ByVal iCol%)
11 Rem Function CellString(ByVal oWS as Worksheet, ByVal iRow&, ByVal iCol%)
12 Rem Function CellFormula(ByVal oWS as Worksheet, ByVal iRow&, ByVal iCol%)
13 Rem Sub SelectCell(ByVal eRow&, ByVal eCol%)
14 Rem Sub SelectCellFS(ByVal oWS As Worksheet, ByVal eRow&, ByVal eCol%)
15 Rem Sub ClearCell(ByVal iRow&, ByVal iCol%)
16 Rem Sub SetCellValue(oWS as Worksheet, ByVal iRow&, ByVal iCol%, vValue&)
17 Rem Sub SetCellString(oWS as Worksheet, ByVal iRow&, ByVal iCol%, vValue)
18 Rem Sub SetCellFormula(oWS as Worksheet, ByVal iRow&, ByVal iCol%, vValue$)
19 Rem ===== get CellRange contents·clear CellRange contents =====
20 Rem Sub ClearRange(ByVal oWS as Worksheet, iRow0&, iCol0%, iRow1&, iCol1%)
21 Rem Function RangeSum&(ByVal oWS As Worksheet, StrCol%, StrRow&, EndCol%, EndRow&)
22 Rem ===== get Sheet/Doc attributes =====
23 Rem Function GetSheetNdx%(ByVal pSName$)
24 Rem Function IsWSFound(ByVal pSName$) as Boolean
25 Rem ===== get Row/Column number =====
26 Rem Function getFSRow%(oWS As Object, FindWord$, fCol%)
27 Rem Function getBKRow%(fLine$, fBKCd$)
28 Rem Function CtxInsRow&(iStrtRow%, FindNo%)
29 Rem Function Col_InFR%(ByVal pSName$, sWD$)
30 Rem Function GetLastUsedRow(oWS as Object) as Integer
31 Rem Function GetLastUsedCol(ByVal oWS as Object) as Integer
32 Rem ==== Other convenient functions without relation to cursor position ====
33 Rem Function CAlpha(ByVal iCol%) as string
34 Rem Function myIsEmpty(ByVal iCol%, ByVal iRow&) as Boolean
35 Rem Function myVal(ByVal v$) as Long
36 Rem Function myDVal(ByVal v$) as Double
37 Rem Function myMax(ByVal a%, ByVal b%) as integer
38 Rem Function myMin(ByVal a%, ByVal b%) as integer
39 Rem Function myIIF(ByVal a as boolean, ByVal b, c)
40 Rem Function mySigma(ByVal v) as integer
41 Rem Function IsAWDIgOpen as Boolean
42 Rem Function IndexInACT%(ByVal SearchString$)
43 Rem Function IndexInArray%(ByVal SearchString$, LocList())
44 Rem Function Acct2Txcd$(ByVal SearchString$)
45 Rem F:get1stLetter$(ByVal fName$)
46
47 Rem =====
48 Option Explicit
49 Rem ===== Get Cell content·Write into Cell·Move cursor from cel to cell =====
50 Function SelectionRow&()
51     SelectionRow& = Selection.Row&
52 End Function
53 Function SelectionColumn%()
54     SelectionColumn% = Selection.Column%
55 End Function
56 Function SelectionAddress$()
57     SelectionAddress$ = GetCellName$(SelectionRow&, SelectionColumn%)
58 End Function
59 Function GetCellName$(ByVal iRow&, ByVal iCol%)
60     GetCellName$ = CAlpha(iCol%) & "$" & CStr(iRow&)
61 End Function
62 Function CellValue(ByVal oWS As Worksheet, ByVal iRow&, ByVal iCol%)
63     If iCol% >= 1 And iRow& >= 1 Then
64         CellValue = oWS.Cells(iRow&, iCol%).Value
65         If TypeName(CellValue) = "String" Then
66             CellValue = 0
67         End If
68     End If
69 End Function
70 Function CellString(ByVal oWS As Worksheet, ByVal iRow&, ByVal iCol%)

```

```

        If iCol%>=1 and iRow%>=1 then
            CellString = trim(oWS.GetCellbyPosition(iCol% - 1, iRow% - 1).String)
        End If
    End Function
Function CellFormula(ByVal oWS as object, ByVal iRow%, ByVal iCol%)
    If iCol%>=1 and iRow%>=1 then
        CellFormula = oWS.GetCellbyPosition(iCol% - 1, iRow% - 1).Formula
    End If
End Function
Sub SelectCell(ByVal eRow%, ByVal eCol%)
    dim args(0) as new com.sun.star.beans.PropertyValue
    args(0).Name = "ToPoint"
    args(0).Value = GetCellName$(eRow%, eCol%)
    dispatcher.executeDispatch(aController.Frame, ".uno:GoToCell", "", 0, args())
End Sub
Sub SelectCellFS(ByVal oWS As Object, ByVal eRow%, ByVal eCol%)
    dim oCell As Object
    oCell=oWS.getCellByPosition(eCol%, eRow%)
    aController.Select(oCell)
End Sub
Sub ClearCell(ByVal iRow%, ByVal iCol%)
    If iCol%>=1 and iRow%>=1 then
        aSheet.getCellByPosition( iCol%-1, iRow%-1).ClearContents(23)
    EndIf
End Sub
Sub SetCellValue(ByVal oWS as object, ByVal iRow%, ByVal iCol%, vValue$)
    If iCol%>=1 and iRow%>=1 then
        oWS.GetCellbyPosition(iCol% - 1, iRow% - 1).Value = vValue$
    End If
End Sub
Sub SetCellString(ByVal oWS as object, ByVal iRow%, ByVal iCol%, vValue$)
    If iCol%>=1 and iRow%>=1 then
        oWS.GetCellbyPosition(iCol% - 1, iRow% - 1).String = vValue$
    End If
End Sub
Sub SetCellFormula(ByVal oWS as object, ByVal iRow%, ByVal iCol%, vValue$)
    If iCol%>=1 and iRow%>=1 then
        oWS.GetCellbyPosition(iCol% - 1, iRow% - 1).Formula = vValue$
    End If
End Sub
Rem ===== get CellRange contents•clear CellRange contents =====
Sub ClearRange(ByVal oWS as object, iRow0%, iCol0%, iRow1%, iCol1%)
    dim fRNM$
    If iCol0%>=1 and iRow0%>=1 then
        if iCol1%>=iCol0% and iRow1%>= iRow0% then
            fRNM$ = CAlpha(iCol0%) & Cstr(iRow0%) & "•"
            fRNM$ = fRNM$ & CAlpha(iCol1%) & Cstr(iRow1%)
            oWS.getCellRangeByName(fRNM$).ClearContents(23)
        end if
    EndIf
End Sub
Function RangeSum&(ByVal oWS As Object, StrCol%, StrRow%, EndCol%, EndRow%)
    dim aArgs(0) As Variant, fRNM$
    If Not (StrCol%<=EndCol% and StrRow% <= EndRow%) then
        RangeSum& = 0
    Else
        fRNM$ = CAlpha(StrCol%) & Cstr(StrRow%) & "•"
        fRNM$ = fRNM$ & CAlpha(EndCol%) & Cstr(EndRow%)
        aArgs(0) = oWS.getCellRangeByName(fRNM$)
        RangeSum& = oFunctionAccess.callFunction("SUM", aArgs() )
    End If
End Function
Rem ===== get Sheet/Doc attributes =====
Function GetSheetNdx%(ByVal pSName$)
    dim oSheet as object, oEnum as object
    GetSheetNdx%=0
    oEnum = oSheets.createEnumeration
    Do while oEnum.hasMoreElements()
        Set oSheet = oEnum.nextElement()
        if oSheet.getName() = pSName$ then

```

```

71     If iCol% >= 1 And iRow% >= 1 Then
72         CellString = CStr(Trim(oWS.Cells(iRow%, iCol%).Value))
73     End If
74 End Function
75 Function CellFormula(ByVal oWS As Worksheet, ByVal iRow%, ByVal iCol%)
76     If iCol% >= 1 And iRow% >= 1 Then
77         CellFormula = oWS.Cells(iRow%, iCol%).FormulaR1C1
78     End If
79 End Function
80 Sub SelectCell(ByVal eRow%, ByVal eCol%)
81     Cells(eRow%, eCol%).Select
82
83
84
85 End Sub
86 Sub SelectCellFS(ByVal oWS As Worksheet, ByVal eRow%, ByVal eCol%)
87     oWS.Cells(eRow%, eCol%).Select
88
89
90 End Sub
91 Sub ClearCell(ByVal iRow%, ByVal iCol%)
92     If iCol% >= 1 And iRow% >= 1 Then
93         Cells(iRow%, iCol%).ClearContents
94     End If
95 End Sub
96 Sub SetCellValue(ByVal oWS As Worksheet, ByVal iRow%, ByVal iCol%, vValue&)
97     If iCol% >= 1 And iRow% >= 1 Then
98         oWS.Cells(iRow%, iCol%).Value = vValue&
99     End If
100 End Sub
101 Sub SetCellString(ByVal oWS As Worksheet, ByVal iRow%, ByVal iCol%, vValue)
102     If iCol% >= 1 And iRow% >= 1 Then
103         oWS.Cells(iRow%, iCol%).Value = vValue
104     End If
105 End Sub
106 Sub SetCellFormula(ByVal oWS As Worksheet, ByVal iRow%, ByVal iCol%, vValue$)
107     If iCol% >= 1 And iRow% >= 1 Then
108         oWS.Cells(iRow%, iCol%).FormulaR1C1 = vValue$
109     End If
110 End Sub
111 Rem ===== get Sheet/Doc attributes =====
112 Sub ClearRange(ByVal oWS As Worksheet, iRow0%, iCol0%, iRow1%, iCol1%)
113     Dim fRNM$
114     fRNM$ = CAlpha$(iCol0%) & iRow0% & "•"
115     fRNM$ = fRNM$ & CAlpha$(iCol1%) & iRow1%
116     If iCol0% >= 1 And iRow0% >= 1 Then
117         If iCol1% >= iCol0% And iRow1% >= iRow0% Then
118             oWS.Range(fRNM$).ClearContents
119         End If
120     End If
121 End Sub
122 Function RangeSum&(ByVal oWS As Worksheet, StrCol%, StrRow%, EndCol%, EndRow%)
123     Dim oRange As Range, fRNM$
124     If Not (StrCol% <= EndCol% And StrRow% <= EndRow%) Then
125         RangeSum& = 0
126     Else
127         fRNM$ = CAlpha(StrCol%) & CStr(StrRow%) & "•"
128         fRNM$ = fRNM$ & CAlpha(EndCol%) & CStr(EndRow%)
129         Set oRange = oWS.Range(fRNM$)
130         RangeSum& = Application.WorksheetFunction.Sum(oRange)
131     End If
132 End Function
133 Rem ===== get Sheet/Doc attributes =====
134 Function GetSheetNdx%(ByVal pSName$)
135     Dim oSheet As Object
136     GetSheetNdx% = 1
137
138     For Each oSheet In Sheets
139
140         If oSheet.Name = pSName$ Then

```

```

        exit do
    endif
    GetSheetNdx%=GetSheetNdx%+1
Loop
End Function
Function IsWSFound(ByVal pSName$) as Boolean
    If GetSheetNdx(pSName$) <= oSheets.getCount()-1 then
        IsWSFound = True
    Else
        IsWSFound = False
    End If
End Function
Rem ===== get Row/Column number =====
Function getFSRow%(oWS As Object, FindWord$, fCol%)
    dim fRNM$
    dim oFound As object, oBPD Desc As Object, fRange As Object
    Set oBPD Desc = oWS.createSearchDescriptor()
    oBPD Desc.SearchWords = true
    oBPD Desc.SearchString = FindWord$
    oBPD Desc.SearchBackwards = False
    fRNM$=CAlpha$(fCol%)&CStr(CNRow%) & "."
    fRNM$=fRNM$ & CAlpha$(fCol%) & Cstr(1200)
    Set fRange = oWS.getCellRangeByName(fRNM$)
    Set oFound = fRange.findFirst(oBPD Desc)
    If IsNull(oFound) then
        getFSRow% = 1200
    Else
        getFSRow% = oFound.CellAddress.Row + 1
    Endif
End Function
Function getBKRow&(fLine$, fBKCD%)
    dim oFound As object, oBPD Desc As Object, fRange As Object
    dim fRNM$, i%, iCount%
    Set oBPD Desc = WkBK.createSearchDescriptor()
    oBPD Desc.SearchWords = true
    oBPD Desc.SearchString = fLine$
    oBPD Desc.SearchBackwards = False
    fRNM$ = CAlpha$(Col_LineNo%) & CStr(dbMonthTop%) & "-"
    fRNM$ = fRNM$ & CAlpha$(Col_LineNo%) & CStr(dbMonthBtm%)
    Set fRange = WkBK.getCellRangeByName(fRNM$)
    Set oFound = fRange.findAll(oBPD Desc)
    If IsNull(oFound) then
        getBKRow& = dbMonthBtm&
        Exit Function
    End If
    iCount%=oFound.getCount()
    For i% = 0 to iCount%-1
        getBKRow& = oFound(i%).CellAddress.Row+1
        If CellValue(WkBK, getBKRow&, Col_BKCD%)=fBKCD% then
            Exit For
        End If
    Next i%
End Function
Function Col_InFR%(ByVal pSName$, sWD$)
    dim WkWk as object, oFound As object, fRNM$
    Set WkWk = oSheets.getByName(pSName$)
    fRNM$ = cna$ & CStr(CNRow%) & cjc$ & CAlpha(CoLMT%) & CStr(CNRow%)
    Set oCNM Desc = WkWk.createSearchDescriptor()
    Set oCNM Range = WkWk.getCellRangeByName(fRNM$)
    oCNM Desc.SearchWords = True
    oCNM Desc.SearchString = sWD$
    oCNM Desc.SearchBackwards = False
    Set oFound = oCNM Range.findFirst(oCNM Desc)
    If IsNull(oFound) then
        Col_InFR% = 0
    Else
        Col_InFR% = oFound.CellAddress.Column + 1
    Endif
End Function

```

```

141         Exit For
142     End If
143     GetSheetNdx% = GetSheetNdx% + 1
144     Next oSheet
145 End Function
146 Function IsWSFound(ByVal pSName$) As Boolean
147     If GetSheetNdx(pSName$) <= Sheets.Count Then
148         IsWSFound = True
149     Else
150         IsWSFound = False
151     End If
152 End Function
153 Rem ===== get Row/Column number =====
154 Function getFSRow%(oWS As Worksheet, FindWord$, fCol%)
155     Dim fRNM$
156     Dim oFound As Object, fRange As Range
157
158
159
160
161     fRNM$ = CAlpha$(fCol%) & CStr(CNRow%) & "."
162     fRNM$ = fRNM$ & CAlpha$(fCol%) & CStr(1200)
163     Set fRange = oWS.Range(fRNM$)
164     Set oFound = fRange.Find(FindWord$, Lookat:=xlWhole)
165     If oFound Is Nothing Then
166         getFSRow% = 1200
167     Else
168         getFSRow% = oFound.Row
169     End If
170 End Function
171 Function getBKRow&(fLine$, fBKCD%)
172     Dim oFound As Object, fRange As Range
173     Dim fRNM$, foundAddress
174
175
176
177
178     fRNM$ = CAlpha$(Col_LineNo%) & CStr(dbMonthTop%) & "-"
179     fRNM$ = fRNM$ & CAlpha$(Col_LineNo%) & CStr(dbMonthBtm%)
180     Set fRange = WkBK.Range(fRNM$)
181     Set oFound = fRange.Find(fLine$, Lookat:=xlWhole)
182     If oFound Is Nothing Then
183         getBKRow& = dbMonthBtm&
184         Exit Function
185     End If
186     Do
187         foundAddress = oFound.Address
188         If CellValue(WkBK, oFound.Row, Col_BKCD%) = fBKCD% Then
189             getBKRow& = oFound.Row
190             Exit Do
191         Else
192             Set oFound = fRange.FindNext(oFound)
193         End If
194     Loop While Not oFound Is Nothing And oFound.Address <> foundAddress
195 End Function
196 Function Col_InFR%(ByVal pSName$, sWD$)
197     Dim WkWk As Worksheet, oFound As Object, fRNM$
198     Set WkWk = Workbooks(GLName).Worksheets(pSName$)
199     fRNM$ = "$A$" & CStr(CNRow%) & "." & CAlpha(CoLMT%) & CStr(CNRow%)
200
201     Set oCNM Range = WkWk.Range(fRNM$)
202
203
204
205     Set oFound = oCNM Range.Find(sWD$, Lookat:=xlWhole)
206     If oFound Is Nothing Then
207         Col_InFR% = 0
208     Else
209         Col_InFR% = oFound.Column
210     End If
End Function

```

```

End Function
Function GetLastUsedRow&(ByVal oWS as Object, ByVal iCol%)
    dim oCell As Object, oCursor As Object
    dim aAddress As Variant
    Set oCell = oWS.GetCellbyPosition(0, 0)
    Set oCursor = oWS.createCursorByRange(oCell)
    oCursor.gotoStartOfUsedArea( False )
    oCursor.gotoEndOfUsedArea( True )
    aAddress = oCursor.RangeAddress
    GetLastUsedRow& = aAddress.EndRow + 5
    Do while myIsEmpty(oWS, GetLastUsedRow&, iCol%)
        GetLastUsedRow& = GetLastUsedRow&-1
    Loop
End Function
Function GetLastUsedCol%(ByVal oWS as Object)
    dim oCell As Object, oCursor As Object
    dim aAddress As Variant
    Set oCell = oWS.GetCellbyPosition(0, 0)
    Set oCursor = oWS.createCursorByRange(oCell)
    oCursor.gotoStartOfUsedArea( False )
    oCursor.gotoEndOfUsedArea( True )
    aAddress = oCursor.RangeAddress
    GetLastUsedCol% = aAddress.EndColumn
End Function
Rem ==== Other convenient functions without relation to cursor position ====
Function CAlpha$(ByVal iCol%)
    dim s_preidx%, s_posidx%, aABC$( )
    aABC$( ) = Array("A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z")
    iCol% = iCol% - 1
    If iCol% > -1 then
        s_preidx% = int(iCol% / 26)
        s_posidx% = iCol% MOD 26
        if s_preidx% = 0 then
            CAlpha$ = "$" & aABC$(s_posidx%)
        else
            CAlpha$ = "$" & aABC$(s_preidx% - 1) & aABC$(s_posidx%)
        end if
    End If
End Function
Function myIsEmpty(ByVal pSheet as object, ByVal iRow&, ByVal iCol%) as Boolean
    If iCol%>=1 and iRow&>=1 then
        if Len(pSheet.getCellByPosition(iCol%-1, iRow& -1).String) = 0 then
            myIsEmpty = True
        else
            myIsEmpty = False
        end if
    End If
End Function
Function myVal&(ByVal v$)
    dim newv$
    If Left(v$, 1) = "(" then
        newv$ = Mid(v$, 2, Len(v$) - 2)
        myVal& = - Val(newv$)
    Else
        myVal& = Val(v$)
    End If
End Function
Function myDVal(ByVal v$) as Double
    If Left(v, 1) = "(" then
        newv = Mid(v, 2, Len(v) - 2)
        myDVal = - Val(newv)
    Else
        myDVal = Val(v)
    End If
End Function
Rem Functions free from Data Types (without restrictions by Data Type)
Function myMax(ByVal a, ByVal b)
    If a > b then
        myMax = a
    Else

```

```

211 End Function
212 Function GetLastUsedRow&(ByVal oWS As Worksheet, ByVal iCol%)
213     Dim oCursor As Range, i%
214     oWS.Unprotect ("pwd")
215     Set oCursor = oWS.Cells(1, iCol%).CurrentRegion
216     GetLastUsedRow& = oCursor.Rows.Count
217
218     For i% = 1 To 11
219         If UCase(oWS.Cells(GetLastUsedRow& + i%, 1).Value) = "ENDLN" Then
220             GetLastUsedRow& = GetLastUsedRow& + i% + 2
221             Exit For
222         End If
223     Next i%
224 End Function
225 Function GetLastUsedCol%(ByVal oWS As Worksheet)
226     Dim oCursor As Range
227     oWS.Unprotect ("pwd")
228
229     Set oCursor = oWS.Cells(1, 1).CurrentRegion
230
231
232
233     GetLastUsedCol% = oCursor.Columns.Count
234 End Function
235 Rem ==== Other convenient functions without relation to cursor position ====
236 Function CAlpha$(ByVal iCol%)
237     Dim s_preidx%, s_posidx%, aABC
238     aABC = Array("A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z")
239     iCol% = iCol% - 1
240     If iCol% > -1 Then
241         s_preidx% = Int(iCol% / 26)
242         s_posidx% = iCol% Mod 26
243         If s_preidx% = 0 Then
244             CAlpha$ = "$" & aABC(s_posidx%)
245         Else
246             CAlpha$ = "$" & aABC(s_preidx% - 1) & aABC(s_posidx%)
247         End If
248     End If
249 End Function
250 Function myIsEmpty(ByVal pSheet As Worksheet, ByVal iRow&, ByVal iCol%) As Boolean
251     If iCol% >= 1 And iRow& >= 1 Then
252         If Len(pSheet.Cells(iRow&, iCol%).Value) = 0 Then
253             myIsEmpty = True
254         Else
255             myIsEmpty = False
256         End If
257     End If
258 End Function
259 Function myVal&(ByVal v$)
260     Dim newv$
261     If Left(v$, 1) = "(" Then
262         newv$ = Mid(v$, 2, Len(v$) - 2)
263         myVal& = -Val(newv$)
264     Else
265         myVal& = Val(v$)
266     End If
267 End Function
268 Function myDVal(ByVal v$) As Double
269     If Left(v, 1) = "(" Then
270         newv = Mid(v, 2, Len(v) - 2)
271         myDVal = -Val(newv)
272     Else
273         myDVal = Val(v)
274     End If
275 End Function
276 Rem Functions free from Data Types (without restrictions by Data Type)
277 Function myMax(ByVal a, ByVal b)
278     If a > b Then
279         myMax = a
280     Else

```

```

        myMax = b
    End If
End Function
Rem Freed from data type(2006/12/21)
Function myMin(ByVal a, ByVal b)
    If a < b then
        myMin = a
    Else
        myMin = b
    End If
End Function
Function myIIF(ByVal a as boolean, ByVal b, c)
    If a then
        myIIF = b
    Else
        myIIF = c
    End If
End Function
Function mySigma(ByVal v) as integer
    If v < 0 then
        mySigma = -1
    Else
        mySigma = 1
    End If
End Function
Function IsAWDIgOpen() as Boolean
    If IsNull(oAWDIg) then
        IsAWDIgOpen = False
    Else
        IsAWDIgOpen = True
    End If
End Function
Function IndexInACT%(ByVal SearchString$)
    Dim i%, aList()
    aList()=oAWAct.Model.StringItemList
    IndexInACT% = -1
    For i% = Lbound(aList()) to Ubound(aList())
        If Ucase(Left(aList(i%), ADigit)) = Ucase(SearchString$) Then
            IndexInACT% = i%
            Exit For
        End If
    Next i%
End Function
Function IndexInArray%(ByVal SearchString$, LocList())
    Dim i%
    IndexInArray% = -1
    For i% = Lbound(LocList()) to Ubound(LocList())
        If Ucase(LocList(i%)) = Ucase(SearchString$) Then
            IndexInArray% = i%
            Exit For
        End If
    Next i%
End Function
Function Acct2Txcd$(ByVal SearchString$)
    Dim i%, mLen%, mLen2%, mAcct$, aList()
    aList()=oAWAct.Model.StringItemList
    i% = IndexInACT%(SearchString$)
    If i% = -1 Then
        Acct2Txcd$ = "N"
    Else
        mLen% = Len(aList(i%))
        mAcct$ = Trim(Right(aList(i%),mLen%-ADigit))
        mLen2% = Len(mAcct$)
        Acct2Txcd$ = Left(Right(mAcct$,mLen2%-1), 1)
    Endif
End Function
Function get1stLetter$(ByVal fName$)
    dim iPos%, gName$
    iPos% = InStr(fName$, "—")+InStr(fName$, "—")+InStr(fName$, "-")
    gName$=Trim(Right(fName$, Len(fName$)-iPos%))

```

```

281         myMax = b
282     End If
283 End Function
284 Rem Freed from data type(2006/12/21)
285 Function myMin(ByVal a, ByVal b)
286     If a < b Then
287         myMin = a
288     Else
289         myMin = b
290     End If
291 End Function
292 Function myIIF(ByVal a As Boolean, ByVal b, c)
293     If a Then
294         myIIF = b
295     Else
296         myIIF = c
297     End If
298 End Function
299 Function mySigma(ByVal v) As Integer
300     If v < 0 Then
301         mySigma = -1
302     Else
303         mySigma = 1
304     End If
305 End Function
306 Function IsAWDIgOpen() As Boolean
307     On Error GoTo ErrorHandler
308     IsAWDIgOpen = False
309     IsAWDIgOpen = oAWDIg.OpenFlag.Value
310 ErrorHandler:
311     Exit Function
312 End Function
313 Function IndexInACT%(ByVal SearchString$)
314     Dim i%, Lcnt%
315     Lcnt% = oAWAct.ListCount
316     IndexInACT% = -1
317     For i% = 0 To Lcnt% - 1
318         If UCase(Left(oAWAct.List(i%), ADigit)) = UCase(SearchString$) Then
319             IndexInACT% = i%
320             Exit For
321         End If
322     Next i%
323 End Function
324 Function IndexInArray%(ByVal SearchString$, LocList As Object)
325     Dim i%
326     IndexInArray% = -1
327     For i% = LBound(LocList.List()) To UBound(LocList.List())
328         If UCase(LocList.List(i%)) = UCase(SearchString$) Then
329             IndexInArray% = i%
330             Exit For
331         End If
332     Next i%
333 End Function
334 Function Acct2Txcd$(ByVal SearchString$)
335     Dim i%, mLen%, mLen2%, mAcct$, aList()
336     i% = IndexInACT%(SearchString$)
337
338     If i% = -1 Then
339         Acct2Txcd$ = "N"
340     Else
341         mLen% = Len(oAWAct.List(i%))
342         mAcct$ = Trim(Right(oAWAct.List(i%), mLen% - ADigit))
343         mLen2% = Len(mAcct$)
344         Acct2Txcd$ = Left(Right(mAcct$, mLen2% - 1), 1)
345     End If
346 End Function
347 Function get1stLetter$(ByVal fName$)
348     Dim iPos%, gName$
349     iPos% = InStr(fName$, "—") + InStr(fName$, "—") + InStr(fName$, "-")
350     gName$ = Trim(Right(fName$, Len(fName$) - iPos%))

```

```
If Left(gName$,1)=chr(&H3000) Then
    gName$=Right(gName$, Len(gName$)-1)
End If
get1stLetter$=Left(gName$,1)
End Function
```

```
351 If Left(gName$, 1) = Chr(&H3000) Then
352     gName$ = Right(gName$, Len(gName$) - 1)
353 End If
354 get1stLetter$ = Left(gName$, 1)
355 End Function
```