

```

Rem =====
Rem V_Date Administration OOoBasic Program (Copyright:Shuichi Sunaga)
Rem (2006/10/14:289)(2009/01/28:387)(2009/10/13:352)
Rem (2011/01/28:434)(2011/12/30:488)
Rem Translation into English(2012/08/16:401)
Rem -----
Rem Sub DDList()
Rem Sub DefCMoList()
Rem Sub DefSMoList()
Rem Sub setCurrYMD()
Rem Sub DirectSetCurrYMD(ByVal ForB%)
Rem Sub setStrtYMD()
Rem Sub GiveDTHHeader()
Rem Sub GetFiscalDates()
Rem Sub CheckCurrDate()
Rem Sub CheckStrtDate()
Rem Sub FilValMo() For input of monthly fixed amount transactions(not used)
Rem Function NewNextDate(ByVal mCDate as string, ByVal iVar%, ByVal SvE$)
Rem Function GetDate(ByVal when$, ByVal what$)
Rem Function GetYear(ByVal cYMD$) as integer
Rem Function GetMonth(ByVal cYMD$) as integer
Rem Function GetDay(ByVal cYMD$) as integer
Rem Function MthDue(mWhen%) As Integer
Rem Function NxtMonth%(ByVal mWhen%) 'Called from SeekTBRow()
Rem Function BfrMonth%(ByVal mWhen%) 'Called from SeekTBRow()
Rem Function MthDiff%(ByVal sWhen%, eWhen%) Called from Rem GetFiscalDates()
Rem Function Mth_Col%(eCol%) Rem used in double_click analysis
Rem Function GoodUntilDate(ByVal rDate$, ByVal NumDates%)
Rem Function IllegalDT(ByVal mYr%, ByVal mMo%, ByVal mDT%) as Boolean
Rem =====
Option Explicit
Sub DDList()
    Dim i%, DTB4Change$, oCD As Object
    IntroTo3D()
    Set oDTDlg = CreateUnoDialog(DialogLibraries.OOo3D.DTDlg)
    Set oCurrYMD = oDTDlg.GetControl("CurrYMD")
    Set oStrtYMD = oDTDlg.GetControl("StrtYMD")
    DefCMoList
    DefSMoList
    Rem ----- Initialization of PrintJobListner & MouseListner -----
    oMListener = CreateUnoListener("MouseListener_", "com.sun.star.awt.XMouseListener")
    oDTDlg.Model.PositionX = 3
    oDTDlg.Model.PositionY = 3
    oDTDlg.execute()
End Sub
Sub DefCMoList()
    Dim i%, preList$(11+3)
    Rem -----(Current months listed in List box)-----
    preList$(0) = GetDate("Beg", "Date")
    preList$(0) = NewNextDate(preList$(0), 45, "E")
    For i% = 1 to 11+3
        preList$(i) = NewNextDate(preList$(i-1), 45, "E")
    Next i%
    oCurrYMD.AddItem(preList$(i), 1)
    oCurrYMD.setText(GetDate("Curr", "Date"))
End Sub
Sub DefSMoList()
    Dim i%, preList$(30), ArrayIndex%, m$
    Rem -----(Current months listed in List box)-----
    preList$(0) = CStr(GetDate("Beg", "Year")) & "/" & "01/01"
    oStrtYMD.removeItems(0, oStrtYMD.ItemCount)
    oStrtYMD.addItem(preList$(0), 0)
    i% = 1
    Do while True
        preList$(i) = NewNextDate(preList$(i-1), 45, "S")
        m$ = (GetYear(preList$(i))-1900)*100 + GetMonth(preList$(i))
        If Val(m$) > cMonth then
            Exit Do
        End If
        oStrtYMD.addItem(preList$(i), i%)
    End Sub

```

```

1 Rem =====
2 Rem V_Date Administration Excel VBA Program (Copyright:Shuichi Sunaga)
3 Rem
4 Rem (2011/08/16:377)(2012/01/26:390)
5 Rem Translation into English(2012/08/16:384)
6 Rem -----
7 Rem Sub DDList()
8 Rem Sub DefCMoList()
9 Rem Sub DefSMoList()
10 Rem Sub setCurrYMD()
11 Rem Sub DirectSetCurrYMD(ByVal ForB%)
12 Rem Sub setStrtYMD()
13 Rem Sub GiveDTHHeader()
14 Rem Sub GetFiscalDates()
15 Rem Sub CheckCurrDate()
16 Rem Sub CheckStrtDate()
17 Rem Sub FilValMo() For input of monthly fixed amount transactions(not used)
18 Rem Function NewNextDate(ByVal mCDate as string, ByVal iVar%, ByVal SvE$)
19 Rem Function GetDate(ByVal when$, ByVal what$)
20 Rem Function GetYear(ByVal cYMD$) as integer
21 Rem Function GetMonth(ByVal cYMD$) as integer
22 Rem Function GetDay(ByVal cYMD$) as integer
23 Rem Function MthDue(mWhen%) As Integer
24 Rem Function NxtMonth%(ByVal mWhen%) 'Called from SeekTBRow()
25 Rem Function BfrMonth%(ByVal mWhen%) 'Called from SeekTBRow()
26 Rem Function MthDiff%(ByVal sWhen%, eWhen%) Called from Rem GetFiscalDates()
27 Rem Function Mth_Col%(eCol%) Rem used in double_click analysis
28 Rem Function GoodUntilDate(ByVal rDate$, ByVal NumDates%)
29 Rem Function IllegalDT(ByVal mYr%, ByVal mMo%, ByVal mDT%) as Boolean
30 Rem =====
31 Option Explicit
32 Sub DDList()
33     Dim i%, DTB4Change$
34     IntroTo3D
35     Set oDTDlg = DTDlg
36     Set oCurrYMD = oDTDlg.CurrYMD
37     Set oStrtYMD = oDTDlg.StrtYMD
38     DefCMoList
39     DefSMoList
40     Rem ----- Initialization of PrintJobListner & MouseListner -----
41
42     oDTDlg.Left = 3
43     oDTDlg.Top = 3
44     oDTDlg.Show
45 End Sub
46 Sub DefCMoList()
47     Dim i%, preList$, k%
48     Rem -----(Current months listed in List box)-----
49     preList$ = GetDate("Beg", "Date")
50     oCurrYMD.AddItem NewNextDate(preList$, 45, "E")
51     For i% = 1 To 11 + 3
52         oCurrYMD.AddItem NewNextDate(oCurrYMD.List(i - 1), 45, "E")
53     Next i%
54
55     oCurrYMD.Value = GetDate("Curr", "Date")
56 End Sub
57 Sub DefSMoList()
58     Dim ArrayIndex%, m$, preList$
59     Rem -----(Current months listed in List box)-----
60     preList$ = CStr(GetDate("Beg", "Year")) & "/" & "01/01"
61     oStrtYMD.Clear
62     oStrtYMD.AddItem preList$
63
64     Do While True
65         preList$ = NewNextDate(preList$, 45, "S")
66         m$ = (GetYear(preList$) - 1900) * 100 + GetMonth(preList$)
67         If Val(m$) > cMonth Then
68             Exit Do
69         End If
70         oStrtYMD.AddItem preList$

```

```

        i% = i% + 1
    Loop
    ArrayIndex% = IndexInArray(GetDate("Strt","Date"), oStrtYMD.Model.StringItemList)
    If ArrayIndex% > -1 then
        oStrtYMD.setText(oStrtYMD.Model.StringItemList(ArrayIndex%))
    Else
        CheckStrtDate()
    End If
End Sub
Sub setCurrYMD() 'triggered by cahnge action of StrtYMD in DTDlg
    dim newYMD$, i%, ArrayIndex%
    newYMD$ = oCurrYMD.getText()
    If newYMD$ <> GetDate("Curr","Date") then
        SetCellString WkCL, 6, 2, newYMD$
        ThisComponent.calculateAll()
        GiveDTHHeader()
        GetFiscalDates()
        For i% = 0 to 4
            RowArray(2, i%) = 0
        Next i%
        Rem -----(refresh starting dates in the List Box
        DefSMoList()
        ArrayIndex% = IndexInArray(GetDate("Strt","Date"), oStrtYMD.Model.StringItemList)
        If ArrayIndex% > -1 then
            oStrtYMD.setText(oStrtYMD.Model.StringItemList(ArrayIndex%))
        End If
    End If
End Sub
Sub DirectSetCurrYMD() 'triggered by change action of StrtYMD in DTDlg
    dim iRow&, preMonth%
    preMonth%=cMonth%
    setCurrYMD()
    Set aSheet = aController.ActiveSheet
    aSName$ = aSheet.getName()
    If preMonth%<>cMonth% Then
        getPagePlan(aSName$, cMonth%)
        aSheet.UnProtect("pwd")
        If aSName$="BK" or aSName$="JV" then
            If Not IsCMDTNull Then
                SelectCell dbMonthTop&, SelectionColumn%
                RNMS=cna$ & CStr(dbMonthTop&) & cjc$ & cna$ & Cstr(dbMonthBtm&)
                aSheet.getCellRangeByName(RNMS).Rows.IsVisible = True
                If aSName$="BK" Then
                    iRow&=SelectionRow&
                    PostBKBF(iRow&)
                    CalcBKBal(iRow&+1)
                End If
            Else
                MsgTtl$ = aSName$ & " sheet has no current month transactions input."
                MsgBox(MsgTxt$, 0, MsgTtl$)
            End If
        End If
    End If
End Sub
Sub setStrtYMD()
    StrtYMD = oStrtYMD.getText()
    If StrtYMD <> GetDate("Strt","Date") Then
        SetCellString WkCL, 7, 2, StrtYMD
        ThisComponent.calculateAll()
        StrtMo% = GetDate("Strt","FYM")
    End If
End Sub
Sub GiveDTHHeader()
    dim oStyleFamilies As Object, oPageStyles As Object
    dim oPageStyle As Object, HContent as Object
    dim i%, n%, mSpace$, sReportDT$
    Set oStyleFamilies = ThisComponent.getStyleFamilies()
    Set oPageStyles = oStyleFamilies.getByname( "PageStyles" )
    mSpace$=chr$(32)
    For i% = 1 to 10

```

```

71
72 Loop
73 ArrayIndex% = IndexInArray(GetDate("Strt", "Date"), oStrtYMD)
74 If ArrayIndex% > -1 Then
75     oStrtYMD.Value = oStrtYMD.List(ArrayIndex%)
76 Else
77     CheckStrtDate
78 End If
79 End Sub
80 Sub setCurrYMD()
81     Dim newYMD$, i%, ArrayIndex%
82     newYMD$ = oCurrYMD.Value
83     If newYMD$ <> GetDate("Curr", "Date") Then
84         SetCellString WkCL, 6, 2, newYMD$
85         Calculate
86         GiveDTHHeader
87         GetFiscalDates
88         For i% = 0 To 4
89             RowArray(2, i%) = 0
90         Next i%
91         Rem -----(refresh starting dates in the List Box
92         DefSMoList
93         ArrayIndex% = IndexInArray(GetDate("Strt", "Date"), oStrtYMD)
94         If ArrayIndex% > -1 Then
95             oStrtYMD.Value = oStrtYMD.List(ArrayIndex%)
96         End If
97     End If
98 End Sub
99 Sub DirectSetCurrYMD() 'triggered by change action of StrtYMD in DTDlg
100     Dim iRow&, preMonth%
101     preMonth% = cMonth%
102     setCurrYMD
103     Set aSheet = ActiveSheet
104     aSName$ = aSheet.Name
105     If preMonth% <> cMonth% Then
106         getPagePlan aSName$, cMonth%
107         aSheet.Unprotect ("pwd")
108         If aSName$ = "BK" Or aSName$ = "JV" Then
109             If Not IsCMDTNull Then
110                 SelectCell dbMonthTop&, SelectionColumn%
111                 RNMS = "$A$" & CStr(dbMonthTop&) & "$A$" & CStr(dbMonthBtm&)
112                 aSheet.Range(RNMS).EntireRow.Hidden = False
113                 If aSName$ = "BK" Then
114                     iRow& = SelectionRow&
115                     PostBKBF (iRow&)
116                     CalcBKBal (iRow& + 1)
117                 End If
118             Else
119                 MsgTtl$ = aSName$ & " sheet has no current month transactions input."
120                 resp% = MsgBox(MsgTxt$, 0, MsgTtl$)
121             End If
122         End If
123     End If
124 End Sub
125 Sub setStrtYMD()
126     StrtYMD = oStrtYMD.Value
127     If StrtYMD <> GetDate("Strt", "Date") Then
128         SetCellString WkCL, 7, 2, StrtYMD
129         Calculate
130         StrtMo% = GetDate("Strt", "FYM")
131     End If
132 End Sub
133 Sub GiveDTHHeader()
134
135
136     Dim i%, mSpace$, sReportDT$, oSheet As Worksheet
137
138
139     mSpace$ = Chr$(32)
140     For i% = 1 To 10

```

```

        mSPace$=mSpace$ & chr$(32)
Next i%
sReportDT$ = "As of " & GetDate("Curr","Date") & mSpace$
n% = oPageStyles.getcount() - 1
For i% = 0 to n%
    Set oPageStyle = oPageStyles.getByIndex(i)
    HContent = oPageStyle.RightPageHeaderContent
    If HContent.LeftText.String = CoName Then
        HContent.RightText.String = sReportDT$
        oPageStyle.RightPageHeaderContent = HContent
        oPageStyle.HeaderIsOn = True
    End If
Next i%
End Sub
Rem =====
Sub GetFiscalDates()
    Dim Col_RB$, oRange As object
    cMonth% = GetDate("Curr","FYM")
    StrtYMD$ = GetDate("Strt","Date")
    StrtMo% = GetDate("Strt","FYM")
    NumMo% = MthDiff%(GetDate("Beg","FYM"), cMonth%)+1
    Rem -----(Get column number for BS/PL/Dept sheets: 2008/10/03)-----
    Col_RB$ = "Col_ " & CStr(CellValue(WkCL, 5, 3) * 100 + CellValue(WkCL, 5, 4))
    CBS_YE% = Col_InFR%("BS", Col_RB$)
    Col_RB$ = "Col_ " & CStr(CellValue(WkCL, 4, 3) * 100 + CellValue(WkCL, 4, 4))
    CBS_YB% = Col_InFR%("BS", Col_RB$)
    Col_RB$ = "Col_ " & CStr(CellValue(WkCL, 6, 3) * 100 + CellValue(WkCL, 6, 4))
    CBS_CM% = Col_InFR%("BS", Col_RB$)
    Rem -----
    CheckStrtDate()
End Sub
Sub CheckStrtDate()
    If GetDate("Strt","FYM") > GetDate("Curr","FYM") then
        StrtYMD = DateSerial(GetDate("Curr","Year"), GetDate("Curr","Month"), 1)
        MsgTxt$ = "The starting date is after the current month." & chr(13)
        MsgTxt$ = MsgTxt$ & " Will let the starting date change to " & StrtYMD & "."
        MsgTtl$ = "Test of starting date"
        resp% = MsgBox(MsgTxt$, 0, MsgTtl$)
        SetCellString WkCL, 7, 2, StrtYMD
    End If
End Sub
Sub FilValMo()
    Dim i%, preList%(11), iRow&, EndDate%
    MsgTtl$ = "Test of the cursor position"
    If aSName$="AP" Or aSName$="AR" Or aSName$="MW" then
        iRow& = SelectionRow()
        getRowArray aSName$, "ValMo", cMonth%, cMonth%
        If iRow&>=dbBtmBorder&-1 Then
            MsgTxt$ = "The cursor is outside of the data input area" & chr(13)
            MsgTxt$ = MsgTxt$ & "Move the cursor inside of the data input area and retry."
            MsgBox(MsgTxt$, 0, MsgTtl$)
            Exit Sub
        Else
            MsgTtl$ = "Confirmation of row insertion by the year end"
            MsgTxt$ = "Will insert rows beneath the current row and input " & chr(13)
            MsgTxt$ = MsgTxt$ & "each month from starting month to year-end month."
            resp% = MsgBox(MsgTxt$, 1, MsgTtl$)
            If resp%=2 then
                Exit Sub
            End If
        End If
    End If
    getColArray(aSName$)
    preList%(0)=StrtMo%
    EndDate% = GetDate("End","FYM")
    aSheet.UnProtect("pwd")
    For i% = 0 to 11
        aSheet.Rows.InsertByIndex(iRow&, 1)
        iRow& = iRow& + 1
        SetCellValue aSheet, iRow&, Col_ValMo%, preList%(i%)
        preList%(i%+1) = NxtMonth%(preList%(i%))
    
```

```

        mSPace$ = mSpace$ & Chr$(32)
Next i%
sReportDT$ = "As of " & GetDate("Curr", "Date") & mSpace$
For Each oSheet In Sheets
    If oSheet.PageSetup.LeftHeader = CoName Then
        oSheet.PageSetup.RightHeader = sReportDT$
    End If
Next oSheet
End Sub
Rem =====
Sub GetFiscalDates()
    Dim Col_RB$
    cMonth% = GetDate("Curr", "FYM")
    StrtYMD = GetDate("Strt", "Date")
    StrtMo% = GetDate("Strt", "FYM")
    NumMo% = MthDiff%(GetDate("Beg", "FYM"), cMonth%) + 1
    Rem -----(Get column number for BS/PL/Dept sheets: 2008/10/03)-----
    Col_RB$ = "Col_ " & CStr(CellValue(WkCL, 5, 3) * 100 + CellValue(WkCL, 5, 4))
    CBS_YE% = Col_InFR%("BS", Col_RB$)
    Col_RB$ = "Col_ " & CStr(CellValue(WkCL, 4, 3) * 100 + CellValue(WkCL, 4, 4))
    CBS_YB% = Col_InFR%("BS", Col_RB$)
    Col_RB$ = "Col_ " & CStr(CellValue(WkCL, 6, 3) * 100 + CellValue(WkCL, 6, 4))
    CBS_CM% = Col_InFR%("BS", Col_RB$)
    Rem -----
    CheckStrtDate()
End Sub
Sub CheckStrtDate()
    If GetDate("Strt", "FYM") > GetDate("Curr", "FYM") Then
        StrtYMD = DateSerial(GetDate("Curr", "Year"), GetDate("Curr", "Month"), 1)
        MsgTxt$ = "The starting date is after the current month." & Chr(13)
        MsgTxt$ = MsgTxt$ & " Will let the starting date change to " & StrtYMD & "."
        MsgTtl$ = "Test of starting date"
        resp% = MsgBox(MsgTxt$, vbOKOnly, MsgTtl$)
        SetCellString WkCL, 7, 2, StrtYMD
    End If
End Sub
Sub FilValMo()
    Dim i%, preList%(11), iRow&, EndDate%
    MsgTtl$ = "Test of the cursor position"
    If aSName$ = "AP" Or aSName$ = "AR" Or aSName$ = "MW" Then
        iRow& = SelectionRow()
        getRowArray aSName$, "ValMo", cMonth%, cMonth%
        If iRow& >= dbBtmBorder& - 1 Then
            MsgTxt$ = "The cursor is outside of the data input area" & Chr(13)
            MsgTxt$ = MsgTxt$ & "Move the cursor inside of the data input area and retry."
            resp% = MsgBox(MsgTxt$, vbOKOnly, MsgTtl$)
            Exit Sub
        Else
            MsgTtl$ = "Confirmation of row insertion by the year end"
            MsgTxt$ = "Will insert rows beneath the current row and input " & Chr(13)
            MsgTxt$ = MsgTxt$ & "each month from starting month to year-end month."
            resp% = MsgBox(MsgTxt$, vbOKCancel, MsgTtl$)
            If resp% = vbCancel Then
                Exit Sub
            End If
        End If
    End If
    getColArray(aSName$)
    preList%(0) = StrtMo%
    EndDate% = GetDate("End", "FYM")
    aSheet.Unprotect ("pwd")
    For i% = 0 To 11
        aSheet.Cells(iRow& + 1, 1).EntireRow.Insert
        iRow& = iRow& + 1
        SetCellValue aSheet, iRow&, Col_ValMo%, preList%(i%)
        preList%(i% + 1) = NxtMonth%(preList%(i%))
    
```

```

        If preList%(i%)=EndDate% Then
            Exit For
        End If
        Next i%
        aSheet.Protect("pwd")
    Else
        MsgTxt$ = "This command is unusable on this "&aSName$& " sheet"
        MsgBox (MsgTxt$, 0, MsgTtl$)
    End If
End Sub
Rem =====(Functions)=====
Function NewNextDate(ByVal mCDate$, ByVal iVar&, ByVal SvE$)
    Dim i%, nYMD&, mYMD&
    mYMD& = DateValue(CDate(mCDate$)) + iVar&
    If SvE$ = "E" then
        nYMD& = DateValue(DateSerial(Year(mYMD&), Month(mYMD&), 1)) - 1
        NewNextDate = DateSerial(Year(nYMD&), Month(nYMD&), Day(nYMD&))
    ElseIf SvE$ = "S" then
        nYMD& = DateValue(DateSerial(Year(mYMD&), Month(mYMD&), 1))
        NewNextDate = DateSerial(Year(nYMD&), Month(nYMD&), 1)
    End If
End Function
Function GetDate(ByVal when$, ByVal what$)
    Dim sRow&, sCol%
    If IsNull(WkCL) then
        Set aController = ThisComponent.CurrentController
        Set oSheets = ThisComponent.Sheets
        Set WkCL = oSheets.getByName("CL")
    End If
    sRow& = 4
    sCol% = 3
    Select Case when$
        Case "Beg"
            select case what$
                case "Date"
                    GetDate = CellString(WkCL,sRow&,sCol%-1)
                case "Year"
                    GetDate = CellValue(WkCL,sRow&,sCol%)
                case "Month"
                    GetDate = CellValue(WkCL,sRow&,sCol%+1)
                case "Day"
                    GetDate = CellValue(WkCL,sRow&,sCol%+2)
                case "FYM"
                    GetDate = CellValue(WkCL,sRow&,sCol%+3)
            end select
        Case "End"
            select case what$
                case "Date"
                    GetDate = CellString(WkCL,sRow&+1,sCol%-1)
                case "Year"
                    GetDate = CellValue(WkCL,sRow&+1,sCol%)
                case "Month"
                    GetDate = CellValue(WkCL,sRow&+1,sCol%+1)
                case "Day"
                    GetDate = CellValue(WkCL,sRow&+1,sCol%+2)
                case "FYM"
                    GetDate = CellValue(WkCL,sRow&+1,sCol%+3)
            end select
        Case "Curr"
            select case what$
                case "Date"
                    GetDate = CellString(WkCL,sRow&+2,sCol%-1)
                case "Year"
                    GetDate = CellValue(WkCL,sRow&+2,sCol%)
                case "Month"
                    GetDate = CellValue(WkCL,sRow&+2,sCol%+1)
                case "Day"
                    GetDate = CellValue(WkCL,sRow&+2,sCol%+2)
                case "FYM"
                    GetDate = CellValue(WkCL,sRow&+2,sCol%+3)
            end select
    End Select
End Function

```

```

211         If preList%(i%) = EndDate% Then
212             Exit For
213         End If
214         Next i%
215         aSheet.Protect ("pwd")
216     Else
217         MsgTxt$ = "This command is unusable on this " & aSName$ & " sheet"
218         resp% = MsgBox(MsgTxt$, vbOKOnly, MsgTtl$)
219     End If
220 End Sub
221 Rem =====(Functions)=====
222 Function NewNextDate(ByVal mCDate$, ByVal iVar&, ByVal SvE$)
223     Dim i%, nYMD&, mYMD&
224     mYMD& = DateValue(CDate(mCDate$)) + iVar&
225     If SvE$ = "E" Then
226         nYMD& = DateValue(DateSerial(Year(mYMD&), Month(mYMD&), 1)) - 1
227         NewNextDate = DateSerial(Year(nYMD&), Month(nYMD&), Day(nYMD&))
228     ElseIf SvE$ = "S" Then
229         nYMD& = DateValue(DateSerial(Year(mYMD&), Month(mYMD&), 1))
230         NewNextDate = DateSerial(Year(nYMD&), Month(nYMD&), 1)
231     End If
232 End Function
233 Function GetDate(ByVal when$, ByVal what$)
234     Dim sRow&, sCol%
235
236
237
238     Set WkCL = Workbooks(GLName).Worksheets("CL")
239
240     sRow& = 4
241     sCol% = 3
242     Select Case when$
243         Case "Beg"
244             Select Case what$
245                 Case "Date"
246                     GetDate = CellString(WkCL, sRow&, sCol% - 1)
247                 Case "Year"
248                     GetDate = CellValue(WkCL, sRow&, sCol%)
249                 Case "Month"
250                     GetDate = CellValue(WkCL, sRow&, sCol% + 1)
251                 Case "Day"
252                     GetDate = CellValue(WkCL, sRow&, sCol% + 2)
253                 Case "FYM"
254                     GetDate = CellValue(WkCL, sRow&, sCol% + 3)
255             End Select
256         Case "End"
257             Select Case what$
258                 Case "Date"
259                     GetDate = CellString(WkCL, sRow& + 1, sCol% - 1)
260                 Case "Year"
261                     GetDate = CellValue(WkCL, sRow& + 1, sCol%)
262                 Case "Month"
263                     GetDate = CellValue(WkCL, sRow& + 1, sCol% + 1)
264                 Case "Day"
265                     GetDate = CellValue(WkCL, sRow& + 1, sCol% + 2)
266                 Case "FYM"
267                     GetDate = CellValue(WkCL, sRow& + 1, sCol% + 3)
268             End Select
269         Case "Curr"
270             Select Case what$
271                 Case "Date"
272                     GetDate = CellString(WkCL, sRow& + 2, sCol% - 1)
273                 Case "Year"
274                     GetDate = CellValue(WkCL, sRow& + 2, sCol%)
275                 Case "Month"
276                     GetDate = CellValue(WkCL, sRow& + 2, sCol% + 1)
277                 Case "Day"
278                     GetDate = CellValue(WkCL, sRow& + 2, sCol% + 2)
279                 Case "FYM"
280                     GetDate = CellValue(WkCL, sRow& + 2, sCol% + 3)
            End Select
        End Select
    End Function

```

```

        end select
    Case "Strt"
        select case what$
            case "Date"
                GetDate = CellString(WkCL,sRow&+3,sCol%-1)
            case "Year"
                GetDate = CellValue(WkCL,sRow&+3,sCol%)
            case "Month"
                GetDate = CellValue(WkCL,sRow&+3,sCol%+1)
            case "Day"
                GetDate = CellValue(WkCL,sRow&+3,sCol%+2)
            case "FYM"
                GetDate = CellValue(WkCL,sRow&+3,sCol%+3)
        end select
    End Select
End Function
Function GetYear%(ByVal cYMD$)
    GetYear% = Val(Left(cYMD,4))
End Function
Function GetMonth%(ByVal cYMD$)
    GetMonth% = Val(Mid(cYMD,6,2))
End Function
Function GetDay%(ByVal cYMD$)
    GetDay% = Val(Right(cYMD,2))
End Function
Function MthDue%(ByVal mWhen%)
    dim pyrmo$, NewYrMo as Date
    pyrmo$ = GetDate("Curr", "Date")
    If GetMonth(pyrmo$) + mWhen% + 1 < 13 then
        NewYrMo = DateSerial(GetYear(pyrmo$), GetMonth(pyrmo$) + mWhen% + 1, 1) - 1
    Else
        NewYrMo = DateSerial(GetYear(pyrmo$)+1, GetMonth(pyrmo$) + mWhen% - 11, 1) - 1
    End If
    MthDue% = (Year(NewYrMo) - 1900) * 100 + Month(NewYrMo)
End Function
Function NxtMonth%(ByVal mWhen%) 'called from SeekTbRow()
    dim iYear%, iMonth%
    iYear% = Int(mWhen/100)
    iMonth% = mWhen% - iYear% * 100
    If iMonth% < 12 then
        NxtMonth% = iYear% * 100 + iMonth% + 1
    Else
        NxtMonth% = (iYear% + 1)*100 + 1
    End If
End Function
Function BfrMonth%(ByVal mWhen%) 'called from SeekTbRow()
    dim iYear%, iMonth%
    iYear% = Int(mWhen%/100)
    iMonth% = mWhen% - iYear% * 100
    If iMonth% > 1 then
        BfrMonth% = iYear% * 100 + iMonth% - 1
    Else
        BfrMonth% = (iYear% - 1)*100 + 12
    End If
End Function
Function MthDiff%(ByVal sWhen%, eWhen%) 'called from GetFiscalDates()
    dim wMonth%
    MthDiff% = 0
    wMonth% = sWhen%
    Do While wMonth < eWhen%
        wMonth = NxtMonth(wMonth%)
        MthDiff% = MthDiff% + 1
    Loop
End Function
Function Mth_Col%(eCol%) 'used in Double Click analysis
    Dim EndMo%, i%
    Mth_Col% = GetDate("Beg", "FYM")
    If eCol% <= RMCOL+5 Then
        Exit Function
    Elseif eCol% >= RMCOL+17 Then

```

```

281     End Select
282     Case "Strt"
283         Select Case what$
284             Case "Date"
285                 GetDate = CellString(WkCL, sRow& + 3, sCol% - 1)
286             Case "Year"
287                 GetDate = CellValue(WkCL, sRow& + 3, sCol%)
288             Case "Month"
289                 GetDate = CellValue(WkCL, sRow& + 3, sCol% + 1)
290             Case "Day"
291                 GetDate = CellValue(WkCL, sRow& + 3, sCol% + 2)
292             Case "FYM"
293                 GetDate = CellValue(WkCL, sRow& + 3, sCol% + 3)
294         End Select
295     End Select
296 End Function
297 Function GetYear%(ByVal cYMD$)
298     GetYear% = Val(Left(cYMD$, 4))
299 End Function
300 Function GetMonth%(ByVal cYMD$)
301     GetMonth% = Val(Mid(cYMD$, 6, 2))
302 End Function
303 Function GetDay%(ByVal cYMD$)
304     GetDay% = Val(Right(cYMD$, 2))
305 End Function
306 Function MthDue%(ByVal mWhen%)
307     Dim pyrmo$, NewYrMo As Date
308     pyrmo$ = GetDate("Curr", "Date")
309     If GetMonth(pyrmo$) + mWhen% + 1 < 13 Then
310         NewYrMo = DateSerial(GetYear(pyrmo$), GetMonth(pyrmo$) + mWhen% + 1, 1) - 1
311     Else
312         NewYrMo = DateSerial(GetYear(pyrmo$) + 1, GetMonth(pyrmo$) + mWhen% - 11, 1) - 1
313     End If
314     MthDue% = (Year(NewYrMo) - 1900) * 100 + Month(NewYrMo)
315 End Function
316 Function NxtMonth%(ByVal mWhen%) 'called from SeekTbRow()
317     Dim iYear%, iMonth%
318     iYear% = Int(mWhen / 100)
319     iMonth% = mWhen% - iYear% * 100
320     If iMonth% < 12 Then
321         NxtMonth% = iYear% * 100 + iMonth% + 1
322     Else
323         NxtMonth% = (iYear% + 1) * 100 + 1
324     End If
325 End Function
326 Function BfrMonth%(ByVal mWhen%) 'called from SeekTbRow()
327     Dim iYear%, iMonth%
328     iYear% = Int(mWhen / 100)
329     iMonth% = mWhen% - iYear% * 100
330     If iMonth% > 1 Then
331         BfrMonth% = iYear% * 100 + iMonth% - 1
332     Else
333         BfrMonth% = (iYear% - 1) * 100 + 12
334     End If
335 End Function
336 Function MthDiff%(ByVal sWhen%, eWhen%) 'called from GetFiscalDates()
337     Dim wMonth%
338     MthDiff% = 0
339     wMonth% = sWhen%
340     Do While wMonth < eWhen%
341         wMonth = NxtMonth(wMonth%)
342         MthDiff% = MthDiff% + 1
343     Loop
344 End Function
345 Function Mth_Col%(eCol%) 'used in Double Click analysis
346     Dim EndMo%, i%
347     Mth_Col% = GetDate("Beg", "FYM")
348     If eCol% <= RMCOL + 5 Then
349         Exit Function
350     Elseif eCol% >= RMCOL + 17 Then

```

```

Mth_Col% = GetDate("End", "FYM")
Else
    For i% = 1 to eCol% - (RMCOL+5)
        Mth_Col% = NxtMonth(Mth_Col%)
    Next i%
End If
End Function
Function IllegalDT(ByVal mYr%, ByVal mMo%, ByVal mDT%) as Boolean
    dim MsgAll$
    MsgTtl$ = "Test of information completeness"
    IllegalDT = False
    If mMo% = 0 or mDT% = 0 then
        MsgTxt$ = "Input "&myIIF(mMo%=0, "month", "day")
        MsgTxt$ = MsgTxt$ & " of the planned settlement date."
        MsgBox(MsgTxt$, 16, MsgTtl$)
        IllegalDT = True
    Exit Function
End If
If mMo% > 12 then
    MsgTxt$ = "The month of planned settlement date must be 12 or lesser."
    IllegalDT = True
End If
Select Case mMo%
    Case 1, 3, 5, 7, 8, 10, 12
        If mDT% > 31 then
            MsgTxt$ = "The day of planned settlement date must be 31 or lesser."
            IllegalDT = True
        End If
    Case 2
        If mYr%/4 - Int(mYr%/4) = 0 then
            if mDT% > 29 then
                MsgTxt$ = "The day of planned settlement date must be 29 or lesser."
                IllegalDT = True
            end if
        Else
            if mDT% > 28 then
                MsgTxt$ = "The day of planned settlement date must be 28 or lesser."
                IllegalDT = True
            end if
        End If
    Case 4, 6, 9, 11
        If mDT% > 30 then
            MsgTxt$ = "The day of planned settlement date must be 30 or lesser."
            IllegalDT = True
        End If
End Select
If IllegalDT = True then
    MsgAll$ = "The planned settlement date is not on the calendar"
    MsgBox(MsgAll$, 16, MsgTtl$)
End If
End Function

```

```

351 Mth_Col% = GetDate("End", "FYM")
352 Else
353     For i% = 1 To eCol% - (RMCOL + 5)
354         Mth_Col% = NxtMonth(Mth_Col%)
355     Next i%
356 End If
357 End Function
358 Function IllegalDT(ByVal mYr%, ByVal mMo%, ByVal mDT%) As Boolean
359     Dim MsgAll$
360     MsgTtl$ = "Test of information completeness"
361     IllegalDT = False
362     If mMo% = 0 Or mDT% = 0 Then
363         MsgTxt$ = "Input " & myIIF(mMo% = 0, "month", "day")
364         MsgTxt$ = MsgTxt$ & " of the planned settlement date."
365         resp% = MsgBox(MsgTxt$, vbCritical, MsgTtl$)
366         IllegalDT = True
367     Exit Function
368 End If
369 If mMo% > 12 Then
370     MsgTxt$ = "The month of planned settlement date must be 12 or lesser."
371     IllegalDT = True
372 End If
373 Select Case mMo%
374     Case 1, 3, 5, 7, 8, 10, 12
375         If mDT% > 31 Then
376             MsgTxt$ = "The day of planned settlement date must be 31 or lesser."
377             IllegalDT = True
378         End If
379     Case 2
380         If mYr% / 4 - Int(mYr% / 4) = 0 Then
381             If mDT% > 29 Then
382                 MsgTxt$ = "The day of planned settlement date must be 29 or lesser."
383                 IllegalDT = True
384             End If
385         Else
386             If mDT% > 28 Then
387                 MsgTxt$ = "The day of planned settlement date must be 28 or lesser."
388                 IllegalDT = True
389             End If
390         End If
391     Case 4, 6, 9, 11
392         If mDT% > 30 Then
393             MsgTxt$ = "The day of planned settlement date must be 30 or lesser."
394             IllegalDT = True
395         End If
396 End Select
397 If IllegalDT = True Then
398     MsgAll$ = "The planned settlement date is not on the calendar"
399     resp% = MsgBox(MsgAll$, vbCritical, MsgTtl$)
400 End If
401 End Function

```