```
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 GiveDTHeader()
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.AddItems(preList$(), 1)
	oCurrYMD.setText(GetDate("Curr","Date"))
End Sub
Sub DefSMoList()
	dim i%, preList$(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%)
```

```
Rem ============================================================
Rem V_Date Administration Excel VBA Program (Copyright:Shuichi Sunaga)
Rem
Rem (2011/08/16:377)(2012/01/26:390)
Rem Translation into English(2012/08/16:384)
Rem ------------------------------------------------------------
Rem Sub DDList()
Rem Sub DefCMoList()
Rem Sub DefSMoList()
Rem Sub setCurrYMD()
Rem Sub DirectSetCurrYMD(ByVal ForB%)
Rem Sub setStrtYMD()
Rem Sub GiveDTHeader()
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$, ByVal NumDates%)
Rem Function IllegalDT(ByVal mYr%, ByVal mMo%, ByVal mDT%) as Boolean
Rem ============================================================
Option Explicit
Sub DDList()
	Dim i%, DTB4Change$
	IntroTo3D
	Set oDTDlg = DTDlg
	Set oCurrYMD = oDTDlg.CurrYMD
	Set oStrtYMD = oDTDlg.StrtYMD
	DefCMoList
	DefSMoList
	Rem ---------- Initialization of PrintJobListner & MouseListner -----------------

	oDTDlg.Left = 3
	oDTDlg.Top = 3
	oDTDlg.Show
End Sub
Sub DefCMoList()
	Dim i%, preList$, k%
	Rem ----------(Current months listed in List box)-----
	preList$ = GetDate("Beg", "Date")
	oCurrYMD.AddItem NewNextDate(preList$, 45, "E")
	For i% = 1 To 11 + 3
		oCurrYMD.AddItem NewNextDate(oCurrYMD.List(i - 1), 45, "E")
	Next i%

	oCurrYMD.Value = GetDate("Curr", "Date")
End Sub
Sub DefSMoList()
	Dim ArrayIndex%, m$, preList$
	Rem ----------(Current months listed in List box)-----
	preList$ = CStr(GetDate("Beg", "Year")) & "/01/01"
	oStrtYMD.Clear
	oStrtYMD.AddItem preList$

	Do While True
		preList$ = NewNextDate(preList$, 45, "S")
		m$ = (GetYear(preList$) - 1900) * 100 + GetMonth(preList$)
		If Val(m$) > cMonth Then
			Exit Do
		End If
		oStrtYMD.AddItem preList$
```

```
        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()
        GiveDTHeader()
        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$="JV" then
            If Not IsCMDTNull Then
                SelectCell dbMonthTop&,SelectionColumn%
                RNM$=cna$ & CStr(dbMonthTop&) & cjc$ & cna$ & Cstr(dbMonthBtm&)
                aSheet.getCellRangeByName(RNM$).Rows.IsVisible = True
                If aSName$="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 GiveDTHeader()
    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          GiveDTHeader
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                 RNM$ = "$A$" & CStr(dbMonthTop&) & ":$A$" & CStr(dbMonthBtm&)
112                 aSheet.Range(RNM$).EntireRow.Hidden = False
113                 If aSName$ = "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 GiveDTHeader()
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
        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
        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)
```

```
                If preList%(i%) = EndDate% Then
                    Exit For
                End If
            Next i%
            aSheet.Protect ("pwd")
        Else
            MsgTxt$ = "This command is unusable on this " & aSName$ & " sheet"
            resp% = MsgBox(MsgTxt$, vbOKOnly, MsgTtl$)
        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%



    Set WkCL = Workbooks(GLName).Worksheets("CL")

    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
            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
```

```
                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
```

```
            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
```