

```

Rem =====
Rem A_CashFlowStatement Preparation OOoBasic Program (Copyright:Shuichi Sunaga)
Rem (Ver. 2006/04/05:326)Adopted FindALL in FCAcctSum()(2006/05/14:549)
Rem (Ver. 2006/10/14:293) integration to ND(2009/01/28:306)
Rem Integration to ND(Ver. 2011/06/02:304)
Rem Added CFVeIByWS() and IsExistInCF()(Ver 2011/06/10:408)(Ver 2012/08/07:400)
Rem -----
Rem Sub CFCalc()
Rem Sub CFPostXLn(ByVal StrtLn%)
Rem Function CFAcctSum(WkWk as object, FkAcct$, oSearchDesc, oSRange as object) As Long
Rem Sub CFfindPost(ByVal selActCD$, ByVal iStrtMo%)
Rem Sub CFVeriByWS()
Rem Sub IsExistInCF(ByVal WkWk As Object, ByVal StrtRow&)
Rem Function IsExclBC(fAcct$) As Boolean
Rem =====
Option Explicit
Sub CFCalc()
    dim BSAudLN&
    dim args(0) as new com.sun.star.beans.PropertyValue
    IntroTo3D()
    If CellValue(WkBS, getFSRow%(WkBS, "ENDLN", ActCol) + 1, CBS_CM% - 1) <> 0 Then
        MsgShow("CFInCompletePriorFS")
        Rem "Unable to prepare current month Cashflow statemnt" & Chr(13)
        Rem "because prior month Financial Statement is not complete."
        Rem MsgTtl$ = "Preparation of CF for "& CStr(GetDate("Curr", "Date"))
        End
    End If
    Rem ①--- Quit CFCalc where current month transaction is non-existent -----
    targetRow& = SeekTbRow&(WkBK, "ValMo", cMonth%, cMonth%, "Top")
    If IsCMDTNull then
        MsgTxt$ = "Abort preparation of Cashflow Statement because, " & Chr(13)
        MsgTxt$ = "Bank Sheet has no current month ("& CStr(cMonth%) & ") transactions input."
        MsgTtl$ = "Test for input transactions"
        msgbox(MsgTxt$, 0, MsgTtl$)
        End
    End If
    Rem ②----- Change to manual calculation mode -----
    args(0).Name = "AutomaticCalculation"
    args(0).Value = False
    dispatcher.executeDispatch(aController, ".uno:AutomaticCalculation", "", 0, args())
    ThisComponent.calculateAll()
    Rem ③--- Set Search Range for Data Summation (Do Not Change The Order) ---
    SetSrcRange "PaidMo", cMonth%, cMonth%, "ActCD"
    Rem ④----- Make CF Sheet Visible And Active -----
    WkCF.IsVisible = True
    aController.ActiveSheet = WkCF
    Rem ⑤ ----- Make sum of posting source columns on the CF -----
    WkCF.Unprotect("pwd")
    CFPostXLn(7)
    CFPostXFm
    Rem ----- Calculation for Due from Sole Proprietor -----
    If CorpOrIndv = "Indv" then
        CFToOwner()
    End If
    Rem =====
    CFPostBKBal()
    ThisComponent.calculateAll()
    BSAudLN&=getFSRow%(WkCF, "ENDLN", ActCol) + 2
    If CellValue(WkCF, BSAudLN&, BFCol-1)<>0 Then
        CFVeriByWS()
    End If
    args(0).Value = true
    dispatcher.executeDispatch(aController, ".uno:AutomaticCalculation", "", 0, args())
    oDTDlg.endExecute()
    BSPLShowCurrent("CF")
    WkCF.Protect("pwd")
End Sub
Sub CFPostXLn(ByVal StrtLn%)
    dim mSigma%, MBKCRow&, MBKCS$, tBFVal&, tCFVal&, vEndRow&
    dim sAName$, sEndCD$, WkAcct$, JVCalc$

```

```

1 Rem =====
2 Rem A_CashFlowStatement Preparation Excel VBA Program (Copyright:Shuichi Sunaga)
3 Rem
4 Rem
5 Rem (Ver 2012/08/07:389行)
6 Rem Translation to English(2012/08/09:389)
7 Rem -----
8 Rem Sub CFCalc()
9 Rem Sub CFPostXLn(ByVal StrtLn%)
10 Rem Function CFAcctSum(WkWk as object, FkAcct$, oSearchDesc, oSRange as object) As Long
11 Rem Sub CFfindPost(ByVal selActCD$, ByVal iStrtMo%)
12 Rem Sub CFVeriByWS()
13 Rem Sub IsExistInCF(ByVal WkWk As Object, ByVal StrtRow&)
14 Rem Function IsExclBC(fAcct$) As Boolean
15 Rem =====
16 Option Explicit
17 Sub CFCalc()
18     Dim BSAudLN&
19
20     IntroTo3D
21     If CellValue(WkBS, getFSRow%(WkBS, "ENDLN", ActCol) + 1, CBS_CM% - 1) <> 0 Then
22         MsgShow("CFInCompletePriorFS")
23         Rem "Unable to prepare current month Cashflow statemnt" & Chr(13)
24         Rem "because prior month Cashflow statement is not complete."
25         Rem MsgTtl$ = "Preparation of CF for "& CStr(GetDate("Curr", "Date"))
26         End
27     End If
28     Rem ①--- Quit CFCalc where current month transaction is non-existent -----
29     targetRow& = SeekTbRow&(WkBK, "ValMo", cMonth%, cMonth%, "Top")
30     If IsCMDTNull Then
31         MsgTxt$ = "Abort preparation of Cashflow Statement because, " & Chr(13)
32         MsgTxt$ = "Bank Sheet has no current month ("& CStr(cMonth%) & ") transactions input."
33         MsgTtl$ = "Test for input transactions"
34         resp% = MsgBox(MsgTxt$, 0, MsgTtl$)
35         End
36     End If
37     Rem ②----- Change to manual calculation mode -----
38     Application.Calculation = xlManual
39
40
41     Calculate
42     Rem ③--- Set Search Range for Data Summation (Do Not Change The Order) ---
43     SetSrcRange "PaidMo", cMonth%, cMonth%, "ActCD"
44     Rem ④----- Make CF Sheet Visible And Active -----
45     WkCF.Visible = True
46     WkCF.Activate
47     Rem ⑤ ----- Make sum of posting source columns on the CF -----
48     WkCF.Unprotect ("pwd")
49     CFPostXLn (7)
50     CFPostXFm
51     Rem ----- Calculation for Due from Sole Proprietor -----
52     If CorpOrIndv = "Indv" Then
53         CFToOwner
54     End If
55     Rem =====
56     CFPostBKBal
57     Calculate
58     BSAudLN&= getFSRow%(WkCF, "ENDLN", ActCol) + 2
59     If CellValue(WkCF, BSAudLN&, BFCol - 1) <> 0 Then
60         CFVeriByWS
61     End If
62
63     Application.Calculation = xlAutomatic
64
65     BSPLShowCurrent ("CF")
66     WkCF.Protect ("pwd")
67 End Sub
68 Sub CFPostXLn(ByVal StrtLn%)
69     Dim mSigma%, MBKCRow&, MBKCS$, tBFVal&, tCFVal&, vEndRow&
70     Dim sAName$, sEndCD$, WkAcct$, JVCalc$

```

```

dim BFamt&, CurrAmt&, absCurrAmt&, ArrBP
dim yCursor as object
dim bQuitKey as boolean
BPROw& = StrtLN%
Do While True
    bQuitKey = False
    SelectCellFS WkCF, BPROw&, BKCol
    WkAcct$ = CellString(WkCF, BPROw&, 2)
    sName$ = CellString(WkCF, BPROw&+1, CFCol-1)
    sEndCD$ = Ucase(Trim(WkAcct$))
    Rem -----(The order of If statement is optimized)-----
    If sEndCD$ = "ENDLN" Then
        If sName$ <> "BankBalance per G/L" then
            SetCellString WkCF, BPROw&+1, CFCol-1, "BankBalance per G/L:"
        Else
            vEndRow& = BPROw&
            BPROw&=CellValue(WkCL, 1, 28)
            tBFVal&=0
            tCFVal&=0
            Do While Len(CellString(WkCL, BPROw&, 28)) > 0
                MBKC$=CellString(WkCL, BPROw&, 28)
                MBKCRow& = getFSRow%(WkBS, MBKC$, ActCol)
                If CellValue(WkBS, MBKCRow&, ActCol)<>OWNCD Then
                    tBFVal&=tBFVal&+CellValue(WkBS, MBKCRow&, CBS_CM%-1)
                    tCFVal&=tCFVal&+CellValue(WkBS, MBKCRow&, CBS_CM%)
                End If
                BPROw&=BPROw& + 1
            Loop
            SetCellValue WkCF, vEndRow&+1, BFCol, tBFVal&
            SetCellValue WkCF, vEndRow&+1, CFCol, tCFVal&
        End If
        Exit Sub
    ElseIf sName$ = "BankBalance per G/L:" Then
        If sEndCD$ <> "ENDLN" then
            SetCellString WkCF, BPROw&, ActCol, "ENDLN"
        End If
        Exit Sub
    ElseIf BPROw& > 300 Then
        Do While Cellstring(WkCF, BPROw&, CFCol-1) = ""
            BPROw& = BPROw& - 1
        Loop
        BPROw& = BPROw& - 1
        SetCellString WkCF, BPROw&-1, ActCol, "ENDLN"
        SetCellString WkCF, BPROw&, CFCol-1, "BankBalance per G/L:"
        SelectCellFS WkCF, BPROw&, BKCol
        Exit Sub
    ElseIf myIsEmpty(WkCF, BPROw&, 2) Then
        bQuitKey = True
    Else
        If Val(WkAcct$) = 0 then
            bQuitKey = True
        end if
    End If
    If bQuitKey = False then
        Rem -----(Get Brought forward amount)-----
        BFamt& = RangeSum(WkCF, BFCol+8, BPROw&, BFCol+7+NumMo%, BPROw&)
        If CBS_CM% = CBS_YE% +1 Or CBS_CM% = CBS_YB% Then
            BFamt& = 0
        Else
            If CBS_CM% <= CBS_YE% Then
                BFamt& = RangeSum(WkCF, CBS_YB%, BPROw&, CBS_CM%-1, BPROw&)
            Else
                BFamt& = RangeSum(WkCF, CBS_YE%+1, BPROw&, CBS_CM%-1, BPROw&)
            End If
        End If
        Rem ----- BK summation -----
        AmtArray&(0, 0) = -CFAcctSum(WkBK, WkAcct$, oBKSDesc, BKRange)
        Rem ----- JV summation -----
        Rem Use the "Y" mark in column "CF" as flag for inclusion in CF preparation (2011/12/15)
        AmtArray&(0, 1) = -CFAcctSum(WkJV, WkAcct$, oJVSDesc, JVRRange)
    End If

```

```

71 Dim BFamt&, CurrAmt&, absCurrAmt&, ArrBP
72
73 Dim bQuitKey As Boolean
74 BPROw& = StrtLN%
75 Do While True
76     bQuitKey = False
77     SelectCellFS WkCF, BPROw&, BKCol
78     WkAcct$ = CellString(WkCF, BPROw&, 2)
79     sName$ = CellString(WkCF, BPROw& + 1, CFCol - 1)
80     sEndCD$ = Ucase(Trim(WkAcct$))
81     Rem -----(The order of If statement is optimized)-----
82     If sEndCD$ = "ENDLN" Then
83         If sName$ <> "BankBalance per G/L" Then
84             SetCellString WkCF, BPROw& + 1, CFCol - 1, "BankBalance per G/L:"
85         Else
86             vEndRow& = BPROw&
87             BPROw& = CellValue(WkCL, 1, 28)
88             tBFVal& = 0
89             tCFVal& = 0
90             Do While Len(CellString(WkCL, BPROw&, 28)) > 0
91                 MBKC$ = CellString(WkCL, BPROw&, 28)
92                 MBKCRow& = getFSRow%(WkBS, MBKC$, ActCol)
93                 If CellValue(WkBS, MBKCRow&, ActCol) <> OWNCD Then
94                     tBFVal& = tBFVal& + CellValue(WkBS, MBKCRow&, CBS_CM% - 1)
95                     tCFVal& = tCFVal& + CellValue(WkBS, MBKCRow&, CBS_CM%)
96                 End If
97                 BPROw& = BPROw& + 1
98             Loop
99             SetCellValue WkCF, vEndRow& + 1, BFCol, tBFVal&
100             SetCellValue WkCF, vEndRow& + 1, CFCol, tCFVal&
101         End If
102         Exit Sub
103     ElseIf sName$ = "BankBalance per G/L:" Then
104         If sEndCD$ <> "ENDLN" Then
105             SetCellString WkCF, BPROw&, ActCol, "ENDLN"
106         End If
107         Exit Sub
108     ElseIf BPROw& > 300 Then
109         Do While CellString(WkCF, BPROw&, CFCol - 1) = ""
110             BPROw& = BPROw& - 1
111         Loop
112         BPROw& = BPROw& - 1
113         SetCellString WkCF, BPROw& - 1, ActCol, "ENDLN"
114         SetCellString WkCF, BPROw&, CFCol - 1, "BankBalance per G/L:"
115         SelectCellFS WkCF, BPROw&, BKCol
116         Exit Sub
117     ElseIf myIsEmpty(WkCF, BPROw&, 2) Then
118         bQuitKey = True
119     Else
120         If Val(WkAcct$) = 0 Then
121             bQuitKey = True
122         End If
123     End If
124     If bQuitKey = False Then
125         Rem -----(Get Brought forward amount)-----
126         BFamt& = RangeSum(WkCF, BFCol + 8, BPROw&, BFCol + 7 + NumMo%, BPROw&)
127         If CBS_CM% = CBS_YE% + 1 Or CBS_CM% = CBS_YB% Then
128             BFamt& = 0
129         Else
130             If CBS_CM% <= CBS_YE% Then
131                 BFamt& = RangeSum(WkCF, CBS_YB%, BPROw&, CBS_CM% - 1, BPROw&)
132             Else
133                 BFamt& = RangeSum(WkCF, CBS_YE% + 1, BPROw&, CBS_CM% - 1, BPROw&)
134             End If
135         End If
136         Rem ----- BK summation -----
137         AmtArray&(0, 0) = -CFAcctSum(WkBK, WkAcct$, BKRange)
138         Rem ----- JV summation -----
139         Rem Use the "Y" mark in column "CF" as flag for inclusion in CF preparation (2011/12/15)
140         AmtArray&(0, 1) = -CFAcctSum(WkJV, WkAcct$, JVRRange)

```

```

Rem ----- AP summation -----
if APUse Then
    AmtArray&(0, 2) = -CFAcctSum(WkAP, WkAcct$, oAPSDesc, APRange)
end if
Rem ----- AR summation -----
if ARUse Then
    If CStr(WkAcct$) = CStr(ARCD) Then
        mSigma% = 1
    Else
        mSigma% = -1
    End If
    AmtArray&(0, 3) = -CFAcctSum(WkAR, WkAcct$, oARSDesc, ARRange) * mSigma%
end if
Rem ----- MW summation -----
if MWUse Then
    AmtArray&(0, 4) = CFAcctSum(WkMW, WkAcct$, oMWSDesc, MWRRange)
end if
Rem ----- Posting to CF -----
Set yCursor = WkCF.getCellRangeByPosition(BFCol-1, BPRow&-1, RMCol-1, BPRow&-1)

Rem ----- preparation for setDataArray() -----
if ARUse Then
    ArrBP=Array(BFAmt&, AmtArray&(0, 0), AmtArray&(0, 1), AmtArray&(0, 2), AmtArray&(0, 3))
    CurrAmt& = AmtArray&(0, 0) + AmtArray&(0, 1) + AmtArray&(0, 2) + AmtArray&(0, 3)
elseif MWUse Then
    ArrBP=Array(BFAmt&, AmtArray&(0, 0), AmtArray&(0, 1), AmtArray&(0, 2), AmtArray&(0, 4))
    CurrAmt& = AmtArray&(0, 0) + AmtArray&(0, 1) + AmtArray&(0, 2) + AmtArray&(0, 4)
end if
yCursor.setDataArray(Array(ArrBP)) '-(Posting B/F and Current only on BS.:PostBFCF will post to the Trend
Rem ----- Posting to CF trends -----
SetCellValue WkCF, BPRow&, CBS_CM%, CurrAmt&
absCurrAmt& = ABS(AmtArray&(0, 0)) + ABS(AmtArray&(0, 1)) + ABS(AmtArray&(0, 2)) + ABS(AmtArray&(0, 3))
if ABS(BFAmt&) + absCurrAmt& = 0 then
    if WkCF.getCellByPosition(0, BPRow& - 1).Rows.IsVisible = True then
        WkCF.getCellByPosition(0, BPRow& - 1).Rows.IsVisible = False
    End If
else
    if WkCF.getCellByPosition(0, BPRow& - 1).Rows.IsVisible = False then
        WkCF.getCellByPosition(0, BPRow& - 1).Rows.IsVisible = True
    End If
end if
End If
BPRow& = BPRow& + 1
Loop
End Sub
Sub CFPostXFm()
    Rem ---- Current month summation of W/H Tax and Bank Charges ----
    Dim WHTaxRow&, BKChgRow&, CurrAmt&
    Dim BKChgAmt&, tVal&, WHTax&
    getColArray ("BK") 'get from ColArray at once
    targetRow& = SeekTBRow&(WkBK, "ValMo", cMonth%, cMonth%, "Top")
    Rem ----- Calculation of Bank Charges -----
    BKChgRow& = getFSRow&(WkCF, BKChgCD, ActCol)
    tVal& = Val(CellValue(WkCF, BKChgRow&, BKCol))
    BKChgAmt& = -RangeSum&(WkBK, Col_BKChg%, dbMonthTop&, Col_BKChg%, dbMonthBtm&)
    SetCellValue WkCF, BKChgRow&, BKCol, tVal& + BKChgAmt&
    CurrAmt& = RangeSum(WkCF, BKCol, BKChgRow&, RMCol, BKChgRow&)
    SetCellValue WkCF, BKChgRow&, CBS_CM%, CurrAmt&
    Rem ----- Calculation of WithHolding Taxes -----
    WHTaxRow& = getFSRow&(WkCF, WHTaxCD, ActCol)
    tVal& = Val(CellValue(WkCF, WHTaxRow&, BKCol))
    WHTax& = -RangeSum&(WkBK, Col_WHT%, dbMonthTop&, Col_WHT%, dbMonthBtm&)
    SetCellValue WkCF, WHTaxRow&, BKCol, tVal& + WHTax&
    CurrAmt& = RangeSum(WkCF, BKCol, WHTaxRow&, RMCol, WHTaxRow&)
    SetCellValue WkCF, WHTaxRow&, CBS_CM%, CurrAmt&
End Sub
Function CFAcctSum&(ByVal WkWk as object, ByVal FkAcct$, ByVal oSearchDesc, oSRange as object)
    dim i%, j%, xRow&, fName$

```

```

141 Rem ----- AP summation -----
142 if APUse Then
143     AmtArray&(0, 2) = -CFAcctSum(WkAP, WkAcct$, APRange)
144 End if
145 Rem ----- AR summation -----
146 if ARUse Then
147     if CStr(WkAcct$) = CStr(ARCD) Then
148         mSigma% = 1
149     Else
150         mSigma% = -1
151     End if
152     AmtArray&(0, 3) = -CFAcctSum(WkAR, WkAcct$, ARRange) * mSigma%
153 End if
154 Rem ----- MW summation -----
155 if MWUse Then
156     AmtArray&(0, 4) = CFAcctSum(WkMW, WkAcct$, MWRRange)
157 End if
158 Rem ----- Posting to CF -----
159 SetCellValue WkCF, BPRow&, BFCol, BFAmt&
160 SetCellValue WkCF, BPRow&, BKCol, AmtArray&(0, 0)
161 SetCellValue WkCF, BPRow&, JVCOL, AmtArray&(0, 1)
162 SetCellValue WkCF, BPRow&, APCol, AmtArray&(0, 2)
163 if ARUse Then
164     SetCellValue WkCF, BPRow&, ARCol, AmtArray&(0, 3)
165     CurrAmt& = AmtArray&(0, 0) + AmtArray&(0, 1) + AmtArray&(0, 2) + AmtArray&(0, 3)
166     absCurrAmt& = Abs(AmtArray&(0, 0)) + Abs(AmtArray&(0, 1)) + Abs(AmtArray&(0, 2))
167 Elseif MWUse Then
168     SetCellValue WkCF, BPRow&, MWCol, AmtArray&(0, 4)
169     CurrAmt& = AmtArray&(0, 0) + AmtArray&(0, 1) + AmtArray&(0, 2) + AmtArray&(0, 4)
170     absCurrAmt& = Abs(AmtArray&(0, 0)) + Abs(AmtArray&(0, 1)) + Abs(AmtArray&(0, 2))
171 End if
172 Rem ----- Posting to CF Trend -----
173 SetCellValue WkCF, BPRow&, CBS_CM%, CurrAmt&
174 Rem ----- Hide zero rows on CF -----
175 if Abs(BFAmt&) + absCurrAmt& = 0 Then
176     if WkCF.Rows(BPRow&).Hidden = False Then
177         WkCF.Rows(BPRow&).Hidden = True
178     End if
179 Else
180     if WkCF.Rows(BPRow&).Hidden = True Then
181         WkCF.Rows(BPRow&).Hidden = False
182     End if
183 End if
184 End if
185 BPRow& = BPRow& + 1
186 Loop
187 End Sub
188 Sub CFPostXFm()
189     Rem ---- Current month summation of W/H Tax and Bank Charges ----
190     Dim WHTaxRow&, BKChgRow&, CurrAmt&
191     Dim BKChgAmt&, tVal&, WHTax&
192     getColArray ("BK") 'get from ColArray at once
193     targetRow& = SeekTBRow&(WkBK, "ValMo", cMonth%, cMonth%, "Top")
194     Rem ----- Calculation of Bank Charges -----
195     BKChgRow& = getFSRow&(WkCF, BKChgCD, ActCol)
196     tVal& = Val(CellValue(WkCF, BKChgRow&, BKCol))
197     BKChgAmt& = -RangeSum&(WkBK, Col_BKChg%, dbMonthTop&, Col_BKChg%, dbMonthBtm&)
198     SetCellValue WkCF, BKChgRow&, BKCol, tVal& + BKChgAmt&
199     CurrAmt& = RangeSum(WkCF, BKCol, BKChgRow&, RMCOL, BKChgRow&)
200     SetCellValue WkCF, BKChgRow&, CBS_CM%, CurrAmt&
201     Rem ----- Calculation of WithHolding Taxes -----
202     WHTaxRow& = getFSRow&(WkCF, WHTaxCD, ActCol)
203     tVal& = Val(CellValue(WkCF, WHTaxRow&, BKCol))
204     WHTax& = -RangeSum&(WkBK, Col_WHT%, dbMonthTop&, Col_WHT%, dbMonthBtm&)
205     SetCellValue WkCF, WHTaxRow&, BKCol, tVal& + WHTax&
206     CurrAmt& = RangeSum(WkCF, BKCol, WHTaxRow&, RMCOL, WHTaxRow&)
207     SetCellValue WkCF, WHTaxRow&, CBS_CM%, CurrAmt&
208 End Sub
209 Function CFAcctSum&(ByVal WkWk As Object, ByVal FkAcct$, ByVal oSRange As Object)
210     Dim i%, j%, xRow&, fName$, firstAddress

```

```

dim oFound as object, oCell As Object, oLength%
dim fCol_Month%, fCol_CTax%, fCol_ActCD%, fCol_Amount%, fCol_LineNo%
dim ICTax%, IAmount%, LnMonth%, LnAcct$, sPM$
fSName$ = WkWk.getName() 'define fSName$ here
oSearchDesc.SearchWords = true
oSearchDesc.SearchString = FkAcct
oSearchDesc.SearchBackwards = False
Set oFound = oSRange.findAll(oSearchDesc)
If IsNull(oFound) then
    Exit Function
EndIf
sPM$ = CellString(WkCF, BPRow&, 3)
CFAcctSum& = 0
Select Case fSName$
    Case "BK", "JV"
        fCol_CTax%=Col_InArray(fSName$, "CXDr")
        fCol_Month%=Col_InArray(fSName$, "ValMo")
        fCol_LineNo%=Col_InArray(fSName$, "LineNo")
    Case "AP"
        fCol_CTax%=Col_InArray(fSName$, "CXDr")
        fCol_Month%=Col_InArray(fSName$, "PaidMo")
    Case "AR", "MW"
        fCol_CTax%=Col_InArray(fSName$, "CXCcr")
        fCol_Month%=Col_InArray(fSName$, "PaidMo")
End Select
fCol_ActCD%=Col_InArray(fSName$, "ActCD")
fCol_Amount%=Col_InArray(fSName$, "Amount")
oLength% = oFound.getCount()

For i% = 0 to oLength%-1
    xRow& = oFound.getByIndex(i%).RangeAddress.StartRow + 1
    for j% = 1 to oFound.getByIndex(i%).Rows.Count
        LnMonth% = Val(CellString(WkWk, xRow&, xRow&, fCol_Month%))
        If fSName$="JV" Then
            Rem Use the "Y" mark in column "CF" as flag for inclusion in CF preparation (2011/12/15)
            If CellString(WkWk, xRow&, fCol_LineNo%+1) <>"Y" Then
                xRow& = xRow& + 1
                Exit For
            End If
        End If
        IAmount& = CellValue(WkWk, xRow&, fCol_Amount%)
        ICTax& = CellValue(WkWk, xRow&, fCol_CTax%)
        If A_VBSCD(FkAcct) then
            If sPM$="Dr" And IAmount& <= 0 then
                IAmount& = 0
                ICTax& = 0
            Elseif sPM$="Cr" And IAmount& > 0 then
                IAmount& = 0
                ICTax& = 0
            End If
        end if
        CFAcctSum& = CFAcctSum& + ICTax& + IAmount&
    End If
    xRow& = xRow& + 1
End Function
Sub CFPostBKBal()
    Dim sWD$, BKTtILN%, BSAudLN%
    sWD$= "Cash Total"
    BKTtILN% = getFSRow%(WkBS, sWD$, CFCol-1)
    BSAudLN% = getFSRow%(WkCF, "ENDLN", ActCol) + 1
    SetCellValue WkCF, BSAudLN%, BFCol, CellValue(WkBS, BKTtILN%, CBS_CM%-1)
    SetCellValue WkCF, BSAudLN%, CFCol, CellValue(WkBS, BKTtILN%, CBS_CM%)
End Sub
Sub CFfindPost(ByVal selActCD$, ByVal iStrtMo%, ByVal iEndMo%)
    dim i%, mFormula$
    ClearRange WkLN, 3, 1, 3000, 10 + NumDpt
    WkLN.getCellRangeByPosition(6, 0, 6, 0).Columns.IsVisible = False
    LnRow& = 3
    Rem ----- Define Data Range Subject to Account Analysis ----

```

```

211 Dim oFound As Object, oCell As Object, oLength%
212 Dim fCol_Month%, fCol_CTax%, fCol_ActCD%, fCol_Amount%, fCol_LineNo%
213 Dim ICTax%, IAmount%, LnMonth%, LnAcct$, sPM$
214 fSName$ = WkWk.Name 'define fSName$ here
215 Set oFound = oSRange.Find(FkAcct$, Lookat:=xlWhole, SearchDirection:=xlNext)
216
217
218
219 If oFound Is Nothing Then
220     Exit Function
221 End If
222 sPM$ = CellString(WkCF, BPRow&, 3)
223 CFAcctSum& = 0
224 Select Case fSName$
225     Case "BK", "JV"
226         fCol_CTax% = Col_InArray(fSName$, "CXDr")
227         fCol_Month% = Col_InArray(fSName$, "ValMo")
228         fCol_LineNo% = Col_InArray(fSName$, "LineNo")
229     Case "AP"
230         fCol_CTax% = Col_InArray(fSName$, "CXDr")
231         fCol_Month% = Col_InArray(fSName$, "PaidMo")
232     Case "AR", "MW"
233         fCol_CTax% = Col_InArray(fSName$, "CXCcr")
234         fCol_Month% = Col_InArray(fSName$, "PaidMo")
235 End Select
236 fCol_ActCD% = Col_InArray(fSName$, "ActCD")
237 fCol_Amount% = Col_InArray(fSName$, "Amount")
238 firstAddress = oFound.Address
239 Do
240     xRow& = oFound.Row
241     LnMonth% = Val(CellString(WkWk, xRow&, fCol_Month%))
242     LnAcct$ = CellString(WkWk, xRow&, fCol_ActCD%)
243     If LnMonth% = cMonth% And LnAcct$ = FkAcct Then
244         IAmount& = CellValue(WkWk, xRow&, fCol_Amount%)
245         ICTax& = CellValue(WkWk, xRow&, fCol_CTax%)
246         If A_VBSCD(FkAcct) Then
247             If sPM$ = "Dr" And IAmount& <= 0 Then
248                 IAmount& = 0
249                 ICTax& = 0
250             Elseif sPM$ = "Cr" And IAmount& > 0 Then
251                 IAmount& = 0
252                 ICTax& = 0
253             End If
254         End If
255         If fSName$ = "JV" Then
256             Rem Use the "Y" mark in column "CF" as flag for inclusion in CF preparation (2011/12
257             If CellString(WkWk, xRow&, fCol_LineNo% + 1) <>"Y" Then
258                 IAmount& = 0
259                 ICTax& = 0
260             End If
261         End If
262         CFAcctSum& = CFAcctSum& + ICTax& + IAmount&
263     End If
264     Set oFound = oSRange.FindNext(oFound)
265 Loop While Not oFound Is Nothing And oFound.Address <> firstAddress
266 End Function
267 Sub CFPostBKBal()
268     Dim sWD$, BKTtILN%, BSAudLN%
269     sWD$ = "Cash Total"
270     BKTtILN% = getFSRow%(WkBS, sWD$, CFCol - 1)
271     BSAudLN% = getFSRow%(WkCF, "ENDLN", ActCol) + 1
272     SetCellValue WkCF, BSAudLN%, BFCol, CellValue(WkBS, BKTtILN%, CBS_CM% - 1)
273     SetCellValue WkCF, BSAudLN%, CFCol, CellValue(WkBS, BKTtILN%, CBS_CM%)
274 End Sub
275 Sub CFfindPost(ByVal selActCD$, ByVal iStrtMo%, ByVal iEndMo%)
276     Dim i%, mFormula$
277     ClearRange WkLN, 3, 1, 3000, 10 + NumDpt
278     WkLN.Columns(7).Hidden = True
279     LnRow& = 3
280     Rem ----- Define Data Range Subject to Account Analysis ----

```

```

SetSrcRange "PaidMo", iStrtMo%, iEndMo%, "ActCD"
Rem ----- From BK -----
PstOTAMts(WkBK, iStrtMo%, iEndMo%, selActCD$, oBKSDesc, BKRRange, "CF")
Rem ----- From JV -----
PstOTAMts(WkJV, iStrtMo%, iEndMo%, selActCD$, oJVSDesc, JVRRange, "CF")
Rem ----- From AP -----
If APUse then
    PstOTAMts(WkAP, iStrtMo%, iEndMo%, selActCD$, oAPSDesc, APRRange, "CF")
End If
Rem ----- From AR -----
If ARUse then
    PstOTAMts(WkAR, iStrtMo%, iEndMo%, selActCD$, oARSDesc, ARRange, "CF")
End If
Rem ----- From MW -----
If MWUse then
    PstOTAMts(WkMW, iStrtMo%, iEndMo%, selActCD$, oMWSDesc, MWRRange, "CF")
End If
Rem -----(Steps for Withholding Tax)-----
If selActCD$ = WHTaxCD then
    WkLn.getCellRangeByPosition(6, 0, 6, 0).Columns.IsVisible = True
    SetSrcRange "ValMo", iStrtMo%, iEndMo%, "WHT" "BK only
    PstWHTax iStrtMo%, iEndMo%, "CF"
End If
If selActCD$ = BKChgCD Then
    PostBKChrg iStrtMo%, iEndMo%, "CF"
End If
SelectCell LnRow&, 6
SetCellString WkLn, LnRow&, 5, "Total"
If LnRow& > 3 Then
    For i = 6 To 10 + NumDpt
        mFormula$ = "=SUM(" & CAlpha(i) & ":3:" & CAlpha(i) & Cstr(LnRow& - 1) & ")")
        SetCellFormula WkLn, LnRow&, i, mFormula$
    Next i
End If
End Sub
Rem To find out account code which is not used on CF sheet
Sub CFVeriByWS()
    Dim oRange As Range, MsgAdd$
    dim MsgAdd$
    IntroTo3D()
    MakeAcctList(WkCF, )
    Rem ----- Preparation of message -----
    MsgAdd$ = Chr(13) & "Continue checking ?"
    MsgTxt$ = "This account code is not used on CF sheet."
    MsgTxt$ = MsgTxt$ & MsgAdd$
    MsgTit$ = "Verification of Account code"
    Rem -----
    setOptState "BK"
    IsExistInCF(WkBK, dbMonthTop&+1)
    setOptState "JV"
    IsExistInCF(WkJV, dbMonthTop&+1)
    getColArray "AP"
    setRowArray "AP", "PaidMo", cMonth%, cMonth%
    IsExistInCF(WkAP, dbMonthTop&+1)
    if ARUse Then
        getColArray "AR"
        setRowArray "AR", "PaidMo", cMonth%, cMonth%
        IsExistInCF(WkAR, dbMonthTop&+1)
    end if
    if MWUse Then
        getColArray "MW"
        setRowArray "MW", "PaidMo", cMonth%, cMonth%
        IsExistInCF(WkMW, dbMonthTop&+1)
    end if
    oAWDIg.endExecute()
End Sub
Sub IsExistInCF(ByVal WkWk As Object, ByVal StrtRow&)
    dim oRange as object
    dim VeriLN&, WKAcct0$, WkAcct$, AcctIndex%
    aController.ActiveSheet=WkWk

```

```

281 SetSrcRange "PaidMo", iStrtMo%, iEndMo%, "ActCD"
282 Rem ----- From BK -----
283 PstOTAMts WkBK, iStrtMo%, iEndMo%, selActCD$, "ActCD", BKRRange, "CF"
284 Rem ----- From JV -----
285 PstOTAMts WkJV, iStrtMo%, iEndMo%, selActCD$, "ActCD", JVRRange, "CF"
286 Rem ----- From AP -----
287 If APUse Then
288     PstOTAMts WkAP, iStrtMo%, iEndMo%, selActCD$, "ActCD", APRRange, "CF"
289 End If
290 Rem ----- From AR -----
291 If ARUse Then
292     PstOTAMts WkAR, iStrtMo%, iEndMo%, selActCD$, "ActCD", ARRange, "CF"
293 End If
294 Rem ----- From MW -----
295 If MWUse Then
296     PstOTAMts WkMW, iStrtMo%, iEndMo%, selActCD$, "ActCD", MWRRange, "CF"
297 End If
298 Rem -----(Steps for Withholding Tax)-----
299 If selActCD$ = WHTaxCD Then
300     WkLn.Columns(7).Hidden = False
301     SetSrcRange "ValMo", iStrtMo%, iEndMo%, "WHT" "BK only
302     PstWHTax iStrtMo%, iEndMo%, "CF"
303 End If
304 If selActCD$ = BKChgCD Then
305     PostBKChrg iStrtMo%, iEndMo%, "CF"
306 End If
307 SelectCell LnRow&, 6
308 SetCellString WkLn, LnRow&, 5, "Total"
309 If LnRow& > 3 Then
310     For i = 6 To 10 + NumDpt
311         mFormula$ = "=SUM(R[-" & CStr(LnRow& - 3) & "]C:R[-1]C)"
312         SetCellFormula WkLn, LnRow&, i, mFormula$
313     Next i
314 End If
315 End Sub
316 Rem To detect account code not used on CF sheet
317 Sub CFVeriByWS()
318     Dim oRange As Range, MsgAdd$
319
320     IntroTo3D
321     MakeAcctList WkCF
322     Rem ----- Preparation of message -----
323     MsgAdd$ = Chr(13) & "Continue checking ?"
324     MsgTxt$ = "This account code is not used on CF sheet."
325     MsgTxt$ = MsgTxt$ & MsgAdd$
326     MsgTit$ = "Verification of Account code"
327     Rem -----
328     setOptState ("BK")
329     IsExistInCF WkBK, dbMonthTop&
330     setOptState ("JV")
331     IsExistInCF WkJV, dbMonthTop&
332     getColArray ("AP")
333     setRowArray "AP", "PaidMo", cMonth%, cMonth%
334     IsExistInCF WkAP, dbMonthTop& + 1
335     If ARUse Then
336         getColArray "AR"
337         setRowArray "AR", "PaidMo", cMonth%, cMonth%
338         IsExistInCF WkAR, dbMonthTop& + 1
339     End If
340     If MWUse Then
341         getColArray "MW"
342         setRowArray "MW", "PaidMo", cMonth%, cMonth%
343         IsExistInCF WkMW, dbMonthTop& + 1
344     End If
345
346 End Sub
347 Sub IsExistInCF(ByVal WkWk As Object, ByVal StrtRow&)
348     Dim oRange As Range
349     Dim VeriLN&, WKAcct0$, WkAcct$, AcctIndex%
350     WkWk.Activate

```

```

VeriLN& = StrtRow&
Do while VeriLN& <= dbMonthBtm& + 1
    If WkWk.Name="JV" Then
        If CellValue(WkWk, VeriLN&, Col_LineNo%+1) <> "Y" Then
            Exit Do
        End If
    End If
    SelectCell VeriLN&, Col_ActCD%
    WkAcct0$ = Trim(CellString(WkWk, VeriLN&, Col_ActCD%))
    WkAcct$ = FillSpace(Len(WkAcct0$)) & WkAcct0$
    AcctIndex% = IndexInACT(WkAcct$)
    Set oRange = WkWk.getCellByPosition(Col_ActCD%-1, VeriLN&-1)
    If AcctIndex% > -1 then 'where WkAcct is used on CF
        if Hex(oRange.CellBackColor) <> "FFFFFF" then
            oRange.CellBackColor = "&HFFFFFF"
        end if
    Else
        if VeriLN& > 2 AND Len(WkAcct0$) > 0 Then
            If Not IsExclBC(WkAcct0$) Then
                oRange.CellBackColor = RGB(221,160,221)
                resp%=MsgBox(MsgTxt$, 276, MsgTtl$)
                If resp%<>6 Then
                    End
                End If
            End If
        end if
    End If
    VeriLN& = VeriLN& + 1
    If WkWk.Name()= "AP" or WkWk.Name() = "AR" or WkWk.Name() = "MW" Then
        Do While WkWk.getCellByPosition(1, VeriLN&-1).Rows.IsVisible = False And VeriLN& < dbMonthBtm&
            VeriLN& = VeriLN& + 1
        Loop
    End If
Loop
End Sub
Function IsExclBC(fAcct$) As Boolean
    dim bkRow&
    IsExclBC=False
    bkRow& = CellValue(WkCL,1,28)
    Do While Not myIsEmpty(WkCL, bkRow&, 28)
        If fAcct$=CellString(WkCL,bkRow&,28) Then
            IsExclBC=True
            Exit Do
        End If
        bkRow&=bkRow&+1
    Loop
    Select Case fAcct$
        Case ARCD, APCD, CRTXCD, PYTXCD
            IsExclBC=True
    End Select
End Function

```

```

351 VeriLN& = StrtRow&
352 Do While VeriLN& < dbMonthBtm& + 1
353     If WkWk.Name = "JV" Then
354         If CellValue(WkWk, VeriLN&, Col_LineNo% + 1) <> "Y" Then
355             Exit Do
356         End If
357     End If
358     SelectCell VeriLN&, Col_ActCD%
359     WKAcct0$ = Trim(CellString(WkWk, VeriLN&, Col_ActCD%))
360     WkAcct$ = FillSpace(Len(WKAcct0$)) & WKAcct0$
361     AcctIndex% = IndexInACT(WkAcct$)
362     Set oRange = WkWk.Cells(VeriLN&, Col_ActCD%)
363     If AcctIndex% > -1 Then 'where WkAcct is used on CF
364         If oRange.Interior.Color <> RGB(255, 255, 255) Then
365             oRange.Interior.Color = RGB(255, 255, 255)
366         End If
367     Else
368         If VeriLN& > 2 And Len(WKAcct0$) > 0 Then
369             If Not IsExclBC(WKAcct0$) Then
370                 oRange.Interior.Color = RGB(221, 160, 221)
371                 resp% = MsgBox(MsgTxt$, 276, MsgTtl$)
372                 If resp% <> 6 Then
373                     End
374                 End If
375             End If
376         End If
377     End If
378     VeriLN& = VeriLN& + 1
379     If WkWk.Name = "AP" Or WkWk.Name = "AR" Or WkWk.Name = "MW" Then
380         Do While WkWk.Rows(VeriLN&).Hidden = True And VeriLN& < dbMonthBtm&
381             VeriLN& = VeriLN& + 1
382         Loop
383     End If
384 Loop
385 End Sub
386 Function IsExclBC(fAcct$) As Boolean
387     Dim bkRow&
388     IsExclBC = False
389     bkRow& = CellValue(WkCL, 1, 28)
390     Do While Not myIsEmpty(WkCL, bkRow&, 28)
391         If fAcct$ = CellString(WkCL, bkRow&, 28) Then
392             IsExclBC = True
393             Exit Do
394         End If
395         bkRow& = bkRow& + 1
396     Loop
397     Select Case fAcct$
398         Case ARCD, APCD, CRTXCD, PYTXCD
399             IsExclBC = True
400     End Select
401 End Function

```