

```

Rem =====
Rem   A_Reporting_BSPL preparation OoBasic Program (Copyright:Shuichi Sunaga)
Rem   (2005/10/31:493)Adopted FindALL in AcctSum()(2006/05/14:549)
Rem   Segregated prior and post WsCalc procedures(2006/11/17:603)
Rem   Introduced PostBKChrg()(2007/12/10:783)
Rem   Inroduced RangeSum for posting to trend columns in PostXLn(): (2008/12/5:849)
Rem   Concentration of Post2Owner and IndvBF into ND(2009/09/22:770)(2012/07/06:616)
Rem   Translation into English(2012/08/12:567)
Rem =====
Rem Sub WsCalc()
Rem Sub ChkB4WsCalc()
Rem Sub PostXLn(ByVal WkNow As Object, ByVal adjCIAct%, ByVal StrtLN%)
Rem Sub PostXFm()
Rem Sub PostBKChrg()
Rem Sub PostBFCF(ByVal WkNow As Object, ByVal StrtLN%)
Rem Function AcctSum&(ByVal WkWk as object, FkAcct$, fCol_Acct%, oSearchDesc, oSRange as object)
Rem Sub ChkAftWSCalc()
Rem Sub VeriDept()
Rem =====
Option Explicit
Sub WsCalc()
    dim args(0) as new com.sun.star.beans.PropertyValue
    dim i%, mTopRow&
    oCurrYMD.setFocus()
    IntroTo3D()
    Rem (1)--- Quit WsCalc where current month transaction is non-existent -----
    If CellValue(WkBS, getFSRow%(WkBS, "ENDLN", ActCol) + 1, CBS_CM% - 1) <> 0 Then
        MsgBox("InCompletePriorFS")
    End
End If
setCol4All()
If ExamAllTopRow(cMonth%, "Top") then
    MsgTxt$ = "Will abort BS/PL Preparation because no transaction for the" & chr(13)
    MsgTxt$ = MsgTxt$ & "current month (" & CStr(cMonth%) & ") is found."
    MsgTit$ = "Test of input transaction data"
    msgbox(MsgTxt$, 0, MsgTit$)
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)-----Define Data Range subject to BS/PL preparation -----
SetSrcRange "ValMo", cMonth%, cMonth%, "ActCD"
Rem (4)----- Verify Dr/Cr Balance for JV,AP,AR,MW -----
ChkB4WsCalc()
Rem (5-1)----- Make sum of posting source columns on the BS ----
aController.ActiveSheet = WkBS
WkBS.Unprotect("pwd")
PostXLn WkBS, 6
Rem (5-2)----- Make sum of posting source columns on the PL ----
aController.ActiveSheet = WkPL
WkPL.Unprotect("pwd")
If IsMultiDP then
    For i% = 1 To NumDpt
        WkDpt(i).Unprotect("pwd")
    Next i%
End If
PostXLn WkPL, 6
Rem (5-3)----- Make sum of transactions for bank accounts -----
aController.ActiveSheet = WkBS
PostXFm
Rem (5-4)----- Automatic JE for due from sole proprietorship-----
If CorpOrIndv = "Indv" then
    Post2Owner() 'see INDIV
End If
Rem (5-5)----- Posting to trend analysis range (after PostXFm) -----
PostBFCF WkBS, 6 '---Posting to BS trend
BSPLShowCurrent("BP")

```

```

1 Rem =====
2 Rem   A_Reporting_BSPL preparation OoBasic Program (Copyright:Shuichi Sunaga)
3 Rem
4 Rem
5 Rem
6 Rem
7 Rem   (2012/04/24:573行)
8 Rem   Translation into English(2012/08/12:567)
9 Rem =====
10 Rem Sub WsCalc()
11 Rem Sub ChkB4WsCalc()
12 Rem Sub PostXLn(ByVal WkNow As Object, ByVal adjCIAct%, ByVal StrtLN%)
13 Rem Sub PostXFm()
14 Rem Sub PostBKChrg()
15 Rem Sub PostBFCF(ByVal WkNow As Object, ByVal StrtLN%)
16 Rem Function AcctSum&(ByVal WkWk as object, FkAcct$, fCol_Acct%, oSearchDesc, oSRange
17 Rem Sub ChkAftWSCalc()
18 Rem Sub VeriDept()
19 Rem =====
20 Option Explicit
21 Sub WsCalc()
22
23     Dim i%, mTopRow&, a%
24     oCurrYMD.SetFocus
25     IntroTo3D
26     Rem (1)--- Quit WsCalc where current month transaction is non-existent -----
27     If CellValue(WkBS, getFSRow%(WkBS, "ENDLN", ActCol) + 1, CBS_CM% - 1) <> 0 Then
28         MsgBox ("InCompletePriorFS")
29     End
30 End If
31 setCol4All
32 If ExamAllTopRow(cMonth%, "Top") Then
33     MsgTxt$ = "Will abort BS/PL Preparation because no transaction for the" & Chr(13)
34     MsgTxt$ = MsgTxt$ & "current month (" & CStr(cMonth%) & ") is found."
35     MsgTit$ = "Test of input transaction data"
36     resp% = MsgBox(MsgTxt$, 0, MsgTit$)
37 End
38 End If
39 Rem (2)----- Change to manual calculation mode -----
40
41
42 Application.Calculation = xlManual
43 Calculate
44 Rem (3)-----Define Data Range subject to BS/PL preparation -----
45 SetSrcRange "ValMo", cMonth%, cMonth%, "ActCD"
46 Rem (4)----- Verify Dr/Cr Balance for JV,AP,AR,MW -----
47 ChkB4WsCalc
48 Rem (5-1)----- Make sum of posting source columns on the BS ----
49 WkBS.Activate
50 WkBS.Unprotect ("pwd")
51 PostXLn WkBS, 6
52 Rem (5-2)----- Make sum of posting source columns on the PL ----
53 WkPL.Activate
54 WkPL.Unprotect ("pwd")
55 If IsMultiDP Then
56     For i% = 1 To NumDpt
57         WkDpt(i).Unprotect ("pwd")
58     Next i%
59 End If
60 PostXLn WkPL, 6
61 Rem (5-3)----- Make sum of transactions for bank accounts -----
62 WkBS.Activate
63 PostXFm
64 Rem (5-4)----- Automatic JE for due from sole proprietorship-----
65 If CorpOrIndv = "Indv" Then
66     Post2Owner 'See INDIV
67 End If
68 Rem (5-5)----- Posting to trend analysis range (after PostXFm) -----
69 PostBFCF WkBS, 6 '---Posting to BS trend
70 BSPLShowCurrent ("BP")

```

```

Rem (6)----- Change to Automatic calculation mode -----
args(0).Value = true
dispatcher.executeDispatch(aController, ".uno:AutomaticCalculation", "", 0, args())
Rem ②----- Protect BS,PL, WkDpt(i) after FS Calculation -----
WkBS.Protect("pwd")
WkPL.Protect("pwd")
If IsMultiDP then
    For i% = 1 To NumDpt
        WkDpt(i).Protect("pwd")
    Next i%
End If
Rem ⑧----- Handling where B/S Audit line shows other than zero -----
ChkAftWSCalc()
Rem -----
If IsAWDlOpen() then
    oAWDlG.endExecute()
End If
GiveDTHeader()
ThisComponent.calculateAll()
End Sub
Rem ----- Verify Dr/Cr Balance for JV,AP,AR,MW -----
Sub ChkB4WsCalc()
    setCol4All()
    Rem -----
    If APUse Then
        getRowArray "AP", "ValMo", cMonth%, cMonth%
        Col_Gross% = Col_InArray("AP", "Gross")
        Col_Amount% = Col_InArray("AP", "Amount")
        If CellValue(WkAP, dbBtmBorder&, Col_Gross%) <> CellValue(WkAP, dbBtmBorder&, Col_Amount%) Then
            setOptState("AP")
            VeriByWS dbMonthTop&, 276
        End
    end if
End If
Rem -----
If ARUse Then
    getRowArray "AR", "ValMo", cMonth%, cMonth%
    Col_Gross% = Col_InArray("AR", "Gross")
    Col_Amount% = Col_InArray("AR", "Amount")
    If CellValue(WkAR, dbBtmBorder&, Col_Gross%) <> CellValue(WkAR, dbBtmBorder&, Col_Amount%) Then
        setOptState("AR")
        VeriByWS dbMonthTop&, 276
    End
end If
End If
Rem -----
If MWUse Then
    getRowArray "MW", "ValMo", cMonth%, cMonth%
    Col_Gross% = Col_InArray("MW", "Gross")
    Col_Amount% = Col_InArray("MW", "Amount")
    If CellValue(WkMW, dbBtmBorder&, Col_Gross%) <> CellValue(WkMW, dbBtmBorder&, Col_Amount%) Then
        setOptState("MW")
        VeriByWS dbMonthTop&, 276
    End
end if
End If
Rem -----
GetRowArray "JV", "ValMo", cMonth%, cMonth%
Col_Gross% = Col_InArray("JV", "Gross")
Col_Amount% = Col_InArray("JV", "Amount")
If CellValue(WkJV, dbBtmBorder&, Col_Gross%) <> 0 or CellValue(WkJV, dbBtmBorder&, Col_Amount%) <> 0 Then
    setOptState("JV")
    VeriByWS dbMonthTop&, 276
End
End If
End Sub
Sub PostXLn(ByVal WkNow As Object, ByVal StrtLN%)
    dim i%, k%, mSigma%, sAName$, sAudWD$
    dim yCursor as object
    dim sEndCD$, WkAcct$

```

```

71 Rem (6)----- Change to Automatic calculation mode -----
72
73 Application.Calculation = xlAutomatic
74 Rem (7)----- Protect BS,PL, WkDpt(i) after FS Calculation -----
75 WkBS.Protect ("pwd")
76 WkPL.Protect ("pwd")
77 If IsMultiDP Then
78     For i% = 1 To NumDpt
79         WkDpt(i).Protect ("pwd")
80     Next i%
81 End If
82 Rem(8)----- Handling where B/S Audit line shows other than zero -----
83 ChkAftWSCalc
84 Rem -----
85 If IsAWDlOpen() Then
86     oAWDlG.Unload
87 End If
88 GiveDTHeader
89 Calculate
90 End Sub
91 Rem ----- Verify Dr/Cr Balance for JV,AP,AR,MW -----
92 Sub ChkB4WsCalc()
93     setCol4All
94     Rem -----
95     If APUse Then
96         getRowArray "AP", "ValMo", cMonth%, cMonth%
97         Col_Gross% = Col_InArray("AP", "Gross")
98         Col_Amount% = Col_InArray("AP", "Amount")
99         If CellValue(WkAP, dbBtmBorder&, Col_Gross%) <> CellValue(WkAP, dbBtmBorder&, Col_Amount%) Then
100             setOptState ("AP")
101             VeriByWS dbMonthTop&, 276
102         End
103     End If
104 End If
105 Rem -----
106 If ARUse Then
107     getRowArray "AR", "ValMo", cMonth%, cMonth%
108     Col_Gross% = Col_InArray("AR", "Gross")
109     Col_Amount% = Col_InArray("AR", "Amount")
110     If CellValue(WkAR, dbBtmBorder&, Col_Gross%) <> CellValue(WkAR, dbBtmBorder&, Col_Amount%) Then
111         setOptState ("AR")
112         VeriByWS dbMonthTop&, 276
113     End
114 End If
115 End If
116 Rem -----
117 If MWUse Then
118     getRowArray "MW", "ValMo", cMonth%, cMonth%
119     Col_Gross% = Col_InArray("MW", "Gross")
120     Col_Amount% = Col_InArray("MW", "Amount")
121     If CellValue(WkMW, dbBtmBorder&, Col_Gross%) <> CellValue(WkMW, dbBtmBord
122         setOptState ("MW")
123         VeriByWS dbMonthTop&, 276
124     End
125 End If
126 End If
127 Rem -----
128 getRowArray "JV", "ValMo", cMonth%, cMonth%
129 Col_Gross% = Col_InArray("JV", "Gross")
130 Col_Amount% = Col_InArray("JV", "Amount")
131 If CellValue(WkJV, dbBtmBorder&, Col_Gross%) <> 0 Or CellValue(WkJV, dbBtmBorder&
132     setOptState ("JV")
133     VeriByWS dbMonthTop&, 276
134 End
135 End If
136 End Sub
137 Sub PostXLn(ByVal WkNow As Object, ByVal StrtLN%)
138     Dim i%, k%, mSigma%, sAName$, sAudWD$
139     Dim sEndCD$, WkAcct$
140     Dim bQuitKey As Boolean

```

```

141 dim bQuitKey as boolean
142 dim BFamt&, CurrAmt&, ArrBP
143 BPROw& = StrtLN%
144 Do While True
145     bQuitKey = False
146     SelectCellFS(WkNow, BPROw&, BKCol)
147     WkAcct$ = CellString(WkNow, BPROw&, ActCol)
148     sAName$ = CellString(WkNow, BPROw&+1, CFCol-1)
149     sEndCD$ = UCase(Trim(WkAcct$))
150     sAudWD$ = myIIF(WkNow.Name = "BS", BSAudCD, PLAudCD)
151     Rem -----(Do not change the order of the following if statements)-----
152     If sEndCD$ = "ENDLN" Then
153         If sAName$ <> sAudWD$ then
154             SetCellString WkNow, BPROw&+1, CFCol-1, sAudWD$
155         End If
156         Exit Sub
157     ElseIf sAName$ = sAudWD$ Then
158         If sEndCD$ <> "ENDLN" then
159             SetCellString WkNow, BPROw&, ActCol, "ENDLN"
160         End If
161         Exit Sub
162     ElseIf BPROw& > 300 Then
163         Do While Cellstring(WkNow, BPROw&, CFCol-1) = ""
164             BPROw& = BPROw& - 1
165         Loop
166         SetCellString WkNow, BPROw&, ActCol, "ENDLN"
167         SetCellString WkNow, BPROw&+1, CFCol-1, sAudWD$
168         Exit Sub
169     ElseIf myIsEmpty(WkNow, BPROw&, 2) Then
170         bQuitKey = True
171     ElseIf Val(WkAcct$) = 0 then
172         bQuitKey = True
173     ElseIf (Val(WkAcct$) = Val(REEndCD)) Then
174         bQuitKey = True
175     End If
176     If bQuitKey = False then
177         Rem -----(Get brought forward amounts)-----
178         if CorpOrIndv = "Indv" Then
179             BFamt& = IndvBF&(WkNow, WkAcct$, BPROw&) 'see INDIV
180         else
181             if A_VBSCD(WkAcct$) then 'BS Accounts
182                 If (Val(WkAcct$) = Val(REBegCD)) Then
183                     If CBS_CM% = CBS_YE% + 1 Or CBS_CM% = CBS_YB% Then
184                         BFamt& = CellValue(WkNow, getFSRow%(WkBS, REEndCD, ActCol), CBS_CM%-1)
185                     Else
186                         BFamt& = CellValue(WkNow, getFSRow%(WkBS, REBegCD, ActCol), CBS_CM%-1)
187                     End If
188                 Else
189                     BFamt& = CellValue(WkNow, BPROw&, CBS_CM%-1)
190                 End If
191             elseif A_VPLCD(WkAcct$) then
192                 Rem Only when PL accounts also appear on BS
193                 Rem For each Dpt, posting will be done when dataArray is posted
194                 If CBS_CM% = CBS_YE% + 1 Or CBS_CM% = CBS_YB% Then
195                     BFamt& = 0
196                 Else
197                     If CBS_CM% <= CBS_YE% Then
198                         BFamt& = RangeSum(WkNow, CBS_YB%, BPROw&, CBS_CM%-1, BPROw&)
199                     Else
200                         BFamt& = RangeSum(WkNow, CBS_YE%+1, BPROw&, CBS_CM%-1, BPROw&)
201                     End If
202                 End If
203             end if
204         End If
205         Rem ----- BK summation -----
206         AmtArray&(0, 0) = AcctSum(WkBK, WkAcct$, oBKSDesc, BKRRange)
207         Rem ----- JV summation -----
208         AmtArray&(0, 1) = AcctSum(WkJV, WkAcct$, oJVSDesc, JVRRange)
209         Rem ----- AP summation -----
210         if APUse Then

```

```

141 Dim BFamt&, CurrAmt&, ArrBP
142 BPROw& = StrtLN
143 Do While True
144     bQuitKey = False
145     SelectCellFS WkNow, BPROw&, BKCol
146     WkAcct$ = CellString(WkNow, BPROw&, ActCol)
147     sAName$ = CellString(WkNow, BPROw& + 1, CFCol - 1)
148     sEndCD$ = UCase(Trim(WkAcct$))
149     sAudWD$ = myIIF(WkNow.Name = "BS", BSAudCD, PLAudCD)
150     Rem -----(Do not change the order of the following if statements)-----
151     If sEndCD$ = "ENDLN" Then
152         If sAName$ <> sAudWD$ Then
153             SetCellString WkNow, BPROw& + 1, CFCol - 1, sAudWD$
154         End If
155         Exit Sub
156     ElseIf sAName$ = sAudWD$ Then
157         If sEndCD$ <> "ENDLN" Then
158             SetCellString WkNow, BPROw&, ActCol, "ENDLN"
159         End If
160         Exit Sub
161     ElseIf BPROw& > 300 Then
162         Do While CellString(WkNow, BPROw&, CFCol - 1) = ""
163             BPROw& = BPROw& - 1
164         Loop
165         SetCellString WkNow, BPROw&, ActCol, "ENDLN"
166         SetCellString WkNow, BPROw& + 1, CFCol - 1, sAudWD$
167         Exit Sub
168     ElseIf myIsEmpty(WkNow, BPROw&, 2) Then
169         bQuitKey = True
170     ElseIf Val(WkAcct$) = 0 Then
171         bQuitKey = True
172     ElseIf (Val(WkAcct$) = Val(REEndCD)) Then
173         bQuitKey = True
174     End If
175     If bQuitKey = False Then
176         Rem -----(Get brought forward amounts)-----
177         If CorpOrIndv = "Indv" Then
178             BFamt& = IndvBF&(WkNow, WkAcct$, BPROw&) 'see INDIV
179         Else
180             If A_VBSCD(WkAcct$) Then 'BS Accounts
181                 If (Val(WkAcct$) = Val(REBegCD)) Then
182                     If CBS_CM% = CBS_YE% + 1 Or CBS_CM% = CBS_YB% Ther
183                         BFamt& = CellValue(WkNow, getFSRow%(WkBS, REEndC
184                     Else
185                         BFamt& = CellValue(WkNow, getFSRow%(WkBS, REBegC
186                     End If
187                 Else
188                     BFamt& = CellValue(WkNow, BPROw&, CBS_CM% - 1)
189                 End If
190             ElseIf A_VPLCD(WkAcct$) Then
191                 Rem Only when PL accounts also appear on BS
192                 Rem For each Dpt, posting will be done when dataArray is posted
193                 If CBS_CM% = CBS_YE% + 1 Or CBS_CM% = CBS_YB% Then
194                     BFamt& = 0
195                 Else
196                     If CBS_CM% <= CBS_YE% Then
197                         BFamt& = RangeSum(WkNow, CBS_YB%, BPROw&, CBS
198                     Else
199                         BFamt& = RangeSum(WkNow, CBS_YE% + 1, BPROw&, C
200                     End If
201                 End If
202             End If
203         End If
204         Rem ----- BK summation -----
205         AmtArray&(0, 0) = AcctSum(WkBK, WkAcct$, BKRRange)
206         Rem ----- JV summation -----
207         AmtArray&(0, 1) = AcctSum(WkJV, WkAcct$, JVRRange)
208         Rem ----- AP summation -----
209         If APUse Then
210             mSigma% = myIIF(CStr(WkAcct$) = CStr(APCD), -1, 1)

```

```

        mSigma%=myIIF(CStr(WkAcct$) = CStr(APCD), -1, 1)
        AmtArray&(0, 2) = AcctSum(WkAP, WkAcct$, oAPSDesc, APRange) * mSigma%
    end if
    Rem ----- AR summation -----
    if ARUse Then
        AmtArray&(0, 3) = AcctSum(WkAR, WkAcct$, oARSDesc, ARRange)
    end if
    Rem ----- MW summation -----
    if MWUse Then
        AmtArray&(0, 4) = AcctSum(WkMW, WkAcct$, oMWSDesc, MWRRange)
    end if
    Rem -----posting to BS/PL -----
    Set yCursor = WkNow.getCellRangeByPosition(BFCol-1, BPRow&-1, RMCol-1, BPRow&-1)
    Rem -----setDataArray() corresponding to row array -----
    if ARUse Then
        mSigma%=myIIF(CStr(WkAcct$) = CStr(ARCD), 1, -1)
        ArrBP=Array(BFAmt&, AmtArray&(0, 0), AmtArray&(0, 1), AmtArray&(0, 2), AmtArray&(0, 3)*mSigma%)
    elseif MWUse Then
        ArrBP=Array(BFAmt&, AmtArray&(0, 0), AmtArray&(0, 1), AmtArray&(0, 2), AmtArray&(0, 4))
    end if
    yCursor.setDataArray(Array(ArrBP))
    CurrAmt& = RangeSum(WkNow, BKCol, BPRow&, RMCol, BPRow&)
    if A_VPLCD(WkAcct$) then
        Rem ----- posting of PL current month data to trend range -----
        If Not WS_BS(WkAcct) then
            SetCellValue WkPL, BPRow&, CBS_CM%, CurrAmt&
        End If
    end if
    Rem -----(compilation and posting for departmental accounting)-----
    If IsMultiDP then
        if A_VAllocCD(WkAcct$) then
            For i% = 1 to NumDpt
                Set yCursor = WkDpt(i%).getCellRangeByPosition(BFCol-1, BPRow&-1, RMCol-1, BPRow&-1)
                Rem -----(get cumulative to prior month)-----
                If CBS_CM% = CBS_YE% + 1 Or CBS_CM% = CBS_YB% Then
                    BFAmt& = 0
                Else
                    If CBS_CM% <= CBS_YE% Then
                        BFAmt& = RangeSum(WkDpt(i), CBS_YB%, BPRow&, CBS_CM%-1, BPRow&)
                    Else
                        BFAmt& = RangeSum(WkDpt(i), CBS_YE%+1, BPRow&, CBS_CM%-1, BPRow&)
                    End If
                End If
            End If
            Rem -----

            if ARUse Then
                Rem note the minus sign in front of AmtArray&(i%, 3) for AR
                ArrBP=Array(BFAmt&, AmtArray&(i%, 0), AmtArray&(i%, 1), AmtArray&(i%, 2), -AmtArray&(i%, 3))
            elseif MWUse Then
                ArrBP=Array(BFAmt&, AmtArray&(i%, 0), AmtArray&(i%, 1), AmtArray&(i%, 2), AmtArray&(i%, 4))
            end if
            yCursor.setDataArray(Array(ArrBP))
            CurrAmt& = RangeSum(WkDpt(i), BKCol, BPRow&, RMCol, BPRow&)
            SetCellValue WkDpt(i), BPRow&, CBS_CM%, CurrAmt&
            for k% = 0 to 4
                AmtArray&(i%, k%) = 0
            next k%
        Next i%
    end if
End If
BPRow& = BPRow& + 1
Loop
End Sub
Sub PostXFm()
    Rem ----- Current month summation of bank transaction and consumption tax -----
    Dim aArgs(0) As Variant
    Dim oFound as object, tBKRange As object
    Dim icCD%, WHTaxRow&, CTXRow&, CXPRow&, BKChgRow&

```

```

        AmtArray&(0, 2) = AcctSum(WkAP, WkAcct$, APRange) * mSigma%
    End If
    Rem ----- AR summation -----
    If ARUse Then
        AmtArray&(0, 3) = AcctSum(WkAR, WkAcct$, ARRange)
    End If
    Rem ----- MW summation -----
    If MWUse Then
        AmtArray&(0, 4) = AcctSum(WkMW, WkAcct$, MWRRange)
    End If
    Rem -----posting to BS/PL -----
    SetCellValue WkNow, BPRow&, BFCol, BFAmt&
    SetCellValue WkNow, BPRow&, BKCol, AmtArray&(0, 0)
    SetCellValue WkNow, BPRow&, JVCOL, AmtArray&(0, 1)
    SetCellValue WkNow, BPRow&, APCol, AmtArray&(0, 2)
    If ARUse Then
        mSigma% = myIIF(CStr(WkAcct$) = CStr(ARCD), 1, -1)
        SetCellValue WkNow, BPRow&, ARCol, AmtArray&(0, 3) * mSigma%
    Elseif MWUse Then
        SetCellValue WkNow, BPRow&, MWCol, AmtArray&(0, 4)
    End If
    CurrAmt& = RangeSum(WkNow, BKCol, BPRow&, RMCOL, BPRow&)
    If A_VPLCD(WkAcct$) Then
        Rem ----- posting of PL current month data to trend range -----
        If Not WS_BS(WkAcct) Then
            SetCellValue WkPL, BPRow&, CBS_CM%, CurrAmt&
        End If
    End If
    Rem -----(compilation and posting for departmental accounting)-----
    If NumDpt > 1 Then
        If A_VAllocCD(WkAcct$) Then
            For i% = 1 To NumDpt
                Rem -----(get cumulative to prior month)-----
                If CBS_CM% = CBS_YE% + 1 Or CBS_CM% = CBS_YB% Then
                    BFAmt& = 0
                Else
                    If CBS_CM% <= CBS_YE% Then
                        BFAmt& = RangeSum(WkDpt(i%), CBS_YB%, BPRo
                    Else
                        BFAmt& = RangeSum(WkDpt(i%), CBS_YE% + 1, BI
                    End If
                End If
            End If
            Rem -----
            SetCellValue WkDpt(i%), BPRow&, BFCOL, BFAmt&
            SetCellValue WkDpt(i%), BPRow&, BKCol, AmtArray&(i%, 0)
            SetCellValue WkDpt(i%), BPRow&, JVCOL, AmtArray&(i%, 1)
            SetCellValue WkDpt(i%), BPRow&, APCOL, AmtArray&(i%, 2)
            If ARUse Then
                Rem note the minus sign in front of AmtArray&(i%, 3) for AF
                SetCellValue WkDpt(i%), BPRow&, ARCol, -AmtArray&(i%,
            Elseif MWUse Then
                SetCellValue WkDpt(i%), BPRow&, MWCol, AmtArray&(i%,
            End If
            CurrAmt& = RangeSum(WkDpt(i), BKCol, BPRow&, RMCOL, BPR
            SetCellValue WkDpt(i%), BPRow&, CBS_CM%, CurrAmt&
            For k% = 0 To 4
                AmtArray&(i%, k%) = 0
            Next k%
        Next i%
    End If
End If
BPRow& = BPRow& + 1
Loop
End Sub
Sub PostXFm()
    Rem ----- Current month summation of bank transaction and consumption tax -----
    Dim oFound As Object, tBKRange As Range, oRange As Range
    Dim icCD%, WHTaxRow&, CTXRow&, CXPRow&, BKChgRow&

```

```

dim iStrRow&, iEndRow&
dim BKChgAmt&, tVal&, WHTax&, BKAmt&, CTPAY&, CTxAmt&, nwVal&, CurrAmt&
dim ac%, MBKC%, ColName$
getPagePlan "BK", cMonth%
Rem If WkBK.AutoFilterMode Then
Rem   WkBK.Columns("A:M").AutoFilter
Rem End If
icCD% = Col_BKCD%
Rem ----- Calculation by bank -----
CTXRow& = getFSRow%(WkBS, CRTXCD, ActCol)
CXPRow& = getFSRow%(WkBS, PYTXCD, ActCol)
WHTaxRow& = getFSRow%(WkBS, WHTaxCD, ActCol)
BKChgRow& = getFSRow%(WkPL, BKChgCD, ActCol)
iEndRow& = dbMonthTop& - 1
Do While True
  RNM$ = CAlpha(icCD%) & CStr(iEndRow& + 1) & cjc$
  RNM$ = RNM$ & CAlpha(icCD%) & CStr(dbMonthBtm&)
  Set tBKRange = WkBK.getCellRangeByName(RNM$)
  MBKC% = CellValue(WkBK, iEndRow&+1, Col_BKCD%)
  BPRow& = getFSRow%(WkBS, CStr(MBKC%), ActCol)
  If iEndRow& = dbMonthTop& then
    SetCellValue WkCL, 1, 28, BPRow&
  End If
  SetCellValue WkCL, BPRow&, 28, CellValue(WkBS, BPRow&, 2)
  iStrRow& = getFirstRow&(tBKRange, CStr(MBKC%))
  iEndRow& = getLastRow&(tBKRange, CStr(MBKC%))
  SetCellValue WkCL, BPRow&, 30, CellValue(WkBK, iEndRow&, Col_Gross%+3)
  SelectCellFS WkBS, BPRow&, BKCol
  Rem ===Summation of increase and decrease of Bank amounts =====
  If CellValue(WkBK, tBKRange.RangeAddress.StartRow+1, Col_ValMo%) = cMonth% then
    BKAmt& = RangeSum&(WkBK, Col_Amount%, iStrRow&, Col_Amount%, iEndRow&)
  Else
    BKAmt& = 0
  End If
  Rem ===Summation of VAT received only when the column is setup =====
  If Col_CXCr% < Col_Border% then
    CTPAY& = RangeSum&(WkBK, Col_CXCr%, iStrRow&, Col_CXCr%, iEndRow&)
  Else
    CTPAY& = 0
  End If
  Rem =====(Summation of bank transfer charges)=====
  BKChgAmt& = RangeSum&(WkBK, Col_BKChg%, iStrRow&, Col_BKChg%, iEndRow&)
  CTxAmt& = Int((BKChgAmt& / (100 + CTXRate)) * CTXRate)
  tVal& = Val(CellValue(WkPL, BKChgRow&, BKCol))
  BKChgAmt& = BKChgAmt& - CTxAmt&
  SetCellValue WkPL, BKChgRow&, BKCol, tVal& + BKChgAmt&
  CurrAmt& = RangeSum(WkPL, BKCol, BKChgRow&, RMCOL, BKChgRow&)
  SetCellValue WkPL, BKChgRow&, CBS_CM%, CurrAmt&
  If IsMultiDP Then
    tVal& = Val(CellValue(WkDpt(NumDpt), BKChgRow&, BKCol))
    SetCellValue WkDpt(NumDpt), BKChgRow&, BKCol, tVal& + BKChgAmt&
    CurrAmt& = RangeSum(WkDpt(NumDpt), BKCol, BKChgRow&, RMCOL, BKChgRow&)
    SetCellValue WkDpt(NumDpt), BKChgRow&, CBS_CM%, CurrAmt&
  End If
  Rem =====Summation of VAT paid=====
  CTxAmt& = RangeSum&(WkBK, Col_CXDr%, iStrRow&, Col_CXDr%, iEndRow&) + CTxAmt&
  Rem =====Posting of VAT paid and VAT received =====
  tVal& = CellValue(WkBS, CTXRow&, BKCol)
  SetCellValue WkBS, CTXRow&, BKCol, tVal& + CTxAmt&
  If Col_CXCr% < Col_Border% then
    tVal& = CellValue(WkBS, CXPRow&, BKCol)
    SetCellValue WkBS, CXPRow&, BKCol, tVal& + CTPAY&
  End If
  Rem ===== Summation of Withholding tax received and posting to BS =====
  WHTax& = RangeSum&(WkBK, Col_WHT%, iStrRow&, Col_WHT%, iEndRow&)
  tVal& = CellValue(WkBS, WHTaxRow&, BKCol)
  SetCellValue WkBS, WHTaxRow&, BKCol, tVal& + WHTax&
  Rem ===Posting to BK account on BS, and month end balance to the BK row on CL =====
  tVal& = CellValue(WkBS, BPRow&, BKCol)
  tVal& = tVal& - BKAmt& - CTPAY& - CTxAmt& - BKChgAmt& - WHTax& 'note CTxAmt

```

```

281 Dim iStrRow&, iEndRow&
282 Dim BKChgAmt&, tVal&, WHTax&, BKAmt&, CTPAY&, CTxAmt&, nwVal&, CurrAmt&
283 Dim AC%, MBKC%, tBKC%, ColName$
284 getPagePlan "BK", cMonth%
285 Rem If WkBK.AutoFilterMode Then
286 Rem   WkBK.Columns("A:M").AutoFilter
287 Rem End If
288 icCD% = Col_BKCD%
289 Rem ----- Calculation by bank -----
290 CTXRow& = getFSRow%(WkBS, CRTXCD, ActCol)
291 CXPRow& = getFSRow%(WkBS, PYTXCD, ActCol)
292 WHTaxRow& = getFSRow%(WkBS, WHTaxCD, ActCol)
293 BKChgRow& = getFSRow%(WkPL, BKChgCD, ActCol)
294 iEndRow& = dbMonthTop& - 1
295 Do While True
296   RNM$ = CAlpha(icCD%) & CStr(iEndRow& + 1) & ".:"
297   RNM$ = RNM$ & CAlpha(icCD%) & CStr(dbMonthBtm&)
298   Set tBKRange = WkBK.Range(RNM$)
299   MBKC% = WkBK.Cells(iEndRow& + 1, Col_BKCD%).Value
300   BPRow& = getFSRow%(WkBS, CStr(MBKC%), ActCol)
301   If iEndRow& = dbMonthTop& Then
302     SetCellValue WkCL, 1, 28, BPRow&
303   End If
304   SetCellValue WkCL, BPRow&, 28, CellValue(WkBS, BPRow&, 2)
305   iStrRow& = getFirstRow&(tBKRange, CStr(MBKC%))
306   iEndRow& = getLastRow&(tBKRange, CStr(MBKC%))
307   SetCellValue WkCL, BPRow&, 30, CellValue(WkBK, iEndRow&, Col_Gross% + 3)
308   SelectCell BPRow&, BKCol
309   Rem ===Summation of increase and decrease of Bank amounts =====
310   If CellValue(WkBK, iStrRow&, Col_ValMo%) = cMonth% Then
311     BKAmt& = RangeSum&(WkBK, Col_Amount%, iStrRow&, Col_Amount%, iEndRow&)
312   Else
313     BKAmt& = 0
314   End If
315   Rem ===Summation of VAT received only when the column is setup =====
316   If Col_CXCr% < Col_Border% Then
317     CTPAY& = RangeSum&(WkBK, Col_CXCr%, iStrRow&, Col_CXCr%, iEndRow&)
318   Else
319     CTPAY& = 0
320   End If
321   Rem =====(Summation of bank transfer charges)=====
322   BKChgAmt& = RangeSum&(WkBK, Col_BKChg%, iStrRow&, Col_BKChg%, iEndRow&)
323   CTxAmt& = Int((BKChgAmt& / (100 + CTXRate)) * CTXRate)
324   tVal& = Val(CellValue(WkPL, BKChgRow&, BKCol))
325   BKChgAmt& = BKChgAmt& - CTxAmt&
326   SetCellValue WkPL, BKChgRow&, BKCol, tVal& + BKChgAmt&
327   CurrAmt& = RangeSum(WkPL, BKCol, BKChgRow&, RMCOL, BKChgRow&)
328   SetCellValue WkPL, BKChgRow&, CBS_CM%, CurrAmt&
329   If IsMultiDP Then
330     tVal& = Val(CellValue(WkDpt(NumDpt), BKChgRow&, BKCol))
331     SetCellValue WkDpt(NumDpt), BKChgRow&, BKCol, tVal& + BKChgAmt&
332     CurrAmt& = RangeSum(WkDpt(NumDpt), BKCol, BKChgRow&, RMCOL, BKChgRow&)
333     SetCellValue WkDpt(NumDpt), BKChgRow&, CBS_CM%, CurrAmt&
334   End If
335   Rem =====Summation of VAT paid=====
336   CTxAmt& = RangeSum&(WkBK, Col_CXDr%, iStrRow&, Col_CXDr%, iEndRow&) +
337   Rem =====Posting of VAT paid and VAT received =====
338   tVal& = CellValue(WkBS, CTXRow&, BKCol)
339   SetCellValue WkBS, CTXRow&, BKCol, tVal& + CTxAmt&
340   If Col_CXCr% < Col_Border% Then
341     tVal& = CellValue(WkBS, CXPRow&, BKCol)
342     SetCellValue WkBS, CXPRow&, BKCol, tVal& + CTPAY&
343   End If
344   Rem ===== Summation of Withholding tax received and posting to BS =====
345   WHTax& = RangeSum&(WkBK, Col_WHT%, iStrRow&, Col_WHT%, iEndRow&)
346   tVal& = CellValue(WkBS, WHTaxRow&, BKCol)
347   SetCellValue WkBS, WHTaxRow&, BKCol, tVal& + WHTax&
348   Rem ===Posting to BK account on BS, and month end balance to the BK row on CL =
349   tVal& = CellValue(WkBS, BPRow&, BKCol)
350   tVal& = tVal& - BKAmt& - CTPAY& - CTxAmt& - BKChgAmt& - WHTax& 'note CTxAmt

```



```

SetCellValue WkBS, BPRow&, BKCol, tVal&
tVal& = CellValue(WkBS, BPRow&, BFCol) + tVal& + CellValue(WkBS, BPRow&, JVCol)
SetCellValue WkCL, BPRow&, 29, tVal&
Rem =====
If iEndRow&+1 > dbMonthBtm& then
    exit do
End if
Loop
Rem ----- Amounts to AR bottom row -----
If ARUse Then
    BPRow& = getFSRow%(WkBS, ARCD, ActCol)
    dbBtmBorder&=RowArray(2, 3)
    Col_Amount%=Col_InArray("AR", "Amount")
    ac=myIIF(bAllocDpt, NumDpt, 0)
    tVal& = CellValue(WkBS, BPRow&, BKCol - 1) 'Balance at beginning
    SetCellValue WkAR, dbBtmBorder& + 1, Col_Amount% + ac + 1, tVal&
    tVal& = CellValue(WkBS, BPRow&, ARCol) 'Current month Increase
    SetCellValue WkAR, dbBtmBorder& + 1, Col_Amount% + ac + 2, tVal&
    tVal& = -CellValue(WkBS, BPRow&, BKCol) - CellValue(WkBS, BPRow&, JVCol)
    SetCellValue WkAR, dbBtmBorder& + 1, Col_Amount% + ac + 3, tVal&
    tVal& = RangeSum(WkBS, BFCol, BPRow&, ARCol, BPRow&) 'Balance at ending
    SetCellValue WkAR, dbBtmBorder& + 1, Col_Amount% + ac + 4, tVal&
End If
Rem ----- Amounts to MW bottom row -----
If MWUse Then
    BPRow& = getFSRow%(WkBS, MWCD, ActCol)
    dbBtmBorder&=RowArray(2, 4)
    Col_Amount%=Col_InArray("MW", "Amount")
    ac=myIIF(bAllocDpt, NumDpt, 0)
    tVal& = CellValue(WkBS, BPRow&, BKCol - 1) 'Balance at beginning
    SetCellValue WkMW, dbBtmBorder& + 1, Col_Amount% + ac + 1, tVal&
    tVal& = CellValue(WkBS, BPRow&, BKCol) + CellValue(WkBS, BPRow&, JVCol)
    SetCellValue WkMW, dbBtmBorder& + 1, Col_Amount% + ac + 2, tVal&
    tVal& = -CellValue(WkBS, BPRow&, MWCol) 'Current month payment
    SetCellValue WkMW, dbBtmBorder& + 1, Col_Amount% + ac + 3, tVal&
    tVal& = RangeSum(WkBS, BFCol, BPRow&, MWCol, BPRow&) 'Balance at ending
    SetCellValue WkMW, dbBtmBorder& + 1, Col_Amount% + ac + 4, tVal&
End If
Rem ----- Amounts to AP bottom row -----
If APUse Then
    BPRow& = getFSRow%(WkBS, APCD, ActCol)
    dbBtmBorder&=RowArray(2, 2)
    Col_Amount%=Col_InArray("AP", "Amount")
    ac=myIIF(bAllocDpt, NumDpt, 0)
    tVal& = -CellValue(WkBS, BPRow&, BKCol-1) 'Balance at beginning
    SetCellValue WkAP, dbBtmBorder& + 1, Col_Amount% + ac + 1, tVal&
    tVal& = -CellValue(WkBS, BPRow&, APCol) 'Increase
    SetCellValue WkAP, dbBtmBorder& + 1, Col_Amount% + ac + 2, tVal&
    tVal& = CellValue(WkBS, BPRow&, BKCol) + CellValue(WkBS, BPRow&, JVCol) 'Decrease
    SetCellValue WkAP, dbBtmBorder& + 1, Col_Amount% + ac + 3, tVal&
    tVal& = -RangeSum(WkBS, BFCol, BPRow&, APCol, BPRow&) 'Balance at ending
    SetCellValue WkAP, dbBtmBorder& + 1, Col_Amount% + ac + 4, tVal&
End If
End Sub
Sub PostBFCF(ByVal WkNow As Object, ByVal StrtLN%)
    Dim CumAmount&, WkAcct$
    BPRow& = StrtLN%
    Do While True
        SelectCellIFS(WkNow, BPRow&, BKCol)
        WkAcct$ = CellString(WkNow, BPRow&, 2)
        If Val(WkAcct$) > 0 And Len(WkAcct$) > 0 Then
            If A_VBSCD(WkAcct$) Then Rem BS Accounts
                If Val(WkAcct$) = Val(REEndCD) Then
                    Rem do nothing
                Else
                    CumAmount& = RangeSum(WkNow, BFCol, BPRow&, RMCOL, BPRow&)
                    SetCellValue WkNow, BPRow&, CBS_CM%, CumAmount&
                End If
            elseif A_VPLCD(WkAcct$) Then Rem PL Accounts
                CumAmount& = RangeSum(WkNow, BKCol, BPRow&, RMCOL, BPRow&)
            end if
        End If
    End Sub

```

```

351 SetCellValue WkBS, BPRow&, BKCol, tVal&
352 tVal& = CellValue(WkBS, BPRow&, BFCol) + tVal& + CellValue(WkBS, BPRow&, JVCol)
353 SetCellValue WkCL, BPRow&, 29, tVal&
354 Rem =====
355 If iEndRow& + 1 > dbMonthBtm& Then
356     Exit Do
357 End If
358 Loop
359 Rem ----- Amounts to AR bottom row -----
360 If ARUse Then
361     BPRow& = getFSRow%(WkBS, ARCD, ActCol)
362     dbBtmBorder& = RowArray(2, 3)
363     Col_Amount% = Col_InArray("AR", "Amount")
364     AC = myIIF(bAllocDpt, NumDpt, 0)
365     tVal& = CellValue(WkBS, BPRow&, BKCol - 1) 'Balance at beginning
366     SetCellValue WkAR, dbBtmBorder& + 1, Col_Amount% + AC + 1, tVal&
367     tVal& = CellValue(WkBS, BPRow&, ARCol) 'Current month Increase
368     SetCellValue WkAR, dbBtmBorder& + 1, Col_Amount% + AC + 2, tVal&
369     tVal& = -CellValue(WkBS, BPRow&, BKCol) - CellValue(WkBS, BPRow&, JVCOL)
370     SetCellValue WkAR, dbBtmBorder& + 1, Col_Amount% + AC + 3, tVal&
371     tVal& = RangeSum(WkBS, BFCol, BPRow&, ARCol, BPRow&) 'Balance at ending
372     SetCellValue WkAR, dbBtmBorder& + 1, Col_Amount% + AC + 4, tVal&
373 End If
374 Rem ----- Amounts to MW bottom row -----
375 If MWUse Then
376     BPRow& = getFSRow%(WkBS, MWCD, ActCol)
377     dbBtmBorder& = RowArray(2, 4)
378     Col_Amount% = Col_InArray("MW", "Amount")
379     AC = myIIF(bAllocDpt, NumDpt, 0)
380     tVal& = CellValue(WkBS, BPRow&, BKCol - 1) 'Balance at beginning
381     SetCellValue WkMW, dbBtmBorder& + 1, Col_Amount% + AC + 1, tVal&
382     tVal& = CellValue(WkBS, BPRow&, BKCol) + CellValue(WkBS, BPRow&, JVCOL)
383     SetCellValue WkMW, dbBtmBorder& + 1, Col_Amount% + AC + 2, tVal&
384     tVal& = -CellValue(WkBS, BPRow&, MWCol) 'Current month payment
385     SetCellValue WkMW, dbBtmBorder& + 1, Col_Amount% + AC + 3, tVal&
386     tVal& = RangeSum(WkBS, BFCol, BPRow&, MWCol, BPRow&) 'Balance at ending
387     SetCellValue WkMW, dbBtmBorder& + 1, Col_Amount% + AC + 4, tVal&
388 End If
389 Rem ----- Amounts to AP bottom row -----
390 If APUse Then
391     BPRow& = getFSRow%(WkBS, APCD, ActCol)
392     dbBtmBorder& = RowArray(2, 2)
393     Col_Amount% = Col_InArray("AP", "Amount")
394     AC = myIIF(bAllocDpt, NumDpt, 0)
395     tVal& = -CellValue(WkBS, BPRow&, BKCol - 1) 'Balance at beginning
396     SetCellValue WkAP, dbBtmBorder& + 1, Col_Amount% + AC + 1, tVal&
397     tVal& = -CellValue(WkBS, BPRow&, APCol) 'Increase
398     SetCellValue WkAP, dbBtmBorder& + 1, Col_Amount% + AC + 2, tVal&
399     tVal& = CellValue(WkBS, BPRow&, BKCol) + CellValue(WkBS, BPRow&, JVCOL)
400     SetCellValue WkAP, dbBtmBorder& + 1, Col_Amount% + AC + 3, tVal&
401     tVal& = -RangeSum(WkBS, BFCol, BPRow&, APCol, BPRow&) 'Balance at ending
402     SetCellValue WkAP, dbBtmBorder& + 1, Col_Amount% + AC + 4, tVal&
403 End If
404 End Sub
405 Sub PostBFCF(ByVal WkNow As Object, ByVal StrtLN%)
406     Dim CumAmount&, WkAcct$, oRange As Range
407     BPRow& = StrtLN%
408     Do While True
409         SelectCellIFS WkNow, BPRow&, BKCol
410         WkAcct$ = CellString(WkNow, BPRow&, 2)
411         If Val(WkAcct$) > 0 And Len(WkAcct$) > 0 Then
412             If A_VBSCD(WkAcct$) Then 'BS Accounts
413                 If Val(WkAcct$) = Val(REEndCD) Then
414                     Rem do nothing
415                 Else
416                     CumAmount& = RangeSum(WkNow, BFCol, BPRow&, RMCOL, BPRow&, BPF)
417                     SetCellValue WkNow, BPRow&, CBS_CM%, CumAmount&
418                 End If
419             elseif A_VPLCD(WkAcct$) Then 'PL Accounts
420                 CumAmount& = RangeSum(WkNow, BKCol, BPRow&, RMCOL, BPRow&

```

```

        SetCellValue WkNow, BPRow&, CBS_CM%, CumAmount&
    end if
End If
If WkAcct$ = "ENDLN" Then
    Exit Do
Else
    BPRow& = BPRow& + 1
End If
Loop
End Sub
Function AcctSum&(ByVal WkWk as object, FkAcct$, oSearchDesc, oSRange as object)
    dim i%, j%, k%, xRow&, iCol%, oFLength%
    dim oFound as object, LnAcct$, LnMonth%, pSName$
    AcctSum& = 0
    pSName$ = WkWk.getName()
    iCol% = iWS(pSName$)
    If RowArray(3,iCol%) then
        Exit Function
    End If
    oSearchDesc.SearchWords = true
    oSearchDesc.SearchString = FkAcct$
    oSearchDesc.SearchBackwards = False
    Set oFound = oSRange.findAll(oSearchDesc)
    If IsNull(oFound) then
        Exit Function
    Endif
    getColArray(pSName$)
    oFLength% = oFound.getCount()
    For i% = 0 to oFLength%-1
        xRow& = oFound.getByIndex(i%).RangeAddress.StartRow + 1
        for j% = 1 to oFound.getByIndex(i%).Rows.Count
            LnMonth% = Val(CellString(WkWk, xRow&, Col_ValMo%))
            LnAcct$ = CellString(WkWk, xRow&, Col_ActCD%)
            If LnMonth% = cMonth% and LnAcct$ = FkAcct$ then
                AcctSum& = AcctSum& + CellValue(WkWk, xRow&, Col_Amount%)
                if IsMultiDP AND bAllocDpt Then
                    If A_VallocCD(FkAcct$) Then
                        For k% = 1 To NumDpt
                            AmtArray&(k%, iCol%) = AmtArray&(k%, iCol%) + CellValue(WkWk, xRow&, Col_Amount%)
                        Next k%
                    End If
                end if
            End If
            xRow& = xRow& + 1
        next j%
    Next i%
End Function
Rem ----- Handling where B/S Audit line shows other than zero -----
Sub ChkAftWSCalc()
    dim i%, BSAudLN%, VeriV&
    dim iRow&, sBK$, bkRow&, cBKCD%
    Rem -----Tie in BS and BK balances-----
    iRow = CellValue(WkCL, 1, 28)
    Do While Len(CellString(WkCL, iRow, 28)) > 0
        If CellValue(WkCL, iRow, 29) <> CellValue(WkCL, iRow, 30) then
            sBK$ = CellString(WkBS, iRow, 2)
            MsgTxt$ = "There is an imbalance between BS and BK regarding" & chr(13)
            MsgTxt$ = MsgTxt$ & sBK$ & ">" & CellString(WkBS, iRow, 4) & "." & Chr(13)
            MsgTxt$ = MsgTxt$ & "Will proceed to Verification by row to identify the cause."
            cBKCD% = Col_InArray("BK", "BKCD")
            bkRow& = RowArray(0,0)+1
            Do While bkRow& <= RowArray(1,0)
                If CellString(WkBK, bkRow&, cBKCD%) = sBK$ Then
                    Exit Do
                End If
                bkRow& = bkRow& + 1
            Loop
            setOptState("BK")
            SelectCell bkRow&, cBKCD%
            MsgTitl$ = "Test of agreement of BS and BK balances"
        End If
    End Do
End Sub

```

```

        SetCellValue WkNow, BPRow&, CBS_CM%, CumAmount&
    End If
End If
If WkAcct$ = "ENDLN" Then
    Exit Do
Else
    BPRow& = BPRow& + 1
End If
Loop
End Sub
Function AcctSum&(ByVal WkWk As Object, FkAcct$, oSRange As Object)
    Dim k%, xRow&, iCol%, firstAddress
    Dim oFound As Object, LnAcct$, LnMonth%, pSName$
    AcctSum& = 0
    pSName$ = WkWk.Name
    iCol% = iWS(pSName$)
    If RowArray(3, iCol%) Then
        Exit Function
    End If
    End If
    getColArray (pSName$)
    firstAddress = oFound.Address
    Do
        xRow& = oFound.Row
        LnMonth% = Val(CellString(WkWk, xRow&, Col_ValMo%))
        LnAcct$ = CellString(WkWk, xRow&, Col_ActCD%)
        If LnMonth% = cMonth% And LnAcct$ = FkAcct$ Then
            AcctSum& = AcctSum& + CellValue(WkWk, xRow&, Col_Amount%)
            If IsMultiDP And bAllocDpt Then
                If A_VallocCD(FkAcct$) Then
                    For k% = 1 To NumDpt
                        AmtArray&(k%, iCol%) = AmtArray&(k%, iCol%) + CellValue(Wk
                    Next k%
                End If
            End If
        End If
    Do
        Set oFound = oSRange.FindNext(oFound)
    Loop While Not oFound Is Nothing And oFound.Address <> firstAddress
End Function
Rem ----- Handling where B/S Audit line shows other than zero -----
Sub ChkAftWSCalc()
    Dim i%, BSAudLN%, VeriV&, a%
    Dim iRow&, sBK$, bkRow&, cBKCD%
    Rem -----Tie in BS and BK balances-----
    iRow& = CellValue(WkCL, 1, 28)
    Do While Len(CellString(WkCL, iRow&, 28)) > 0
        If CellValue(WkCL, iRow&, 29) <> CellValue(WkCL, iRow&, 30) Then
            sBK$ = CellString(WkBS, iRow&, 2)
            MsgTxt$ = "There is an imbalance between BS and BK regarding" & Chr(13)
            MsgTxt$ = MsgTxt$ & sBK$ & ">" & CellString(WkBS, iRow, 4) & "." & Chr(13)
            MsgTxt$ = MsgTxt$ & "Will proceed to Verification by row to identify the cause."
            cBKCD% = Col_InArray("BK", "BKCD")
            bkRow& = RowArray(0, 0) + 1
            Do While bkRow& <= RowArray(1, 0)
                If CellString(WkBK, bkRow&, cBKCD%) = sBK$ Then
                    Exit Do
                End If
                bkRow& = bkRow& + 1
            Loop
            setOptState ("BK")
            SelectCell bkRow&, cBKCD%
            MsgTitl$ = "Test of agreement of BS and BK balances"
        End If
    End Do
End Sub

```

```

        resp% = MsgBox(MsgTxt$, 16, MsgTitl$)
        oDTDIg.EndExecute()
        VeriByWS bkRow&, "WsCalc"
    End
End If
iRow = iRow + 1
Loop
BSAudLN%=getFSRow%(WkBS, "ENDLN", ActCol) + 1
SelectCell BSAudLN%, BKCol
VeriV& = 0
For i% = BKCol To RMCol
    VeriV& = VeriV& + CellValue(WkBS, BSAudLN%, i%)
Next i%
If VeriV& <> 0 Then
    If CellValue(WkBS, BSAudLN%, BKCol) <> 0 Then
        setOptState("BK")
        VeriByWS dbMonthTop&, 276
    end if
    If CellValue(WkBS, BSAudLN%, JVCOL) <> 0 Then
        setOptState("JV")
        VeriByWS dbMonthTop&, 276
    end if
    If CellValue(WkBS, BSAudLN%, APCol) <> 0 Then
        setOptState("AP")
        VeriByWS dbMonthTop&, 276
    end if
    If ARUse Then
        If CellValue(WkBS, BSAudLN%, ARCol) <> 0 Then
            setOptState("AR")
            VeriByWS dbMonthTop&, 276
        End If
    end if
    If MWUse Then
        If CellValue(WkBS, BSAudLN%, MWCol) <> 0 Then
            setOptState("MW")
            VeriByWS dbMonthTop&, 276
        End If
    end if
End If
Rem -----
If IsMultiDP then
    VeriDept()
End If
End Sub
Rem =====(Verification after BS/PL Preparation)=====
Sub VeriDept()
    Dim VD(BKCol To RMCol) As Variant
    Dim i%, j%, DPTAudLN%
    DPTAudLN%=getFSRow%(WkPL, UCase(DPTAudCD), CFCOL-1)
    For i% = BKCol To RMCol
        for j% = 1 To NumDpt
            VD(i%) = VD(i%) + CellValue(WkDpt(j%), DPTAudLN%, i%)
        next j%
        VD(i%) = CellValue(WkPL, DPTAudLN%, i%) - VD(i%)
    Next i%
    If VD(BKCol) <> 0 Then
        setOptState("BK")
        VeriByWS dbMonthTop&, 276
    ElseIf VD(JVCOL) <> 0 Then
        setOptState("JV")
        VeriByWS dbMonthTop&, 276
    ElseIf VD(APCOL) <> 0 Then
        setOptState("AP")
        VeriByWS dbMonthTop&, 276
    End If
    If ARUse then
        If VD(ARCol) <> 0 Then
            setOptState("AR")
            VeriByWS dbMonthTop&, 276
        end if
    end if

```

```

        resp% = MsgBox(MsgTxt$, 16, MsgTitl$)
        oDTDIg.Hide
        VeriByWS bkRow&, "WsCalc"
    End
End If
iRow& = iRow& + 1
Loop
BSAudLN% = getFSRow%(WkBS, "ENDLN", ActCol) + 1
SelectCell BSAudLN%, BKCol
VeriV& = 0
For i% = BKCol To RMCol
    VeriV& = VeriV& + CellValue(WkBS, BSAudLN%, i%)
Next i%
If VeriV& <> 0 Then
    If CellValue(WkBS, BSAudLN%, BKCol) <> 0 Then
        setOptState ("BK")
        VeriByWS dbMonthTop&, 276
    End If
    If CellValue(WkBS, BSAudLN%, JVCOL) <> 0 Then
        setOptState ("JV")
        VeriByWS dbMonthTop&, 276
    End If
    If CellValue(WkBS, BSAudLN%, APCol) <> 0 Then
        setOptState ("AP")
        VeriByWS dbMonthTop&, 276
    End If
    If ARUse Then
        If CellValue(WkBS, BSAudLN%, ARCol) <> 0 Then
            setOptState ("AR")
            VeriByWS dbMonthTop&, 276
        End If
    End If
    If MWUse Then
        If CellValue(WkBS, BSAudLN%, MWCol) <> 0 Then
            setOptState ("MW")
            VeriByWS dbMonthTop&, 276
        End If
    End If
End If
Rem -----
If IsMultiDP Then
    VeriDept
End If
End Sub
Rem =====(Verification after BS/PL Preparation)=====
Sub VeriDept()
    Dim VD(BKCol To RMCol) As Variant
    Dim i%, j%, DPTAudLN%
    DPTAudLN% = getFSRow%(WkPL, UCase(DPTAudCD), CFCOL - 1)
    For i% = BKCol To RMCol
        For j% = 1 To NumDpt
            VD(i%) = VD(i%) + CellValue(WkDpt(j%), DPTAudLN%, i%)
        Next j%
        VD(i%) = CellValue(WkPL, DPTAudLN%, i%) - VD(i%)
    Next i%
    If VD(BKCol) <> 0 Then
        setOptState ("BK")
        VeriByWS dbMonthTop&, 276
    ElseIf VD(JVCOL) <> 0 Then
        setOptState ("JV")
        VeriByWS dbMonthTop&, 276
    ElseIf VD(APCOL) <> 0 Then
        setOptState ("AP")
        VeriByWS dbMonthTop&, 276
    End If
    If ARUse Then
        If VD(ARCol) <> 0 Then
            setOptState ("AR")
            VeriByWS dbMonthTop&, 276
        End If
    End If

```



```
End If
If MWUse then
  if VD(MWCol) <> 0 Then
    setOptState("MW")
    VeriByWS dbMonthTop&, 276
  end if
End If
End Sub
```

```
561 End If
562 If MWUse Then
563   If VD(MWCol) <> 0 Then
564     setOptState ("MW")
565     VeriByWS dbMonthTop&, 276
566   End If
567 End If
568 End Sub
```