

```

Rem =====
Rem A_Accounts and Account attributes:OOoBasic Program (Copyright:Shuichi Sunaga)
Rem (Ver. 2005/10/20:378)(2006/04/06:433)(2006/11/18:488)
Rem integration to ND(2011/09/05:622)Moved DDlist to VD(2011/12/30:351)
Rem Moved ChngAlloc toAX(2012/04/25:270)
Rem Moved Account attributes from VZ(2012/07/02:346)
Rem Adjusted AR portion of ShowJE(2012/08/07:334)
Rem =====
Rem Sub ShowNameFromCD()
Rem Sub ShowAcctName()
Rem Sub PostDAllocate()
Rem Sub ShowJE()
Rem Sub MakeAcctList(ByVal WA As Object, Optional WB As object)
Rem Function FillSpace(ByVal iDigit%)
Rem =====
Option Explicit
Sub ShowNameFromCD()
    IntroTo3D
    ShowAcctName (0)
End Sub
Sub ShowAcctName(sPU%)
    dim actName$, actcode$, AcctIndex%
    If Not StopOrGo("ShowACName") And sPU%=0 then
        MsgTxt$ = "This command is not usable on this sheet."
        MsgTtl$ = "Test of command usability"
        msgbox(MsgTxt$, 16 , MsgTtl$)
    End
End If
MakeAcctList WkBS, WkPL
getPagePlan aSName$, cMonth%
actcode$=CellString(aSheet, SelectionRow&, Col_ActCD%)
If Len(actcode$) = 0 Then
    MsgTxt$ = "Account Code on this row is blank."
    resp% = MsgBox(MsgTxt$, 16, "Blank Code !")
End
End If
Rem -----(Verification accounts)-----
actcode$ = FillSpace(Len(actcode$)) & actcode$
AcctIndex% = IndexInACT(actcode$)
If AcctIndex% > -1 then Rem where WkAcct is registered on BS/PL
    If sPU%=0 Then
        oAWDlg.Title = "Account Code Container"
    else
        oAWDlg.Title = "Account Code Container for addition to BS or PL"
    end if
    oAWAct.setText(oAWAct.Model.StringItemList(AcctIndex%))
    oAWDlg.Model.PositionX = 3
    oAWDlg.Model.PositionY = 3
    oAWDlg.Execute()
Elseif AcctIndex% = -1 Then
    MsgTxt$ = actcode$ & " is not registered neither on BS nor PL." & Chr(13)
    MsgTxt$ = MsgTxt$ & "Use only after registration."
    MsgTtl$ = "Test of Account Code"
    resp% = MsgBox(MsgTxt$, 16, MsgTtl$)
End If
oAWDlg.EndExecute()
End Sub
Sub PostDAllocate()
    dim fRow&, xActCD&
    fRow& = SelectionRow()
    If aSName$ ="BS" Or aSName$ ="PL" Or aSName$ = "CF" Then
        BSPL.Insert(fRow&)
        Exit Sub
    End if
    xActCD&=Val(Left(oAWAct.getText(), ADigit))
    SetCellValue aSheet, fRow&, Col_ActCD%, xActCD&
End Sub
Sub ShowJE()
    dim drAmt as Variant, crAmt as Variant
    dim BKCG as Variant, CXDr as Variant, CXCr as Variant, CXBkCG as Variant

```

```

1 Rem =====
2 Rem A_Accounts and Accountattribute Excel_VBA Program (Copyright:Shuichi Sunaga)
3 Rem
4 Rem Moved ChngAlloc toAX(2012/04/25:270)
5 Rem Moved Account attributes from VZ(2012/07/02:346)
6 Rem Adjusted AR portion of ShowJE(2012/08/07:329)
7 Rem Translation to English(2012/08/09:330)
8 Rem =====
9 Rem Sub ShowNameFromCD()
10 Rem Sub ShowAcctName()
11 Rem Sub PostACDOnly()
12 Rem Sub ShowJE()
13 Rem Sub MakeAcctList(ByVal WA As Object, Optional WB As object)
14 Rem Function FillSpace(ByVal iDigit%)
15 Rem =====
16 Option Explicit
17 Sub ShowNameFromCD()
18     IntroTo3D
19     ShowAcctName (0)
20 End Sub
21 Sub ShowAcctName(sPU%)
22     Dim actname$, actcode$, AcctIndex%
23     If Not StopOrGo("ShowACName") And sPU% = 0 Then
24         MsgTxt$ = "This command is not usable on this sheet."
25         MsgTtl$ = "Test of command usability"
26         resp% = MsgBox(MsgTxt$, vbCritical, MsgTtl$)
27     End
28 End If
29 MakeAcctList WkBS, WkPL
30 getPagePlan aSName$, cMonth%
31 actcode$ = CellString(aSheet, SelectionRow&, Col_ActCD)
32 If Len(actcode$) = 0 Then
33     MsgTxt$ = "Account Code on this row is blank."
34     resp% = MsgBox(MsgTxt$, vbCritical, "Blank Code !")
35 End
36 End If
37 Rem -----(Verification accounts)-----
38 actcode$ = FillSpace(Len(actcode$)) & actcode$
39 AcctIndex% = IndexInACT(actcode$)
40 If AcctIndex% > -1 Then 'where WkAcct is registered on BS/PL
41     If sPU% = 0 Then
42         oAWDlg.Caption = "Account Code Container"
43     Else
44         oAWDlg.Caption = "Account Code Container for addition to BS or PL"
45     End If
46     oAWAct.Value = oAWAct.List(AcctIndex%)
47     oAWDlg.Left = 3
48     oAWDlg.Top = 3
49     oAWDlg.Show
50 Elseif AcctIndex% = -1 Then
51     MsgTxt$ = actcode$ & " is not registered neither on BS nor PL." & Chr(13)
52     MsgTxt$ = MsgTxt$ & "Use only after registration."
53     MsgTtl$ = "Test of Account Code"
54     resp% = MsgBox(MsgTxt$, vbCritical, MsgTtl$)
55 End If
56
57 End Sub
58 Sub PostACDOnly()
59     Dim fRow&, xActCD&
60     fRow& = SelectionRow
61     If aSName$ = "BS" Or aSName$ = "PL" Or aSName$ = "CF" Then
62         BSPL.Insert (fRow&)
63         Exit Sub
64     End If
65     xActCD& = Val(Left(oAWAct.Text, ADigit))
66     SetCellValue aSheet, fRow&, Col_ActCD%, xActCD&
67 End Sub
68 Sub ShowJE()
69     Dim drAmt As Variant, crAmt As Variant, vAmt As Variant
70     Dim BKCG As Variant, CXDr As Variant, CXCr As Variant, CXBkCG As Variant

```

```

dim drAct$, drName$, drLen%, tmpAct$, tmpName$
dim crAct$, crName$, crLen%
dim fRow&, NameL%, AcctIndex%, CRTXTxt$, PYTXTxt$, BKChgTxt$
If StopOrGo("ShowJE") then
    getColArray(aSName$)
Else
    MsgTxt$ = "This command is not usable on this sheet."
    MsgTtl$ = "Test of command usability"
    resp% = MsgBox(MsgTxt$, 16 , MsgTtl$)
End
End If
fRow& = SelectionRow()
Rem =====(DR)=====
drAct$ = CellString(aSheet, fRow&, Col_ActCD%)
drAct$ = FillSpace(Len(drAct$)) & drAct$
AcctIndex% = IndexInACT(drAct$)
If AcctIndex% <= -1 then
    MsgTxt$ = "Unable to present Journal Entry due to no Account Code on this row."
    MsgTtl$ = "Test of Account Code"
    MsgBox(MsgTxt$, 0 , MsgTtl$)
End
End If
drName$ = oAWAct.Model.StringItemList(AcctIndex%)
NameL% = Len(drName$)
drName$ = Trim(Right(drName$, NameL%-Len(drAct$) -6))
drAct$ = drAct$ & Space(1) & drName$ & Space(1) 'Add one charactor worth space to Cr amount
drLen%=Len(drName$)
drAmt= CellValue(aSheet, fRow&, Col_Amount%)
Rem =====(CR)=====
Select Case aSName$
    Case "BK"
        crAct$= CellString(aSheet, fRow&, Col_BKCD%)
    Case "AP"
        crAct$= APCD
    Case "AR"
        crAct$= ARCD
    Case "MW"
        crAct$= MWCD
End Select
crAct$ = FillSpace(Len(crAct$)) & crAct$
AcctIndex% = IndexInACT(crAct$)
crName$ = oAWAct.Model.StringItemList(AcctIndex%)
NameL% = Len(crName$)
crName$ = Trim(Right(crName$, NameL%-Len(crAct$) -6))
crAct$ = crAct$ & Space(1) & CrName$
crLen% = Len(crName$)
crAmt= -drAmt
Rem -----(Adjust string length of Dr and Cr side)-----
If drLen%-crLen%>0 then
    crAct$ = crAct$ & space((drLen%-crLen%)*2)
Else
    drAct$ = drAct$ & space((crLen%-drLen%)*2)
End If
BKCg=CellValue(aSheet, fRow&, Col_BKChg%)
CXDr=CellValue(aSheet, fRow&, Col_CXDr%)
CXCr=CellValue(aSheet, fRow&, Col_CXCr%)
CRTXTxt$ = "#" & CRTXCD & space(1) & "VAT Paid"
PYTXTxt$ = "#" & PYTXCD & space(1) & "VAT Received"
BKChgTxt$ = "#" & BKChgCD & space(1) & "Bank Chages Paid"
If aSName$="AR" Then
    tmpAct$=drAct$
    tmpName$=drName$
    drAct$=crAct$
    drName$=crName$
    crAct$=tmpAct$
    crName$=tmpName$
    drLen%=Len(drName$)
    crLen% = Len(crName$)
    If drLen%-crLen%>0 then
        crAct$ = crAct$ & space((drLen%-crLen%)*2)

```

```

Dim drAct$, drName$, drLen%, tmpAct$, tmpName$
Dim crAct$, crName$, crLen%
Dim fRow&, NameL%, AcctIndex%, CRTXTxt$, PYTXTxt$, BKChgTxt$
If StopOrGo("ShowJE") Then
    getColArray (aSName$)
Else
    MsgTxt$ = "This command is not usable on this sheet."
    MsgTtl$ = "Test of command usability"
    resp% = MsgBox(MsgTxt$, 16, MsgTtl$)
End
End If
fRow& = SelectionRow
Rem =====(DR)=====
drAct$ = CellString(aSheet, fRow&, Col_ActCD%)
drAct$ = FillSpace(Len(drAct$)) & drAct$
AcctIndex% = IndexInACT(drAct$)
If AcctIndex% <= -1 Then
    MsgTxt$ = "Unable to present Journal Entry due to no Account Code on this row."
    MsgTtl$ = "Test of Account Code"
    resp% = MsgBox(MsgTxt$, vbOKOnly, MsgTtl$)
End
End If
drName$ = oAWAct.List(AcctIndex%)
NameL% = Len(drName$)
drName$ = Trim(Right(drName$, NameL% - Len(drAct$) - 6))
drAct$ = drAct$ & Space(1) & drName$ & Space(1) 'Add one charactor worth space to Cr amount
drLen% = Len(drName$)
drAmt = CellValue(aSheet, fRow&, Col_Amount%)
Rem =====(CR)=====
Select Case aSName$
    Case "BK"
        crAct$ = CellString(aSheet, fRow&, Col_BKCD%)
    Case "AP"
        crAct$ = APCD
    Case "AR"
        crAct$ = ARCD
    Case "MW"
        crAct$ = MWCD
End Select
crAct$ = FillSpace(Len(crAct$)) & crAct$
AcctIndex% = IndexInACT(crAct$)
crName$ = oAWAct.List(AcctIndex%)
NameL% = Len(crName$)
crName$ = Trim(Right(crName$, NameL% - Len(crAct$) - 6))
crAct$ = crAct$ & Space(1) & crName$
crLen% = Len(crName$)
crAmt = -drAmt
Rem -----(Adjust string length of Dr and Cr side)-----
If drLen% - crLen% > 0 Then
    crAct$ = crAct$ & Space((drLen% - crLen%) * 2)
Else
    drAct$ = drAct$ & Space((crLen% - drLen%) * 2)
End If
BKCg = CellValue(aSheet, fRow&, Col_BKChg%)
CXDr = CellValue(aSheet, fRow&, Col_CXDr%)
CXCr = CellValue(aSheet, fRow&, Col_CXCr%)
CRTXTxt$ = "#" & CRTXCD & Space(1) & "C_Tax Paid"
PYTXTxt$ = "#" & PYTXCD & Space(1) & "C_Tax Received"
BKChgTxt$ = "#" & BKChgCD & Space(1) & "Bank Chages Paid"
If aSName$ = "AR" Then
    tmpAct$ = drAct$
    tmpName$ = drName$
    drAct$ = crAct$
    drName$ = crName$
    crAct$ = tmpAct$
    crName$ = tmpName$
    drLen% = Len(drName$)
    crLen% = Len(crName$)
    If drLen% - crLen% > 0 Then
        crAct$ = crAct$ & Space((drLen% - crLen%) * 2)

```

```

Else
    drAct$ = drAct$ & space((crLen%-drLen%)*2)
End If
End If
If BkCg=0 then
    if CXCr=0 then
        If CXDr=0 then
            drAmt= " \" & Format(drAmt,"#,##0 ;(##,##0)")
            crAmt= " \" & Format(crAmt,"#,##0 ;(##,##0)")
            MsgTxt$ = "#" & drAct & drAmt & Chr(13) & "#" & crAct & crAmt
        Else
            drAmt= " \" & Format(drAmt,"#,##0 ;(##,##0)")
            crAmt= " \" & Format(crAmt-CXDr,"#,##0 ;(##,##0)")
            MsgTxt$ = "#" & drAct & Space(10) & drAmt & Chr(13)
            MsgTxt$ = MsgTxt$ & CRTXTxt$
            MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(CXDr)) & Format(CXDr,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & Chr(13) & "#" & crAct & crAmt
        End If
    else
        drAmt= " \" & Format(drAmt+CXCr,"#,##0 ;(##,##0)")
        crAmt= " \" & Format(crAmt,"#,##0 ;(##,##0)")
        MsgTxt$ = "#" & drAct & drAmt & Chr(13) & "#" & crAct & crAmt & Chr(13)
        MsgTxt$ = MsgTxt$ & PYTXXt$
        MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(CXCr)-1) & Format(-CXCr,"#,##0 ;(##,##0)")
    end if
Else
    CXBkCg = Int(BkCg/105*5)
    BkCg = BkCg - CXBkCg
    if CXCr=0 then
        If CXDr=0 then
            drAmt= " \" & Format(drAmt, " ,##0 ;(##,##0)")
            crAmt= " \" & Format(crAmt-BkCg-CXBkCg, " ,##0 ;(##,##0)")
            MsgTxt$ = "#" & drAct & drAmt & Chr(13)
            MsgTxt$ = MsgTxt$ & BKChgTxt$
            MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(BKCG)) & Format(BkCg,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & CRTXTxt$
            MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(CXBkCg)) & Format(CXBkCg,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & "#" & crAct & crAmt
        Else
            drAmt= " \" & Format(drAmt, " ,##0 ;(##,##0)")
            crAmt= " \" & Format(crAmt+BkCg+CXDr, " ,##0 ;(##,##0)")
            CXDr = CXDr + CXBkCg
            MsgTxt$ = "#" & drAct & drAmt & Chr(13)
            MsgTxt$ = MsgTxt$ & BKChgTxt$
            MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(BKCG)) & Format(BkCg,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & CRTXTxt$
            MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(CXDr)) & Format(CXDr,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & "#" & crAct & crAmt
        End If
    else
        drAmt= " \" & Format(vAmt+CXCr, " ,##0 ;(##,##0)")
        crAmt= " \" & Format(vAmt+BkCg, " ,##0 ;(##,##0)")
        MsgTxt$ = "#" & drAct & drAmt & Chr(13)
        MsgTxt$ = MsgTxt$ & BKChgTxt$
        MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(BKCG)) & Format(BkCg,"#,##0 ;(##,##0)") & (
        MsgTxt$ = MsgTxt$ & CRTXTxt$
        MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(CXBkCg)) & Format(CXBkCg,"#,##0 ;(##,##0)")
        MsgTxt$ = MsgTxt$ & "#" & crAct & crAmt & Chr(13)
        MsgTxt$ = MsgTxt$ & PYTXXt$
        MsgTxt$ = MsgTxt$ & space((crLen-5)*2+Len(crAmt)-Len(CXCr)-1) & Format(-CXCr,"#,##0 ;(##,##0)")
    end if
End If
MsgTtl$ = "Journal Entry in Dr/Cr mode"
resp% = MsgBox(MsgTxt$, 0, MsgTtl$)
End Sub
Sub MakeAcctList(ByVal WA As Object, Optional WB As object)
    dim actcode$, taxcode$, actname$, al$, actItem$
    dim BPRow&, i%
    Set oAWDIg = CreateUnoDialog(DialogLibraries.OOo3D.AWLlist)
    Set oAWAct = oAWDIg.GetControl("ACTList")

```

```

Else
    drAct$ = drAct$ & Space((crLen% - drLen%) * 2)
End If
End If
If BKCg = 0 Then
    If CXCr = 0 Then
        If CXDr = 0 Then
            drAmt = " \" & Format(drAmt, " ,##0 ;(##,##0)")
            crAmt = " \" & Format(crAmt, " ,##0 ;(##,##0)")
            MsgTxt$ = "#" & drAct & drAmt & Chr(13) & "#" & crAct & crAmt
        Else
            drAmt = " \" & Format(drAmt, " ,##0 ;(##,##0)")
            crAmt = " \" & Format(crAmt - CXDr, " ,##0 ;(##,##0)")
            MsgTxt$ = "#" & drAct & Space(10) & drAmt & Chr(13)
            MsgTxt$ = MsgTxt$ & CRTXTxt$
            MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(CXDr)) & Format(CXDr,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & Chr(13) & "#" & crAct & crAmt
        End If
    Else
        drAmt = " \" & Format(drAmt + CXCr, " ,##0 ;(##,##0)")
        crAmt = " \" & Format(crAmt, " ,##0 ;(##,##0)")
        MsgTxt$ = "#" & drAct & drAmt & Chr(13) & "#" & crAct & crAmt & Chr(13)
        MsgTxt$ = MsgTxt$ & PYTXXt$
        MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(CXCr) - 1) & Format(-CXCr,"#,##0 ;(##,##0)")
    End If
Else
    CXBkCg = Int(BKCg / 105 * 5)
    BKCg = BKCg - CXBkCg
    If CXCr = 0 Then
        If CXDr = 0 Then
            drAmt = " \" & Format(drAmt, " ,##0 ;(##,##0)")
            crAmt = " \" & Format(crAmt - BKCg - CXBkCg, " ,##0 ;(##,##0)")
            MsgTxt$ = "#" & drAct & drAmt & Chr(13)
            MsgTxt$ = MsgTxt$ & BKChgTxt$
            MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(BKCG)) & Format(BKCG,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & CRTXTxt$
            MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(CXBkCg)) & Format(CXBkCg,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & "#" & crAct & crAmt
        Else
            drAmt = " \" & Format(drAmt, " ,##0 ;(##,##0)")
            crAmt = " \" & Format(crAmt + BKCg + CXDr, " ,##0 ;(##,##0)")
            CXDr = CXDr + CXBkCg
            MsgTxt$ = "#" & drAct & drAmt & Chr(13)
            MsgTxt$ = MsgTxt$ & BKChgTxt$
            MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(BKCG)) & Format(BKCG,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & CRTXTxt$
            MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(CXDr)) & Format(CXDr,"#,##0 ;(##,##0)")
            MsgTxt$ = MsgTxt$ & "#" & crAct & crAmt
        End If
    Else
        drAmt = " \" & Format(vAmt + CXCr, " ,##0 ;(##,##0)")
        crAmt = " \" & Format(vAmt + BKCg, " ,##0 ;(##,##0)")
        MsgTxt$ = "#" & drAct & drAmt & Chr(13)
        MsgTxt$ = MsgTxt$ & BKChgTxt$
        MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(BKCG)) & Format(BKCG,"#,##0 ;(##,##0)")
        MsgTxt$ = MsgTxt$ & CRTXTxt$
        MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(CXBkCg)) & Format(CXBkCg,"#,##0 ;(##,##0)")
        MsgTxt$ = MsgTxt$ & "#" & crAct & crAmt & Chr(13)
        MsgTxt$ = MsgTxt$ & PYTXXt$
        MsgTxt$ = MsgTxt$ & Space((crLen - 5) * 2 + Len(crAmt) - Len(CXCr) - 1) & Format(-CXCr,"#,##0 ;(##,##0)")
    End If
End If
MsgTtl$ = "Journal Entry in Dr/Cr mode"
resp% = MsgBox(MsgTxt$, 0, MsgTtl$)
End Sub
Sub MakeAcctList(ByVal WA As Object, Optional WB As Object)
    Dim actcode$, taxcode$, actname$, al$, actItem$
    Dim BPRow&
    Set oAWDIg = AWList
    Set oAWAct = oAWDIg.ACTList

```

```

oAWAct.removeItems(0, oAWAct.ItemCount)
BPRow& = 5
i% = 0
Do While true
    actcode$ = trim(CellString(WA, BPRow&, 2))
    taxcode$ = trim(CellString(WA, BPRow&, 3))
    actname$ = trim(CellString(WA, BPRow&, 4))
    if actcode$ = "ENDLN" or BPRow& > 500 then
        Exit Do
    elseif actcode$ = "U_Border" Or actcode$ = "L_Border" Then
        'Do nothing !
    else
        If Len(actcode$) > 0 Then
            aL$ = FillSpace(Len(actcode$)) & actcode$
            actItem$ = aL$ & " |" & taxcode$ & "|" " & actname$
            oAWAct.AddItem(actItem$,i%)
            i% = i% + 1
        End If
    end if
    BPRow& = BPRow& + 1
Loop
If Not IsMissing(WB) Then
    BPRow& = 5
    Do While true
        actcode$ = trim(CellString(WkPL, BPRow&, 2))
        taxcode$ = trim(CellString(WkPL, BPRow&, 3))
        actname$ = trim(CellString(WkPL, BPRow&, 4))
        if actcode$ = "ENDLN" or BPRow& > 500 then
            Exit Do
        elseif actcode$ = "U_Border" Or actcode$ = "L_Border" Then
            'Do nothing !
        else
            If Len(actcode$) > 0 Then
                aL$ = FillSpace(Len(actcode$)) & actcode$
                actItem$ = aL$ & " |" & taxcode$ & "|" " & actname$
                oAWAct.AddItem(actItem$,i%)
                i% = i% + 1
            End If
        end if
        BPRow& = BPRow& + 1
    Loop
End If
End Sub
Function FillSpace(ByVal iDigit%)
    dim i%
    FillSpace=""
    For i% = 1 to Adigit - iDigit%
        FillSpace=FillSpace & chr$(32)
    Next i%
End Function
Rem ===== User Specific Data =====
Rem To Determine if the Account belongs to BS=True | nonBS=False
Function A_VBSCD(WkAcct) As Boolean
    If Val(WkAcct) < CellValue(WkCL, 21, 3) Then
        A_VBSCD = True
    Else
        A_VBSCD = False
    End If
End Function
Rem To Determine if the Account belongs to PL=True | nonPL=False
Function A_VPLCD(WkAcct) As Boolean
    If Val(WkAcct) >= CellValue(WkCL, 22, 2) Then
        A_VPLCD = True
    Else
        A_VPLCD = False
    End If
End Function
Rem To Determine if the Account appears on BS=True | no=False
Function WS_BS(WkAcct) As Boolean
    If Val(WkAcct) < CellValue(WkCL, 25, 3) Then

```

```

211 oAWAct.Clear
212 BPRow& = 5
213
214 Do While True
215     actcode$ = Trim(CellString(WA, BPRow&, 2))
216     taxcode$ = Trim(CellString(WA, BPRow&, 3))
217     actname$ = Trim(CellString(WA, BPRow&, 4))
218     If actcode$ = "ENDLN" Or BPRow& > 150 Then
219         Exit Do
220     ElseIf actcode$ = "U_Border" Or actcode$ = "L_Border" Then
221         'Do nothing !
222     Else
223         If Len(actcode$) > 0 Then
224             aL$ = FillSpace(Len(actcode$)) & actcode$
225             actItem$ = aL$ & " |" & taxcode$ & "|" " & actname$
226             oAWAct.AddItem actItem$
227         End If
228     End If
229     BPRow& = BPRow& + 1
230 Loop
231 If Not IsMissing(WB) Then
232     BPRow& = 5
233     Do While True
234         actcode$ = Trim(CellString(WkPL, BPRow&, 2))
235         taxcode$ = Trim(CellString(WkPL, BPRow&, 3))
236         actname$ = Trim(CellString(WkPL, BPRow&, 4))
237         If actcode$ = "ENDLN" Or BPRow& > 150 Then
238             Exit Do
239         ElseIf actcode$ = "U_Border" Or actcode$ = "L_Border" Then
240             'Do nothing !
241         Else
242             If Len(actcode$) > 0 Then
243                 aL$ = FillSpace(Len(actcode$)) & actcode$
244                 actItem$ = aL$ & " |" & taxcode$ & "|" " & actname$
245                 oAWAct.AddItem actItem$
246             End If
247         End If
248         BPRow& = BPRow& + 1
249     Loop
250 End If
251 End Sub
252 Function FillSpace(ByVal iDigit%)
253     Dim i%
254     FillSpace = ""
255     For i% = 1 To Adigit - iDigit%
256         FillSpace = FillSpace & Chr$(32)
257     Next i%
258 End Function
259 Rem ===== User Specific Data =====
260 Rem To Determine if the Account belongs to BS=True | nonBS=False
261 Function A_VBSCD(WkAcct) As Boolean
262     If Val(WkAcct) < CellValue(WkCL, 21, 3) Then
263         A_VBSCD = True
264     Else
265         A_VBSCD = False
266     End If
267 End Function
268 Rem To Determine if the Account belongs to PL=True | nonPL=False
269 Function A_VPLCD(WkAcct) As Boolean
270     If Val(WkAcct) >= CellValue(WkCL, 22, 2) Then
271         A_VPLCD = True
272     Else
273         A_VPLCD = False
274     End If
275 End Function
276 Rem To Determine if the Account appears on BS=True | no=False
277 Function WS_BS(WkAcct) As Boolean
278     If Val(WkAcct) < CellValue(WkCL, 25, 3) Then

```

```

        WS_BS = True
    Elseif Val(WkAcct) >= CellValue(WkCL, 25, 2) Then
        WS_BS = True
    Else
        WS_BS = False
    End If
End Function
Rem To Determine if the Account requires apportionment =True | no=False
Function A_VALlocCD(WkAcct) As Boolean
    If (Val(WkAcct) >= CellValue(WkCL, 26, 2) And Val(WkAcct) < CellValue(WkCL, 26, 3)) Then
        A_VALlocCD = True
    Else
        A_VALlocCD = False
    End If
End Function
Rem Determine if the account code is Not Greater Than those for apportionment
Function NGTAllocCD(WkAcct) As Boolean
    If (Val(WkAcct) >= CellValue(WkCL, 22, 2) And Val(WkAcct) < CellValue(WkCL, 26, 3)) Then
        NGTAllocCD = True
    Else
        NGTAllocCD = False
    End If
End Function
Rem To Determine if the Account belongs to Revenues =True | no=False
Function A_VSLsCD(WkAcct) As Boolean
    If (Val(WkAcct) >= CellValue(WkCL, 22, 2) And Val(WkAcct) < CellValue(WkCL, 22, 3)) Then
        A_VSLsCD = True
    Else
        A_VSLsCD = False
    End If
End Function
Rem To Determine if the Account belongs to Cost of Revenues =True | no=False
Function A_VCstCD(WkAcct) As Boolean
    If (Val(WkAcct) >= CellValue(WkCL, 23, 2) And Val(WkAcct) < CellValue(WkCL, 23, 3)) Then
        A_VCstCD = True
    Else
        A_VCstCD = False
    End If
End Function
Rem To Determine if the Account is included in SvC report =True | no=False
Function A_VSvCCD(WkAcct) As Boolean
    If (Val(WkAcct) >= CellValue(WkCL, 27, 2) And Val(WkAcct) < CellValue(WkCL, 27, 3)) Then
        A_VSvCCD = True
    Else
        A_VSvCCD = False
    End If
End Function
Rem To Determine if the Account belongs to non-operating revenues & exp. =True | no=False
Function A_VEOPX(WkAcct) As Boolean
    If Val(WkAcct) >= CellValue(WkCL, 24, 2) Then
        A_VEOPX = True
    Else
        A_VEOPX = False
    End If
End Function
<!-- End of Page -->

```

```

281         WS_BS = True
282     Elseif Val(WkAcct) >= CellValue(WkCL, 25, 2) Then
283         WS_BS = True
284     Else
285         WS_BS = False
286     End If
287 End Function
288 Rem To Determine if the Account requires apportionment =True | no=False
289 Function A_VALlocCD(WkAcct) As Boolean
290     If (Val(WkAcct) >= CellValue(WkCL, 26, 2) And Val(WkAcct) < CellValue(WkCL, 26, 3)) Then
291         A_VALlocCD = True
292     Else
293         A_VALlocCD = False
294     End If
295 End Function
296 Rem Determine if the account code is Not Greater Than those for apportionment
297 Function NGTAllocCD(WkAcct) As Boolean
298     If (Val(WkAcct) >= CellValue(WkCL, 22, 2) And Val(WkAcct) < CellValue(WkCL, 26, 3)) Then
299         NGTAllocCD = True
300     Else
301         NGTAllocCD = False
302     End If
303 End Function
304 Rem To Determine if the Account belongs to Revenues =True | no=False
305 Function A_VSLsCD(WkAcct) As Boolean
306     If (Val(WkAcct) >= CellValue(WkCL, 22, 2) And Val(WkAcct) < CellValue(WkCL, 22, 3)) Then
307         A_VSLsCD = True
308     Else
309         A_VSLsCD = False
310     End If
311 End Function
312 Rem To Determine if the Account belongs to Cost of Revenues =True | no=False
313 Function A_VCstCD(WkAcct) As Boolean
314     If (Val(WkAcct) >= CellValue(WkCL, 23, 2) And Val(WkAcct) < CellValue(WkCL, 23, 3)) Then
315         A_VCstCD = True
316     Else
317         A_VCstCD = False
318     End If
319 End Function
320 Rem To Determine if the Account is included in SvC report =True | no=False
321 Function A_VSvCCD(WkAcct) As Boolean
322     If (Val(WkAcct) >= CellValue(WkCL, 27, 2) And Val(WkAcct) < CellValue(WkCL, 27, 3)) Then
323         A_VSvCCD = True
324     Else
325         A_VSvCCD = False
326     End If
327 End Function
328 Rem To Determine if the Account belongs to non-operating revenues & exp. =True | no=False
329 Function A_VEOPX(WkAcct) As Boolean
330     If Val(WkAcct) >= CellValue(WkCL, 24, 2) Then
331         A_VEOPX = True
332     Else
333         A_VEOPX = False
334     End If
335 End Function
<!-- End of Page -->

```