

```

Sub C4DATACOPY_SelectedAccounts(pov_base)
' This subroutine copies all of the base, local currency data from DefaultRate into the
various Custom4 members
' that have been set to "1" in the trigger accounts, in the [None] entity.

' Limit this to run only in base entities
If pov_base = True Then

' Step 1: Get the amount in the current period in each of the trigger accounts.
' Note these accounts are configured as Balance Recurring which means that if any prior
' period has been set as "1", the current period will also be "1", until someone
changes it to zero.
' Balance Recurring type accounts do not carry into the next year.
CURPLANtrigger = HS.GetCell("A#CURPLAN_Trigger.E#[None].V#[None]" & AllNones)
CURACTUALtrigger = HS.GetCell("A#CURACTUAL_Trigger.E#[None].V#[None]" & AllNones)
CURFCSTtrigger = HS.GetCell("A#CURFCST_Trigger.E#[None].V#[None]" & AllNones)
CURFCST2_10trigger = HS.GetCell("A#CURFCST2_10_Trigger.E#[None].V#[None]" & AllNones)
CURFCST5_7trigger = HS.GetCell("A#CURFCST5_7_Trigger.E#[None].V#[None]" & AllNones)
CURFCST8_4trigger = HS.GetCell("A#CURFCST8_4_Trigger.E#[None].V#[None]" & AllNones)
CURFCST10_2trigger = HS.GetCell("A#CURFCST10_2_Trigger.E#[None].V#[None]" & AllNones)
CURFCST11_1trigger = HS.GetCell("A#CURFCST11_1_Trigger.E#[None].V#[None]" & AllNones)
CURFCSTRVSD3_9trigger = HS.GetCell("A#CURFCSTRVSD3_9_Trigger.E#[None].V#[None]" &
AllNones)
CURFCSTRVSD6_6trigger = HS.GetCell("A#CURFCSTRVSD6_6_Trigger.E#[None].V#[None]" &
AllNones)
CURFCSTRVSD9_3trigger = HS.GetCell("A#CURFCSTRVSD9_3_Trigger.E#[None].V#[None]" &
AllNones)
PRACTUALtrigger = HS.GetCell("A#PRACTUAL_Trigger.E#[None].V#[None]" & AllNones)
PRPLANtrigger = HS.GetCell("A#PRPLAN_Trigger.E#[None].V#[None]" & AllNones)
NEXTPLANtrigger = HS.GetCell("A#NEXTPLAN_Trigger.E#[None].V#[None]" & AllNones)
NEXTACTUALtrigger = HS.GetCell("A#NEXTACTUAL_Trigger.E#[None].V#[None]" & AllNones)
NEXTFCSTtrigger = HS.GetCell("A#NEXTFCST_Trigger.E#[None].V#[None]" & AllNones)
NEXTFCST2_10trigger = HS.GetCell("A#NEXTFCST2_10_Trigger.E#[None].V#[None]" & AllNones)
NEXTFCST5_7trigger = HS.GetCell("A#NEXTFCST5_7_Trigger.E#[None].V#[None]" & AllNones)
NEXTFCST8_4trigger = HS.GetCell("A#NEXTFCST8_4_Trigger.E#[None].V#[None]" & AllNones)
NEXTFCST10_2trigger = HS.GetCell("A#NEXTFCST10_2_Trigger.E#[None].V#[None]" & AllNones)
NEXTFCST11_1trigger = HS.GetCell("A#NEXTFCST11_1_Trigger.E#[None].V#[None]" & AllNones)
NEXTFCSTRVSD3_9trigger = HS.GetCell("A#NEXTFCSTRVSD3_9_Trigger.E#[None].V#[None]" &
AllNones)
NEXTFCSTRVSD6_6trigger = HS.GetCell("A#NEXTFCSTRVSD6_6_Trigger.E#[None].V#[None]" &
AllNones)
NEXTFCSTRVSD9_3trigger = HS.GetCell("A#NEXTFCSTRVSD9_3_Trigger.E#[None].V#[None]" &
AllNones)
NEXTPLAN2trigger = HS.GetCell("A#NEXTFCST_Trigger.E#[None].V#[None]" & AllNones)
FXRevaltrigger = HS.GetCell("A#FXReval_Trigger.E#[None].V#[None]" & AllNones)
FXREVAL2trigger = HS.GetCell("A#FXREVAL2_Trigger.E#[None].V#[None]" & AllNones)
STRATPLANtrigger = HS.GetCell("A#STRATPLAN_Trigger.E#[None].V#[None]" & AllNones)

' Step 2: clear out all the target Custom4 members first, to remove any residual data.
' This also addresses the situation where the trigger has been later cleared or set to
any value other than "1"
HS.Clear "C4#CURPLAN"
HS.Clear "C4#CURACTUAL"
HS.Clear "C4#CURFCST"
HS.Clear "C4#CURFCST2_10"
HS.Clear "C4#CURFCST5_7"
HS.Clear "C4#CURFCST8_4"
HS.Clear "C4#CURFCST10_2"
HS.Clear "C4#CURFCST11_1"
HS.Clear "C4#CURFCSTRVSD3_9"
HS.Clear "C4#CURFCSTRVSD6_6"
HS.Clear "C4#CURFCSTRVSD9_3"
HS.Clear "C4#PRACTUAL"
HS.Clear "C4#PRPLAN"
HS.Clear "C4#NEXTPLAN"
HS.Clear "C4#NEXTACTUAL"

```

```

HS.Clear "C4#NEXTFCST"
HS.Clear "C4#NEXTFCST2_10"
HS.Clear "C4#NEXTFCST5_7"
HS.Clear "C4#NEXTFCST8_4"
HS.Clear "C4#NEXTFCST10_2"
HS.Clear "C4#NEXTFCST11_1"
HS.Clear "C4#NEXTFCSTRVSD3_9"
HS.Clear "C4#NEXTFCSTRVSD6_6"
HS.Clear "C4#NEXTFCSTRVSD9_3"
HS.Clear "C4#NEXTPLAN2"
HS.Clear "C4#FXReval"
HS.Clear "C4#FXReval2"
HS.Clear "C4#StratPlan"

```

'Step 3: for each of the trigger accounts that has "1" in it, set the respective C4 member equal to the source member DefaultRate.

```

'Step 3A: define lists of accounts so we can copy selective data
aINCOMEACCTLIST = HS.Account.List("Income", "[Base]")
aSTDPROFITACCTLIST = HS.Account.List("StdProfitDetail", "[Base]")
aNiATTRIBLIST = HS.Account.List("NI_Attrib_SH", "[Base]")
aBALSHEETACCTLIST = HS.Account.List("BalanceSheet", "[Base]")
aROLLPPEACCTLIST = HS.Account.List("Roll_PPENET", "[Base]")
'Create listings of base level accounts below selected account dimension members

```

```

'Step 3B: fetch all of the data residing in C4#DefaultRate
Set DefaultRateData = HS.OpenDataUnit("I#[ICP
Top].C1#TopC1.C2#TopC2.C3#TopC3.C4#DefaultRate")

```

```

'identifies the number of data records in the sub-cube (array table)
NumItems = DefaultRateData.GetNumItems

```

```

'Set up a For loop to read each data record in the sub-cube (array table)
For i = 0 To NumItems - 1

```

```

'Get the next item (record) from the dataunit (array table) and store it in
variables sAcct, etc.

```

```

Call DefaultRateData.GetItem(i, sAcct, sICP, sC1, sC2, sC3, sC4, dData)
'InList is a custom function. The syntax presents a member name, sAcct in this
case, and a list name aINCOMEACCTLIST, and asks whether the account is in the
list

```

```

If InList(sAcct,aINCOMEACCTLIST) Or InList(sAcct,aSTDPROFITACCTLIST) Or
InList(sAcct,aNiATTRIBLIST) Or InList(sAcct,aBALSHEETACCTLIST) Or
InList(sAcct,aROLLPPEACCTLIST) Then

```

```

    If CURPLANtrigger = 1 Then
        HS.EXP "A#" & sAcct & ".C4#CURPLAN = A#" & sAcct & ".C4#DefaultRate"
    End If

```

```

    If CURACTUALtrigger = 1 Then
        HS.EXP "A#" & sAcct & ".C4#CURACTUAL = A#" & sAcct & ".C4#DefaultRate"
    End If

```

```

    If CURFCSTtrigger = 1 Then
        HS.EXP "A#" & sAcct & ".C4#CURFCST = A#" & sAcct & ".C4#DefaultRate"
    End If

```

```

    If CURFCST2_10trigger = 1 Then
        HS.EXP "A#" & sAcct & ".C4#CURFCST2_10 = A#" & sAcct & ".C4#DefaultRate"
    End If

```

```

    If CURFCST5_7trigger = 1 Then
        HS.EXP "A#" & sAcct & ".C4#CURFCST5_7 = A#" & sAcct & ".C4#DefaultRate"
    End If

```

```

    If CURFCST8_4trigger = 1 Then
        HS.EXP "A#" & sAcct & ".C4#CURFCST8_4 = A#" & sAcct & ".C4#DefaultRate"
    End If

```

```

    If CURFCST10_2trigger = 1 Then
        HS.EXP "A#" & sAcct & ".C4#CURFCST10_2 = A#" & sAcct & ".C4#DefaultRate"
    End If

```

```

    If CURFCST11_1trigger = 1 Then

```

```

HS.EXP "A#" & sAcct & ".C4#CURFCST11_1 = A#" & sAcct & ".C4#DefaultRate"
End If
If CURFCSTRVSD3_9trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#CURFCSTRVSD3_9 = A#" & sAcct &
  ".C4#DefaultRate"
End If
If CURFCSTRVSD6_6trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#CURFCSTRVSD6_6 = A#" & sAcct &
  ".C4#DefaultRate"
End If
If CURFCSTRVSD9_3trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#CURFCSTRVSD9_3 = A#" & sAcct &
  ".C4#DefaultRate"
End If
If PRACTUALtrigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#PRACTUAL = A#" & sAcct & ".C4#DefaultRate"
End If
If PRPLANtrigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#PRPLAN = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTPLANtrigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTPLAN = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTACTUALtrigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTACTUAL = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTFCSTtrigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCST = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTFCST2_10trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCST2_10 = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTFCST5_7trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCST5_7 = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTFCST8_4trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCST8_4 = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTFCST10_2trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCST10_2 = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTFCST11_1trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCST11_1 = A#" & sAcct & ".C4#DefaultRate"
End If
If NEXTFCSTRVSD3_9trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCSTRVSD3_9 = A#" & sAcct &
  ".C4#DefaultRate"
End If
If NEXTFCSTRVSD6_6trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCSTRVSD6_6 = A#" & sAcct &
  ".C4#DefaultRate"
End If
If NEXTFCSTRVSD9_3trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTFCSTRVSD9_3 = A#" & sAcct &
  ".C4#DefaultRate"
End If
If NEXTPLAN2trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#NEXTPLAN2 = A#" & sAcct & ".C4#DefaultRate"
End If
If FXRevaltrigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#FXReval = A#" & sAcct & ".C4#DefaultRate"
End If
If FXReval2trigger = 1 Then
  HS.EXP "A#" & sAcct & ".C4#FXReval2 = A#" & sAcct & ".C4#DefaultRate"
End If
If StratPlantrigger = 1 Then

```

```

                HS.EXP "A#" & sAcct & ".C4#StratPlan = A#" & sAcct & ".C4#DefaultRate"
            End If
        End If 'InList(s)
    End If 'pov_base

End Sub 'C4DATACOPY_SelectedAccounts

'-----
'-----
' This is a custom function. Usage is InList(MemberName, ListName) and this returns a
True/False if the member is in the list or not.
' The variable SearchList is the member name. The variable SearchValue is the list name.
Function InList(ByVal SearchList, ByVal SearchValue)
    'Initialize the variable by making it false by default
    InList = False
    'Cycle through the members of the list
    For SearchLoop = LBound(SearchList) To UBound(SearchList)
        'If the member is found in the list, we're done and the boolean InList = True
        If StrComp(SearchList(SearchLoop), SearchValue, vbTextCompare) = 0 Then
            InList = True
            Exit For 'We're done and the answer is True
        End If 'StrComp
    Next 'member in the list
    'If we exhausted the list and didn't find the member in it, InList = False
End Function 'InList

```