PawCom Application Note 105

Changing PhaseIDs

www.MultiwareSoftware.com

Revised 4/14/04

PhaseID values are stored in three places: Phases.dat, JrnlRow.dat, and Ticket.dat. In order to modify a PhaseID you must search for it in all three places and make the necessary changes. The routine below illustrates how to change PhaseIDs using a subroutine in Ms Access. The routine assumes you have created a table "aa PhaseID Changes", which contains the fields OldPhaseID, NewPhaseID, and Merge.

Peachtree has defined the indices for the Phase data so that you cannot simply change the PhaseID in the Phases.dat table. Instead you need to save the data, delete the original record, then create a new record.

You must be careful ordering the PhaseID changes in the table. For example, suppose you want to make the following changes:

Old	New
------	------
Phase1	Phase2
Phase2	Phase3

If you list the changes in this order the first change will fail because Phase2 already exists (it hasn't been changed yet). Therefore you need to reverse the order of these two entries so that a conflict doesn't occur.

The Merge flag indicates those records where you want to merge the data for the OldPhaseID with data for an existing PhaseID. This situation is a bit more complex because the JobEstimate data for the two phases needs to be combined. In order to assist in merging data two queries are used. The SQL for the queries is shown below.

```
Private Sub UpdatePhaseIDs()

    Dim oJobEst  As New PAW.JobEstimate
    Dim oJrnlRow As New PAW.JrnlRow
    Dim oPhase   As New PAW.Phase
    Dim oTicket  As New PAW.Ticket

    Dim DB As DAO.Database
    Dim RS As DAO.Recordset

    Dim OldPhaseID() As String
    Dim NewPhaseID() As String
    Dim Merge()      As Boolean

    Dim i            As Integer
    Dim myPhaseID    As String
    Dim myIndex      As Long
    Dim Description  As String
    Dim CostType     As String
    Dim UseCostCodes As Boolean
    Dim Inactive     As Boolean
    Dim N            As Integer
    Dim Status       As Integer

' Load PhaseID data from local table to arrays to make processing faster

    DoCmd.Hourglass True
    Set DB = CurrentDb
    Set RS = DB.OpenRecordset("aa PhaseID Changes")

    RS.MoveLast
    N = RS.RecordCount
    RS.MoveFirst
    ReDim OldPhaseID(N)
    ReDim NewPhaseID(N)
    ReDim Merge(N)

    For i = 1 To N
        OldPhaseID(i) = RS!OldPhaseID
        NewPhaseID(i) = RS!NewPhaseID
        Merge(i) = RS!Merge
        RS.MoveNext
    Next i

    RS.Close
    DB.Close
    Set RS = Nothing
    Set DB = Nothing

' Read the old Job Estimates and Phase Codes into Access tables

    mwReadAll_JobEstimate
    mwReadAll_PhaseCode

' Update the Phases.dat data

    BtrieveConnect
    Status = oPhase.OpenFile
    For i = 1 To N
        Status = oPhase.StepFirst
        Do Until Status <> btStatusOK
            myPhaseID = oPhase.PhaseID
            If OldPhaseID(i) = myPhaseID Then
                ' Peachtree has defined the data such that the PhaseID cannot
                ' be simply changed.  What we have to do instead is save the
                ' record data, delete the record, and create a new record.
                myIndex = oPhase.Index
                Description = oPhase.Description
                CostType = oPhase.CostType
                UseCostCodes = oPhase.UseCostCodes
                Inactive = oPhase.Inactive
                Status = oPhase.Delete
                If Status = btStatusOK Then
                    oPhase.PhaseID = NewPhaseID(i)
                    oPhase.Index = myIndex
                    oPhase.Description = Description
                    oPhase.UseCostCodes = UseCostCodes
                    oPhase.Inactive = Inactive
                    Status = oPhase.Insert
                    If Status <> btStatusOK And Not Merge(i) Then
                        MsgBox "PHASE.DAT: Insert of new record failed: Status = " & Status &
vbCrLf & _
                                "Old PhaseID = " & OldPhaseID(i) & vbCrLf & _
                                "New PhaseID = " & NewPhaseID(i)
                    End If
                Else
                    MsgBox "PHASE.DAT: Delete of old PhaseID " & OldPhaseID(i) & " failed: Status
= " & Status
                End If
            End If
            Status = oPhase.StepNext
        Loop
    Next i

    Status = oPhase.CloseFile
    Set oPhase = Nothing

' Update the Ticket.dat data

    Status = oTicket.OpenFile
    Status = oTicket.StepFirst
    Do Until Status <> btStatusOK
        myPhaseID = oTicket.PhaseID
        For i = 1 To N
            If OldPhaseID(i) = myPhaseID Then
                oTicket.PhaseID = NewPhaseID(i)
                Status = oTicket.Update
                If Status <> btStatusOK Then
                    MsgBox "TICKET.DAT: Update failed changing " & OldPhaseID(i) & " to " &
NewPhaseID(i)
                End If
                Exit For
            End If
        Next i
        If oTicket.CostID <> "" Then
            oTicket.CostID = ""
            Status = oTicket.Update
        End If
        Status = oTicket.StepNext
    Loop
    Status = oTicket.CloseFile
    Set oTicket = Nothing

' Fix Job Estimate records.
' All records are read into the Access table, the records are removed
' from Peachtree, and new records are inserted from the data in
' query "aa qry Job Estimate rollup 1", which rolls up the values
' of NumberOfUnits, Revenues, and Expenses for all Cost Codes that are
' not merge records. For the merge records data is obtained from the
' query "aa qry Job Estimate rollup 2".

    Me.txtStatus = "Rebuilding Job Estimate records"
    DoEvents
    mwReadAll_JobEstimate
    Status = oJobEst.OpenFile
    Status = oJobEst.StepFirst
    Do Until Status <> btStatusOK
        Status = oJobEst.Delete
        Status = oJobEst.StepNext
    Loop
    Set DB = CurrentDb
    Set RS = DB.OpenRecordset("aa qry PhaseID rollup 1")
    Do Until RS.EOF
        oJobEst.JobIndex = RS!JobIndex
        oJobEst.PhaseIndex = RS!PhaseIndex
        oJobEst.CostIndex = 0
        oJobEst.NumberOfUnits = RS!NumberOfUnits
        oJobEst.Revenues = RS!Revenues
        oJobEst.Expenses = RS!Expenses
        Status = oJobEst.Insert
        RS.MoveNext
    Loop
    RS.Close
    Set RS = DB.OpenRecordset("aa qry PhaseID rollup 2")
    Status = oPhase.OpenFile
    Do Until RS.EOF
        oJobEst.JobIndex = RS!JobIndex
        myPhaseID = RS!NewPhaseID
        Status = oPhase.GetEqual(myPhaseID)
        oJobEst.PhaseIndex = oPhase.Index
        oJobEst.CostIndex = 0
        oJobEst.NumberOfUnits = RS!NumberOfUnits
        oJobEst.Revenues = RS!Revenues
        oJobEst.Expenses = RS!Expenses
        Status = oJobEst.Insert
        RS.MoveNext
    Loop
    RS.Close
    Set RS = DB.OpenRecordset("aa qry PhaseID rollup 3")
    Do Until RS.EOF
        oJobEst.JobIndex = RS!JobIndex
        oJobEst.PhaseIndex = RS!PhaseIndex
        oJobEst.CostIndex = 0
        oJobEst.NumberOfUnits = RS!NumberOfUnits
        oJobEst.Revenues = RS!Revenues
        oJobEst.Expenses = RS!Expenses
        Status = oJobEst.Insert
        RS.MoveNext
    Loop
    RS.Close
    DB.Close
    Set RS = Nothing
    Set DB = Nothing
    Status = oJobEst.CloseFile
    Set oJobEst = Nothing

' Update the JrnlRow.dat data

    Status = oJrnlRow.OpenFile
    Status = oJrnlRow.StepFirst
    Do Until Status <> btStatusOK
        myPhaseID = oJrnlRow.PhaseID
        For i = 1 To N
            If OldPhaseID(i) = myPhaseID Then
                oJrnlRow.PhaseID = NewPhaseID(i)
                Status = oJrnlRow.Update
                If Status <> btStatusOK Then
                    MsgBox "JRNLROW.DAT: Update failed changing " & OldPhaseID(i) & " to " &
NewPhaseID(i)
                End If
                Exit For
            End If
        Next i
        If oJrnlRow.CostID <> "" Then
            oJrnlRow.CostID = ""
            Status = oJrnlRow.Update
        End If
        Status = oJrnlRow.StepNext
    Loop
    Status = oJrnlRow.CloseFile
    Set oJrnlRow = Nothing

' Clean up

    BtrieveDisconnect
    ReDim OldPhaseID(0)
    ReDim NewPhaseID(0)
    DoCmd.Hourglass False

End Sub

-----------------------------------------------------------------------------------------
```

```
aa qry PhaseID rollup 1:

SELECT [Job Estimate].JobIndex,
[Job Estimate].PhaseIndex,
Sum([Job Estimate].NumberOfUnits) AS NumberOfUnits,
Sum([Job Estimate].Revenues) AS Revenues,
Sum([Job Estimate].Expenses) AS Expenses
FROM ([Job Estimate]
INNER JOIN [Phase Codes] ON [Job Estimate].PhaseIndex = [Phase Codes].Index)
INNER JOIN [aa PhaseID Changes]
ON [Phase Codes].PhaseID = [aa PhaseID Changes].OldPhaseID
WHERE ((([aa PhaseID Changes].Merge)=False))
GROUP BY [Job Estimate].JobIndex, [Job Estimate].PhaseIndex
ORDER BY [Job Estimate].JobIndex, [Job Estimate].PhaseIndex;

aa qry PhaseID rollup 2:

SELECT [Job Estimate].JobIndex,
[aa PhaseID Changes].NewPhaseID,
Sum([Job Estimate].NumberOfUnits) AS NumberOfUnits,
Sum([Job Estimate].Revenues) AS Revenues,
Sum([Job Estimate].Expenses) AS Expenses
FROM ([Job Estimate]
INNER JOIN [Phase Codes] ON [Job Estimate].PhaseIndex = [Phase Codes].Index)
INNER JOIN [aa PhaseID Changes] ON [Phase Codes].PhaseID = [aa PhaseID Changes].OldPhaseID
WHERE ((([aa PhaseID Changes].Merge)=True))
GROUP BY [Job Estimate].JobIndex, [aa PhaseID Changes].NewPhaseID;

aa qry PhaseID rollup 3:

SELECT [Job Estimate].JobIndex,
[Job Estimate].PhaseIndex,
Sum([Job Estimate].NumberOfUnits) AS NumberOfUnits,
Sum([Job Estimate].Revenues) AS Revenues,
Sum([Job Estimate].Expenses) AS Expenses
FROM ([Job Estimate]
INNER JOIN [Phase Codes] ON [Job Estimate].PhaseIndex = [Phase Codes].Index)
LEFT JOIN [aa PhaseID Changes] ON [Phase Codes].PhaseID = [aa PhaseID Changes].OldPhaseID
GROUP BY [Job Estimate].JobIndex, [Job Estimate].PhaseIndex, [aa PhaseID Changes].OldPhaseID
HAVING ((([aa PhaseID Changes].OldPhaseID) Is Null))
ORDER BY [Job Estimate].JobIndex, [Job Estimate].PhaseIndex;
```