Private Sub RecalcAssembles10(alngNewEstimateIDs() As Long, lngNewEstimateIDsCounter As Long)

Dim db As Database, lngCounter As Long, lngCounter2 As Long, qdf1 As QueryDef, lngMeterCounter As Long

Dim lngCityID2 As Long, rst1 As Recordset, rst2 As Recordset

lngCounter = 0

lngMeterCounter = lngNewEstimateIDsCounter

Set db = CurrentDb

'step thru the newly imported estimates

Do Until lngCounter = lngNewEstimateIDsCounter

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " A", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

lngCityID2 = luLabCityID(alngNewEstimateIDs(lngCounter))

'\*\*\* Insert crew records (CrewID exists) into ElinesCrewT, ElinesCrewE, ComponentsCrewT, ComponentsCrewE \*\*\*

'ElinesCrewT

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqElinesCrewT ( ElineID, OT, MTradeID, TradeName, TradeQty, TradeRatePerHr, HrsPerDay ) " \_

& "SELECT TqElines.ElineID, False AS OT, TqCrewTrades.MTradeID, TqMasterTrades.MTradeName, TqCrewTrades.TradeQty, HourlyCost([TqCrewTrades.mTradeID]," & lngCityID2 & ") AS TradeRatePerHr, TradeHoursPerDay([TqCrewTrades.mTradeID]," & lngCityID2 & ") AS Hours " \_

& "FROM (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN (TqCrewTrades INNER JOIN TqMasterTrades ON TqCrewTrades.MTradeID = TqMasterTrades.MTradeID) ON TqElines.CrewID = TqCrewTrades.CrewID " \_

& "WHERE (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((TqElines.CrewID) Is Not Null));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " B", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ElinesCrewE

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqElinesCrewE ( ElineID, MEquipID, EquipName, EquipQty, EquipRatePerDay ) " \_

& "SELECT TqElines.ElineID, TqMasterEquip.MEquipID, TqMasterEquip.MEquipName, TqCrewEquip.EquipQty, DailyECost([TqCrewEquip.mEquipID]," & lngCityID2 & ") AS EquipRatePerDay " \_

& "FROM (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN (TqCrewEquip INNER JOIN TqMasterEquip ON TqCrewEquip.MEquipID = TqMasterEquip.MEquipID) ON TqElines.CrewID = TqCrewEquip.CrewID " \_

& "WHERE (((TqCrewEquip.CrewID) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & "));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " C", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ComponentsCrewT

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqComponentsCrewT ( ComponentID, OT, MTradeID, TradeName, TradeQty, TradeRatePerHr, HrsPerDay ) " \_

& "SELECT TqComponents.ComponentID, False AS OT, TqCrewTrades.MTradeID, TqMasterTrades.MTradeName, TqCrewTrades.TradeQty, HourlyCost([TqCrewTrades.mTradeID]," & lngCityID2 & ") AS TradeRatePerHr, TradeHoursPerDay([TqCrewTrades.mTradeID]," & lngCityID2 & ") AS Hours " \_

& "FROM ((TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID) INNER JOIN (TqCrewTrades INNER JOIN TqMasterTrades ON TqCrewTrades.MTradeID = TqMasterTrades.MTradeID) ON TqComponents.CrewID = TqCrewTrades.CrewID " \_

& "WHERE (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((TqComponents.CrewID) Is Not Null));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " D", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ComponentsCrewE

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqComponentsCrewE ( ComponentID, MEquipID, EquipName, EquipQty, EquipRatePerDay ) " \_

& "SELECT TqComponents.ComponentID, TqMasterEquip.MEquipID, TqMasterEquip.MEquipName, TqCrewEquip.EquipQty, DailyECost([TqCrewEquip.mEquipID]," & lngCityID2 & ") AS EquipRatePerDay " \_

& "FROM ((TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID) INNER JOIN (TqCrewEquip INNER JOIN TqMasterEquip ON TqCrewEquip.MEquipID = TqMasterEquip.MEquipID) ON TqComponents.CrewID = TqCrewEquip.CrewID " \_

& "WHERE (((TqCrewEquip.CrewID) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & "));")

qdf1.Execute

'====================

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " E", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'\*\*\* Insert Trade records (no crew) into ElinesCrewT, ElinesCrewE, ComponentsCrewT, ComponentsCrewE \*\*\*

'ElinesCrewT

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqElinesCrewT ( ElineID, OT, MTradeID, TradeName, TradeQty, TradeRatePerHr, HrsPerDay ) " \_

& "SELECT TqElines.ElineID, False AS OT, TqMasterTrades.MTradeID, TqMasterTrades.MTradeName, TqElines.TradeQty, HourlyCost([TqMasterTrades.mTradeID]," & lngCityID2 & ") AS TradeRatePerHr, TradeHoursPerDay([TqMasterTrades.mTradeID]," & lngCityID2 & ") AS Hours " \_

& "FROM (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqMasterTrades ON TqElines.MTradeID = TqMasterTrades.MTradeID " \_

& "WHERE (((TqMasterTrades.MTradeID) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & "));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " F", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ElinesCrewE

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqElinesCrewE ( ElineID, MEquipID, EquipName, EquipQty, EquipRatePerDay ) " \_

& "SELECT TqElines.ElineID, TqMasterEquip.MEquipID, TqMasterEquip.MEquipName, TqElines.EquipQty, DailyECost([TqMasterEquip.mEquipID]," & lngCityID2 & ") AS EquipRatePerDay " \_

& "FROM (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqMasterEquip ON TqElines.MEquipID = TqMasterEquip.MEquipID " \_

& "WHERE (((TqMasterEquip.MEquipID) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & "));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " G", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ComponentsCrewT

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqComponentsCrewT ( ComponentID, OT, MTradeID, TradeName, TradeQty, TradeRatePerHr, HrsPerDay ) " \_

& "SELECT TqComponents.ComponentID, False AS OT, TqMasterTrades.MTradeID, TqMasterTrades.MTradeName, TqComponents.TradeQty, HourlyCost([TqMasterTrades.mTradeID]," & lngCityID2 & ") AS TradeRatePerHr, TradeHoursPerDay([TqMasterTrades.mTradeID]," & lngCityID2 & ") AS Hours " \_

& "FROM ((TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID) INNER JOIN TqMasterTrades ON TqComponents.MTradeID = TqMasterTrades.MTradeID " \_

& "WHERE (((TqMasterTrades.MTradeID) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & "));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " H", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ComponentsCrewE

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqComponentsCrewE ( ComponentID, MEquipID, EquipName, EquipQty, EquipRatePerDay ) " \_

& "SELECT TqComponents.ComponentID, TqMasterEquip.MEquipID, TqMasterEquip.MEquipName, TqComponents.EquipQty, DailyECost([TqMasterEquip.mEquipID]," & lngCityID2 & ") AS EquipRatePerDay " \_

& "FROM ((TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID) INNER JOIN TqMasterEquip ON TqComponents.MEquipID = TqMasterEquip.MEquipID " \_

& "WHERE (((TqMasterEquip.MEquipID) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & "));")

qdf1.Execute

'Thus far we have filled the CrewE & CrewT tables based on finding matching Crew & Trade ID's

'but matching ID's may not exist. Let's check the total $/day in CrewT CrewE and add Men to make $ match

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " H.1", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ElinesCrewT

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqElinesCrewT ( ElineID, TradeQty, HrsPerDay, TradeRatePerHr, TradeName, OT ) " \_

& "SELECT TqElines.ElineID, ([DailyLabCost]-nz([Cost],0))/320 AS MyQty, 8 AS MyHrs, 40 AS MyRate, 'Men' AS MyName, False AS MyOT " \_

& "FROM (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) LEFT JOIN qFC300ECT ON TqElines.ElineID = qFC300ECT.ElineID " \_

& "WHERE (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((TqElines.SeeDetails)=True) AND ((TqElines.DailyLabCost) Is Not Null And (TqElines.DailyLabCost)<>0) AND ((nz([DailyLabCost],0)-nz([Cost],0)) Not Between -0.1 And 0.1));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " H.2", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ElinesCrewE

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqElinesCrewE ( ElineID, EquipQty, EquipRatePerDay, EquipName ) " \_

& "SELECT TqElines.ElineID, ([DailyEqCost]-nz([Cost],0))/1000 AS MyQty, 1000 AS MyRate, 'Equipment' AS MyName " \_

& "FROM (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) LEFT JOIN qFC300ECE ON TqElines.ElineID = qFC300ECE.ElineID " \_

& "WHERE (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((TqElines.SeeDetails)=True) AND ((TqElines.DailyEqCost) Is Not Null And (TqElines.DailyEqCost)<>0) AND ((nz([DailyEqCost],0)-nz([Cost],0)) Not Between -0.1 And 0.1));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " H.3", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ComponentsCrewT

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqComponentsCrewT ( ComponentID, TradeQty, HrsPerDay, TradeRatePerHr, TradeName, OT ) " \_

& "SELECT TqComponents.ComponentID, ([TqComponents.DailyLabCost]-nz([Cost],0))/320 AS MyQty, 8 AS MyHrs, 40 AS MyRate, 'Men' AS MyName, False AS MyOT " \_

& "FROM ((TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID) LEFT JOIN qFC300CCT ON TqComponents.ComponentID = qFC300CCT.ComponentID " \_

& "WHERE (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((TqComponents.SeeDetails)=True) AND ((TqComponents.DailyLabCost) Is Not Null And (TqComponents.DailyLabCost)<>0) AND ((nz([TqComponents.DailyLabCost],0)-nz([Cost],0)) Not Between -0.1 And 0.1));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " H.4", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'ComponentsCrewE

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqComponentsCrewE ( ComponentID, EquipQty, EquipRatePerDay, EquipName ) " \_

& "SELECT TqComponents.ComponentID, ([TqComponents.DailyEqCost]-nz([Cost],0))/1000 AS MyQty, 1000 AS MyRate, 'Equipment' AS MyName " \_

& "FROM ((TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID) LEFT JOIN qFC300CCE ON TqComponents.ComponentID = qFC300CCE.ComponentID " \_

& "WHERE (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((TqComponents.SeeDetails)=True) AND ((TqComponents.DailyEqCost) Is Not Null And (TqComponents.DailyEqCost)<>0) AND ((nz([TqComponents.DailyEqCost],0)-nz([Cost],0)) Not Between -0.1 And 0.1));")

qdf1.Execute

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\* now we must recalc all AssMhrsPUnit and AssEqDaysPUnit fields

'because versions earlier than 2R76 did not use these fields so they may be blank

'screens now use these fields to display total MHs & Mdays \*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'1) Clear these fields of any values

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " I", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'Clear Elines AssMhrsPUnit and AssEqDaysPUnit fields

Set qdf1 = db.CreateQueryDef("", "UPDATE TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID " \_

& "SET TqElines.AssMhrsPunit = Null, TqElines.AssEqDaysPunit = Null " \_

& "WHERE (((TqElines.AssMhrsPunit) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ")) OR (((TqElines.AssEqDaysPunit) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & "));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " J", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'Clear Components AssMhrsPUnit and AssEqDaysPUnit fields

Set qdf1 = db.CreateQueryDef("", "UPDATE (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID " \_

& "SET TqComponents.AssMhrsPunit = Null, TqComponents.AssEqDaysPunit = Null " \_

& "WHERE (((TqComponents.AssMhrsPunit) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ")) OR (((TqComponents.AssEqDaysPunit) Is Not Null) AND ((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & "));")

qdf1.Execute

'2) Calc AssMhrsPUnit and AssEqDaysPUnit fields for all \*Detail Lines\*

'Mhrs/SF = Mhrs/Day / SF/Day or sum of (TradeQty\*HrsPerDay) / Output

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " K", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'Elines

Set qdf1 = db.CreateQueryDef("", "UPDATE TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID " \_

& "SET TqElines.AssMhrsPunit = Calc2ElineAssMhrsPUnit([ElineID],[DailyOutput]), TqElines.AssEqDaysPunit = Calc2ElineAssEqDaysPUnit([ElineID],[DailyOutput]) " \_

& "WHERE (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((TqElines.SeeDetails)=True));")

qdf1.Execute

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " L", (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'Components

Set qdf1 = db.CreateQueryDef("", "UPDATE (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID " \_

& "SET TqComponents.AssMhrsPunit = Calc2CompAssMhrsPUnit([ComponentID],[TqComponents].[DailyOutput]), TqComponents.AssEqDaysPunit = Calc2CompAssEqDaysPUnit([ComponentID],[TqComponents].[DailyOutput]) " \_

& "WHERE (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((TqComponents.SeeDetails)=True));")

qdf1.Execute

'3) Calc AssMhrsPUnit and AssEqDaysPUnit fields for all \*Assembly Lines\*

'the basic calc is Total Mhrs/Ass / SF/Ass = Mhrs/SF

'or Total Mhrs/Ass / Divider = Mhrs/SF

'to calc the Total Mhrs in an assembly (assume assembly is a partition with a stud line and a gypbd line)

'Studs >> Qty = 300 LF AssMhrsPunit = 2 then 300 x 2 Mhrs/LF = 600 Mhs per Assembly

'Gypbd >> Qty = 800 SF AssMhrsPunit = 3 then 800 x 3 Mhrs/SF = 2400 Mhs per Assembly

' Total Mhs per Assembly = 3000 Mhrs per Assembly

'Thus Total Mhrs/Ass / Divider = Mhrs/SF

'or 3000 Mhrs/Ass / 400 Sf/Ass = 7.5 Mhrs/SF

'To Calc AssMhrsPUnit and AssEqDaysPUnit fields for \*Assembly Lines\*, all higher level AssMhrsPUnit must be known.

'thus A) find all lines where SeeComponents = True and and AssMhrsPUnit is null

'and all sub Assembly lines have AssMhrsPUnit values

'now calc per the above procedure

' B) Repeat A until Records effected = 0 or Counter = 30

' C) Do A) & B) with ComponentID linked to Component2ID until all assemblies are calced

' D) Then to A) & B) for ElineID linked to ComponentID until all assemblies are calced

'Calc Components AssMhrsPUnit

lngCounter2 = 0

'set the qdf1.RecordsAffected property so it's <> 0

Set qdf1 = db.CreateQueryDef("", "UPDATE TqMyCompanyGlob " \_

& "SET TqMyCompanyGlob.MyCompZip = [MyCompZip] " \_

& "WHERE (((TqMyCompanyGlob.MyCompanyID)=1));")

qdf1.Execute

Do Until qdf1.RecordsAffected = 0 Or lngCounter2 = 30

'RecordsAffected should become 0 after a few loops - lngCounter2 is just a safety item

'to prevent an infinate loop in case of a problem

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " M " & lngCounter2, (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'Empty TempID table

Set qdf1 = db.CreateQueryDef("", "DELETE TqTempIDs.\* FROM TqTempIDs;")

qdf1.Execute

'Populate TempID table with eligable ComponentIDs

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqTempIDs ( TempID ) " \_

& "SELECT TqComponents.ComponentID " \_

& "FROM (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN (TqComponents INNER JOIN TqComponents AS TqComponents\_1 ON TqComponents.ComponentID = TqComponents\_1.Component2ID) ON TqElines.ElineID = TqComponents.ElineID " \_

& "GROUP BY TqComponents.ComponentID, TqCsiLines.EstimateID " \_

& "HAVING (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((Sum(IIf([TqComponents\_1].[SeeComponents]=Yes And IsNull([TqComponents\_1].[AssMhrsPunit]),1,0)))=0));")

qdf1.Execute

If qdf1.RecordsAffected = 0 Then Exit Do '<<this is slow method. See ImportPC.ImportEstmates2R80 for Do Until lngCountTot = lngCount method which is much faster

'Use TempID table to limit records updated

Set qdf1 = db.CreateQueryDef("", "UPDATE TqTempIDs INNER JOIN TqComponents ON TqTempIDs.TempID = TqComponents.ComponentID " \_

& "SET TqComponents.AssMhrsPunit = Calc3CompAssMhrPUnit([ComponentID],[TqComponents].[Divider]) " \_

& "WHERE (((TqComponents.AssMhrsPunit) Is Null));")

qdf1.Execute

lngCounter2 = lngCounter2 + 1

Loop

'Calc Elines AssMhrsPUnit

lngCounter2 = 0

'set the qdf1.RecordsAffected property so it's <> 0

Set qdf1 = db.CreateQueryDef("", "UPDATE TqMyCompanyGlob " \_

& "SET TqMyCompanyGlob.MyCompZip = [MyCompZip] " \_

& "WHERE (((TqMyCompanyGlob.MyCompanyID)=1));")

qdf1.Execute

Do Until qdf1.RecordsAffected = 0 Or lngCounter2 = 30

'RecordsAffected should become 0 after a few loops - lngCounter2 is just a safety item

'to prevent an infinate loop in case of a problem

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " N " & lngCounter2, (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'Empty TempID table

Set qdf1 = db.CreateQueryDef("", "DELETE TqTempIDs.\* FROM TqTempIDs;")

qdf1.Execute

'Populate TempID table with eligable ElineIDs

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqTempIDs ( TempID ) " \_

& "SELECT TqElines.ElineID " \_

& "FROM TqCsiLines INNER JOIN (TqElines INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID) ON TqCsiLines.CsiLineID = TqElines.CsiLineID " \_

& "GROUP BY TqElines.ElineID, TqCsiLines.EstimateID " \_

& "HAVING (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((Sum(IIf([TqComponents].[SeeComponents]=Yes And IsNull([TqComponents].[AssMhrsPunit]),1,0)))=0));")

qdf1.Execute

If qdf1.RecordsAffected = 0 Then Exit Do '<<this is slow method. See ImportPC.ImportEstmates2R80 for Do Until lngCountTot = lngCount method which is much faster

'Use TempID table to limit records updated

Set qdf1 = db.CreateQueryDef("", "UPDATE TqTempIDs INNER JOIN TqElines ON TqTempIDs.TempID = TqElines.ElineID " \_

& "Set TqElines.AssMhrsPunit = Calc3ElineAssMhrPUnit([ElineID], [Divider]) " \_

& "WHERE (((TqElines.AssMhrsPunit) Is Null));")

qdf1.Execute

lngCounter2 = lngCounter2 + 1

Loop

'Calc Components AssEqDaysPUnit

lngCounter2 = 0

'set the qdf1.RecordsAffected property so it's <> 0

Set qdf1 = db.CreateQueryDef("", "UPDATE TqMyCompanyGlob " \_

& "SET TqMyCompanyGlob.MyCompZip = [MyCompZip] " \_

& "WHERE (((TqMyCompanyGlob.MyCompanyID)=1));")

qdf1.Execute

Do Until qdf1.RecordsAffected = 0 Or lngCounter2 = 30

'RecordsAffected should become 0 after a few loops - lngCounter2 is just a safety item

'to prevent an infinate loop in case of a problem

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " O " & lngCounter2, (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'Empty TempID table

Set qdf1 = db.CreateQueryDef("", "DELETE TqTempIDs.\* FROM TqTempIDs;")

qdf1.Execute

'Populate TempID table with eligable ComponentIDs

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqTempIDs ( TempID ) " \_

& "SELECT TqComponents.ComponentID " \_

& "FROM (TqCsiLines INNER JOIN TqElines ON TqCsiLines.CsiLineID = TqElines.CsiLineID) INNER JOIN (TqComponents INNER JOIN TqComponents AS TqComponents\_1 ON TqComponents.ComponentID = TqComponents\_1.Component2ID) ON TqElines.ElineID = TqComponents.ElineID " \_

& "GROUP BY TqComponents.ComponentID, TqCsiLines.EstimateID " \_

& "HAVING (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((Sum(IIf([TqComponents\_1].[SeeComponents]=Yes And IsNull([TqComponents\_1].[AssEqDaysPUnit]),1,0)))=0));")

qdf1.Execute

If qdf1.RecordsAffected = 0 Then Exit Do '<<this is slow method. See ImportPC.ImportEstmates2R80 for Do Until lngCountTot = lngCount method which is much faster

'Use TempID table to limit records updated

Set qdf1 = db.CreateQueryDef("", "UPDATE TempIDs INNER JOIN TqComponents ON TempIDs.TempID = TqComponents.ComponentID " \_

& "Set TqComponents.AssEqDaysPunit = Calc3CompAssEqDaysPUnit([ComponentID], [TqComponents].[Divider]) " \_

& "WHERE (((TqComponents.AssEqDaysPunit) Is Null));")

qdf1.Execute

lngCounter2 = lngCounter2 + 1

Loop

'Calc Elines AssEqDaysPUnit

lngCounter2 = 0

'set the qdf1.RecordsAffected property so it's <> 0

Set qdf1 = db.CreateQueryDef("", "UPDATE TqMyCompanyGlob " \_

& "SET TqMyCompanyGlob.MyCompZip = [MyCompZip] " \_

& "WHERE (((TqMyCompanyGlob.MyCompanyID)=1));")

qdf1.Execute

Do Until qdf1.RecordsAffected = 0 Or lngCounter2 = 30

'RecordsAffected should become 0 after a few loops - lngCounter2 is just a safety item

'to prevent an infinate loop in case of a problem

ShowMeter "Importing... FC3 Estimate " & lngMeterCounter & " P " & lngCounter2, (lngCounter + 1) / lngNewEstimateIDsCounter \* 100

'Empty TempID table

Set qdf1 = db.CreateQueryDef("", "DELETE TqTempIDs.\* FROM TqTempIDs;")

qdf1.Execute

'Populate TempID table with eligable ElineIDs

Set qdf1 = db.CreateQueryDef("", "INSERT INTO TqTempIDs ( TempID ) " \_

& "SELECT TqElines.ElineID " \_

& "FROM TqCsiLines INNER JOIN (TqElines INNER JOIN TqComponents ON TqElines.ElineID = TqComponents.ElineID) ON TqCsiLines.CsiLineID = TqElines.CsiLineID " \_

& "GROUP BY TqElines.ElineID, TqCsiLines.EstimateID " \_

& "HAVING (((TqCsiLines.EstimateID)=" & alngNewEstimateIDs(lngCounter) & ") AND ((Sum(IIf([TqComponents].[SeeComponents]=Yes And IsNull([TqComponents].[AssEqDaysPUnit]),1,0)))=0));")

qdf1.Execute

If qdf1.RecordsAffected = 0 Then Exit Do '<<this is slow method. See ImportPC.ImportEstmates2R80 for Do Until lngCountTot = lngCount method which is much faster

'Use TempID table to limit records updated

Set qdf1 = db.CreateQueryDef("", "UPDATE TqTempIDs INNER JOIN TqElines ON TqTempIDs.TempID = TqElines.ElineID " \_

& "Set TqElines.AssEqDaysPunit = Calc3ElineAssEqDaysPUnit([ElineID], [Divider]) " \_

& "WHERE (((TqElines.AssEqDaysPunit) Is Null));")

qdf1.Execute

lngCounter2 = lngCounter2 + 1

Loop

lngCounter = lngCounter + 1

lngMeterCounter = lngMeterCounter - 1

Loop

End Sub