Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmAllocated"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub cmdSerialNo_Click()
'
'   Verify that first record has been generated and open form to generate multiple records
'
'   09 Mar 2012 TM
'
If IsNull(Me![AllocatedID]) Then
    MsgBox "Enter Traceability information for first record"
    Me![PO].SetFocus
End If


Forms![frmWareHouse]![lstOrderLineBOM].SetFocus    '   To save current record before opening form
DoCmd.OpenForm "frmPopulateAllocated", acNormal, , "[AllocatedID]=" & Me![AllocatedID]
End Sub

Private Sub PO_AfterUpdate()
Me![AllocatedDate] = Now
Me![AllocatedSign] = Forms![Start]![txtOperatorID]
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmAllocatedUB"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboPart_GotFocus()
cboPart.Requery
End Sub

```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmAsBuiltBOM"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboAllocatedID_AfterUpdate()
'
'   Set RecordedTime
'
    Me!RecordedTime = Now

End Sub

Private Sub cboAllocatedID_GotFocus()
    If Me![Scrap] Then
        Me!Description.SetFocus
        Exit Sub
    End If

cboAllocatedID.Requery
End Sub

Private Sub cboOperationBOMID_GotFocus()
cboOperationBOMID.Requery
End Sub

'------------------------------------------------------------
' cmdDuplicate_Click
'
'------------------------------------------------------------
Private Sub cmdDuplicate_Click()
On Error GoTo cmdDuplicate_Click_Err

    Dim dbs As Database
    Dim rst As DAO.Recordset

    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset("tblAsBuiltBOM", dbOpenDynaset)

    rst.AddNew
        rst!AssemblyID = Me![AssemblyID]
        rst!OperationBOMID = Me![OperationBOMID]
        rst!Qty = 1
    rst.Update
```

```

   rst.Close
   dbs.Close

   Forms![frmAssembly]![frmlAsBuiltOperation].Form![frmAsBuiltBOM].Requery

   MsgBox "Check and eventually update Qty in both records"

cmdDuplicate_Click_Exit:
   Exit Sub

cmdDuplicate_Click_Err:
   MsgBox Error$
   Resume cmdDuplicate_Click_Exit

End Sub


Private Sub Scrap_AfterUpdate()
'
'  Set StatusChangeTime
'
   Me![StatusChangeTime] = Now

End Sub

Private Sub Scrap_GotFocus()
   If Me![Scrap] Then
      If Now - Nz(Me![StatusChangeTime], 0) > 0.005 Then
         Me!Description.SetFocus
      End If
   End If

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmAsBuiltBOMAll"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboAllocatedID_AfterUpdate()
'
'   Set RecordedTime
'
    Me!RecordedTime = Now

End Sub

Private Sub cboAllocatedID_GotFocus()
    If DLookup("[LockOperation]", "tblAsBuiltOperation", "(AssemblyID) = " & Me!AssemblyID & "AND (OperationID)=" & Me!OperationID) Then
        Me!PartNo.SetFocus
        Exit Sub
    End If

    If Me![Scrap] Then
        Me!PartNo.SetFocus
        Exit Sub
    End If

cboAllocatedID.Requery
End Sub

Private Sub Detail_DblClick(Cancel As Integer)
MsgBox Me![Qty]
End Sub

Private Sub Qty_Click()
If DLookup("[LockOperation]", "tblAsBuiltOperation", "(AssemblyID) = " & Me!AssemblyID & "AND (OperationID)=" & Me!OperationID) Then
    Me!PartNo.SetFocus
End If
End Sub

Private Sub Scrap_AfterUpdate()
'
'   Set StatusChangeTime
'
    Me![StatusChangeTime] = Now
```

```

End Sub

Private Sub Scrap_GotFocus()
    If Me![Scrap] Then
        If Now - Nz(Me![StatusChangeTime], 0) > 0.005 Then
            Me!PartNo.SetFocus
        End If
    End If
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmAsBuiltOperationSlot"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboSlotSign_AfterUpdate()
On Error GoTo Err_cboSlotSign_AfterUpdate
'
'   Add basic slot data to tblStatistics and
'   add content to tblSlotResult and if applicable to tblScanLargeSlot
'   If data already has been recorded, hide old data
'   Lock forms
'   To unlock forms, signature must be removed
'

    Dim AssemblyID As Long
    Dim Stat As Integer
    Dim strSql As String, strDocName As String

    Stat = 0
    AssemblyID = Me![AssemblyID]
'
'   Slot requirements based on qrySlotData31 entered into tblSlotStatistics
'
    strSql = "[AssemblyID]=" & AssemblyID & " and [HideSlotStatistics]=False"
    If DCount("[AssemblyID]", "tblSlotStatistics", strSql) Then
        Stat = 10
        DoCmd.SetWarnings False
        DoCmd.OpenQuery "qupdSlotStatistics", acNormal, acEdit
        DoCmd.SetWarnings True
    End If
    Stat = Stat + 1
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "qryappSlotData31", acNormal, acEdit
    DoCmd.SetWarnings True
'
'   Slot requirements based on qrySlotData31 entered into tblSlotResult
'
    Stat = Stat + 1
    strSql = "[AssemblyID]=" & AssemblyID & " and [HideSlotResult]=False"
    If DCount("[AssemblyID]", "tblSlotResult", strSql) Then
        Stat = Stat + 20
        DoCmd.SetWarnings False
        DoCmd.OpenQuery "qupdSlotResult", acNormal, acEdit
        DoCmd.SetWarnings True
    End If
    Stat = Stat + 1
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "qryappSlotData3", acNormal, acEdit
    DoCmd.SetWarnings True
'
'   Slot requirements based on qrySlotExp1111 entered into tblScanLargeSlot
'
    Stat = Stat + 1
    strSql = "[AssemblyID]=" & AssemblyID & " and [HideScanLargeSlot]=False"
    If DCount("[AssemblyID]", "tblScanLargeSlot", strSql) Then
        Stat = Stat + 30
        DoCmd.SetWarnings False
        DoCmd.OpenQuery "qupdScanLargeSlot", acNormal, acEdit
        DoCmd.SetWarnings True
    End If
    Stat = Stat + 1
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "qryappSlotExp1111", acNormal, acEdit
    DoCmd.SetWarnings True
'
'   Lock record, forms and controls
'
    Stat = Stat + 1
    Me![LockOperation] = True
    Forms![frmSlotResult]![frmSlotLarge].Locked = True
    Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Locked = True
    Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Enabled = False
    Forms![frmSlotResult]![frmSlotFile].Locked = True
    Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Locked = True
    Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Enabled = False
'
'   Set date
'
    Me![OperationDate] = Now
'
'   If this is last operation, set sign and date in tblAssembly
'
    Stat = Stat + 1
    strSql = "[HideOperation] = 0 And [OrderLineID] = " &
Forms![frmSlotResult]![cboOrderLine]
    If DMax("[OperationNo]", "tblOperation", strSql) = Forms![frmSlotResult]![txtOperationNo]
* 1 Then
        Me![AssemblyDate] = Me![OperationDate]
        Me![AssemblySign] = Me![cboSlotSign]
    End If

Exit_cboSlotSign_AfterUpdate:
    Exit Sub

Err_cboSlotSign_AfterUpdate:
'   Write error message to tblBug
    Forms![Start]![txtBug] = "cboSlotSign_AfterUpdate: " & Stat & Err.Description
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "qryappBug", acNormal, acEdit
    DoCmd.SetWarnings True
'
    MsgBox Err.Description
    Resume Exit_cboSlotSign_AfterUpdate

End Sub

Private Sub cboSlotSign_GotFocus()

    If Nz(Me![cboSlotSign], 0) Then
        MsgBox "Record has been signed off"
        Me![NCText].SetFocus
        Exit Sub
    End If

    If Forms![frmSlotResult]![txtStatusVal] = 99 Then
        MsgBox "Slot data not imported"
        Me![NCText].SetFocus
        Exit Sub
    End If

    If DLookup("[MaxOfExprStatus]", "qryPreviousOpSlotStatusGB") = 0 Then
        MsgBox "Previous operation not completed"
        Me![NCText].SetFocus
        Exit Sub
    End If

    If DLookup("[MaxOfExprStatus]", "qryPreviousOpQCStatusGB") = 0 Then
        MsgBox "Previous operation not signed off by QC"
        Me![NCText].SetFocus
        Exit Sub
    End If

    If Me![fraStatus] = 1 Then
        If Forms![frmSlotResult]![txtStatusVal] > 0 Then
            MsgBox "Check data! Screen do not meet requirements."
            Me![NCText].SetFocus
            Exit Sub
        End If
    Else
        If IsNull(Me![NCText]) Then
            MsgBox "Check status or write explanation in Non Conformancy text box"
            Me![NCText].SetFocus
        End If
    End If

End Sub

Private Sub cmdUnlock_Click()
'
'   Can be unlocked if next operation is not signed off
'   Can remove signature if current record is unlocked
'
```

```
   If IsNull(Me![cboSlotSign]) Then
      Exit Sub
   End If

   If Me![cboSlotSign] Then
      If DFirst("[OperationSign]", "qryListOperationStatus") Then
         MsgBox "Record must be ulocked by QC!"
         Exit Sub
      End If
   End If
'
'  If OK on delete signature
'  Lock/unlock forms and controls in frmSlotResult
'
   If MsgBox("Do you want to delete signature to open for new import?", vbOKCancel, "ResLog") = vbOK Then
      Me![cboSlotSign] = Null
      Me![LockOperation] = False
      Forms![frmSlotResult]![frmSlotLarge].Locked = False
      Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Locked = False
      Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Enabled = True
      Forms![frmSlotResult]![frmSlotFile].Locked = False
      Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Locked = False
      Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Enabled = True
   End If

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmAssembly"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboAssemblyID_AfterUpdate()

'
'   Set default values after assembly has been selected
'

On Error GoTo Err_cboAssemblyID_AfterUpdate

    Dim stSql As String
'
'   Set AssemblyID in form Start as reference for Slot opening data queries
'

'    Forms![frmAssembly]![frmOrderLine].Form![chkAllinWO] = 0
    Forms![Start]![txtMinAssemblyID] = Me![cboAssemblyID]
    Forms![Start]![txtMaxAssemblyID] = Me![cboAssemblyID]
    Forms![Start]![txtOperationNo] = Me![txtOperationNo]

'
'   Requery list box to reflect selected assembly
'   Select record in frmWrapDetail form based on selected assembly
'

    lstWrapID.Requery
    Me![lstWrapID] = DMin("[WrapID]", "tblWrap", "[AssemblyID]=" & Me![cboAssemblyID])

'
'   Lock Operation form if also operation is selected and next operation is signed off
'
    stSql = "[AssemblyID]=" & [Forms]![frmAssembly]![cboAssemblyID]
    stSql = stSql & " AND [OperationID]=" & [Forms]![frmAssembly]![cboOperationID]

    If Not IsNull([Forms]![frmAssembly]![cboOperationID]) Then
        Forms![frmAssembly]![frmlAsBuiltOperation].Locked = DLookup("[LockOperation]", "tblAsBuiltOperation", stSql)
        Forms![frmAssembly]![frmlAsBuiltOperation].Enabled = Not DLookup("[LockOperation]", "tblAsBuiltOperation", stSql)
    End If

```

```
'
'   Lock wrapping form if next operation is signed off
'
    If DLookup("[LockOperation]", "qryWrapLock", "[AssemblyID]=" & [Forms]![frmAssembly]![cboAssemblyID]) Then
        Forms![frmAssembly]![frmAsBuiltOperationWrapping].Locked = True
        Forms![frmAssembly]![frmAsBuiltOperationWrapping].Enabled = False
        Forms![frmAssembly]![frmWrapDetail].Locked = True
        Forms![frmAssembly]![frmWrapDetail].Enabled = False
        Forms![frmAssembly]![cmdNewWrapSection].Enabled = False
    Else
        Forms![frmAssembly]![frmAsBuiltOperationWrapping].Locked = False
        Forms![frmAssembly]![frmAsBuiltOperationWrapping].Enabled = True
        Forms![frmAssembly]![frmWrapDetail].Locked = False
        Forms![frmAssembly]![frmWrapDetail].Enabled = True
        Forms![frmAssembly]![cmdNewWrapSection].Enabled = True
    End If

Exit_cboAssemblyID_AfterUpdate:
    Exit Sub

Err_cboAssemblyID_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboAssemblyID_AfterUpdate

End Sub

Private Sub cboAssemblyID_GotFocus()
'cboAssemblyID.Requery
End Sub

Private Sub cboOperationID_AfterUpdate()
'
'   Requery forms and controls after update of this control
'
'   10 Mar 2012 TM

    Dim stSql As String

    cboOperationID.Requery
    Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboOperationSign].Requery
    Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboOperatorSign2].Requery
    Forms![Start]![txtOperationNo] = Me![txtOperationNo]

'
'   Set second sign visible if required
'
'   04 Dec 2011 TM

    If IsNull(Me![chkSecondSign]) Then
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboOperatorSign2].Visible = False
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![OperationDate2].Visible = False
    ElseIf Me![chkSecondSign] Then
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboOperatorSign2].Visible = True
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![OperationDate2].Visible = True
    Else
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboOperatorSign2].Visible = False
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![OperationDate2].Visible = False
    End If

'
'   Set QC sign visible if required and unlock if operator has privileges
'

    If IsNull(Me![chkQCSign]) Then
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboQCOpSign].Visible = False
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![QCOpDate].Visible = False
    ElseIf Me![chkQCSign] Then
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboQCOpSign].Visible = True
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![QCOpDate].Visible = True
        If Forms![Start]![txtOperatorID] Then
            Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboQCOpSign].Locked = False
            Forms![frmAssembly]![frmlAsBuiltOperation].Form![QCOpDate].Locked = False
        Else
            Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboQCOpSign].Locked = True
            Forms![frmAssembly]![frmlAsBuiltOperation].Form![QCOpDate].Locked = True
        End If
    Else
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![cboQCOpSign].Visible = False
        Forms![frmAssembly]![frmlAsBuiltOperation].Form![QCOpDate].Visible = False
    End If

'
'   Lock form if assembly is selected
'
    If Not IsNull([Forms]![frmAssembly]![cboAssemblyID]) Then
        stSql = "[AssemblyID]=" & [Forms]![frmAssembly]![cboAssemblyID]
        stSql = stSql & " AND [OperationID]=" & [Forms]![frmAssembly]![cboOperationID]
        If Not IsNull(DLookup("[LockOperation]", "tblAsBuiltOperation", stSql)) Then
            Forms![frmAssembly]![frmlAsBuiltOperation].Locked = DLookup("[LockOperation]", "tblAsBuiltOperation", stSql)
            Forms![frmAssembly]![frmlAsBuiltOperation].Enabled = Not DLookup("[LockOperation]", "tblAsBuiltOperation", stSql)
        End If
    End If

End Sub

Private Sub cboOrder_AfterUpdate()
Forms![Start]![txtLocationID] = Me![txtLocationID]
End Sub

Private Sub cboOrder_GotFocus()
'   Terje 29Jan11
'   Filtration of orders
'
'   SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location, tblOrder.CompletedSign
'   FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
'   ORDER BY tblOrder.ProdOrder;

    Dim Sql As String

    Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign], [tblLocation.LocationID] "
    Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID "


    If Me![optionOrderSelect] = 1 Then
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    End If

    Forms![frmAssembly]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"


End Sub

Private Sub cboOrderLine_AfterUpdate()
'
'   If emty, set Order
'
On Error GoTo Err_cboOrderLine_AfterUpdate

    If IsNull(Me![cboOrder]) Then
        Me![cboOrder] = DLookup("[OrderID]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine])
    End If

    Forms![frmAssembly]![frmWrapDetail].Form![cboAxialWireID].Requery
    cboOperationID.Requery

'
'   Set parameters for slot data queries in Start form
'
    Forms![Start]![txtOrderLineID] = Me![cboOrderLine]
    Forms![Start]![txtLocationID] = Me![txtLocationID]

    If DLookup("[InclAllLines]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine]) Then
        Forms![Start]![txtInclAllLines] = 360
    Else
        Forms![Start]![txtInclAllLines] = 0
    End If

    Forms![Start]![txtMinAssemblyID] = 0
    Forms![Start]![txtMaxAssemblyID] = 0
'
'   Requery Assembly and clear combo boxes
'
    cboAssemblyID.Requery
    Me![cboAssemblyID] = Null
    Me![cboOperationID] = Null

Exit_cboOrderLine_AfterUpdate:
    Exit Sub

Err_cboOrderLine_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboOrderLine_AfterUpdate

End Sub

Private Sub cboOrderLine_GotFocus()
'   SELECT tblOrderLine.OrderLineID, tblOrderLine.WorkOrder, tblOrderLine.PartNo, tblOrderLine.Description, tblOrder.CompletedSign
'   FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID
'   WHERE (((tblOrder.CompletedSign)=0));
On Error GoTo Err_cboOrderLine_GotFocus

    Dim Sql As String

    Sql = "SELECT [OrderLineID], [WorkOrder], [PartNo], [tblOrderLine.Description], [CompletedSign] "
    Sql = Sql & "FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID "
    If Me![optionOrderSelect] = 1 Then
'   noe muffens når denne linjen ikke inkluderes - WHERE mangler
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    Else
        Sql = Sql & "WHERE [tblOrder.CompletedSign] >= 0 "
    End If

    If Not IsNull(Me![cboOrder]) Then
        Sql = Sql & "AND [tblOrder.OrderID] = " & Me![cboOrder]
    End If

    Forms![frmAssembly]![cboOrderLine].RowSource = Sql & " ORDER BY tblOrderLine.WorkOrder;"

Exit_cboOrderLine_GotFocus:
    Exit Sub

Err_cboOrderLine_GotFocus:
    MsgBox Err.Description
    Resume Exit_cboOrderLine_GotFocus

End Sub

Private Sub cboWrapMachin_GotFocus()
cboWrapMachin.Requery
End Sub

Private Sub cboWrapSection_AfterUpdate()
'
'   Set default values in form frmWrapDetail
'
If IsNull(Me![cboWrapMachin]) Then
    MsgBox "Select wrapping machine to set default wrapping parameters"
    Exit Sub
```

```
End If


If IsNull(Forms![frmAssembly]![frmWrapDetail].Form![cboMachineID]) Then
   Forms![frmAssembly]![frmWrapDetail].Form![cboMachineID] = Me![cboWrapMachin]
End If

End Sub

Private Sub cboWrapSection_GotFocus()
cboWrapSection.Requery
End Sub

Private Sub cmdNewWrapSection_Click()
'
'   Add new record to tblWrap and copy default values from first record
'
'   NOE SOM IKKE STEMMER MED DEKLARERING
'
'SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
'
On Error GoTo Err_cmdNewWrapSection_Click

   Dim stDocName As String
   Dim stSql As String
   Dim dbs As Database
   Dim rst As DAO.Recordset, rst1 As DAO.Recordset, rst2 As DAO.Recordset

   stSql = "SELECT tblWrap.* "
   stSql = stSql & "FROM tblWrap "
   stSql = stSql & "WHERE (tblWrap.AssemblyID)=" & Me![cboAssemblyID]


   Set dbs = CurrentDb
   Set recset = dbs.OpenRecordset(stSql, dbOpenDynaset)
   Set recset1 = dbs.OpenRecordset("tblWrap", dbOpenDynaset)

   recset.MoveFirst
   recset1.AddNew
   recset1!AssemblyID = Me![cboAssemblyID]
   recset1!MachineID = recset!MachineID
   recset1!WrapPitch = recset!WrapPitch
   recset1!WeldTime = recset!WeldTime
   recset1!SetCurrent = recset!SetCurrent
   recset1!WrapSpeed = recset!WrapSpeed
   recset1!WeldPointPos = recset!WeldPointPos
   recset1!ForceWeldWeel = recset!ForceWeldWeel
   recset1.Update

   lstWrapID.Requery
   Me![lstWrapID] = DMax("[WrapID]", "tblWrap", "[AssemblyID]=" & Me![cboAssemblyID])

   recset.Close
```

```
    recset1.Close
    dbs.Close

Exit_cmdNewWrapSection_Click:
    Exit Sub

Err_cmdNewWrapSection_Click:
    MsgBox Err.Description
    Resume Exit_cmdNewWrapSection_Click

End Sub
'----------------------------------------------------------
' cmdFileImport_Click
'
'----------------------------------------------------------
Private Sub cmdFileImport_Click()
On Error GoTo cmdFileImport_Click_Err

    DoCmd.OpenForm "frmSlotFileImport", acNormal, "", "", , acNormal


cmdFileImport_Click_Exit:
    Exit Sub

cmdFileImport_Click_Err:
    MsgBox Error$
    Resume cmdFileImport_Click_Exit

End Sub


'----------------------------------------------------------
' cmdSlotDetails_Click
' Not active TM 18 Nov 2012
'----------------------------------------------------------
Private Sub cmdSlotDetails_Click()
On Error GoTo cmdSlotDetails_Click_Err

    DoCmd.OpenForm "frmSlotResult", acNormal, "", "", , acNormal

    If Not IsNull(Me![cboOrder]) Then
        Forms![frmSlotResult]![cboOrder] = Me![cboOrder]
    End If
    If Not IsNull(Me![cboOrderLine]) Then
        Forms![frmSlotResult]![cboOrderLine] = Me![cboOrderLine]
    End If

cmdSlotDetails_Click_Exit:
    Exit Sub

cmdSlotDetails_Click_Err:
    MsgBox Error$
    Resume cmdSlotDetails_Click_Exit

```

```
End Sub


```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmAssyOpStatus"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboNCSign_AfterUpdate()
    If IsNull(Me![NCDate]) Then Me![NCDate] = Now
End Sub

Private Sub Option22_GotFocus()
'
'   Check signature and NC report no before this option can be selected
'
    If Me![fraStatus] = 4 Then
        MsgBox "This option should not be choosen"
        If IsNull(Me![cboNCSign]) Then
            MsgBox "Sign off before selecting this option"
            Me![cboNCSign].SetFocus
            Exit Sub
        End If

        If IsNull(Me![NCNo]) Then
            MsgBox "Enter Non Conformancy No before selecting this option"
            Me![NCNo].SetFocus
            Exit Sub
        End If
    End If

End Sub

Private Sub Option24_GotFocus()
'
'   Check signature and NC report no before this option can be selected
'
    If Me![fraStatus] = 4 Then
        If IsNull(Me![cboNCSign]) Then
            MsgBox "Sign off before selecting this option"
            Me![cboNCSign].SetFocus
            Exit Sub
        End If

        If IsNull(Me![NCNo]) Then
            MsgBox "Enter Non Conformancy No before selecting this option"
            Me![NCNo].SetFocus
            Exit Sub
        End If
    End If

End Sub

Private Sub Option30_GotFocus()
'
'   Check signature and NC report no before this option can be selected
'
    If Me![fraStatus] = 4 Then
        If IsNull(Me![cboNCSign]) Then
            MsgBox "Sign off before selecting this option"
            Me![cboNCSign].SetFocus
            Exit Sub
        End If

        If IsNull(Me![NCNo]) Then
            MsgBox "Enter Non Conformancy No before selecting this option"
            Me![NCNo].SetFocus
            Exit Sub
        End If
    End If

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmBug"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmlAsBuiltOperation"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboOperationID_GotFocus()
cboOperationID.Requery
End Sub

Private Sub cboOperationSign_AfterUpdate()
'
'   Lock form for previous records by running update query
'   This query sets all previous operations to status Lock
'   Set date for current record
'
    Dim strDocName As String

    strDocName = "qupdAsBuiltOperationLock"
    DoCmd.SetWarnings False
    DoCmd.OpenQuery strDocName, acNormal, acEdit
    DoCmd.SetWarnings True

    Me![OperationDate] = Now
'
'   If this is last operation, set sign and date in tblAssembly
'
    strSql = "[HideOperation] = 0 And [OrderLineID] = " & Forms![frmAssembly]![cboOrderLine]
    If DMax("[OperationNo]", "tblOperation", strSql) = Forms![frmAssembly]![cboOperationID] * 1 Then
        Me![AssemblyDate] = Me![OperationDate]
        Me![AssemblySign] = Me![cboOperationSign]
    End If

End Sub

Private Sub cboOperationSign_GotFocus()
'
'   Verify that all information has been recorded
'


    If Forms![frmAssembly]![txtWorkStationID] < 3 Then  '   Wrapping and ResGauge are 1 and 2
        MsgBox "Wrapping or ResGauge operations cannot be signed off in this form!"
        Me![NCText].SetFocus
        Exit Sub
    End If

    If IsNull(Me![Status]) Then
        Me![Status] = 1
    End If

    If DLookup("[MaxOfExprStatus]", "qryPreviousOpSlotStatusGB") = 0 Then
        MsgBox "Previous operation not completed"
        Me![NCText].SetFocus
        Exit Sub
    End If

    If DLookup("[MaxOfExprStatus]", "qryPreviousOpQCStatusGB") = 0 Then
        MsgBox "Previous operation not signed off by QC"
        Me![NCText].SetFocus
        Exit Sub
    End If

    If Me![Status] = 1 Then
        If DLookup("[TestStatus]", "qryOpStatTest") = 0 Then
            MsgBox "Inspection results missing"
            Me![fraStatus].SetFocus
        End If
        If DLookup("[MaxOfTraceability]", "qryOpStatBOM2") = 0 Then
            MsgBox "Traceability information missing"
            Me![fraStatus].SetFocus
        End If
        If DLookup("[MaxOfQtyRec]", "qryOpStatBOM2") = 0 Then
            MsgBox "Wrong traceability quantity recorded"
            Me![fraStatus].SetFocus
        End If
        If DLookup("[Sign2]", "qryOpStat2Sign") = 0 Then
            MsgBox "Second signature missing"
            Me![fraStatus].SetFocus
        End If
    Else
        If IsNull(Me![NCText]) Then
            MsgBox "Write explanation"
            Me![NCText].SetFocus
        End If
    End If

End Sub

Private Sub cboOperatorSign2_AfterUpdate()
Me![OperationDate2] = Now
Forms![frmAssembly]![txtOrder.Column(2)].SetFocus
End Sub

Private Sub cboQCOpSign_AfterUpdate()
Me![QCOpDate] = Now
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmNewWorkOrder"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub WorkOrder_AfterUpdate()
Me![OrderID] = [Forms]![frmOrder]![cboOrder]
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmOperation"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub HideOperation_AfterUpdate()
Forms![frmOrder]![cboOrderLine].Requery
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmOperationBOM"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboOperationID_AfterUpdate()
'
'   Copy quantity as default value if blank
'
On Error GoTo Err_cboOperationID_AfterUpdate

    If IsNull(Me![tblOperationBOM.Qty]) Then
        Me![tblOperationBOM.Qty] = Me![tblOrderLineBOM.Qty]
    End If

Exit_cboOperationID_AfterUpdate:
    Exit Sub

Err_cboOperationID_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboOperationID_AfterUpdate

End Sub



Private Sub cmdNew_Click()
'
'   Copy current part for new operation
'
On Error GoTo Err_cmdNew_Click

    Dim dbs As Database
    Dim rst As Recordset
    Dim OLbomid As Long

    OLbomid = Me![tblOrderLineBOM.OrderLineBOMID]

    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset("tblOperationBOM", dbOpenDynaset)
        rst.AddNew
        rst!OrderLineBOMID = OLbomid
        rst.Update
    rst.Close
    dbs.Close

Forms![frmOrder]![cboOrderLine].Requery
    'frmOrder.cboOrderLine.Requery

Exit_cmdNew_Click:
    Exit Sub

Err_cmdNew_Click:
    MsgBox Err.Description
    Resume Exit_cmdNew_Click

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmOperationTest"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboOperationID_AfterUpdate()
Me![OrderLineID] = Forms![frmOrder]![cboOrderLine]
End Sub

Private Sub cboOperationID_GotFocus()
cboOperationID.Requery
End Sub

Private Sub MaxVal_AfterUpdate()
'
'  Set code for operation test type
'
'  02 Dec 2011 TM
'

    If Not IsNull(Me![MaxVal]) Then
        Me![OpTestType] = 2
'   Else
'       If IsNull(Me![MinVal]) Then
'           Me![OpTestType] = 1
'       End If
    End If

End Sub

Private Sub MinVal_AfterUpdate()
'
'  Set code for operation test type
'
'  02 Dec 2011 TM
'

    If Not IsNull(Me![MinVal]) Then
        Me![OpTestType] = 2
'   Else
'       If IsNull(Me![MaxVal]) Then
'           Me![OpTestType] = 1
'       End If
    End If

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmOperator"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cmdNewRecOperator_Click()
On Error GoTo Err_cmdNewRecOperator_Click


    'DoCmd.OpenForm "frmOperatorSub"
    'DoCmd.GoToRecord acDataForm, "frmOperatorSub", acNewRec
    'Forms![frmOperator]![frmOperatorSub].Form
   ' DoCmd.GoToRecord acDataForm, Forms![frmOperator]![frmOperatorSub], acNewRec

Exit_cmdNewRecOperator_Click:
    Exit Sub

Err_cmdNewRecOperator_Click:
    MsgBox Err.Description
    Resume Exit_cmdNewRecOperator_Click


End Sub

Private Sub cmdNewOperator_Click()
On Error GoTo Err_cmdNewOperator_Click


    DoCmd.OpenForm "frmNewOperator"
    DoCmd.GoToRecord acDataForm, "frmNewOperator", acNewRec

Exit_cmdNewOperator_Click:
    Exit Sub

Err_cmdNewOperator_Click:
    MsgBox Err.Description
    Resume Exit_cmdNewOperator_Click

End Sub

Private Sub OperatorID_GotFocus()
OperatorID.Requery
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmOperatorSub"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub chk0_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk1_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk2_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk3_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk4_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk5_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk6_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk7_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk8_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk9_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk10_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub chk11_AfterUpdate()
Me![Privileges] = Me![Text38]
End Sub
Private Sub Form_Current()
'
'   Set status of check boxes according to existing value of field Privileges
```

```
'
Me![chk0] = Bitc(Me![Privileges], 0)   '  Generate orders
Me![chk1] = Bitc(Me![Privileges], 1)   '  Wrapping
Me![chk2] = Bitc(Me![Privileges], 2)   '  ResGauge
Me![chk3] = Bitc(Me![Privileges], 3)   '  Base pipe perf.
Me![chk4] = Bitc(Me![Privileges], 4)   '  Welding
Me![chk5] = Bitc(Me![Privileges], 5)   '  Heat treatment
Me![chk6] = Bitc(Me![Privileges], 6)   '  General assembly
Me![chk7] = Bitc(Me![Privileges], 7)   '  Torque machine
Me![chk8] = Bitc(Me![Privileges], 8)   '  Pressure testing
Me![chk9] = Bitc(Me![Privileges], 9)   '  Quality control
Me![chk10] = Bitc(Me![Privileges], 10) '  Drift of base pipe
Me![chk11] = Bitc(Me![Privileges], 11) '  Allocate parts

End Sub

```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmOrder"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboOrder_AfterUpdate()
'cboOrderLine.Requery
End Sub

Private Sub cboOrder_GotFocus()
'   Terje 29Jan11
'   Filtration of orders
'
'   SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location, tblOrder.CompletedSign
'   FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
'   ORDER BY tblOrder.ProdOrder;

    Dim Sql As String

    Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign] "
    Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID "


    If Me![optionOrderSelect] = 1 Then
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    End If

    Forms![frmOrder]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"


End Sub

Private Sub cboOrderLine_AfterUpdate()
'
'   Run append query to add records in tblAsBuiltBOM
'
On Error GoTo Err_cboOrderLine_AfterUpdate

cboOrderLine.Requery
Forms![frmOrder]![frmOperationTest].Form![cboOperationID].Requery
Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID1].Requery
Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID2].Requery
```

```
Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID3].Requery
Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID4].Requery
Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID5].Requery
Forms![frmOrder]![frmReportSetup].Form![cboSignFromOperationID].Requery

'
'   Set AssemblyID in form Start as reference for Slot opening data queries
'

'    Forms![frmAssembly]![frmOrderLine].Form![chkAllinWO] = 0
    Forms![Start]![txtMinAssemblyID] = 0
    Forms![Start]![txtMaxAssemblyID] = DMax("[AssemblyID]", "tblAssembly")
    Forms![Start]![txtOrderLineID] = Me![cboOrderLine]

    If DLookup("[InclAllLines]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine]) Then
        Forms![Start]![txtInclAllLines] = 360
    Else
        Forms![Start]![txtInclAllLines] = 0
    End If


Exit_cboOrderLine_AfterUpdate:
    Exit Sub

Err_cboOrderLine_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboOrderLine_AfterUpdate

End Sub

Private Sub cboOrderLine_Change()
'
'   Updates values in cboOperation based on teh selected WO
'
Forms![frmOrder]![frmOperationBOM].Form![cboOperationID].Requery
End Sub

Private Sub cboOrderLine_GotFocus()
cboOrderLine.Requery
End Sub

Private Sub cmdAsBuiltBOM_Click()
'
'   Generate new records in tblAsBuiltBOM: AssemblyID, OperationBOMID, Qty
'   For the given work order, it is an OperationBOMID for each occurance of a part in an operation
'   Qty denotes the number of a given part being used in the given operation
'
'   02 Dec 2011 TM
'
On Error GoTo Err_cmdAsBuiltBOM_Click

'   06Dec2011 TM: kontrol på at vikletråd og aksialtråd er merket av bør legges inn her!!!

```

```
Dim dbs As Database
Dim rst1 As Recordset, rst2 As Recordset, rst3 As Recordset
Dim stSql1 As String, stSql2 As String
Dim N As Integer




   stSql1 = "SELECT tblAssembly.OrderLineID, tblAssembly.AssemblyID, "
   stSql1 = stSql1 & "tblOperationBOM.OperationBOMID, tblOperationBOM.Qty, "
   stSql1 = stSql1 & "tblOrderLineBOM.Usage "
   stSql1 = stSql1 & "FROM tblAssembly INNER JOIN (tblOrderLineBOM INNER JOIN "
   stSql1 = stSql1 & "tblOperationBOM ON tblOrderLineBOM.OrderLineBOMID = "
   stSql1 = stSql1 & "tblOperationBOM.OrderLineBOMID) ON tblAssembly.OrderLineID = "
   stSql1 = stSql1 & "tblOrderLineBOM.OrderLineID "
   stSql1 = stSql1 & "WHERE (tblAssembly.OrderLineID)=" & [Forms]![frmOrder]![cboOrderLine]
   stSql1 = stSql1 & " AND (tblOrderLineBOM.Usage)<>2"

   N = 0   '   Counter for each record being appended to tblAsBuiltBOM

   DoCmd.Hourglass True

   Set dbs = CurrentDb
'
'   Open list of all assemblies in work order
'
   Set rst1 = dbs.OpenRecordset(stSql1, dbOpenDynaset)
   Set rst2 = dbs.OpenRecordset("tblAsBuiltBOM", dbOpenDynaset)

   Do While Not rst1.EOF
      stSql3 = "(AssemblyID)=" & rst1!AssemblyID & "AND (OperationBOMID)=" & rst1!OperationBOMID
         If DCount("[AssemblyID]", "tblAsBuiltBOM", stSql3) = 0 Then
            rst2.AddNew
            rst2!AssemblyID = rst1!AssemblyID
            rst2!OperationBOMID = rst1!OperationBOMID
            rst2!Qty = rst1!Qty
            rst2.Update
            N = N + 1
         End If
      rst1.MoveNext
   Loop


   DoCmd.Hourglass False

   rst1.Close
   rst2.Close

   dbs.Close

   If N > 0 Then
      Forms![frmOrder]![frmOrderLineDetails].Form![BOMUpdate] = Now
      MsgBox N & " Records are generated"
```

```
    Else
        MsgBox "Records have been updated"
    End If









'Dim stDocName As String
'
''   dcount("[OrderLineID]","tblOperation","[OrderLineID]=" & [Forms]![frmOrder]![cboOrderLine] & " and [WorkStationID]=1")
'
'   If DCount("[OrderLineID]", "qryAsBuiltBOMList") = 0 Then
'       DoCmd.SetWarnings False
'           stDocName = "qryappAsBuiltBOM"
'           DoCmd.OpenQuery stDocName, acViewNormal, acEdit
'       DoCmd.SetWarnings True
'   Else
'       MsgBox "Records has been updated"
'   End If

Exit_cmdAsBuiltBOM_Click:
    Exit Sub

Err_cmdAsBuiltBOM_Click:
    MsgBox Err.Description
    Resume Exit_cmdAsBuiltBOM_Click

End Sub

Private Sub cmdAsBuiltOperations_Click()
'
'  Generate new records in tblAsBuiltOperation
'  inserting OperationID for each AssemblyID
'  The code must be rerun if more assemblies or new operations are added
'
'  23 Nov 2011 TM
'

On Error GoTo Err_cmdAsBuiltOperations_Click

Dim dbs As Database
Dim rst1 As Recordset, rst2 As Recordset, rst3 As Recordset
Dim stSql1 As String, stSql2 As String, stSql3 As String, stSql4 As String
Dim N As Integer

    stSql1 = "SELECT tblAssembly.AssemblyID, tblAssembly.OrderLineID "
    stSql1 = stSql1 & "FROM tblAssembly "
    stSql1 = stSql1 & "WHERE (tblAssembly.OrderLineID) = " & [Forms]![frmOrder]![cboOrderLine]

    stSql2 = "SELECT tblOperation.OrderLineID, tblOperation.OperationID "
    stSql2 = stSql2 & "FROM tblOperation "
    stSql2 = stSql2 & "WHERE (tblOperation.OrderLineID) = " & [Forms]![frmOrder]![cboOrderLine]

    N = 0   '   Counter for each record being appended to tblAsBuiltOperation

'    DoCmd.Hourglass True

    Set dbs = CurrentDb
'
'   Open list of all assemblies in work order
'
    Set rst1 = dbs.OpenRecordset(stSql1, dbOpenDynaset)
'
'   Open list of operations in work order
'
    Set rst2 = dbs.OpenRecordset(stSql2, dbOpenDynaset)

'
'   loop through each assembly
'
    Do While Not rst1.EOF

    stSql3 = "SELECT tblAsBuiltOperation.AssemblyID, tblAsBuiltOperation.OperationID "
    stSql3 = stSql3 & "FROM tblAsBuiltOperation "
    stSql3 = stSql3 & "WHERE (tblAsBuiltOperation.AssemblyID) = " & rst1!AssemblyID
'
'   Open list of as built operations for given assembly
'
    Set rst3 = dbs.OpenRecordset(stSql3, dbOpenDynaset)

'
'   Loop through each operation for the given assembly
'
        Do While Not rst2.EOF
        stSql4 = "(AssemblyID)=" & rst1!AssemblyID & " AND (OperationID)=" & rst2!OperationID
            If DCount("[AsBuiltOperationID]", "tblAsBuiltOperation", stSql4) = 0 Then
                rst3.AddNew
                rst3!AssemblyID = rst1!AssemblyID
                rst3!OperationID = rst2!OperationID
                rst3.Update
                N = N + 1
            End If
            rst2.MoveNext
        Loop
        rst2.MoveFirst
        rst1.MoveNext
    Loop

'    DoCmd.Hourglass False

    rst1.Close
```

```
   rst2.Close
   rst3.Close

   dbs.Close

   If N > 0 Then
      Forms![frmOrder]![frmOrderLineDetails].Form![OperationUpdate] = Now
      MsgBox N & " Records are generated"
   Else
      MsgBox "Records have been updated"
   End If

Exit_cmdAsBuiltOperations_Click:
   Exit Sub

Err_cmdAsBuiltOperations_Click:
   MsgBox Err.Description
   Resume Exit_cmdAsBuiltOperations_Click

End Sub

Private Sub cmdAsBuiltTest_Click()
'
'   Generate records in tblAsBuiltTest
'   Based on records in tblAsBuiltOperations and tblOperationTest
'
'   24 Nov 2011 TM
'   updated 25 Apr 2012 TM: qrydelAsBuiltTestObsolete
'
On Error GoTo Err_cmdAsBuiltTest_Click

Dim dbs As Database
Dim rst1 As Recordset, rst2 As Recordset, rst3 As Recordset
Dim stSql1 As String, stSql2 As String
Dim N As Integer


   stSql1 = "SELECT tblAsBuiltOperation.AsBuiltOperationID, "
   stSql1 = stSql1 & "tblOperationTest.OperationTestID, tblAssembly.OrderLineID "
   stSql1 = stSql1 & "FROM tblAssembly INNER JOIN (tblOperationTest INNER JOIN "
   stSql1 = stSql1 & "tblAsBuiltOperation ON tblOperationTest.OperationID = "
   stSql1 = stSql1 & "tblAsBuiltOperation.OperationID) ON tblAssembly.AssemblyID = "
   stSql1 = stSql1 & "tblAsBuiltOperation.AssemblyID "
   stSql1 = stSql1 & "WHERE (tblAssembly.OrderLineID)=" & [Forms]![frmOrder]![cboOrderLine]

   N = 0   '   Counter for each record being appended to tblAsBuiltTest

   DoCmd.Hourglass True

'
'   Remove records in tblAsBuiltTest if operation has been changed for given test
'
   DoCmd.SetWarnings False
```

```
    DoCmd.OpenQuery "qrydelAsBuiltTestObsolete", acNormal, acEdit
    DoCmd.SetWarnings True

    Set dbs = CurrentDb
'
'   Open list of all assemblies in work order
'
    Set rst1 = dbs.OpenRecordset(stSql1, dbOpenDynaset)
    Set rst2 = dbs.OpenRecordset("tblAsBuiltTest", dbOpenDynaset)

    Do While Not rst1.EOF
        stSql3 = "(AsBuiltOperationID)=" & rst1!AsBuiltOperationID & " AND (OperationTestID)=" & rst1!OperationTestID
            If DCount("[AsBuiltTestID]", "tblAsBuiltTest", stSql3) = 0 Then
                rst2.AddNew
                rst2!AsBuiltOperationID = rst1!AsBuiltOperationID
                rst2!OperationTestID = rst1!OperationTestID
                rst2.Update
                N = N + 1
            End If
        rst1.MoveNext
    Loop


    DoCmd.Hourglass False

    rst1.Close
    rst2.Close

    dbs.Close

    If N > 0 Then
        Forms![frmOrder]![frmOrderLineDetails].Form![TestsUpdate] = Now
        MsgBox N & " Records are generated"
    Else
        MsgBox "Records have been updated"
    End If


'Dim stDocName As String
'
'    If DCount("[OrderLineID]", "qryAsBuiltTestList") = 0 Then
'        DoCmd.SetWarnings False
'            stDocName = "qryappAsBuiltTest"
'            DoCmd.OpenQuery stDocName, acViewNormal, acEdit
'        DoCmd.SetWarnings True
'    Else
'        MsgBox "Records has been updated"
'    End If




Exit_cmdAsBuiltTest_Click:
    Exit Sub

Err_cmdAsBuiltTest_Click:
    MsgBox Err.Description
    Resume Exit_cmdAsBuiltTest_Click

End Sub

Private Sub cmdNewOrder_Click()
'
'   Open form to add new Order as generated by MfgPro
'
On Error GoTo Err_cmdNewOrder_Click


    DoCmd.OpenForm "frmNewOrder"
    DoCmd.GoToRecord acDataForm, "frmNewOrder", acNewRec

Exit_cmdNewOrder_Click:
    Exit Sub

Err_cmdNewOrder_Click:
    MsgBox Err.Description
    Resume Exit_cmdNewOrder_Click

End Sub

Private Sub cmdNewWorkOrder_Click()
'
'   Open form to add new Order as generated by MfgPro
'
On Error GoTo Err_cmdNewWorkOrder_Click


    DoCmd.OpenForm "frmNewWorkOrder"
    DoCmd.GoToRecord acDataForm, "frmNewWorkOrder", acNewRec

Exit_cmdNewWorkOrder_Click:
    Exit Sub

Err_cmdNewWorkOrder_Click:
    MsgBox Err.Description
    Resume Exit_cmdNewWorkOrder_Click

End Sub

```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_FrmOrderDetail"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Private Sub cboOrderRegSign_GotFocus()
'
'   Check that information has been entered in the fields
'
    If IsNull(Me![ProdOrder]) Then
        MsgBox "Order number not entered!"
        Exit Sub
    End If

    If IsNull(Me![LocationID]) Then
        MsgBox "Location not selected!"
        Exit Sub
    End If

    If IsNull(Me![Customer]) Then
        MsgBox "Customer not entered!"
        Exit Sub
    End If

    If IsNull(Me![Description]) Then
        MsgBox "Description not entered!"
        Exit Sub
    End If

    If IsNull(Me![OrderRegDate]) Then
        Me![OrderRegDate] = Now
    End If

End Sub

Private Sub OrderRegDate_DblClick(Cancel As Integer)
Me![OrderRegDate] = Now
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmOrderLineDetails"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboCsgID_AfterUpdate()
'
'   Set base pipe ID
'
Me![BpID] = Me![txtPipeID]
End Sub

Private Sub cboPipeOD_AfterUpdate()
Me![cboCsgID].Requery
'
'   set default values
'
If IsNull(Me![BpOD]) Then Me![BpOD] = Me![txtPipeOD]
If IsNull(Me![nAxialRods]) Then Me![nAxialRods] = Me![txtAxialRods]
If IsNull(Me![PerfDia]) Then Me![PerfDia] = Me![txtPerfD]
If IsNull(Me![NomScreenOD]) Then Me![NomScreenOD] = Me![BpOD] + 2 * Me![txtWrapWireThichness]


End Sub

Private Sub cboProductTypeID_AfterUpdate()
'
'   Add default values
'
If IsNull(Me![WrapWireMass]) Then Me![WrapWireMass] = Me![txtWrapWireMass]
If IsNull(Me![AxialWireMass]) Then Me![AxialWireMass] = Me![txtAxialWireMass]

End Sub

Private Sub cmdNewAssyRecords_Click()
On Error GoTo Err_cmdNewAssyRecords_Click

'
'   Generate new records in tblAssembly and tblWrap
'
'   03 Dec 2011 TM
'
Dim dbs As Database
Dim rst As Recordset, rst2 As Recordset
Dim tblAssyQty As Integer, Qty As Integer, num_pipes As Integer
Dim NSec As Integer ', NRods As Integer
Dim I As Integer, J As Integer, M As Integer, N As Integer
Dim StartPos As Single, WrapLength As Single, L1 As Single, L2 As Single
Dim Screen As Boolean
Dim WO As String, OLID As Long
Dim stSql As String, stSql1 As String, stSql2 As String, stStatus As String


   If IsNull(Me![OrderLineID]) Then
      MsgBox "Work Order not registered"
      Exit Sub
   End If
   OLID = Nz(Me![OrderLineID], 0)

   If IsNull(Me![WorkOrder]) Then
      MsgBox "Work Order number not registered"
      Exit Sub
   End If
   WO = Me![WorkOrder]

   If Nz(Me![Quantity], 0) < 1 Then
      MsgBox "Quantity not recorded"
      Exit Sub
   End If
   Qty = Nz(Me![Quantity], 0)

'
'  Check if product is screen
'  if so, check screen details
'
   Screen = False
   If IsNull(Me![txtProductTypeScreen]) Then
      MsgBox "Product type not recorded"
      Exit Sub
   ElseIf Me![txtProductTypeScreen] Then
      If Nz(Me![NSections], 0) < 1 Then
         MsgBox "Number of wrap sections not recorded"
         Exit Sub
      End If
      NSec = Me![NSections]
      If DCount("[Section]", "tblWrapLength", "[OrderLineID]=" & OLID) > NSec Then
         MsgBox "Too many Sec. recorded in Wrap Length table"
         Exit Sub
      End If
      If DCount("[StartPos]", "tblWrapLength", "[OrderLineID]=" & OLID) > NSec Then
         MsgBox "Too many StartPos recorded in Wrap Length table"
         Exit Sub
      End If
      If DCount("[WrapLength]", "tblWrapLength", "[OrderLineID]=" & OLID) > NSec Then
         MsgBox "Too many WrapLength recorded in Wrap Length table"
         Exit Sub
      End If
```

```

'   check records in tblWrapLength - nominal start position and wrap length for each section
'   if both lengths are entered for all sections

        If NSec > 1 Then
            If DCount("[Section]", "tblWrapLength", "[OrderLineID]=" & OLID) = NSec Then
                If DCount("[StartPos]", "tblWrapLength", "[OrderLineID]=" & OLID) = NSec Then
                    If DCount("[WrapLength]", "tblWrapLength", "[OrderLineID]=" & OLID) = NSec Then
                        J = 1
                        L2 = 0
                        Do While J <= NSec
                            stSql2 = "(OrderLineID)=" & OLID & " AND (Section)=" & J
                            If IsNull(DLookup("[Section]", "tblWrapLength", stSql2)) Then
                                MsgBox "Section " & J & " is missing"
                                Exit Sub
                            End If
                            L1 = DLookup("[StartPos]", "tblWrapLength", stSql2)
                            If L2 > L1 Then
                                MsgBox "StartPos for section " & J & " is too small!"
                                Exit Sub
                            End If
                            L2 = DLookup("[StartPos]", "tblWrapLength", stSql2) + DLookup("[WrapLength]", "tblWrapLength", stSql2)
                            J = J + 1
                        Loop
                    End If
                End If
            End If
        End If

        If Nz(Me![nAxialRods], 0) < 1 Then
            MsgBox "Number of axial rods not recorded"
            Exit Sub
        End If
        Screen = True
    End If

'
'   Count if any or how many records already in tblAssembly
'
    stSql = "(tblAssembly.OrderLineID) = " & OLID
    tblAssyQty = Nz(DCount("[OrderLineID]", "tblAssembly", stSql), 0)

    Set dbs = CurrentDb
    DoCmd.Hourglass True

    If tblAssyQty = Qty Then
'       MsgBox "Records has already been updated"
'       Exit Sub
    ElseIf tblAssyQty > Qty Then
'       MsgBox "Quantity has been reduced. Records should be deleted. This feature is not developed."
        Me![Quantity] = tblAssyQty
'       Exit Sub
```

```
    ElseIf tblAssyQty < Qty Then
        Set rst = dbs.OpenRecordset("tblAssembly", dbOpenDynaset)
        For num_pipes = tblAssyQty + 1 To Qty
            s_nr = WO & "-" & mk3digit(num_pipes, Qty)
            rst.AddNew
            rst!OrderLineID = OrderLineID
            rst!Serialno = s_nr
            rst.Update
            M = M + 1
        Next num_pipes
        rst.Close
    End If

'
'   Loop through records in tblAssembly and check if record for each wrap section exists in tblWrap
'   If not, add records in tblWrap
'
    If Screen Then
        stSql = "SELECT tblAssembly.OrderLineID, tblAssembly.AssemblyID "
        stSql = stSql & "FROM tblAssembly "
        stSql = stSql & "WHERE(tblAssembly.OrderLineID) = " & OLID
        Set rst = dbs.OpenRecordset(stSql, dbOpenDynaset)
        Set rst2 = dbs.OpenRecordset("tblWrap", dbOpenDynaset)
        rst.MoveFirst
        Do While Not rst.EOF
            I = 0
            Do While I < NSec
                I = I + 1
                stSql1 = "(AssemblyID)=" & rst!AssemblyID & " AND (WrapSection)=" & I
                If DCount("[WrapID]", "tblWrap", stSql1) = 0 Then
'   lookup start pos and lenght for given section
                    stSql2 = "(OrderLineID)=" & OLID & " AND (Section)=" & I
                    StartPos = Nz(DLookup("[StartPos]", "tblWrapLength", stSql2), 0)
                    WrapLength = Nz(DLookup("[WrapLength]", "tblWrapLength", stSql2), 0)
                    rst2.AddNew
                    rst2!AssemblyID = rst!AssemblyID
                    rst2!WrapSection = I
                    rst2!WrapTypeID = 1
                    rst2!WrapStartPos = StartPos
                    rst2!WrapLength = WrapLength
                    rst2.Update
                    N = N + 1
                End If

            Loop
            rst.MoveNext
        Loop
    End If

    rst.Close
    dbs.Close
    DoCmd.Hourglass False

```

```
   If M > 0 Then
      stStatus = M & " assemblys added"
   End If

   If N > 0 Then
      If M > 0 Then
         Status = Status & " and total " & N & " wrap sections added"
      Else
         Status = "Total " & N & " wrap sections added"
      End If
   End If

   If M + N > 0 Then
      Me!AssemblyUpdate = Now
      MsgBox Status
   Else
      MsgBox "Records has been updated"
      If IsNull(Me!AssemblyUpdate) Then
         Me!AssemblyUpdate = Now
      End If
   End If

Exit_cmdNewAssyRecords_Click:
   Exit Sub

Err_cmdNewAssyRecords_Click:
   DoCmd.Hourglass False
'  Write error message to tblBug
   Forms![Start]![txtBug] = "cmdNewAssyRecords_Click: " & Err.Description
   DoCmd.SetWarnings False
   DoCmd.OpenQuery "qryappBug", acNormal, acEdit
   DoCmd.SetWarnings True
'
   MsgBox Err.Description
   Resume Exit_cmdNewAssyRecords_Click

End Sub

Private Sub Option73_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
Me![OLStatusDate] = Now
End Sub

Private Sub Option75_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
Me![OLStatusDate] = Now
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmOrderLineSlot"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub chkInclAllLines_AfterUpdate()

    If Me![chkInclAllLines] Then
        Forms![Start]![txtInclAllLines] = 360
    Else
        Forms![Start]![txtInclAllLines] = 0
    End If

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmPopulateAllocated"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cmdCancel_Click()
DoCmd.Close
End Sub

Private Sub cmdUpdateRecords_Click()
'
'   Add records to tblAllocated
'
    Dim OperatorID As Long
    Dim nRecords As Integer
    Dim dbs As Database
    Dim rst As DAO.Recordset

    OperatorID = Forms![Start]![txtOperatorID]
    If OperatorID = 0 Then
        Exit Sub
    End If

    If IsNull(Me![Serialno]) Then
        MsgBox "Enter Serial no from"
        Me![Serialno].SetFocus
        Exit Sub
    End If

    If IsNull(Me![QtySerialNo]) Then
        MsgBox "Enter Quantity in serial number range"
        Me![QtySerialNo].SetFocus
        Exit Sub
    End If

    Me![Qty] = 1

    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset("tblAllocated", dbOpenDynaset)
    For nRecords = Me![Serialno] + 1 To Me![QtySerialNo] + Me![Serialno] - 1
        rst.AddNew
        rst!OrderLineBOMID = Me![OrderLineBOMID]
        rst!PO = Me![PO]
        rst!POline = Me![POline]
        rst!POship = Me![POship]
        rst!Heat = Me![Heat]
        rst!Qty = 1
        rst!Serialno = nRecords
        rst!AllocatedSign = OperatorID
        rst!AllocatedDate = Now
        rst.Update
    Next nRecords
    rst.Close
    dbs.Close

    Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Requery
    DoCmd.Close
    Forms![frmWareHouse]![lstOrderLineBOM].Requery

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmProgressBar"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub txtProgress_AfterUpdate()

Me![Box1].Width = (Me![txtProgress] - Int(Me![txtProgress])) * Me![Box0].Width
Select Case Int(Me![txtProgress])
Case 1
Me![Box1].BackColor = vbGreen
Case 2
Me![Box1].BackColor = vbBlack
Case 3
Me![Box1].BackColor = vbYellow
Case 4
Me![Box1].BackColor = vbRed
End Select

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmQC"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboAssemblyDateFrom_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub cboAssemblyDateFrom_LostFocus()

    If IsNull(Me![cboAssemblyDateFrom]) Then
        Me![fraStatus] = 1
    End If

End Sub

Private Sub cboAssemblyDateTo_AfterUpdate()

    If Me![cboAssemblyDateFrom] > 0 Then
        Me![fraStatus] = 4
    ElseIf Me![fraStatus] = 4 Then
        Me![fraStatus] = 1
    End If

End Sub

Private Sub cboAssemblyID_AfterUpdate()

'
'   Set default values after assembly has been selected
'

On Error GoTo Err_cboAssemblyID_AfterUpdate

    Dim stSql As String
'
'   Set AssemblyID in form Start as reference for Slot opening data queries
'

'    Forms![frmAssembly]![frmOrderLine].Form![chkAllinWO] = 0
    Forms![Start]![txtMinAssemblyID] = Me![cboAssemblyID]
    Forms![Start]![txtMaxAssemblyID] = Me![cboAssemblyID]

```

```
'
'   Requery list box to reflect selected assembly
'   Select record in frmWrapDetail form based on selected assembly
'

    lstWrapID.Requery
    Me![lstWrapID] = DMin("[WrapID]", "tblWrap", "[AssemblyID]=" & Me![cboAssemblyID])

    lstWrapIDHardness.Requery
    Me![lstWrapIDHardness] = Me![lstWrapID]


'
'   Select assemblies in report tab
'
    Me![cboAssyFrom] = Me![txtSerialno]
    Me![cboAssyTo] = Me![txtSerialno]
    Me![fraSelectAssy] = 1




Exit_cboAssemblyID_AfterUpdate:
    Exit Sub

Err_cboAssemblyID_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboAssemblyID_AfterUpdate

End Sub

Private Sub cboAssemblyID_GotFocus()
'cboAssemblyID.Requery
End Sub



Private Sub cboAssyFrom_Click()
    Me![fraSelectAssy] = 3
End Sub

Private Sub cboAssyTo_Click()
    Me![fraSelectAssy] = 3
End Sub

Private Sub cboOrder_AfterUpdate()
'cboOrderLine.Requery
End Sub

Private Sub cboOrder_GotFocus()
'   Terje 29Jan11
'   Filtration of orders
'
```

```
'   SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location, tblOrder.CompletedSign
'   FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
'   ORDER BY tblOrder.ProdOrder;

    Dim Sql As String

    Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign], [tblLocation.LocationID] "
    Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID "


    If Me![optionOrderSelect] = 1 Then
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    End If

    Forms![frmQC]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"


End Sub

Private Sub cboOrderLine_AfterUpdate()
'
'   If emty, set Order
'
On Error GoTo Err_cboOrderLine_AfterUpdate

    If IsNull(Me![cboOrder]) Then
        Me![cboOrder] = DLookup("[OrderID]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine])
    End If


'
'   Set parameters for slot data queries in Start form
'
    Forms![Start]![txtOrderLineID] = Me![cboOrderLine]
    Forms![Start]![txtLocationID] = Me![txtLocationID]

    If DLookup("[InclAllLines]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine]) Then
        Forms![Start]![txtInclAllLines] = 360
    Else
        Forms![Start]![txtInclAllLines] = 0
    End If

    Forms![Start]![txtMinAssemblyID] = 0
    Forms![Start]![txtMaxAssemblyID] = 0
'
'   Requery Assembly, cboWrapID and clear combo boxes
'
    cboAssemblyID.Requery
    Me![cboAssemblyID] = Null
    lstOrderLineBOM.Requery
    cboWrapID.Requery
    cboAssyFrom.Requery
    cboAssyTo.Requery
    cboOperationID.Requery
    Me![cboReportID] = Null
    Me![cboAssemblyDateFrom] = Null
    Me![cboAssemblyDateTo] = Null
    Me![cboShipDateFrom] = Null
    Me![cboShipDateTo] = Null
    Me![cboOperationID] = Null
    Me![fraStatus] = 1
    Me![fraOperations] = 2
    Me![cboAssemblyDateFrom].Requery
    Me![cboAssemblyDateTo].Requery
    Me![cboShipDateFrom].Requery
    Me![cboShipDateTo].Requery

'
'   Select assemblies in report tab
'
    Me![cboAssyFrom] = DMin("[Serialno]", "tblAssembly", "[OrderLineID] = " & Me![cboOrderLine])
    Me![cboAssyTo] = DMax("[Serialno]", "tblAssembly", "[OrderLineID] = " & Me![cboOrderLine])
    Me![fraSelectAssy] = 2


Exit_cboOrderLine_AfterUpdate:
    Exit Sub

Err_cboOrderLine_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboOrderLine_AfterUpdate

End Sub

Private Sub cboOrderLine_GotFocus()
'   SELECT tblOrderLine.OrderLineID, tblOrderLine.WorkOrder, tblOrderLine.PartNo, tblOrderLine.Description, tblOrder.CompletedSign
'   FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID
'   WHERE (((tblOrder.CompletedSign)=0));
On Error GoTo Err_cboOrderLine_GotFocus

    Dim Sql As String

    Sql = "SELECT [OrderLineID], [WorkOrder], [PartNo], [tblOrderLine.Description], [CompletedSign] "
    Sql = Sql & "FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID "
    If Me![optionOrderSelect] = 1 Then
'   noe muffens når denne linjen ikke inkluderes - WHERE mangler
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    Else
        Sql = Sql & "WHERE [tblOrder.CompletedSign] >= 0 "
    End If

```

```
   If Not IsNull(Me![cboOrder]) Then
      Sql = Sql & "AND [tblOrder.OrderID] = " & Me![cboOrder]
   End If

   Forms![frmQC]![cboOrderLine].RowSource = Sql & " ORDER BY tblOrderLine.WorkOrder;"

Exit_cboOrderLine_GotFocus:
   Exit Sub

Err_cboOrderLine_GotFocus:
   MsgBox Err.Description
   Resume Exit_cboOrderLine_GotFocus

End Sub

Private Sub cboReportID_GotFocus()
cboReportID.Requery
End Sub

Private Sub cboShipDateFrom_LostFocus()
   If IsNull(Me![cboShipDateFrom]) Then
      Me![fraStatus] = 1
   End If
End Sub

Private Sub cboShipDateTo_AfterUpdate()

   If Me![cboShipDateFrom] > 0 Then
      Me![fraStatus] = 6
   ElseIf Me![fraStatus] = 6 Then
      Me![fraStatus] = 1
   End If

End Sub

Private Sub cboWrapID_AfterUpdate()
'
'  Set AssemblyID and lstWrapID to select the corect record in frmQCWrap
'
   Me![cboAssemblyID] = Me![cboWrapID].Column(2)
   Me![lstWrapID] = Me![cboWrapID]
   lstWrapID.Requery

End Sub

Private Sub cboWrapIDHardness_AfterUpdate()
'
'  Set AssemblyID and lstWrapIDHardness to select the corect record in frmQCWrap
'
   Me![cboAssemblyID] = Me![cboWrapIDHardness].Column(2)
   Me![lstWrapIDHardness] = Me![cboWrapIDHardness]
   lstWrapIDHardness.Requery

End Sub
```

```

Private Sub cmdPullTestReport_Click()
DoCmd.OpenReport "rptPullTest2", acViewPreview
End Sub

Private Sub cmdrepAllocated_Click()
    DoCmd.OpenReport "rptAllocated", acViewPreview
End Sub

Private Sub cmdrepTest_Click()
On Error GoTo cmdrepTest_Click_Err
'
'   open report for assembly by operation
'
    Dim stSql As String


    If IsNull(Me![cboAssyFrom]) Then
        MsgBox "Select Assembly"
        Exit Sub
    End If
    If IsNull(Me![cboAssyTo]) Then
        MsgBox "Select Assembly"
        Exit Sub
    End If
    If IsNull(Me![cboReportID]) Then
        MsgBox "Select Report"
        Exit Sub
    End If

    stSql = "[OrderLineID]=" & Me![cboOrderLine] & " and [ReportID]=" & Me![cboReportID]
    If DCount("[ReportID]", "tblReport", stSql) = 0 Then
        MsgBox "Select Report"
        Exit Sub
    End If

    Select Case Me![fraStatus]
    Case 4
        DoCmd.OpenReport "rptTestsColComplDate", acViewPreview
        Exit Sub
    Case 6
        DoCmd.OpenReport "rptTestsColShipDate", acViewPreview
        Exit Sub
    Case Else
        DoCmd.OpenReport "rptTestsCol", acViewPreview
        Exit Sub
    End Select


cmdrepTest_Click_Exit:
    Exit Sub

cmdrepTest_Click_Err:
    MsgBox Error$
    Resume cmdrepTest_Click_Exit

End Sub

Private Sub cmdrptAssy_Click()
On Error GoTo cmdrptAssy_Click_Err
'
'   open report based on selections
'

    If IsNull(Me![cboAssyFrom]) Then
        MsgBox "Select Assembly"
        Exit Sub
    End If
    If IsNull(Me![cboAssyTo]) Then
        MsgBox "Select Assembly"
        Exit Sub
    End If

    If Me![fraOperations] = 1 Then
        Select Case Me![fraStatus]
        Case 4
            DoCmd.OpenReport "rptOperationsComplDate", acViewPreview
            Exit Sub
        Case 6
            DoCmd.OpenReport "rptOperationsShipDate", acViewPreview
            Exit Sub
        Case Else
            DoCmd.OpenReport "rptOperations", acViewPreview
            Exit Sub
        End Select
    End If

    If IsNull(Me![cboOperationID]) Then
        MsgBox "Select Operation"
        Exit Sub
    End If

    Select Case Me![fraStatus]
    Case 4
        DoCmd.OpenReport "rptOperationComplDate", acViewPreview
        Exit Sub
    Case 6
        DoCmd.OpenReport "rptOperationShipDate", acViewPreview
        Exit Sub
    Case Else
        DoCmd.OpenReport "rptOperation", acViewPreview
        Exit Sub
    End Select

cmdrptAssy_Click_Exit:
    Exit Sub

cmdrptAssy_Click_Err:
   MsgBox Error$
   Resume cmdrptAssy_Click_Exit

End Sub

Private Sub cmdrptAssyAllOperations_Click()
   DoCmd.OpenReport "rptOperations", acViewPreview

End Sub

Private Sub cmdSlotDetail_Click()
On Error GoTo cmdSlotDetail_Click_Err
'
'   open report for assembly by operation
'
   If IsNull(Me![cboAssyFrom]) Then
      MsgBox "Select Assembly"
      Exit Sub
   End If
   If IsNull(Me![cboAssyTo]) Then
      MsgBox "Select Assembly"
      Exit Sub
   End If

   Select Case Me![fraStatus]
   Case 4
      DoCmd.OpenReport "rptSlotDetailComplDate", acViewPreview
      Exit Sub
   Case 6
      DoCmd.OpenReport "rptSlotDetailShipDate", acViewPreview
      Exit Sub
   Case Else
      DoCmd.OpenReport "rptSlotDetail", acViewPreview
      Exit Sub
   End Select


cmdSlotDetail_Click_Exit:
   Exit Sub

cmdSlotDetail_Click_Err:
   MsgBox Error$
   Resume cmdSlotDetail_Click_Exit

End Sub

'------------------------------------------------------------
' cmdSlotDetails_Click
'
'------------------------------------------------------------
Private Sub cmdSlotDetails_Click()
On Error GoTo cmdSlotDetails_Click_Err

   DoCmd.OpenForm "frmSlotResult", acNormal, "", "", , acNormal
```

```


cmdSlotDetails_Click_Exit:
    Exit Sub

cmdSlotDetails_Click_Err:
    MsgBox Error$
    Resume cmdSlotDetails_Click_Exit

End Sub

Private Sub fraSelectAssy_AfterUpdate()
'
'   Set values in cbo boxes based on selection
'

    If IsNull(Me![cboOrderLine]) Then
        MsgBox "Work Order ID not selected"
        Me![cboOrderLine].SetFocus
        Exit Sub
    End If

    Select Case Me![fraSelectAssy]
        Case 1
            If IsNull(Me![cboAssemblyID]) Then
                MsgBox "Serial number not selected"
                Me![fraSelectAssy] = 2
                Exit Sub
            End If
            Me![cboAssyFrom] = Me![txtSerialno]
            Me![cboAssyTo] = Me![txtSerialno]
        Case 2
            Me![cboAssyFrom] = DMin("[Serialno]", "tblAssembly", "[OrderLineID]=" & Me![cboOrderLine])
            Me![cboAssyTo] = DMax("[Serialno]", "tblAssembly", "[OrderLineID]=" & Me![cboOrderLine])
        Case 3

    End Select

End Sub

Private Sub lstOrderLineBOM_AfterUpdate()
'
'   Unhide text boxes for entering of wrap wire width and standard deviation when wrap wire is selected
'   hide text boxes wehn other part is selected
'
'   TM 20120106
'

    If DLookup("[Usage]", "tblOrderLineBOM", "[OrderLineBOMID]=" & Me![lstOrderLineBOM]) = 4 Then
        Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireWidt].Visible = True
        Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireStDev].Visible = True
    Else
        Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![Traceability].SetFocus
        Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireWidt].Visible = False
        Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireStDev].Visible = False
    End If


End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmQCWirePullTest"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboWirePullTestSign_AfterUpdate()
Me![WirePullTestDate] = Now
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmScreen"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboOrder_AfterUpdate()
cboOrderLine.Requery
End Sub

Private Sub cboOrder_GotFocus()
'   Terje 10.10.14
'   Filtration of orders
    Dim Sql As String

    Sql = "SELECT [OrderID], [ProdOrder], [Description], [tblLocation.LocationID], [Location], [CompletedSign] "
    Sql = Sql & "FROM tblLocation INNER JOIN tblOrder ON tblLocation.LocationID = tblOrder.LocationID "


    If Me![optionOrderSelect] = 1 Then
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    End If

    Forms![frmScreen]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"


End Sub

Private Sub cboOrderLine_AfterUpdate()
ScreenID.Requery
End Sub

Private Sub cmdWrapping_Click()

'Open form for Wrapping details

On Error GoTo Err_cmdWrapping_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmWrappingOrderLine"

    stLinkCriteria = "[OrderLineID]=" & Me![cboOrderLine]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdWrapping_Click:
    Exit Sub

Err_cmdWrapping_Click:
    MsgBox Err.Description
    Resume Exit_cmdWrapping_Click

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmShip"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub ShipDate_DblClick(Cancel As Integer)
'
'   Set todays date if assembly is signed off
'
    If Not IsNull(Me![OperatorSign]) Then
        Me![ShipDate] = Int(Now())
    End If

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmSlotFile"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub Angle_AfterUpdate()

    If Nz(Me![SlotSign]) Then
        If MsgBox("Enter signature after modification", vbOKCancel, "ResLog") = 1 Then
            Me![SlotDate] = Now()
            Me![SlotSign] = ""
        Else
            Me![Angle] = Me![Temp]
        End If
    End If

    Me![Temp] = ""

End Sub

Private Sub Angle_Change()
Me![Temp] = Me![Angle]
End Sub

Private Sub cboSlotSign_AfterUpdate()
'
'   Use current date and time if SlotDate is Null
'
    If IsNull(Me![SlotDate]) Then
    Me![SlotDate] = Now
    End If

End Sub

Private Sub Form_DblClick(Cancel As Integer)
'Open form for slot details
On Error GoTo Err_Form_DblClick

    Dim stDocName As String
    Dim stLinkCriteria As String

    Forms![frmSlotResult]![txtSlotFileID] = Me![SlotFileID]

    stDocName = "frmSlotFileDetail"

    stLinkCriteria = "[SlotFileID]=" & Me![SlotFileID]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Form_DblClick:
    Exit Sub

Err_Form_DblClick:
    MsgBox Err.Description
    Resume Exit_Form_DblClick

End Sub


Private Sub SlotDate_DblClick(Cancel As Integer)
    Me![SlotDate] = Now()
End Sub

Private Sub SlotFileStatus_AfterUpdate()
On Error GoTo Err_SlotFileStatus_AfterUpdate

Dim Status

    Status = Me![SlotFileStatus]

    If Nz(Me![SlotSign]) Then
        If MsgBox("Enter signature after modification", vbOKCancel, "ResLog") = 1 Then
            Me![SlotDate] = Now()
            Me![SlotSign] = ""
        Else
            Me![SlotFileStatus] = Not (Status)
        End If
    End If

Exit_SlotFileStatus_AfterUpdate:
    Exit Sub

Err_SlotFileStatus_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_SlotFileStatus_AfterUpdate

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmSlotFileImport"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database



Private Sub cmd_getfil_Click()
Dim fil_sti As String, kun_sti As String, kun_fil As String
Dim slash_pos As Integer, path_pos As Integer

cdlg.ShowOpen
fil_sti = cdlg.FileName

If fil_sti <> "" Then
    slash_pos = 1
    Do While slash_pos > 0
        path_pos = slash_pos
        slash_pos = InStr(slash_pos + 1, fil_sti, "\")
    Loop
    If path_pos > 1 Then
        txtFileNameUnb = Right(fil_sti, Len(fil_sti) - path_pos)
        cboPath = Left(fil_sti, path_pos)
        txtFileNameUnbx = fil_sti
        'MsgBox "sti : " & Left(fil_sti, path_pos)
        'MsgBox "fil : " & Right(fil_sti, Len(fil_sti) - path_pos)
    End If
End If

End Sub

Private Sub LoadSlotDataFile_Click()
On Error GoTo Err_LoadSlotDataFile_Click

    Dim I As Integer
    Dim AssemblyID As Long, OrderLineID As Long

    Forms![frmSlotResult]![txtFile] = Me![txtFileNameUnb]
    Forms![frmSlotResult]![txtPath] = Me![cboPath]

    AssemblyID = Forms![frmSlotResult]![cboAssemblyID]
    OrderLineID = [Forms]![Start]![txtOrderLineID]
    I = SlotAngle(Me![txtFileNameUnbx], OrderLineID, AssemblyID)
    DoCmd.Close acForm, "frmSlotFileImport"
```

```

Exit_LoadSlotDataFile_Click:
    Exit Sub

Err_LoadSlotDataFile_Click:
    MsgBox Err.Description
    Resume Exit_LoadSlotDataFile_Click

End Sub

Private Sub txtFileNameUnb_AfterUpdate()
Forms![frmProduction]![ScreenID] = Me![txtScreenID]
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmSlotImportAngle"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cmdSlotImportAngleOK_Click()
Dim I As Integer
Dim PathFileName As String
Dim OrderLineID As Long, AssemblyID As Long

    PathFileName = Forms![frmSlotImportAngle]![txtPathFileName]
    OrderLineID = Forms![frmSlotImportAngle]![txtOrderLineID]
    AssemblyID = Forms![frmSlotImportAngle]![txtAssemblyID]
    Forms![frmSlotResult]![txtAngle] = Me![cboAngle]
    DoCmd.Close acForm, "frmSlotImportAngle"

    I = SlotImport(PathFileName, OrderLineID, AssemblyID)

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmSlotLarge"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboSlotManSign_GotFocus()
'
'   Check that type of modification has been selected
'

    If Me![fraSlotMod] < 1 Then
        MsgBox "Select type of slot modification!"
        Me![fraSlotMod].SetFocus
        Exit Sub
    End If

    If IsNull(Me![SlotManDate]) Then
        Me![SlotManDate] = Now()
    End If

End Sub

Private Sub fraSlotMod_AfterUpdate()
'
'   If slot modification is done and ticked off, the slot on each side
'   are marked as "Next" to the actual slot.
'   This causes the slot to apear in the large slot list and evetuell
'   modification to this slot can also be recorded.
'   A verification is done to make sure the actual slots are actual neighbour
'   slots by comparing slot positions.
'
Dim dbs As DAO.Database
Dim rst As DAO.Recordset
Dim stSql As String

    If Me![fraSlotMod] = 2 Or Me![fraSlotMod] = 3 Then
        Set dbs = CurrentDb
        stSql = "SELECT tblSlotData.SlotID, tblSlotData.SlotFileID, "
        stSql = stSql & "tblSlotData.SlotPos, tblSlotData.SlotMod "
        stSql = stSql & "FROM tblSlotData "
        stSql = stSql & "WHERE (tblSlotData.SlotID) >= " & Me![SlotID] - 1
        stSql = stSql & " AND (tblSlotData.SlotID) <= " & Me![SlotID] + 1
        stSql = stSql & " AND (tblSlotData.SlotFileID) = " & Me![SlotFileID]
        Set rst = dbs.OpenRecordset(stSql, dbOpenDynaset)
        rst.MoveFirst
        If rst!SlotID = (Me![SlotID] - 1) Then
            If (Me![SlotPos] - rst!SlotPos) < 5 Then
                If rst!SlotMod < 2 Then     '   Only if slot is not modified
                    rst.Edit
                    rst!SlotMod = 4
                    rst.Update
                End If
            End If
        End If
        rst.MoveLast
        If rst!SlotID = (Me![SlotID] + 1) Then
            If (rst!SlotPos - Me![SlotPos]) < 5 Then
                If rst!SlotMod < 2 Then     '   Only if slot is not modified
                    rst.Edit
                    rst!SlotMod = 4
                    rst.Update
                End If
            End If
        End If
    rst.Close
    dbs.Close
    Forms![frmSlotResult]![cboAssemblyID].Requery
    End If



End Sub

Private Sub SlotManDate_DblClick(Cancel As Integer)
Me![SlotManDate] = Now
End Sub

Private Sub SlotWidth_AfterUpdate()
'    Me![fraSlotMod] = 2
    Me![SlotManDate] = Now
    If Me![SlotOrg] <> Me![SlotWidth] Then
        Me![SlotChanged] = -1
    Else
        Me![SlotChanged] = 0
    End If
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmSlotOutOfSpec"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub Combo13_AfterUpdate()

    If Nz(Me![SlotManDate], 0) = 0 Then
        Me![SlotManDate] = Now()
    End If

End Sub

Private Sub Form_AfterUpdate()

    If (Nz(Me![SlotManSign], 0) = 0) And (Nz(Me![SlotMan], 0) > 0) Then
        MsgBox "Enter signature after update of slot width"
    End If

End Sub

Private Sub SlotManDate_DblClick(Cancel As Integer)
    Me![SlotManDate] = Now()
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmSlotResult"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboAssemblyID_AfterUpdate()
On Error GoTo Err_cboAssemblyID_AfterUpdate
'
'   Set AssemblyID in frmStart as this is being used by queries
'
    Forms![Start]![txtMinAssemblyID] = Me![cboAssemblyID]
    Forms![Start]![txtMaxAssemblyID] = Me![cboAssemblyID]
    Forms![Start]![txtOperationNo] = Me![txtOperationNo]
'
'   Lock/unlock forms and controls in frmSlotResult
'
    Forms![frmSlotResult]![frmSlotLarge].Locked = Me![txtOperationLocked]
    Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Locked = Me![txtOperationLocked]
    Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Enabled = Not Me![txtOperationLocked]

    Forms![frmSlotResult]![frmSlotFile].Locked = Me![txtOperationLocked]
    Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Locked = Me![txtOperationLocked]
    Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Enabled = Not Me![txtOperationLocked]

Exit_cboAssemblyID_AfterUpdate:
    Exit Sub

Err_cboAssemblyID_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboAssemblyID_AfterUpdate

End Sub

Private Sub cboAssemblyID_GotFocus()
cboAssemblyID.Requery
End Sub

Private Sub cboOrder_AfterUpdate()
Forms![Start]![txtLocationID] = Me![txtLocationID]

End Sub

Private Sub cboOrder_GotFocus()
```

```
'   Terje 16Nov12
'   Filtration of orders
'
'   SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location, tblOrder.CompletedSign
'   FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
'   ORDER BY tblOrder.ProdOrder;

    Dim Sql As String

    Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign], [tblLocation.LocationID] "
    Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID "


    If Me![optionOrderSelect] = 1 Then
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    End If

    Forms![frmSlotResult]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"

End Sub

Private Sub cboOrderLine_AfterUpdate()
'
'   If emty, set Order
'
On Error GoTo Err_cboOrderLine_AfterUpdate

    If IsNull(Me![cboOrder]) Then
        Me![cboOrder] = DLookup("[OrderID]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine])
    End If
'
'   Set parameters for slot data queries in Start form
'
    Forms![Start]![txtOrderLineID] = Me![cboOrderLine]
    Forms![Start]![txtLocationID] = Me![txtLocationID]

    If DLookup("[InclAllLines]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine]) Then
        Forms![Start]![txtInclAllLines] = 360
    Else
        Forms![Start]![txtInclAllLines] = 0
    End If

    Forms![Start]![txtMinAssemblyID] = 0
    Forms![Start]![txtMaxAssemblyID] = 0
'
'   Requery Assembly and clear combo boxes
'
    cboAssemblyID.Requery
    Me![cboAssemblyID] = Null

Exit_cboOrderLine_AfterUpdate:
    Exit Sub

Err_cboOrderLine_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboOrderLine_AfterUpdate

End Sub

Private Sub cboOrderLine_GotFocus()
On Error GoTo Err_cboOrderLine_GotFocus
'
'   Set sql statement based on selected orders
'
    Dim Sql As String

    Sql = "SELECT [OrderLineID], [WorkOrder], [PartNo], [tblOrderLine.Description], [CompletedSign] "
    Sql = Sql & "FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID "
    If Me![optionOrderSelect] = 1 Then
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    Else
        Sql = Sql & "WHERE [tblOrder.CompletedSign] >= 0 "
    End If

    If Not IsNull(Me![cboOrder]) Then
        Sql = Sql & "AND [tblOrder.OrderID] = " & Me![cboOrder]
    End If

    Forms![frmSlotResult]![cboOrderLine].RowSource = Sql & " ORDER BY tblOrderLine.WorkOrder;"

Exit_cboOrderLine_GotFocus:
    Exit Sub

Err_cboOrderLine_GotFocus:
    MsgBox Err.Description
    Resume Exit_cboOrderLine_GotFocus

End Sub

Private Sub cmdImportSlotData_Click()
On Error GoTo Err_cmdImportSlotData_Click
'   Open frmSlotFileImport for selection of ResGauge data file
'   Record must not be locked

    Dim stDocName As String
    Dim stLinkCriteria As String

    If Me![txtOperationLocked] Then
        MsgBox "Form is locked. Go to Results and unlock."
        Exit Sub
    End If

    stDocName = "frmSlotFileImport"
```

```
   DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdImportSlotData_Click:
   Exit Sub

Err_cmdImportSlotData_Click:
   MsgBox Err.Description
   Resume Exit_cmdImportSlotData_Click

End Sub

Private Sub cmdImportSlotData2007_Click()
'Requires reference to Microsoft Office 12.0 Object Library.

   Dim fDialog As Office.FileDialog
   Dim varFile As Variant, strFile As String
   Dim slash_pos As Integer, path_pos As Integer, I As Integer
   Dim OrderLineID As Long


   Set fDialog = Application.FileDialog(msoFileDialogFilePicker)
   With fDialog
     .AllowMultiSelect = False
     .Title = "Please select a ResGauge log file"
     .Filters.Clear
     .Filters.Add "ResGauge data", "*.txt"
      If .Show = True Then
         For Each varFile In .SelectedItems
            strFile = varFile
         Next
      Else
         MsgBox "You clicked Cancel in the file dialog box. No data has been imported."
         Exit Sub
      End If
   End With

   If IsNull(varFile) Then
      MsgBox "Error in slot file selection"
      Exit Sub
   Else
      slash_pos = 1
      Do While slash_pos > 0
         path_pos = slash_pos
         slash_pos = InStr(slash_pos + 1, strFile, "\")
      Loop
      If path_pos > 1 Then
         Me![txtFile] = Right(strFile, Len(strFile) - path_pos)
         Me![txtPath] = Left(strFile, path_pos)
      End If
   End If

   OrderLineID = [Forms]![Start]![txtOrderLineID]
   I = SlotAngle(strFile, OrderLineID, Me![cboAssemblyID])

```

```

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmWareHouse"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboAssemblyID_AfterUpdate()

'
'   Set default values after assembly has been selected
'

On Error GoTo Err_cboAssemblyID_AfterUpdate

    Dim stSql As String
'
'   Set AssemblyID in form Start as reference for Slot opening data queries
'

'    Forms![frmAssembly]![frmOrderLine].Form![chkAllinWO] = 0
    Forms![Start]![txtMinAssemblyID] = Me![cboAssemblyID]
    Forms![Start]![txtMaxAssemblyID] = Me![cboAssemblyID]

'
'   Requery list box to reflect selected assembly
'   Select record in frmWrapDetail form based on selected assembly
'

    lstWrapID.Requery
    Me![lstWrapID] = DMin("[WrapID]", "tblWrap", "[AssemblyID]=" & Me![cboAssemblyID])

    lstWrapIDHardness.Requery
    Me![lstWrapIDHardness] = Me![lstWrapID]

'
'   Select assemblies in report tab
'
    Me![cboAssyFrom] = Me![cboAssemblyID]
    Me![cboAssyTo] = Me![cboAssemblyID]
    Me![fraSelectAssy] = 1




Exit_cboAssemblyID_AfterUpdate:
    Exit Sub

Err_cboAssemblyID_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboAssemblyID_AfterUpdate

End Sub

Private Sub cboAssemblyID_GotFocus()
'cboAssemblyID.Requery
End Sub

Private Sub cboOrder_AfterUpdate()
'cboOrderLine.Requery
End Sub

Private Sub cboOrder_GotFocus()
'   Terje 29Jan11
'   Filtration of orders
'
'   SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location, tblOrder.CompletedSign
'   FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
'   ORDER BY tblOrder.ProdOrder;

    Dim Sql As String

    Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign], [tblLocation.LocationID] "
    Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID "


    If Me![optionOrderSelect] = 1 Then
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    End If

    Forms![frmWareHouse]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"


End Sub

Private Sub cboOrderLine_AfterUpdate()
'
'   If emty, set Order
'
On Error GoTo Err_cboOrderLine_AfterUpdate

    If IsNull(Me![cboOrder]) Then
        Me![cboOrder] = DLookup("[OrderID]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine])
    End If


    lstOrderLineBOM.Requery
```

```


Exit_cboOrderLine_AfterUpdate:
    Exit Sub

Err_cboOrderLine_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboOrderLine_AfterUpdate

End Sub

Private Sub cboOrderLine_GotFocus()
'   SELECT tblOrderLine.OrderLineID, tblOrderLine.WorkOrder, tblOrderLine.PartNo, tblOrderLine.Description, tblOrder.CompletedSign
'   FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID
'   WHERE (((tblOrder.CompletedSign)=0));
On Error GoTo Err_cboOrderLine_GotFocus

    Dim Sql As String

    Sql = "SELECT [OrderLineID], [WorkOrder], [PartNo], [tblOrderLine.Description], [CompletedSign] "
    Sql = Sql & "FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID "
    If Me![optionOrderSelect] = 1 Then
'   noe muffens når denne linjen ikke inkluderes - WHERE mangler
        Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
    Else
        Sql = Sql & "WHERE [tblOrder.CompletedSign] >= 0 "
    End If

    If Not IsNull(Me![cboOrder]) Then
        Sql = Sql & "AND [tblOrder.OrderID] = " & Me![cboOrder]
    End If

    Forms![frmWareHouse]![cboOrderLine].RowSource = Sql & " ORDER BY tblOrderLine.WorkOrder;"

Exit_cboOrderLine_GotFocus:
    Exit Sub

Err_cboOrderLine_GotFocus:
    MsgBox Err.Description
    Resume Exit_cboOrderLine_GotFocus

End Sub

Private Sub cmdrepAllocated_Click()
    DoCmd.OpenReport "rptAllocated", acViewPreview
End Sub




Private Sub lstOrderLineBOM_AfterUpdate()
```

```
'
'   Unhide text boxes for entering of wrap wire width and standard deviation when wrap wire is selected
'   hide text boxes wehn other part is selected
'
'   TM 20120106
'
    lstOrderLineBOM.Requery

    If DLookup("[Usage]", "tblOrderLineBOM", "[OrderLineBOMID]=" & Me![lstOrderLineBOM]) = 4 Then
        Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireWidt].Visible = True
        Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireStDev].Visible = True
    Else
        Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![Traceability].SetFocus
        Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireWidt].Visible = False
        Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireStDev].Visible = False
    End If

End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmWirePullTest"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub Force_AfterUpdate()
Me![TestMachine] = 1
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmWrapDetail"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cboAxialWireID_AfterUpdate()
Me![cboWrapWireID].Requery
End Sub

Private Sub cboWrapSign_AfterUpdate()
'
'   Check if all normal sections has been wrapped and
'   copy date and sign from last normal section to (tblAssembly) tblAsBuiltOperation
'
On Error GoTo Err_cboWrapSign_AfterUpdate

    Dim stSql As String, stSql1 As String, stSql2 As String
    Dim Status As Boolean
    Dim WireMass As Single
    Dim AssemblyID As Long, OperationBOMID As Long
    Dim dbs As Database
    Dim rst As DAO.Recordset

    If Me!AxialWireAsBuiltBOMID > 0 And Me!WrapWireAsBuiltBOMID > 0 Then
        GoTo UpdateAsBuiltOperation
    End If

    stSql = "SELECT tblAsBuiltBOM.AsBuiltBOMID, tblAsBuiltBOM.AssemblyID, "
    stSql = stSql & "tblAsBuiltBOM.OperationBOMID, tblAsBuiltBOM.AllocatedID, "
    stSql = stSql & "tblAsBuiltBOM.Qty, tblAsBuiltBOM.Scrap, tblOrderLineBOM.Usage, "
    stSql = stSql & "tblOperationBOM.HideOperationBOM, tblOrderLineBOM.HideOLBOM, "
    stSql = stSql & "tblAsBuiltBOM.RecordedTime, tblAsBuiltBOM.StatusChangeTime "
    stSql = stSql & "FROM tblOrderLineBOM INNER JOIN (tblAsBuiltBOM INNER JOIN "
    stSql = stSql & "tblOperationBOM ON tblAsBuiltBOM.OperationBOMID = "
    stSql = stSql & "tblOperationBOM.OperationBOMID) ON tblOrderLineBOM.OrderLineBOMID = "
    stSql = stSql & "tblOperationBOM.OrderLineBOMID "
    stSql = stSql & "WHERE (tblAsBuiltBOM.AssemblyID)=" & Me![AssemblyID]
    stSql = stSql & " AND (tblOperationBOM.HideOperationBOM)=False "
    stSql = stSql & "AND (tblOrderLineBOM.HideOLBOM)=False "
    stSql1 = stSql & "AND (tblOrderLineBOM.Usage)=4 "
    stSql2 = stSql & "AND (tblOrderLineBOM.Usage)=5 "

    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset(stSql1, dbOpenDynaset)

    If rst.RecordCount = 0 Then
        MsgBox "As built BOM has not been properly populated"
        rst.Close
        dbs.Close
        Exit Sub
    End If

    WireMass = Round(Me!WrapWireMass * Me!NomScreenOD * Pi() * Me!WrapLength /
Me!WrapPitch / 1000, 1)

    If Me!cboWrapTypeID = 1 Then
        Status = False
    Else
        Status = True
    End If

    If Me!WrapWireAsBuiltBOMID > 0 Then
        GoTo UpdateAxialWire
    End If

    rst.MoveFirst

    If IsNull(rst!AllocatedID) Then
        rst.Edit
        rst!AllocatedID = Me!cboWrapWireID
        rst!Scrap = Status
        If Status Then
            rst!StatusChangeTime = Now
        End If
        rst!Qty = WireMass
        rst!RecordedTime = Now
        rst.Update
    Else
        AssemblyID = Me!AssemblyID
        OperationBOMID = rst!OperationBOMID
        rst.AddNew
        rst!AssemblyID = AssemblyID
        rst!OperationBOMID = OperationBOMID
        rst!AllocatedID = Me!cboWrapWireID
        rst!Scrap = Status
        If Status Then
            rst!StatusChangeTime = Now
        End If
        rst!Qty = WireMass
        rst!RecordedTime = Now
        rst.Update
    End If

    Me!WrapWireAsBuiltBOMID = rst!AsBuiltBOMID

    rst.Close

```

```
UpdateAxialWire:

   Set rst = dbs.OpenRecordset(stSql2, dbOpenDynaset)

   If rst.RecordCount = 0 Then
      MsgBox "As built BOM has not been properly populated"
      rst.Close
      dbs.Close
      Exit Sub
   End If

   WireMass = Round(Me!AxialWireMass * Me!nAxialRods * Me!WrapLength / 1000, 1)

   rst.MoveFirst

   If IsNull(rst!AllocatedID) Then
      rst.Edit
      rst!AllocatedID = Me!cboAxialWireID
      rst!Scrap = Status
      rst!Qty = WireMass
      rst!RecordedTime = Now
      rst.Update
   Else
      AssemblyID = Me!AssemblyID
      OperationBOMID = rst!OperationBOMID
      rst.AddNew
      rst!AssemblyID = AssemblyID
      rst!OperationBOMID = OperationBOMID
      rst!AllocatedID = Me!cboAxialWireID
      rst!Scrap = Status
      rst!Qty = WireMass
      rst!RecordedTime = Now
      rst.Update
   End If

   Me!AxialWireAsBuiltBOMID = rst!AsBuiltBOMID

   rst.Close
   dbs.Close

UpdateAsBuiltOperation:

   If Me![cboWrapTypeID] > 1 Then
      Exit Sub
   End If

   stSql = "(tblWrap.WrapTypeID)=1 AND (tblWrap.WrapSign)>0 AND "
   stSql = stSql & "(tblWrap.AssemblyID)=" & Me![AssemblyID]

   If Me![NSections] = DCount("[WrapSign]", "tblWrap", stSql) Then
      Me![AssyWrapSign] = Me![cboWrapSign]
      Forms![frmAssembly]![frmAsBuiltOperationWrapping].Form![cboOperationSignWrap] =
Me![cboWrapSign]
      Forms![frmAssembly]![frmAsBuiltOperationWrapping].Form![OperationDate] = Now
```

```
        Me![AssyWrapDate] = Now
    Else
        Forms![frmAssembly]![frmAsBuiltOperationWrapping].Form![cboOperationSignWrap] = Null
        Forms![frmAssembly]![frmAsBuiltOperationWrapping].Form![OperationDate] = Null
    End If

Exit_cboWrapSign_AfterUpdate:
    Exit Sub

Err_cboWrapSign_AfterUpdate:
    MsgBox Err.Description
    Resume Exit_cboWrapSign_AfterUpdate

End Sub

Private Sub cboWrapSign_GotFocus()
'
'   Check that all data has been entered
'

    If IsNull(Me![cboMachineID]) Then
        Me![cboMachineID].SetFocus
        MsgBox "Select wrapping machin"
        Exit Sub
    End If

    If IsNull(Me![cboWrapWireID]) Then
        Me![cboWrapWireID].SetFocus
        MsgBox "Select wrap wire traceability"
        Exit Sub
    End If

    If IsNull(Me![cboAxialWireID]) Then
        Me![cboAxialWireID].SetFocus
        MsgBox "Select axial wire traceability"
        Exit Sub
    End If

    If IsNull(Me![WrapSection]) Then
        Me![WrapSection].SetFocus
        MsgBox "Enter section that has been wrapped"
        Exit Sub
    End If

    If IsNull(Me![WrapStartPos]) Then
        Me![WrapStartPos].SetFocus
        MsgBox "Enter start position"
        Exit Sub
    End If

    If IsNull(Me![WrapLength]) Then
        Me![WrapLength].SetFocus
        MsgBox "Enter wrapping length"
        Exit Sub
    End If

    If IsNull(Me!AxialWireMass) Then
        MsgBox "Axial wire mass not entered in Order form"
        Me!WrapLength.SetFocus
        Exit Sub
    End If

    If IsNull(Me!nAxialRods) Then
        MsgBox "Number of axial rods not entered in Order form"
        Me!WrapLength.SetFocus
        Exit Sub
    End If

    If IsNull(Me!WrapWireMass) Then
        MsgBox "Wrap wire mass not entered in Order form"
        Me!WrapLength.SetFocus
        Exit Sub
    End If

    If IsNull(Me!NomScreenOD) Then
        MsgBox "Nominel Screen OD not entered in Order form"
        Me!WrapLength.SetFocus
        Exit Sub
    End If

    If IsNull(Me!WrapPitch) Then
        MsgBox "Wrap pitch missing"
        Me!WrapPitch.SetFocus
        Exit Sub
    End If

    If IsNull(Me![WrapDate]) Then
        Me![WrapDate] = Now
    End If

End Sub

Private Sub cboWrapTypeID_AfterUpdate()
'
'   If status is changed, change status in tblAsBuiltBOM
'   IF A NORMAL SECTION IS SET TO SCRAP, SIGN IN TBLWRAP SHOULD BE REMOVED
'   THIS IS NOT IMPLEMENTED
'
    Dim stSql As String
    Dim Status As Boolean
'    Dim WireMass As Single
'    Dim AssemblyID As Long, OperationBOMID As Long
    Dim dbs As Database
    Dim rst As DAO.Recordset
    Dim I As Integer
    If IsNull(Me!WrapWireAsBuiltBOMID) Then
        MsgBox "Record wrap wire traceability"
        Me!cboWrapWireID.SetFocus
        Exit Sub
    End If

    If IsNull(Me!AxialWireAsBuiltBOMID) Then
        MsgBox "Record axial wire traceability"
        Me!cboAxialWireID.SetFocus
        Exit Sub
    End If

    stSql = "SELECT tblAsBuiltBOM.AsBuiltBOMID, tblAsBuiltBOM.AssemblyID, "
    stSql = stSql & "tblAsBuiltBOM.Scrap, tblAsBuiltBOM.StatusChangeTime "
    stSql = stSql & "FROM tblAsBuiltBOM "
    stSql = stSql & "WHERE (tblAsBuiltBOM.AssemblyID)=" & Me!AssemblyID

    If Me!cboWrapTypeID = 1 Then
        Status = False
    Else
        Status = True
    End If

    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset(stSql, dbOpenDynaset)

    rst.MoveFirst
    Do While Not rst.EOF
        If rst!AsBuiltBOMID = Me!WrapWireAsBuiltBOMID Then
            rst.Edit
            rst!Scrap = Status
            rst!StatusChangeTime = Now
            rst.Update
            Exit Do
        End If
        rst.MoveNext
    Loop

    rst.MoveFirst
    Do While Not rst.EOF
        If rst!AsBuiltBOMID = Me!AxialWireAsBuiltBOMID Then
            rst.Edit
            rst!Scrap = Status
            rst!StatusChangeTime = Now
            rst.Update
            Exit Do
        End If
        rst.MoveNext
    Loop

    rst.Close
    dbs.Close

End Sub
```

```

Private Sub cboWrapWireID_AfterUpdate()
'
'   Add traceability information to tblAsBuiltBOM
'   If the traceability is selected again, the record will not be updated
'
'   14Dec2011 TM
'   Code deactivated as this is updated after signing the record
'   11Feb2012 TM
'
Exit Sub
    Dim stSql As String
    Dim Status As Boolean
    Dim WireMass As Single
    Dim AssemblyID As Long, OperationBOMID As Long
    Dim dbs As Database
    Dim rst As DAO.Recordset

    If Me!WrapWireAsBuiltBOMID > 0 Then
        Exit Sub
    End If

    stSql = "SELECT tblAsBuiltBOM.AsBuiltBOMID, tblAsBuiltBOM.AssemblyID, "
    stSql = stSql & "tblAsBuiltBOM.OperationBOMID, tblAsBuiltBOM.AllocatedID, "
    stSql = stSql & "tblAsBuiltBOM.Qty, tblAsBuiltBOM.Scrap, tblOrderLineBOM.Usage, "
    stSql = stSql & "tblOperationBOM.HideOperationBOM, tblOrderLineBOM.HideOLBOM, "
    stSql = stSql & "tblAsBuiltBOM.RecordedTime "
    stSql = stSql & "FROM tblOrderLineBOM INNER JOIN (tblAsBuiltBOM INNER JOIN "
    stSql = stSql & "tblOperationBOM ON tblAsBuiltBOM.OperationBOMID = "
    stSql = stSql & "tblOperationBOM.OperationBOMID) ON tblOrderLineBOM.OrderLineBOMID = "
    stSql = stSql & "tblOperationBOM.OrderLineBOMID "
    stSql = stSql & "WHERE (tblAsBuiltBOM.AssemblyID)=" & Me![AssemblyID]
    stSql = stSql & " AND (tblOrderLineBOM.Usage)=4 "
    stSql = stSql & "AND (tblOperationBOM.HideOperationBOM)=False "
    stSql = stSql & "AND (tblOrderLineBOM.HideOLBOM)=False"

    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset(stSql, dbOpenDynaset)


    If rst.RecordCount = 0 Then
        MsgBox "As built BOM has not been properly populated"
        Exit Sub
    End If

    WireMass = Me!WrapWireMass * Me!NomScreenOD * Pi() * Me!WrapLength / Me!WrapPitch / 1000

    If Me!cboWrapTypeID = 1 Then
        Status = False
    Else
        Status = True
    End If

```

```
   rst.MoveFirst

   If IsNull(rst!AllocatedID) Then
      rst.Edit
      rst!AllocatedID = Me!cboWrapWireID
      rst!Scrap = Status
      rst!Qty = WireMass
      rst!RecordedTime = Now
      rst.Update
   Else
      AssemblyID = Me!AssemblyID
      OperationBOMID = rst!OperationBOMID
      rst.AddNew
      rst!AssemblyID = AssemblyID
      rst!OperationBOMID = OperationBOMID
      rst!AllocatedID = Me!cboWrapWireID
      rst!Scrap = Status
      rst!Qty = WireMass
      rst!RecordedTime = Now
      rst.Update
   End If

   Me!WrapWireAsBuiltBOMID = rst!AsBuiltBOMID

   rst.Close
   dbs.Close


End Sub

Private Sub cboWrapWireID_GotFocus()
If IsNull(Me![cboAxialWireID]) Then
   MsgBox "Select axial wire first"
   Me![cboAxialWireID].SetFocus
End If
End Sub
```

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_Start"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdBug_Click()
On Error GoTo Err_cmdBug_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmBug"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.GoToRecord , , acNewRec

Exit_cmdBug_Click:
    Exit Sub

Err_cmdBug_Click:
    MsgBox Err.Description
    Resume Exit_cmdBug_Click


End Sub

Private Sub cmdOperators_Click()
    Dim Priv As Long

    Priv = Nz(DLookup("[Privileges]", "tblOperator", "[LDapUser]='" & Me![LogOnName] & "'"), 0)
    If Bitc(Priv, 0) Or Bitc(Priv, 9) Then
        DoCmd.OpenForm "frmOperator", acNormal
    End If

End Sub


Private Sub cmdQC_Click()
On Error GoTo Err_cmdQC_Click

    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim Priv As Long
```

```

   Priv = Nz(DLookup("[Privileges]", "tblOperator", "[LDapUser]='" & Me![LogOnName] & "'"), 0)
   If Bitc(Priv, 0) Or Bitc(Priv, 9) Or Bitc(Priv, 11) Then
      stDocName = "frmQC"
      DoCmd.OpenForm stDocName, , , stLinkCriteria
   End If

Exit_cmdQC_Click:
   Exit Sub

Err_cmdQC_Click:
'  Write error message to tblBug
   Forms![Start]![txtBug] = "cmdQC_Click: " & Err.Description
   DoCmd.SetWarnings False
   DoCmd.OpenQuery "qryappBug", acNormal, acEdit
   DoCmd.SetWarnings True
'
   MsgBox Err.Description
   Resume Exit_cmdQC_Click

End Sub

Private Sub cmdSpooling_Click()
On Error GoTo Err_cmdSpooling_Click

   Dim stDocName As String
   Dim stLinkCriteria As String

   stDocName = "frmSpooling"
   DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdSpooling_Click:
   Exit Sub

Err_cmdSpooling_Click:
   MsgBox Err.Description
   Resume Exit_cmdSpooling_Click
   End Sub

Sub cmdAssembly_Click()
On Error GoTo Err_cmdAssembly_Click

   Dim stDocName As String
   Dim stLinkCriteria As String

   stDocName = "frmAssembly"
   DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdAssembly_Click:
   Exit Sub

Err_cmdAssembly_Click:
   MsgBox Err.Description
```

```
    Resume Exit_cmdAssembly_Click

End Sub

Private Sub cmdOrder_Click()
On Error GoTo Err_cmdOrder_Click

    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim Priv As Long

    Priv = Nz(DLookup("[Privileges]", "tblOperator", "[LDapUser]='" & Me![LogOnName] & "'"), 0)
    If Bitc(Priv, 0) Or Bitc(Priv, 9) Or Bitc(Priv, 11) Then
        stDocName = "frmOrder"
        DoCmd.OpenForm stDocName, , , stLinkCriteria
    End If

Exit_cmdOrder_Click:
    Exit Sub

Err_cmdOrder_Click:
    MsgBox Err.Description
    Resume Exit_cmdOrder_Click

End Sub
Private Sub cmdSpool_Click()
End Sub

Private Sub cmdSlotData_Click()
On Error GoTo Err_cmdSlotData_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmSlotResult"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdSlotData_Click:
    Exit Sub

Err_cmdSlotData_Click:
    MsgBox Err.Description
    Resume Exit_cmdSlotData_Click

End Sub

Private Sub cmdWareHouse_Click()
On Error GoTo Err_cmdWareHouse_Click

    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim Priv As Long

    Priv = Nz(DLookup("[Privileges]", "tblOperator", "[LDapUser]='" & Me![LogOnName] & "'"), 0)
    If Bitc(Priv, 0) Or Bitc(Priv, 9) Or Bitc(Priv, 11) Then
        stDocName = "frmWareHouse"
        DoCmd.OpenForm stDocName, , , stLinkCriteria
    End If

Exit_cmdWareHouse_Click:
    Exit Sub

Err_cmdWareHouse_Click:
    MsgBox Err.Description
    Resume Exit_cmdWareHouse_Click

End Sub
```