

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmAllocated"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Private Sub cmdSerialNo_Click()
11 '
12 '  Verify that first record has been generated and open form to generate multiple records
13 '
14 '  09 Mar 2012 TM
15 '
16 If IsNull(Me![AllocatedID]) Then
17     MsgBox "Enter Traceability information for first record"
18     Me![PO].SetFocus
19 End If
20
21
22 Forms![frmWareHouse]![lstOrderLineBOM].SetFocus ' To save current record before
opening form
23 DoCmd.OpenForm "frmPopulateAllocated", acNormal, , "[AllocatedID]=" & Me![AllocatedID]
24 End Sub
25
26 Private Sub PO_AfterUpdate()
27 Me![AllocatedDate] = Now
28 Me![AllocatedSign] = Forms![Start]![txtOperatorID]
29 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmAllocatedUB"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboPart_GotFocus()
13     cboPart.Requery
14 End Sub
15
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmAsBuiltBOM"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboAllocatedID_AfterUpdate()
13 '
14 '   Set RecordedTime
15 '
16     Me!RecordedTime = Now
17
18 End Sub
19
20 Private Sub cboAllocatedID_GotFocus()
21     If Me![Scrap] Then
22         Me!Description.SetFocus
23     Exit Sub
24 End If
25
26 cboAllocatedID.Requery
27 End Sub
28
29 Private Sub cboOperationBOMID_GotFocus()
30     cboOperationBOMID.Requery
31 End Sub
32
33 '-----
34 ' cmdDuplicate_Click
35 '
36 '-----
37 Private Sub cmdDuplicate_Click()
38 On Error GoTo cmdDuplicate_Click_Err
39
40     Dim dbs As Database
41     Dim rst As DAO.Recordset
42
43     Set dbs = CurrentDb
44     Set rst = dbs.OpenRecordset("tblAsBuiltBOM", dbOpenDynaset)
45
46     rst.AddNew
47     rst!AssemblyID = Me![AssemblyID]
48     rst!OperationBOMID = Me![OperationBOMID]
49     rst!Qty = 1
50     rst.Update
```

```
51
52  rst.Close
53  dbs.Close
54
55  Forms![frmAssembly]![frmAsBuiltOperation].Form![frmAsBuiltBOM].Requery
56
57  MsgBox "Check and eventually update Qty in both records"
58
59  cmdDuplicate_Click_Exit:
60  Exit Sub
61
62  cmdDuplicate_Click_Err:
63  MsgBox Error$
64  Resume cmdDuplicate_Click_Exit
65
66  End Sub
67
68
69  Private Sub Scrap_AfterUpdate()
70  '
71  ' Set StatusChangeTime
72  '
73  Me![StatusChangeTime] = Now
74
75  End Sub
76
77  Private Sub Scrap_GotFocus()
78  If Me![Scrap] Then
79      If Now - Nz(Me![StatusChangeTime], 0) > 0.005 Then
80          Me!Description.SetFocus
81      End If
82  End If
83
84  End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmAsBuiltBOMAll"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboAllocatedID_AfterUpdate()
13 '
14 ' Set RecordedTime
15 '
16     Me!RecordedTime = Now
17
18 End Sub
19
20 Private Sub cboAllocatedID_GotFocus()
21     If DLookup("[LockOperation]", "tblAsBuiltOperation", "(AssemblyID) = " & Me!AssemblyID & "AND (OperationID)=" & Me!OperationID) Then
22         Me!PartNo.SetFocus
23     Exit Sub
24 End If
25
26 If Me![Scrap] Then
27     Me!PartNo.SetFocus
28     Exit Sub
29 End If
30
31 cboAllocatedID.Requery
32 End Sub
33
34 Private Sub Detail_Db1Click(Cancel As Integer)
35 MsgBox Me![Qty]
36 End Sub
37
38 Private Sub Qty_Click()
39 If DLookup("[LockOperation]", "tblAsBuiltOperation", "(AssemblyID) = " & Me!AssemblyID & "AND (OperationID)=" & Me!OperationID) Then
40     Me!PartNo.SetFocus
41 End If
42 End Sub
43
44 Private Sub Scrap_AfterUpdate()
45 '
46 ' Set StatusChangeTime
47 '
48     Me![StatusChangeTime] = Now
```

```
49
50 End Sub
51
52 Private Sub Scrap_GotFocus()
53     If Me![Scrap] Then
54         If Now - Nz(Me![StatusChangeTime], 0) > 0.005 Then
55             Me!PartNo.SetFocus
56         End If
57     End If
58 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmAsBuiltOperationSlot"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboSlotSign_AfterUpdate()
13 On Error GoTo Err_cboSlotSign_AfterUpdate
14 '
15 ' Add basic slot data to tblStatistics and
16 ' add content to tblSlotResult and if applicable to tblScanLargeSlot
17 ' If data already has been recorded, hide old data
18 ' Lock forms
19 ' To unlock forms, signature must be removed
20 '
21
22 Dim AssemblyID As Long
23 Dim Stat As Integer
24 Dim strSql As String, strDocName As String
25
26 Stat = 0
27 AssemblyID = Me![AssemblyID]
28 '
29 ' Slot requirements based on qrySlotData31 entered into tblSlotStatistics
30 '
31 strSql = "[AssemblyID]=" & AssemblyID & " and [HideSlotStatistics]=False"
32 If DCount("[AssemblyID]", "tblSlotStatistics", strSql) Then
33     Stat = 10
34     DoCmd.SetWarnings False
35     DoCmd.OpenQuery "qupdSlotStatistics", acNormal, acEdit
36     DoCmd.SetWarnings True
37 End If
38 Stat = Stat + 1
39 DoCmd.SetWarnings False
40 DoCmd.OpenQuery "qryappSlotData31", acNormal, acEdit
41 DoCmd.SetWarnings True
42 '
43 ' Slot requirements based on qrySlotData31 entered into tblSlotResult
44 '
45 Stat = Stat + 1
46 strSql = "[AssemblyID]=" & AssemblyID & " and [HideSlotResult]=False"
47 If DCount("[AssemblyID]", "tblSlotResult", strSql) Then
48     Stat = Stat + 20
49     DoCmd.SetWarnings False
50     DoCmd.OpenQuery "qupdSlotResult", acNormal, acEdit
```

```
51      DoCmd.SetWarnings True
52  End If
53  Stat = Stat + 1
54  DoCmd.SetWarnings False
55  DoCmd.OpenQuery "qryappSlotData3", acNormal, acEdit
56  DoCmd.SetWarnings True
57  '
58  ' Slot requirements based on qrySlotExp1111 entered into tblScanLargeSlot
59  '
60  Stat = Stat + 1
61  strSql = "[AssemblyID]=" & AssemblyID & " and [HideScanLargeSlot]=False"
62  If DCount("[AssemblyID]", "tblScanLargeSlot", strSql) Then
63      Stat = Stat + 30
64      DoCmd.SetWarnings False
65      DoCmd.OpenQuery "qupdScanLargeSlot", acNormal, acEdit
66      DoCmd.SetWarnings True
67  End If
68  Stat = Stat + 1
69  DoCmd.SetWarnings False
70  DoCmd.OpenQuery "qryappSlotExp1111", acNormal, acEdit
71  DoCmd.SetWarnings True
72  '
73  ' Lock record, forms and controls
74  '
75  Stat = Stat + 1
76  Me![LockOperation] = True
77  Forms![frmSlotResult]![frmSlotLarge].Locked = True
78  Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Locked = True
79  Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Enabled = False
80  Forms![frmSlotResult]![frmSlotFile].Locked = True
81  Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Locked = True
82  Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Enabled = False
83  '
84  ' Set date
85  '
86  Me![OperationDate] = Now
87  '
88  ' If this is last operation, set sign and date in tblAssembly
89  '
90  Stat = Stat + 1
91  strSql = "[HideOperation] = 0 And [OrderLineID] = " &
Forms![frmSlotResult]![cboOrderLine]
92  If DMax("[OperationNo]", "tblOperation", strSql) = Forms![frmSlotResult]![txtOperationNo]
* 1 Then
93      Me![AssemblyDate] = Me![OperationDate]
94      Me![AssemblySign] = Me![cboSlotSign]
95  End If
96
97  Exit_cboSlotSign_AfterUpdate:
98  Exit Sub
99
100 Err_cboSlotSign_AfterUpdate:
101  ' Write error message to tblBug
102  Forms![Start]![txtBug] = "cboSlotSign_AfterUpdate: " & Stat & Err.Description
103  DoCmd.SetWarnings False
```



```
104 DoCmd.OpenQuery "qryappBug", acNormal, acEdit
105 DoCmd.SetWarnings True
106 '
107 MsgBox Err.Description
108 Resume Exit_cboSlotSign_AfterUpdate
109
110 End Sub
111
112 Private Sub cboSlotSign_GotFocus()
113
114 If Nz(Me![cboSlotSign], 0) Then
115     MsgBox "Record has been signed off"
116     Me![NCText].SetFocus
117     Exit Sub
118 End If
119
120 If Forms![frmSlotResult]![txtStatusVal] = 99 Then
121     MsgBox "Slot data not imported"
122     Me![NCText].SetFocus
123     Exit Sub
124 End If
125
126 If DLookup("[MaxOfExprStatus]", "qryPreviousOpSlotStatusGB") = 0 Then
127     MsgBox "Previous operation not completed"
128     Me![NCText].SetFocus
129     Exit Sub
130 End If
131
132 If DLookup("[MaxOfExprStatus]", "qryPreviousOpQCStatusGB") = 0 Then
133     MsgBox "Previous operation not signed off by QC"
134     Me![NCText].SetFocus
135     Exit Sub
136 End If
137
138 If Me![fraStatus] = 1 Then
139     If Forms![frmSlotResult]![txtStatusVal] > 0 Then
140         MsgBox "Check data! Screen do not meet requirements."
141         Me![NCText].SetFocus
142         Exit Sub
143     End If
144 Else
145     If IsNull(Me![NCText]) Then
146         MsgBox "Check status or write explanation in Non Conformance text box"
147         Me![NCText].SetFocus
148     End If
149 End If
150
151 End Sub
152
153 Private Sub cmdUnlock_Click()
154 '
155 ' Can be unlocked if next operation is not signed off
156 ' Can remove signature if current record is unlocked
157 '
```

```
158   If IsNull(Me![cboSlotSign]) Then
159       Exit Sub
160   End If
161
162   If Me![cboSlotSign] Then
163       If DFirst("[OperationSign]", "qryListOperationStatus") Then
164           MsgBox "Record must be unlocked by QC!"
165           Exit Sub
166       End If
167   End If
168 '
169 ' If OK on delete signature
170 ' Lock/unlock forms and controls in frmSlotResult
171 '
172   If MsgBox("Do you want to delete signature to open for new import?", vbOKCancel,
"ResLog") = vbOK Then
173       Me![cboSlotSign] = Null
174       Me![LockOperation] = False
175       Forms![frmSlotResult]![frmSlotLarge].Locked = False
176       Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Locked = False
177       Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Enabled = True
178       Forms![frmSlotResult]![frmSlotFile].Locked = False
179       Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Locked = False
180       Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Enabled = True
181   End If
182
183 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmAssembly"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboAssemblyID_AfterUpdate()
13
14 '
15 ' Set default values after assembly has been selected
16 '
17
18 On Error GoTo Err_cboAssemblyID_AfterUpdate
19
20 Dim stSql As String
21 '
22 ' Set AssemblyID in form Start as reference for Slot opening data queries
23 '
24
25 ' Forms![frmAssembly]![frmOrderLine].Form![chkAllinWO] = 0
26 Forms![Start]![txtMinAssemblyID] = Me![cboAssemblyID]
27 Forms![Start]![txtMaxAssemblyID] = Me![cboAssemblyID]
28 Forms![Start]![txtOperationNo] = Me![txtOperationNo]
29
30 '
31 ' Requery list box to reflect selected assembly
32 ' Select record in frmWrapDetail form based on selected assembly
33 '
34
35 lstWrapID.Requery
36 Me![lstWrapID] = DMin("[WrapID]", "tblWrap", "[AssemblyID]=" & Me![cboAssemblyID])
37
38 '
39 ' Lock Operation form if also operation is selected and next operation is signed off
40 '
41 stSql = "[AssemblyID]=" & [Forms]![frmAssembly]![cboAssemblyID]
42 stSql = stSql & " AND [OperationID]=" & [Forms]![frmAssembly]![cboOperationID]
43
44 If Not IsNull([Forms]![frmAssembly]![cboOperationID]) Then
45     Forms![frmAssembly]![frmAsBuiltOperation].Locked = DLookup("[LockOperation]",
46 "tblAsBuiltOperation", stSql)
47     Forms![frmAssembly]![frmAsBuiltOperation].Enabled = Not DLookup("[LockOperation]",
48 "tblAsBuiltOperation", stSql)
49 End If
50
```

```
49 '
50 ' Lock wrapping form if next operation is signed off
51 '
52 If DLookup("[LockOperation]", "qryWrapLock", "[AssemblyID]=" &
[Forms]![frmAssembly]![cboAssemblyID]) Then
53     Forms![frmAssembly]![frmAsBuiltOperationWrapping].Locked = True
54     Forms![frmAssembly]![frmAsBuiltOperationWrapping].Enabled = False
55     Forms![frmAssembly]![frmWrapDetail].Locked = True
56     Forms![frmAssembly]![frmWrapDetail].Enabled = False
57     Forms![frmAssembly]![cmdNewWrapSection].Enabled = False
58 Else
59     Forms![frmAssembly]![frmAsBuiltOperationWrapping].Locked = False
60     Forms![frmAssembly]![frmAsBuiltOperationWrapping].Enabled = True
61     Forms![frmAssembly]![frmWrapDetail].Locked = False
62     Forms![frmAssembly]![frmWrapDetail].Enabled = True
63     Forms![frmAssembly]![cmdNewWrapSection].Enabled = True
64 End If
65
66 Exit_cboAssemblyID_AfterUpdate:
67 Exit Sub
68
69 Err_cboAssemblyID_AfterUpdate:
70 MsgBox Err.Description
71 Resume Exit_cboAssemblyID_AfterUpdate
72
73 End Sub
74
75 Private Sub cboAssemblyID_GotFocus()
76 'cboAssemblyID.Requery
77 End Sub
78
79 Private Sub cboOperationID_AfterUpdate()
80 '
81 ' Requery forms and controls after update of this control
82 '
83 ' 10 Mar 2012 TM
84
85 Dim stSql As String
86
87 cboOperationID.Requery
88 Forms![frmAssembly]![frmAsBuiltOperation].Form![cboOperationSign].Requery
89 Forms![frmAssembly]![frmAsBuiltOperation].Form![cboOperatorSign2].Requery
90 Forms![Start]![txtOperationNo] = Me![txtOperationNo]
91
92 '
93 ' Set second sign visible if required
94 '
95 ' 04 Dec 2011 TM
96
97 If IsNull(Me![chkSecondSign]) Then
98     Forms![frmAssembly]![frmAsBuiltOperation].Form![cboOperatorSign2].Visible = False
99     Forms![frmAssembly]![frmAsBuiltOperation].Form![OperationDate2].Visible = False
100 ElseIf Me![chkSecondSign] Then
101     Forms![frmAssembly]![frmAsBuiltOperation].Form![cboOperatorSign2].Visible = True
```

```
102 Forms![frmAssembly]![frmAsBuiltOperation].Form![OperationDate2].Visible = True
103 Else
104 Forms![frmAssembly]![frmAsBuiltOperation].Form![cboOperatorSign2].Visible = False
105 Forms![frmAssembly]![frmAsBuiltOperation].Form![OperationDate2].Visible = False
106 End If
107
108 '
109 ' Set QC sign visible if required and unlock if operator has privileges
110 '
111
112 If IsNull(Me![chkQCSign]) Then
113 Forms![frmAssembly]![frmAsBuiltOperation].Form![cboQCOPSign].Visible = False
114 Forms![frmAssembly]![frmAsBuiltOperation].Form![QCOPDate].Visible = False
115 ElseIf Me![chkQCSign] Then
116 Forms![frmAssembly]![frmAsBuiltOperation].Form![cboQCOPSign].Visible = True
117 Forms![frmAssembly]![frmAsBuiltOperation].Form![QCOPDate].Visible = True
118 If Forms![Start]![txtOperatorID] Then
119 Forms![frmAssembly]![frmAsBuiltOperation].Form![cboQCOPSign].Locked = False
120 Forms![frmAssembly]![frmAsBuiltOperation].Form![QCOPDate].Locked = False
121 Else
122 Forms![frmAssembly]![frmAsBuiltOperation].Form![cboQCOPSign].Locked = True
123 Forms![frmAssembly]![frmAsBuiltOperation].Form![QCOPDate].Locked = True
124 End If
125 Else
126 Forms![frmAssembly]![frmAsBuiltOperation].Form![cboQCOPSign].Visible = False
127 Forms![frmAssembly]![frmAsBuiltOperation].Form![QCOPDate].Visible = False
128 End If
129
130 '
131 ' Lock form if assembly is selected
132 '
133 If Not IsNull([Forms]![frmAssembly]![cboAssemblyID]) Then
134 stSql = "[AssemblyID]=" & [Forms]![frmAssembly]![cboAssemblyID]
135 stSql = stSql & " AND [OperationID]=" & [Forms]![frmAssembly]![cboOperationID]
136 If Not IsNull(DLookup("[LockOperation]", "tblAsBuiltOperation", stSql)) Then
137 Forms![frmAssembly]![frmAsBuiltOperation].Locked = DLookup("[LockOperation]",
"tblAsBuiltOperation", stSql)
138 Forms![frmAssembly]![frmAsBuiltOperation].Enabled = Not
DLookup("[LockOperation]", "tblAsBuiltOperation", stSql)
139 End If
140 End If
141
142 End Sub
143
144 Private Sub cboOrder_AfterUpdate()
145 Forms![Start]![txtLocationID] = Me![txtLocationID]
146 End Sub
147
148 Private Sub cboOrder_GotFocus()
149 ' Terje 29Jan11
150 ' Filtration of orders
151 '
152 ' SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location,
tblOrder.CompletedSign
153 ' FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
```



```
154 ' ORDER BY tblOrder.ProdOrder;
155
156 Dim Sql As String
157
158 Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign],
[tblLocation.LocationID] "
159 Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID =
tblLocation.LocationID "
160
161
162 If Me![optionOrderSelect] = 1 Then
163     Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
164 End If
165
166 Forms![frmAssembly]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"
167
168
169 End Sub
170
171 Private Sub cboOrderLine_AfterUpdate()
172 '
173 ' If emty, set Order
174 '
175 On Error GoTo Err_cboOrderLine_AfterUpdate
176
177 If IsNull(Me![cboOrder]) Then
178     Me![cboOrder] = DLookup("[OrderID]", "tblOrderLine", "[OrderLineID] = " &
Me![cboOrderLine])
179 End If
180
181 Forms![frmAssembly]![frmWrapDetail].Form![cboAxialWireID].Requery
182 cboOperationID.Requery
183
184 '
185 ' Set parameters for slot data queries in Start form
186 '
187 Forms![Start]![txtOrderLineID] = Me![cboOrderLine]
188 Forms![Start]![txtLocationID] = Me![txtLocationID]
189
190 If DLookup("[InclAllLines]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine]) Then
191     Forms![Start]![txtInclAllLines] = 360
192 Else
193     Forms![Start]![txtInclAllLines] = 0
194 End If
195
196 Forms![Start]![txtMinAssemblyID] = 0
197 Forms![Start]![txtMaxAssemblyID] = 0
198 '
199 ' Requery Assembly and clear combo boxes
200 '
201 cboAssemblyID.Requery
202 Me![cboAssemblyID] = Null
203 Me![cboOperationID] = Null
204
205 Exit_cboOrderLine_AfterUpdate:
```

```
206 Exit Sub
207
208 Err_cboOrderLine_AfterUpdate:
209 MsgBox Err.Description
210 Resume Exit_cboOrderLine_AfterUpdate
211
212 End Sub
213
214 Private Sub cboOrderLine_GotFocus()
215 ' SELECT tblOrderLine.OrderLineID, tblOrderLine.WorkOrder, tblOrderLine.PartNo,
tblOrderLine.Description, tblOrder.CompletedSign
216 ' FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID
217 ' WHERE (((tblOrder.CompletedSign)=0));
218 On Error GoTo Err_cboOrderLine_GotFocus
219
220 Dim Sql As String
221
222 Sql = "SELECT [OrderLineID], [WorkOrder], [PartNo], [tblOrderLine.Description],
[CompletedSign] "
223 Sql = Sql & "FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID =
tblOrderLine.OrderID "
224 If Me![optionOrderSelect] = 1 Then
225 ' noe muffens når denne linjen ikke inkluderes - WHERE mangler
226 Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
227 Else
228 Sql = Sql & "WHERE [tblOrder.CompletedSign] >= 0 "
229 End If
230
231 If Not IsNull(Me![cboOrder]) Then
232 Sql = Sql & "AND [tblOrder.OrderID] = " & Me![cboOrder]
233 End If
234
235 Forms![frmAssembly]![cboOrderLine].RowSource = Sql & " ORDER BY
tblOrderLine.WorkOrder;"
236
237 Exit_cboOrderLine_GotFocus:
238 Exit Sub
239
240 Err_cboOrderLine_GotFocus:
241 MsgBox Err.Description
242 Resume Exit_cboOrderLine_GotFocus
243
244 End Sub
245
246 Private Sub cboWrapMachin_GotFocus()
247 cboWrapMachin.Requery
248 End Sub
249
250 Private Sub cboWrapSection_AfterUpdate()
251 '
252 ' Set default values in form frmWrapDetail
253 '
254 If IsNull(Me![cboWrapMachin]) Then
255 MsgBox "Select wrapping machine to set default wrapping parameters"
256 Exit Sub
```

```
257 End If
258
259
260 If IsNull(Forms![frmAssembly]![frmWrapDetail].Form![cboMachineID]) Then
261     Forms![frmAssembly]![frmWrapDetail].Form![cboMachineID] = Me![cboWrapMachin]
262 End If
263
264 End Sub
265
266 Private Sub cboWrapSection_GotFocus()
267     cboWrapSection.Requery
268 End Sub
269
270 Private Sub cmdNewWrapSection_Click()
271     '
272     ' Add new record to tblWrap and copy default values from first record
273     '
274     ' NOE SOM IKKE STEMME MED DEKLARERING
275     '
276     '
277     '
278     On Error GoTo Err_cmdNewWrapSection_Click
279
280     Dim stDocName As String
281     Dim stSql As String
282     Dim dbs As Database
283     Dim rst As DAO.Recordset, rst1 As DAO.Recordset, rst2 As DAO.Recordset
284
285     stSql = "SELECT tblWrap.* "
286     stSql = stSql & "FROM tblWrap "
287     stSql = stSql & "WHERE (tblWrap.AssemblyID)=" & Me![cboAssemblyID]
288
289
290     Set dbs = CurrentDb
291     Set recset = dbs.OpenRecordset(stSql, dbOpenDynaset)
292     Set recset1 = dbs.OpenRecordset("tblWrap", dbOpenDynaset)
293
294     recset.MoveFirst
295     recset1.AddNew
296     recset1!AssemblyID = Me![cboAssemblyID]
297     recset1!MachineID = recset!MachineID
298     recset1!WrapPitch = recset!WrapPitch
299     recset1!WeldTime = recset!WeldTime
300     recset1!SetCurrent = recset!SetCurrent
301     recset1!WrapSpeed = recset!WrapSpeed
302     recset1!WeldPointPos = recset!WeldPointPos
303     recset1!ForceWeldWeel = recset!ForceWeldWeel
304     recset1.Update
305
306     lstWrapID.Requery
307     Me![lstWrapID] = DMax("[WrapID]", "tblWrap", "[AssemblyID]=" & Me![cboAssemblyID])
308
309     recset.Close
```

```
310  recset1.Close
311  dbs.Close
312
313  Exit_cmdNewWrapSection_Click:
314  Exit Sub
315
316  Err_cmdNewWrapSection_Click:
317  MsgBox Err.Description
318  Resume Exit_cmdNewWrapSection_Click
319
320  End Sub
321  '-----
322  ' cmdFileImport_Click
323  '
324  '-----
325  Private Sub cmdFileImport_Click()
326  On Error GoTo cmdFileImport_Click_Err
327
328  DoCmd.OpenForm "frmSlotFileImport", acNormal, "", "", acNormal
329
330
331  cmdFileImport_Click_Exit:
332  Exit Sub
333
334  cmdFileImport_Click_Err:
335  MsgBox Error$
336  Resume cmdFileImport_Click_Exit
337
338  End Sub
339
340
341  '-----
342  ' cmdSlotDetails_Click
343  ' Not active TM 18 Nov 2012
344  '-----
345  Private Sub cmdSlotDetails_Click()
346  On Error GoTo cmdSlotDetails_Click_Err
347
348  DoCmd.OpenForm "frmSlotResult", acNormal, "", "", acNormal
349
350  If Not IsNull(Me![cboOrder]) Then
351  Forms![frmSlotResult]![cboOrder] = Me![cboOrder]
352  End If
353  If Not IsNull(Me![cboOrderLine]) Then
354  Forms![frmSlotResult]![cboOrderLine] = Me![cboOrderLine]
355  End If
356
357  cmdSlotDetails_Click_Exit:
358  Exit Sub
359
360  cmdSlotDetails_Click_Err:
361  MsgBox Error$
362  Resume cmdSlotDetails_Click_Exit
363
```

364 End Sub

365

366

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmAssyOpStatus"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboNCSign_AfterUpdate()
13     If IsNull(Me![NCDate]) Then Me![NCDate] = Now
14 End Sub
15
16 Private Sub Option22_GotFocus()
17 '
18 ' Check signature and NC report no before this option can be selected
19 '
20 If Me![fraStatus] = 4 Then
21     MsgBox "This option should not be choosen"
22     If IsNull(Me![cboNCSign]) Then
23         MsgBox "Sign off before selecting this option"
24         Me![cboNCSign].SetFocus
25     Exit Sub
26 End If
27
28 If IsNull(Me![NCNo]) Then
29     MsgBox "Enter Non Conformancy No before selecting this option"
30     Me![NCNo].SetFocus
31 Exit Sub
32 End If
33 End If
34
35 End Sub
36
37 Private Sub Option24_GotFocus()
38 '
39 ' Check signature and NC report no before this option can be selected
40 '
41 If Me![fraStatus] = 4 Then
42     If IsNull(Me![cboNCSign]) Then
43         MsgBox "Sign off before selecting this option"
44         Me![cboNCSign].SetFocus
45     Exit Sub
46 End If
47
48 If IsNull(Me![NCNo]) Then
49     MsgBox "Enter Non Conformancy No before selecting this option"
50     Me![NCNo].SetFocus
```



```
51      Exit Sub
52      End If
53      End If
54
55 End Sub
56
57 Private Sub Option30_GotFocus()
58 '
59 ' Check signature and NC report no before this option can be selected
60 '
61 If Me![fraStatus] = 4 Then
62     If IsNull(Me![cboNCSign]) Then
63         MsgBox "Sign off before selecting this option"
64         Me![cboNCSign].SetFocus
65         Exit Sub
66     End If
67
68     If IsNull(Me![NCNo]) Then
69         MsgBox "Enter Non Conformancy No before selecting this option"
70         Me![NCNo].SetFocus
71         Exit Sub
72     End If
73 End If
74
75 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmBug"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmAsBuiltOperation"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboOperationID_GotFocus()
13     cboOperationID.Requery
14 End Sub
15
16 Private Sub cboOperationSign_AfterUpdate()
17     '
18     ' Lock form for previous records by running update query
19     ' This query sets all previous operations to status Lock
20     ' Set date for current record
21     '
22     Dim strDocName As String
23
24     strDocName = "qupdAsBuiltOperationLock"
25     DoCmd.SetWarnings False
26     DoCmd.OpenQuery strDocName, acNormal, acEdit
27     DoCmd.SetWarnings True
28
29     Me![OperationDate] = Now
30     '
31     ' If this is last operation, set sign and date in tblAssembly
32     '
33     strSql = "[HideOperation] = 0 And [OrderLineID] = " & Forms![frmAssembly]![cboOrderLine]
34     If DMax("[OperationNo]", "tblOperation", strSql) = Forms![frmAssembly]![cboOperationID]
35         * 1 Then
36             Me![AssemblyDate] = Me![OperationDate]
37             Me![AssemblySign] = Me![cboOperationSign]
38         End If
39     End Sub
40
41 Private Sub cboOperationSign_GotFocus()
42     '
43     ' Verify that all information has been recorded
44     '
45
46     If Forms![frmAssembly]![txtWorkStationID] < 3 Then ' Wrapping and ResGauge are 1 and
47         2
48         MsgBox "Wrapping or ResGauge operations cannot be signed off in this form!"
49         Me![NCText].SetFocus
```

```
49      Exit Sub
50  End If
51
52  If IsNull(Me![Status]) Then
53      Me![Status] = 1
54  End If
55
56  If DLookup("[MaxOfExprStatus]", "qryPreviousOpSlotStatusGB") = 0 Then
57      MsgBox "Previous operation not completed"
58      Me![NCText].SetFocus
59      Exit Sub
60  End If
61
62  If DLookup("[MaxOfExprStatus]", "qryPreviousOpQCStatusGB") = 0 Then
63      MsgBox "Previous operation not signed off by QC"
64      Me![NCText].SetFocus
65      Exit Sub
66  End If
67
68  If Me![Status] = 1 Then
69      If DLookup("[TestStatus]", "qryOpStatTest") = 0 Then
70          MsgBox "Inspection results missing"
71          Me![fraStatus].SetFocus
72      End If
73      If DLookup("[MaxOfTraceability]", "qryOpStatBOM2") = 0 Then
74          MsgBox "Traceability information missing"
75          Me![fraStatus].SetFocus
76      End If
77      If DLookup("[MaxOfQtyRec]", "qryOpStatBOM2") = 0 Then
78          MsgBox "Wrong traceability quantity recorded"
79          Me![fraStatus].SetFocus
80      End If
81      If DLookup("[Sign2]", "qryOpStat2Sign") = 0 Then
82          MsgBox "Second signature missing"
83          Me![fraStatus].SetFocus
84      End If
85  Else
86      If IsNull(Me![NCText]) Then
87          MsgBox "Write explanation"
88          Me![NCText].SetFocus
89      End If
90  End If
91
92  End Sub
93
94  Private Sub cboOperatorSign2_AfterUpdate()
95      Me![OperationDate2] = Now
96      Forms![frmAssembly]![txtOrder.Column(2)].SetFocus
97  End Sub
98
99  Private Sub cboQCOpSign_AfterUpdate()
100     Me![QCOpDate] = Now
101  End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmNewWorkOrder"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub WorkOrder_AfterUpdate()
13     Me![OrderID] = [Forms]![frmOrder]![cboOrder]
14 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmOperation"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub HideOperation_AfterUpdate()
13     Forms![frmOrder]![cboOrderLine].Requery
14 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmOperationBOM"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboOperationID_AfterUpdate()
13 '
14 '   Copy quantity as default value if blank
15 '
16 On Error GoTo Err_cboOperationID_AfterUpdate
17
18     If IsNull(Me![tblOperationBOM.Qty]) Then
19         Me![tblOperationBOM.Qty] = Me![tblOrderLineBOM.Qty]
20     End If
21
22 Exit_cboOperationID_AfterUpdate:
23     Exit Sub
24
25 Err_cboOperationID_AfterUpdate:
26     MsgBox Err.Description
27     Resume Exit_cboOperationID_AfterUpdate
28
29 End Sub
30
31
32
33 Private Sub cmdNew_Click()
34 '
35 '   Copy current part for new operation
36 '
37 On Error GoTo Err_cmdNew_Click
38
39     Dim dbs As Database
40     Dim rst As Recordset
41     Dim OLbomid As Long
42
43     OLbomid = Me![tblOrderLineBOM.OrderLineBOMID]
44
45     Set dbs = CurrentDb
46     Set rst = dbs.OpenRecordset("tblOperationBOM", dbOpenDynaset)
47     rst.AddNew
48     rst!OrderLineBOMID = OLbomid
49     rst.Update
50     rst.Close
```

```
51     dbs.Close
52
53 Forms![frmOrder]![cboOrderLine].Requery
54 'frmOrder.cboOrderLine.Requery
55
56 Exit_cmdNew_Click:
57     Exit Sub
58
59 Err_cmdNew_Click:
60     MsgBox Err.Description
61     Resume Exit_cmdNew_Click
62
63 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmOperationTest"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboOperationID_AfterUpdate()
13     Me![OrderLineID] = Forms![frmOrder]![cboOrderLine]
14 End Sub
15
16 Private Sub cboOperationID_GotFocus()
17     cboOperationID.Requery
18 End Sub
19
20 Private Sub MaxVal_AfterUpdate()
21     '
22     ' Set code for operation test type
23     '
24     ' 02 Dec 2011 TM
25     '
26
27     If Not IsNull(Me![MaxVal]) Then
28         Me![OpTestType] = 2
29     Else
30         ' If IsNull(Me![MinVal]) Then
31         '     Me![OpTestType] = 1
32         ' End If
33     End If
34
35 End Sub
36
37 Private Sub MinVal_AfterUpdate()
38     '
39     ' Set code for operation test type
40     '
41     ' 02 Dec 2011 TM
42     '
43
44     If Not IsNull(Me![MinVal]) Then
45         Me![OpTestType] = 2
46     Else
47         ' If IsNull(Me![MaxVal]) Then
48         '     Me![OpTestType] = 1
49         ' End If
50     End If
```

51

52 End Sub

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmOperator"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cmdNewRecOperator_Click()
13 On Error GoTo Err_cmdNewRecOperator_Click
14
15
16     'DoCmd.OpenForm "frmOperatorSub"
17     'DoCmd.GoToRecord acDataForm, "frmOperatorSub", acNewRec
18     'Forms![frmOperator]![frmOperatorSub].Form
19     ' DoCmd.GoToRecord acDataForm, Forms![frmOperator]![frmOperatorSub], acNewRec
20
21 Exit_cmdNewRecOperator_Click:
22     Exit Sub
23
24 Err_cmdNewRecOperator_Click:
25     MsgBox Err.Description
26     Resume Exit_cmdNewRecOperator_Click
27
28
29 End Sub
30
31 Private Sub cmdNewOperator_Click()
32 On Error GoTo Err_cmdNewOperator_Click
33
34
35     DoCmd.OpenForm "frmNewOperator"
36     DoCmd.GoToRecord acDataForm, "frmNewOperator", acNewRec
37
38 Exit_cmdNewOperator_Click:
39     Exit Sub
40
41 Err_cmdNewOperator_Click:
42     MsgBox Err.Description
43     Resume Exit_cmdNewOperator_Click
44
45 End Sub
46
47 Private Sub OperatorID_GotFocus()
48 OperatorID.Requery
49 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmOperatorSub"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub chk0_AfterUpdate()
13     Me![Privileges] = Me![Text38]
14 End Sub
15 Private Sub chk1_AfterUpdate()
16     Me![Privileges] = Me![Text38]
17 End Sub
18 Private Sub chk2_AfterUpdate()
19     Me![Privileges] = Me![Text38]
20 End Sub
21 Private Sub chk3_AfterUpdate()
22     Me![Privileges] = Me![Text38]
23 End Sub
24 Private Sub chk4_AfterUpdate()
25     Me![Privileges] = Me![Text38]
26 End Sub
27 Private Sub chk5_AfterUpdate()
28     Me![Privileges] = Me![Text38]
29 End Sub
30 Private Sub chk6_AfterUpdate()
31     Me![Privileges] = Me![Text38]
32 End Sub
33 Private Sub chk7_AfterUpdate()
34     Me![Privileges] = Me![Text38]
35 End Sub
36 Private Sub chk8_AfterUpdate()
37     Me![Privileges] = Me![Text38]
38 End Sub
39 Private Sub chk9_AfterUpdate()
40     Me![Privileges] = Me![Text38]
41 End Sub
42 Private Sub chk10_AfterUpdate()
43     Me![Privileges] = Me![Text38]
44 End Sub
45 Private Sub chk11_AfterUpdate()
46     Me![Privileges] = Me![Text38]
47 End Sub
48 Private Sub Form_Current()
49     '
50     ' Set status of check boxes according to existing value of field Privileges
```

51 '
52 Me![chk0] = Bitc(Me![Privileges], 0) ' Generate orders
53 Me![chk1] = Bitc(Me![Privileges], 1) ' Wrapping
54 Me![chk2] = Bitc(Me![Privileges], 2) ' ResGauge
55 Me![chk3] = Bitc(Me![Privileges], 3) ' Base pipe perf.
56 Me![chk4] = Bitc(Me![Privileges], 4) ' Welding
57 Me![chk5] = Bitc(Me![Privileges], 5) ' Heat treatment
58 Me![chk6] = Bitc(Me![Privileges], 6) ' General assembly
59 Me![chk7] = Bitc(Me![Privileges], 7) ' Torque machine
60 Me![chk8] = Bitc(Me![Privileges], 8) ' Pressure testing
61 Me![chk9] = Bitc(Me![Privileges], 9) ' Quality control
62 Me![chk10] = Bitc(Me![Privileges], 10) ' Drift of base pipe
63 Me![chk11] = Bitc(Me![Privileges], 11) ' Allocate parts
64
65 End Sub
66

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmOrder"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboOrder_AfterUpdate()
13     'cboOrderLine.Requery
14 End Sub
15
16 Private Sub cboOrder_GotFocus()
17     ' Terje 29Jan11
18     ' Filtration of orders
19     '
20     ' SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location,
tblOrder.CompletedSign
21     ' FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
22     ' ORDER BY tblOrder.ProdOrder;
23
24     Dim Sql As String
25
26     Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign] "
27     Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID =
tblLocation.LocationID "
28
29
30     If Me![optionOrderSelect] = 1 Then
31         Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
32     End If
33
34     Forms![frmOrder]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"
35
36
37 End Sub
38
39 Private Sub cboOrderLine_AfterUpdate()
40     '
41     ' Run append query to add records in tblAsBuiltBOM
42     '
43     On Error GoTo Err_cboOrderLine_AfterUpdate
44
45     cboOrderLine.Requery
46     Forms![frmOrder]![frmOperationTest].Form![cboOperationID].Requery
47     Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID1].Requery
48     Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID2].Requery
```

```
49 Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID3].Requery
50 Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID4].Requery
51 Forms![frmOrder]![frmReportSetup].Form![cboOperationTestID5].Requery
52 Forms![frmOrder]![frmReportSetup].Form![cboSignFromOperationID].Requery
53
54 '
55 ' Set AssemblyID in form Start as reference for Slot opening data queries
56 '
57
58 ' Forms![frmAssembly]![frmOrderLine].Form![chkAllinWO] = 0
59 Forms![Start]![txtMinAssemblyID] = 0
60 Forms![Start]![txtMaxAssemblyID] = DMax("[AssemblyID]", "tblAssembly")
61 Forms![Start]![txtOrderLineID] = Me![cboOrderLine]
62
63 If DLookup("[InclAllLines]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine]) Then
64     Forms![Start]![txtInclAllLines] = 360
65 Else
66     Forms![Start]![txtInclAllLines] = 0
67 End If
68
69
70 Exit_cboOrderLine_AfterUpdate:
71 Exit Sub
72
73 Err_cboOrderLine_AfterUpdate:
74 MsgBox Err.Description
75 Resume Exit_cboOrderLine_AfterUpdate
76
77 End Sub
78
79 Private Sub cboOrderLine_Change()
80 '
81 ' Updates values in cboOperation based on teh selected WO
82 '
83 Forms![frmOrder]![frmOperationBOM].Form![cboOperationID].Requery
84 End Sub
85
86 Private Sub cboOrderLine_GotFocus()
87     cboOrderLine.Requery
88 End Sub
89
90 Private Sub cmdAsBuiltBOM_Click()
91 '
92 ' Generate new records in tblAsBuiltBOM: AssemblyID, OperationBOMID, Qty
93 ' For the given work order, it is an OperationBOMID for each occurrence of a part in an
    operation
94 ' Qty denotes the number of a given part being used in the given operation
95 '
96 ' 02 Dec 2011 TM
97 '
98 On Error GoTo Err_cmdAsBuiltBOM_Click
99
100 ' 06Dec2011 TM: kontrol på at vikletråd og aksialtråd er merket av bør legges inn her!!!
101
```

```
102 Dim dbs As Database
103 Dim rst1 As Recordset, rst2 As Recordset, rst3 As Recordset
104 Dim stSql1 As String, stSql2 As String
105 Dim N As Integer
106
107
108
109
110 stSql1 = "SELECT tblAssembly.OrderLineID, tblAssembly.AssemblyID, "
111 stSql1 = stSql1 & "tblOperationBOM.OperationBOMID, tblOperationBOM.Qty, "
112 stSql1 = stSql1 & "tblOrderLineBOM.Usage "
113 stSql1 = stSql1 & "FROM tblAssembly INNER JOIN (tblOrderLineBOM INNER JOIN "
114 stSql1 = stSql1 & "tblOperationBOM ON tblOrderLineBOM.OrderLineBOMID = "
115 stSql1 = stSql1 & "tblOperationBOM.OrderLineBOMID) ON tblAssembly.OrderLineID = "
116 stSql1 = stSql1 & "tblOrderLineBOM.OrderLineID "
117 stSql1 = stSql1 & "WHERE (tblAssembly.OrderLineID)=" &
[Forms]![frmOrder]![cboOrderLine]
118 stSql1 = stSql1 & " AND (tblOrderLineBOM.Usage)<>2"
119
120 N = 0 ' Counter for each record being appended to tblAsBuiltBOM
121
122 DoCmd.Hourglass True
123
124 Set dbs = CurrentDb
125 '
126 ' Open list of all assemblies in work order
127 '
128 Set rst1 = dbs.OpenRecordset(stSql1, dbOpenDynaset)
129 Set rst2 = dbs.OpenRecordset("tblAsBuiltBOM", dbOpenDynaset)
130
131 Do While Not rst1.EOF
132 stSql3 = "(AssemblyID)=" & rst1!AssemblyID & "AND (OperationBOMID)=" &
rst1!OperationBOMID
133 If DCount("[AssemblyID]", "tblAsBuiltBOM", stSql3) = 0 Then
134 rst2.AddNew
135 rst2!AssemblyID = rst1!AssemblyID
136 rst2!OperationBOMID = rst1!OperationBOMID
137 rst2!Qty = rst1!Qty
138 rst2.Update
139 N = N + 1
140 End If
141 rst1.MoveNext
142 Loop
143
144
145 DoCmd.Hourglass False
146
147 rst1.Close
148 rst2.Close
149
150 dbs.Close
151
152 If N > 0 Then
153 Forms![frmOrder]![frmOrderLineDetails].Form![BOMUpdate] = Now
154 MsgBox N & " Records are generated"
```

```
155 Else
156     MsgBox "Records have been updated"
157 End If
158
159
160
161
162
163
164
165
166
167 'Dim stDocName As String
168 '
169 " dcount("[OrderLineID]", "tblOperation", "[OrderLineID]=" &
[Forms]![frmOrder]![cboOrderLine] & " and [WorkStationID]=1")
170 '
171 ' If DCount("[OrderLineID]", "qryAsBuiltBOMList") = 0 Then
172 '     DoCmd.SetWarnings False
173 '     stDocName = "qryappAsBuiltBOM"
174 '     DoCmd.OpenQuery stDocName, acViewNormal, acEdit
175 '     DoCmd.SetWarnings True
176 ' Else
177 '     MsgBox "Records has been updated"
178 ' End If
179
180 Exit_cmdAsBuiltBOM_Click:
181 Exit Sub
182
183 Err_cmdAsBuiltBOM_Click:
184 MsgBox Err.Description
185 Resume Exit_cmdAsBuiltBOM_Click
186
187 End Sub
188
189 Private Sub cmdAsBuiltOperations_Click()
190 '
191 ' Generate new records in tblAsBuiltOperation
192 ' inserting OperationID for each AssemblyID
193 ' The code must be rerun if more assemblies or new operations are added
194 '
195 ' 23 Nov 2011 TM
196 '
197
198 On Error GoTo Err_cmdAsBuiltOperations_Click
199
200 Dim dbs As Database
201 Dim rst1 As Recordset, rst2 As Recordset, rst3 As Recordset
202 Dim stSql1 As String, stSql2 As String, stSql3 As String, stSql4 As String
203 Dim N As Integer
204
205 stSql1 = "SELECT tblAssembly.AssemblyID, tblAssembly.OrderLineID "
206 stSql1 = stSql1 & "FROM tblAssembly "
207 stSql1 = stSql1 & "WHERE (tblAssembly.OrderLineID) = " &
[Forms]![frmOrder]![cboOrderLine]
```

```
208
209     stSql2 = "SELECT tblOperation.OrderLineID, tblOperation.OperationID "
210     stSql2 = stSql2 & "FROM tblOperation "
211     stSql2 = stSql2 & "WHERE (tblOperation.OrderLineID) = " &
[Forms]![frmOrder]![cboOrderLine]
212
213     N = 0 ' Counter for each record being appended to tblAsBuiltOperation
214
215     ' DoCmd.Hourglass True
216
217     Set dbs = CurrentDb
218     '
219     ' Open list of all assemblies in work order
220     '
221     Set rst1 = dbs.OpenRecordset(stSql1, dbOpenDynaset)
222     '
223     ' Open list of operations in work order
224     '
225     Set rst2 = dbs.OpenRecordset(stSql2, dbOpenDynaset)
226
227     '
228     ' loop through each assembly
229     '
230     Do While Not rst1.EOF
231
232     stSql3 = "SELECT tblAsBuiltOperation.AssemblyID, tblAsBuiltOperation.OperationID "
233     stSql3 = stSql3 & "FROM tblAsBuiltOperation "
234     stSql3 = stSql3 & "WHERE (tblAsBuiltOperation.AssemblyID) = " & rst1!AssemblyID
235     '
236     ' Open list of as built operations for given assembly
237     '
238     Set rst3 = dbs.OpenRecordset(stSql3, dbOpenDynaset)
239
240     '
241     ' Loop through each operation for the given assembly
242     '
243     Do While Not rst2.EOF
244     stSql4 = "(AssemblyID)=" & rst1!AssemblyID & " AND (OperationID)=" &
rst2!OperationID
245     If DCount("[AsBuiltOperationID]", "tblAsBuiltOperation", stSql4) = 0 Then
246     rst3.AddNew
247     rst3!AssemblyID = rst1!AssemblyID
248     rst3!OperationID = rst2!OperationID
249     rst3.Update
250     N = N + 1
251     End If
252     rst2.MoveNext
253     Loop
254     rst2.MoveFirst
255     rst1.MoveNext
256     Loop
257
258     ' DoCmd.Hourglass False
259
260     rst1.Close
```

```
261  rst2.Close
262  rst3.Close
263
264  dbs.Close
265
266  If N > 0 Then
267      Forms![frmOrder]![frmOrderLineDetails].Form![OperationUpdate] = Now
268      MsgBox N & " Records are generated"
269  Else
270      MsgBox "Records have been updated"
271  End If
272
273  Exit_cmdAsBuiltOperations_Click:
274      Exit Sub
275
276  Err_cmdAsBuiltOperations_Click:
277      MsgBox Err.Description
278      Resume Exit_cmdAsBuiltOperations_Click
279
280  End Sub
281
282  Private Sub cmdAsBuiltTest_Click()
283  '
284  '  Generate records in tblAsBuiltTest
285  '  Based on records in tblAsBuiltOperations and tblOperationTest
286  '
287  '  24 Nov 2011 TM
288  '  updated 25 Apr 2012 TM: qrydelAsBuiltTestObsolete
289  '
290  On Error GoTo Err_cmdAsBuiltTest_Click
291
292  Dim dbs As Database
293  Dim rst1 As Recordset, rst2 As Recordset, rst3 As Recordset
294  Dim stSql1 As String, stSql2 As String
295  Dim N As Integer
296
297
298  stSql1 = "SELECT tblAsBuiltOperation.AsBuiltOperationID, "
299  stSql1 = stSql1 & "tblOperationTest.OperationTestID, tblAssembly.OrderLineID "
300  stSql1 = stSql1 & "FROM tblAssembly INNER JOIN (tblOperationTest INNER JOIN "
301  stSql1 = stSql1 & "tblAsBuiltOperation ON tblOperationTest.OperationID = "
302  stSql1 = stSql1 & "tblAsBuiltOperation.OperationID) ON tblAssembly.AssemblyID = "
303  stSql1 = stSql1 & "tblAsBuiltOperation.AssemblyID "
304  stSql1 = stSql1 & "WHERE (tblAssembly.OrderLineID)=" &
[Forms]![frmOrder]![cboOrderLine]
305
306  N = 0 ' Counter for each record being appended to tblAsBuiltTest
307
308  DoCmd.Hourglass True
309
310  '
311  '  Remove records in tblAsBuiltTest if operation has been changed for given test
312  '
313  DoCmd.SetWarnings False
```



```
314 DoCmd.OpenQuery "qrydelAsBuiltTestObsolete", acNormal, acEdit
315 DoCmd.SetWarnings True
316
317 Set dbs = CurrentDb
318 '
319 ' Open list of all assemblies in work order
320 '
321 Set rst1 = dbs.OpenRecordset(stSql1, dbOpenDynaset)
322 Set rst2 = dbs.OpenRecordset("tblAsBuiltTest", dbOpenDynaset)
323
324 Do While Not rst1.EOF
325     stSql3 = "(AsBuiltOperationID)=" & rst1!AsBuiltOperationID & " AND
(OperationTestID)=" & rst1!OperationTestID
326     If DCount("[AsBuiltTestID]", "tblAsBuiltTest", stSql3) = 0 Then
327         rst2.AddNew
328         rst2!AsBuiltOperationID = rst1!AsBuiltOperationID
329         rst2!OperationTestID = rst1!OperationTestID
330         rst2.Update
331         N = N + 1
332     End If
333     rst1.MoveNext
334 Loop
335
336
337 DoCmd.Hourglass False
338
339 rst1.Close
340 rst2.Close
341
342 dbs.Close
343
344 If N > 0 Then
345     Forms![frmOrder]![frmOrderLineDetails].Form![TestsUpdate] = Now
346     MsgBox N & " Records are generated"
347 Else
348     MsgBox "Records have been updated"
349 End If
350
351
352 Dim stDocName As String
353 '
354 ' If DCount("[OrderLineID]", "qryAsBuiltTestList") = 0 Then
355 '     DoCmd.SetWarnings False
356 '     stDocName = "qryappAsBuiltTest"
357 '     DoCmd.OpenQuery stDocName, acViewNormal, acEdit
358 '     DoCmd.SetWarnings True
359 ' Else
360 '     MsgBox "Records has been updated"
361 ' End If
362
363
364
365
366 Exit_cmdAsBuiltTest_Click:
```

```
367 Exit Sub
368
369 Err_cmdAsBuiltTest_Click:
370 MsgBox Err.Description
371 Resume Exit_cmdAsBuiltTest_Click
372
373 End Sub
374
375 Private Sub cmdNewOrder_Click()
376 '
377 ' Open form to add new Order as generated by MfgPro
378 '
379 On Error GoTo Err_cmdNewOrder_Click
380
381
382 DoCmd.OpenForm "frmNewOrder"
383 DoCmd.GoToRecord acDataForm, "frmNewOrder", acNewRec
384
385 Exit_cmdNewOrder_Click:
386 Exit Sub
387
388 Err_cmdNewOrder_Click:
389 MsgBox Err.Description
390 Resume Exit_cmdNewOrder_Click
391
392 End Sub
393
394 Private Sub cmdNewWorkOrder_Click()
395 '
396 ' Open form to add new Order as generated by MfgPro
397 '
398 On Error GoTo Err_cmdNewWorkOrder_Click
399
400
401 DoCmd.OpenForm "frmNewWorkOrder"
402 DoCmd.GoToRecord acDataForm, "frmNewWorkOrder", acNewRec
403
404 Exit_cmdNewWorkOrder_Click:
405 Exit Sub
406
407 Err_cmdNewWorkOrder_Click:
408 MsgBox Err.Description
409 Resume Exit_cmdNewWorkOrder_Click
410
411 End Sub
412
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_FrmOrderDetail"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11 Private Sub cboOrderRegSign_GotFocus()
12 '
13 ' Check that information has been entered in the fields
14 '
15     If IsNull(Me![ProdOrder]) Then
16         MsgBox "Order number not entered!"
17         Exit Sub
18     End If
19
20     If IsNull(Me![LocationID]) Then
21         MsgBox "Location not selected!"
22         Exit Sub
23     End If
24
25     If IsNull(Me![Customer]) Then
26         MsgBox "Customer not entered!"
27         Exit Sub
28     End If
29
30     If IsNull(Me![Description]) Then
31         MsgBox "Description not entered!"
32         Exit Sub
33     End If
34
35     If IsNull(Me![OrderRegDate]) Then
36         Me![OrderRegDate] = Now
37     End If
38
39 End Sub
40
41 Private Sub OrderRegDate_DbClick(Cancel As Integer)
42     Me![OrderRegDate] = Now
43 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmOrderLineDetails"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboCsgID_AfterUpdate()
13 '
14 ' Set base pipe ID
15 '
16 Me![BpID] = Me![txtPipeID]
17 End Sub
18
19 Private Sub cboPipeOD_AfterUpdate()
20 Me![cboCsgID].Requery
21 '
22 ' set default values
23 '
24 If IsNull(Me![BpOD]) Then Me![BpOD] = Me![txtPipeOD]
25 If IsNull(Me![nAxialRods]) Then Me![nAxialRods] = Me![txtAxialRods]
26 If IsNull(Me![PerfDia]) Then Me![PerfDia] = Me![txtPerfD]
27 If IsNull(Me![NomScreenOD]) Then Me![NomScreenOD] = Me![BpOD] + 2 *
    Me![txtWrapWireThickness]
28
29
30 End Sub
31
32 Private Sub cboProductTypeID_AfterUpdate()
33 '
34 ' Add default values
35 '
36 If IsNull(Me![WrapWireMass]) Then Me![WrapWireMass] = Me![txtWrapWireMass]
37 If IsNull(Me![AxialWireMass]) Then Me![AxialWireMass] = Me![txtAxialWireMass]
38
39 End Sub
40
41 Private Sub cmdNewAssyRecords_Click()
42 On Error GoTo Err_cmdNewAssyRecords_Click
43
44 '
45 ' Generate new records in tblAssembly and tblWrap
46 '
47 ' 03 Dec 2011 TM
48 '
49 Dim dbs As Database
```

```
50 Dim rst As Recordset, rst2 As Recordset
51 Dim tblAssyQty As Integer, Qty As Integer, num_pipes As Integer
52 Dim NSec As Integer, NRods As Integer
53 Dim I As Integer, J As Integer, M As Integer, N As Integer
54 Dim StartPos As Single, WrapLength As Single, L1 As Single, L2 As Single
55 Dim Screen As Boolean
56 Dim WO As String, OLID As Long
57 Dim stSql As String, stSql1 As String, stSql2 As String, stStatus As String
58
59
60 If IsNull(Me![OrderLineID]) Then
61     MsgBox "Work Order not registered"
62     Exit Sub
63 End If
64 OLID = Nz(Me![OrderLineID], 0)
65
66 If IsNull(Me![WorkOrder]) Then
67     MsgBox "Work Order number not registered"
68     Exit Sub
69 End If
70 WO = Me![WorkOrder]
71
72 If Nz(Me![Quantity], 0) < 1 Then
73     MsgBox "Quantity not recorded"
74     Exit Sub
75 End If
76 Qty = Nz(Me![Quantity], 0)
77
78 '
79 ' Check if product is screen
80 ' if so, check screen details
81 '
82 Screen = False
83 If IsNull(Me![txtProductTypeScreen]) Then
84     MsgBox "Product type not recorded"
85     Exit Sub
86 ElseIf Me![txtProductTypeScreen] Then
87     If Nz(Me![NSections], 0) < 1 Then
88         MsgBox "Number of wrap sections not recorded"
89         Exit Sub
90     End If
91     NSec = Me![NSections]
92     If DCount("[Section]", "tblWrapLength", "[OrderLineID]=" & OLID) > NSec Then
93         MsgBox "Too many Sec. recorded in Wrap Length table"
94         Exit Sub
95     End If
96     If DCount("[StartPos]", "tblWrapLength", "[OrderLineID]=" & OLID) > NSec Then
97         MsgBox "Too many StartPos recorded in Wrap Length table"
98         Exit Sub
99     End If
100     If DCount("[WrapLength]", "tblWrapLength", "[OrderLineID]=" & OLID) > NSec Then
101         MsgBox "Too many WrapLength recorded in Wrap Length table"
102         Exit Sub
103     End If
```

```
104
105 ' check records in tblWrapLength - nominal start position and wrap length for each section
106 ' if both lengths are entered for all sections
107
108     If NSec > 1 Then
109         If DCount("[Section]", "tblWrapLength", "[OrderLineID]=" & OLID) = NSec Then
110             If DCount("[StartPos]", "tblWrapLength", "[OrderLineID]=" & OLID) = NSec Then
111                 If DCount("[WrapLength]", "tblWrapLength", "[OrderLineID]=" & OLID) = NSec
Then
112                     J = 1
113                     L2 = 0
114                     Do While J <= NSec
115                         stSql2 = "(OrderLineID)=" & OLID & " AND (Section)=" & J
116                         If IsNull(DLookup("[Section]", "tblWrapLength", stSql2)) Then
117                             MsgBox "Section " & J & " is missing"
118                             Exit Sub
119                         End If
120                         L1 = DLookup("[StartPos]", "tblWrapLength", stSql2)
121                         If L2 > L1 Then
122                             MsgBox "StartPos for section " & J & " is too small!"
123                             Exit Sub
124                         End If
125                         L2 = DLookup("[StartPos]", "tblWrapLength", stSql2) +
DLookup("[WrapLength]", "tblWrapLength", stSql2)
126                         J = J + 1
127                     Loop
128                 End If
129             End If
130         End If
131     End If
132
133     If Nz(Me![nAxialRods], 0) < 1 Then
134         MsgBox "Number of axial rods not recorded"
135         Exit Sub
136     End If
137     Screen = True
138 End If
139
140 '
141 ' Count if any or how many records already in tblAssembly
142 '
143 stSql = "(tblAssembly.OrderLineID) = " & OLID
144 tblAssyQty = Nz(DCount("[OrderLineID]", "tblAssembly", stSql), 0)
145
146 Set dbs = CurrentDb
147 DoCmd.Hourglass True
148
149 If tblAssyQty = Qty Then
150     ' MsgBox "Records has already been updated"
151     ' Exit Sub
152 ElseIf tblAssyQty > Qty Then
153     ' MsgBox "Quantity has been reduced. Records should be deleted. This feature is not
developed."
154     Me![Quantity] = tblAssyQty
155     ' Exit Sub
```



```
156 ElseIf tblAssyQty < Qty Then
157     Set rst = dbs.OpenRecordset("tblAssembly", dbOpenDynaset)
158     For num_pipes = tblAssyQty + 1 To Qty
159         s_nr = WO & "-" & mk3digit(num_pipes, Qty)
160         rst.AddNew
161         rst!OrderLineID = OrderLineID
162         rst!Serialno = s_nr
163         rst.Update
164         M = M + 1
165     Next num_pipes
166     rst.Close
167 End If
168
169 '
170 ' Loop through records in tblAssembly and check if record for each wrap section exists in
tblWrap
171 ' If not, add records in tblWrap
172 '
173 If Screen Then
174     stSql = "SELECT tblAssembly.OrderLineID, tblAssembly.AssemblyID "
175     stSql = stSql & "FROM tblAssembly "
176     stSql = stSql & "WHERE(tblAssembly.OrderLineID) = " & OLID
177     Set rst = dbs.OpenRecordset(stSql, dbOpenDynaset)
178     Set rst2 = dbs.OpenRecordset("tblWrap", dbOpenDynaset)
179     rst.MoveFirst
180     Do While Not rst.EOF
181         I = 0
182         Do While I < NSec
183             I = I + 1
184             stSql1 = "(AssemblyID)=" & rst!AssemblyID & " AND (WrapSection)=" & I
185             If DCount("[WrapID]", "tblWrap", stSql1) = 0 Then
186                 ' lookup start pos and lenght for given section
187                 stSql2 = "(OrderLineID)=" & OLID & " AND (Section)=" & I
188                 StartPos = Nz(DLookup("[StartPos]", "tblWrapLength", stSql2), 0)
189                 WrapLength = Nz(DLookup("[WrapLength]", "tblWrapLength", stSql2), 0)
190                 rst2.AddNew
191                 rst2!AssemblyID = rst!AssemblyID
192                 rst2!WrapSection = I
193                 rst2!WrapTypeID = 1
194                 rst2!WrapStartPos = StartPos
195                 rst2!WrapLength = WrapLength
196                 rst2.Update
197                 N = N + 1
198             End If
199         Loop
200     Loop
201     rst.MoveNext
202 Loop
203 End If
204
205 rst.Close
206 dbs.Close
207 DoCmd.Hourglass False
208
```

```
209   If M > 0 Then
210       stStatus = M & " assemblys added"
211   End If
212
213   If N > 0 Then
214       If M > 0 Then
215           Status = Status & " and total " & N & " wrap sections added"
216       Else
217           Status = "Total " & N & " wrap sections added"
218       End If
219   End If
220
221   If M + N > 0 Then
222       Me!AssemblyUpdate = Now
223       MsgBox Status
224   Else
225       MsgBox "Records has been updated"
226       If IsNull(Me!AssemblyUpdate) Then
227           Me!AssemblyUpdate = Now
228       End If
229   End If
230
231 Exit_cmdNewAssyRecords_Click:
232 Exit Sub
233
234 Err_cmdNewAssyRecords_Click:
235 DoCmd.Hourglass False
236 ' Write error message to tblBug
237 Forms![Start]![txtBug] = "cmdNewAssyRecords_Click: " & Err.Description
238 DoCmd.SetWarnings False
239 DoCmd.OpenQuery "qryappBug", acNormal, acEdit
240 DoCmd.SetWarnings True
241 '
242 MsgBox Err.Description
243 Resume Exit_cmdNewAssyRecords_Click
244
245 End Sub
246
247 Private Sub Option73_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
248 Me![OLStatusDate] = Now
249 End Sub
250
251 Private Sub Option75_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
252 Me![OLStatusDate] = Now
253 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmOrderLineSlot"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub chkInclAllLines_AfterUpdate()
13
14     If Me![chkInclAllLines] Then
15         Forms![Start]![txtInclAllLines] = 360
16     Else
17         Forms![Start]![txtInclAllLines] = 0
18     End If
19
20 End Sub
21
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmPopulateAllocated"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cmdCancel_Click()
13     DoCmd.Close
14 End Sub
15
16 Private Sub cmdUpdateRecords_Click()
17     '
18     ' Add records to tblAllocated
19     '
20     Dim OperatorID As Long
21     Dim nRecords As Integer
22     Dim dbs As Database
23     Dim rst As DAO.Recordset
24
25     OperatorID = Forms![Start]![txtOperatorID]
26     If OperatorID = 0 Then
27         Exit Sub
28     End If
29
30     If IsNull(Me![SerialNo]) Then
31         MsgBox "Enter Serial no from"
32         Me![SerialNo].SetFocus
33         Exit Sub
34     End If
35
36     If IsNull(Me![QtySerialNo]) Then
37         MsgBox "Enter Quantity in serial number range"
38         Me![QtySerialNo].SetFocus
39         Exit Sub
40     End If
41
42     Me![Qty] = 1
43
44     Set dbs = CurrentDb
45     Set rst = dbs.OpenRecordset("tblAllocated", dbOpenDynaset)
46     For nRecords = Me![SerialNo] + 1 To Me![QtySerialNo] + Me![SerialNo] - 1
47         rst.AddNew
48         rst!OrderLineBOMID = Me![OrderLineBOMID]
49         rst!PO = Me![PO]
50         rst!POline = Me![POline]
```

```
51      rst!POship = Me![POship]
52      rst!Heat = Me![Heat]
53      rst!Qty = 1
54      rst!Serialno = nRecords
55      rst!AllocatedSign = OperatorID
56      rst!AllocatedDate = Now
57      rst.Update
58  Next nRecords
59  rst.Close
60  dbs.Close
61
62  Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Requery
63  DoCmd.Close
64  Forms![frmWareHouse]![IstOrderLineBOM].Requery
65
66 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmProgressBar"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub txtProgress_AfterUpdate()
13
14     Me![Box1].Width = (Me![txtProgress] - Int(Me![txtProgress])) * Me![Box0].Width
15     Select Case Int(Me![txtProgress])
16     Case 1
17         Me![Box1].BackColor = vbGreen
18     Case 2
19         Me![Box1].BackColor = vbBlack
20     Case 3
21         Me![Box1].BackColor = vbYellow
22     Case 4
23         Me![Box1].BackColor = vbRed
24     End Select
25
26 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmQC"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboAssemblyDateFrom_BeforeUpdate(Cancel As Integer)
13
14 End Sub
15
16 Private Sub cboAssemblyDateFrom_LostFocus()
17
18     If IsNull(Me![cboAssemblyDateFrom]) Then
19         Me![fraStatus] = 1
20     End If
21
22 End Sub
23
24 Private Sub cboAssemblyDateTo_AfterUpdate()
25
26     If Me![cboAssemblyDateFrom] > 0 Then
27         Me![fraStatus] = 4
28     ElseIf Me![fraStatus] = 4 Then
29         Me![fraStatus] = 1
30     End If
31
32 End Sub
33
34 Private Sub cboAssemblyID_AfterUpdate()
35
36 '
37 ' Set default values after assembly has been selected
38 '
39
40 On Error GoTo Err_cboAssemblyID_AfterUpdate
41
42 Dim stSql As String
43 '
44 ' Set AssemblyID in form Start as reference for Slot opening data queries
45 '
46
47 ' Forms![frmAssembly]![frmOrderLine].Form![chkAllinWO] = 0
48 Forms![Start]![txtMinAssemblyID] = Me![cboAssemblyID]
49 Forms![Start]![txtMaxAssemblyID] = Me![cboAssemblyID]
50
```



```
51 '
52 ' Requery list box to reflect selected assembly
53 ' Select record in frmWrapDetail form based on selected assembly
54 '
55
56 IstWrapID.Requery
57 Me![IstWrapID] = DMin("[WrapID]", "tblWrap", "[AssemblyID]=" & Me![cboAssemblyID])
58
59 IstWrapIDHardness.Requery
60 Me![IstWrapIDHardness] = Me![IstWrapID]
61
62
63 '
64 ' Select assemblies in report tab
65 '
66 Me![cboAssyFrom] = Me![txtSerialNo]
67 Me![cboAssyTo] = Me![txtSerialNo]
68 Me![fraSelectAssy] = 1
69
70
71
72
73 Exit_cboAssemblyID_AfterUpdate:
74 Exit Sub
75
76 Err_cboAssemblyID_AfterUpdate:
77 MsgBox Err.Description
78 Resume Exit_cboAssemblyID_AfterUpdate
79
80 End Sub
81
82 Private Sub cboAssemblyID_GotFocus()
83 'cboAssemblyID.Requery
84 End Sub
85
86
87
88 Private Sub cboAssyFrom_Click()
89 Me![fraSelectAssy] = 3
90 End Sub
91
92 Private Sub cboAssyTo_Click()
93 Me![fraSelectAssy] = 3
94 End Sub
95
96 Private Sub cboOrder_AfterUpdate()
97 'cboOrderLine.Requery
98 End Sub
99
100 Private Sub cboOrder_GotFocus()
101 ' Terje 29Jan11
102 ' Filtration of orders
103 '
```

```
104 ' SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location,
    tblOrder.CompletedSign
105 ' FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
106 ' ORDER BY tblOrder.ProdOrder;
107
108 Dim Sql As String
109
110 Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign],
    [tblLocation.LocationID] "
111 Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID =
    tblLocation.LocationID "
112
113
114 If Me![optionOrderSelect] = 1 Then
115     Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
116 End If
117
118 Forms![frmQC]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"
119
120
121 End Sub
122
123 Private Sub cboOrderLine_AfterUpdate()
124 '
125 ' If empty, set Order
126 '
127 On Error GoTo Err_cboOrderLine_AfterUpdate
128
129 If IsNull(Me![cboOrder]) Then
130     Me![cboOrder] = DLookup("[OrderID]", "tblOrderLine", "[OrderLineID] = " &
        Me![cboOrderLine])
131 End If
132
133
134 '
135 ' Set parameters for slot data queries in Start form
136 '
137 Forms![Start]![txtOrderLineID] = Me![cboOrderLine]
138 Forms![Start]![txtLocationID] = Me![txtLocationID]
139
140 If DLookup("[InclAllLines]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine]) Then
141     Forms![Start]![txtInclAllLines] = 360
142 Else
143     Forms![Start]![txtInclAllLines] = 0
144 End If
145
146 Forms![Start]![txtMinAssemblyID] = 0
147 Forms![Start]![txtMaxAssemblyID] = 0
148 '
149 ' Requery Assembly, cboWrapID and clear combo boxes
150 '
151 cboAssemblyID.Requery
152 Me![cboAssemblyID] = Null
153 lstOrderLineBOM.Requery
154 cboWrapID.Requery
```

```
155     cboAssyFrom.Requery
156     cboAssyTo.Requery
157     cboOperationID.Requery
158     Me![cboReportID] = Null
159     Me![cboAssemblyDateFrom] = Null
160     Me![cboAssemblyDateTo] = Null
161     Me![cboShipDateFrom] = Null
162     Me![cboShipDateTo] = Null
163     Me![cboOperationID] = Null
164     Me![fraStatus] = 1
165     Me![fraOperations] = 2
166     Me![cboAssemblyDateFrom].Requery
167     Me![cboAssemblyDateTo].Requery
168     Me![cboShipDateFrom].Requery
169     Me![cboShipDateTo].Requery
170
171 '
172 ' Select assemblies in report tab
173 '
174     Me![cboAssyFrom] = DMin("[Serialno]", "tblAssembly", "[OrderLineID] = " &
Me![cboOrderLine])
175     Me![cboAssyTo] = DMax("[Serialno]", "tblAssembly", "[OrderLineID] = " &
Me![cboOrderLine])
176     Me![fraSelectAssy] = 2
177
178
179 Exit_cboOrderLine_AfterUpdate:
180     Exit Sub
181
182 Err_cboOrderLine_AfterUpdate:
183     MsgBox Err.Description
184     Resume Exit_cboOrderLine_AfterUpdate
185
186 End Sub
187
188 Private Sub cboOrderLine_GotFocus()
189 ' SELECT tblOrderLine.OrderLineID, tblOrderLine.WorkOrder, tblOrderLine.PartNo,
tblOrderLine.Description, tblOrder.CompletedSign
190 ' FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID
191 ' WHERE (((tblOrder.CompletedSign)=0));
192 On Error GoTo Err_cboOrderLine_GotFocus
193
194     Dim Sql As String
195
196     Sql = "SELECT [OrderLineID], [WorkOrder], [PartNo], [tblOrderLine.Description],
[CompletedSign] "
197     Sql = Sql & "FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID =
tblOrderLine.OrderID "
198     If Me![optionOrderSelect] = 1 Then
199 ' noe muffens når denne linjen ikke inkluderes - WHERE mangler
200         Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
201     Else
202         Sql = Sql & "WHERE [tblOrder.CompletedSign] >= 0 "
203     End If
204
```

```
205 If Not IsNull(Me![cboOrder]) Then
206     Sql = Sql & "AND [tblOrder.OrderID] = " & Me![cboOrder]
207 End If
208
209 Forms![frmQC]![cboOrderLine].RowSource = Sql & " ORDER BY tblOrderLine.WorkOrder;"
210
211 Exit_cboOrderLine_GotFocus:
212 Exit Sub
213
214 Err_cboOrderLine_GotFocus:
215 MsgBox Err.Description
216 Resume Exit_cboOrderLine_GotFocus
217
218 End Sub
219
220 Private Sub cboReportID_GotFocus()
221     cboReportID.Requery
222 End Sub
223
224 Private Sub cboShipDateFrom_LostFocus()
225     If IsNull(Me![cboShipDateFrom]) Then
226         Me![fraStatus] = 1
227     End If
228 End Sub
229
230 Private Sub cboShipDateTo_AfterUpdate()
231
232     If Me![cboShipDateFrom] > 0 Then
233         Me![fraStatus] = 6
234     ElseIf Me![fraStatus] = 6 Then
235         Me![fraStatus] = 1
236     End If
237
238 End Sub
239
240 Private Sub cboWrapID_AfterUpdate()
241 '
242 ' Set AssemblyID and IstWrapID to select the corect record in frmQCWrap
243 '
244     Me![cboAssemblyID] = Me![cboWrapID].Column(2)
245     Me![IstWrapID] = Me![cboWrapID]
246     IstWrapID.Requery
247
248 End Sub
249
250 Private Sub cboWrapIDHardness_AfterUpdate()
251 '
252 ' Set AssemblyID and IstWrapIDHardness to select the corect record in frmQCWrap
253 '
254     Me![cboAssemblyID] = Me![cboWrapIDHardness].Column(2)
255     Me![IstWrapIDHardness] = Me![cboWrapIDHardness]
256     IstWrapIDHardness.Requery
257
258 End Sub
```

```
259
260 Private Sub cmdPullTestReport_Click()
261 DoCmd.OpenReport "rptPullTest2", acViewPreview
262 End Sub
263
264 Private Sub cmdrepAllocated_Click()
265 DoCmd.OpenReport "rptAllocated", acViewPreview
266 End Sub
267
268 Private Sub cmdrepTest_Click()
269 On Error GoTo cmdrepTest_Click_Err
270 '
271 ' open report for assembly by operation
272 '
273 Dim stSql As String
274
275
276 If IsNull(Me![cboAssyFrom]) Then
277 MsgBox "Select Assembly"
278 Exit Sub
279 End If
280 If IsNull(Me![cboAssyTo]) Then
281 MsgBox "Select Assembly"
282 Exit Sub
283 End If
284 If IsNull(Me![cboReportID]) Then
285 MsgBox "Select Report"
286 Exit Sub
287 End If
288
289 stSql = "[OrderLineID]=" & Me![cboOrderLine] & " and [ReportID]=" & Me![cboReportID]
290 If DCount("[ReportID]", "tblReport", stSql) = 0 Then
291 MsgBox "Select Report"
292 Exit Sub
293 End If
294
295 Select Case Me![fraStatus]
296 Case 4
297 DoCmd.OpenReport "rptTestsColComplDate", acViewPreview
298 Exit Sub
299 Case 6
300 DoCmd.OpenReport "rptTestsColShipDate", acViewPreview
301 Exit Sub
302 Case Else
303 DoCmd.OpenReport "rptTestsCol", acViewPreview
304 Exit Sub
305 End Select
306
307
308 cmdrepTest_Click_Exit:
309 Exit Sub
310
311 cmdrepTest_Click_Err:
312 MsgBox Error$
```

```
313 Resume cmdrepTest_Click_Exit
314
315 End Sub
316
317 Private Sub cmdrptAssy_Click()
318 On Error GoTo cmdrptAssy_Click_Err
319 '
320 ' open report based on selections
321 '
322
323 If IsNull(Me![cboAssyFrom]) Then
324     MsgBox "Select Assembly"
325     Exit Sub
326 End If
327 If IsNull(Me![cboAssyTo]) Then
328     MsgBox "Select Assembly"
329     Exit Sub
330 End If
331
332 If Me![fraOperations] = 1 Then
333     Select Case Me![fraStatus]
334     Case 4
335         DoCmd.OpenReport "rptOperationsComplDate", acViewPreview
336         Exit Sub
337     Case 6
338         DoCmd.OpenReport "rptOperationsShipDate", acViewPreview
339         Exit Sub
340     Case Else
341         DoCmd.OpenReport "rptOperations", acViewPreview
342         Exit Sub
343     End Select
344 End If
345
346 If IsNull(Me![cboOperationID]) Then
347     MsgBox "Select Operation"
348     Exit Sub
349 End If
350
351 Select Case Me![fraStatus]
352 Case 4
353     DoCmd.OpenReport "rptOperationComplDate", acViewPreview
354     Exit Sub
355 Case 6
356     DoCmd.OpenReport "rptOperationShipDate", acViewPreview
357     Exit Sub
358 Case Else
359     DoCmd.OpenReport "rptOperation", acViewPreview
360     Exit Sub
361 End Select
362
363 cmdrptAssy_Click_Exit:
364 Exit Sub
365
366 cmdrptAssy_Click_Err:
```

```
367     MsgBox Error$
368     Resume cmdrptAssy_Click_Exit
369
370 End Sub
371
372 Private Sub cmdrptAssyAllOperations_Click()
373     DoCmd.OpenReport "rptOperations", acViewPreview
374
375 End Sub
376
377 Private Sub cmdSlotDetail_Click()
378     On Error GoTo cmdSlotDetail_Click_Err
379 '
380 ' open report for assembly by operation
381 '
382 If IsNull(Me![cboAssyFrom]) Then
383     MsgBox "Select Assembly"
384     Exit Sub
385 End If
386 If IsNull(Me![cboAssyTo]) Then
387     MsgBox "Select Assembly"
388     Exit Sub
389 End If
390
391 Select Case Me![fraStatus]
392 Case 4
393     DoCmd.OpenReport "rptSlotDetailComplDate", acViewPreview
394     Exit Sub
395 Case 6
396     DoCmd.OpenReport "rptSlotDetailShipDate", acViewPreview
397     Exit Sub
398 Case Else
399     DoCmd.OpenReport "rptSlotDetail", acViewPreview
400     Exit Sub
401 End Select
402
403
404 cmdSlotDetail_Click_Exit:
405     Exit Sub
406
407 cmdSlotDetail_Click_Err:
408     MsgBox Error$
409     Resume cmdSlotDetail_Click_Exit
410
411 End Sub
412
413 '-----
414 ' cmdSlotDetails_Click
415 '
416 '-----
417 Private Sub cmdSlotDetails_Click()
418     On Error GoTo cmdSlotDetails_Click_Err
419
420     DoCmd.OpenForm "frmSlotResult", acNormal, "", "", acNormal
```

```
421
422
423 cmdSlotDetails_Click_Exit:
424     Exit Sub
425
426 cmdSlotDetails_Click_Err:
427     MsgBox Error$
428     Resume cmdSlotDetails_Click_Exit
429
430 End Sub
431
432 Private Sub fraSelectAssy_AfterUpdate()
433 '
434 ' Set values in cbo boxes based on selection
435 '
436
437 If IsNull(Me![cboOrderLine]) Then
438     MsgBox "Work Order ID not selected"
439     Me![cboOrderLine].SetFocus
440     Exit Sub
441 End If
442
443 Select Case Me![fraSelectAssy]
444     Case 1
445         If IsNull(Me![cboAssemblyID]) Then
446             MsgBox "Serial number not selected"
447             Me![fraSelectAssy] = 2
448             Exit Sub
449         End If
450         Me![cboAssyFrom] = Me![txtSerialno]
451         Me![cboAssyTo] = Me![txtSerialno]
452     Case 2
453         Me![cboAssyFrom] = DMin("[Serialno]", "tblAssembly", "[OrderLineID]=" &
Me![cboOrderLine])
454         Me![cboAssyTo] = DMax("[Serialno]", "tblAssembly", "[OrderLineID]=" &
Me![cboOrderLine])
455     Case 3
456
457 End Select
458
459 End Sub
460
461 Private Sub IstOrderLineBOM_AfterUpdate()
462 '
463 ' Unhide text boxes for entering of wrap wire width and standard deviation when wrap wire
is selected
464 ' hide text boxes wehn other part is selected
465 '
466 ' TM 20120106
467 '
468
469 If DLookup("[Usage]", "tblOrderLineBOM", "[OrderLineBOMID]=" & Me![IstOrderLineBOM])
= 4 Then
470     Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireWidt].Visible =
True
```

```
471      Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireStDev].Visible =  
      True  
472      Else  
473      Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![Traceability].SetFocus  
474      Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireWidt].Visible =  
      False  
475      Forms![frmQC]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireStDev].Visible =  
      False  
476      End If  
477  
478  
479      End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmQCWirePullTest"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboWirePullTestSign_AfterUpdate()
13     Me![WirePullTestDate] = Now
14 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmScreen"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboOrder_AfterUpdate()
13     cboOrderLine.Requery
14 End Sub
15
16 Private Sub cboOrder_GotFocus()
17     ' Terje 10.10.14
18     ' Filtration of orders
19     Dim Sql As String
20
21     Sql = "SELECT [OrderID], [ProdOrder], [Description], [tblLocation.LocationID], [Location],
22     [CompletedSign] "
23
24     Sql = Sql & "FROM tblLocation INNER JOIN tblOrder ON tblLocation.LocationID =
25     tblOrder.LocationID "
26
27     If Me![optionOrderSelect] = 1 Then
28         Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
29     End If
30
31     Forms![frmScreen]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"
32 End Sub
33
34 Private Sub cboOrderLine_AfterUpdate()
35     ScreenID.Requery
36 End Sub
37
38 Private Sub cmdWrapping_Click()
39
40     'Open form for Wrapping details
41
42     On Error GoTo Err_cmdWrapping_Click
43
44     Dim stDocName As String
45     Dim stLinkCriteria As String
46
47     stDocName = "frmWrappingOrderLine"
48
```

```
49  stLinkCriteria = "[OrderLineID]=" & Me![cboOrderLine]
50  DoCmd.OpenForm stDocName, , , stLinkCriteria
51
52  Exit_cmdWrapping_Click:
53  Exit Sub
54
55  Err_cmdWrapping_Click:
56  MsgBox Err.Description
57  Resume Exit_cmdWrapping_Click
58
59  End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmShip"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub ShipDate_DblClick(Cancel As Integer)
13 '
14 ' Set todays date if assembly is signed off
15 '
16     If Not IsNull(Me![OperatorSign]) Then
17         Me![ShipDate] = Int(Now())
18     End If
19
20 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmSlotFile"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub Angle_AfterUpdate()
13
14     If Nz(Me![SlotSign]) Then
15         If MsgBox("Enter signature after modification", vbOKCancel, "ResLog") = 1 Then
16             Me![SlotDate] = Now()
17             Me![SlotSign] = ""
18         Else
19             Me![Angle] = Me![Temp]
20         End If
21     End If
22
23     Me![Temp] = ""
24
25 End Sub
26
27 Private Sub Angle_Change()
28     Me![Temp] = Me![Angle]
29 End Sub
30
31 Private Sub cboSlotSign_AfterUpdate()
32 '
33 ' Use current date and time if SlotDate is Null
34 '
35     If IsNull(Me![SlotDate]) Then
36         Me![SlotDate] = Now
37     End If
38
39 End Sub
40
41 Private Sub Form_DbClick(Cancel As Integer)
42 'Open form for slot details
43 On Error GoTo Err_Form_DbClick
44
45     Dim stDocName As String
46     Dim stLinkCriteria As String
47
48     Forms![frmSlotResult]![txtSlotFileID] = Me![SlotFileID]
49
50     stDocName = "frmSlotFileDetail"
```

```
51
52     stLinkCriteria = "[SlotFileID]=" & Me![SlotFileID]
53     DoCmd.OpenForm stDocName, , , stLinkCriteria
54
55 Exit_Form_DblClick:
56     Exit Sub
57
58 Err_Form_DblClick:
59     MsgBox Err.Description
60     Resume Exit_Form_DblClick
61
62 End Sub
63
64
65 Private Sub SlotDate_DblClick(Cancel As Integer)
66     Me![SlotDate] = Now()
67 End Sub
68
69 Private Sub SlotFileStatus_AfterUpdate()
70     On Error GoTo Err_SlotFileStatus_AfterUpdate
71
72     Dim Status
73
74     Status = Me![SlotFileStatus]
75
76     If Nz(Me![SlotSign]) Then
77         If MsgBox("Enter signature after modification", vbOKCancel, "ResLog") = 1 Then
78             Me![SlotDate] = Now()
79             Me![SlotSign] = ""
80         Else
81             Me![SlotFileStatus] = Not (Status)
82         End If
83     End If
84
85 Exit_SlotFileStatus_AfterUpdate:
86     Exit Sub
87
88 Err_SlotFileStatus_AfterUpdate:
89     MsgBox Err.Description
90     Resume Exit_SlotFileStatus_AfterUpdate
91
92 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmSlotFileImport"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12
13
14 Private Sub cmd_getfil_Click()
15     Dim fil_sti As String, kun_sti As String, kun_fil As String
16     Dim slash_pos As Integer, path_pos As Integer
17
18     cdlg.ShowOpen
19     fil_sti = cdlg.FileName
20
21     If fil_sti <> "" Then
22         slash_pos = 1
23         Do While slash_pos > 0
24             path_pos = slash_pos
25             slash_pos = InStr(slash_pos + 1, fil_sti, "\")
26         Loop
27         If path_pos > 1 Then
28             txtFileNameUnb = Right(fil_sti, Len(fil_sti) - path_pos)
29             cboPath = Left(fil_sti, path_pos)
30             txtFileNameUnbx = fil_sti
31             'MsgBox "sti : " & Left(fil_sti, path_pos)
32             'MsgBox "fil : " & Right(fil_sti, Len(fil_sti) - path_pos)
33         End If
34     End If
35
36 End Sub
37
38 Private Sub LoadSlotDataFile_Click()
39     On Error GoTo Err_LoadSlotDataFile_Click
40
41     Dim I As Integer
42     Dim AssemblyID As Long, OrderLineID As Long
43
44     Forms![frmSlotResult]![txtFile] = Me![txtFileNameUnb]
45     Forms![frmSlotResult]![txtPath] = Me![cboPath]
46
47     AssemblyID = Forms![frmSlotResult]![cboAssemblyID]
48     OrderLineID = [Forms]![Start]![txtOrderLineID]
49     I = SlotAngle(Me![txtFileNameUnbx], OrderLineID, AssemblyID)
50     DoCmd.Close acForm, "frmSlotFileImport"
```

```
51
52 Exit_LoadSlotDataFile_Click:
53     Exit Sub
54
55 Err_LoadSlotDataFile_Click:
56     MsgBox Err.Description
57     Resume Exit_LoadSlotDataFile_Click
58
59 End Sub
60
61 Private Sub txtFileNameUnb_AfterUpdate()
62     Forms![frmProduction]![ScreenID] = Me![txtScreenID]
63 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmSlotImportAngle"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cmdSlotImportAngleOK_Click()
13     Dim I As Integer
14     Dim PathFileName As String
15     Dim OrderLineID As Long, AssemblyID As Long
16
17     PathFileName = Forms![frmSlotImportAngle]![txtPathFileName]
18     OrderLineID = Forms![frmSlotImportAngle]![txtOrderLineID]
19     AssemblyID = Forms![frmSlotImportAngle]![txtAssemblyID]
20     Forms![frmSlotResult]![txtAngle] = Me![cboAngle]
21     DoCmd.Close acForm, "frmSlotImportAngle"
22
23     I = SlotImport(PathFileName, OrderLineID, AssemblyID)
24
25 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmSlotLarge"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboSlotManSign_GotFocus()
13 '
14 ' Check that type of modification has been selected
15 '
16
17 If Me![fraSlotMod] < 1 Then
18     MsgBox "Select type of slot modification!"
19     Me![fraSlotMod].SetFocus
20     Exit Sub
21 End If
22
23 If IsNull(Me![SlotManDate]) Then
24     Me![SlotManDate] = Now()
25 End If
26
27 End Sub
28
29 Private Sub fraSlotMod_AfterUpdate()
30 '
31 ' If slot modification is done and ticked off, the slot on each side
32 ' are marked as "Next" to the actual slot.
33 ' This causes the slot to appear in the large slot list and eventuell
34 ' modification to this slot can also be recorded.
35 ' A verification is done to make sure the actual slots are actual neighbour
36 ' slots by comparing slot positions.
37 '
38 Dim dbs As DAO.Database
39 Dim rst As DAO.Recordset
40 Dim stSql As String
41
42 If Me![fraSlotMod] = 2 Or Me![fraSlotMod] = 3 Then
43     Set dbs = CurrentDb
44     stSql = "SELECT tblSlotData.SlotID, tblSlotData.SlotFileID, "
45     stSql = stSql & "tblSlotData.SlotPos, tblSlotData.SlotMod "
46     stSql = stSql & "FROM tblSlotData "
47     stSql = stSql & "WHERE (tblSlotData.SlotID) >= " & Me![SlotID] - 1
48     stSql = stSql & " AND (tblSlotData.SlotID) <= " & Me![SlotID] + 1
49     stSql = stSql & " AND (tblSlotData.SlotFileID) = " & Me![SlotFileID]
50     Set rst = dbs.OpenRecordset(stSql, dbOpenDynaset)
```

```
51     rst.MoveFirst
52     If rst!SlotID = (Me![SlotID] - 1) Then
53         If (Me![SlotPos] - rst!SlotPos) < 5 Then
54             If rst!SlotMod < 2 Then ' Only if slot is not modified
55                 rst.Edit
56                 rst!SlotMod = 4
57                 rst.Update
58             End If
59         End If
60     End If
61     rst.MoveLast
62     If rst!SlotID = (Me![SlotID] + 1) Then
63         If (rst!SlotPos - Me![SlotPos]) < 5 Then
64             If rst!SlotMod < 2 Then ' Only if slot is not modified
65                 rst.Edit
66                 rst!SlotMod = 4
67                 rst.Update
68             End If
69         End If
70     End If
71     rst.Close
72     dbs.Close
73     Forms![frmSlotResult]![cboAssemblyID].Requery
74     End If
75
76
77
78 End Sub
79
80 Private Sub SlotManDate_DblClick(Cancel As Integer)
81     Me![SlotManDate] = Now
82 End Sub
83
84 Private Sub SlotWidth_AfterUpdate()
85     ' Me![fraSlotMod] = 2
86     Me![SlotManDate] = Now
87     If Me![SlotOrg] <> Me![SlotWidth] Then
88         Me![SlotChanged] = -1
89     Else
90         Me![SlotChanged] = 0
91     End If
92 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmSlotOutOfSpec"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub Combo13_AfterUpdate()
13
14     If Nz(Me![SlotManDate], 0) = 0 Then
15         Me![SlotManDate] = Now()
16     End If
17
18 End Sub
19
20 Private Sub Form_AfterUpdate()
21
22     If (Nz(Me![SlotManSign], 0) = 0) And (Nz(Me![SlotMan], 0) > 0) Then
23         MsgBox "Enter signature after update of slot width"
24     End If
25
26 End Sub
27
28 Private Sub SlotManDate_DbClick(Cancel As Integer)
29     Me![SlotManDate] = Now()
30 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmSlotResult"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboAssemblyID_AfterUpdate()
13 On Error GoTo Err_cboAssemblyID_AfterUpdate
14 '
15 ' Set AssemblyID in frmStart as this is being used by queries
16 '
17 Forms![Start]![txtMinAssemblyID] = Me![cboAssemblyID]
18 Forms![Start]![txtMaxAssemblyID] = Me![cboAssemblyID]
19 Forms![Start]![txtOperationNo] = Me![txtOperationNo]
20 '
21 ' Lock/unlock forms and controls in frmSlotResult
22 '
23 Forms![frmSlotResult]![frmSlotLarge].Locked = Me![txtOperationLocked]
24 Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Locked = Me![txtOperationLocked]
25 Forms![frmSlotResult]![frmSlotLarge]![cboSlotManSign].Enabled = Not
Me![txtOperationLocked]
26
27 Forms![frmSlotResult]![frmSlotFile].Locked = Me![txtOperationLocked]
28 Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Locked = Me![txtOperationLocked]
29 Forms![frmSlotResult]![frmSlotFile]![cboSlotSign].Enabled = Not Me![txtOperationLocked]
30
31 Exit_cboAssemblyID_AfterUpdate:
32 Exit Sub
33
34 Err_cboAssemblyID_AfterUpdate:
35 MsgBox Err.Description
36 Resume Exit_cboAssemblyID_AfterUpdate
37
38 End Sub
39
40 Private Sub cboAssemblyID_GotFocus()
41 cboAssemblyID.Requery
42 End Sub
43
44 Private Sub cboOrder_AfterUpdate()
45 Forms![Start]![txtLocationID] = Me![txtLocationID]
46
47 End Sub
48
49 Private Sub cboOrder_GotFocus()
```

```
50 ' Terje 16Nov12
51 ' Filtration of orders
52 '
53 ' SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location,
tblOrder.CompletedSign
54 ' FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
55 ' ORDER BY tblOrder.ProdOrder;
56
57 Dim Sql As String
58
59 Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign],
[tblLocation.LocationID] "
60 Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID =
tblLocation.LocationID "
61
62
63 If Me![optionOrderSelect] = 1 Then
64     Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
65 End If
66
67 Forms![frmSlotResult]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"
68
69 End Sub
70
71 Private Sub cboOrderLine_AfterUpdate()
72 '
73 ' If emty, set Order
74 '
75 On Error GoTo Err_cboOrderLine_AfterUpdate
76
77 If IsNull(Me![cboOrder]) Then
78     Me![cboOrder] = DLookup("[OrderID]", "tblOrderLine", "[OrderLineID] = " &
Me![cboOrderLine])
79 End If
80 '
81 ' Set parameters for slot data queries in Start form
82 '
83 Forms![Start]![txtOrderLineID] = Me![cboOrderLine]
84 Forms![Start]![txtLocationID] = Me![txtLocationID]
85
86 If DLookup("[InclAllLines]", "tblOrderLine", "[OrderLineID] = " & Me![cboOrderLine]) Then
87     Forms![Start]![txtInclAllLines] = 360
88 Else
89     Forms![Start]![txtInclAllLines] = 0
90 End If
91
92 Forms![Start]![txtMinAssemblyID] = 0
93 Forms![Start]![txtMaxAssemblyID] = 0
94 '
95 ' Requery Assembly and clear combo boxes
96 '
97 cboAssemblyID.Requery
98 Me![cboAssemblyID] = Null
99
100 Exit_cboOrderLine_AfterUpdate:
```

```
101 Exit Sub
102
103 Err_cboOrderLine_AfterUpdate:
104 MsgBox Err.Description
105 Resume Exit_cboOrderLine_AfterUpdate
106
107 End Sub
108
109 Private Sub cboOrderLine_GotFocus()
110 On Error GoTo Err_cboOrderLine_GotFocus
111 '
112 ' Set sql statement based on selected orders
113 '
114 Dim Sql As String
115
116 Sql = "SELECT [OrderLineID], [WorkOrder], [PartNo], [tblOrderLine.Description],
[CompletedSign] "
117 Sql = Sql & "FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID =
tblOrderLine.OrderID "
118 If Me![optionOrderSelect] = 1 Then
119     Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
120 Else
121     Sql = Sql & "WHERE [tblOrder.CompletedSign] >= 0 "
122 End If
123
124 If Not IsNull(Me![cboOrder]) Then
125     Sql = Sql & "AND [tblOrder.OrderID] = " & Me![cboOrder]
126 End If
127
128 Forms![frmSlotResult]![cboOrderLine].RowSource = Sql & " ORDER BY
tblOrderLine.WorkOrder;"
129
130 Exit_cboOrderLine_GotFocus:
131 Exit Sub
132
133 Err_cboOrderLine_GotFocus:
134 MsgBox Err.Description
135 Resume Exit_cboOrderLine_GotFocus
136
137 End Sub
138
139 Private Sub cmdImportSlotData_Click()
140 On Error GoTo Err_cmdImportSlotData_Click
141 ' Open frmSlotFileImport for selection of ResGauge data file
142 ' Record must not be locked
143
144 Dim stDocName As String
145 Dim stLinkCriteria As String
146
147 If Me![txtOperationLocked] Then
148     MsgBox "Form is locked. Go to Results and unlock."
149     Exit Sub
150 End If
151
152 stDocName = "frmSlotFileImport"
```

```
153 DoCmd.OpenForm stDocName, , , stLinkCriteria
154
155 Exit_cmdImportSlotData_Click:
156 Exit Sub
157
158 Err_cmdImportSlotData_Click:
159 MsgBox Err.Description
160 Resume Exit_cmdImportSlotData_Click
161
162 End Sub
163
164 Private Sub cmdImportSlotData2007_Click()
165 'Requires reference to Microsoft Office 12.0 Object Library.
166
167 Dim fDialog As Office.FileDialog
168 Dim varFile As Variant, strFile As String
169 Dim slash_pos As Integer, path_pos As Integer, I As Integer
170 Dim OrderLineID As Long
171
172
173 Set fDialog = Application.FileDialog(msoFileDialogFilePicker)
174 With fDialog
175 .AllowMultiSelect = False
176 .Title = "Please select a ResGauge log file"
177 .Filters.Clear
178 .Filters.Add "ResGauge data", "*.txt"
179 If .Show = True Then
180 For Each varFile In .SelectedItems
181 strFile = varFile
182 Next
183 Else
184 MsgBox "You clicked Cancel in the file dialog box. No data has been imported."
185 Exit Sub
186 End If
187 End With
188
189 If IsNull(varFile) Then
190 MsgBox "Error in slot file selection"
191 Exit Sub
192 Else
193 slash_pos = 1
194 Do While slash_pos > 0
195 path_pos = slash_pos
196 slash_pos = InStr(slash_pos + 1, strFile, "\")
197 Loop
198 If path_pos > 1 Then
199 Me![txtFile] = Right(strFile, Len(strFile) - path_pos)
200 Me![txtPath] = Left(strFile, path_pos)
201 End If
202 End If
203
204 OrderLineID = [Forms]![Start]![txtOrderLineID]
205 I = SlotAngle(strFile, OrderLineID, Me![cboAssemblyID])
206
```

207

208 End Sub

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmWareHouse"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboAssemblyID_AfterUpdate()
13
14 '
15 ' Set default values after assembly has been selected
16 '
17
18 On Error GoTo Err_cboAssemblyID_AfterUpdate
19
20 Dim stSql As String
21 '
22 ' Set AssemblyID in form Start as reference for Slot opening data queries
23 '
24
25 ' Forms![frmAssembly]![frmOrderLine].Form![chkAllinWO] = 0
26 Forms![Start]![txtMinAssemblyID] = Me![cboAssemblyID]
27 Forms![Start]![txtMaxAssemblyID] = Me![cboAssemblyID]
28
29 '
30 ' Requery list box to reflect selected assembly
31 ' Select record in frmWrapDetail form based on selected assembly
32 '
33
34 IstWrapID.Requery
35 Me![IstWrapID] = DMin("[WrapID]", "tblWrap", "[AssemblyID]=" & Me![cboAssemblyID])
36
37 IstWrapIDHardness.Requery
38 Me![IstWrapIDHardness] = Me![IstWrapID]
39
40 '
41 ' Select assemblies in report tab
42 '
43 Me![cboAssyFrom] = Me![cboAssemblyID]
44 Me![cboAssyTo] = Me![cboAssemblyID]
45 Me![fraSelectAssy] = 1
46
47
48
49
50 Exit_cboAssemblyID_AfterUpdate:
```

```
51 Exit Sub
52
53 Err_cboAssemblyID_AfterUpdate:
54 MsgBox Err.Description
55 Resume Exit_cboAssemblyID_AfterUpdate
56
57 End Sub
58
59 Private Sub cboAssemblyID_GotFocus()
60 'cboAssemblyID.Requery
61 End Sub
62
63 Private Sub cboOrder_AfterUpdate()
64 'cboOrderLine.Requery
65 End Sub
66
67 Private Sub cboOrder_GotFocus()
68 ' Terje 29Jan11
69 ' Filtration of orders
70 '
71 ' SELECT tblOrder.OrderID, tblOrder.ProdOrder, tblOrder.Description, tblLocation.Location,
tblOrder.CompletedSign
72 ' FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID = tblLocation.LocationID
73 ' ORDER BY tblOrder.ProdOrder;
74
75 Dim Sql As String
76
77 Sql = "SELECT [OrderID], [ProdOrder], [Description], [Location], [CompletedSign],
[tblLocation.LocationID] "
78 Sql = Sql & "FROM tblOrder LEFT JOIN tblLocation ON tblOrder.LocationID =
tblLocation.LocationID "
79
80
81 If Me![optionOrderSelect] = 1 Then
82 Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
83 End If
84
85 Forms![frmWareHouse]![cboOrder].RowSource = Sql & "ORDER BY tblOrder.ProdOrder;"
86
87
88 End Sub
89
90 Private Sub cboOrderLine_AfterUpdate()
91 '
92 ' If emty, set Order
93 '
94 On Error GoTo Err_cboOrderLine_AfterUpdate
95
96 If IsNull(Me![cboOrder]) Then
97 Me![cboOrder] = DLookup("[OrderID]", "tblOrderLine", "[OrderLineID] = " &
Me![cboOrderLine])
98 End If
99
100
101 IstOrderLineBOM.Requery
```

```
102
103
104 Exit_cboOrderLine_AfterUpdate:
105     Exit Sub
106
107 Err_cboOrderLine_AfterUpdate:
108     MsgBox Err.Description
109     Resume Exit_cboOrderLine_AfterUpdate
110
111 End Sub
112
113 Private Sub cboOrderLine_GotFocus()
114     ' SELECT tblOrderLine.OrderLineID, tblOrderLine.WorkOrder, tblOrderLine.PartNo,
115     '   tblOrderLine.Description, tblOrder.CompletedSign
116     ' FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID = tblOrderLine.OrderID
117     ' WHERE (((tblOrder.CompletedSign)=0));
118     On Error GoTo Err_cboOrderLine_GotFocus
119
120     Dim Sql As String
121
122     Sql = "SELECT [OrderLineID], [WorkOrder], [PartNo], [tblOrderLine.Description],
123     [CompletedSign] "
124     Sql = Sql & "FROM tblOrder INNER JOIN tblOrderLine ON tblOrder.OrderID =
125     tblOrderLine.OrderID "
126     If Me![optionOrderSelect] = 1 Then
127         ' noe muffens når denne linjen ikke inkluderes - WHERE mangler
128         Sql = Sql & "WHERE [tblOrder.CompletedSign] = 0 "
129     Else
130         Sql = Sql & "WHERE [tblOrder.CompletedSign] >= 0 "
131     End If
132
133     If Not IsNull(Me![cboOrder]) Then
134         Sql = Sql & "AND [tblOrder.OrderID] = " & Me![cboOrder]
135     End If
136
137     Forms![frmWareHouse]![cboOrderLine].RowSource = Sql & " ORDER BY
138     tblOrderLine.WorkOrder;"
139
140 Exit_cboOrderLine_GotFocus:
141     Exit Sub
142
143 Err_cboOrderLine_GotFocus:
144     MsgBox Err.Description
145     Resume Exit_cboOrderLine_GotFocus
146
147 End Sub
148
149
150 Private Sub cmdrepAllocated_Click()
151     DoCmd.OpenReport "rptAllocated", acViewPreview
152 End Sub
153
154 Private Sub IstOrderLineBOM_AfterUpdate()
```

```
153 '
154 ' Unhide text boxes for entering of wrap wire width and standard deviation when wrap wire
    is selected
155 ' hide text boxes wehn other part is selected
156 '
157 ' TM 20120106
158 '
159 IstOrderLineBOM.Requery
160
161 If DLookup("[Usage]", "tblOrderLineBOM", "[OrderLineBOMID]=" & Me![IstOrderLineBOM])
    = 4 Then
162     Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireWidt].Visible
        = True
163     Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireStDev].Visible
        = True
164     Else
165     Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![Traceability].SetFocus
166     Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireWidt].Visible
        = False
167     Forms![frmWareHouse]![frmAllocatedUB].Form![frmAllocated].Form![WrapWireStDev].Visible
        = False
168     End If
169
170 End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmWirePullTest"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub Force_AfterUpdate()
13     Me![TestMachine] = 1
14 End Sub
```

Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_frmWrapDetail"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11
12 Private Sub cboAxialWireID_AfterUpdate()
13     Me![cboWrapWireID].Requery
14 End Sub
15
16 Private Sub cboWrapSign_AfterUpdate()
17     '
18     ' Check if all normal sections has been wrapped and
19     ' copy date and sign from last normal section to (tblAssembly) tblAsBuiltOperation
20     '
21 On Error GoTo Err_cboWrapSign_AfterUpdate
22
23     Dim stSql As String, stSql1 As String, stSql2 As String
24     Dim Status As Boolean
25     Dim WireMass As Single
26     Dim AssemblyID As Long, OperationBOMID As Long
27     Dim dbs As Database
28     Dim rst As DAO.Recordset
29
30     If Me!AxialWireAsBuiltBOMID > 0 And Me!WrapWireAsBuiltBOMID > 0 Then
31         GoTo UpdateAsBuiltOperation
32     End If
33
34     stSql = "SELECT tblAsBuiltBOM.AsBuiltBOMID, tblAsBuiltBOM.AssemblyID, "
35     stSql = stSql & "tblAsBuiltBOM.OperationBOMID, tblAsBuiltBOM.AllocatedID, "
36     stSql = stSql & "tblAsBuiltBOM.Qty, tblAsBuiltBOM.Scrap, tblOrderLineBOM.Usage, "
37     stSql = stSql & "tblOperationBOM.HideOperationBOM, tblOrderLineBOM.HideOLBOM, "
38     stSql = stSql & "tblAsBuiltBOM.RecordedTime, tblAsBuiltBOM.StatusChangeTime "
39     stSql = stSql & "FROM tblOrderLineBOM INNER JOIN (tblAsBuiltBOM INNER JOIN "
40     stSql = stSql & "tblOperationBOM ON tblAsBuiltBOM.OperationBOMID = "
41     stSql = stSql & "tblOperationBOM.OperationBOMID) ON tblOrderLineBOM.OrderLineBOMID
42     stSql = stSql & "tblOperationBOM.OrderLineBOMID "
43     stSql = stSql & "WHERE (tblAsBuiltBOM.AssemblyID)= " & Me![AssemblyID]
44     stSql = stSql & " AND (tblOperationBOM.HideOperationBOM)=False "
45     stSql = stSql & "AND (tblOrderLineBOM.HideOLBOM)=False "
46     stSql1 = stSql & "AND (tblOrderLineBOM.Usage)=4 "
47     stSql2 = stSql & "AND (tblOrderLineBOM.Usage)=5 "
48
49     Set dbs = CurrentDb
```

```
50  Set rst = dbs.OpenRecordset(stSql1, dbOpenDynaset)
51
52  If rst.RecordCount = 0 Then
53      MsgBox "As built BOM has not been properly populated"
54      rst.Close
55      dbs.Close
56      Exit Sub
57  End If
58
59  WireMass = Round(Me!WrapWireMass * Me!NomScreenOD * Pi() * Me!WrapLength /
Me!WrapPitch / 1000, 1)
60
61  If Me!cboWrapTypeID = 1 Then
62      Status = False
63  Else
64      Status = True
65  End If
66
67  If Me!WrapWireAsBuiltBOMID > 0 Then
68      GoTo UpdateAxialWire
69  End If
70
71  rst.MoveFirst
72
73  If IsNull(rst!AllocatedID) Then
74      rst.Edit
75      rst!AllocatedID = Me!cboWrapWireID
76      rst!Scrap = Status
77      If Status Then
78          rst!StatusChangeTime = Now
79      End If
80      rst!Qty = WireMass
81      rst!RecordedTime = Now
82      rst.Update
83  Else
84      AssemblyID = Me!AssemblyID
85      OperationBOMID = rst!OperationBOMID
86      rst.AddNew
87      rst!AssemblyID = AssemblyID
88      rst!OperationBOMID = OperationBOMID
89      rst!AllocatedID = Me!cboWrapWireID
90      rst!Scrap = Status
91      If Status Then
92          rst!StatusChangeTime = Now
93      End If
94      rst!Qty = WireMass
95      rst!RecordedTime = Now
96      rst.Update
97  End If
98
99  Me!WrapWireAsBuiltBOMID = rst!AsBuiltBOMID
100
101  rst.Close
102
```

```
103 UpdateAxialWire:
104
105 Set rst = dbs.OpenRecordset(stSql2, dbOpenDynaset)
106
107 If rst.RecordCount = 0 Then
108     MsgBox "As built BOM has not been properly populated"
109     rst.Close
110     dbs.Close
111     Exit Sub
112 End If
113
114 WireMass = Round(Me!AxialWireMass * Me!nAxialRods * Me!WrapLength / 1000, 1)
115
116 rst.MoveFirst
117
118 If IsNull(rst!AllocatedID) Then
119     rst.Edit
120     rst!AllocatedID = Me!cboAxialWireID
121     rst!Scrap = Status
122     rst!Qty = WireMass
123     rst!RecordedTime = Now
124     rst.Update
125 Else
126     AssemblyID = Me!AssemblyID
127     OperationBOMID = rst!OperationBOMID
128     rst.AddNew
129     rst!AssemblyID = AssemblyID
130     rst!OperationBOMID = OperationBOMID
131     rst!AllocatedID = Me!cboAxialWireID
132     rst!Scrap = Status
133     rst!Qty = WireMass
134     rst!RecordedTime = Now
135     rst.Update
136 End If
137
138 Me!AxialWireAsBuiltBOMID = rst!AsBuiltBOMID
139
140 rst.Close
141 dbs.Close
142
143 UpdateAsBuiltOperation:
144
145 If Me![cboWrapTypeID] > 1 Then
146     Exit Sub
147 End If
148
149 stSql = "(tblWrap.WrapTypeID)=1 AND (tblWrap.WrapSign)>0 AND "
150 stSql = stSql & "(tblWrap.AssemblyID)=" & Me![AssemblyID]
151
152 If Me![NSections] = DCount("[WrapSign]", "tblWrap", stSql) Then
153     Me![AssyWrapSign] = Me![cboWrapSign]
154     Forms![frmAssembly]![frmAsBuiltOperationWrapping].Form![cboOperationSignWrap] =
Me![cboWrapSign]
155     Forms![frmAssembly]![frmAsBuiltOperationWrapping].Form![OperationDate] = Now
```

```
156     Me![AssyWrapDate] = Now
157 Else
158     Forms![frmAssembly]![frmAsBuiltOperationWrapping].Form![cboOperationSignWrap] =
Null
159     Forms![frmAssembly]![frmAsBuiltOperationWrapping].Form![OperationDate] = Null
160 End If
161
162 Exit_cboWrapSign_AfterUpdate:
163 Exit Sub
164
165 Err_cboWrapSign_AfterUpdate:
166 MsgBox Err.Description
167 Resume Exit_cboWrapSign_AfterUpdate
168
169 End Sub
170
171 Private Sub cboWrapSign_GotFocus()
172 '
173 ' Check that all data has been entered
174 '
175
176 If IsNull(Me![cboMachineID]) Then
177     Me![cboMachineID].SetFocus
178     MsgBox "Select wrapping machin"
179     Exit Sub
180 End If
181
182 If IsNull(Me![cboWrapWireID]) Then
183     Me![cboWrapWireID].SetFocus
184     MsgBox "Select wrap wire traceability"
185     Exit Sub
186 End If
187
188 If IsNull(Me![cboAxialWireID]) Then
189     Me![cboAxialWireID].SetFocus
190     MsgBox "Select axial wire traceability"
191     Exit Sub
192 End If
193
194 If IsNull(Me![WrapSection]) Then
195     Me![WrapSection].SetFocus
196     MsgBox "Enter section that has been wrapped"
197     Exit Sub
198 End If
199
200 If IsNull(Me![WrapStartPos]) Then
201     Me![WrapStartPos].SetFocus
202     MsgBox "Enter start position"
203     Exit Sub
204 End If
205
206 If IsNull(Me![WrapLength]) Then
207     Me![WrapLength].SetFocus
208     MsgBox "Enter wrapping length"
```

```
209      Exit Sub
210  End If
211
212  If IsNull(Me!AxialWireMass) Then
213      MsgBox "Axial wire mass not entered in Order form"
214      Me!WrapLength.SetFocus
215      Exit Sub
216  End If
217
218  If IsNull(Me!nAxialRods) Then
219      MsgBox "Number of axial rods not entered in Order form"
220      Me!WrapLength.SetFocus
221      Exit Sub
222  End If
223
224  If IsNull(Me!WrapWireMass) Then
225      MsgBox "Wrap wire mass not entered in Order form"
226      Me!WrapLength.SetFocus
227      Exit Sub
228  End If
229
230  If IsNull(Me!NomScreenOD) Then
231      MsgBox "Nominel Screen OD not entered in Order form"
232      Me!WrapLength.SetFocus
233      Exit Sub
234  End If
235
236  If IsNull(Me!WrapPitch) Then
237      MsgBox "Wrap pitch missing"
238      Me!WrapPitch.SetFocus
239      Exit Sub
240  End If
241
242  If IsNull(Me![WrapDate]) Then
243      Me![WrapDate] = Now
244  End If
245
246  End Sub
247
248  Private Sub cboWrapTypeID_AfterUpdate()
249  '
250  ' If status is changed, change status in tblAsBuiltBOM
251  ' IF A NORMAL SECTION IS SET TO SCRAP, SIGN IN TBLWRAP SHOULD BE REMOVED
252  ' THIS IS NOT IMPLEMENTED
253  '
254      Dim stSql As String
255      Dim Status As Boolean
256      ' Dim WireMass As Single
257      ' Dim AssemblyID As Long, OperationBOMID As Long
258      Dim dbs As Database
259      Dim rst As DAO.Recordset
260      Dim I As Integer
261
262
```

```
263     If IsNull(Me!WrapWireAsBuiltBOMID) Then
264         MsgBox "Record wrap wire traceability"
265         Me!cboWrapWireID.SetFocus
266         Exit Sub
267     End If
268
269     If IsNull(Me!AxialWireAsBuiltBOMID) Then
270         MsgBox "Record axial wire traceability"
271         Me!cboAxialWireID.SetFocus
272         Exit Sub
273     End If
274
275     stSql = "SELECT tblAsBuiltBOM.AsBuiltBOMID, tblAsBuiltBOM.AssemblyID, "
276     stSql = stSql & "tblAsBuiltBOM.Scrap, tblAsBuiltBOM.StatusChangeTime "
277     stSql = stSql & "FROM tblAsBuiltBOM "
278     stSql = stSql & "WHERE (tblAsBuiltBOM.AssemblyID)=" & Me!AssemblyID
279
280     If Me!cboWrapTypeID = 1 Then
281         Status = False
282     Else
283         Status = True
284     End If
285
286     Set dbs = CurrentDb
287     Set rst = dbs.OpenRecordset(stSql, dbOpenDynaset)
288
289     rst.MoveFirst
290     Do While Not rst.EOF
291         If rst!AsBuiltBOMID = Me!WrapWireAsBuiltBOMID Then
292             rst.Edit
293             rst!Scrap = Status
294             rst!StatusChangeTime = Now
295             rst.Update
296             Exit Do
297         End If
298         rst.MoveNext
299     Loop
300
301     rst.MoveFirst
302     Do While Not rst.EOF
303         If rst!AsBuiltBOMID = Me!AxialWireAsBuiltBOMID Then
304             rst.Edit
305             rst!Scrap = Status
306             rst!StatusChangeTime = Now
307             rst.Update
308             Exit Do
309         End If
310         rst.MoveNext
311     Loop
312
313     rst.Close
314     dbs.Close
315
316 End Sub
```



```
317
318 Private Sub cboWrapWireID_AfterUpdate()
319 '
320 ' Add traceability information to tblAsBuiltBOM
321 ' If the traceability is selected again, the record will not be updated
322 '
323 ' 14Dec2011 TM
324 ' Code deactivated as this is updated after signing the record
325 ' 11Feb2012 TM
326 '
327 Exit Sub
328 Dim stSql As String
329 Dim Status As Boolean
330 Dim WireMass As Single
331 Dim AssemblyID As Long, OperationBOMID As Long
332 Dim dbs As Database
333 Dim rst As DAO.Recordset
334
335 If Me!WrapWireAsBuiltBOMID > 0 Then
336 Exit Sub
337 End If
338
339 stSql = "SELECT tblAsBuiltBOM.AsBuiltBOMID, tblAsBuiltBOM.AssemblyID, "
340 stSql = stSql & "tblAsBuiltBOM.OperationBOMID, tblAsBuiltBOM.AllocatedID, "
341 stSql = stSql & "tblAsBuiltBOM.Qty, tblAsBuiltBOM.Scrap, tblOrderLineBOM.Usage, "
342 stSql = stSql & "tblOperationBOM.HideOperationBOM, tblOrderLineBOM.HideOLBOM, "
343 stSql = stSql & "tblAsBuiltBOM.RecordedTime "
344 stSql = stSql & "FROM tblOrderLineBOM INNER JOIN (tblAsBuiltBOM INNER JOIN "
345 stSql = stSql & "tblOperationBOM ON tblAsBuiltBOM.OperationBOMID = "
346 stSql = stSql & "tblOperationBOM.OperationBOMID) ON tblOrderLineBOM.OrderLineBOMID
= "
347 stSql = stSql & "tblOperationBOM.OrderLineBOMID "
348 stSql = stSql & "WHERE (tblAsBuiltBOM.AssemblyID)= " & Me![AssemblyID]
349 stSql = stSql & " AND (tblOrderLineBOM.Usage)=4 "
350 stSql = stSql & "AND (tblOperationBOM.HideOperationBOM)=False "
351 stSql = stSql & "AND (tblOrderLineBOM.HideOLBOM)=False"
352
353 Set dbs = CurrentDb
354 Set rst = dbs.OpenRecordset(stSql, dbOpenDynaset)
355
356
357 If rst.RecordCount = 0 Then
358 MsgBox "As built BOM has not been properly populated"
359 Exit Sub
360 End If
361
362 WireMass = Me!WrapWireMass * Me!NomScreenOD * Pi() * Me!WrapLength /
Me!WrapPitch / 1000
363
364 If Me!cboWrapTypeID = 1 Then
365 Status = False
366 Else
367 Status = True
368 End If
369
```

```
370  rst.MoveFirst
371
372  If IsNull(rst!AllocatedID) Then
373      rst.Edit
374      rst!AllocatedID = Me!cboWrapWireID
375      rst!Scrap = Status
376      rst!Qty = WireMass
377      rst!RecordedTime = Now
378      rst.Update
379  Else
380      AssemblyID = Me!AssemblyID
381      OperationBOMID = rst!OperationBOMID
382      rst.AddNew
383      rst!AssemblyID = AssemblyID
384      rst!OperationBOMID = OperationBOMID
385      rst!AllocatedID = Me!cboWrapWireID
386      rst!Scrap = Status
387      rst!Qty = WireMass
388      rst!RecordedTime = Now
389      rst.Update
390  End If
391
392  Me!WrapWireAsBuiltBOMID = rst!AsBuiltBOMID
393
394  rst.Close
395  dbs.Close
396
397
398  End Sub
399
400  Private Sub cboWrapWireID_GotFocus()
401  If IsNull(Me![cboAxialWireID]) Then
402      MsgBox "Select axial wire first"
403      Me![cboAxialWireID].SetFocus
404  End If
405  End Sub
```


Code

```
1  VERSION 1.0 CLASS
2  BEGIN
3      MultiUse = -1 'True
4  END
5  Attribute VB_Name = "Form_Start"
6  Attribute VB_GlobalNameSpace = False
7  Attribute VB_Creatable = True
8  Attribute VB_PredeclaredId = True
9  Attribute VB_Exposed = False
10 Option Compare Database
11 Option Explicit
12
13 Private Sub cmdBug_Click()
14 On Error GoTo Err_cmdBug_Click
15
16     Dim stDocName As String
17     Dim stLinkCriteria As String
18
19     stDocName = "frmBug"
20     DoCmd.OpenForm stDocName, , , stLinkCriteria
21     DoCmd.GoToRecord , , acNewRec
22
23 Exit_cmdBug_Click:
24     Exit Sub
25
26 Err_cmdBug_Click:
27     MsgBox Err.Description
28     Resume Exit_cmdBug_Click
29
30
31 End Sub
32
33 Private Sub cmdOperators_Click()
34     Dim Priv As Long
35
36     Priv = Nz(DLookup("[Privileges]", "tblOperator", "[LDapUser]=" & Me![LogOnName] & """),
37 0)
38     If Bitc(Priv, 0) Or Bitc(Priv, 9) Then
39         DoCmd.OpenForm "frmOperator", acNormal
40     End If
41 End Sub
42
43
44 Private Sub cmdQC_Click()
45 On Error GoTo Err_cmdQC_Click
46
47     Dim stDocName As String
48     Dim stLinkCriteria As String
49     Dim Priv As Long
```

```
50
51   Priv = Nz(DLookup("[Privileges]", "tblOperator", "[LDapUser]=" & Me![LogOnName] & ""),
0)
52   If Bitc(Priv, 0) Or Bitc(Priv, 9) Or Bitc(Priv, 11) Then
53       stDocName = "frmQC"
54       DoCmd.OpenForm stDocName, , , stLinkCriteria
55   End If
56
57 Exit_cmdQC_Click:
58   Exit Sub
59
60 Err_cmdQC_Click:
61   ' Write error message to tblBug
62   Forms![Start]![txtBug] = "cmdQC_Click: " & Err.Description
63   DoCmd.SetWarnings False
64   DoCmd.OpenQuery "qryappBug", acNormal, acEdit
65   DoCmd.SetWarnings True
66   '
67   MsgBox Err.Description
68   Resume Exit_cmdQC_Click
69
70 End Sub
71
72 Private Sub cmdSpooling_Click()
73 On Error GoTo Err_cmdSpooling_Click
74
75   Dim stDocName As String
76   Dim stLinkCriteria As String
77
78   stDocName = "frmSpooling"
79   DoCmd.OpenForm stDocName, , , stLinkCriteria
80
81 Exit_cmdSpooling_Click:
82   Exit Sub
83
84 Err_cmdSpooling_Click:
85   MsgBox Err.Description
86   Resume Exit_cmdSpooling_Click
87   End Sub
88
89 Sub cmdAssembly_Click()
90 On Error GoTo Err_cmdAssembly_Click
91
92   Dim stDocName As String
93   Dim stLinkCriteria As String
94
95   stDocName = "frmAssembly"
96   DoCmd.OpenForm stDocName, , , stLinkCriteria
97
98 Exit_cmdAssembly_Click:
99   Exit Sub
100
101 Err_cmdAssembly_Click:
102   MsgBox Err.Description
```

```
103 Resume Exit_cmdAssembly_Click
104
105 End Sub
106
107 Private Sub cmdOrder_Click()
108 On Error GoTo Err_cmdOrder_Click
109
110 Dim stDocName As String
111 Dim stLinkCriteria As String
112 Dim Priv As Long
113
114 Priv = Nz(DLookup("[Privileges]", "tblOperator", "[LDapUser]=" & Me![LogOnName] & """),
0)
115 If Bitc(Priv, 0) Or Bitc(Priv, 9) Or Bitc(Priv, 11) Then
116 stDocName = "frmOrder"
117 DoCmd.OpenForm stDocName, , , stLinkCriteria
118 End If
119
120 Exit_cmdOrder_Click:
121 Exit Sub
122
123 Err_cmdOrder_Click:
124 MsgBox Err.Description
125 Resume Exit_cmdOrder_Click
126
127 End Sub
128 Private Sub cmdSpool_Click()
129 End Sub
130
131 Private Sub cmdSlotData_Click()
132 On Error GoTo Err_cmdSlotData_Click
133
134 Dim stDocName As String
135 Dim stLinkCriteria As String
136
137 stDocName = "frmSlotResult"
138 DoCmd.OpenForm stDocName, , , stLinkCriteria
139
140 Exit_cmdSlotData_Click:
141 Exit Sub
142
143 Err_cmdSlotData_Click:
144 MsgBox Err.Description
145 Resume Exit_cmdSlotData_Click
146
147 End Sub
148
149 Private Sub cmdWareHouse_Click()
150 On Error GoTo Err_cmdWareHouse_Click
151
152 Dim stDocName As String
153 Dim stLinkCriteria As String
154 Dim Priv As Long
155
```

```
156     Priv = Nz(DLookup("[Privileges]", "tblOperator", "[LDapUser]='" & Me![LogOnName] & "'"),
157     0)
157     If Bitc(Priv, 0) Or Bitc(Priv, 9) Or Bitc(Priv, 11) Then
158         stDocName = "frmWareHouse"
159         DoCmd.OpenForm stDocName, , , stLinkCriteria
160     End If
161
162 Exit_cmdWareHouse_Click:
163     Exit Sub
164
165 Err_cmdWareHouse_Click:
166     MsgBox Err.Description
167     Resume Exit_cmdWareHouse_Click
168
169 End Sub
```