

Option Explicit

Private Sub Worksheet_Change(ByVal Target As Range)

On Error GoTo ErrorHandler

Application.EnableEvents = False

Dim answer As VbMsgBoxResult

Dim changeStatus As Boolean

changeStatus = False

' Check if the change occurred in filename cell

If Target.Address = "\$D\$7" Then

'Check if the address has been changed from the default rule book

If Target.Value <> oldRBfile Then

' Ask if the user would like to copy the previous custom rule book file name

answer = MsgBox("Are you sure want to change the rule book file name?", vbYesNo + vbQuestion, "Confirm Rule Book Change")

If answer = vbNo Then

Target.Value = oldRBfile

Else

oldRBfile = Target.Value

changeStatus = True

If Target.Value = defaultRuleFile Then

oldRBtype = "Master Rule Book"

Range("C7").Value = "Master Rule Book"

Else

oldRBtype = "Custom Rule Book"

Range("C7").Value = "Custom Rule Book"

End If

End If

End If

ElseIf Target.Address = "\$C\$7" Then

' Check if status was changed from Custom to Master

If Target.Value = "Master Rule Book" And Target.Value <> oldRBtype And Range("D7").Value <> defaultRuleFile Then

' Ask if the user would like to copy the previous custom rule book file name

answer = MsgBox("The rule book will be reset and your previous file name will be recorded.", vbYesNo + vbQuestion, "Reset Rule Book")

If answer = vbYes Then

Range("F7") = "Previous custom rule book: " & Range("D7").Value

changeStatus = True

' Store the new status value as the old value for the next change event

oldRBtype = Target.Value

' Reset the file name to default

Range("D7").Value = defaultRuleFile

End If

End If

End If

If changeStatus Then

' change the status of previously completed routines to FALSE due to the rule book change

' check if rawlines imported and also filtered based on rule book file

If Range("E18").Value = True And Range("E20").Value = True Then

Dim RBmacrostatus As Variant

Dim i As Integer

RBmacrostatus = Array("E18", "E20", "E22", "E24", "E26", "E28", "E30")

For i = LBound(RBmacrostatus) To UBound(RBmacrostatus)

If Range(RBmacrostatus(i)).Value = True Then

Range(RBmacrostatus(i)).Value = False

Range(RBmacrostatus(i)).Offset(0, 1).Value = Range(RBmacrostatus(i)).Offset(0, 1).Value & " *rule file changed!"

End If

Next i

End If

End If

```
Application.EnableEvents = True  
Exit Sub
```

ErrorHandler:

```
MsgBox "An error occurred: " & Err.Description  
Application.EnableEvents = True
```

```
End Sub
```

ThisWorkbook - 1

Option Explicit

Private Sub Workbook_Open()

On Error GoTo ErrorHandler

Application.EnableEvents = False

Dim thistemplate As Workbook

Dim homesheet As Worksheet

Dim RBfilename As Range

Dim RBfiletype As Range

Set thistemplate = ThisWorkbook

Set homesheet = thistemplate.Worksheets("Home")

Set RBfilename = homesheet.Range("D7")

Set RBfiletype = homesheet.Range("C7")

If RBfiletype.Value = "Master Rule Book" And RBfilename.Value <> defaultRuleFile Then

MsgBox "This template is set for the Master rule book but has a different filename value."

& vbCrLf & _

"The rule book file status has been changed to 'Custom', please check the filename and select 'Master' if you wish to reset"

RBfiletype.Value = "Custom Rule Book"

ElseIf RBfiletype.Value = "Master Rule Book" And RBfilename.Value = defaultRuleFile Then

MsgBox "Your template is set to utilize the Master rule book file: " & RBfilename.Value & vbCrLf &

Lf & _

"Ensure this file is open to complete the DDF construction macros on this sheet"

ElseIf RBfiletype.Value = "Custom Rule Book" Then

MsgBox "Your template is set to have a Custom rule book file: " & RBfilename.Value & vbCrLf &

_

"Ensure that your file name is correct and the matching rule book is open."

Else

' Invalid entry in ruleBookFile_Status

MsgBox "Invalid entry in rule book file status. Please select either 'Master' or 'Custom' Rule Book.", vbExclamation

End If

oldRBtype = RBfiletype 'set the old file type to check for future changes

oldRBfile = RBfilename 'set the old file name to check for future changes, allows for user to revert if changed

RBIsOpen

Application.EnableEvents = True

'homesheet.Range("D3").Value = Environ("UserName") & " at " & Format(Now, "yyyy/mm/dd hh:mm")

homesheet.Activate

Exit Sub

ErrorHandler:

MsgBox "An error occurred, please ensure the template hasn't been altered: " & Err.Description

Application.EnableEvents = True

End Sub

Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)

'Worksheets("Home").Range("D2").Value = Environ("UserName") & " at " & Format(Now, "yyyy/mm/dd hh:mm")

End Sub

TemplateModule - 1

Option Explicit

Public Const defaultRuleFile As String = "DDF_RuleBook_r0.xlsm" 'current standard rulebook filename

Public Const integrityBlue As Long = 9790002 'colour for placeholder fonts

Public oldRBtype As Variant 'Allows for check and update of rule book of status

Public oldRBfile As Variant 'Allows for check and revert of rule book file name

Public Function RBisOpen() As Boolean

'Allows other subs that call rule book to check open first

Dim rulebook As String

Dim rulebookStatus As Range

Dim wb As Workbook

'get rule book name and status cell

rulebook = Worksheets("Home").Range("D7").Value

Set rulebookStatus = Worksheets("Home").Range("E7")

RBisOpen = False

'check open books for match

For Each wb In Workbooks

 If rulebook = wb.Name Then

 RBisOpen = True

 End If

Next wb

'update status

rulebookStatus = RBisOpen

End Function

Sub clear_Home()

'this sub will clear the template status cells and the associated user and timestamp notes on the home sheet

Dim statusclear As Variant

Dim i As Integer

Dim placeholderRange As Range

Dim placeholderCell As Range

With Worksheets("Home")

 statusclear = Array("F6", "D8:E15", "E18:F18", "E20:F20", "E22:F22", "E24:F24", "E26:F26", "E28:F28", "E30:F30", "E32:F32")

 For i = LBound(statusclear) To UBound(statusclear)

 Range(statusclear(i)).ClearContents

 Next i

 Set placeholderRange = Range("D8:D16")

 For Each placeholderCell In placeholderRange

 placeholderCell.Value = placeholderCell.Offset(0, 3).Value

 placeholderCell.HorizontalAlignment = xlCenter

 placeholderCell.Font.FontStyle = "Italic"

 placeholderCell.Font.Size = 10

 placeholderCell.Font.Color = integrityBlue

 Next placeholderCell

End With

RBisOpen

End Sub

Sub call_check_EstInfo_Template_Files()

'check that rule book is open

If Not RBisOpen Then

 MsgBox "The Rule Book is not open"

 Exit Sub

End If

```

Dim homesheet As Worksheet
Set homesheet = Worksheets("Home")

'get this template name
Dim thisfilename As String
thisfilename = ThisWorkbook.Name

'get the rule book name
Dim rulebookfile As String
rulebookfile = homesheet.Range("D7").Value

'get the estimate info range
Dim estName_T As Range
Dim estInfoName_T As Range
Dim location_T As Range
Dim storeys_T As Range
Dim foundation_T As Range
Dim mechLevel_T As Range
Dim rawLinesFile_T As Range
Dim reportFile_T As Range
Dim checkStamp_T As Range

Set estName_T = homesheet.Range("D6")
Set estInfoName_T = homesheet.Range("D8")
Set location_T = homesheet.Range("D9")
Set storeys_T = homesheet.Range("D10")
Set foundation_T = homesheet.Range("D11")
Set mechLevel_T = homesheet.Range("D12")
Set rawLinesFile_T = homesheet.Range("D13")
Set reportFile_T = homesheet.Range("D14")
Set checkStamp_T = homesheet.Range("F6")

'call the check routine from the rule book
Application.Run rulebookfile & "!check_EstInfo_Template_Files", thisfilename, checkStamp_T, estName_T, estInfoName_T, location_T, storeys_T, foundation_T, mechLevel_T, rawLinesFile_T, reportFile_T
End Sub

Sub call_getEstimateData()

'check that rule book is open
If Not RBisOpen Then
    MsgBox "The Rule Book is not open"
    Exit Sub
End If

'get this template and associated rule book names
Dim thisfilename As String
thisfilename = ThisWorkbook.Name
Dim rulebookfile As String
rulebookfile = Worksheets("Home").Range("D7").Value

'get the estimate's data file name and status cell
Dim dataFile As Range
Set dataFile = Worksheets("Home").Range("D13")
Dim dataStatus As Range
Set dataStatus = Worksheets("Home").Range("E18")

Application.Run rulebookfile & "!getEstimateData", thisfilename, dataFile, dataStatus

End Sub

Sub call_RuleCheck()

'check that rule book is open
If Not RBisOpen Then
    MsgBox "The Rule Book is not open"
    Exit Sub
End If

'get this template and associated rule book names
Dim thisfilename As String

```

TemplateModule - 3

```
thisfilename = ThisWorkbook.Name
Dim rulebookfile As String
rulebookfile = Worksheets("Home").Range("D7").Value

'get the other required variables for the rule book sub:
'Sub RuleCheck(TemplateFile As String, rcheckStatus As Range, storeys As Integer, foundation As String, mechLevel As String)
Dim rcheckStatus_T As Range
Dim storeys_T As Integer
Dim foundation_T As String
Dim mechLevel_T As String

Set rcheckStatus_T = Worksheets("Home").Range("E20")
storeys_T = Worksheets("Home").Range("D10").Value
foundation_T = Worksheets("Home").Range("D11").Value
mechLevel_T = Worksheets("Home").Range("D12").Value

Application.Run rulebookfile & "!RuleCheck", thisfilename, rcheckStatus_T, storeys_T, foundation_T, mechLevel_T

End Sub

Sub call_getGroups()
'calls the rule book sub: getGroups(TemplateFile As String, layoutStatus As Range, estInfoName As String, GFAcell As Range, storeyCell As Range, bsmtPDcell As Range, foundationCell As Range)
'to compile the group and room data for final layout

'check that the rule check has been completed
Dim rulecheckStatus As Range
Set rulecheckStatus = Worksheets("Home").Range("E20")
If Not rulecheckStatus = True Then
    MsgBox "The rule check status is not TRUE, please complete before proceeding"
    Exit Sub
End If

'check that rule book is open
If Not RBisOpen Then
    MsgBox "The Rule Book is not open"
    Exit Sub
End If

'get this template and associated rule book names
Dim thisfilename As String
thisfilename = ThisWorkbook.Name
Dim rulebookfile As String
rulebookfile = Worksheets("Home").Range("D7").Value

'get the remaining required variables to pass
Dim estInfoName_T As String
Dim GFAcell_T As Range
Dim storeyCell_T As Range
Dim foundationCell_T As Range
Dim bsmtPDcell_T As Range
Dim layoutStatus_T As Range

estInfoName_T = Worksheets("Home").Range("D8").Value
Set GFAcell_T = Worksheets("Groups-Layout").Range("B7")
Set storeyCell_T = Worksheets("Groups-Layout").Range("B8")
Set bsmtPDcell_T = Worksheets("Groups-Layout").Range("B9")
Set foundationCell_T = Worksheets("Groups-Layout").Range("B10")
Set layoutStatus_T = Worksheets("Home").Range("E22")

Application.Run rulebookfile & "!getGroups", thisfilename, layoutStatus_T, estInfoName_T, GFAcell_T, storeyCell_T, bsmtPDcell_T, foundationCell_T

call_getDimensions

End Sub

Sub call_getDimensions()

'calls the rule book sub: getDimensions(TemplateFile As String, layoutStatus As Range, reportXLSX As String)
'to extract pricelist and room dimensions from the report pdf to excel export
```

```

'check that rule book is open
' If Not RBisOpen Then
'     MsgBox "The Rule Book is not open"
'     Exit Sub
' End If

'ensure the getGroups routine has been completed
Dim layoutStatus_T As Range
Set layoutStatus_T = Worksheets("Home").Range("E22")

If Not layoutStatus_T.Value = "1/4" Then
    MsgBox "The layout status is: " & layoutStatus_T & vbCrLf & _
        "Please complete the getGroups routine before proceeding"
    Exit Sub
End If

'get this template and associated rule book names
Dim thisfilename As String
thisfilename = ThisWorkbook.Name
Dim rulebookfile As String
rulebookfile = Worksheets("Home").Range("D7").Value

Dim reportXLSX_T As String
reportXLSX_T = Worksheets("Home").Range("D14").Value

Application.Run rulebookfile & "!getDimensions", thisfilename, layoutStatus_T, reportXLSX_T

call_makeGrpTable

End Sub

Sub call_makeGrpTable()
'calls the rule book sub: makeGrpTable(TemplateFile As String, layoutStatus As Range, estInfoName As String,
'footPrint As Double, pList As String)
'to create or update the group and layout table

'check that rule book is open
If Not RBisOpen Then
    MsgBox "The Rule Book is not open"
    Exit Sub
End If

'check that getDimensions routine has been completed
Dim layoutStatus_T As Range
Set layoutStatus_T = Worksheets("Home").Range("E22")

If Not (layoutStatus_T.Value = "2/4" Or layoutStatus_T.Value = "3/4" Or layoutStatus_T.Value = True) Then
    MsgBox "The layout status is: " & layoutStatus_T & vbCrLf & _
        "Please complete the getDimensions routine before proceeding"
    Exit Sub
End If

'get this template and associated rule book names
Dim thisfilename As String
thisfilename = ThisWorkbook.Name
Dim rulebookfile As String
rulebookfile = Worksheets("Home").Range("D7").Value

Dim estInfoName_T As String
estInfoName_T = Worksheets("Home").Range("D8").Value

Application.Run rulebookfile & "!makeGrpTable", thisfilename, layoutStatus_T, estInfoName_T, Range("GFAcell").Value, Range("PriceList").Value

End Sub

Sub verifyLayout()

Dim layoutStatus As Range

```

TemplateModule - 5

```
Set layoutStatus = Worksheets("Home").Range("E22")
```

```
'ensure the three required layout routines have been completed
```

```
If Not layoutStatus.Value = "3/4" Then
```

```
    MsgBox "The layout status is: " & layoutStatus & vbCrLf & _  
        "It cannot be verified. Please complete all required steps"
```

```
    Exit Sub
```

```
End If
```

```
layoutStatus.Value = True
```

```
layoutStatus.Offset(0, 1).Value = Environ("UserName") & " at " & Format(Now, "yyyy/mm/dd hh:mm")
```

```
Worksheets("Home").Activate
```

```
End Sub
```

```
Sub call_addGroupLineItems()
```

```
'calls the rule book sub: addGroupLineItems(TemplateFile As String, groupLineStatus As Range)
```

```
'to add the group data to the line items after review and revisions
```

```
    If Worksheets("Home").Range("E22").Value <> True Then
```

```
        MsgBox "The layout and groups have not been verified, please complete before proceeding"
```

```
        Exit Sub
```

```
    End If
```

```
    'check that rule book is open
```

```
    If Not RBisOpen Then
```

```
        MsgBox "The Rule Book is not open"
```

```
        Exit Sub
```

```
    End If
```

```
    'get this template and associated rule book names
```

```
    Dim thisfilename As String
```

```
    thisfilename = ThisWorkbook.Name
```

```
    Dim rulebookfile As String
```

```
    rulebookfile = Worksheets("Home").Range("D7").Value
```

```
    Dim groupLineStatus_T As Range
```

```
    Set groupLineStatus_T = Worksheets("Home").Range("E24")
```

```
    Application.Run rulebookfile & "!addGroupLineItems", thisfilename, groupLineStatus_T
```

```
End Sub
```

```
Sub call_PerformPercentageLookup()
```

```
'calls the rule book sub: PerformPercentageLookup(TemplateFile As String, replaceItemStatus As Range)
```

```
    'check the previous routines completed
```

```
    If Worksheets("Home").Range("E24").Value <> True Then
```

```
        MsgBox "The group information status is FALSE, please complete before proceeding"
```

```
        Exit Sub
```

```
    End If
```

```
    'check that rule book is open
```

```
    If Not RBisOpen Then
```

```
        MsgBox "The Rule Book is not open"
```

```
        Exit Sub
```

```
    End If
```

```
    'get this template and associated rule book names
```

```
    Dim thisfilename As String
```

```
    thisfilename = ThisWorkbook.Name
```

```
    Dim rulebookfile As String
```

```
    rulebookfile = Worksheets("Home").Range("D7").Value
```

```
    Dim statusCell As Range
```

```
    Set statusCell = Worksheets("Home").Range("E26")
```

```
    Dim lineData As Worksheet
```

```
    Set lineData = Worksheets("LineItems")
```

```
    If statusCell.Value = True Then
```

```
        ' Sub has been run, prompt the user to overwrite or exit
```



```

        Dim answer As Variant
        answer = MsgBox("The replacement rule lookup has already been performed. Do you want to overwr
ite the information?", vbYesNo + vbQuestion, "Sub Run Check")
        If answer = vbNo Then
            Exit Sub
        ElseIf answer = vbYes Then
            If lineData.Range("P1") <> "Depth-Replacement" Then
                answer = MsgBox("The formatting of LineItems is not as expected. If you continue, the
first row and all columns beyond O will be deleted. Do you wish to continue?", vbYesNo + vbQuestion, "
Sub Run Check")
                If answer = vbNo Then
                    statusCell = False
                    ' use offset instead    infoCell = "check previous steps completed"
                    Exit Sub
                End If
            Else
                lineData.Rows(1).Delete
                lineData.Columns("P:XFD").Delete
            End If
        End If
    End If
End If

```

```

Application.Run rulebookfile & "!PerformPercentageLookup", thisfilename, statusCell

```

```

End Sub

```

```

Sub call_UpdatePivotTable()

```

```

'calls the rule book sub: UpdatePivotTable(TemplateFile As String, ptUpStatus As Range)

```

```

'to update the pivot tables with the depth/damage value range

```

```

'check the previous routines completed

```

```

If Worksheets("Home").Range("E26").Value <> True Then

```

```

    MsgBox "The group information status is FALSE, please complete before proceeding"

```

```

    Exit Sub

```

```

End If

```

```

'check that rule book is open

```

```

If Not RBisOpen Then

```

```

    MsgBox "The Rule Book is not open"

```

```

    Exit Sub

```

```

End If

```

```

'get this template and associated rule book names

```

```

Dim thisfilename As String

```

```

thisfilename = ThisWorkbook.Name

```

```

Dim rulebookfile As String

```

```

rulebookfile = Worksheets("Home").Range("D7").Value

```

```

Dim ptUpStatus As Range

```

```

Set ptUpStatus = Worksheets("Home").Range("E28")

```

```

Application.Run rulebookfile & "!UpdatePivotTable", thisfilename, ptUpStatus

```

```

End Sub

```

```

Sub call_CopyPvtTables()

```

```

'calls the rule book sub: CopyPvtTables(TemplateFile As String, ptCopyStatus As Range)

```

```

'to create a copy of the pivot data on the DDF worksheets

```

```

'check the previous routines completed

```

```

If Worksheets("Home").Range("E28").Value <> True Then

```

```

    MsgBox "The group information status is FALSE, please complete before proceeding"

```

```

    Exit Sub

```

```

End If

```

```

'check that rule book is open

```

```

If Not RBisOpen Then

```

```

    MsgBox "The Rule Book is not open"

```

```

    Exit Sub

```

```

End If

```

```

'get this template and associated rule book names

```

```

Dim thisfilename As String
thisfilename = ThisWorkbook.Name
Dim rulebookfile As String
rulebookfile = Worksheets("Home").Range("D7").Value

Dim ptCopyStatus_T As Range
Set ptCopyStatus_T = Worksheets("Home").Range("E30")

Application.Run rulebookfile & "!CopyPvtTables", thisfilename, ptCopyStatus_T

copyResponseCosts

End Sub
Sub copyResponseCosts()

Dim ddfGrp2 As Worksheet

Set ddfGrp2 = Worksheets("DDFwrk_grp2")

ddfGrp2.Range("B17") = Range("respMain")
ddfGrp2.Range("B18") = Range("respBase")
ddfGrp2.Range("B19") = Range("respUpper")
ddfGrp2.Range("B19") = Range("respUpper")
ddfGrp2.Range("B7") = Range("GFAcell")

End Sub

Sub call_CanFloodDDF()
'calls the rule book sub: CanFloodDDF(TemplateFile As String, estInfoName As String, ddfStatus As Range)
'to create the CanFlood formatted function

'check the previous routines completed
If Worksheets("Home").Range("E30").Value <> True Then
    MsgBox "The pivot copy status is FALSE, please complete before proceeding"
    Exit Sub
End If

'check that rule book is open
If Not RBisOpen Then
    MsgBox "The Rule Book is not open"
    Exit Sub
End If

'get this template and associated rule book names
Dim thisfilename As String
thisfilename = ThisWorkbook.Name
Dim rulebookfile As String
rulebookfile = Worksheets("Home").Range("D7").Value

Dim estInfoSheet As String
estInfoSheet = Worksheets("Home").Range("D8").Value

Dim ddfStatus_T As Range
Set ddfStatus_T = Worksheets("Home").Range("E32")

Application.Run rulebookfile & "!CanFloodDDF", thisfilename, estInfoSheet, ddfStatus_T

End Sub

```