

Trabajo de Fin de Máster

Máster en Ingeniería en Tecnologías Industriales

Optimización del Sistema de Gestión del Cambio (CMS)

Anexos

Autor: Xavier Palacio Fernàndez
Director: Isidre Fabregues
Convocatoria: Julio 2018



Escuela Técnica Superior
De Ingeniería Industrial de Barcelona




```
'-----
'| Revision History :                               |
'|-----
'| Version No:| Author   | Changes Made:         | Mod. Date:  |
'|-----
'| 1.0   | X. Palacio | First Release           | 30/06/2017 |
'|-----
'|
'|-----
'| Design Hierarchy :
'|-----
'|
'| The complete design hierarchy is as shown below (left to right) :
'|
'| -> NO Class Creation
'|-----
'| DEVELOPER NOTES
'|-----
'|
'|-----
'| EXTERNAL REFERENCES
'|-----
```

```
Public Sub CMS_Global_Numbers()
    Dim QueryRow As Integer

    Dim State As String

    Dim NApproved As Integer
    Dim NPending As Integer
    Dim NPendingYear As Integer
    Dim NApprovedYear As Integer

    Dim ELECNAproved As Integer
    Dim ELECNPending As Integer
    Dim ELECNPendingYear As Integer
    Dim ELECNAprovedYear As Integer

    NApproved = 0
    NPending = 0
    NCancelled = 0
    NCompleted = 0
    NRejected = 0

    NApprovedYear = 0
    NPendingYear = 0
    NCancelledYear = 0
```

NCompletedYear = 0
NRejectedYear = 0

ELECNAproved = 0
ELECNPending = 0
ELECNCancelled = 0
ELECNCompleted = 0
ELECNRjected = 0

ELECNAprovedYear = 0
ELECNPendingYear = 0
ELECNCancelledYear = 0
ELECNCompletedYear = 0
ELECNRjectedYear = 0

QueryRow = 2
WriteRow = 3

Do

'Information of the CMS since year 2016

State = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 3)
CreationDate = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1)
Plant = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 8)
GlobalStatus = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 7)
CMSPM = UCase(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 9))
CMSNumber = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 2)

If GlobalStatus = "On going" Then
 If State = "Approved" Then
 NApproved = NApproved + 1
 Elseif State = "Pending" Then
 NPending = NPending + 1
 Elseif State = "Rejected" Then
 NRejected = NRejected + 1
 End If

Else

 If GlobalStatus = "Completed" Then
 NCompleted = NCompleted + 1
 Elseif GlobalStatus = "Cancelled" Then
 NCancelled = NCancelled + 1
 End If
End If

'Information of the CMS of the present year

If Year(CreationDate) = Year(Date) Or Year(CreationDate) = Year(Date) - 1 Then
 If GlobalStatus = "On going" Then
 If State = "Approved" Then
 NApprovedYear = NApprovedYear + 1
 Elseif State = "Pending" Then

```
        NPendingYear = NPendingYear + 1
    Elself State = "Rejected" Then
        NRejectedYear = NRejectedYear + 1
    End If
Else
    If GlobalStatus = "Completed" Then
        NCompletedYear = NCompletedYear + 1
    Elself GlobalStatus = "Cancelled" Then
        NCancelledYear = NCancelledYear + 1
    End If
End If
End If

'Information for the CMS of Electronics
If Plant = "V2 Ficosa Electronics VLDC - EMS" Then
    If GlobalStatus = "On going" Then
        If State = "Approved" Then
            ELECNAproved = ELECNAproved + 1
        Elself State = "Pending" Then
            ELECNPending = ELECNPending + 1
        Elself State = "Rejected" Then
            ELECNRjected = ELECNRjected + 1
        End If
    Elself GlobalStatus <> "On going" Then
        If GlobalStatus = "Completed" Then
            ELECNCpleted = ELECNCpleted + 1
        Elself GlobalStatus = "Cancelled" Then
            ELECNCancelled = ELECNCancelled + 1
        End If
    End If
'Information of the CMS of the present year
If Year(CreationDate) = Year(Date) Or Year(CreationDate) = (Year(Date) - 1) Then
    If GlobalStatus = "On going" Then
        If State = "Approved" Then
            ELECNAprovedYear = ELECNAprovedYear + 1
        Elself State = "Pending" Then
            ELECNPendingYear = ELECNPendingYear + 1
        Elself State = "Rejected" Then
            ELECNRjectedYear = ELECNRjectedYear + 1
        End If
    Elself GlobalStatus <> "On going" Then
        If GlobalStatus = "Cancelled" Then
            ELECNCancelledYear = ELECNCancelledYear + 1
        Elself GlobalStatus = "Completed" Then
            ELECNCpletedYear = ELECNCpletedYear + 1
        End If
    End If
End If
End If
```

"We are going to do the Sheet with the Completed CMS prepared for the Change Board

If Year(CreationDate) = Year(Date) Or Year(CreationDate) = Year(Date) - 1 Then
If GlobalStatus = "Completed" And Plant = "V2 Ficosa Electronics VLDC - EMS"

Then

ThisWorkbook.Sheets("CMS Completed").Cells(WriteRow, 1) = CMSNumber
ThisWorkbook.Sheets("CMS Completed").Cells(WriteRow, 2) = CMSPM
ThisWorkbook.Sheets("CMS Completed").Cells(WriteRow, 3) = CreationDate
ThisWorkbook.Sheets("CMS Completed").Cells(WriteRow, 4) = Plant
ThisWorkbook.Sheets("CMS Completed").Cells(WriteRow, 5) = GlobalStatus
ThisWorkbook.Sheets("CMS Completed").Cells(WriteRow, 6) = State
WriteRow = WriteRow + 1

End If

End If

QueryRow = QueryRow + 1

Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 4) = ""

ActualWeek = Format(Date, "ww")

LastWeek = ThisWorkbook.Sheets("CMS Identification").Cells(22, 3)

If ActualDate <> LastDate Then

"We copy all the values from ActualWeek to PreviousWeek2

ThisWorkbook.Sheets("CMS Identification").Cells(6, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(6, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(13, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(13, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(22, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(22, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(29, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(29, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(7, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(7, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(8, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(8, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(9, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(9, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(10, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(10, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(11, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(11, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(12, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(12, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(14, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(14, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(15, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(15, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(16, 3) =

ThisWorkbook.Sheets("CMS Identification").Cells(16, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(17, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(17, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(18, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(18, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(19, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(19, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(23, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(23, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(24, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(24, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(25, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(25, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(26, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(26, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(27, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(27, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(28, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(28, 2)

ThisWorkbook.Sheets("CMS Identification").Cells(30, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(30, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(31, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(31, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(32, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(32, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(33, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(33, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(34, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(34, 2)
ThisWorkbook.Sheets("CMS Identification").Cells(35, 3) =
ThisWorkbook.Sheets("CMS Identification").Cells(35, 2)

End If

ThisWorkbook.Sheets("CMS Identification").Cells(6, 2) = ActualWeek
ThisWorkbook.Sheets("CMS Identification").Cells(13, 2) = ActualWeek
ThisWorkbook.Sheets("CMS Identification").Cells(22, 2) = ActualWeek
ThisWorkbook.Sheets("CMS Identification").Cells(29, 2) = ActualWeek

ThisWorkbook.Sheets("CMS Identification").Cells(7, 2) = NApproved
ThisWorkbook.Sheets("CMS Identification").Cells(8, 2) = NPending
ThisWorkbook.Sheets("CMS Identification").Cells(9, 2) = NCancelled
ThisWorkbook.Sheets("CMS Identification").Cells(10, 2) = NCompleted
ThisWorkbook.Sheets("CMS Identification").Cells(11, 2) = NRejected
ThisWorkbook.Sheets("CMS Identification").Cells(12, 2) = NRejected + NCompleted +
NCancelled + NPending + NApproved

ThisWorkbook.Sheets("CMS Identification").Cells(14, 2) = NApprovedYear
ThisWorkbook.Sheets("CMS Identification").Cells(15, 2) = NPendingYear
ThisWorkbook.Sheets("CMS Identification").Cells(16, 2) = NCancelledYear

```
ThisWorkbook.Sheets("CMS Identification").Cells(17, 2) = NCompletedYear  
ThisWorkbook.Sheets("CMS Identification").Cells(18, 2) = NRejectedYear  
ThisWorkbook.Sheets("CMS Identification").Cells(19, 2) = NRejectedYear +  
NCompletedYear + NCancelledYear + NPendingYear + NApprovedYear
```

```
ThisWorkbook.Sheets("CMS Identification").Cells(23, 2) = ELECNAproved  
ThisWorkbook.Sheets("CMS Identification").Cells(24, 2) = ELECNPending  
ThisWorkbook.Sheets("CMS Identification").Cells(25, 2) = ELECNCancelled  
ThisWorkbook.Sheets("CMS Identification").Cells(26, 2) = ELECNCCompleted  
ThisWorkbook.Sheets("CMS Identification").Cells(27, 2) = ELECNRjected  
ThisWorkbook.Sheets("CMS Identification").Cells(28, 2) = ELECNRjected +  
ELECNCCompleted + ELECNCancelled + ELECNPending + ELECNAproved
```

```
ThisWorkbook.Sheets("CMS Identification").Cells(30, 2) = ELECNAprovedYear  
ThisWorkbook.Sheets("CMS Identification").Cells(31, 2) = ELECNPendingYear  
ThisWorkbook.Sheets("CMS Identification").Cells(32, 2) = ELECNCancelledYear  
ThisWorkbook.Sheets("CMS Identification").Cells(33, 2) = ELECNCCompletedYear  
ThisWorkbook.Sheets("CMS Identification").Cells(34, 2) = ELECNRjectedYear  
ThisWorkbook.Sheets("CMS Identification").Cells(35, 2) = ELECNRjectedYear +  
ELECNCCompletedYear + ELECNCancelledYear + ELECNPendingYear +  
ELECNAprovedYear
```

End Sub

```
Public Sub CMS_Aproved()  
    Dim QueryRow As Integer  
    Dim WriteRow As Integer
```

```
    Dim QueryPhaseDate As Date
```

```
    Dim Project As String  
    Dim State As String  
    Dim Plant As String  
    Dim InitialDate As Date  
    Dim Lotus_DB As LotusDB_Query_Type
```

```
    Dim EndDate As Date
```

```
    InitialDate = ThisWorkbook.Sheets("CMS Identification").Cells(1, 16)  
    EndDate = Date  
    Call LotusDataImport_2(CMS_v3, CMS, InitialDate, EndDate)
```

```
    QueryRow = 2  
    WriteRow = 6  
    Do  
        Project =  
Clean_Project_Code(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 4))
```



```

State = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 3)
GlobalStatus = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 7)
CreationDate = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1)
Plant = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 8)
Leader = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 10)
PM = UCase(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 9))

```

```

If State = "Approved" And Plant = "V2 Ficosa Electronics VLDC - EMS" And
CreationDate > InitialDate And GlobalStatus <> "Completed" Then
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 5) = Project
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 7) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 2)
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 8) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 10)

```

```

'Identify if there is PM
If PM = "" Or PM = " " Then
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 6) = "NO PM"
Else
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 6) = PM
End If

```

```

'Know if there is leader o there is no leader in the CMS
If IsEmpty(Leader) Then
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 8) = "NO
LEADER"
Else
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 8) = Leader
End If

```

```

'Write the date of the different phases
For I = 0 To 3
    QueryPhaseDate =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 11 + I)
    If QueryPhaseDate = "0:00:00" Then
        ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Interior.Color = RGB(174, 174, 174)
    Else
        'Due date not reached (Nothing)
        If QueryPhaseDate > Date And
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 30 + I) = 2 Then
            ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 + I) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 11 + I)
            'Due date + 15 days and it still open (RED stuffing)
            Elself QueryPhaseDate + 15 <= Date And
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 30 + I) = 2 Then
                ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Interior.ColorIndex = 3
                ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 + I) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 11 + I)
                'Less than 15 days after the due date (YELLOW stuffing)

```

```

Elseif QueryPhaseDate <= Date And QueryPhaseDate + 15 > Date And
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 30 + I) = 2 Then
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 + I) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 11 + I)
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Interior.ColorIndex = 6
    End If
    If ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 30 + I) = 3
Then
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 + I) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 11 + I)
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Font.ColorIndex = 4
    Elseif ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 30 + I) =
2 Then
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 + I) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 11 + I)
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Font.Color = 1
    Elseif ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 30 + I) =
1 Then
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 + I) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 11 + I)
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Font.ColorIndex = 3
    Elseif ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 30 + I) =
0 Then
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 + I) =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 11 + I)
    ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Font.ColorIndex = 1
    End If
    End If
Next I

```

'We want to identify if the document only have PH5

```

If ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 12) <> "" And
IsEmpty(ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9)) And
IsEmpty(ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 10)) And
IsEmpty(ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 11)) Then
    For I = 0 To 2
        ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 + I) = "-"
        ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Interior.ColorIndex = 2
        ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 9 +
I).Font.ColorIndex = 1
    Next I
End If

```

'Check if the PartNumber is affected

```

PartNumber = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 15)

```

```
        If IsEmpty(PartNumber) Then                'Or PartNumber = "xxx" Or PartNumber
= "x" Or PartNumber = "N/A" Or PartNumber = "NA" Then
            ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 14) = "NO"
            ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow,
14).Interior.ColorIndex = 3
        Else
            ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow, 14) = "YES"
            ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow,
14).Interior.ColorIndex = 4
        End If
```

```
        'Check if the table of Timming&Cost is empty.
        ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow,
13).Interior.ColorIndex = 4
```

```
        DeltaEBITPercentage =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 28)
        DeltaEBITEuros = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow,
29)
```

```
        If IsEmpty(DeltaEBITPercentage) Or IsEmpty(DeltaEBITEuros) Then
            ThisWorkbook.Sheets("CMS Identification").Cells(WriteRow,
13).Interior.ColorIndex = 3
        End If
```

```
        WriteRow = WriteRow + 1
        End If
        QueryRow = QueryRow + 1
        Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1) = ""
    End Sub
```

```
Public Sub CMS_PM()
    Dim QueryPM As Integer
    Dim WritePM As Integer
    Dim QueryInPM As Integer

    Dim ProgressPM As Double

    Dim IsIn As Boolean
    Dim PM As String

    Dim NumberApproved As Integer
    Dim Progress As Double

    Dim ActualWeek As Integer
```

```
    QueryPM = 6
    WritePM = 6
    Do
```

```

    PM = UCase(ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 6))
    IsIn = False
    QueryInPM = 6
    ProgressPM = ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM,
15).Value
    Do
        If PM = UCase(ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 17))
Then
            ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 18) =
ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 18) + 1
            ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 20) =
ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 20) + ProgressPM
            IsIn = True
            Exit Do
        End If
        QueryInPM = QueryInPM + 1
    Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 17) = ""

If IsIn = False Then
    ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 17) = PM
    ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 18) = 1
    ThisWorkbook.Sheets("CMS Identification").Cells(QueryInPM, 20) = ProgressPM
    WritePM = WritePM + 1
End If
QueryPM = QueryPM + 1
Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 6) = ""

'Loop to write the progress
WritePM = 6
TotalProgress = 0
Do
    NumberApproved = ThisWorkbook.Sheets("CMS Identification").Cells(WritePM, 18)
    If ThisWorkbook.Sheets("CMS Identification").Cells(WritePM, 20) = "-" Then
        Progress = 0
    Else
        Progress = ThisWorkbook.Sheets("CMS Identification").Cells(WritePM, 20)
    End If
    TotalProgress = TotalProgress + Progress

    If NumberApproved <> 0 Then
        ThisWorkbook.Sheets("CMS Identification").Cells(WritePM, 20) = Progress /
NumberApproved
    Else
        ThisWorkbook.Sheets("CMS Identification").Cells(WritePM, 20) = "-"
    End If

    If IsEmpty(ThisWorkbook.Sheets("CMS Identification").Cells(WritePM, 19)) Then
        ThisWorkbook.Sheets("CMS Identification").Cells(WritePM, 19) = 0
    End If

```

```
WritePM = WritePM + 1
Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(WritePM, 17) = ""

ActualWeek = Format(Date, "ww")
PreviousWeek = ThisWorkbook.Sheets("CMS Identification").Cells(5, 24)
If ActualWeek = PreviousWeek Then
    ThisWorkbook.Sheets("CMS Identification").Cells(6, 24) = TotalProgress / (QueryPM
- 6)
Else
    ThisWorkbook.Sheets("CMS Identification").Cells(6, 25) =
ThisWorkbook.Sheets("CMS Identification").Cells(6, 24)
    ThisWorkbook.Sheets("CMS Identification").Cells(6, 24) = TotalProgress / (QueryPM
- 6)
    ThisWorkbook.Sheets("CMS Identification").Cells(5, 25) =
ThisWorkbook.Sheets("CMS Identification").Cells(5, 24)
    ThisWorkbook.Sheets("CMS Identification").Cells(5, 24) = ActualWeek
End If
End Sub

Public Sub CMS_DELETE()
    MsgBox "Values Deleted"
    ThisWorkbook.Sheets("CMS Identification").Range("E6:O5000").ClearContents
    ThisWorkbook.Sheets("CMS Identification").Range("E6:O5000").Interior.ColorIndex = 2

    ThisWorkbook.Sheets("CMS Identification").Range("Q6:V5000").ClearContents

    ThisWorkbook.Sheets("CMS Pending").Range("C2:D5000").ClearContents
    ThisWorkbook.Sheets("CMS Pending").Range("I2:K5000").ClearContents
    ThisWorkbook.Sheets("CMS Pending").Range("L9:S5000").ClearContents
End Sub

Public Sub CMS_Progress()
    Dim QueryRow As Integer
    Dim Progress As Integer

    Dim Leader As String

    Dim ProgressPercen As Double

    QueryRow = 6
    Do
        Progress = 0
        Leader = UCase(ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow, 8))
        Ph2 = ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow, 9)
        Ph3 = ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow, 10)
        Ph4 = ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow, 11)
        Ph5 = ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow, 12)
        TC = ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow,
13).Interior.ColorIndex
        PN = ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow,
14).Interior.ColorIndex
```

```
'Add Progress if there is Leader
If Leader = "NO LEADER" Then
Else
    Progress = Progress + 100
End If
```

```
'Add Progress Related with the Target Dates
If IsEmpty(Ph5) Then
Else
    Progress = Progress + 100
End If
```

```
'Add Progress related with Timing&Cost
If TC = 4 Then
    Progress = Progress + 100
End If
```

```
'Add Progresss reltaed with the Part Number
If PN = 4 Then
    Progress = Progress + 100
End If
ProgressPercen = Progress / 400
ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow, 15) = ProgressPercen
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryRow, 5) = ""
End Sub
```

```
Public Sub Email_PM()
```

```
    Dim UserName As String
    Dim Maildb As Object
    Dim Maildoc As Object
    Dim Mailbody As Object
    Dim Mailsession As Object
```

```
    Dim Progress As Double
```

```
'Start session to Notes
Savelt = True
Set Mailsession = CreateObject("Notes.NotesSession")
'UserName = Mailsession.UserName
'Open the mail database in notes
Set Maildb = Mailsession.GetDatabase("MOLLET/FICOSA", "mail\cpmo.nsf")
If Maildb.IsOpen = True Then
Else
    Maildb.Openmail
End If
```

```
'Define the Calendar Week
IntWeekNumber = DatePart("ww", Date)
```

```
If IntWeekNumber < 10 Then
    StrCW = "CW0" & IntWeekNumber
Else
    StrCW = "CW" & IntWeekNumber
End If

QueryRow = 4
Do
    Position = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 1)
    Name = UCase(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3))
    Punctuation = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 19)
    CMSPending = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 4)
    CMSNumberN4 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 6)
    CMSNumberN3 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 5)
    DelayCHK1 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 7)
    DelayCHK2 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 12)
    DelayCHK3 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 16)

    IdentificationIN = False
    QueryIdentificationRow = 6
    Do
        PMIdentification = ThisWorkbook.Sheets("CMS
Identification").Cells(QueryIdentificationRow, 17)
        N4Name = ThisWorkbook.Sheets("CMS
Identification").Cells(QueryIdentificationRow, 21)
        N3Name = ThisWorkbook.Sheets("CMS
Identification").Cells(QueryIdentificationRow, 22)

        If PMIdentification = Name Then
            IdentificationIN = True
            Exit Do
        End If
        QueryIdentificationRow = QueryIdentificationRow + 1
    Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryIdentificationRow,
17) = ""

    If DelayCHK1 = 0 Or IsEmpty(DelayCHK1) Then
        DelayCHK1 = "Because there is no CMS with more than 30 days pending, the
delay is 0."
    Else
        DelayCHK1 = "The total delay of the CMS with more than 30 days is: " &
DelayCHK1 & " days."
    End If

    If IsEmpty(CMSNumberN4) Or CMSNumberN4 = 0 Then
        CMSNumberN4 = "At the moment you don't have any pending CMS with more
than 45 days from the day it was opened. (N4)"
    Else
```

CMSNumberN4 = "The number of pending CMS with more than 45 days from the day it was open is/are: " & CMSNumberN4 & "(" & N4Name & "). This CMS will be automatically cancelled on Monday."

End If

If IsEmpty(CMSNumberN3) Or CMSNumberN3 = 0 Then

CMSNumberN3 = "At the moment you don't have any pending CMS with more than 30 days and less than 45 days from the day it was opened. (N3)"

Else

CMSNumberN3 = "The number of pending CMS with more than 30 days from the day it was open is: " & CMSNumberN3 & "(" & N3Name & "). If you don't take any action, this CMS will be cancelled automatically after 45 days."

End If

QueryCHK2 = 3

Do

PMCHK2 = UCase(ThisWorkbook.Sheets("CMS CHK").Cells(QueryCHK2, 1))

CHK2NOKNumber = ThisWorkbook.Sheets("CMS CHK").Cells(QueryCHK2, 3)

CHK2NOKName = ThisWorkbook.Sheets("CMS CHK").Cells(QueryCHK2, 4)

If PMCHK2 = Name Then

If CHK2NOKNumber <> 0 Then

CHK2NOKNumber = "The number of CMS without Implementation Date is: " & CHK2NOKNumber & "(" & CHK2NOKName & ")"

Else

CHK2NOKNumber = "There isn't any CMS without Implementation Date."

End If

If IsEmpty(DelayCHK2) Or DelayCHK2 = 0 Then

DelayCHK2 = "Because there is no CMS with more than 7 days in CHK2, the delay is 0."

Else

DelayCHK2 = "The total delay of the CMS in CHK2 with more than 7 days is: " & DelayCHK2 & " days."

End If

Exit Do

End If

QueryCHK2 = QueryCHK2 + 1

Loop Until ThisWorkbook.Sheets("CMS CHK").Cells(QueryCHK2, 1) = ""

QueryCHK3 = 3

Do

PMCHK3 = UCase(ThisWorkbook.Sheets("CMS CHK").Cells(QueryCHK2, 10))

CHK3NOKNumber = ThisWorkbook.Sheets("CMS CHK").Cells(QueryCHK2, 12)

CHK3NOKName = ThisWorkbook.Sheets("CMS CHK").Cells(QueryCHK2, 13)

If PMCHK3 = Name Then

If CHK3NOKNumber <> 0 Then


```

        CHK3NOKNumber = "The number of CMS with delay is: " &
CHK3NOKNumber & " ( " & CHK3NOKName & " )"
        Else
            CHK3NOKNumber = "There isn't any CMS with delay."
        End If

        If IsEmpty(DelayCHK3) Or DelayCHK3 = 0 Then
            DelayCHK3 = "Because there is no CMS with more than 7 days in CHK3, the
delay is 0."
        Else
            DelayCHK3 = "The total delay of the CMS in CHK3 with more than 7 days is:
" & DelayCHK3 & " days."
        End If
        Exit Do
    End If
    QueryCHK3 = QueryCHK3 + 1
    Loop Until ThisWorkbook.Sheets("CMS CHK").Cells(QueryCHK3, 10) = ""

    If CHK2NOKNumber = 0 Or CHK2NOKNumber = 1 Then
        CHK2NOKNumber = "There isn't any CMS without Implementation Date."
    End If

    If IsEmpty(CHK3NOKNumber) Then
        CHK3NOKNumber = "There isn't any CMS with delay."
    End If

    If ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 8) = "-" Then
        Progress = 0
    Else
        Progress = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 8)
    End If

    Progress = Progress * 100
    Progress = Round(Progress, 2)

    If Name <> "NO PM" Then
        'Information of the Email
        Set Maildoc = Maildb.CreateDocument
        Maildoc.form = "Memo"
        Maildoc.sendto = Name
        Maildoc.CopyTo = "Enrique Osorio"
        Maildoc.Subject = "PM - CMS Flash Review -> Update - " & StrCW
        Maildoc.body = "Dear " + Name + "," & vbCrLf & vbCrLf & _
            "Please find below the status of the CMS at the end of " & StrCW & ":" & vbCrLf
& vbCrLf & _
            & " - The current position in the CMS Ranking is: " & Position & vbCrLf &
vbCrLf & _
            " - The current number of points by CMS in the CMS Ranking is: " &
Puntuacion & "." & vbCrLf & vbCrLf & _
            " - The current number of CMS pending is: " & CMSPending & vbCrLf &
vbCrLf & _

```

```
" CHECK 1 SUMMARY " & vbCrLf & vbCrLf & _  
" " & CMSNumberN4 & vbCrLf & vbCrLf & _  
" " & CMSNumberN3 & vbCrLf & vbCrLf & _  
" " & DelayCHK1 & vbCrLf & vbCrLf & _  
" CHECK 2 SUMMARY " & vbCrLf & vbCrLf & _  
" " & CHK2NOKNumber & vbCrLf & vbCrLf & _  
" " & DelayCHK2 & vbCrLf & vbCrLf & _  
" CHECK 3 SUMMARY " & vbCrLf & vbCrLf & _  
" " & CHK3NOKNumber & vbCrLf & vbCrLf & _  
" " & DelayCHK3 & vbCrLf & vbCrLf & _  
"" & vbCrLf & vbCrLf & _  
"Comments:" & vbCrLf & vbCrLf & _  
"Happy new year! " & vbCrLf & vbCrLf & _  
"The punctuation in the ranking has change. This year we will focus into the  
delay of each CMS in all the CHKS." & vbCrLf & vbCrLf & _  
"" & vbCrLf & vbCrLf & _  
"Thanks" & vbCrLf & vbCrLf & _  
" Corporate PMO " & StrTKOOverdueWOFforecast
```

```
Maildoc.SaveMessageOnSend = True  
Maildoc.send 0, Recipient  
MsgBox "Milestone email successfully sent to " + Name, vbOKOnly  
End If  
QueryRow = QueryRow + 1  
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""  
End Sub
```

CMS_Import_New_Products

```
Public Sub CMS_Import()
    Dim QueryProjectRow As Integer           'Query in the CMS Report NEW
    PRODUCTS Sheet
    Dim QueryRow As Integer                 'Query in the DB_AUX_SHEET Sheet

    Dim Project As String                   'Name of the project we are looking for
    Dim QueryProject As String              'Name of the project we are querying in the
    DATA Sheet

    Dim ECRStatus As String                  'Status of the project

    Dim NApproved As Integer
    Dim NPending As Integer
    Dim NRejected As Integer

    Dim NApprovedYear As Integer
    Dim NPendingYear As Integer
    Dim NRejectedYear As Integer

    Dim EndDate As Date
    Dim InitialDate As Date
```

```
InitialDate = ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(1, 2)
EndDate = Date
Call LotusDataImport_2(CMS_v3, CMS, InitialDate, EndDate)
ThisWorkbook.Sheets("CMS_CHK").Range("A3:H5000").ClearContents
ThisWorkbook.Sheets("CMS_CHK").Range("J3:Q5000").ClearContents
QueryRow = 2
QueryWriteRow = 3
Do
    CMSNumber = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 2)
    CMSDate = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1)
    CMSStatus = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 3)
    CMSGlobalStatus = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 7)
    CMSPlant = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 8)
    CMSPM = UCase(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 9))
    CMSApprovedDate = CDate(ApprovedDate(QueryRow))
    CMSPH5Date = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 14)
    CMSPH5StreetLight = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow,
```

33)

```
DelayApprovedDate = Date - CMSApprovedDate
```

```
If IsEmpty(CMSPH5Date) Then
    DelayPH5Date = 0
Else
    CMSPH5Date = CDate(CMSPH5Date)
    DelayPH5Date = Date - CMSPH5Date
End If
```

```

If CMSPlant = "V2 Ficosa Electronics VLDC - EMS" And CMSGlobalStatus = "On
going" And CMSStatus = "Approved" And Year(CMSDate) =
Year(ThisWorkbook.Sheets("CMS Identification").Cells(1, 16)) Then
    'Looking for the PM in the list of the sheet
    IsIn = False
    QueryPMRow = 3
    Do
        If CMSPM = UCase(ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow,
1)) Then
            IsIn = True
            ActualDate = Date
            If CMSPH5Date <> "" Then
                ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 2) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 2) + 1
                If CMSPH5Date >= ActualDate And CMSPH5StreetLight = 2 Then
                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 11) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 11) + 1
                    ElseIf CMSPH5Date < ActualDate And CMSPH5Date >= Format(Date - 7,
"yyyy-mm-dd") And CMSPH5StreetLight = 2 Then
                        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 12) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 12) + 1
                        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 13) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 13) + " " + CMSNumber
                        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 14) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 14) + 1
                        If IsEmpty(ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 14))
Then
                            ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 15) = 0
                            ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 16) = 0
                            End If
                            ElseIf CMSPH5Date < ActualDate - 7 And CMSPH5Date >= ActualDate -
15 And CMSPH5StreetLight = 2 Then
                                ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 12) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 12) + 1
                                ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 13) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 13) + " " + CMSNumber
                                ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 15) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 15) + 1
                                ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 17) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 17) + DelayPH5Date
                                If IsEmpty(ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 13))
Then
                                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 14) = 0
                                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 16) = 0
                                    End If
                                    ElseIf CMSPH5Date < ActualDate - 15 Then
                                        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 12) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 12) + 1
                                        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 13) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 13) + " " + CMSNumber

```

```

        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 16) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 16) + 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 17) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 17) + DelayPH5Date
        If IsEmpty(ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 12))
Then
            ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 14) = 0
            ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 15) = 0
        End If
    End If
Else
    ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 3) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 3) + 1
    ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 4) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 4) + " " + CMSNumber
    If CMSStatus = "Approved" Then
        If CMSApprovedDate >= ActualDate Then
            ElseIf CMSApprovedDate < ActualDate And CMSApprovedDate >=
ActualDate - 7 Then
                ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 5) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 5) + 1
                ElseIf CMSApprovedDate < ActualDate - 7 And CMSApprovedDate >=
ActualDate - 15 Then
                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 6) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 6) + 1
                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 8) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 8) + DelayApprovedDate
                    ElseIf CMSApprovedDate < ActualDate - 15 Then
                        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 7) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 7) + 1
                        ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 8) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 8) + DelayApprovedDate
                    End If
                End If
            End If
        End If
    Exit Do
End If
QueryPMRow = QueryPMRow + 1
Loop Until ThisWorkbook.Sheets("CMS CHK").Cells(QueryPMRow, 1) = ""

If IsIn = False And CMSPM <> " " Then
ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 1) = CMSPM
ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 10) = CMSPM
    If CMSPH5Date <> "" Then
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 2) = 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 3) = 0
        If CMSPH5Date >= ActualDate And CMSPH5StreetLight = 2 Then
            ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 11) = 1
            ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 12) = 0
            ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 13) = ""
            ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 14) = 0
        End If
    End If
End If

```

```

        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 15) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 16) = 0
    ElseIf CMSPH5Date < ActualDate And CMSPH5Date >= ActualDate - 7 And
CMSPH5StreetLight = 2 Then
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 11) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 12) = 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 13) =
CMSNumber
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 14) = 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 15) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 16) = 0
        ElseIf CMSPH5Date < ActualDate - 7 And CMSPH5Date >= ActualDate - 15
And CMSPH5StreetLight = 2 Then
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 11) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 12) = 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 13) =
CMSNumber
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 14) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 15) = 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 16) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 17) =
DelayPH5Date
        ElseIf CMSPH5Date < ActualDate - 15 And CMSPH5StreetLight = 2 Then
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 11) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 12) = 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 13) =
CMSNumber
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 14) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 15) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 16) = 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 17) =
DelayPH5Date
    End If
    Else
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 2) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 3) = 1
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 11) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 12) = 0
        ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 4) = CMSNumber
        If CMSStatus = "Approved" Then
            If CMSApprovedDate >= ActualDate Then
                ElseIf CMSApprovedDate < ActualDate And CMSApprovedDate >=
ActualDate - 7 Then
                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 5) = 1
                ElseIf CMSApprovedDate < ActualDate - 7 And CMSApprovedDate >=
ActualDate - 15 Then
                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 6) = 1
                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 8) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 8) + DelayApprovedDate
                ElseIf CMSApprovedDate < ActualDate - 15 Then
                    ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 7) = 1

```

```
                ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 8) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryWriteRow, 8) + DelayApprovedDate
            End If
        End If
    End If
    QueryWriteRow = QueryWriteRow + 1
End If
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1) = ""
```

'To know how many CMS doesn't have PH4 Date

```
QueryRow = 3
TotalNOK2 = 0
TotalN22 = 0
TotalN32 = 0
TotalN42 = 0
```

```
TotalNOK3 = 0
TotalN23 = 0
TotalN33 = 0
TotalN43 = 0
```

Do

```
TotalNOK2 = TotalNOK2 + ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 3)
TotalN22 = TotalN22 + ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 5)
TotalN32 = TotalN32 + ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 6)
TotalN42 = TotalN42 + ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 7)
```

```
TotalNOK3 = TotalNOK3 + ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 12)
TotalN23 = TotalN23 + ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 14)
TotalN33 = TotalN33 + ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 15)
TotalN43 = TotalN43 + ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 16)
```

```
    QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS CHK").Cells(QueryRow, 1) = ""
```

```
ThisWorkbook.Sheets("CMS CHK").Cells(2, 9) = TotalNOK2
ThisWorkbook.Sheets("CMS CHK").Cells(4, 9) = TotalN22
ThisWorkbook.Sheets("CMS CHK").Cells(6, 9) = TotalN32
ThisWorkbook.Sheets("CMS CHK").Cells(8, 9) = TotalN42
```

```
ThisWorkbook.Sheets("CMS CHK").Cells(2, 18) = TotalNOK3
ThisWorkbook.Sheets("CMS CHK").Cells(4, 18) = TotalN23
ThisWorkbook.Sheets("CMS CHK").Cells(6, 18) = TotalN33
ThisWorkbook.Sheets("CMS CHK").Cells(8, 18) = TotalN43
```

```
QueryProjectRow = 5
Do
```

```

Project = Clean_Project_Code(ThisWorkbook.Sheets("CMS Report NEW
PRODUCTS").Cells(QueryProjectRow, 5))
QueryRow = 2
NApproved = 0
NPending = 0
NRejected = 0
NCompleted = 0
NCancelled = 0

NApprovedYear = 0
NPendingYear = 0
NRejectedYear = 0
NCompletedYear = 0
NCancelledYear = 0

Do
  QueryProject =
Clean_Project_Code(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 4))
  ECRStatus = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 3)
  ECRGlobalStatus = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow,
7)
  CMSCreationDate = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow,
1)

'Calculating with all the information in CMS
If QueryProject = Project And ECRGlobalStatus = "On going" Then
  If InStr(1, ECRStatus, "Approved") Then
    NApproved = NApproved + 1
  ElseIf InStr(1, ECRStatus, "Pending") Then
    NPending = NPending + 1
  ElseIf InStr(1, ECRStatus, "Rejected") Then
    NRejected = NRejected + 1
  End If
  ElseIf QueryProject = Project And ECRGlobalStatus <> "On going" Then
  If InStr(1, ECRGlobalStatus, "Completed") Then
    NCompleted = NCompleted + 1
  ElseIf InStr(1, ECRGlobalStatus, "Cancelled") Then
    NCancelled = NCancelled + 1
  End If
End If

'Calculating with the information of the present year
If QueryProject = Project And ECRGlobalStatus = "On going" Then
  If Year(CMSCreationDate) = Year(Date) Or Year(CMSCreationDate) =
Year(Date) - 1 Then
    If InStr(1, ECRStatus, "Approved") Then
      NApprovedYear = NApprovedYear + 1
    ElseIf InStr(1, ECRStatus, "Pending") Then
      NPendingYear = NPendingYear + 1
    ElseIf InStr(1, ECRStatus, "Rejected") Then
      NRejectedYear = NRejectedYear + 1

```



```
        End If
    End If
    Elseif QueryProject = Project And ECRGlobalStatus <> "On going" Then
        If Year(CMSCreationDate) = Year(Date) Or Year(CMSCreationDate) =
Year(Date) - 1 Then
            If InStr(1, ECRGlobalStatus, "Completed") Then
                NCompletedYear = NCompletedYear + 1
            Elseif InStr(1, ECRGlobalStatus, "Cancelled") Then
                NCancelledYear = NCancelledYear + 1
            End If
        End If
    End If
End If

QueryRow = QueryRow + 1

Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 4) = ""

    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow, 6)
= NPending
    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow, 7)
= NApproved
    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow, 8)
= NRejected
    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow, 9)
= NCompleted
    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow,
10) = NCancelled

    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow,
11) = NPendingYear
    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow,
12) = NApprovedYear
    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow,
14) = NRejectedYear
    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow,
15) = NCompletedYear
    ThisWorkbook.Sheets("CMS Report NEW PRODUCTS").Cells(QueryProjectRow,
16) = NCancelledYear

    QueryProjectRow = QueryProjectRow + 1

Loop Until ThisWorkbook.Sheets("CMS Report NEW
PRODUCTS").Cells(QueryProjectRow, 5) = ""
End Sub

Function ApprovedDate(ByVal Consulta As Integer) As String
    Column = 36
    Approved = 0
    Do
        ConsultingDate = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(Consulta,
Column)
```

```
If ConsultingDate >= Approved Then
  Approved = ConsultingDate
End If
Column = Column + 1
Loop Until Column = 66
If Approved <> 0 Then
  ApprovedDate = Left(Approved, 10)
Else
  ApprovedDate = 0
End If
End Function
```



```
' | 1.0 | X. Palacio | First Release | 30/06/2017 |  
' | | | | |
```

```
'-----  
'  
'-----  
' Design Hierarchy :  
'-----  
'
```

```
' The complete design hierarchy is as shown below (left to right) :  
'
```

```
' -> NO Class Creation  
'-----
```

```
' DEVELOPER NOTES  
'-----  
'  
'-----
```

```
' EXTERNAL REFERENCES  
'-----
```

```
Public Sub CMS_Pending_Update()
```

```
    Dim QueryRow As Integer
```

```
    Dim Status As String
```

```
    Dim People As String
```

```
    Dim NPeople As Integer
```

```
    Dim InitialDate As Date
```

```
    Dim WritePeople As Integer
```

```
    Dim QueryPeople As Integer
```

```
    Dim EndDate As Date
```

```
    Dim IsIn As Boolean
```

```
    QueryRow = 2
```

```
    WritePeople = 2
```

```
    InitialDate = ThisWorkbook.Sheets("CMS Pending").Cells(1, 7)
```

```
    EndDate = Date
```

```
    Call LotusDataImport_2(CMS_v3, CMS_Pen, InitialDate, EndDate)
```

```
    Do
```

```
        Status = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 2)
```

```
        People = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 3)
```

```
        CMSDate = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1)
```

```
        Plant = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 7)
```

```
If Status = "Pending" And People <> "-" And People <> " -" And CMSDate >
InitialDate And InStr(Plant, "V2 Ficosa Electronics") Then
    PeopleArray = Split(People, "; ")
    NPeople = UBound(PeopleArray) + 1
    For I = 0 To NPeople - 1
        QueryPeople = 2
        IsIn = False
        Do
            If ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 3) =
PeopleArray(I) Then
                ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 4) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 4) + 1
                    IsIn = True
                    Exit Do
            End If
            QueryPeople = QueryPeople + 1
        Loop Until ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 3) = ""

        If IsIn = False Then
            ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 3) =
PeopleArray(I)
                ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 4) = 1
                WritePeople = WritePeople + 1
            End If
        Next I
    End If
    QueryRow = QueryRow + 1
    Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1) = ""
End Sub

Public Sub CMS_Pending_Update_PM()

    Dim QueryRow As Integer

    Dim Status As String
    Dim People As String
    Dim NPeople As Integer

    Dim InitialDate As Date

    Dim WritePeople As Integer
    Dim QueryPeople As Integer

    Dim IsIn As Boolean

    QueryRow = 2
    WritePeople = 2
    InitialDate = ThisWorkbook.Sheets("CMS Pending").Cells(1, 7)
    Do
        Status = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 2)
```

```
CMSDate = CDate(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1))
Plant = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 7)
PM = UCase(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 6))
Delay = Date - CMSDate
```

```
'Know is Empty the PM value
```

```
If PM = " " Then
    PM = "NO PM"
End If
```

```
If Status = "Pending" And InStr(Plant, "V2 Ficosa Electronics") Then
    If Year(CMSDate) = Year(Date) Or Year(CMSDate) = Year(Date) - 1 Then
        QueryPeople = 2
        IsIn = False
        Do
```

```
            If UCase(ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 9)) =
PM Then
                ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 10) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 10) + 1
                If Delay >= 30 Then
                    ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 11) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 11) + Delay
                End If
                IsIn = True
                Exit Do
            End If
            QueryPeople = QueryPeople + 1
        Loop Until ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 3) = ""
```

```
    If IsIn = False Then
        ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 9) = PM
        ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 10) = 1
        If Delay >= 30 Then
            ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 11) = Delay
        End If
        WritePeople = WritePeople + 1
    End If
End If
```

```
End If
```

```
QueryRow = QueryRow + 1
```

```
Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1) = ""
```

```
'Copy the values in CMS Identification
```

```
QueryRow = 2
```

```
Do
```

```
    Identificator = ThisWorkbook.Sheets("CMS Pending").Cells(QueryRow, 9)
```

```
    IsIn = False
```

```
    QueryRowIdentification = 5
```

```
Do
```

```

    QueryPM = UCase(ThisWorkbook.Sheets("CMS
Identification").Cells(QueryRowIdentification, 17))
    If QueryPM = Identificator Then
        ThisWorkbook.Sheets("CMS Identification").Cells(QueryRowIdentification, 19) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryRow, 10)
        ThisWorkbook.Sheets("CMS Identification").Cells(QueryRowIdentification, 23) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryRow, 11)
        IsIn = True
        Exit Do
    End If
    QueryRowIdentification = QueryRowIdentification + 1
    Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryRowIdentification,
17) = ""

```

```

    If IsIn = False Then
        ThisWorkbook.Sheets("CMS Identification").Cells(QueryRowIdentification, 17) =
Identificator
        ThisWorkbook.Sheets("CMS Identification").Cells(QueryRowIdentification, 18) = 0
        ThisWorkbook.Sheets("CMS Identification").Cells(QueryRowIdentification, 19) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryRow, 10)
    End If
    QueryRow = QueryRow + 1
    Loop Until ThisWorkbook.Sheets("CMS Pending").Cells(QueryRow, 9) = ""
End Sub

```

Public Sub CMS45()

'What we are going to do in this sub is to identify the CMS that are in phase N4 (more than 45 days pending)

```

    Dim Status As String
    Dim CMSDate As Date
    Dim Plant As String
    Dim PM As String
    Dim NumberCMS As String

```

```

    Dim QueryRow As Integer
    Dim WritePeople As Integer
    Dim InitialDate As Date

```

```

    QueryRow = 2
    WritePeople = 9
    InitialDate = ThisWorkbook.Sheets("CMS Pending").Cells(1, 7)

```

Do

```

    Status = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 2)
    CMSDate = CDate(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1))
    Plant = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 7)
    PM = UCase(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 6))
    NumberCMS = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 4)
    Delay = Date - CMSDate

```

'Know is Empty the PM value

If PM = " " Then

 PM = "NO PM"

End If

'Identification of CMS Pending N4

If Status = "Pending" And InStr(Plant, "V2 Ficosa Electronics") And CMSDate + 45
 <= Date Then

 If Year(CMSDate) = Year(Date) Or Year(CMSDate) = Year(Date) - 1 Then

 QueryPeople = 2

 IsIn = False

 Do

 If UCase(ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 12)) =

PM Then

 ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 13) =

ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 13) + 1

 ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 14) =

ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 14) & ", " & NumberCMS

 ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 20) =

ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 20) + Delay

 IsIn = True

 Exit Do

 End If

 QueryPeople = QueryPeople + 1

 Loop Until ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 3) = ""

 If IsIn = False Then

 ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 12) = PM

 ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 13) = 1

 ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 14) =

NumberCMS

 ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 20) = Delay

 WritePeople = WritePeople + 1

 End If

End If

End If

'Identification of CMS Pending N3

If Status = "Pending" And InStr(Plant, "V2 Ficosa Electronics") And CMSDate + 30
 <= Date And CMSDate + 45 >= Date Then

 If Year(CMSDate) = Year(Date) Or Year(CMSDate) = Year(Date) - 1 Then

 QueryPeople = 2

 IsIn = False

 Do

 If ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 12) = PM Then

 ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 17) =

ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 17) & ", " & NumberCMS

 ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 20) =

ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 20) + Delay

 IsIn = True

 Exit Do


```

        End If
        QueryPeople = QueryPeople + 1
    Loop Until ThisWorkbook.Sheets("CMS Pending").Cells(QueryPeople, 3) = ""

    If IsIn = False Then
        ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 12) = PM
        ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 17) =
NumberCMS
        ThisWorkbook.Sheets("CMS Pending").Cells(WritePeople, 20) = Delay
        WritePeople = WritePeople + 1
    End If
End If
End If
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 2) = ""

'COPY THE VALUES TO THE CMS PENDING SHEET IN THE ROW OF EACH PM
QueryIdentificationRow = 6
Do
    PM = UCase(ThisWorkbook.Sheets("CMS
Identification").Cells(QueryIdentificationRow, 17))
    QueryPendingRow = 9
    Do
        PMPending = UCase(ThisWorkbook.Sheets("CMS
Pending").Cells(QueryPendingRow, 12))
        If PMPending = PM Then
            ThisWorkbook.Sheets("CMS Identification").Cells(QueryIdentificationRow, 21) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryPendingRow, 14)
            ThisWorkbook.Sheets("CMS Identification").Cells(QueryIdentificationRow, 22) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryPendingRow, 17)
            ThisWorkbook.Sheets("CMS Identification").Cells(QueryIdentificationRow, 23) =
ThisWorkbook.Sheets("CMS Pending").Cells(QueryPendingRow, 20)
        End If
        QueryPendingRow = QueryPendingRow + 1
    Loop Until ThisWorkbook.Sheets("CMS Pending").Cells(QueryPendingRow, 12) = ""
    QueryIdentificationRow = QueryIdentificationRow + 1
    Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryIdentificationRow,
17) = ""
End Sub
Public Sub Graph()
    Dim QueryRow As Integer
    Dim N1 As Integer
    Dim N2 As Integer
    Dim N3 As Integer
    Dim N4 As Integer

    Dim DateN1 As Date
    Dim DateN2 As Date
    Dim DateN3 As Date
    Dim DateN4 As Date

```

Dim ActualWeek As Integer
Dim LastWeek As Integer

N1 = 0
N2 = 0
N3 = 0
N4 = 0

DateN1 = Date
DateN2 = Date - 15
DateN3 = Date - 30
DateN4 = Date - 45
ActualWeek = Format(Date, "ww")
LastWeek = ThisWorkbook.Sheets("CMS Pending").Cells(4, 17)

QueryRow = 2
Do

 QueryDate = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1)
 State = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 2)
 Plant = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 7)

 If State = "Pending" And Plant = "V2 Ficosa Electronics VLDC - EMS" Then
 If Year(QueryDate) = Year(Date) Or Year(QueryDate) = Year(Date) - 1 Then
 If QueryDate <= DateN1 And QueryDate >= DateN2 Then
 N1 = N1 + 1
 Elseif QueryDate < DateN2 And QueryDate >= DateN3 Then
 N2 = N2 + 1
 Elseif QueryDate < DateN3 And QueryDate >= DateN4 Then
 N3 = N3 + 1
 Elseif QueryDate < DateN4 Then
 N4 = N4 + 1
 End If
 End If

 End If
 QueryRow = QueryRow + 1

Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1) = ""

If LastWeek = ActualWeek Then

 ThisWorkbook.Sheets("CMS Pending").Cells(4, 13) = N1
 ThisWorkbook.Sheets("CMS Pending").Cells(4, 14) = N2
 ThisWorkbook.Sheets("CMS Pending").Cells(4, 15) = N3
 ThisWorkbook.Sheets("CMS Pending").Cells(4, 16) = N4
 ThisWorkbook.Sheets("CMS Pending").Cells(4, 17) = ActualWeek

Else

 ThisWorkbook.Sheets("CMS Pending").Cells(5, 13) = ThisWorkbook.Sheets("CMS Pending").Cells(4, 13)
 ThisWorkbook.Sheets("CMS Pending").Cells(5, 14) = ThisWorkbook.Sheets("CMS Pending").Cells(4, 14)
 ThisWorkbook.Sheets("CMS Pending").Cells(5, 15) = ThisWorkbook.Sheets("CMS Pending").Cells(4, 15)

```
ThisWorkbook.Sheets("CMS Pending").Cells(5, 16) = ThisWorkbook.Sheets("CMS Pending").Cells(4, 16)
```

```
ThisWorkbook.Sheets("CMS Pending").Cells(5, 17) = ThisWorkbook.Sheets("CMS Pending").Cells(4, 17)
```

```
ThisWorkbook.Sheets("CMS Pending").Cells(4, 13) = N1
```

```
ThisWorkbook.Sheets("CMS Pending").Cells(4, 14) = N2
```

```
ThisWorkbook.Sheets("CMS Pending").Cells(4, 15) = N3
```

```
ThisWorkbook.Sheets("CMS Pending").Cells(4, 16) = N4
```

```
ThisWorkbook.Sheets("CMS Pending").Cells(4, 17) = ActualWeek
```

```
End If
```

```
End Sub
```

```
Public Sub Email_Approvers()
```

```
Dim UserName As String
```

```
Dim Maildb As Object
```

```
Dim Maildoc As Object
```

```
Dim Mailbody As Object
```

```
Dim Mailsession As Object
```

```
Dim Progress As Double
```

```
'Start session to Notes
```

```
Savelt = True
```

```
Set Mailsession = CreateObject("Notes.NotesSession")
```

```
'UserName = Mailsession.UserName
```

```
'Open the mail database in notes
```

```
Set Maildb = Mailsession.GetDatabase("MOLLET/FICOSA", "mail\cpmo.nsf")
```

```
If Maildb.IsOpen = True Then
```

```
Else
```

```
Maildb.Openmail
```

```
End If
```

```
'Define the Calendar Week
```

```
IntWeekNumber = DatePart("ww", Date)
```

```
If IntWeekNumber < 10 Then
```

```
StrCW = "CW0" & IntWeekNumber
```

```
Else
```

```
StrCW = "CW" & IntWeekNumber
```

```
End If
```

```
QueryRow = 2
```

```
Do
```

```
Name = ThisWorkbook.Sheets("CMS Pending").Cells(QueryRow, 3)
```

```
NumberOpen = ThisWorkbook.Sheets("CMS Pending").Cells(QueryRow, 4)
```

```
If Name <> "NO PM" Then
```

```
Information of the Email
```

```
Set Maildoc = Maildb.CreateDocument
```

```
Maildoc.form = "Memo"
```

```
Maildoc.sendto = Name
```

```
Maildoc.CopyTo = "Xavier Palacio"
Maildoc.Subject = "Approvers - CMS Flash Review -> Update - " & StrCW
Maildoc.body = "Dear " + Name + "," & vbCrLf & vbCrLf & _
    "Please find below the status of your PENDING CMS at the end of " & StrCW &
"." & vbCrLf & vbCrLf _
    & "    - Number of PENDING CMS: " & NumberOpen & vbCrLf & vbCrLf & _
    "REMAINDER: You will find your PENDING CMS in CMS DB v3.0 (CMS ->
Pendings -> By Approver)" & vbCrLf & vbCrLf & _
    "Thanks" & vbCrLf & vbCrLf & _
    " Corporate PMO " & StrTKOOverdueWOFforecast

Maildoc.SaveMessageOnSend = True
Maildoc.send 0, Recipient
MsgBox "CMS email successfully sent to " + Name, vbOKOnly
End If
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Pending").Cells(QueryRow, 3) = ""
End Sub
```



```
' | | | | |
'-----
'
'-----
' Design Hierarchy :
'-----
'
' The complete design hierarchy is as shown below (left to right) :
'
' -> NO Class Creation
'
' -> FDS_Update() -----> Generic_Module, LotusNotesDB_Module
'
'-----
' DEVELOPER NOTES
'-----
'
'-----
' EXTERNAL REFERENCES
'-----
Private Declare Function SetCurrentDirectory Lib "kernel32" Alias "SetCurrentDirectoryA"
(ByVal lpPathName As String) As Long
'-----
' CONSTANT DEFINITION
'-----
Const FDS_row_SeekStart As Integer = 4 'FDS Worksheet row starting sweep, make
sure no data is before this point
Const FDS_row_SeekEnd As Integer = 100 'FDS Worksheet row finishing sweep,
make sure no data is beyond this point
Const FDS_MS_row_SeekStart As Integer = 2 'FDS_MS Worksheet row starting sweep,
make sure no data is before this point
Const FDS_MS_row_SeekEnd As Integer = 2000 'FDS_MS Worksheet row finishing
sweep, make sure no data is beyond this point
Const FDS_col_CalendarStart As Integer = 11 'FDS Worksheet column where schedule
starts
Const FDS_col_CalendarEnd As Integer = 62 'FDS Worksheet column where schedule
ends. Increase this number to increase schedule lenght
Const FDS_Project_rowinc As Integer = 2 'FDS Worksheet number of rows used for
every project
'-----
' GLOBAL VARIABLES DEFINITION
'-----
'-----
' PROCEDURES DEFINITION
'-----
Public Sub FDS_Update()
Dim I As Integer 'Auxiliar variables used for FOR loops
Dim J As Integer 'Auxiliar variables used for FOR loops
```

```

Dim K As Integer           'Auxiliar variables used for FOR loops
Dim MsgBox_Result As VbMsgBoxResult 'Auxiliar variable used to store MsgBox
Results

Dim FDS_CalendarStartDate As Date    ' Stores initial schedule date. Stored in cell
"G2"
Dim Now_Date As Date                ' Stores TODAY date
Dim Now_WK_Str As String             ' Stores TODAY year week
Dim FDS_Date As Date                ' Write schedule-calendar headers text
Dim Week_Str As String              ' Write schedule-calendar headers text
Dim Date_Str As String              ' Write schedule-calendar headers text
Dim Now_Col As Integer              ' Specify which column represents TODAY week
Dim FDS_Project_Size As Integer     ' Stores number of projects in FDS Worksheet
Dim FDS_MS_Project_Size As Integer  ' Stores number of project Milestones in
FDS_MS Worksheet

Dim FDS_Project_Code As String      ' Stores project code from FDS Worksheet
Dim FDS_Project_Date As Date        ' Stores date from FDS Worksheet
Dim FDS_Project_DateWK As Integer   ' Stores date (as wk) from FDS Worksheet
Dim FDS_Project_DateYY As Integer   ' Stores date (as year) from FDS Worksheet

Dim FDS_FC_Project_Date As Date     ' Stores forecasted MS closing date from
FDS_MS Worksheet
Dim FDS_FC_Project_DateWK As Integer ' Stores forecasted MS closing date (as
wk) from FDS_MS Worksheet
Dim FDS_FC_Project_DateYY As Integer ' Stores forecasted MS closing date (as
year) from FDS_MS Worksheet
Dim FDS_RE_Project_Date As Date     ' Stores real MS closing date from FDS_MS
Worksheet
Dim FDS_RE_Project_DateWK As Integer ' Stores real MS closing date (as wk) from
FDS_MS Worksheet
Dim FDS_RE_Project_DateYY As Integer ' Stores real MS closing date (as year) from
FDS_MS Worksheet
Dim FDS_RE_Project_Delay As Integer ' Stores calculated MS deviation for
Milestone (in days)
Dim FDS_MS_Name As String           ' Stores MS name from FDS_MS Worksheet
Dim FDS_MS_Symbol As String         ' Stores MS symbol (See "Generic_Module"
Sub "FDS_MS_Name_Process")

Dim FDS_RE_Project_Date_Closed As Boolean ' Used to identify when a MS is closed
(True) according FDS_MS Worksheet
Dim Exit_Project_MS_sweep As Boolean   ' Used to identify whe FDS_MS worksheet
sweep is finished

Dim FDS_DB As LotusDB_Type          ' Variable used to sweep all possible FDS
Databases

'Read Calendar Start date from Schedule Start Cell "FDS" Worksheet
FDS_CalendarStartDate = ThisWorkbook.Sheets("FDS").Cells(FDS_row_SeekStart - 2,
7).Value

```

```

'Store present/Today date
Now_Date = Now
Now_WK_Str = Date_to_WeekString(Now)

'Find how many projects do we have to create the timeline/schedule
FDS_Project_Size = 0
I = 0
For I = FDS_row_SeekStart To FDS_row_SeekEnd Step FDS_Project_rowinc
  If ThisWorkbook.Sheets("FDS").Cells(I, 7).Value <> "" Then
    FDS_Project_Size = FDS_Project_Size + 1
  Else
    I = FDS_row_SeekEnd
  End If
Next

'Add proper dates to timeline and format them properly
Sheets("FDS").Range(Cells(FDS_row_SeekStart, FDS_col_CalendarStart - 1),
Cells(FDS_row_SeekStart + FDS_Project_Size * FDS_Project_rowinc - 1,
FDS_col_CalendarStart - 1)).Select
  Sheets("FDS").Range(Cells(FDS_row_SeekStart, FDS_col_CalendarStart - 1),
Cells(FDS_row_SeekStart + FDS_Project_Size * FDS_Project_rowinc - 1,
FDS_col_CalendarStart - 1)).Copy

FDS_Date = FDS_CalendarStartDate
Now_Col = -1
For I = FDS_col_CalendarStart To FDS_col_CalendarEnd
  Week_Str = Date_to_WeekString(FDS_Date)
  Date_Str = Date_to_DateString(FDS_Date)
  ThisWorkbook.Sheets("FDS").Cells(FDS_row_SeekStart - 1, I).Value = Week_Str
  ThisWorkbook.Sheets("FDS").Cells(FDS_row_SeekStart - 2, I).Value = Date_Str
  FDS_Date = FDS_Date + 7

  Sheets("FDS").Range(Cells(FDS_row_SeekStart, I), Cells(FDS_row_SeekStart +
FDS_Project_Size * FDS_Project_rowinc - 1, I)).PasteSpecial Paste:=xlPasteFormats,
Operation:=xlNone, SkipBlanks:=False, Transpose:=False

  'If written week is present week, store column for future formatting
  If Now_WK_Str = Week_Str Then
    Now_Col = I
  End If
Next

'Clear all possible MS and red thick borders marking present date
ThisWorkbook.Sheets("FDS").Range(Cells(FDS_row_SeekStart,
FDS_col_CalendarStart), Cells(FDS_row_SeekStart + FDS_Project_Size *
FDS_Project_rowinc - 1, FDS_col_CalendarEnd)).Value = ""
Call
Format_Range_Internal_Borders(ThisWorkbook.Sheets("FDS").Range(Cells(FDS_row_S
eekStart - 2, FDS_col_CalendarStart), Cells(FDS_row_SeekStart + FDS_Project_Size *
FDS_Project_rowinc - 1, FDS_col_CalendarEnd)))
'Add Red thick border to present week

```



```

If Now_Col <> -1 Then
    Call
    Format_Range_RedThickBorder(ThisWorkbook.Sheets("FDS").Range(Cells(FDS_row_S
    eekStart - 2, Now_Col), Cells(FDS_row_SeekStart + FDS_Project_Size *
    FDS_Project_rowinc - 1, Now_Col)))
    End If

'Import data from FDS for Milestones and apply it
'-----
-----
For FDS_DB = LotusDB_Type.[_First] To LotusDB_Type.[_Last]
    'Check if user wants to open this FDS database
    MsgBox_Result = MsgBox("Do you want to import Milestones from " +
    LotusDB_Type_to_String(FDS_DB) + " DB?", vbYesNo, "Lotus DB Data import")

    If MsgBox_Result = vbYes Then

        'Import Milestone data from Lotus DB
        Call LotusDataImport_2(FDS_DB, FDS_Milestones, FDS_CalendarStartDate,
        FDS_Date)

        'Find how many projects MS do we have in FDS MS Worksheet
        FDS_MS_Project_Size = 0
        I = FDS_MS_row_SeekStart
        While ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(I, 1).Value <> ""
            FDS_MS_Project_Size = FDS_MS_Project_Size + 1
            I = I + 1
        Wend

        'Sweep all project codes in order to match it with "DB_AUX_SHEET" Worksheet
        I = FDS_row_SeekStart
        ThisWorkbook.Sheets("FDS").Activate
        While ThisWorkbook.Sheets("FDS").Cells(I, 7).Value <> ""
            FDS_Project_Code = ThisWorkbook.Sheets("FDS").Cells(I, 7).Value
            FDS_Project_Code = Strings.UCase(FDS_Project_Code)

            'When a project code is selected, sweep all calendar weeks in "FDS" Worksheet
            For J = FDS_col_CalendarStart To FDS_col_CalendarEnd

                'Read selected colum date in "FDS" Worksheet
                FDS_Project_Date =
                ThisWorkbook.Sheets("FDS").Cells(FDS_row_SeekStart - 2, J).Value
                FDS_Project_DateWK = CInt(Format(FDS_Project_Date, "ww", FW)) - 1
                FDS_Project_DateYY = Format(FDS_Project_Date, "yy", FW)

                'Start sweeping "DB_AUX_SHEET" Worksheet trying to match Project code
                first
                K = FDS_MS_row_SeekStart
                Exit_Project_MS_sweep = False

                While Not (Exit_Project_MS_sweep)

```

```

If Strings.UCase(ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(K,
1).Value) = FDS_Project_Code Then
    'When project code is matched, means a MS of the selected project is in
the line
    'Read Forecast date, Real Date and detect if MS is closed

    FDS_FC_Project_Date =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(K, 4).Value
    FDS_FC_Project_DateWK = CInt(Format(FDS_FC_Project_Date, "ww",
FW)) - 1
    FDS_FC_Project_DateYY = Format(FDS_FC_Project_Date, "yy", FW)
    If ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(K, 5).Value = "" Then
        FDS_RE_Project_Date_Closed = False ' MS is open, discard closing
dates data
        FDS_RE_Project_DateWK = -1
        FDS_RE_Project_DateYY = -1
        FDS_RE_Project_Delay = -1
    Else
        FDS_RE_Project_Date_Closed = True ' MS is closed, fill variables
with closing date
        FDS_RE_Project_Date =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(K, 5).Value
        FDS_RE_Project_DateWK = CInt(Format(FDS_RE_Project_Date,
"ww", FW)) - 1
        FDS_RE_Project_DateYY = Format(FDS_RE_Project_Date, "yy",
FW)
        FDS_RE_Project_Delay =
ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(K, 6).Value
    End If
    FDS_MS_Name = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(K,
3).Value
    Call FDS_MS_Name_Process(FDS_MS_Name, FDS_MS_Symbol,
FDS_RE_Project_Date_Closed)

    'We start comparing MS Forecast Date with selected date in "FDS"
Worksheet
    If FDS_FC_Project_DateWK = FDS_Project_DateWK And _
    FDS_FC_Project_DateYY = FDS_Project_DateYY Then
        'Forecast Date is equal to selected date column in "FDS" Worksheet
        If ThisWorkbook.Sheets("FDS").Cells(I, J).Value = "" Then
            ThisWorkbook.Sheets("FDS").Cells(I, J).Value = FDS_MS_Name
            ThisWorkbook.Sheets("FDS").Cells(I + 1, J).Value =
FDS_MS_Symbol
        Else
            ThisWorkbook.Sheets("FDS").Cells(I, J).Value =
CStr(ThisWorkbook.Sheets("FDS").Cells(I, J).Value) + "," + FDS_MS_Name
            ThisWorkbook.Sheets("FDS").Cells(I + 1, J).Value =
CStr(ThisWorkbook.Sheets("FDS").Cells(I + 1, J).Value) + FDS_MS_Symbol
        End If

        If FDS_RE_Project_Date_Closed Then

```

```

'MS has been closed
If FDS_RE_Project_Delay > 15 Then
    'Delay is outside expectation, write forecasted date in timeline in
RED
    Call
    Format_Text_Normal(ThisWorkbook.Sheets("FDS").Range(Cells(I, J), Cells(I, J)))
    Call
    Format_Last_Symbol_Red(ThisWorkbook.Sheets("FDS").Range(Cells(I + 1, J), Cells(I +
1, J)))
    Elseif FDS_RE_Project_Delay <= 15 And FDS_RE_Project_Delay
> 0 Then
        'Delay is inside expectation, write forecasted date in timeline in
Yellow
        Call
        Format_Text_Normal(ThisWorkbook.Sheets("FDS").Range(Cells(I, J), Cells(I, J)))
        Call
        Format_Last_Symbol_Yellow(ThisWorkbook.Sheets("FDS").Range(Cells(I + 1, J), Cells(I
+ 1, J)))
    Else
        'Delay is inside expectation, write forecasted date in timeline in
GREEN
        Call
        Format_Text_Normal(ThisWorkbook.Sheets("FDS").Range(Cells(I, J), Cells(I, J)))
        Call
        Format_Last_Symbol_Green(ThisWorkbook.Sheets("FDS").Range(Cells(I + 1, J), Cells(I
+ 1, J)))
    End If
Else
    FDS_RE_Project_Delay = Now - FDS_FC_Project_Date
    If FDS_RE_Project_Delay > 15 Then
        'MS has not been closed and delay is outside expectation, write
forecasted date in timeline in RED
        Call
        Format_Text_Normal(ThisWorkbook.Sheets("FDS").Range(Cells(I, J), Cells(I, J)))
        Call
        Format_Last_Symbol_Red(ThisWorkbook.Sheets("FDS").Range(Cells(I + 1, J), Cells(I +
1, J)))
        Elseif FDS_RE_Project_Delay <= 15 And FDS_RE_Project_Delay
> 0 Then
            'MS has not been closed and delay is inside expectation, write
forecasted date in timeline in Yellow
            Call
            Format_Text_Normal(ThisWorkbook.Sheets("FDS").Range(Cells(I, J), Cells(I, J)))
            Call
            Format_Last_Symbol_Yellow(ThisWorkbook.Sheets("FDS").Range(Cells(I + 1, J), Cells(I
+ 1, J)))
        Else
            'MS has not been closed and delay is 0 or less days, write
forecasted Date in timeline in Black
            Call
            Format_Text_Normal(ThisWorkbook.Sheets("FDS").Range(Cells(I, J), Cells(I, J)))

```

```
                Call
Format_Last_Symbol_Normal(ThisWorkbook.Sheets("FDS").Range(Cells(I + 1, J), Cells(I
+ 1, J)))
                End If
            End If
        End If
    Else
        If K > FDS_MS_Project_Size Then
            Exit_Project_MS_sweep = True
        End If
    End If
    K = K + 1
Wend
Next
I = I + FDS_Project_rowinc
Wend
End If
Next

'In WorkSheets, hide marked rows and columns
For I = 1 To 1024
    If ThisWorkbook.Sheets("FDS").Cells(1, I).Value = "HIDE" Then
        ThisWorkbook.Sheets("FDS").Cells(1, I).EntireColumn.Hidden = True
    End If
    If ThisWorkbook.Sheets("FDS").Cells(I, 1).Value = "HIDE" Then
        ThisWorkbook.Sheets("FDS").Cells(I, 1).EntireRow.Hidden = True
    End If
Next

'Show finalization message to user
MsgBox_Result = MsgBox("Automatic data loading process ended", vbInformation,
"FDS Milestone Report")

End Sub
```



```
'-----  
' Design Hierarchy :  
'-----  
'  
' The complete design hierarchy is as shown below (left to right) :  
'  
' -> NO Class Creation  
'  
' -> Format_Text_Red(ByRef CellsRange As Range)  
' -> Format_Text_Green(ByRef CellsRange As Range)  
' -> Format_Text_Normal(ByRef CellsRange As Range)  
' -> Format_Last_Symbol_Red(ByRef SymbolCell As Range)  
' -> Format_Last_Symbol_Yellow(ByRef SymbolCell As Range)  
' -> Format_Last_Symbol_Green(ByRef SymbolCell As Range)  
' -> Format_Last_Symbol_Normal(ByRef SymbolCell As Range)  
' -> Format_Range_Internal_Borders(ByRef CellsRange As Range)  
' -> Format_Range_Clear(ByRef CellsRange As Range)  
' -> Format_Range_RedThickBorder(ByRef CellsRange As Range)  
' -> Date_to_WeekString(ByRef CellsRange As Range) As String  
' -> Date_to_DateString(ByRef CellsRange As Range) As String  
' -> Clean_Project_Code(ByVal Project_Code As String) As String  
' -> Clean_PM_Name(ByVal PM_Name As String) As String  
' -> Clean_FDS_Phase(ByVal FDS_Phase As String) As String  
' -> Clean_BU_String(ByVal BU_String As String) As String  
' -> Clean_Region_String(ByVal Region_String As String) As String  
' -> Clean_PLANT_String(ByVal PLANT_String As String) As String  
' -> Clean_String(ByVal String_In As String) As String  
' -> FDS_MS_Name_Process(ByRef FDS_MS_Name As String, ByRef  
FDS_MS_Symbol As String, ByVal FDS_MS_Closed As Boolean)  
' -> SL_to_INT(ByVal SL_String As String) As Integer  
'
```

```
'-----  
' DEVELOPER NOTES  
'-----  
'  
'
```

```
'-----  
' CONSTANT DEFINITION  
'-----
```

```
Const Symbol_Circled0_Empty As Integer = 128 ' ASCII code for "Windings" font  
character. Used for Schedule MS representation  
Const Symbol_Circled1_Empty As Integer = 129 ' ASCII code for "Windings" font  
character. Used for Schedule MS representation  
Const Symbol_Circled2_Empty As Integer = 130 ' ASCII code for "Windings" font  
character. Used for Schedule MS representation  
Const Symbol_Circled3_Empty As Integer = 131 ' ASCII code for "Windings" font  
character. Used for Schedule MS representation  
Const Symbol_Circled4_Empty As Integer = 132 ' ASCII code for "Windings" font  
character. Used for Schedule MS representation  
Const Symbol_Circled5_Empty As Integer = 133 ' ASCII code for "Windings" font  
character. Used for Schedule MS representation
```

Const Symbol_Circled6_Empty As Integer = 134 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled7_Empty As Integer = 135 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled8_Empty As Integer = 136 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled9_Empty As Integer = 137 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled10_Empty As Integer = 138 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled0_Fill As Integer = 139 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled1_Fill As Integer = 140 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled2_Fill As Integer = 141 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled3_Fill As Integer = 142 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled4_Fill As Integer = 143 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled5_Fill As Integer = 144 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled6_Fill As Integer = 145 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled7_Fill As Integer = 146 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled8_Fill As Integer = 147 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled9_Fill As Integer = 148 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circled10_Fill As Integer = 149 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circle_Empty As Integer = 162 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Circle_Fill As Integer = 108 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Square_Empty As Integer = 112 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_Square_Fill As Integer = 110 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_4Star_Empty As Integer = 178 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_4Star_Fill As Integer = 170 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_5Star_Empty As Integer = 182 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_5Star_Fill As Integer = 171 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_5Star_Circled As Integer = 181 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_CheckOK_Empty As Integer = 254 ' ASCII code for "Windings" font character. Used for Schedule MS representation

Const Symbol_CheckNG_Fill As Integer = 253 ' ASCII code for "Windings" font character. Used for Schedule MS representation

```
'-----  
' GLOBAL VARIABLES DEFINITION  
'-----  
  
'-----  
' PROCEDURES DEFINITION  
'-----  
  
'     PROCEDURES DEFINITION - Format_Text_Red  
'     -PURPOSE: Formats selected cells range to red background and white text  
'     -PARAMETERS: CellsRange (ByRef) -> Range to be text formatted  
'     -RETURN VALUES: None  
'-----  
  
Public Sub Format_Text_Red(ByRef CellsRange As Range)  
    With CellsRange  
        .Interior.Pattern = xlSolid  
        .Interior.PatternColorIndex = xlAutomatic  
        .Interior.Color = 255  
        .Interior.TintAndShade = 0  
        .Interior.PatternTintAndShade = 0  
        .Font.ThemeColor = xlThemeColorDark1  
        .Font.TintAndShade = 0  
        .Font.ThemeFont = xlThemeFontNone  
        .Font.Name = "Arial"  
        .Font.Size = 8  
        .Font.Bold = True  
        .Font.Strikethrough = False  
        .Font.Superscript = False  
        .Font.Subscript = False  
        .Font.OutlineFont = False  
        .Font.Shadow = False  
        .Font.Underline = xlUnderlineStyleNone  
    End With  
End Sub  
  
'-----  
'     PROCEDURES DEFINITION - Format_Text_Green  
'     -PURPOSE: Formats selected cells range to green text  
'     -PARAMETERS: CellsRange (ByRef) -> Range to be text formatted  
'     -RETURN VALUES: None  
'-----  
  
Public Sub Format_Text_Green(ByRef CellsRange As Range)  
    'Formatear la fuente en verde  
    With CellsRange  
        .Font.Color = -16711936  
        .Font.TintAndShade = 0  
        .Font.ThemeFont = xlThemeFontNone  
    End With  
End Sub
```



```

        .Font.Name = "Arial"
        .Font.Size = 8
        .Font.Bold = True
        .Font.Strikethrough = False
        .Font.Superscript = False
        .Font.Subscript = False
        .Font.OutlineFont = False
        .Font.Shadow = False
        .Font.Underline = xlUnderlineStyleNone
    End With
End Sub
'-----
'      PROCEDURES DEFINITION - Format_Text_Normal
'      -PURPOSE: Formats selected cells range to black text
'      -PARAMETERS: CellsRange (ByRef) -> Range to be text formatted
'      -RETURN VALUES: None
'-----
Public Sub Format_Text_Normal(ByRef CellsRange As Range)
    'Formatear la fuente en verde
    With CellsRange
        .Font.Color = 0
        .Font.TintAndShade = 0
        .Font.ThemeFont = xlThemeFontNone
        .Font.Name = "Arial"
        .Font.Size = 8
        .Font.Bold = False
        .Font.Strikethrough = False
        .Font.Superscript = False
        .Font.Subscript = False
        .Font.OutlineFont = False
        .Font.Shadow = False
        .Font.Underline = xlUnderlineStyleNone
    End With
End Sub
'-----
'      PROCEDURES DEFINITION - Format_Last_Symbol_Red
'      -PURPOSE: Formats last introduced character in a cell to be Red
'      -PARAMETERS: SymbolCell (ByRef) -> Range to be text formatted
'      -RETURN VALUES: None
'-----
Public Sub Format_Last_Symbol_Red(ByRef SymbolCell As Range)
    'Format last symbol in RED
    With SymbolCell.Font
        .Name = "Wdings"
        .Size = 14
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
    End With
End Sub

```

```
.TintAndShade = 0
.ThemeFont = xlThemeFontNone
End With
With SymbolCell.Characters(Start:=Strings.Len(SymbolCell.Value), Length:=1).Font
.Color = 255
End With
End Sub
```

```
-----
'      PROCEDURES DEFINITION - Format_Last_Symbol_Yellow
'      -PURPOSE: Formats last introduced character in a cell to be Yellow
'      -PARAMETERS: SymbolCell (ByRef) -> Range to be text formatted
'      -RETURN VALUES: None
'-----
```

```
Public Sub Format_Last_Symbol_Yellow(ByRef SymbolCell As Range)
'Format last symbol in Yellow
With SymbolCell.Font
.Name = "Windings"
.Size = 14
.Strikethrough = False
.Superscript = False
.Subscript = False
.OutlineFont = False
.Shadow = False
.Underline = xlUnderlineStyleNone
.TintAndShade = 0
.ThemeFont = xlThemeFontNone
End With
With SymbolCell.Characters(Start:=Strings.Len(SymbolCell.Value), Length:=1).Font
.Color = -16727809
End With
End Sub
```

```
-----
'      PROCEDURES DEFINITION - Format_Last_Symbol_Green
'      -PURPOSE: Formats last introduced character in a cell to be Green
'      -PARAMETERS: SymbolCell (ByRef) -> Range to be text formatted
'      -RETURN VALUES: None
'-----
```

```
Public Sub Format_Last_Symbol_Green(ByRef SymbolCell As Range)
'Format last symbol in Green
With SymbolCell.Font
.Name = "Windings"
.Size = 14
.Strikethrough = False
.Superscript = False
.Subscript = False
.OutlineFont = False
.Shadow = False
.Underline = xlUnderlineStyleNone
.TintAndShade = 0
.ThemeFont = xlThemeFontNone
End With
```

```
With SymbolCell.Characters(Start:=Strings.Len(SymbolCell.Value), Length:=1).Font
    .Color = -16711936
End With
End Sub
'-----
'      PROCEDURES DEFINITION - Format_Last_Symbol_Normal
'      -PURPOSE: Formats last introduced character in a cell to be Black
'      -PARAMETERS: SymbolCell (ByRef) -> Range to be text formatted
'      -RETURN VALUES: None
'-----
Public Sub Format_Last_Symbol_Normal(ByRef SymbolCell As Range)
    'Format last symbol in Black
    With SymbolCell.Font
        .Name = "Windings"
        .Size = 14
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .TintAndShade = 0
        .ThemeFont = xlThemeFontNone
    End With
    With SymbolCell.Characters(Start:=Strings.Len(SymbolCell.Value), Length:=1).Font
        .Color = 0
    End With
End Sub
'-----
'      PROCEDURES DEFINITION - Format_Range_Internal_Borders
'      -PURPOSE: Formats introduced range to have internal borders
'      -PARAMETERS: CellsRange (ByRef) -> Range to be formatted
'      -RETURN VALUES: None
'-----
Public Sub Format_Range_Internal_Borders(ByRef CellsRange As Range)
    With CellsRange
        .Borders(xlDiagonalDown).LineStyle = xlNone
        .Borders(xlDiagonalUp).LineStyle = xlNone
        .Borders(xlEdgeLeft).LineStyle = xlContinuous
        .Borders(xlEdgeLeft).ColorIndex = 0
        .Borders(xlEdgeLeft).TintAndShade = 0
        .Borders(xlEdgeLeft).Weight = xlThin
        .Borders(xlEdgeTop).LineStyle = xlContinuous
        .Borders(xlEdgeTop).ColorIndex = 0
        .Borders(xlEdgeTop).TintAndShade = 0
        .Borders(xlEdgeTop).Weight = xlThin
        .Borders(xlEdgeBottom).LineStyle = xlContinuous
        .Borders(xlEdgeBottom).ColorIndex = 0
        .Borders(xlEdgeBottom).TintAndShade = 0
        .Borders(xlEdgeBottom).Weight = xlThin
        .Borders(xlEdgeRight).LineStyle = xlContinuous
    End With
End Sub
```

```
.Borders(xlEdgeRight).ColorIndex = 0
.Borders(xlEdgeRight).TintAndShade = 0
.Borders(xlEdgeRight).Weight = xlThin
.Borders(xlEdgeLeft).LineStyle = xlContinuous
.Borders(xlEdgeLeft).ColorIndex = 0
.Borders(xlEdgeLeft).TintAndShade = 0
.Borders(xlEdgeLeft).Weight = xlThin
.Borders(xlInsideVertical).LineStyle = xlContinuous
.Borders(xlInsideVertical).ColorIndex = 0
.Borders(xlInsideVertical).TintAndShade = 0
.Borders(xlInsideVertical).Weight = xlThin
.Borders(xlInsideHorizontal).LineStyle = xlContinuous
.Borders(xlInsideHorizontal).ColorIndex = 0
.Borders(xlInsideHorizontal).TintAndShade = 0
.Borders(xlInsideHorizontal).Weight = xlThin
```

End With

End Sub

' PROCEDURES DEFINITION - Format_Range_Clear

' -PURPOSE: Formats introduced range to have no borders

' -PARAMETERS: CellsRange (ByRef) -> Range to be formatted

' -RETURN VALUES: None

Public Sub Format_Range_Clear(ByRef CellsRange As Range)

With CellsRange

```
.Borders(xlDiagonalDown).LineStyle = xlNone
.Borders(xlDiagonalUp).LineStyle = xlNone
.Borders(xlEdgeLeft).LineStyle = xlNone
.Borders(xlEdgeTop).LineStyle = xlNone
.Borders(xlEdgeBottom).LineStyle = xlNone
.Borders(xlEdgeRight).LineStyle = xlNone
.Borders(xlEdgeLeft).LineStyle = xlNone
.Borders(xlInsideVertical).LineStyle = xlNone
.Borders(xlInsideHorizontal).LineStyle = xlNone
```

End With

End Sub

' PROCEDURES DEFINITION - Format_Range_RedThickBorder

' -PURPOSE: Formats introduced range to have thick external red border

' -PARAMETERS: CellsRange (ByRef) -> Range to be formatted

' -RETURN VALUES: None

Public Sub Format_Range_RedThickBorder(ByRef CellsRange As Range)

With CellsRange

```
.Borders(xlEdgeLeft).LineStyle = xlContinuous
.Borders(xlEdgeLeft).Color = -16776961
.Borders(xlEdgeLeft).TintAndShade = 0
.Borders(xlEdgeLeft).Weight = xlThick
.Borders(xlEdgeTop).LineStyle = xlContinuous
.Borders(xlEdgeTop).Color = -16776961
.Borders(xlEdgeTop).TintAndShade = 0
```

```

.Borders(xlEdgeTop).Weight = xlThick
.Borders(xlEdgeBottom).LineStyle = xlContinuous
.Borders(xlEdgeBottom).Color = -16776961
.Borders(xlEdgeBottom).TintAndShade = 0
.Borders(xlEdgeBottom).Weight = xlThick
.Borders(xlEdgeRight).LineStyle = xlContinuous
.Borders(xlEdgeRight).Color = -16776961
.Borders(xlEdgeRight).TintAndShade = 0
.Borders(xlEdgeRight).Weight = xlThick
.Borders(xlEdgeLeft).LineStyle = xlContinuous
.Borders(xlEdgeLeft).Color = -16776961
.Borders(xlEdgeLeft).TintAndShade = 0
.Borders(xlEdgeLeft).Weight = xlThick
End With
End Sub
'-----
'      PROCEDURES DEFINITION - Date_to_WeekString
'      -PURPOSE: Creates a string from a date in order to be used as a data
header. String format is WKxx, where xx is the weeeek number
'      -PARAMETERS: Date_Value -> Date to be processed
'      -RETURN VALUES: String with processed text
'-----
Public Function Date_to_WeekString(ByVal Date_Value As Date) As String
    Dim Ret_Str As String

    Dim D As Date
    Dim FW As Integer
    Dim VBAWeekNum As Integer
    Dim WK_Str As String
    Dim YR_Str As String

    WK_Str = CStr(CInt(Format(Date_Value, "ww", FW)) - 1)
    If Strings.Len(WK_Str) = 1 Then WK_Str = "0" + WK_Str

    YR_Str = Format(Date_Value, "yy", FW)
    Ret_Str = "WK" + WK_Str + " / " + YR_Str

    Date_to_WeekString = Ret_Str

Exit Function
End Function
'-----
'      PROCEDURES DEFINITION - Date_to_DateString
'      -PURPOSE: Creates a string from a date in order to be used as a data
header. String format "mm/dd/yy"
'      -PARAMETERS: Date_Value -> Date to be processed
'      -RETURN VALUES: String with processed text
'-----
Public Function Date_to_DateString(ByVal Date_Value As Date) As String

    Date_to_DateString = Format(Date_Value, "mm/dd/yy")

```

```

Exit Function
End Function
'-----
'      PROCEDURES DEFINITION - Clean_Project_Code
'      -PURPOSE: Filters input project code string in order to be usable as as data
input.
'      Processes done are UpperCasing the string, removing unused spaces
and parsing it to have XXX XXXXX XX format.
'      -PARAMETERS: Project_Code -> Project_Code String to be processed
'      -RETURN VALUES: String with processed text
'-----
Public Function Clean_Project_Code(ByVal Project_Code As String) As String
    Dim Temp_Str As String
    Temp_Str = Clean_String(Project_Code)

    If Temp_Str <> "" Then
        If Strings.Len(Temp_Str) = 10 Then
            Temp_Str = Strings.Left(Temp_Str, 3) + " " + Strings.Mid(Temp_Str, 4, 5) + " " +
Strings.Right(Temp_Str, 2)
        ElseIf Strings.Len(Temp_Str) = 11 Then
            If Strings.Mid(Temp_Str, 4, 1) = " " Then
                Temp_Str = Strings.Left(Temp_Str, 9) + " " + Strings.Right(Temp_Str, 2)
            ElseIf Strings.Mid(Temp_Str, 9, 1) = " " Then
                Temp_Str = Strings.Left(Temp_Str, 3) + " " + Strings.Right(Temp_Str, 8)
            Else
                Temp_Str = ""
            End If
        ElseIf Strings.Len(Temp_Str) = 12 Then
            Temp_Str = Temp_Str
        Else
            Temp_Str = ""
        End If

    End If

    Clean_Project_Code = Temp_Str
Exit Function
End Function
'-----
'      PROCEDURES DEFINITION - Clean_PM_Name
'      -PURPOSE: Filters input project manager string in order to be usable as as
data input.
'      Processes done are UpperCasing the string, removing unused
spaces,
'      reduce PM name to a single char and parsing it to have X. XXXXX
format.
'      -PARAMETERS: PM_Name -> PM_Name String to be processed
'      -RETURN VALUES: String with processed text
'-----
Public Function Clean_PM_Name(ByVal PM_Name As String) As String

```

```

Dim Temp_Str As String
Dim Temp_Str_2 As String
Dim I As Integer

Temp_Str = Clean_String(PM_Name)
Temp_Str_2 = Strings.Left(Temp_Str, 1) + ". "
For I = 1 To Strings.Len(Temp_Str)
    If Strings.Mid(Temp_Str, I, 1) = " " Then
        Temp_Str_2 = Temp_Str_2 + Strings.Right(Temp_Str, Strings.Len(Temp_Str) -
l)
        I = Strings.Len(Temp_Str)
    End If
Next

Clean_PM_Name = Temp_Str_2
Exit Function
End Function
'-----
'      PROCEDURES DEFINITION - Clean_FDS_Phase
'      -PURPOSE: Filters input FDS_Phase string in order to be usable as as data
input.
'      Processes done are "cleaning" the string and reduce the string to a
single number.
'      FDS Phase number is checked to be 1-7, in any other case char "X" is
used
'      -PARAMETERS: FDS_Phase -> FDS_Phase String to be processed
'      -RETURN VALUES: String with processed text
'-----
Public Function Clean_FDS_Phase(ByVal FDS_Phase As String) As String
    Dim Temp_Str As String

    Temp_Str = Clean_String(FDS_Phase)
    Temp_Str = Strings.Right(Temp_Str, 1)

    Select Case Temp_Str
        Case "1", "2", "3", "4", "5", "6", "7"
            Temp_Str = Temp_Str
        Case Else
            Temp_Str = "X"
    End Select
    Clean_FDS_Phase = Temp_Str
Exit Function
End Function
'-----
'      PROCEDURES DEFINITION - Clean_BU_String
'      -PURPOSE: Filters input BU string in order to be usable as as data input.
'      Processes done are "cleaning" the string.
'      BU String is checked to be among valid ones, in any other case
original string is used
'      -PARAMETERS: BU_String -> BU_String String to be processed
'      -RETURN VALUES: String with processed text

```

```

'-----
Public Function Clean_BU_String(ByVal BU_String As String) As String
    Dim Temp_Str As String
    Temp_Str = Clean_String(BU_String)
    Select Case Temp_Str
        Case "IDNEO TECHNOLOGIES", "IDNEO"
            Temp_Str = "IDNEO"
        Case "A3", "AC", "ADV. COMMS", "ADVANCED COMMUNICATIONS", "TCU",
"ADVANCED COMMUNICATIONS SYSTEM"
            Temp_Str = "AC"
        Case "ADAS", "ADASENS"
            Temp_Str = "ADAS"
        Case "REAR VIEW SYSTEMS"
            Temp_Str = "MIRROR"
        Case "PLASTICS & ELECTRICAL"
            Temp_Str = "UH"
        Case "COMMERCIAL VEHICLE"
            Temp_Str = "COMMER"
        Case "AFTERMARKET"
            Temp_Str = "AFTER"
        Case "EMS", "EMOBILITY"
            Temp_Str = "EMS"
        Case Else
            Temp_Str = Temp_Str
    End Select

    Clean_BU_String = Temp_Str
Exit Function
End Function
'-----
'      PROCEDURES DEFINITION - Clean_Region_String
'      -PURPOSE: Filters input Region string in order to be usable as as data input.
'      Processes done are "cleaning" the string.
'      Region String is checked to be among valid ones, in any other case
original string is used
'      -PARAMETERS: Region_String -> Region_String String to be processed
'      -RETURN VALUES: String with processed text
'-----
Public Function Clean_Region_String(ByVal Region_String As String) As String
    Dim Temp_Str As String
    Temp_Str = Clean_String(Region_String)
    Select Case Temp_Str
        Case "SOUTH-EUROPE", "IBERIA"
            Temp_Str = "S-EU"
        Case "NORTH EUROPE"
            Temp_Str = "N-EU"
        Case "ASIA - PACIFIC"
            Temp_Str = "A-PAC"
        Case "NAFTA"
            Temp_Str = "NAFTA"
        Case "MERCOSUR"

```



```

        Temp_Str = "MERCO"
    Case Else
        Temp_Str = Temp_Str
    End Select

    Clean_Region_String = Temp_Str
    Exit Function
End Function
'-----
'   PROCEDURES DEFINITION - Clean_PLANT_String
'   -PURPOSE: Filters input Plant string in order to be usable as as data input.
'   Processes done are "cleaning" the string.
'   Plant String is checked to be among valid ones, in any other case
original string is used
'   -PARAMETERS: PLANT_String -> PLANT_String String to be processed
'   -RETURN VALUES: String with processed text
'-----
Public Function Clean_PLANT_String(ByVal PLANT_String As String) As String
    Dim Temp_Str As String
    Temp_Str = Clean_String(PLANT_String)
    Select Case Temp_Str
        Case "SORIA"
            Temp_Str = "SOR"
        Case "DIEUZE"
            Temp_Str = "DIE"
        Case "NALDA"
            Temp_Str = "NAL"
        Case "POLONIA"
            Temp_Str = "POL"
        Case "MORCONE"
            Temp_Str = "MOR"
        Case "BRUYERES"
            Temp_Str = "BRU"
        Case "FR.-PL.-SPAIN"
            Temp_Str = "SOR"
        Case "MEXICO"
            Temp_Str = "MEX"
        Case "BRASIL"
            Temp_Str = "BRA"
        Case "SHANGAI"
            Temp_Str = "SHG"
        Case "BURSA"
            Temp_Str = "BUR"
        Case "FICOSA ELECTRONICS", "FE", "FE - SPAIN", "FE-SPAIN", "FE -SPAIN",
"FE- SPAIN"
            Temp_Str = "FE"
        Case "VILADECAVALLS -SPAIN", "VILADECAVALLS-SPAIN",
"VILADECAVALLS - SPAIN", "VILADECAVALLS", "VLDC", "VLD"
            Temp_Str = "VLD"
    End Select
End Function

```

```

        Case "IDNEO", "IDNEO MFT.", "IDNEO MANUFACTURING - SPAIN", "IDNEO
MANUFACTURING-SPAIN", "IDNEO MANUFACTURING- SPAIN", "IDNEO
MANUFACTURING -SPAIN"
            Temp_Str = "IDN"
        Case Else
            Temp_Str = Temp_Str
        End Select

        Clean_PLANT_String = Temp_Str
    Exit Function
End Function

'-----
'      PROCEDURES DEFINITION - Clean_String
'      -PURPOSE: Processes done are UpperCasing the string and removing
unused spaces
'      -PARAMETERS: String_In -> String_In String to be processed
'      -RETURN VALUES: String with processed text
'-----

Public Function Clean_String(ByVal String_In As String) As String
    Dim Str As String
    Str = Strings.UCase(String_In)

    While Strings.Left(Str, 1) = " "
        Str = Strings.Right(Str, Strings.Len(Str) - 1)
    Wend
    While Strings.Right(Temp_Str, 1) = " "
        Str = Strings.Left(Str, Strings.Len(Str) - 1)
    Wend

    Clean_String = Str
    Exit Function
End Function

'-----
'      PROCEDURES DEFINITION - FDS_MS_Name_Process
'      -PURPOSE: Filters input FDS Milestone String in order to be usable as as
data input.
'      Defines FDS MS Symbol string with the proper char according if it's
open or closed
'      Processes done to FDS Milestone String are "cleaning" the string.
'      and then checked to be among valid ones, in any other case original
string is used
'      if valid MS name is detected, a symbol is assigned as well.
'
'      NOTE: If no valid MS text is detected, Text is unchanged and Symbol
will be a default one
'      -PARAMETERS:   FDS_MS_Name (ByRef) -> FDS_MS_Name String to be
processed
'                   FDS_MS_Symbol (ByRef) -> FDS_MS_Symbol String where FDS
Symbol will be stored

```

```
'          FDS_MS_Closed -> FDS_MS_Closed Boolean indicating if MS is
closed (true) or not (false)
'          -RETURN VALUES: None
'-----
```

```
Public Sub FDS_MS_Name_Process(ByRef FDS_MS_Name As String, ByRef
FDS_MS_Symbol As String, ByVal FDS_MS_Closed As Boolean)
    FDS_MS_Symbol = ""
```

```
    FDS_MS_Name = Clean_String(FDS_MS_Name)
```

```
    Select Case FDS_MS_Name
```

```
        Case "PHASE 2"
```

```
            FDS_MS_Name = "PH2"
```

```
            If Not (FDS_MS_Closed) Then
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled2_Empty)
```

```
            Else
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled2_Fill)
```

```
            End If
```

```
        Case "DPR3 PH3", "DPR3+PH3"
```

```
            FDS_MS_Name = "PH3"
```

```
            If Not (FDS_MS_Closed) Then
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled3_Empty)
```

```
            Else
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled3_Fill)
```

```
            End If
```

```
        Case "DPR4 PH4", "DPR4+PH4"
```

```
            FDS_MS_Name = "PH4"
```

```
            If Not (FDS_MS_Closed) Then
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled4_Empty)
```

```
            Else
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled4_Fill)
```

```
            End If
```

```
        Case "DPR5 HS4 + PH5", "DPR5+HS4+PH5", "PH5 HS4"
```

```
            FDS_MS_Name = "PH5"
```

```
            If Not (FDS_MS_Closed) Then
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled5_Empty)
```

```
            Else
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled5_Fill)
```

```
            End If
```

```
        Case "SOP FICOSA + PH6", "SOP FICOSA+PH6", "PH6 (SOP FICOSA)"
```

```
            FDS_MS_Name = "PH6"
```

```
            If Not (FDS_MS_Closed) Then
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled6_Empty)
```

```
            Else
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled6_Fill)
```

```
            End If
```

```
        Case "PH7"
```

```
            FDS_MS_Name = "PH7"
```

```
            If Not (FDS_MS_Closed) Then
```

```
                FDS_MS_Symbol = Chr(Symbol_Circled7_Empty)
```

```
            Else
```

```
FDS_MS_Symbol = Chr(Symbol_Circled7_Fill)
End If

Case "TKO AUDIT"
FDS_MS_Name = "TKO"
If Not (FDS_MS_Closed) Then
FDS_MS_Symbol = Chr(Symbol_5Star_Empty)
Else
FDS_MS_Symbol = Chr(Symbol_5Star_Fill)
End If
Case "INTERNAL R@R (HS5)", "R@R AUDIT"
FDS_MS_Name = "R@R"
If Not (FDS_MS_Closed) Then
FDS_MS_Symbol = Chr(Symbol_5Star_Empty)
Else
FDS_MS_Symbol = Chr(Symbol_5Star_Fill)
End If

Case "START DVP"
FDS_MS_Name = "DVP"
If Not (FDS_MS_Closed) Then
FDS_MS_Symbol = Chr(Symbol_Square_Empty)
Else
FDS_MS_Symbol = Chr(Symbol_Square_Fill)
End If
Case "START PVT"
FDS_MS_Name = "PVT"
If Not (FDS_MS_Closed) Then
FDS_MS_Symbol = Chr(Symbol_Square_Empty)
Else
FDS_MS_Symbol = Chr(Symbol_Square_Fill)
End If

Case "FRONTLOAD DPR2"
FDS_MS_Name = "FL"
If Not (FDS_MS_Closed) Then
FDS_MS_Symbol = Chr(Symbol_Circle_Empty)
Else
FDS_MS_Symbol = Chr(Symbol_Circle_Fill)
End If
Case "HS2 (1ST ASSEMBLY)", "IDPR5 + HS2 (1ST OFF TOOL ASSY)"
FDS_MS_Name = "HS2"
If Not (FDS_MS_Closed) Then
FDS_MS_Symbol = Chr(Symbol_Circle_Empty)
Else
FDS_MS_Symbol = Chr(Symbol_Circle_Fill)
End If
Case "HS3"
FDS_MS_Name = "HS3"
If Not (FDS_MS_Closed) Then
FDS_MS_Symbol = Chr(Symbol_Circle_Empty)
```

```

Else
    FDS_MS_Symbol = Chr(Symbol_Circle_Fill)
End If
Case "CUSTOMER R@R"
    FDS_MS_Name = "R@R CUS"
    If Not (FDS_MS_Closed) Then
        FDS_MS_Symbol = Chr(Symbol_Circle_Empty)
    Else
        FDS_MS_Symbol = Chr(Symbol_Circle_Fill)
    End If
Case "PSW"
    FDS_MS_Name = "PSW"
    If Not (FDS_MS_Closed) Then
        FDS_MS_Symbol = Chr(Symbol_Circle_Empty)
    Else
        FDS_MS_Symbol = Chr(Symbol_Circle_Fill)
    End If
Case "SOP CUSTOMER"
    FDS_MS_Name = "SOP CUS"
    If Not (FDS_MS_Closed) Then
        FDS_MS_Symbol = Chr(Symbol_Circle_Empty)
    Else
        FDS_MS_Symbol = Chr(Symbol_Circle_Fill)
    End If
Case Else
    FDS_MS_Name = FDS_MS_Name
    If Not (FDS_MS_Closed) Then
        FDS_MS_Symbol = Chr(Symbol_Circle_Empty)
    Else
        FDS_MS_Symbol = Chr(Symbol_Circle_Fill)
    End If
End Select
End Sub
'-----
'   PROCEDURES DEFINITION - SL_to_INT
'   -PURPOSE: Converts Lotus DB data (v,a,r) to integer values (100,50,0)
'   NOTE: If no valid SL text is detected, function will return 0
'   -PARAMETERS:  SL_String -> Lotus DB String to be processed
'   -RETURN VALUES: None
'-----
Public Function SL_to_INT(SL_String As String) As Integer
    Dim SL_VAL As Integer

    SL_VAL = 0
    SL_String = Clean_String(SL_String)

    Select Case SL_String
        Case "R"
            SL_VAL = 0
        Case "V"
            SL_VAL = 100
    End Select

```

```
Case "A"  
  SL_VAL = 50  
Case Else  
  SL_VAL = 0  
End Select  
  
SL_to_INT = SL_VAL  
Exit Function  
End Function
```



```
' -> LotusDB_FDS_Type_to_String(LotusDB As LotusDB_FDS_Type) As String
'
'-----
' DEVELOPER NOTES
'-----
'
'-----
' EXTERNAL REFERENCES
'-----
Private Declare Function SetCurrentDirectory Lib "kernel32" Alias "SetCurrentDirectoryA"
(ByVal lpPathName As String) As Long
'-----
' CONSTANT DEFINITION
'-----

' ODBC Connection String declaration for Lotus Notes FDS DB. Note that different
countries FDS DB require a different connection string
Private Const ConnectionString_Lotus_Generic As String = _
    "UserName=;EncryptPWD=;" & _
    "MaxSubquery=50;MaxStmtLen=4096;MaxRels=20;MaxVarcharLen=254;" & _

"KeepTempldx=1;MaxLongVarcharLen=512;ShowImplicitFlds=0;MapSpecialChars=1;Thre
adTimeout=60;"

Private Const ConnectionString_FDS_SPA As String =
"ODBC;DSN=FDS_SPA;Database=APPS\F25\FDSv7.nsf;Server=APPS/FICOSA;" &
ConnectionString_Lotus_Generic
Private Const ConnectionString_FDS_BRA As String =
"ODBC;DSN=FDS_BRA;Database=DESARROLLO\FDSv7.nsf;Server=FICBRAZIL/FICO
SA;" & ConnectionString_Lotus_Generic
Private Const ConnectionString_FDS_POR As String =
"ODBC;DSN=FDS_POR;Database=Desarrollo\FDSv7.nsf;Server=F32SRV05/FCBP/CTR
L/FICOSA;" & ConnectionString_Lotus_Generic
Private Const ConnectionString_FDS_CHI As String =
"ODBC;DSN=FDS_CHI;Database=Desarrollo\FDSv7.nsf;Server=F98SRV05/FICH/MIRR/
FICOSA;" & ConnectionString_Lotus_Generic
Private Const ConnectionString_FDS_GER As String =
"ODBC;DSN=FDS_GER;Database=desarrollo\FDSv7.nsf;Server=WILKE/FICOSA;" &
ConnectionString_Lotus_Generic
Private Const ConnectionString_FDS_FRA As String =
"ODBC;DSN=FDS_FRA;Database=Desarrollo\FDSV7.NSF;Server=DIEUZE/FICOSA;" &
ConnectionString_Lotus_Generic
Private Const ConnectionString_FDS_ITA As String =
"ODBC;DSN=FDS_ITA;Database=Desarrollo\FDSv7.nsf;Server=VENARIA/FICOSA;" &
ConnectionString_Lotus_Generic
Private Const ConnectionString_FDS_NAF As String =
"ODBC;DSN=FDS_NAF;Database=Desarrollo\FDSv7.nsf;Server=MEXICO/FICOSA;" &
ConnectionString_Lotus_Generic
```

```
Private Const ConnectionString_FDS_TUR As String =  
"ODBC;DSN=FDS_TUR;Database=Desarrollo\FDSv7.nsf;Server=F07SRV05/FITK/SRVC  
/FICOSA;" & ConnectionString_Lotus_Generic  
Private Const ConnectionString_FDS_POL As String =  
"ODBC;DSN=FDS_POL;Database=FDSv7.nsf;Server=F95SRV05/FMPL/MIRR/FICOSA;"  
& ConnectionString_Lotus_Generic  
Private Const ConnectionString_CMS As String =  
"ODBC;DSN=CMS_v3;Database=APPS\F25\CDSv10test.nsf;Server=APPS/FICOSA;" &  
ConnectionString_Lotus_Generic
```

```
'-----  
' GLOBAL DATA ESTRUCTURES DEFINITION  
'-----
```

```
Public Enum LotusDB_Query_Type  
    [_First] = 1  
    FDS_Streetlights = 1  
    FDS_Milestones = 2  
    FDS_Checklists = 3  
    CMS = 4  
    CMS_Pen = 5  
    DocA3_GA = 6  
    [_Last] = 6  
End Enum
```

```
Public Enum LotusDB_FDS_Type  
    [_First] = 1  
    FDS_SPA = 1  
    FDS_BRA = 2  
    FDS_POR = 3  
    FDS_CHI = 4  
    FDS_GER = 5  
    FDS_FRA = 6  
    FDS_ITA = 7  
    FDS_NAF = 8  
    FDS_TUR = 9  
    FDS_POL = 10  
    CMS_v3 = 11  
    DocA3 = 12  
  
    [_Last] = 12  
End Enum
```

```
'-----  
' GLOBAL VARIABLES DEFINITION  
'-----
```

```
'-----  
' PROCEDURES DEFINITION  
'-----  
'-----
```

```

'      PROCEDURES DEFINITION - LotusDataImport
'      -PURPOSE: This function imports data from selected Lotus DB and selected
query.
'      Imported data is stored in a specific Worksheet, sheet is created if it
doesn't exists
'      DB_Link is deleted after data retrieval
'      -PARAMETERS: LotusDB -> Lotus DB to Open
'      LotusDB_Query -> Query to make
'      DateRange_Min -> Minimum date to retrieve date.
'      DateRange_Max -> Maximum date to retrieve date.
'      Project_Code/Project_CodeList -> List of projects to be used as
query, note that if empty array or first element is "*" no
'      project code filtering will be applied.
'      -RETURN VALUES: None
'-----
Public Sub LotusDataImport_1(LotusDB As LotusDB_FDS_Type, LotusDB_Query As
LotusDB_Query_Type, DateRange_Min As Date, DateRange_Max As Date,
Project_Code As String)
' -> Replaces Project_CodeList() String input by Project_Code String
Dim Project_CodeList(1 To 1) As String
Project_CodeList(1) = Project_Code
Call LotusDataImport(LotusDB, LotusDB_Query, DateRange_Min, DateRange_Max,
Project_CodeList)
End Sub
Public Sub LotusDataImport_2(LotusDB As LotusDB_FDS_Type, LotusDB_Query As
LotusDB_Query_Type, DateRange_Min As Date, DateRange_Max As Date)
' -> Removes Project_CodeList() String input, project filtering is disabled
Dim Project_CodeList(1 To 1) As String
Project_CodeList(1) = "*"
Call LotusDataImport(LotusDB, LotusDB_Query, DateRange_Min, DateRange_Max,
Project_CodeList)
End Sub
Public Sub LotusDataImport_3(LotusDB As LotusDB_FDS_Type, LotusDB_Query As
LotusDB_Query_Type, Project_CodeList() As String)
' -> Removes DateRange_Min,DateRange_Max Dates inputs, Date filtering is disabled
Dim DateRange_Min As Date
Dim DateRange_Max As Date
DateRange_Min = "1/01/2015"
DateRange_Max = "1/01/2050"
Call LotusDataImport(LotusDB, LotusDB_Query, DateRange_Min, DateRange_Max,
Project_CodeList)
End Sub
Public Sub LotusDataImport_4(LotusDB As LotusDB_FDS_Type, LotusDB_Query As
LotusDB_Query_Type, Project_Code As String)
' -> Removes DateRange_Min,DateRange_Max Dates inputs, Date filtering is disabled
' -> Replaces Project_CodeList() String input by Project_Code String
Dim DateRange_Min As Date
Dim DateRange_Max As Date
DateRange_Min = "1/01/2015"
DateRange_Max = "1/01/2050"
Dim Project_CodeList(1 To 1) As String

```

```
Project_CodeList(1) = Project_Code
Call LotusDataImport(LotusDB, LotusDB_Query, DateRange_Min, DateRange_Max,
Project_CodeList)
End Sub
Public Sub LotusDataImport_5(LotusDB As LotusDB_FDS_Type, LotusDB_Query As
LotusDB_Query_Type)
' -> Removes DateRange_Min,DateRange_Max Dates inputs, Date filtering is disabled
' -> Removes Project_CodeList() String input, project filtering is disabled
Dim DateRange_Min As Date
Dim DateRange_Max As Date
DateRange_Min = "1/01/2015"
DateRange_Max = "1/01/2050"
Dim Project_CodeList(1 To 1) As String
Project_CodeList(1) = "*"
Call LotusDataImport(LotusDB, LotusDB_Query, DateRange_Min, DateRange_Max,
Project_CodeList)
End Sub

Public Sub LotusDataImport(LotusDB As LotusDB_FDS_Type, LotusDB_Query As
LotusDB_Query_Type, DateRange_Min As Date, DateRange_Max As Date,
Project_CodeList() As String)
'Base implementation of original LotusDataImport Procedure
Dim conString As String 'Auxiliar variable to store DataBase connection definition string
Dim query As String 'Auxiliar variable to store DataBase Query
Dim MsgBox_Result As VbMsgBoxResult 'Auxiliar variable used to store MsgBox
Results
Dim I As Integer
Dim sheet_index As Integer

'Obtain proper Connection and Query strings
If LotusDB_Query = CMS Then
conString = ConnectionString_CMS
Else
conString = GetConnectionString_FDS(LotusDB)
End If

query = GetQuery(LotusDB_Query, DateRange_Min, DateRange_Max,
Project_CodeList())

If conString <> "" And query <> "" Then ' Check if Query and connection strings are
defined
Dim target As Range
sheet_index = -1
For I = 1 To ThisWorkbook.Sheets.Count 'Check if data destination sheet exists
a = ThisWorkbook.Sheets(I).Name
If ThisWorkbook.Sheets(I).Name = "DB_AUX_SHEET" Then
sheet_index = I
End If
Next
If sheet_index = -1 Then ' If data destination sheet doesn't exist, create it
```

```

        ThisWorkbook.Sheets.Add
After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count)
        ThisWorkbook.Sheets(l).Name = "DB_AUX_SHEET"
        sheet_index = l
    End If

    'Select detination area
    Set target = ThisWorkbook.Sheets(sheet_index).Cells(1, 1)
    ThisWorkbook.Sheets(sheet_index).Activate
    target.Select

    'Perform data import form Lotus DB
    Select Case ImportSQLtoQueryTable(conString, query, target)
        Case 0    'Lotus Notes Query done OK
        Case Else 'Lotus Notes Query done with Unexpected Error
            MsgBox_Result = MsgBox("DataBase handling error...", vbInformation, "Lotus
DB access...")
    End Select

    'Remove Data link so no external interfaces are detected by Excel
    ActiveWorkbook.Connections("Connection").Delete
Else
    MsgBox_Result = MsgBox("Error specifying Lotus DB or Query", vbInformation,
"Lotus DB access...")
End If

End Sub
'-----
'    PROCEDURES DEFINITION - GetConnectionString_FDS
'    -PURPOSE: This function generates a proper string to be used as connection
definition
'    -PARAMETERS: LotusDB -> Lotus DB to Open
'    -RETURN VALUES: Returns DB Connection String
'-----
Private Function GetConnectionString_FDS(LotusDB As LotusDB_FDS_Type) As String
    Select Case LotusDB
        ' SQL Query String declaration for Retrieving FDS Streetlights
        Case LotusDB_FDS_Type.FDS_SPA
            GetConnectionString_FDS = ConnectionString_FDS_SPA
        Case LotusDB_FDS_Type.FDS_BRA
            GetConnectionString_FDS = ConnectionString_FDS_BRA
        Case LotusDB_FDS_Type.FDS_POR
            GetConnectionString_FDS = ConnectionString_FDS_POR
        Case LotusDB_FDS_Type.FDS_CHI
            GetConnectionString_FDS = ConnectionString_FDS_CHI
        Case LotusDB_FDS_Type.FDS_GER
            GetConnectionString_FDS = ConnectionString_FDS_GER
        Case LotusDB_FDS_Type.FDS_FRA
            GetConnectionString_FDS = ConnectionString_FDS_FRA
        Case LotusDB_FDS_Type.FDS_ITA
            GetConnectionString_FDS = ConnectionString_FDS_ITA
    End Select
End Function

```

```

Case LotusDB_FDS_Type.FDS_NAF
  GetConnectionString_FDS = ConnectionString_FDS_NAF
Case LotusDB_FDS_Type.FDS_TUR
  GetConnectionString_FDS = ConnectionString_FDS_TUR
Case LotusDB_FDS_Type.FDS_POL
  GetConnectionString_FDS = ConnectionString_FDS_POL
Case LotusDB_FDS_Type.CMS_v3
  GetConnectionString_FDS = ConnectionString_CMS
Case LotusDB_FDS_Type.DocA3
  GetConnectionString_FDS = ConnectionString_DocA3
Case Else
  GetConnectionString_FDS = ""
End Select
End Function
-----
'   PROCEDURES DEFINITION - GetQuery
'   -PURPOSE: This function generates a proper string to be used as query
'   -PARAMETERS: LotusDB_Query -> Query to make
'               DateRange_Min -> Minimum date to retrieve date
'               DateRange_Max -> Maximum date to retrieve date
'   -RETURN VALUES: Returns Query String
-----
Private Function GetQuery(LotusDB_Query As LotusDB_Query_Type, DateRange_Min
As Date, DateRange_Max As Date, Project_CodeList() As String) As String

Dim Date_Min_String As String
Dim Date_Max_String As String
Dim Project_Code_Size As Integer
Dim Project_Code_Query_Str As String
Dim Project_Code_QueryHeader_Str As String
Dim I As Integer

Date_Min_String = Format(DateRange_Min, "yyyy-mm-dd")
Date_Max_String = Format(DateRange_Max, "yyyy-mm-dd")

Project_Code_Size = -1
Project_Code_Query_Str = ""

Select Case LotusDB_Query
Case LotusDB_Query_Type.FDS_Streetlights
  ' SQL Query String declaration for Retrieving FDS Streetlights table projects
  Project_Code_QueryHeader_Str = "semaforobyProject2.semaforo_project="
Case LotusDB_Query_Type.FDS_Milestones
  ' SQL Query String declaration for Retrieving FDS Milestones table projects
  Project_Code_QueryHeader_Str = "Milestone.semaforo_project="
Case LotusDB_Query_Type.FDS_Checklists
  ' SQL Query String declaration for Retrieving FDS Checklists status table projects
  Project_Code_QueryHeader_Str = "StatusPC." + Chr(34) + "_18" + Chr(34) + "="
Case LotusDB_Query_Type.CMS
  ' SQL Query String declaration for Retrieving CMS information
  Project_Code_QueryHeader_Str = "StatusSQL="

```

```

Case LotusDB_Query_Type.CMS_Pen
  ' SQL Query String declaration for Retrieving CMS information
  Project_Code_QueryHeader_Str = "SQLPend="
Case LotusDB_Query_Type.DocA3_GA
  ' SQL Query String declaration for Retrieving CMS information
  """"""""Project_Code_QueryHeader_Str = "SQLPend="
Case Else
  Project_Code_QueryHeader_Str = ""
End Select

On Error Resume Next
  Project_Code_Size = UBound(Project_CodeList) - LBound(Project_CodeList) + 1
If Err.Number = 0 Then
  'Project_CodeList is not empty, generate appropriate query
  If Project_Code_Size >= 1 And Project_CodeList(LBound(Project_CodeList)) <> ""
Then
  Project_Code_Query_Str = " AND ("
  For I = LBound(Project_CodeList) To UBound(Project_CodeList)
    Project_Code_Query_Str = Project_Code_Query_Str +
Project_Code_QueryHeader_Str + Project_CodeList(I) + ""
    If I = UBound(Project_CodeList) Then
      Project_Code_Query_Str = Project_Code_Query_Str + ")"
    Else
      Project_Code_Query_Str = Project_Code_Query_Str + " OR "
    End If
  Next
  End If
Else
  'Project_CodeList is empty, remove Project queries
End If

Select Case LotusDB_Query

Case LotusDB_Query_Type.FDS_Streetlights
  ' SQL Query String declaration for Retrieving FDS Streetlights
  GetQuery = _
  "SELECT semaforobyProject2.semaforo_project, semaforobyProject2.SC,
semaforobyProject2.SD, semaforobyProject2.SCu, semaforobyProject2.SQ,
semaforobyProject2.ST, semaforobyProject2." + Chr(34) + "_99" + Chr(34) + Chr(13) + ""
& Chr(10) & _
  "FROM semaforobyProject2 semaforobyProject2" & Chr(13) & "" & Chr(10) & _
  "WHERE (semaforobyProject2." + Chr(34) + "_99" + Chr(34) + " Is Not Null)" & _
  Project_Code_Query_Str
  '-----
  '-----TO DEFINE HOW TO FILTER PROPERLY DATE
STREETLIGHTS
  'AND (semaforobyProject2." + Chr(34) + "_99" + Chr(34) + ">" + Date_Min_String
+ "") AND
  ' (semaforobyProject2." + Chr(34) + "_99" + Chr(34) + "<" + Date_Max_String +
"") & Chr(13) & "" & Chr(10) &

```

```

'-----
Case LotusDB_Query_Type.FDS_Milestones
' SQL Query String declaration for Retrieving FDS Milestones
GetQuery = _
"SELECT Milestone.semaforo_project, Milestone.Persona, Milestone.ms,
Milestone.msf, Milestone.msr, Milestone.mst" + Chr(13) + "" & Chr(10) & _
"FROM Milestone Milestone" & Chr(13) & "" & Chr(10) & _
"WHERE (Milestone.semaforo_project Is Not Null) AND (Milestone.msf>" +
Date_Min_String + "") AND (Milestone.msf<" + Date_Max_String + "")" & _
Project_Code_Query_Str
Case LotusDB_Query_Type.FDS_Checklists
' SQL Query String declaration for Retrieving FDS Checklists status
GetQuery = _
"SELECT StatusPC." + Chr(34) + "_18" + Chr(34) + ", StatusPC." + Chr(34) + "_16"
+ Chr(34) + ", StatusPC.fds1_fecha_revi, StatusPC.fds1_fecha_previs_fase,
StatusPC.fds1_revi1_1" & Chr(13) & "" & Chr(10) & _
"FROM StatusPC StatusPC" & Chr(13) & "" & Chr(10) '& _
"WHERE (StatusPC.fds1_fecha_revi>" + Date_Min_String + "") AND
(StatusPC.fds1_fecha_revi<" + Date_Max_String + "")"
Case LotusDB_Query_Type.CMS
' SQL Query String declaration for Retrieving CMS information
GetQuery = _
"SELECT StatusSQL." + Chr(34) + "_1" + Chr(34) + ", StatusSQL.fldECRNumber,
StatusSQL.fldECRStatus, StatusSQL.fldProjectReference, StatusSQL.fldRequestor,
StatusSQL.fldSubject, StatusSQL.fldGlobalStatus, StatusSQL.fldPlant,
StatusSQL.fldProjectManager, StatusSQL.fld_Leader_Project,
StatusSQL.fld_Phase2_Target, StatusSQL.fld_Phase3_Target,
StatusSQL.fld_Phase4_Target, StatusSQL.fld_Phase5_Target,
StatusSQL.fldPARefFico_1, StatusSQL.fldDesignCost, StatusSQL.fldDesignTiming,
StatusSQL.fldStockCost, StatusSQL.fldLotSizeCost, StatusSQL.fldAppNextLotDate,
StatusSQL.fldLineModificationCost, StatusSQL.fldLineModificationDate,
StatusSQL.fldAssemblyCost, StatusSQL.fldLabourCost, StatusSQL.fldDeltaPieceCost,
StatusSQL.fldToolingCost, StatusSQL.fldToolingTiming, StatusSQL.fldDeltaSellingCost,
StatusSQL.fldDeltaToolingCost, StatusSQL.fldRygPhase2, StatusSQL.fldRygPhase3,
StatusSQL.fldRygPhase4, StatusSQL.fldRygPhase5, StatusSQL.fldDelta_EBIT_Percent,
StatusSQL.fldDelta_EBIT_EUR" & Chr(13) & "" & Chr(10) & _
", StatusSQL.fldAppDate_1, StatusSQL.fldAppDate_2, StatusSQL.fldAppDate_3,
StatusSQL.fldAppDate_4, StatusSQL.fldAppDate_5, StatusSQL.fldAppDate_6,
StatusSQL.fldAppDate_7, StatusSQL.fldAppDate_8, StatusSQL.fldAppDate_9,
StatusSQL.fldAppDate_10, StatusSQL.fldAppDate_11, StatusSQL.fldAppDate_12,
StatusSQL.fldAppDate_13, StatusSQL.fldAppDate_14, StatusSQL.fldAppDate_15,
StatusSQL.fldDelegDate_1, StatusSQL.fldDelegDate_2, StatusSQL.fldDelegDate_3,
StatusSQL.fldDelegDate_4, StatusSQL.fldDelegDate_5, StatusSQL.fldDelegDate_6,
StatusSQL.fldDelegDate_7, StatusSQL.fldDelegDate_8, StatusSQL.fldDelegDate_9,
StatusSQL.fldDelegDate_10, StatusSQL.fldDelegDate_11, StatusSQL.fldDelegDate_12,
StatusSQL.fldDelegDate_13, StatusSQL.fldDelegDate_14, StatusSQL.fldDelegDate_15"
& Chr(13) & "" & Chr(10) & _
"FROM StatusSQL StatusSQL" & Chr(13) & "" & Chr(10)

Case LotusDB_Query_Type.CMS_Pen
' SQL Query String declaration for Retrieving CMS information

```



```

    GetQuery = _
    "SELECT SQLPend." + Chr(34) + "_1" + Chr(34) + ", SQLPend.fldECRStatus,
SQLPend." + Chr(34) + "_2" + Chr(34) + ", SQLPend.fldECRNumber,
SQLPend.fldProjectReference, SQLPend.fldProjectManager, SQLPend.fldPlant" &
Chr(13) & "" & Chr(10) & _
    "FROM SQLPend SQLPend" & Chr(13) & "" & Chr(10)
    Case LotusDB_Query_Type.DocA3_GA
    ""GetQuery =
    """"SELECT Referencias__3__Por_Type_of_part.Tipo,
Referencias__3__Por_Type_of_part.cod_proyecto,
Referencias__3__Por_Type_of_part.n_referencia,
Referencias__3__Por_Type_of_part.proyecto,
Referencias__3__Por_Type_of_part.referencia" & Chr(13) & "" & Chr(10) & _
    """"FROM Referencias__3__Por_Type_of_part
Referencias__3__Por_Type_of_part" & Chr(13) & "" & Chr(10) & _
    Case Else
    GetQuery = ""
    End Select

```

End Function

```

'-----
'      PROCEDURES DEFINITION - ImportSQLtoQueryTable
'      -PURPOSE: This function imports data from selected Lotus DB and selected
query.
'      Imported data is stored in a specific Worksheet Range
'      -PARAMETERS: conString -> Lotus DB to Open
'      query -> Query to make
'      target -> Worksheet Range where imported data is stored
'      -RETURN VALUES: None
'-----

```

Private Function ImportSQLtoQueryTable(ByVal conString As String, ByVal query As String, ByVal target As Range) As Integer

```

    ImportSQLtoQueryTable = -1
    On Error Resume Next

```

```

    Dim ws As Worksheet
    Set ws = target.Worksheet

```

```

    'Select destination range
    Dim address As String
    address = target.Cells(1, 1).address

```

```

    ' Procedure recreates ListObject or QueryTable
    If Not target.ListObject Is Nothing Then ' Created in Excel 2007 or higher
        target.ListObject.Delete
    ElseIf Not target.QueryTable Is Nothing Then ' Created in Excel 2003
        target.QueryTable.ResultRange.Clear
        target.QueryTable.Delete
    End If

```

```

If Application.Version >= "12.0" Then      ' Excel 2007 or higher
  With ws.ListObjects.Add(SourceType:=0, Source:=Array(conString),
Destination:=Range(address))
    With .QueryTable
      .CommandType = xlCmdSql
      .CommandText = StringToArray(query)
      .BackgroundQuery = True
      .SavePassword = False
      .Refresh BackgroundQuery:=False 'When this command is executed password
will be requested
    End With
  End With
Else                                          ' Excel 2003
  With ws.QueryTables.Add(Connection:=Array(conString),
Destination:=Range(address))
    .CommandType = xlCmdSql
    .CommandText = StringToArray(query)
    .BackgroundQuery = True
    .SavePassword = False
    .Refresh BackgroundQuery:=False 'When this command is executed password will
be requested
  End With
End If

```

```

ImportSQLtoQueryTable = 0

```

```

End Function

```

```

'-----
'   PROCEDURES DEFINITION - StringToArray
'   Source: http://support.microsoft.com/kb/816562
'-----

```

```

Private Function StringToArray(Str As String) As Variant

```

```

  Const StrLen = 127

```

```

  Dim NumElems As Integer

```

```

  Dim Temp() As String

```

```

  Dim I As Integer

```

```

  NumElems = (Len(Str) / StrLen) + 1

```

```

  ReDim Temp(1 To NumElems) As String

```

```

  For I = 1 To NumElems

```

```

    Temp(I) = Mid(Str, ((I - 1) * StrLen) + 1, StrLen)

```

```

  Next I

```

```

  StringToArray = Temp

```

```

End Function

```

```

'-----
'   PROCEDURES DEFINITION - LotusDB_FDS_Type_to_String

```

```
'           -PURPOSE: This function generates a proper string to identify FDS DB String
Name
'           -PARAMETERS: LotusDB -> Lotus DB to convert String
'           -RETURN VALUES: Returns FDS DB Identifier String
'-----
Public Function LotusDB_FDS_Type_to_String(LotusDB As LotusDB_FDS_Type) As
String
    Select Case LotusDB
    ' String to identify which Lotus Database is selected
    Case LotusDB_FDS_Type.FDS_SPA
        LotusDB_FDS_Type_to_String = "FDS SPAIN"
    Case LotusDB_FDS_Type.FDS_BRA
        LotusDB_FDS_Type_to_String = "FDS BRAZIL"
    Case LotusDB_FDS_Type.FDS_POR
        LotusDB_FDS_Type_to_String = "FDS PORTUGAL"
    Case LotusDB_FDS_Type.FDS_CHI
        LotusDB_FDS_Type_to_String = "FDS CHINA"
    Case LotusDB_FDS_Type.FDS_GER
        LotusDB_FDS_Type_to_String = "FDS GERMANY"
    Case LotusDB_FDS_Type.FDS_FRA
        LotusDB_FDS_Type_to_String = "FDS FRANCE"
    Case LotusDB_FDS_Type.FDS_ITA
        LotusDB_FDS_Type_to_String = "FDS ITALY"
    Case LotusDB_FDS_Type.FDS_NAF
        LotusDB_FDS_Type_to_String = "FDS NAFTA"
    Case LotusDB_FDS_Type.FDS_TUR
        LotusDB_FDS_Type_to_String = "FDS TURKEY"
    Case LotusDB_FDS_Type.FDS_POL
        LotusDB_FDS_Type_to_String = "FDS POLAND"
    Case LotusDB_FDS_Type.CMS_v3
        LotusDB_FDS_Type_to_String = "CMS V3.0"
    Case Else
        LotusDB_FDS_Type_to_String = ""
    End Select
End Function
```



```
' | 1.0 | X. Palacio | First Release | 25/09/2017 |
' | | | | |
'-----
'
```

```
' Design Hierarchy :
'-----
'
```

```
' The complete design hierarchy is as shown below (left to right) :
'
```

```
' -> NO Class Creation
'-----
'
```

```
' DEVELOPER NOTES
'-----
'
```

```
' EXTERNAL REFERENCES
'-----
'
```

```
Public Sub Ranking()
```

```
Dim QueryRow As Integer
```

```
Dim PM As String
```

```
Dim PMIN As Boolean
```

```
Dim PMPending As String
```

```
Dim Punctuation As Double
```

```
ThisWorkbook.Sheets("CMS Ranking").Range("D4:P25").ClearContents
```

```
'CMS Pending Ranking
```

```
QueryRow = 4
```

```
Do
```

```
PM = UCase(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3))
```

```
QueryPM = 2
```

```
PMIN = False
```

```
Do
```

```
PMPending = UCase(ThisWorkbook.Sheets("CMS Pending").Cells(QueryPM, 9))
```

```
If PM = PMPending Then
```

```
PMIN = True
```

```
Exit Do
```

```
End If
```

```
QueryPM = QueryPM + 1
```

```
Loop Until ThisWorkbook.Sheets("CMS Pending").Cells(QueryPM, 9) = ""
```

```
If PMIN = True Then
```

```
ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 4) =
```

```
ThisWorkbook.Sheets("CMS Pending").Cells(QueryPM, 10)
```

```
Else
```

```
ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 4) = 0
```

```
End If
```

```
QueryRow = QueryRow + 1
```

```
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""
```

```

'CMS %Approved Ranking
QueryRow = 4
Do
  PM = UCase(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3))
  QueryPM = 6
  PMIN = False
  Do
    PMPending = UCase(ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM,
17))
    If PM = PMPending Then
      PMIN = True
      Exit Do
    End If
    QueryPM = QueryPM + 1
  Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 17) = ""

  If PMIN = True Then
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 8) =
ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 20)
  Else
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 8) = "-"
  End If
  QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

'CMS N3 and N4 Pending Ranking
QueryRow = 4
Do
  PM = UCase(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3))
  QueryPM = 6
  PMIN = False
  Do
    PMPending = UCase(ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM,
17))
    If PM = PMPending Then
      PMIN = True
      Exit Do
    End If
    QueryPM = QueryPM + 1
  Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 17) = ""

  If PMIN = True Then
    N4 = ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 21)
    N3 = ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 22)
    DelayCMS = ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 23)
    If IsEmpty(N4) Then
      ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 6) = 0
      ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 6).Interior.ColorIndex
= 4

```

```

Else
    Count = 1
    For I = 1 To Len(N4)
        If Mid(N4, I, 1) = "," Then
            Count = Count + 1
        End If
    Next I
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 6) = Count
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 6).Interior.ColorIndex
= 3
End If

If IsEmpty(N3) Then
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 5) = 0
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 5).Interior.ColorIndex
= 4
Else
    Count = 1
    For I = 1 To Len(N3)
        If Mid(N3, I, 1) = "," Then
            Count = Count + 1
        End If
    Next I
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 5) = Count
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 5).Interior.ColorIndex
= 3
End If

If IsEmpty(DelayCMS) Then
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 7) = 0
Else
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 7) = DelayCMS
End If

Else
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 6) = 0
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 5) = 0
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 6).Interior.ColorIndex = 4
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 5).Interior.ColorIndex = 4
End If
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

'The calculation of CHECK 2 (Ph5 Date) will be in the module
CMS_Import_New_Products

QueryRow = 4
Do
    PM = UCase(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3))
    QueryPM = 3
    PMIN = False

```

```
Do
  PMPending = ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 1)
  If PM = PMPending Then
    PMIN = True
    Exit Do
  End If
  QueryPM = QueryPM + 1
Loop Until ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 1) = ""

If PMIN = True Then
  ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 9) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 3)
  ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 12) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 8)
Else
  ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 9) = 0
End If
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

'The calculation of CHECK 3 (Ph5 Date ON TIME) will be in the module
CMS_Import_New_Products
QueryRow = 4
Do
  PM = UCase(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3))
  QueryPM = 3
  PMIN = False
  Do
    PMPending = ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 10)
    If PM = PMPending Then
      PMIN = True
      Exit Do
    End If
    QueryPM = QueryPM + 1
  Loop Until ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 10) = ""

  If PMIN = True Then
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 13) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 12)
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 16) =
ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 17)

Else
  ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 13) = 0
End If
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

'CMS N3 and N4 Ranking fot PH4Date ON TIME
QueryRow = 4
Do
```



```

PM = UCase(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3))
QueryPM = 3
PMIN = False
Do
    PMPending = UCase(ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 10))
    If PM = PMPending Then
        PMIN = True
        Exit Do
    End If
    QueryPM = QueryPM + 1
Loop Until ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 10) = ""

If PMIN = True Then
    N3 = ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 15)
    N4 = ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 16)
    If IsEmpty(N3) Or N3 = 0 Then
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 14) = 0
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 14).Interior.ColorIndex
= 4
    Else
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 14) = N3
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 14).Interior.ColorIndex
= 3
    End If

    If IsEmpty(N4) Or N4 = 0 Then
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 15) = 0
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 15).Interior.ColorIndex
= 4
    Else
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 15) = N4
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 15).Interior.ColorIndex
= 3
    End If
    Else
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 14) = 0
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 15) = 0
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 14).Interior.ColorIndex =
4
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 15).Interior.ColorIndex =
4
    End If
    QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

'CMS N3 and N4 Ranking for PH4Date after 7 or 15 days the CMS is approved
QueryRow = 4
Do
    PM = UCase(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3))
    QueryPM = 3
    PMIN = False

```

```

Do
  PMPending = UCase(ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 1))
  If PM = PMPending Then
    PMIN = True
    Exit Do
  End If
  QueryPM = QueryPM + 1
Loop Until ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 1) = ""

If PMIN = True Then
  N3 = ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 6)
  N4 = ThisWorkbook.Sheets("CMS CHK").Cells(QueryPM, 7)
  If IsEmpty(N3) Or N3 = 0 Then
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 10) = 0
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 10).Interior.ColorIndex
= 4
  Else
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 10) = N3
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 10).Interior.ColorIndex
= 3
  End If

  If IsEmpty(N4) Or N4 = 0 Then
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 11) = 0
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 11).Interior.ColorIndex
= 4
  Else
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 11) = N4
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 11).Interior.ColorIndex
= 3
  End If
Else
  ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 10) = 0
  ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 11) = 0
  ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 10).Interior.ColorIndex =
4
  ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 11).Interior.ColorIndex =
4
End If
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

```

'Calculation of Points. The clasification is going to work as follows. The more points you are, the worse your ranking.

QueryRow = 4

Do

```

DelayCHK1 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 7)
DelayCHK2 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 12)
DelayCHK3 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 16)

```

```
PendingCHK1 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 4)
PendingCHK2 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 9)
PendingCHK3 = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 13)

Progress = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 8)

If Progress = "-" Then
    Progress = 1
End If

If IsEmpty(DelayCHK1) Then
    DelayCHK1 = 0
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 7) = 0
End If

If IsEmpty(DelayCHK2) Then
    DelayCHK2 = 0
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 12) = 0
End If

If IsEmpty(DelayCHK3) Then
    DelayCHK3 = 0
    ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 16) = 0
End If

Puntuation = DelayCHK1 + DelayCHK2 * 3 + DelayCHK3 * 3 + 5 * (PendingCHK1 +
PendingCHK2 + PendingCHK3) + (1 - Progress) * 100

ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 17) = Puntuation
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

QueryRow = 4
Do
    PM = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3)
    QueryPM = 6
    Do
        PMQuery = ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 17)
        If PMQuery = PM Then
            ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 18).Value =
ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 18).Value +
ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 19).Value
            ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 19) =
ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 17) /
ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 18)
            ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 19) =
Round(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 19), 2)
        Exit Do
    End If
    QueryPM = QueryPM + 1
    Loop Until ThisWorkbook.Sheets("CMS Identification").Cells(QueryPM, 17) = ""
```

```

    QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

QueryRow = 4
Do
    If IsEmpty(ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 18)) Then
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 18) = 0
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 19) = 0
    End If
    QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

ThisWorkbook.Sheets("CMS Ranking").Sort.SortFields.Clear
ThisWorkbook.Sheets("CMS Ranking").Sort.SortFields.Add Key:=Range("S4:S25"), _
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
With ThisWorkbook.Sheets("CMS Ranking").Sort
    .SetRange Range("B4:T25")
    .Header = xlGuess
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With

'Make the new clasification
QueryRow = 4
RP = 1
Do
    Position = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 1)
    Punctuation = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 19)

    If Punctuation = 0 Then
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 1) = 0
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 1).Interior.ColorIndex = 1
    Else
        ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 1) = RP
        If RP = 1 Or RP = 2 Or RP = 3 Or RP = 4 Then
            ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 1).Interior.ColorIndex
= 4
        End If
        RP = RP + 1
    End If
    QueryRow = QueryRow + 1

Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

'Compare with the last week value
QueryRow = 4
Do

```

```
ActualPosition = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 1)
LastWeekPosition = ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 20)
ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 2) = LastWeekPosition -
ActualPosition
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""

result = MsgBox("Do you want to change the last week position of the PM?
(Recommended to change it on Friday)", vbOKCancel + vbExclamation, "Important")
Select Case result
Case vbOK:
QueryRow = 4
Do
ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 20) =
ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 1)
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("CMS Ranking").Cells(QueryRow, 3) = ""
MsgBox "Last week positions changed"
Case vbCancel:
MsgBox "Last week positions still being the same"
End Select

End Sub
```



```
' |           |           |           |           |
' -----
'
' -----
' Design Hierarchy :
' -----
'
' The complete design hierarchy is as shown below (left to right) :
'
' -> NO Class Creation
' -----
' DEVELOPER NOTES
' -----
'
' -----
' EXTERNAL REFERENCES
' -----
Public Sub Week_CMS()
    Dim QueryColumn As Integer
    Dim QueryWeek As Integer

    'We want to know in which Column is going to appear the weekly CMS information
    Do
        QueryColumn = QueryColumn + 1
        Loop Until ThisWorkbook.Sheets("CMS Summary").Cells(4, QueryColumn) = ""

    'We identify in which week we are and in which column we are going to wirte the
    information
    QueryWeek = ThisWorkbook.Sheets("CMS Identification").Cells(29, 2)
    LastWeek = ThisWorkbook.Sheets("CMS Summary").Cells(4, QueryColumn - 1)
    If QueryWeek <> LastWeek Then
        If QueryColumn = 12 Then
            'Copy values for Global Stats, Pending Stats and Progress
            ThisWorkbook.Sheets("CMS Summary").Range("C4:K10").Copy
            Destination:=ThisWorkbook.Sheets("CMS Summary").Range("B4:J10")
            ThisWorkbook.Sheets("CMS Identification").Range("B29:B35").Copy
            Destination:=ThisWorkbook.Sheets("CMS Summary").Range("K4:K10")
            ThisWorkbook.Sheets("CMS Summary").Range("C12:K16").Copy
            Destination:=ThisWorkbook.Sheets("CMS Summary").Range("B12:J16")
            ThisWorkbook.Sheets("CMS Summary").Range("C18:K20").Copy
            Destination:=ThisWorkbook.Sheets("CMS Summary").Range("B18:J20")
            ThisWorkbook.Sheets("CMS Summary").Range("C22:K22").Copy
            Destination:=ThisWorkbook.Sheets("CMS Summary").Range("B22:J22")
            ThisWorkbook.Sheets("CMS Summary").Range("C24:K25").Copy
            Destination:=ThisWorkbook.Sheets("CMS Summary").Range("B24:J25")
            ThisWorkbook.Sheets("CMS Summary").Range("C27:K30").Copy
            Destination:=ThisWorkbook.Sheets("CMS Summary").Range("B27:J30")
            ThisWorkbook.Sheets("CMS Summary").Range("C32:K35").Copy
            Destination:=ThisWorkbook.Sheets("CMS Summary").Range("B32:J35")
        End If
    End If
End Sub
```

```
ThisWorkbook.Sheets("CMS Summary").Range("C38:K39").Copy
Destination:=ThisWorkbook.Sheets("CMS Summary").Range("B38:J39")
'Copy Pending Stats
ThisWorkbook.Sheets("CMS Summary").Cells(12, 11) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 17)
ThisWorkbook.Sheets("CMS Summary").Cells(13, 11) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 13)
ThisWorkbook.Sheets("CMS Summary").Cells(14, 11) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 14)
ThisWorkbook.Sheets("CMS Summary").Cells(15, 11) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 15)
ThisWorkbook.Sheets("CMS Summary").Cells(16, 11) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 16)
'Copy Approvers and PM
ThisWorkbook.Sheets("CMS Summary").Cells(18, 11) =
ThisWorkbook.Sheets("CMS Pending").Cells(5, 7)
ThisWorkbook.Sheets("CMS Summary").Cells(19, 11) =
ThisWorkbook.Sheets("CMS Pending").Cells(6, 7)
ThisWorkbook.Sheets("CMS Summary").Cells(20, 11) =
ThisWorkbook.Sheets("CMS Summary").Cells(10, 11) - ThisWorkbook.Sheets("CMS
Summary").Cells(10, 10)
'CMS Electronics Progress
ThisWorkbook.Sheets("CMS Summary").Cells(22, 11) =
ThisWorkbook.Sheets("CMS Identification").Cells(6, 24)
ThisWorkbook.Sheets("CMS Summary").Cells(24, 11) =
ThisWorkbook.Sheets("CMS CHK").Cells(2, 9)
ThisWorkbook.Sheets("CMS Summary").Cells(25, 11) =
(ThisWorkbook.Sheets("CMS Summary").Cells(23, 11) - ThisWorkbook.Sheets("CMS
Summary").Cells(24, 11)) / ThisWorkbook.Sheets("CMS Summary").Cells(23, 11)

ThisWorkbook.Sheets("CMS Summary").Cells(27, 11) =
ThisWorkbook.Sheets("CMS Summary").Cells(4, 11)
ThisWorkbook.Sheets("CMS Summary").Cells(28, 11) =
ThisWorkbook.Sheets("CMS CHK").Cells(4, 9)
ThisWorkbook.Sheets("CMS Summary").Cells(29, 11) =
ThisWorkbook.Sheets("CMS CHK").Cells(6, 9)
ThisWorkbook.Sheets("CMS Summary").Cells(30, 11) =
ThisWorkbook.Sheets("CMS CHK").Cells(8, 9)

ThisWorkbook.Sheets("CMS Summary").Cells(32, 11) =
ThisWorkbook.Sheets("CMS Summary").Cells(4, 11)
ThisWorkbook.Sheets("CMS Summary").Cells(33, 11) =
ThisWorkbook.Sheets("CMS CHK").Cells(4, 18)
ThisWorkbook.Sheets("CMS Summary").Cells(34, 11) =
ThisWorkbook.Sheets("CMS CHK").Cells(6, 18)
ThisWorkbook.Sheets("CMS Summary").Cells(35, 11) =
ThisWorkbook.Sheets("CMS CHK").Cells(8, 18)

ThisWorkbook.Sheets("CMS Summary").Cells(39, 11) =
ThisWorkbook.Sheets("CMS Ranking").Cells(29, 19)
```


Else

'Values for Global Stats

ThisWorkbook.Sheets("CMS Summary").Cells(4, QueryColumn) =
ThisWorkbook.Sheets("CMS Identification").Cells(29, 2)
ThisWorkbook.Sheets("CMS Summary").Cells(5, QueryColumn) =
ThisWorkbook.Sheets("CMS Identification").Cells(30, 2)
ThisWorkbook.Sheets("CMS Summary").Cells(6, QueryColumn) =
ThisWorkbook.Sheets("CMS Identification").Cells(31, 2)
ThisWorkbook.Sheets("CMS Summary").Cells(7, QueryColumn) =
ThisWorkbook.Sheets("CMS Identification").Cells(32, 2)
ThisWorkbook.Sheets("CMS Summary").Cells(8, QueryColumn) =
ThisWorkbook.Sheets("CMS Identification").Cells(33, 2)
ThisWorkbook.Sheets("CMS Summary").Cells(9, QueryColumn) =
ThisWorkbook.Sheets("CMS Identification").Cells(34, 2)
ThisWorkbook.Sheets("CMS Summary").Cells(10, QueryColumn) =
ThisWorkbook.Sheets("CMS Identification").Cells(35, 2)

'Values for Pending Stats

ThisWorkbook.Sheets("CMS Summary").Cells(12, QueryColumn) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 17)
ThisWorkbook.Sheets("CMS Summary").Cells(13, QueryColumn) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 13)
ThisWorkbook.Sheets("CMS Summary").Cells(14, QueryColumn) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 14)
ThisWorkbook.Sheets("CMS Summary").Cells(15, QueryColumn) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 15)
ThisWorkbook.Sheets("CMS Summary").Cells(16, QueryColumn) =
ThisWorkbook.Sheets("CMS Pending").Cells(4, 16)

'Values for Approvers and PM

ThisWorkbook.Sheets("CMS Summary").Cells(18, QueryColumn) =
ThisWorkbook.Sheets("CMS Pending").Cells(5, 7)
ThisWorkbook.Sheets("CMS Summary").Cells(19, QueryColumn) =
ThisWorkbook.Sheets("CMS Pending").Cells(6, 7)
ThisWorkbook.Sheets("CMS Summary").Cells(20, QueryColumn) =
ThisWorkbook.Sheets("CMS Summary").Cells(10, QueryColumn) -
ThisWorkbook.Sheets("CMS Summary").Cells(10, QueryColumn - 1)

'CMS Electronics Progress

ThisWorkbook.Sheets("CMS Summary").Cells(22, QueryColumn) =
ThisWorkbook.Sheets("CMS Identification").Cells(6, 24)
ThisWorkbook.Sheets("CMS Summary").Cells(24, QueryColumn) =
ThisWorkbook.Sheets("CMS CHK").Cells(2, 9)
ThisWorkbook.Sheets("CMS Summary").Cells(25, QueryColumn) =
(ThisWorkbook.Sheets("CMS Summary").Cells(23, QueryColumn) -
ThisWorkbook.Sheets("CMS Summary").Cells(24, QueryColumn)) /
ThisWorkbook.Sheets("CMS Summary").Cells(23, QueryColumn)

ThisWorkbook.Sheets("CMS Summary").Cells(27, QueryColumn) =
ThisWorkbook.Sheets("CMS Summary").Cells(4, QueryColumn)
ThisWorkbook.Sheets("CMS Summary").Cells(28, QueryColumn) =
ThisWorkbook.Sheets("CMS CHK").Cells(4, 9)
ThisWorkbook.Sheets("CMS Summary").Cells(29, QueryColumn) =
ThisWorkbook.Sheets("CMS CHK").Cells(6, 9)

```
ThisWorkbook.Sheets("CMS Summary").Cells(30, QueryColumn) =  
ThisWorkbook.Sheets("CMS CHK").Cells(8, 9)
```

```
ThisWorkbook.Sheets("CMS Summary").Cells(32, QueryColumn) =  
ThisWorkbook.Sheets("CMS Summary").Cells(4, QueryColumn)  
ThisWorkbook.Sheets("CMS Summary").Cells(33, QueryColumn) =  
ThisWorkbook.Sheets("CMS CHK").Cells(4, 18)  
ThisWorkbook.Sheets("CMS Summary").Cells(34, QueryColumn) =  
ThisWorkbook.Sheets("CMS CHK").Cells(6, 18)  
ThisWorkbook.Sheets("CMS Summary").Cells(35, QueryColumn) =  
ThisWorkbook.Sheets("CMS CHK").Cells(8, 18)
```

```
ThisWorkbook.Sheets("CMS Summary").Cells(39, QueryColumn) =  
ThisWorkbook.Sheets("CMS Ranking").Cells(29, 19)
```

```
End If
```

```
Elseif QueryWeek = LastWeek Then
```

```
    'Values for Global Stats
```

```
    ThisWorkbook.Sheets("CMS Summary").Cells(4, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Identification").Cells(29, 2)  
    ThisWorkbook.Sheets("CMS Summary").Cells(5, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Identification").Cells(30, 2)  
    ThisWorkbook.Sheets("CMS Summary").Cells(6, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Identification").Cells(31, 2)  
    ThisWorkbook.Sheets("CMS Summary").Cells(7, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Identification").Cells(32, 2)  
    ThisWorkbook.Sheets("CMS Summary").Cells(8, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Identification").Cells(33, 2)  
    ThisWorkbook.Sheets("CMS Summary").Cells(9, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Identification").Cells(34, 2)  
    ThisWorkbook.Sheets("CMS Summary").Cells(10, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Identification").Cells(35, 2)
```

```
    'Values for Pending Stats
```

```
    ThisWorkbook.Sheets("CMS Summary").Cells(12, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Pending").Cells(4, 17)  
    ThisWorkbook.Sheets("CMS Summary").Cells(13, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Pending").Cells(4, 13)  
    ThisWorkbook.Sheets("CMS Summary").Cells(14, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Pending").Cells(4, 14)  
    ThisWorkbook.Sheets("CMS Summary").Cells(15, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Pending").Cells(4, 15)  
    ThisWorkbook.Sheets("CMS Summary").Cells(16, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Pending").Cells(4, 16)
```

```
    'Values for Approvers and PM
```

```
    ThisWorkbook.Sheets("CMS Summary").Cells(18, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Pending").Cells(5, 7)  
    ThisWorkbook.Sheets("CMS Summary").Cells(19, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Pending").Cells(6, 7)  
    ThisWorkbook.Sheets("CMS Summary").Cells(20, QueryColumn - 1) =  
ThisWorkbook.Sheets("CMS Summary").Cells(10, QueryColumn - 1) -  
ThisWorkbook.Sheets("CMS Summary").Cells(10, QueryColumn - 2)
```

```

    'CMS Electronics Progress
    ThisWorkbook.Sheets("CMS Summary").Cells(22, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS Identification").Cells(6, 24)
    ThisWorkbook.Sheets("CMS Summary").Cells(24, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS CHK").Cells(2, 9)
    ThisWorkbook.Sheets("CMS Summary").Cells(25, QueryColumn - 1) =
(ThisWorkbook.Sheets("CMS Summary").Cells(23, QueryColumn - 1) -
ThisWorkbook.Sheets("CMS Summary").Cells(24, QueryColumn - 1)) /
ThisWorkbook.Sheets("CMS Summary").Cells(23, QueryColumn - 1)

    ThisWorkbook.Sheets("CMS Summary").Cells(27, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS Summary").Cells(4, QueryColumn - 1)
    ThisWorkbook.Sheets("CMS Summary").Cells(28, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS CHK").Cells(4, 9)
    ThisWorkbook.Sheets("CMS Summary").Cells(29, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS CHK").Cells(6, 9)
    ThisWorkbook.Sheets("CMS Summary").Cells(30, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS CHK").Cells(8, 9)

    ThisWorkbook.Sheets("CMS Summary").Cells(32, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS Summary").Cells(4, QueryColumn - 1)
    ThisWorkbook.Sheets("CMS Summary").Cells(33, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS CHK").Cells(4, 18)
    ThisWorkbook.Sheets("CMS Summary").Cells(34, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS CHK").Cells(6, 18)
    ThisWorkbook.Sheets("CMS Summary").Cells(35, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS CHK").Cells(8, 18)

    ThisWorkbook.Sheets("CMS Summary").Cells(39, QueryColumn - 1) =
ThisWorkbook.Sheets("CMS Ranking").Cells(29, 19)
    End If
End Sub

Public Sub Open_CMS()
    Dim QueryRow As Integer
    Dim CMSOpen As Date
    Dim NumberOpenPrevious As Integer
    Dim NumberOpen As Integer

    QueryRow = 2
    NumberOpen = 0
    NumberOpenPrevious = 0
    Do
        Plant = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 7)
        CMSOpen = ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1)
        CMSWeek = DatePart("ww", CMSOpen)
        CMSYear = DatePart("yyyy", CMSOpen)
        ActualWeek = DatePart("ww", Date)
        ActualYear = DatePart("yyyy", Date)
        If CMSWeek = ActualWeek And CMSYear = ActualYear And Plant = "V2 Ficosa
Electronics VLDC - EMS" Then

```

```
NumberOpen = NumberOpen + 1
Elseif CMSWeek = ActualWeek - 1 And CMSYear = ActualYear And Plant = "V2
Ficosa Electronics VLDC - EMS" Then
    NumberOpenPrevious = NumberOpenPrevious + 1
End If
QueryRow = QueryRow + 1
Loop Until ThisWorkbook.Sheets("DB_AUX_SHEET").Cells(QueryRow, 1) = ""

ThisWorkbook.Sheets("CMS Pending").Cells(7, 7) = NumberOpen
ThisWorkbook.Sheets("CMS Pending").Cells(8, 7) = NumberOpenPrevious

End Sub
```

