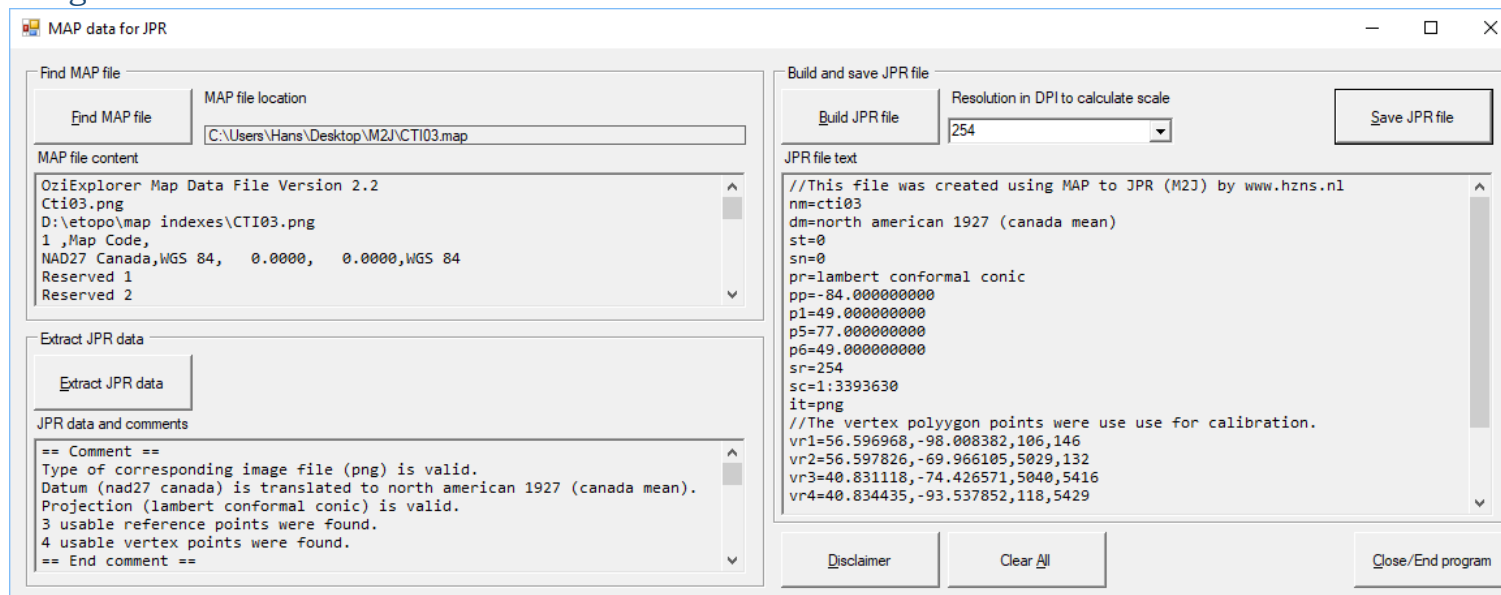


OziExplorer MAP Data to JPR (M2J)

Image of M2J



Code for Module M2J.vb

```
Public Class M2J
    'Structures declarations
    Structure JprStructure
        Public nm As String
        Public sc As Double
        Public sr As Double
        Public it As String
        Public ds As Double
        Public dm As String
        Public st As Double
        Public sn As Double
        Public pr As String
        Public zn As String
        Public pp As String
        Public p1 As String
        Public p2 As String
        Public p3 As String
        Public p4 As String
        Public p5 As String
    End Structure

    'Necessary JPR data (in the context of this application)
    'Name of the Chart
    'Scale of the Chart
    'Resolution of the Chart
    'Type of image of the Chart
    'Size of pixel in meters (no a formal JPF variable)
    'Geographic datum of the Chart
    'Datum shift (latitude)
    'Datum shift (longitude)
    'Projection of the Chart
    'UTM zone of the Chart
    'Central meridian of the Chart
    'Latitude of the origin of the Chart
    'Scalefactor of the Chart
    'False Easting of the Chart
    'Fales Northing of the Chart
    'First standard parallel of the Chart
End Class
```

```

    Public p6 As String      'Second standard parallel of the Chart
End Structure

Structure Coordinate        'Referenced Pixel and Geographical coordinates
    Public x As String      'Pixel X
    Public y As String      'Pixel Y
    Public lat As String    'Latitude
    Public lon As String    'Longitude
    Public Valid As Integer 'Application internal variable used for validation
End Structure

'Variable declarations
Public MapFilePath As String 'Filename and path to MAP-file.
Public MapFile As String     'Filename MAP-file.
Public MapPath As String     'Path to MAP-file.
Public ChartName As String   'Filename MAP-file without extention. (ChartName was used to prevent confusion with the use of map-file)
Public JprFilePath As String 'Filename and path to JPR-file.
Public MapFileContent As String 'Content of MAP-file
Public LineCounter As Integer 'Amount of lines in MapFileContent (will be (re)calculated for every MAP-file)
Public LineContent(60) As String 'Content of each line in MapFileContent (will be (re)dimensioned for every MAP-file)
Public LineField(20) As String 'Content of each field in LineContent
Public JprData As JprStructure 'Variable representing the JPR data
Public ImageWidth As Integer  'Width of the image
Public ImageHeight As Integer 'Height of the image
Public UTM As Boolean         'Indication map projection is UTM
Public TM As Boolean          'Indication map projection is Transverse Mercator
Public MERC As Boolean        'Indication map projection is Mercator
Public LCC As Boolean         'Indication map projection is Lambert Conformal Conic
Public ReferencePoint(3) As Coordinate 'Calibration points (topleft, topright, bottomright, bottomleft, valid)
Public ReferencePointCounter As Integer 'Counted number of reference points in MAP-file
Public ValidReferencePoints As Integer  'Usable Number of reference points in MAP-file
Public VertexPoint(4) As Coordinate    'Vertexpoints (topleft, topright, bottomright, bottomleft, valid)
Public VertexPointNoted As Integer      'Noted number of Vertexpoints in MAP-file
Public VertexPointLLCounter As Integer  'Counted number of vertexpoints based on Latitude/longitude
Public VertexPointXYCounter As Integer  'Counted number of vertexpoints based on X/Y coordinate
Public ValidVertexPoints As Integer     'Usable number of Vertexpoints in MAP-file
Public VertexMax As Integer             'Maximum number of Vertexpoints in MAP-file

Public EndLine As String = Chr(13) + Chr(10) 'Variable representing the end of a line of text.

'Subroutine M2J_Load
'This sub is will be executed starting the application
'- It enables and sets the Tooltips for buttons and boxes.
'- Loading values into DpiBox
'- It enables/disables buttons

Private Sub M2J_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'Declarations
    Dim TipText As New ToolTip()

```

```

'Setting up the delays for the ToolTip (TipText).
TipText.AutoPopDelay = 5000
TipText.InitialDelay = 1000
TipText.ReshowDelay = 500

'Forcing the ToolTip text to be displayed whether or not the form is active.
TipText.ShowAlways = True

' Setting up the TipText text
TipText.SetToolTip(Me.FindButton, "Find and select a MAP-file.")
TipText.SetToolTip(Me.MapInfoBox, "Content of the selected MAP-File.")
TipText.SetToolTip(Me.ExtractButton, "Extract the data form the MAP-file.")
TipText.SetToolTip(Me.JprDataBox, "Comments and extracted JPR-data.")
TipText.SetToolTip(Me.BuildButton, "Build the content for the JPR-file. ")
TipText.SetToolTip(Me.JprFileBox, "Content for the JPR-file.")
TipText.SetToolTip(Me.DpiBox, "Select or change the resolution of the chart, 254 dpi (100 dot per cm) is the default.")
TipText.SetToolTip(Me.SaveButton, "Save the content for JPR-file in the same directory as the MAP-file. If the JPR-file exists, you will be prompted to over write.")
TipText.SetToolTip(Me.JprDataBox, "Comments and extracted JPR-data")
TipText.SetToolTip(Me.DisclaimButton, "Disclaimer for the application 'MAP data to JPR' (M2J).")
TipText.SetToolTip(Me.ClearButton, "Clear all data with in the applicatation, except the selected resolution.")
TipText.SetToolTip(Me.EndButton, "Close the application.")

'Loading DpiBox
DpiBox.Items.Add("72")
DpiBox.Items.Add("96")
DpiBox.Items.Add("100")
DpiBox.Items.Add("127")
DpiBox.Items.Add("150")
DpiBox.Items.Add("200")
DpiBox.Items.Add("254")
DpiBox.Items.Add("300")
DpiBox.Items.Add("508")
'Default setting
DpiBox.Text = "254"

'Initial setting user interface (buttons)
ExtractButton.Enabled = False
BuildButton.Enabled = False
SaveButton.Enabled = False
DpiBox.Enabled = False

End Sub

'Subroutine FindButton_Click
'- Finding/Selecting and opening a MAP file
'- Creating values for variables: ChartFileName, MapFile, ChartName, JprFilePath
'- Breaking content of MAP-file in to lines and load them in an array (LineContent)
Private Sub FindButton_Click(sender As Object, e As EventArgs) Handles FindButton.Click
'Declarations
Dim FileNameLen, Pointer As Integer

```

```

Dim TempFileName As String
Dim MapContentLen, ChrPointer, LinePointer As Integer
Dim TxtChr As String = ""
Dim FindFile As New OpenFileDialog

'Clearing all data on screen and variables
Call ClearButton_Click(sender, e)

'Setting up dialog
FindFile.Filter = "Ozi-MAP file (*.map) | *.map"
FindFile.Multiselect = False
'Opening dialog
FindFile.ShowDialog()

'Retrieve value for MAPfile
TempFileName = FindFile.FileName

'Stopping this subroutine if no file was selected (preventing an error)
'(length TempfileNam = 0)
If Len(TempFileName) = 0 Then
    MsgBox("No file was selected!", 48, "Warning")
    Exit Sub
End If

'Setting values for variables
MapFilePath = TempFileName
MapFileValue.Text = TempFileName

'Retrieving and setting values for variables MapFile and ChartPath
FileNameLen = Len(TempFileName)
For Pointer = FileNameLen To 0 Step -1
    If Mid(TempFileName, Pointer, 1) = "\" Then
        MapPath = Mid(TempFileName, 1, Pointer)
        MapFile = Mid(TempFileName, Pointer + 1, FileNameLen - Pointer)
        Exit For
    End If
Next Pointer

'Retrieving and setting values for variables ChartName and JprFilePath
FileNameLen = Len(MapFile)
For Pointer = FileNameLen To 0 Step -1
    If Mid(MapFile, Pointer, 1) = "." Then
        ChartName = Mid(MapFile, 1, Pointer - 1)
        JprFilePath = MapPath & ChartName & ".jpr"
        Exit For
    End If
Next Pointer

'Retreiving content of the MapFile
MapFileContent = My.Computer.FileSystem.ReadAllText(MapPath + MapFile)

```

```

'Loading MapFileContent into MapInfoBox
MapInfoBox.Text = MapFileContent

'Breaking MapFileContent into Lines (in array LineContent)
'- Determining amount of lines (LineCounter)
MapContentLen = Len(MapFileContent)
LineCounter = 1
For I = 1 To MapContentLen
    If Mid(MapFileContent, I, 1) = Chr(13) Then
        LineCounter = LineCounter + 1
    End If
Next
'- Retrieving the lines and put them in the array LineContent
ReDim LineContent(LineCounter)
ChrPointer = 1
LinePointer = 1
Do While LinePointer < LineCounter And ChrPointer < MapContentLen
    Do
        TxtChr = Mid(MapFileContent, ChrPointer, 1)
        If TxtChr <> Chr(13) And TxtChr <> Chr(10) Then
            LineContent(LinePointer) = LineContent(LinePointer) + TxtChr
        End If
        ChrPointer = ChrPointer + 1
    Loop Until TxtChr = Chr(13) Or TxtChr = Chr(10)
    If TxtChr = Chr(13) Then LinePointer = LinePointer + 1
Loop

'Enabling ExtractButton (to extracting the JPR data)
ExtractButton.Enabled = True
ExtractButton.Select()

End Sub

'Subroutine ExtractButton_Click
'- Checking if the MAP-file is a OZI-file
'- Retrieving all necessary JPR-data

Private Sub ExtractButton_Click(sender As Object, e As EventArgs) Handles ExtractButton.Click
    'Declarations
    Dim ChrPointer As Integer
    Dim TmpPointer As Integer
    Dim LinePointer As Integer
    Dim CommentText As String = "==" Comment == " & EndLine
    Dim TmpText As String

    'Checking if MAP-data contain OziExplorer information
    If Mid(LineContent(1), 1, 11) <> "OziExplorer" Then
        JprDataBox.Text = "File " & MapFile & " wasn't created with or for OziExplorer."
    Exit Sub

```

```

End If

'Setting value ChartName
JprData.nm = LCase(ChartName)

'Checking File format of map image by the file extention in line 3
If Len(LineContent(3)) = 0 Then
    CommentText = CommentText & " file isn't linked to a image file." & EndLine
Else
    For ChrPointer = Len(LineContent(3)) To 0 Step -1
        If Mid(LineContent(3), ChrPointer, 1) = "." Then
            JprData.it = LCase(Mid(LineContent(3), ChrPointer + 1, Len(LineContent(3)) - 1))
            Exit For
        End If
    Next
    Select Case JprData.it
        Case "jpg", "png", "tif", "gif", "bmp"
            CommentText = CommentText & "Type of corresponding image file (" & JprData.it & ") is valid." & EndLine
        Case Else
            CommentText = CommentText & "Type of corresponding image file (" & JprData.it & ") isn't valid." & EndLine
    End Select
End If

'Splitting line 5 into fields to extract JPR data (datum and datum shift)
Call SplitLine(LineContent(5))
JprData.dm = LCase(LineField(1))
JprData.st = LineField(3)
JprData.sn = LineField(4)

'Evaluating and if necessary translating (MAP > JPR) datum
Select Case JprData.dm
    Case "wgs 84"
        CommentText = CommentText & "Datum (" & JprData.dm & ") is translated to wgs84." & EndLine
        JprData.dm = "wgs84"
    Case "nad83"
        CommentText = CommentText & "Datum (" & JprData.dm & ") is valid." & EndLine
    Case "nad27"
        CommentText = CommentText & "Datum (" & JprData.dm & ") is valid." & EndLine
    Case "nad27 canada"
        CommentText = CommentText & "Datum (" & JprData.dm & ") is translated to north american 1927 (canada mean)." & EndLine
        JprData.dm = "north american 1927 (canada mean)"
    Case "nad27 conus"
        CommentText = CommentText & "Datum (" & JprData.dm & ") is translated to north american 1927 (continental us)." & EndLine
        JprData.dm = "north american 1927 (continental us)"
    Case Else
        CommentText = CommentText & "Datum (" & JprData.dm & ") isn't valid for M2J." & EndLine
End Select

'Splitting line 9 into fields to extract JPR data (projection)
Call SplitLine(LineContent(9))

```

```

JprData.pr = LCase(LineField(2))

'Evaluating and if necessary translating (MAP > JPR) projection
Select Case JprData.pr
  Case "mercator"
    CommentText = CommentText & "Projection (" & JprData.pr & ") is valid." & EndLine
  Case "transverse mercator"
    CommentText = CommentText & "Projection (" & JprData.pr & ") is valid." & EndLine
  Case "(utm) universal transverse mercator"
    CommentText = CommentText & "Projection (" & JprData.pr & ") is translated to utm." & EndLine
    JprData.pr = "utm"
  Case "lambert conformal conic"
    CommentText = CommentText & "Projection (" & JprData.pr & ") is valid." & EndLine
  Case Else
    CommentText = CommentText & "Projection (" & JprData.pr & ") isn't valid or isn't supported by M2J." & EndLine
End Select

'Inserting UTM evaluation??

'Splitting line 9 into fields to extract JPR data (projection specific data)
Call SplitLine(LineContent(40))
JprData.pp = Trim(LineField(3))      'Central meridian of the Chart
JprData.p1 = Trim(LineField(2))     'Latitude of the origin of the Chart
JprData.p2 = Trim(LineField(4))     'Scalefactor of the Chart
JprData.p3 = Trim(LineField(6))     'False Easting of the Chart
JprData.p4 = Trim(LineField(5))     'False Northing of the Chart
JprData.p5 = Trim(LineField(7))     'First standard parallel of the Chart
JprData.p6 = Trim(LineField(8))     'Second standard parallel of the Chart

'Evaluation of projections
'- Resetting variables UTM, TM, MERC and LCC
UTM = True
TM = True
MERC = True
LCC = True

'- Checking if all necessary data are in place
If JprData.pp = "" Then LCC = False : TM = False
If JprData.p1 = "" Then LCC = False : TM = False : MERC = False
If JprData.p2 = "" Then TM = False
If JprData.p3 = "" Then TM = False
If JprData.p4 = "" Then TM = False
If JprData.p5 = "" Then LCC = False
If JprData.p6 = "" Then LCC = False

'- Writing to CommentText
If JprData.pp = "transverse mercator" And TM = False Then
  CommentText = CommentText & "The data for a transverse mercator projection are incomplete." & EndLine
End If
If JprData.pp = "mercator" And MERC = False Then

```

```

    CommentText = CommentText & "The data for a mercator projection are incomplete." & EndLine
End If
If JprData.pp = "lambert conformal conic" And LCC = False Then
    CommentText = CommentText & "The data for a lambert conformal conic projection are incomplete." & EndLine
End If

'Retrieving data for Reference points
'- Determining number of reference points (RefPoint array)
ReferencePointCounter = 0
For LinePointer = 10 To 39
    Call SplitLine(LineContent(LinePointer))
    If Trim(LineField(3)) <> "" Then
        ReferencePointCounter = ReferencePointCounter + 1
    Else
        Exit For
    End If
Next

'- Redemtioning/resetting RefPoint Array
ReDim ReferencePoint(ReferencePointCounter)
For TmpPointer = 1 To ReferencePointCounter
    ReferencePoint(TmpPointer).x = 0
    ReferencePoint(TmpPointer).y = 0
    ReferencePoint(TmpPointer).lat = 0
    ReferencePoint(TmpPointer).lon = 0
    ReferencePoint(TmpPointer).Valid = 0
Next

'- Determining values for the Reference points array and number of valid Reference points
ValidReferencePoints = 0
For LinePointer = 10 To 9 + ReferencePointCounter
    Call SplitLine(LineContent(LinePointer))
    If Trim(LineField(7)) = "" Or Trim(LineField(10)) = "" Then
        ReferencePoint(LinePointer - 9).Valid = ReferencePoint(LinePointer - 9).Valid + 1
    Else
        ReferencePoint(LinePointer - 9).x = Val(LineField(3))
        ReferencePoint(LinePointer - 9).y = Val(LineField(4))
        ReferencePoint(LinePointer - 9).lat = Val(LineField(7)) + Val(LineField(8)) / 60
        If Trim(LCase(LineField(9))) = "s" Then ReferencePoint(LinePointer - 9).lat = ReferencePoint(LinePointer - 9).lat * -1
        ReferencePoint(LinePointer - 9).lon = Val(LineField(10)) + Val(LineField(11)) / 60
        If Trim(LCase(LineField(12))) = "w" Then ReferencePoint(LinePointer - 9).lon = ReferencePoint(LinePointer - 9).lon * -1
        ValidReferencePoints = ValidReferencePoints + 1
    End If
Next
CommentText = CommentText & ValidReferencePoints & " usable reference points were found." & EndLine

'Retrieving data for the Vertex Polygon
'- Resetting variables
VertexPointNoted = 0
VertexPointLLCounter = 0

```

```

VertexPointXYCounter = 0

'- Retrieving the number of vertex points
For LinePointer = 41 To LineCounter
    Call SplitLine(LineContent(LinePointer))
    Select Case LCase(LineField(1))
        Case "mmpnum"
            VertexPointNoted = Val(LineField(2))
        Case "mmp11"
            VertexPointLLCounter = VertexPointLLCounter + 1
        Case "mmpxy"
            VertexPointXYCounter = VertexPointXYCounter + 1
    End Select
Next

'- Deteremining size of VertexPoint Array
VertexMax = VertexPointNoted
If VertexPointLLCounter > VertexMax Then VertexMax = VertexPointLLCounter
If VertexPointXYCounter > VertexMax Then VertexMax = VertexPointXYCounter
If VertexMax <> VertexPointNoted Then
    CommentText = CommentText & "The number of noted vertex points (" & VertexPointNoted & ") is different from number counted (" & VertexMax & ")." & EndLine
End If

'- Redemtioning/resetting VertexPoint Array
ReDim VertexPoint(VertexMax)
For TmpPointer = 1 To VertexMax
    VertexPoint(TmpPointer).x = 0
    VertexPoint(TmpPointer).y = 0
    VertexPoint(TmpPointer).lat = 0
    VertexPoint(TmpPointer).lon = 0
    VertexPoint(TmpPointer).Valid = 0
Next

'- Determening values for VertexPoint array and DotSize
For LinePointer = 41 To LineCounter
    Call SplitLine(LineContent(LinePointer))
    Select Case LCase(LineField(1))
        Case "mmp11"
            If Trim(LineField(3)) = "" Or Trim(LineField(4)) = "" Then
                VertexPoint(Val(LineField(2))).Valid = VertexPoint(Val(LineField(2))).Valid + 1
            Else
                VertexPoint(Val(LineField(2))).lon = Val(LineField(3))
                VertexPoint(Val(LineField(2))).lat = Val(LineField(4))
            End If
        Case "mmpxy"
            If Trim(LineField(3)) = "" Or Trim(LineField(4)) = "" Then
                VertexPoint(Val(LineField(2))).Valid = VertexPoint(Val(LineField(2))).Valid + 2
            Else
                VertexPoint(Val(LineField(2))).x = Val(LineField(3))
                VertexPoint(Val(LineField(2))).y = Val(LineField(4))
            End If
        End Select
    End For

```

```

        End If
        Case "mm1b"
            JprData.ds = Val(LineField(2))
        End Select
    Next

    '- Determening number of usable vertex points
    ValidVertexPoints = 0
    For TmpPointer = 1 To VertexMax
        If VertexPoint(TmpPointer).Valid = 0 Then
            ValidVertexPoints = ValidVertexPoints + 1
        End If
    Next
    CommentText = CommentText & ValidVertexPoints & " usable vertex points were found." & EndLine

    '-Checking if chart can be calibrated
    If ValidVertexPoints < 3 And ValidReferencePoints < 3 Then
        CommentText = CommentText & "The chart can not calibrated. There aren't enough points available." & EndLine
    End If

    '-Determening UTM zone
    If JprData.pr = "utm" Then
        '- Determening number (for UTM zone)
        Call SplitLine(LineContent(10))
        If Trim(LineField(14)) <> "" Then
            '-Based on first reference point
            JprData.zn = Trim(LineField(14))
        Else
            '-Based on first vertex point
            JprData.zn = Int((VertexPoint(1).lon + 186) / 6)
            If JprData.zn < 1 Then JprData.zn = JprData.zn + 60
        End If

        '-Determening north/south (for UTM zone)
        If ReferencePoint(1).Valid = 1 Then
            '-Based on longitude first reference pont
            If ReferencePoint(1).lat >= 0 Then
                JprData.zn = JprData.zn & "t"
            Else
                JprData.zn = JprData.zn & "j"
            End If
        Else
            '-Based on longitude first vertex pont
            If VertexPoint(1).lat >= 0 Then
                JprData.zn = JprData.zn & "t"
            Else
                JprData.zn = JprData.zn & "j"
            End If
        End If
        CommentText = CommentText & "UTM zone is " & JprData.zn & "." & EndLine
    End If

```

End If

```

'Last line of comment
CommentText = CommentText & "== End comment ==" & EndLine

'JPR data form MAP file
CommentText = CommentText & "== Start JPR data from MAP file ==" & EndLine
'- General data
CommentText = CommentText & "nm (chartname) = " & JprData.nm & EndLine
CommentText = CommentText & "it (graphics type) = " & JprData.it & EndLine
'- Datum data
CommentText = CommentText & "dm (datum) = " & JprData.dm & EndLine
CommentText = CommentText & "st (datum shift latitude) = " & JprData.st & EndLine
CommentText = CommentText & "sn (datum shift longitude) = " & JprData.sn & EndLine
'- Projectioion data
CommentText = CommentText & "pr (projection) = " & JprData.pr & EndLine
CommentText = CommentText & "pp (central meridian) = " & JprData.pp & EndLine
CommentText = CommentText & "p1 (latitude of origin) = " & JprData.p1 & EndLine
CommentText = CommentText & "p2 (scale factor) = " & JprData.p2 & EndLine
CommentText = CommentText & "p3 (false northing) = " & JprData.p3 & EndLine
CommentText = CommentText & "p4 (false easting) = " & JprData.p4 & EndLine
CommentText = CommentText & "p5 (standard parallel 1) = " & JprData.p5 & EndLine
CommentText = CommentText & "p6 (standard parallel 2) = " & JprData.p6 & EndLine
CommentText = CommentText & "zn (UTM zone) = " & JprData.zn & EndLine
'- Scale data
CommentText = CommentText & "dotsize = " & CommaToDot(JprData.ds) & " meter/dot" & EndLine
'- Reference points
For TmpPointer = 1 To ReferencePointCounter
    If ReferencePoint(TmpPointer).Valid = 0 Then
        TmpText = "Usable"
    Else
        TmpText = "Unusable"
    End If
    CommentText = CommentText & "rp" & TmpPointer & " (referencepoint " & TmpPointer & ") = " &
        CommaToDot(ReferencePoint(TmpPointer).lat) & ", " &
        CommaToDot(ReferencePoint(TmpPointer).lon) & ", " &
        ReferencePoint(TmpPointer).x & ", " &
        ReferencePoint(TmpPointer).y & ", " &
        TmpText & EndLine
Next
'- Vertex points
For TmpPointer = 1 To VertexMax
    If VertexPoint(TmpPointer).Valid = 0 Then
        TmpText = "Usable"
    Else
        TmpText = "Unusable"
    End If
    CommentText = CommentText & "vp" & TmpPointer & " (vertexpoint " & TmpPointer & ") = " &
        CommaToDot(VertexPoint(TmpPointer).lat) & ", " &
        CommaToDot(VertexPoint(TmpPointer).lon) & ", " &

```

```

        VertexPoint(TmpPointer).x & ", " &
        VertexPoint(TmpPointer).y & ", " &
        TmpText & EndLine
Next
CommentText = CommentText & "==" End JPR data from MAP file ==" & EndLine

'Copying Comment to JprDataBox
JprDataBox.Text = CommentText

'Enabling BuildButton
BuildButton.Enabled = True
BuildButton.Select()

End Sub

'Subroutine BuildButton_Click builds the correct content for the JPR-file
Private Sub BuildButton_Click(sender As Object, e As EventArgs) Handles BuildButton.Click

    'Building Content for JPR-File bij using function BuildJprContent
    JprFileBox.Text = BuildJprContent()

    'Enabling SaveButton
    SaveButton.Enabled = True
    SaveButton.Select()
    DpiBox.Enabled = True

End Sub

'Subroutine SaveButton_Click
'- Saving the JprFileBox.text to a JPR-file in the same directory
' as the original MAP file. If this JPR file exists a warning will be prompted.
Private Sub SaveButton_Click(sender As Object, e As EventArgs) Handles SaveButton.Click
    'Declarations
    Dim Answer As Integer          'Used as answer value for MsgBox

    'Checking for an existing file
    If My.Computer.FileSystem.FileExists(JprFilePath) = True Then
        '(Msgbox appearance: 4 = Yes/No, 32 = Question mark, 256 = second button default, 292 = total)
        Answer = MsgBox("JPR-file already exists. Overwrite?", 292, "Warning")
        If Answer = 6 Then
            'Overwriting existing JPR-file if answer on MsgBox = Yes (Answer = 6)
            My.Computer.FileSystem.WriteAllText(JprFilePath, JprFileBox.Text, False)
        End If
    Else
        'Creating JPR-file
        My.Computer.FileSystem.WriteAllText(JprFilePath, JprFileBox.Text, False)
    End If

End Sub

```

```

'Subroutne DisclaimButton shows the disclaimer for hzns.nl
Private Sub DisclaimButton_Click(sender As Object, e As EventArgs) Handles DisclaimButton.Click
    'Declarations
    Dim MsgText As String

    'Disclaimer message
    MsgText = "This application is provided 'As is'. The use of the application is on your own risk. " &
        "Direct or indirect damage by using this application is users responsibility, " &
        "not the application-builders. Redistribution by a third party (commercial of non-commercial) " &
        "is prohibit. Download the application direct from www.hzns.nl."
    MsgBox(MsgText, 48, "Disclaimer")

End Sub

'Subroutine ClearButton_Click
'- Clearing all data from screen
'- disabling CreateButton and SaveButton
Private Sub ClearButton_Click(sender As Object, e As EventArgs) Handles ClearButton.Click
    'Declarations
    Dim Pointer As Integer

    'Clearing textboxes and labes
    MapInfoBox.Text = ""
    JprDataBox.Text = ""
    JprFileBox.Text = ""
    MapFileValue.Text = ""

    'Disabling buttons and DpiBox
    ExtractButton.Enabled = False
    BuildButton.Enabled = False
    SaveButton.Enabled = False
    DpiBox.Enabled = False

    'Clearing MAP file Data
    MapFile = ""
    MapPath = ""
    MapFilePath = ""
    ChartName = ""
    JprFilePath = ""
    MapFileContent = ""
    For Pointer = 0 To LineCounter
        LineContent(Pointer) = ""
    Next

    'Focus on FindButton
    FindButton.Select()

End Sub

'Unloading and closing application

```

```

Private Sub EndButton_Click(sender As Object, e As EventArgs) Handles EndButton.Click
    Me.Close()
End Sub

'Subroutine SplitLine splits a line with comma separated fields into max 20 fields
Private Sub SplitLine(LineText As String)
    'Declarations
    Dim FieldPointer, ChrPointer As Integer
    Dim TextChr As String

    'Clear all information Array LineField
    For FieldPointer = 0 To 20
        LineField(FieldPointer) = ""
    Next

    'Splitting line into fields (comma: Chr=44)
    FieldPointer = 1
    ChrPointer = 1
    Do While ChrPointer <= Len(LineText)
        Do
            TextChr = Mid(LineText, ChrPointer, 1)
            If TextChr <> Chr(44) Then
                LineField(FieldPointer) = LineField(FieldPointer) + TextChr
            End If
            ChrPointer = ChrPointer + 1
        Loop Until TextChr = Chr(44) Or ChrPointer > Len(LineText)
        If TextChr = Chr(44) Then FieldPointer = FieldPointer + 1
    Loop
End Sub

'Subroutine DipBox_LostFocus handles the cases LostFocus and SelectedValueChanged (with focus on DipBox)
'- recalculating the map scale
Private Sub DipBox_LostFocus(sender As Object, e As EventArgs) Handles DipBox.LostFocus, DipBox.SelectedValueChanged
    '(only if JprFileContentBox contains information)
    If JprFileBox.Text <> "" Then
        If JprData.sr <> Val(DipBox.Text) Then
            Call BuildButton_Click(sender, e) '(learned by trail and error)
            SaveButton.Select()
        End If
    End If
End Sub

'Subroutine DipBox_PressKey handles the case "Enter"-key was used (with focus on DipBox)
'- recalculating the map scale
Private Sub DipBox_KeyPress(sender As Object, e As KeyPressEventArgs) Handles DipBox.KeyPress
    If JprFileBox.Text <> "" And e.KeyChar = Chr(13) Then
        If JprData.sr <> Val(DipBox.Text) Then
            JprData.sr = Val(DipBox.Text)
            Call BuildButton_Click(sender, e) '(learned by trail and error)
        End If
    End If
End Sub

```

```

    End If
End If

End Sub

Private Function BuildJprContent()
'Declarations
Dim TmpPointer1, TmpPointer2 As Integer
Dim ContentText As String

'Adding data line by line to content
ContentText = "//This file was created using MAP to JPR (M2J) by www.hzns.nl" & EndLine &
    "nm=" & JprData.nm & EndLine &
    "dm=" & JprData.dm & EndLine &
    "st=" & JprData.st & EndLine &
    "sn=" & JprData.sn & EndLine &
    "pr=" & JprData.pr & EndLine

'Handling projection specific variables
Select Case JprData.pr
Case "utm"
    ContentText = ContentText & "zn=" & JprData.zn & EndLine
Case "mercator"
    ContentText = ContentText & "p1=" & JprData.p1 & EndLine
Case "transverse mercator"
    ContentText = ContentText & "pp=" & JprData.pp & EndLine &
        "p1=" & JprData.p1 & EndLine &
        "p2=" & JprData.p2 & EndLine &
        "p3=" & JprData.p3 & EndLine &
        "p4=" & JprData.p4 & EndLine
Case "lambert conformal conic"
    ContentText = ContentText & "pp=" & JprData.pp & EndLine &
        "p1=" & JprData.p1 & EndLine &
        "p5=" & JprData.p5 & EndLine &
        "p6=" & JprData.p6 & EndLine
Case Else
    ContentText = ContentText & "//This projection is not supported by MAP To JPR, please check en edit 'p-data'" & EndLine &
        "pp=" & JprData.pp & EndLine &
        "p1=" & JprData.p1 & EndLine &
        "p2=" & JprData.p2 & EndLine &
        "p3=" & JprData.p3 & EndLine &
        "p4=" & JprData.p4 & EndLine &
        "p5=" & JprData.p5 & EndLine &
        "p6=" & JprData.p6 & EndLine
End Select

'Handling scale specific variables
JprData.sr = Val(DpiBox.Text)
JprData.sc = Int((JprData.sr / 2.54) * 100 * JprData.ds)
ContentText = ContentText & "sr=" & JprData.sr & EndLine

```

```

ContentText = ContentText & "sc=1:" & JprData.sc & EndLine

'Handling scale graphic variables
ContentText = ContentText & "it=" & JprData.it & EndLine

'Handling Reference points
If ValidVertexPoints < 3 And ValidReferencePoints < 3 Then
    ContentText = ContentText & "//The chart can not calibrated because there aren't enough points available." & EndLine
Else
    If ValidVertexPoints > ValidReferencePoints Then
        ContentText = ContentText & "//The vertex polygon points were use use for calibration." & EndLine
        TmpPointer1 = 0
        For TmpPointer2 = 1 To VertexMax
            If VertexPoint(TmpPointer2).Valid = 0 Then
                TmpPointer1 = TmpPointer1 + 1
                ContentText = ContentText & "vr" & TmpPointer1 & "=" &
                    CommaToDot(VertexPoint(TmpPointer1).lat) & "," &
                    CommaToDot(VertexPoint(TmpPointer1).lon) & "," &
                    VertexPoint(TmpPointer1).x & "," &
                    VertexPoint(TmpPointer1).y & EndLine
            End If
        Next
    Else
        ContentText = ContentText & "//The reference points were use for calibration." & EndLine
        TmpPointer1 = 0
        For TmpPointer2 = 1 To ReferencePointCounter
            If ReferencePoint(TmpPointer2).Valid = 0 Then
                TmpPointer1 = TmpPointer1 + 1
                ContentText = ContentText & "rp" & TmpPointer1 & "=" &
                    CommaToDot(ReferencePoint(TmpPointer1).lat) & "," &
                    CommaToDot(ReferencePoint(TmpPointer1).lon) & "," &
                    ReferencePoint(TmpPointer1).x & "," &
                    ReferencePoint(TmpPointer1).y & EndLine
            End If
        Next
    End If
End If

'Handling vertex polygon points
If ValidVertexPoints < 3 Then
    ContentText = ContentText & "//The vertex polygon couldn't established because there aren't enough points available." & EndLine
Else
    TmpPointer1 = 0
    For TmpPointer2 = 1 To VertexMax
        If VertexPoint(TmpPointer2).Valid = 0 Then
            TmpPointer1 = TmpPointer1 + 1
            ContentText = ContentText & "vp" & TmpPointer1 & "=" &
                VertexPoint(TmpPointer1).x & "," &
                VertexPoint(TmpPointer1).y & EndLine
        End If
    End If
End If

```



```

        Next
    End If

    'Setting return value
    Return ContentText

End Function

'Function CommaToDot Converts the numeric expression (for in for example
'a Latitude Or Longitude) into string expression with a dot (.) as a seperator
'And max 6 decimals. (A fale/save construction)
Private Function CommaToDot(LatLonValue As Double) As String
    'Declarations
    Dim TmpText1, TmpText2 As String
    Dim ChrPointer As Integer
    'Moving character by character form TmpTxt1 to TmpTxt2
    'unless it is a comma (,). The comma will be replaced by a dot (.).
    TmpText1 = Format(LatLonValue, "0.000000")
    TmpText2 = ""
    For ChrPointer = 1 To Len(TmpText1)
        If Mid(TmpText1, ChrPointer, 1) <> "," Then
            TmpText2 = TmpText2 + Mid(TmpText1, ChrPointer, 1)
        Else
            TmpText2 = TmpText2 + "."
        End If
    Next
    'Returning string to function and ending
    Return TmpText2

End Function

End Class

```

Code for Module M2J.Designer.vb

```

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class M2J
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Try

```

`End Sub``'Required by the Windows Form Designer``Private components As System.ComponentModel.IContainer``'NOTE: The following procedure is required by the Windows Form Designer``'It can be modified using the Windows Form Designer.``'Do not modify it using the code editor.``<System.Diagnostics.DebuggerStepThrough()> _``Private Sub InitializeComponent()``Me.ClearButton = New System.Windows.Forms.Button()``Me.EndButton = New System.Windows.Forms.Button()``Me.DisclaimerButton = New System.Windows.Forms.Button()``Me.BuildBox = New System.Windows.Forms.GroupBox()``Me.DpiBox = New System.Windows.Forms.ComboBox()``Me.DpiLabel = New System.Windows.Forms.Label()``Me.SaveButton = New System.Windows.Forms.Button()``Me.JprFileBoxLabel = New System.Windows.Forms.Label()``Me.JprFileBox = New System.Windows.Forms.TextBox()``Me.BuildButton = New System.Windows.Forms.Button()``Me.ExtractBox = New System.Windows.Forms.GroupBox()``Me.JprDataBoxLabel = New System.Windows.Forms.Label()``Me.JprDataBox = New System.Windows.Forms.TextBox()``Me.ExtractButton = New System.Windows.Forms.Button()``Me.FindBox = New System.Windows.Forms.GroupBox()``Me.MapInfoLabel = New System.Windows.Forms.Label()``Me.MapInfoBox = New System.Windows.Forms.TextBox()``Me.MapFileValue = New System.Windows.Forms.Label()``Me.MapFileLabel = New System.Windows.Forms.Label()``Me.FindButton = New System.Windows.Forms.Button()``Me.BuildBox.SuspendLayout()``Me.ExtractBox.SuspendLayout()``Me.FindBox.SuspendLayout()``Me.SuspendLayout()``'``'ClearButton``'``Me.ClearButton.Location = New System.Drawing.Point(720, 371)``Me.ClearButton.Name = "ClearButton"``Me.ClearButton.Size = New System.Drawing.Size(122, 43)``Me.ClearButton.TabIndex = 6``Me.ClearButton.Text = "Clear &All"``Me.ClearButton.UseVisualStyleBackColor = True``'``'EndButton``'``Me.EndButton.Location = New System.Drawing.Point(1032, 371)``Me.EndButton.Name = "EndButton"``Me.EndButton.Size = New System.Drawing.Size(122, 43)``Me.EndButton.TabIndex = 7`

```
Me.EndButton.Text = "&Close/End program"
Me.EndButton.UseVisualStyleBackColor = True
'
'DisclaimButton
'
Me.DisclaimButton.Location = New System.Drawing.Point(592, 371)
Me.DisclaimButton.Name = "DisclaimButton"
Me.DisclaimButton.Size = New System.Drawing.Size(122, 43)
Me.DisclaimButton.TabIndex = 5
Me.DisclaimButton.Text = "&Disclaimer"
Me.DisclaimButton.UseVisualStyleBackColor = True
'
'BuildBox
'
Me.BuildBox.Controls.Add(Me.DpiBox)
Me.BuildBox.Controls.Add(Me.DpiLabel)
Me.BuildBox.Controls.Add(Me.SaveButton)
Me.BuildBox.Controls.Add(Me.JprFileBoxLabel)
Me.BuildBox.Controls.Add(Me.JprFileBox)
Me.BuildBox.Controls.Add(Me.BuildButton)
Me.BuildBox.Location = New System.Drawing.Point(586, 12)
Me.BuildBox.Name = "BuildBox"
Me.BuildBox.Size = New System.Drawing.Size(568, 353)
Me.BuildBox.TabIndex = 11
Me.BuildBox.TabStop = False
Me.BuildBox.Text = "Build and save JPR file"
'
'DpiBox
'
Me.DpiBox.FormattingEnabled = True
Me.DpiBox.Location = New System.Drawing.Point(134, 41)
Me.DpiBox.Name = "DpiBox"
Me.DpiBox.Size = New System.Drawing.Size(174, 21)
Me.DpiBox.TabIndex = 7
'
'DpiLabel
'
Me.DpiLabel.AutoSize = True
Me.DpiLabel.Location = New System.Drawing.Point(134, 19)
Me.DpiLabel.Name = "DpiLabel"
Me.DpiLabel.Size = New System.Drawing.Size(175, 13)
Me.DpiLabel.TabIndex = 6
Me.DpiLabel.Text = "Resolution in DPI to calculate scale"
'
'SaveButton
'
Me.SaveButton.Location = New System.Drawing.Point(431, 19)
Me.SaveButton.Name = "SaveButton"
Me.SaveButton.Size = New System.Drawing.Size(122, 43)
Me.SaveButton.TabIndex = 4
```

```
Me.SaveButton.Text = "&Save JPR file"
Me.SaveButton.UseVisualStyleBackColor = True
'
'JprFileBoxLabel
'
Me.JprFileBoxLabel.AutoSize = True
Me.JprFileBoxLabel.Location = New System.Drawing.Point(6, 65)
Me.JprFileBoxLabel.Name = "JprFileBoxLabel"
Me.JprFileBoxLabel.Size = New System.Drawing.Size(63, 13)
Me.JprFileBoxLabel.TabIndex = 4
Me.JprFileBoxLabel.Text = "JPR file text"
'
'JprFileBox
'
Me.JprFileBox.Font = New System.Drawing.Font("Consolas", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.JprFileBox.Location = New System.Drawing.Point(6, 83)
Me.JprFileBox.Multiline = True
Me.JprFileBox.Name = "JprFileBox"
Me.JprFileBox.ReadOnly = True
Me.JprFileBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
Me.JprFileBox.Size = New System.Drawing.Size(547, 264)
Me.JprFileBox.TabIndex = 3
'
'BuildButton
'
Me.BuildButton.Location = New System.Drawing.Point(6, 19)
Me.BuildButton.Name = "BuildButton"
Me.BuildButton.Size = New System.Drawing.Size(122, 43)
Me.BuildButton.TabIndex = 3
Me.BuildButton.Text = "&Build JPR file"
Me.BuildButton.UseVisualStyleBackColor = True
'
'ExtractBox
'
Me.ExtractBox.Controls.Add(Me.JprDataBoxLabel)
Me.ExtractBox.Controls.Add(Me.JprDataBox)
Me.ExtractBox.Controls.Add(Me.ExtractButton)
Me.ExtractBox.Location = New System.Drawing.Point(12, 216)
Me.ExtractBox.Name = "ExtractBox"
Me.ExtractBox.Size = New System.Drawing.Size(568, 198)
Me.ExtractBox.TabIndex = 10
Me.ExtractBox.TabStop = False
Me.ExtractBox.Text = "Extract JPR data"
'
'JprDataBoxLabel
'
Me.JprDataBoxLabel.AutoSize = True
Me.JprDataBoxLabel.Location = New System.Drawing.Point(6, 65)
Me.JprDataBoxLabel.Name = "JprDataBoxLabel"
Me.JprDataBoxLabel.Size = New System.Drawing.Size(123, 13)
```

```
Me.JprDataBoxLabel.TabIndex = 4
Me.JprDataBoxLabel.Text = "JPR data and comments"
'
'JprDataBox
'
Me.JprDataBox.Font = New System.Drawing.Font("Consolas", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.JprDataBox.Location = New System.Drawing.Point(6, 83)
Me.JprDataBox.Multiline = True
Me.JprDataBox.Name = "JprDataBox"
Me.JprDataBox.ReadOnly = True
Me.JprDataBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
Me.JprDataBox.Size = New System.Drawing.Size(547, 104)
Me.JprDataBox.TabIndex = 3
'
'ExtractButton
'
Me.ExtractButton.Location = New System.Drawing.Point(6, 19)
Me.ExtractButton.Name = "ExtractButton"
Me.ExtractButton.Size = New System.Drawing.Size(122, 43)
Me.ExtractButton.TabIndex = 2
Me.ExtractButton.Text = "&Extract JPR data"
Me.ExtractButton.UseVisualStyleBackColor = True
'
'FindBox
'
Me.FindBox.Controls.Add(Me.MapInfoLabel)
Me.FindBox.Controls.Add(Me.MapInfoBox)
Me.FindBox.Controls.Add(Me.MapFileValue)
Me.FindBox.Controls.Add(Me.MapFileLabel)
Me.FindBox.Controls.Add(Me.FindButton)
Me.FindBox.Location = New System.Drawing.Point(12, 12)
Me.FindBox.Name = "FindBox"
Me.FindBox.Size = New System.Drawing.Size(568, 198)
Me.FindBox.TabIndex = 9
Me.FindBox.TabStop = False
Me.FindBox.Text = "Find MAP file"
'
'MapInfoLabel
'
Me.MapInfoLabel.AutoSize = True
Me.MapInfoLabel.Location = New System.Drawing.Point(6, 65)
Me.MapInfoLabel.Name = "MapInfoLabel"
Me.MapInfoLabel.Size = New System.Drawing.Size(85, 13)
Me.MapInfoLabel.TabIndex = 4
Me.MapInfoLabel.Text = "MAP file content"
'
'MapInfoBox
'
Me.MapInfoBox.Font = New System.Drawing.Font("Consolas", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.MapInfoBox.Location = New System.Drawing.Point(6, 83)
```

```
Me.MapInfoBox.Multiline = True
Me.MapInfoBox.Name = "MapInfoBox"
Me.MapInfoBox.ReadOnly = True
Me.MapInfoBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
Me.MapInfoBox.Size = New System.Drawing.Size(547, 104)
Me.MapInfoBox.TabIndex = 3
'
'MapFileValue
'
Me.MapFileValue.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.MapFileValue.Location = New System.Drawing.Point(137, 47)
Me.MapFileValue.Name = "MapFileValue"
Me.MapFileValue.Size = New System.Drawing.Size(416, 15)
Me.MapFileValue.TabIndex = 2
'
'MapFileLabel
'
Me.MapFileLabel.AutoSize = True
Me.MapFileLabel.Location = New System.Drawing.Point(134, 19)
Me.MapFileLabel.Name = "MapFileLabel"
Me.MapFileLabel.Size = New System.Drawing.Size(86, 13)
Me.MapFileLabel.TabIndex = 1
Me.MapFileLabel.Text = "MAP file location"
'
'FindButton
'
Me.FindButton.Location = New System.Drawing.Point(6, 19)
Me.FindButton.Name = "FindButton"
Me.FindButton.Size = New System.Drawing.Size(122, 43)
Me.FindButton.TabIndex = 1
Me.FindButton.Text = "&Find MAP file"
Me.FindButton.UseVisualStyleBackColor = True
'
'M2J
'
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.ClientSize = New System.Drawing.Size(1160, 423)
Me.Controls.Add(Me.ClearButton)
Me.Controls.Add(Me.EndButton)
Me.Controls.Add(Me.DisclaimButton)
Me.Controls.Add(Me.BuildBox)
Me.Controls.Add(Me.ExtractBox)
Me.Controls.Add(Me.FindBox)
Me.Name = "M2J"
Me.Text = "MAP data for JPR"
Me.BuildBox.ResumeLayout(False)
Me.BuildBox.PerformLayout()
Me.ExtractBox.ResumeLayout(False)
Me.ExtractBox.PerformLayout()
```

```
Me.FindBox.ResumeLayout(False)
Me.FindBox.PerformLayout()
Me.ResumeLayout(False)
```

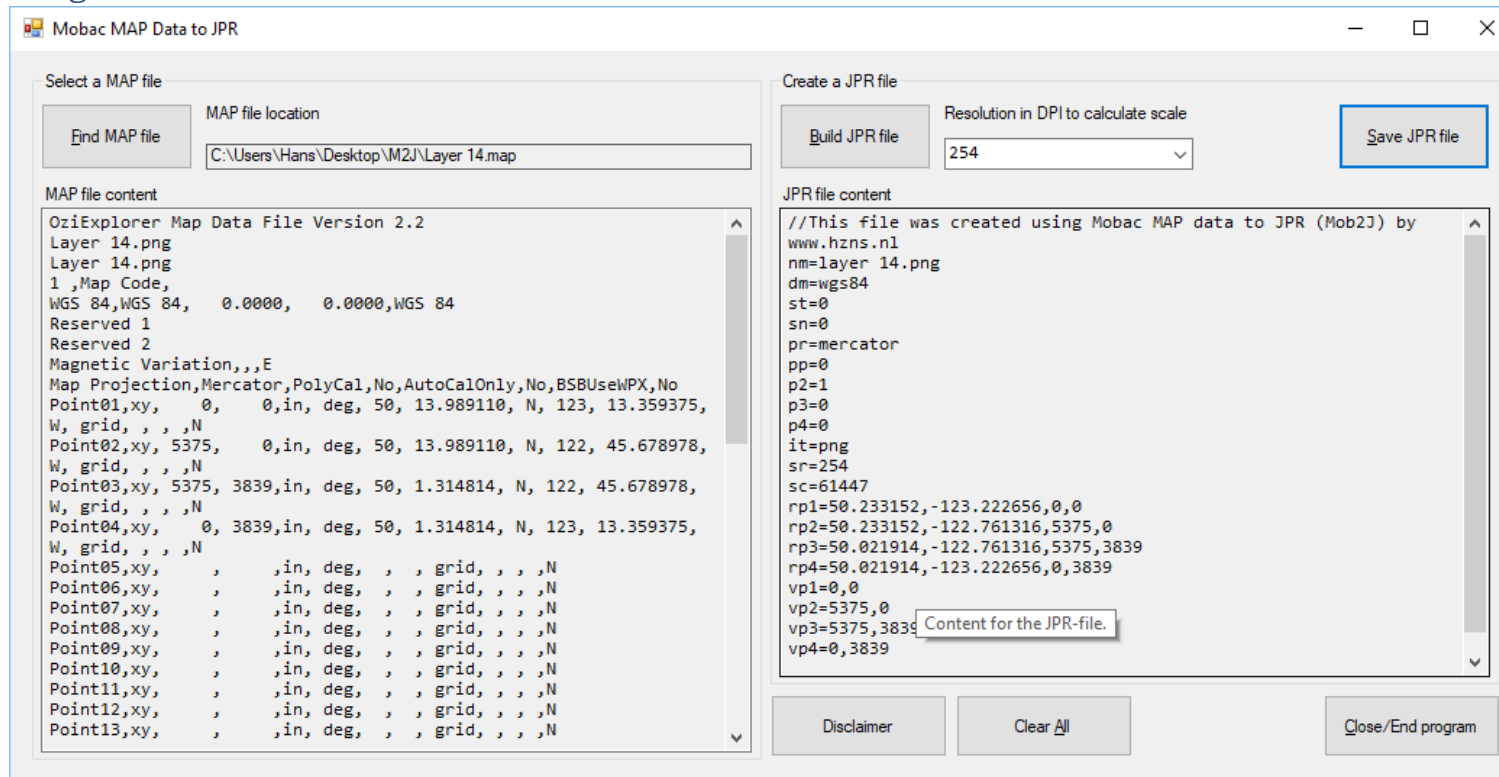
```
End Sub
```

```
Friend WithEvents ClearButton As Button
Friend WithEvents EndButton As Button
Friend WithEvents DisclaimButton As Button
Friend WithEvents BuildBox As GroupBox
Friend WithEvents DpiBox As ComboBox
Friend WithEvents DpiLabel As Label
Friend WithEvents SaveButton As Button
Friend WithEvents JprFileBoxLabel As Label
Friend WithEvents JprFileBox As TextBox
Friend WithEvents BuildButton As Button
Friend WithEvents ExtractBox As GroupBox
Friend WithEvents JprDataBoxLabel As Label
Friend WithEvents JprDataBox As TextBox
Friend WithEvents ExtractButton As Button
Friend WithEvents FindBox As GroupBox
Friend WithEvents MapInfoLabel As Label
Friend WithEvents MapInfoBox As TextBox
Friend WithEvents MapFileValue As Label
Friend WithEvents MapFileLabel As Label
Friend WithEvents FindButton As Button
```

```
End Class
```

Mobac MAP Data to JPR (Mob2J)

Image of Mob2J



Code for Module Mob2J.vb

Public Class Mob2J

'Structure declarations

Structure JprStructure

Public nm As String

Public sc As Double

Public sr As Double

Public ds As Double

Public it As String

Public dm As String

Public st As Double

Public sn As Double

Public pr As String

Public pp As String

'Necessary JPR data (in the context of this application)

'Name of chart

'Scale

'Resolution

'Size of a pixel in meter

'Type of image

'Datum

'Datum shift (latitude)

'Datum shift (longitude)

'Projection

'Central meridian

```

Public p2 As String      'Scale Factor
Public p3 As String      'False Northing
Public p4 As String      'False Easting
End Structure

Structure Coordinate      'Referenced Pixel and Geographical coordinates
Public x As String        'Pixel X
Public y As String        'Pixel Y
Public lat As String      'Latitude
Public lon As String      'Longitude
End Structure

'Variable declarations
Public MapFilePath As String      'Filename and path to MAP-file
Public MapFile As String          'Filename MAP-file
Public MapPath As String          'Path to MAP-file
Public JprFilePath As String      'Filename and path to JPR-file.
Public ChartName As String        'MAP-file name without extension (ChartName was used to prevent confusion with the use of map-file)
Public MapFileContent As String   'Content of .map file
Public LineCounter As Integer     'Amount of lines in MapFileContent (will be (re)calculated for every map)
Public LineContent(60) As String  'Content of each line in MapFileContent (will be (re)dimensioned for every map)
Public LineField(4) As String     'Content of a field in a line (used to extract coordinates)
Public JprData As JprStructure    'Variable representing the JPR data
Public ReferencePoint(4) As Coordinate 'Reference points for calibration and vertex polygon (topleft, topright, bottomright, bottomleft)

Public EndLine As String = Chr(13) + Chr(10) 'Variable representing the end a line of text.

'subroutine Mob2J_Load will be executed starting the application
'- It enables and sets the Tooltips for buttons and boxes.
'- Loading DpiBox with default values
'- It enables/disables buttons
Private Sub Mob2J_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'Declarations
    Dim TipText As New ToolTip()

    'Setting up the delays for the ToolTip (TipText).
    TipText.AutoPopDelay = 5000
    TipText.InitialDelay = 1000
    TipText.ReshowDelay = 500

    'Forcing the ToolTip text to be displayed whether or not the form is active.
    TipText.ShowAlways = True

    ' Setting up the TipText text
    TipText.SetToolTip(Me.FindButton, "Find and select a MAP-file.")
    TipText.SetToolTip(Me.MapInfoBox, "Content of the selected MAP-file.")
    TipText.SetToolTip(Me.BuildButton, "Build the content for the JPR-file.")
    TipText.SetToolTip(Me.DpiBox, "Select or change the resolution of the chart, 254 dpi (100 dot per cm) is the default.")
    TipText.SetToolTip(Me.JprFileBox, "Content for the JPR-file.")
    TipText.SetToolTip(Me.SaveButton, "Save the content for JPR-file in the same directory as the MAP-file. If the JPR-file exists, you will be prompted to over write.")

```

```

TipText.SetToolTip(Me.DisclaimButton, "Disclaimer for the application Mobac MAP data to JPR (M2J).")
TipText.SetToolTip(Me.ClearButton, "Clear all data with in the applicatation, except the selected resolution.")
TipText.SetToolTip(Me.EndButton, "Close the application.")

'Loading DpiBox
DpiBox.Items.Add("72")
DpiBox.Items.Add("96")
DpiBox.Items.Add("100")
DpiBox.Items.Add("127")
DpiBox.Items.Add("150")
DpiBox.Items.Add("200")
DpiBox.Items.Add("254")
DpiBox.Items.Add("300")
DpiBox.Items.Add("508")
'Default value for DpiBox
DpiBox.Text = 254

'Initial setting user interface (buttons)
BuildButton.Enabled = False
SaveButton.Enabled = False
DpiBox.Enabled = False

End Sub

'Subroutine FindButton_Click
'- Finding/Selecting and opening a MAP file
'- Creating values for variables: MapFile, MapPath, JprFileName, MapFileContent
'- Presenting MapFileContent on the screen
'- Loading MapFileContent of MapFile into an array (LineContent)
Private Sub FindButton_Click(sender As Object, e As EventArgs) Handles FindButton.Click
'Declarations
Dim FileNameLen, Pointer As Integer
Dim TempFileName As String
Dim MapFileContentLen, ChrPointer, LinePointer As Integer
Dim TxtChr As String = ""
Dim FindFile As New OpenFileDialog

'Clearing all data on screen and variables
Call ClearButton_Click(sender, e)

'Setting up dialog
FindFile.Filter = "Ozi-MAP file (*.map) | *.map"
FindFile.Multiselect = False
'Opening dialog
FindFile.ShowDialog()

'Retrieving the value for MAPfile
TempFileName = FindFile.FileName

'Stopping this subroutine if no file was selected (preventing an error)

```

```

'(length TempFileName = 0)
If Len(TempFileName) = 0 Then
    MsgBox("No file was selected!", 48, "Warning")
    Exit Sub
End If

'Setting values for variables
MapFilePath = TempFileName
MapFileValue.Text = TempFileName

'Retrieving the value for MapFile and MapPath
FileNameLen = Len(TempFileName)
For Pointer = FileNameLen To 0 Step -1
    If Mid(TempFileName, Pointer, 1) = "\" Then
        MapPath = Mid(TempFileName, 1, Pointer)
        MapFile = Mid(TempFileName, Pointer + 1, FileNameLen - Pointer)
        Exit For
    End If
Next Pointer

'Retrieving the value for ChartName and JprFileName
FileNameLen = Len(MapFile)
For Pointer = FileNameLen To 0 Step -1
    If Mid(MapFile, Pointer, 1) = "." Then
        ChartName = Mid(MapFile, 1, Pointer - 1)
        JprFilePath = MapPath & ChartName & ".jpr"
        Exit For
    End If
Next Pointer

'Retreiving content of the MapFile
MapFileContent = My.Computer.FileSystem.ReadAllText(MapPath + MapFile)

>Loading MapFileContent into MapInfoBox
MapInfoBox.Text = MapFileContent

'Breaking MapFileContent into Lines (in array LineContent)
'- Determening amount of lines (LineCounter)
MapFileContentLen = Len(MapFileContent)
LineCounter = 1
For I = 1 To MapFileContentLen
    If Mid(MapFileContent, I, 1) = Chr(13) Then
        LineCounter = LineCounter + 1
    End If
Next
'- Retrieving the lines and put them in the array LineContent
ReDim LineContent(LineCounter)
ChrPointer = 1
LinePointer = 1
Do While LinePointer < LineCounter And ChrPointer < MapFileContentLen

```

```

    Do
        TxtChr = Mid(MapFileContent, ChrPointer, 1)
        If TxtChr <> Chr(13) And TxtChr <> Chr(10) Then
            LineContent(LinePointer) = LineContent(LinePointer) + TxtChr
        End If
        ChrPointer = ChrPointer + 1
    Loop Until TxtChr = Chr(13) Or TxtChr = Chr(10)
    If TxtChr = Chr(13) Then LinePointer = LinePointer + 1
Loop

'Enabling BuildButton (to build the content of JPR file)
BuildButton.Enabled = True
BuildButton.Select()

End Sub

'Subroutine BuildButton_Click
'- Retrieving all data for JPR File
'- Writing all data to JprFileBox in the correct structure
Private Sub BuildButton_Click(sender As Object, e As EventArgs) Handles BuildButton.Click
    'Retrieving calibrationpoints (ReferencePoint) and Size of a pixel in meter (JprData.ds)
    For LinePointer = 41 To LineCounter
        Call SplitLine(LineContent(LinePointer))
        Select Case LCase(LineField(1))
            Case "mmp1l"
                'latitude and longitude
                ReferencePoint(Val(LineField(2))).lon = Trim(LineField(3))
                ReferencePoint(Val(LineField(2))).lat = Trim(LineField(4))
            Case "mmpxy"
                'pixel X and Y
                ReferencePoint(Val(LineField(2))).x = Trim(LineField(3))
                ReferencePoint(Val(LineField(2))).y = Trim(LineField(4))
            Case "mm1b"
                'Size of pixel in real in meter
                JprData.ds = Val(LineField(2))
        End Select
    Next

    'Retrieving the data for variable JprData
    JprData.nm = LCase(Trim(LineContent(2)))
    JprData.dm = "wgs84"
    JprData.st = 0
    JprData.sn = 0
    JprData.pr = "mercator"
    JprData.pp = "0"
    JprData.p2 = "1"
    JprData.p3 = "0"
    JprData.p4 = "0"
    JprData.it = "png"
    JprData.sr = Val(DpiBox.Text)
    JprData.sc = Int((JprData.sr / 2.54) * 100 * JprData.ds)

    'Writing the data to JprFileBox

```

```

JprFileBox.Text = "//This file was created using Mobac MAP data to JPR (Mob2J) by www.hzns.nl" & EndLine &
"nm=" & JprData.nm & EndLine &
"dm=" & JprData.dm & EndLine &
"st=" & JprData.st & EndLine &
"sn=" & JprData.sn & EndLine &
"pr=" & JprData.pr & EndLine &
"pp=" & JprData.pp & EndLine &
"p2=" & JprData.p2 & EndLine &
"p3=" & JprData.p3 & EndLine &
"p4=" & JprData.p4 & EndLine &
"it=" & JprData.it & EndLine &
"sr=" & JprData.sr & EndLine &
"sc=" & JprData.sc & EndLine &
"rp1=" & ReferencePoint(1).lat & "," & ReferencePoint(1).lon & "," & ReferencePoint(1).x & "," & ReferencePoint(1).y & EndLine &
"rp2=" & ReferencePoint(2).lat & "," & ReferencePoint(2).lon & "," & ReferencePoint(2).x & "," & ReferencePoint(2).y & EndLine &
"rp3=" & ReferencePoint(3).lat & "," & ReferencePoint(3).lon & "," & ReferencePoint(3).x & "," & ReferencePoint(3).y & EndLine &
"rp4=" & ReferencePoint(4).lat & "," & ReferencePoint(4).lon & "," & ReferencePoint(4).x & "," & ReferencePoint(4).y & EndLine &
"vp1=" & ReferencePoint(1).x & "," & ReferencePoint(1).y & EndLine &
"vp2=" & ReferencePoint(2).x & "," & ReferencePoint(2).y & EndLine &
"vp3=" & ReferencePoint(3).x & "," & ReferencePoint(3).y & EndLine &
"vp4=" & ReferencePoint(4).x & "," & ReferencePoint(4).y & EndLine

'Enabling SaveButton (to save the content of JPR file)
DpiBox.Enabled = True
SaveButton.Enabled = True
SaveButton.Select()
End Sub

'Subroutine SaveButton_Click
'- Saving the JprFileContentBox.text to a JPR-file in the same directory
' as the original MAP file. If this JPR file exists a warning will be prompted.
Private Sub SaveButton_Click(sender As Object, e As EventArgs) Handles SaveButton.Click
'Declarations
Dim Answer As Integer 'Used as answer value for MsgBox

'Checking for an existing file
If My.Computer.FileSystem.FileExists(JprFilePath) = True Then
    Answer = MsgBox("JPR-file already exists. Overwrite?", 292, "Warning")
    If Answer = 6 Then
        'Overwriting existing JPR-file if answer on MsgBox = Yes (Answer = 6)
        My.Computer.FileSystem.WriteAllText(JprFilePath, JprFileBox.Text, False)
    End If
Else
    'Creating JPR-file
    My.Computer.FileSystem.WriteAllText(JprFilePath, JprFileBox.Text, False)
End If

End Sub

'Subroutine DisclaimerButton_Click

```

```

'- Showing the disclaimer
Private Sub DisclaimerButton_Click(sender As Object, e As EventArgs) Handles DisclaimButton.Click
    'Declarations
    Dim MsgText As String

    'Disclaimer message
    MsgText = "This application Is provided 'As is'. The use of the application is on your own risk. " &
        "Direct or indirect damage by using this application is users responsibility, " &
        "not the application-builders. Redistribution by a third party (commercial of non-commercial) " &
        "is prohibit. Please download the application direct from www.hzns.nl."
    MsgBox(MsgText, 48, "Disclaimer")

End Sub

'Subroutine ClearButton_Click
'- Clearing all data from screen
'- disabling BuildButton and SaveButton
Private Sub ClearButton_Click(sender As Object, e As EventArgs) Handles ClearButton.Click
    'Declarations
    Dim Pointer As Integer

    'Clearing textboxes and labels
    MapInfoBox.Text = ""
    JprFileBox.Text = ""
    MapFileValue.Text = ""

    'Disabeling buttons
    SaveButton.Enabled = False
    BuildButton.Enabled = False
    DpiBox.Enabled = False

    'Clearing MAP file Data
    MapFile = ""
    MapPath = ""
    MapFilePath = ""
    ChartName = ""
    JprFilePath = ""
    MapFileContent = ""
    For Pointer = 0 To LineCounter
        LineContent(Pointer) = ""
    Next

End Sub

'Suboutine EndButton_Click
'- Closing application
Private Sub EndButton_Click(sender As Object, e As EventArgs) Handles EndButton.Click
    'Unloading and closing application
    Me.Close()

```

```

End Sub

'Subroutine SplitLine
'- Splitting a line With comma separeted fields into 4 fields
' (used for calibration data)
Private Sub SplitLine(LineText As String)
'Declarations
Dim FieldPointer, ChrPointer As Integer
Dim TxtChr As String

'Clearing all information Array LineField
For FieldPointer = 0 To 4
    LineField(FieldPointer) = ""
Next
'Splitting line into fields
FieldPointer = 1
ChrPointer = 1
Do While ChrPointer <= Len(LineText)
    Do
        TxtChr = Mid(LineText, ChrPointer, 1)
        If TxtChr <> Chr(44) Then
            LineField(FieldPointer) = LineField(FieldPointer) + TxtChr
        End If
        ChrPointer = ChrPointer + 1
    Loop Until TxtChr = Chr(44) Or ChrPointer > Len(LineText)
    If TxtChr = Chr(44) Then FieldPointer = FieldPointer + 1
    Loop
End Sub

'Subroutine DipBox_LostFocus handles the cases LostFocus and SelectedValueChanged (with focus on DipBox)
'- recalculating the map scale
Private Sub DipBox_LostFocus(sender As Object, e As EventArgs) Handles DipBox.LostFocus, DipBox.SelectedValueChanged
'(only if JprFileBox contains information)
If JprFileBox.Text <> "" Then
    If JprData.sr <> Val(DipBox.Text) Then
        Call BuildButton_Click(sender, e) '(learned by trail and error)
        SaveButton.Select()
    End If
End If

End Sub

'Subroutine DipBox_PressKey handles the case "Enter"-key was used (with focus on DipBox)
'- recalculating the map scale
Private Sub DipBox_KeyPress(sender As Object, e As KeyPressEventArgs) Handles DipBox.KeyPress
If JprFileBox.Text <> "" And e.KeyChar = Chr(13) Then
    If JprData.sr <> Val(DipBox.Text) Then
        JprData.sr = Val(DipBox.Text)
        Call BuildButton_Click(sender, e) '(learned by trail and error)
    End If
End Sub

```

```

End If

End Sub

End Class

```

Code for Module Mob2J.Designer.vb

```

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Mob2J
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.SelectBox = New System.Windows.Forms.GroupBox()
        Me.MapInfoBox = New System.Windows.Forms.TextBox()
        Me.MapFileContentLabel = New System.Windows.Forms.Label()
        Me.FindButton = New System.Windows.Forms.Button()
        Me.CreateBox = New System.Windows.Forms.GroupBox()
        Me.DpiBox = New System.Windows.Forms.ComboBox()
        Me.DpiLabel = New System.Windows.Forms.Label()
        Me.SaveButton = New System.Windows.Forms.Button()
        Me.JprFileBox = New System.Windows.Forms.TextBox()
        Me.JprFileBoxLabel = New System.Windows.Forms.Label()
        Me.BuildButton = New System.Windows.Forms.Button()
        Me.EndButton = New System.Windows.Forms.Button()
        Me.ClearButton = New System.Windows.Forms.Button()
        Me.DisclaimButton = New System.Windows.Forms.Button()
        Me.MapFileValue = New System.Windows.Forms.Label()
        Me.MapFileLabel = New System.Windows.Forms.Label()
        Me.SelectBox.SuspendLayout()
        Me.CreateBox.SuspendLayout()
    End Sub

```



```
Me.SuspendLayout()
'
'SelectBox
'
Me.SelectBox.Controls.Add(Me.MapFileValue)
Me.SelectBox.Controls.Add(Me.MapFileLabel)
Me.SelectBox.Controls.Add(Me.MapInfoBox)
Me.SelectBox.Controls.Add(Me.MapFileContentLabel)
Me.SelectBox.Controls.Add(Me.FindButton)
Me.SelectBox.Location = New System.Drawing.Point(15, 14)
Me.SelectBox.Name = "SelectBox"
Me.SelectBox.Size = New System.Drawing.Size(504, 476)
Me.SelectBox.TabIndex = 0
Me.SelectBox.TabStop = False
Me.SelectBox.Text = "Select a MAP file"
'
'MapInfoBox
'
Me.MapInfoBox.BackColor = System.Drawing.SystemColors.ButtonFace
Me.MapInfoBox.Font = New System.Drawing.Font("Consolas", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.MapInfoBox.Location = New System.Drawing.Point(6, 94)
Me.MapInfoBox.Multiline = True
Me.MapInfoBox.Name = "MapInfoBox"
Me.MapInfoBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
Me.MapInfoBox.Size = New System.Drawing.Size(491, 376)
Me.MapInfoBox.TabIndex = 1
'
'MapFileContentLabel
'
Me.MapFileContentLabel.AutoSize = True
Me.MapFileContentLabel.Location = New System.Drawing.Point(6, 78)
Me.MapFileContentLabel.Name = "MapFileContentLabel"
Me.MapFileContentLabel.Size = New System.Drawing.Size(85, 13)
Me.MapFileContentLabel.TabIndex = 5
Me.MapFileContentLabel.Text = "MAP file content"
'
'FindButton
'
Me.FindButton.Location = New System.Drawing.Point(6, 22)
Me.FindButton.Name = "FindButton"
Me.FindButton.Size = New System.Drawing.Size(105, 46)
Me.FindButton.TabIndex = 0
Me.FindButton.Text = "&Find MAP file"
Me.FindButton.UseVisualStyleBackColor = True
'
'CreateBox
'
Me.CreateBox.Controls.Add(Me.DpiBox)
Me.CreateBox.Controls.Add(Me.DpiLabel)
Me.CreateBox.Controls.Add(Me.SaveButton)
```

```

Me.CreateBox.Controls.Add(Me.JprFileBox)
Me.CreateBox.Controls.Add(Me.JprFileBoxLabel)
Me.CreateBox.Controls.Add(Me.BuildButton)
Me.CreateBox.Location = New System.Drawing.Point(525, 14)
Me.CreateBox.Name = "CreateBox"
Me.CreateBox.Size = New System.Drawing.Size(504, 424)
Me.CreateBox.TabIndex = 4
Me.CreateBox.TabStop = False
Me.CreateBox.Text = "Create a JPR file"
'
'DpiBox
Me.DpiBox.Font = New System.Drawing.Font("Consolas", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.DpiBox.FormattingEnabled = True
Me.DpiBox.Location = New System.Drawing.Point(120, 46)
Me.DpiBox.Name = "DpiBox"
Me.DpiBox.Size = New System.Drawing.Size(172, 22)
Me.DpiBox.TabIndex = 3
'
'DpiLabel
Me.DpiLabel.AutoSize = True
Me.DpiLabel.Location = New System.Drawing.Point(117, 22)
Me.DpiLabel.Name = "DpiLabel"
Me.DpiLabel.Size = New System.Drawing.Size(175, 13)
Me.DpiLabel.TabIndex = 8
Me.DpiLabel.Text = "Resolution in DPI to calculate scale"
'
'SaveButton
Me.SaveButton.Location = New System.Drawing.Point(392, 22)
Me.SaveButton.Name = "SaveButton"
Me.SaveButton.Size = New System.Drawing.Size(105, 46)
Me.SaveButton.TabIndex = 4
Me.SaveButton.Text = "&Save JPR file"
Me.SaveButton.UseVisualStyleBackColor = True
'
'JprFileBox
Me.JprFileBox.BackColor = System.Drawing.SystemColors.ButtonFace
Me.JprFileBox.Font = New System.Drawing.Font("Consolas", 9.0!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.JprFileBox.Location = New System.Drawing.Point(6, 94)
Me.JprFileBox.Multiline = True
Me.JprFileBox.Name = "JprFileBox"
Me.JprFileBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
Me.JprFileBox.Size = New System.Drawing.Size(491, 324)
Me.JprFileBox.TabIndex = 5
'
'JprFileBoxLabel

```

```
Me.JprFileBoxLabel.AutoSize = True
Me.JprFileBoxLabel.Location = New System.Drawing.Point(6, 78)
Me.JprFileBoxLabel.Name = "JprFileBoxLabel"
Me.JprFileBoxLabel.Size = New System.Drawing.Size(82, 13)
Me.JprFileBoxLabel.TabIndex = 5
Me.JprFileBoxLabel.Text = "JPR file content"
'
'BuildButton
'
Me.BuildButton.Location = New System.Drawing.Point(6, 22)
Me.BuildButton.Name = "BuildButton"
Me.BuildButton.Size = New System.Drawing.Size(105, 46)
Me.BuildButton.TabIndex = 2
Me.BuildButton.Text = "&Build JPR file"
Me.BuildButton.UseVisualStyleBackColor = True
'
'EndButton
'
Me.EndButton.Location = New System.Drawing.Point(907, 444)
Me.EndButton.Name = "EndButton"
Me.EndButton.Size = New System.Drawing.Size(122, 43)
Me.EndButton.TabIndex = 8
Me.EndButton.Text = "&Close/End program"
Me.EndButton.UseVisualStyleBackColor = True
'
'ClearButton
'
Me.ClearButton.Location = New System.Drawing.Point(653, 444)
Me.ClearButton.Name = "ClearButton"
Me.ClearButton.Size = New System.Drawing.Size(122, 43)
Me.ClearButton.TabIndex = 7
Me.ClearButton.Text = "Clear &All"
Me.ClearButton.UseVisualStyleBackColor = True
'
'DisclaimButton
'
Me.DisclaimButton.Location = New System.Drawing.Point(525, 444)
Me.DisclaimButton.Name = "DisclaimButton"
Me.DisclaimButton.Size = New System.Drawing.Size(122, 43)
Me.DisclaimButton.TabIndex = 6
Me.DisclaimButton.Text = "Disclaimer"
Me.DisclaimButton.UseVisualStyleBackColor = True
'
'MapFileValue
'
Me.MapFileValue.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.MapFileValue.Location = New System.Drawing.Point(120, 50)
Me.MapFileValue.Name = "MapFileValue"
Me.MapFileValue.Size = New System.Drawing.Size(377, 18)
Me.MapFileValue.TabIndex = 7
```

```

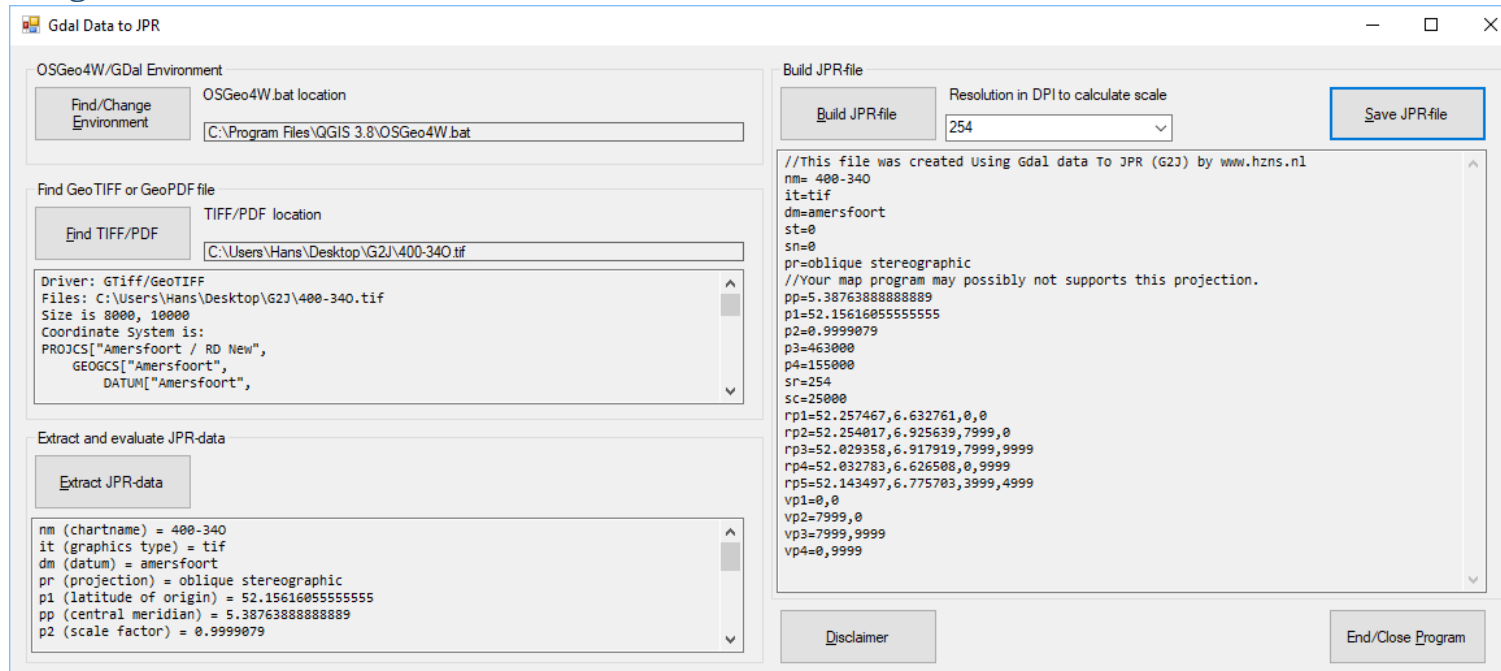
    'MapFileLabel
    '
    Me.MapFileLabel.AutoSize = True
    Me.MapFileLabel.Location = New System.Drawing.Point(117, 22)
    Me.MapFileLabel.Name = "MapFileLabel"
    Me.MapFileLabel.Size = New System.Drawing.Size(86, 13)
    Me.MapFileLabel.TabIndex = 6
    Me.MapFileLabel.Text = "MAP file location"
    '
    'Mob2J
    '
    Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
    Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
    Me.ClientSize = New System.Drawing.Size(1043, 506)
    Me.Controls.Add(Me.DisclaimButton)
    Me.Controls.Add(Me.ClearButton)
    Me.Controls.Add(Me.EndButton)
    Me.Controls.Add(Me.CreateBox)
    Me.Controls.Add(Me.SelectBox)
    Me.Name = "Mob2J"
    Me.Text = "Mobac MAP Data to JPR"
    Me.SelectBox.ResumeLayout(False)
    Me.SelectBox.PerformLayout()
    Me.CreateBox.ResumeLayout(False)
    Me.CreateBox.PerformLayout()
    Me.ResumeLayout(False)

End Sub

Friend WithEvents SelectBox As GroupBox
Friend WithEvents MapInfoBox As TextBox
Friend WithEvents MapFileContentLabel As Label
Friend WithEvents FindButton As Button
Friend WithEvents CreateBox As GroupBox
Friend WithEvents JprFileBox As TextBox
Friend WithEvents JprFileBoxLabel As Label
Friend WithEvents BuildButton As Button
Friend WithEvents SaveButton As Button
Friend WithEvents EndButton As Button
Friend WithEvents ClearButton As Button
Friend WithEvents DisclaimButton As Button
Friend WithEvents DpiBox As ComboBox
Friend WithEvents DpiLabel As Label
Friend WithEvents MapFileValue As Label
Friend WithEvents MapFileLabel As Label
End Class

```

Image of G2J



Code for Module G2J.vb

```
Public Class G2J
    'Structures declarations
    Structure JprStructure
        Public nm As String
        Public sc As Double
        Public sr As Double
        Public it As String
        Public sd As Double
        Public dm As String
        Public st As Double
        Public sn As Double
        Public pr As String
        Public zn As String
        Public pp As String
        Public p1 As String
        Public p2 As String
    End Structure

    'Necessary JPR data (in the context of this application)
    'Name of the Chart
    'Scale of the Chart
    'Resolution of the Chart
    'Type of image of the Chart
    'Size of pixel in meters (no a formal JPF variable)
    'Geographic datum of the Chart
    'Datum shift (latitude)
    'Datum shift (longitude)
    'Projection of the Chart
    'UTM zone of the Chart
    'Central meridian of the Chart
    'Latitude of the origin of the Chart
    'Scalefactor of the Chart
End Class
```

```

Public p3 As String      'False Easting of the Chart
Public p4 As String      'Fales Northing of the Chart
Public p5 As String      'First standard parallel of the Chart
Public p6 As String      'Second standard parallel of the Chart
End Structure

Structure Coordinate      'Referenced Pixel and Geographical coordinates
Public x As String        'Pixel X
Public y As String        'Pixel Y
Public lat As String      'Latitude
Public lon As String      'Longitude
Public Valid As Integer   'Application internal variable used for validation
End Structure

Structure GdalStructure   'Gdal data about Resolution
Public ResolutionX As Single 'Horizontal resolution
Public ResolutionY As Single 'Vertical resolution
Public ResolutionUnit As Integer 'Unit used for resolution (2=DPI/3=Dot per cm)
Public PixelSizeX As Double 'Horizontal size of a pixel in reality (in meter)
Public PixelSizeY As Double 'Vertical size of a pixel in reality (in meter)
End Structure

'Variable declarations
Public IniFilePath As String      'Filename and path to 'g2j.ini'.
Public IniContent As String       'Content of 'g2j.ini'.
Public BatFilePath As String      'Filename and path to 'OSGeo4W.bat'.
Public ChartFilePath As String    'Filename and path to TIFF- or PDF-file.
Public ChartFile As String        'Filename TIFF- or PDF-file.
Public ChartPath As String        'Path to TIFF- or PDF-file.
Public ChartName As String        'Filename TIFF- or PDF-file without extention.
Public JprFilePath As String      'Filename and path to to JPR-file.
Public GdalContent As String      'Geo-Information retrieved with GdalInfo
Public TempFilePath As String     'Filename and path to a temporary file with the Gdal information
Public JprData As JprStructure    'Variable representing the JPR data
Public ReferencePoint(5) As Coordinate 'Calibration points (topleft, topright, bottomright, bottomleft, center)
Public VertexPoint(4) As Coordinate 'Vertexpoints (topleft, topright, bottomright, bottomleft)
Public ImageWidth As Integer      'Width of the image
Public ImageHeight As Integer     'Height of teh image
Public GdalData As GdalStructure  'Variable representing the neccesary Gdal data
Public ResolutionScaleCode As Integer 'This ariable is a 'returncode' when handling Resoution and Scale

Public EndLine As String = Chr(13) + Chr(10) 'Variable representing the end a line of text.

'Subroutine G2J_Load
'This sub is will be executed starting the application
'- It enables and sets the Tooltips for buttons and boxes.
'- It enables/disables buttons
'- It calls sub 'CheckGdalPath' (to load 'BatFilePath').
Private Sub G2J_Load(sender As Object, e As EventArgs) Handles MyBase.Load
'Declarations

```

```

Dim TipText As New ToolTip()

'Setting up the delays for the ToolTip (TipText).
TipText.AutoPopDelay = 5000
TipText.InitialDelay = 1000
TipText.ReshowDelay = 500

'Forcing the ToolTip text to be displayed whether or not the form is active.
TipText.ShowAlways = True

TipText.SetToolTip(Me.OSButton, "Find or change the batch file to start the Gdal environment (OSGeo4W.bat)")
TipText.SetToolTip(Me.BatFileValue, "The location of the batchfile to start the Gdal environment (OSGeo4W.bat)")
TipText.SetToolTip(Me.FindButton, "Find or change the PDF/TIF-file and load the content")
TipText.SetToolTip(Me.ChartFileValue, "The location of the the PDF/TIF-file")
TipText.SetToolTip(Me.GdalInfoBox, "The geographic and image parameters of the the PDF/TIF-file, extracted with Gdalinfo")
TipText.SetToolTip(Me.ExtractButton, "Extract the JPR data from the data provided by Gdalinfo")
TipText.SetToolTip(Me.JprDataBox, "The extracted JPR data from the data provided by Gdalinfo")
TipText.SetToolTip(Me.BuildButton, "Create the content of the JPR-file")
TipText.SetToolTip(Me.SaveJprFileButton, "Save the content of the JPR-file (in the same directory as the PDF/TIF-file)")
TipText.SetToolTip(Me.JprFileBox, "The content of the JPR-file")
TipText.SetToolTip(Me.DisclaimButton, "Show the disclaimer")
TipText.SetToolTip(Me.EndButton, "End en close the application")

'Loading DpiBox
DpiBox.Items.Add("72")
DpiBox.Items.Add("96")
DpiBox.Items.Add("100")
DpiBox.Items.Add("127")
DpiBox.Items.Add("150")
DpiBox.Items.Add("200")
DpiBox.Items.Add("254")
DpiBox.Items.Add("300")
DpiBox.Items.Add("508")

'Initial setting user interface (buttons)
FindButton.Enabled = False
ExtractButton.Enabled = False
BuildButton.Enabled = False
SaveJprFileButton.Enabled = False
DpiBox.Enabled = False

'Initiating the check applications variables
Call CheckGdalPath()

End Sub

'Subroutine CheckGdalPath
'- 'CheckGdalPath' will first look for the 'g2j.ini' file in documents directory of user.
'- If so, it will check the content of 'g2j.ini'. the value of 'BatFilePath' must be a exiting file 'OSGeo4W.bat'.
' OSGeo4W.bat starts the OSGeo4W shell. This shell is needed to execute gdalinfo.exe.

```

```

' If found this information Is showed In the application window('BatFileValue'). and the button 'FindButton' will be enabled.
'- If ini-file is missing or the content of BatFilePath is incorrect, the user will be prompted to took
' For the location of 'OSGeo4W.bat' (using the sub 'FindChangeOS').
Private Sub CheckGdalPath()
    'Declarations
    Dim Answer As Integer

    'Looking of j2g.ini
    ' (My.Computer.FileSystem.SpecialDirectories.MyDocuments points to users 'My documents' directory
    IniFilePath = My.Computer.FileSystem.SpecialDirectories.MyDocuments & "\g2j.ini"
    If My.Computer.FileSystem.FileExists(IniFilePath) = True Then
        IniContent = My.Computer.FileSystem.ReadAllText(IniFilePath)
        'ini-data is only one line; text will be split in two parts (part(0) is "BatFilePath" and the part(1) is the value for GdalFilePath)
        'Before splitting there will be a check if the separating chr (=) exist
        Answer = InStr(IniContent, "=", CompareMethod.Binary)
        If Answer > 0 Then
            Dim Part() As String = Split(IniContent, "=",,)
            If My.Computer.FileSystem.FileExists(Part(1)) = True Then
                BatFilePath = Part(1)
                BatFileValue.Text = BatFilePath
                'Enabling next step of application OSGeo4W.bat
                FindButton.Enabled = True
                FindButton.Select()
            Exit Sub
        End If
    End If
End If

'Initiating the procedure to find 'OSGeo4W.bat'. This batch file is necessary to open the OSGeo4W enviroment
Call FindChangeOS()

End Sub

'Subroutine FindChangeOS
'- This sub launches the OpenFileDialog to find the file 'OSGeo4W.bat'.
'- If found this information Is showed In the application window('BatFileValue'). and the button 'FindButton' will be enabled.
'- If not found there will be a warning. No chances will be made ('BatFileValue' and 'FindButton').
Private Sub FindChangeOS()
    'Declarations
    Dim TempFileName As String
    Dim TempText As String
    Dim FindBatFile As New OpenFileDialog

    'Setting variables for Filedialog
    FindBatFile.InitialDirectory = "c:\"
    FindBatFile.Filter = "OSGeo4W-file | OSGeo4W.bat"
    FindBatFile.Multiselect = False

    'Opening dialog
    FindBatFile.ShowDialog()

```



```

TempFileName = FindBatFile.FileName

'Stopping subroutine if OSGeo4W.bat isn't selected (len TempfileName = 0)
If Len(TempFileName) = 0 Then
    TempText = "No OSGeo4W.bat was selected. Without this file/information the application will not work correctly!"
    MsgBox(TempText, 48, "Warning")
    Exit Sub
End If

'Setting variables
BatFilePath = TempFileName
BatFileValue.Text = TempFileName

'Deleting existing g2j.ini if exists
If My.Computer.FileSystem.FileExists(IniFilePath) Then My.Computer.FileSystem.DeleteFile(IniFilePath)

'Creating text for new g2j.ini
TempText = "BatFilePath=" & BatFilePath
My.Computer.FileSystem.WriteAllText(IniFilePath, TempText, False)

'Enabling next step of application OSGeo4W.bat
FindButton.Enabled = True
FindButton.Select()

End Sub

'OSButton_Click
'- This sub call 'FindChangeOS'.
'- Enables the opportunity to change OSGeo4W
Private Sub OSButton_Click(sender As Object, e As EventArgs) Handles OSButton.Click

    Call FindChangeOS()

End Sub

'Subroutine FindButton_Click
'- Selecting and opening a TIF or PDF file
'- Creating values for variables: ChartFileName, ChartFile, ChartName, JprFilePath, GdalContent
'- Retrieving WKT from TIF or PDG file
'- Presenting MapFileContent on the screen
'- Loading MapFileContent of MapFile into an array (LineContent)
Private Sub FindButton_Click(sender As Object, e As EventArgs) Handles FindButton.Click
    'Declarations
    Dim TempFileName As String
    Dim TempText As String
    Dim Pointer As Integer
    Dim FileNameLen As Integer
    Dim FindChartFile As New OpenFileDialog

    'Clearing Textboxes and disabling buttons

```

```

GdalInfoBox.Text = ""
JprDataBox.Text = ""
JprFileBox.Text = ""
DpiBox.Text = ""
DpiBox.Enabled = False
ExtractButton.Enabled = False
BuildButton.Enabled = False
SaveJprFileButton.Enabled = False

'Setting variables for Filedialog
FindChartFile.Filter = "TIFF/PDF-file|*.tif; *.tiff; *.pdf"
FindChartFile.Multiselect = False

'Opening dialog
FindChartFile.ShowDialog()
TempFileName = FindChartFile.FileName

'Stopping subroutine if no file is selected (len TempfileName = 0)
If Len(TempFileName) = 0 Then
    TempText = "No Chart was selected!"
    MsgBox(TempText, 48, "Warning")
    Exit Sub
End If

'Setting values for variables
ChartFilePath = TempFileName
ChartFileValue.Text = TempFileName

'Retrieve en set values for variables ChartFile and ChartPath
FileNameLen = Len(TempFileName)
For Pointer = FileNameLen To 0 Step -1
    If Mid(TempFileName, Pointer, 1) = "\" Then
        ChartPath = Mid(TempFileName, 1, Pointer)
        ChartFile = Mid(TempFileName, Pointer + 1, FileNameLen - Pointer)
    Exit For
    End If
Next Pointer

'Retrieving and setting values for variables ChartName and JprFilePath
FileNameLen = Len(ChartFile)
For Pointer = FileNameLen To 0 Step -1
    If Mid(ChartFile, Pointer, 1) = "." Then
        ChartName = Mid(ChartFile, 1, Pointer - 1)
        JprFilePath = ChartPath & ChartName & ".jpr"
    Exit For
    End If
Next Pointer

'Retrieving Geo-information with GdalInfo
'- Setting variabel TempFilePath

```

```

' (My.Computer.FileSystem.SpecialDirectories.Temp point to users "Temporary" directory)
TempFilePath = My.Computer.FileSystem.SpecialDirectories.Temp & "\" & "g2j.tmp"

'- Retrieving Gdal (WKT) data and copying to g2j.tmp
' Wait:=True > to be sure process is finalised before executing next statement.
' This prevents an error! (trial and error!)
Call Shell(BatFilePath & " gdalinfo.exe " & ChartFilePath & " > " & TempFilePath, AppWinStyle.Hide, Wait:=True)

'- Copying content of g2j.tmp to "GdalContent"
GdalContent = My.Computer.FileSystem.ReadAllText(TempFilePath)

'- Loading GdalContent into 'GdalInfoBox'
GdalInfoBox.Text = GdalContent

'- Deleting tempfile g2j.tmp
My.Computer.FileSystem.DeleteFile(TempFilePath)

'Testing for Geographic data (looking for the keyword 'PROJCS')
'- If not: Warning
'- If so: enabling 'ExtractButton' for next step
Pointer = InStr(GdalContent, "PROJCS", CompareMethod.Binary)
If Pointer = 0 Then
    MsgBox("The TIFF/PDF file doesn't contain geographic data!", 48, "Warning")
    'Disabling next steps in the application
    ExtractButton.Enabled = False
    BuildButton.Enabled = False
    SaveJprFileButton.Enabled = False
Else
    'Enabling next steps in the application
    ExtractButton.Enabled = True
    ExtractButton.Select()
End If

End Sub

'Function CvtLatLonStr converts the string expression, representing a Latitude or longitude,
'(in degree, minutes, secondes and derection) in a string expression, representing a Latitude
'or longitude in decimal degree.
Private Function CvtLatLonStr(LatLonStr As String) As String
    'Declarations
    Dim ValDeg, ValMin, ValSec, ValSign As Double
    Dim TempText1, TempText2 As String
    Dim ChrPointer As Integer

    'Extracting value degrees (ValDeg)
    TempText1 = LatLonStr
    ChrPointer = InStr(TempText1, "d", CompareMethod.Binary)
    TempText2 = Trim(Mid(TempText1, 1, ChrPointer - 1))
    ValDeg = Val(TempText2)

```

```

'Extracting value minutes (ValMin)
TempText1 = Mid(TempText1, ChrPointer + 1, Len(TempText1) - ChrPointer)
ChrPointer = InStr(TempText1, "'", CompareMethod.Binary)
TempText2 = Trim(Mid(TempText1, 1, ChrPointer - 1))
ValMin = Val(TempText2)

'Extracting value Secondes (ValSec)
TempText1 = Mid(TempText1, ChrPointer + 1, Len(TempText1) - ChrPointer)
ChrPointer = InStr(TempText1, Chr(34), CompareMethod.Binary)
TempText2 = Trim(Mid(TempText1, 1, ChrPointer - 1))
ValSec = Val(TempText2)

'Extracting value Sign (+/-, ValSgn)
TempText2 = Mid(TempText1, ChrPointer + 1, Len(TempText1) - ChrPointer)
Select Case TempText2
    Case "W", "S"
        ValSign = -1
    Case "N", "E"
        ValSign = 1
End Select

'Creating return value of function and ending
Return Format(((ValDeg + (ValMin / 60) + (ValSec / 3600)) * ValSign), "0.000000")

End Function

'Function CommaToDot converts, in a string representing as number, the comma (,)
'as a separator character into a dot (.).
Private Function CommaToDot(CommaString As String) As String
    'Declarations
    Dim TempText As String
    Dim ChrPointer As Integer

    'Moving character by character from 'CommaString' to 'TempText'
    'unless it is a comma (,). In this case the comma will be replaced by a dot (.).
    TempText = ""
    For ChrPointer = 1 To Len(CommaString)
        If Mid(CommaString, ChrPointer, 1) <> "," Then
            TempText = TempText + Mid(CommaString, ChrPointer, 1)
        Else
            TempText = TempText + "."
        End If
    Next

    'Returning string to function and ending
    Return TempText
End Function

'Function UnderScoreToSpace converts, in a string, the underscore (_) character into a space ( ).
Private Function UnderScoreToSpace(UnderScoreString As String) As String

```

```

'Declarations
Dim TempText As String
Dim ChrPointer As Integer

'Moving character by character form 'UnderScoreString' to 'TempText'
'unless it is a comma (,). In this case the comma will be replaced by a dot (.).
TempText = ""
For ChrPointer = 1 To Len(UnderScoreString)
    If Mid(UnderScoreString, ChrPointer, 1) <> "," Then
        TempText = TempText + Mid(UnderScoreString, ChrPointer, 1)
    Else
        TempText = TempText + "."
    End If
Next

'Returning string to function and ending
Return TempText
End Function

'Subroutine ClearData clears all the information of variables used to create a JPR file
Private Sub ClearData()
'Declarations
Dim Pointer As Integer

'- The JPR data
JprData.nm = "nodata"
JprData.sc = -1
JprData.sr = -1
JprData.sd = -1
JprData.it = "nodata"
JprData.dm = "nodata"
JprData.st = 0
JprData.sn = 0
JprData.pr = "nodata"
JprData.zn = "nodata"
JprData.pp = "nodata"
JprData.p1 = "nodata"
JprData.p2 = "nodata"
JprData.p3 = "nodata"
JprData.p4 = "nodata"
JprData.p5 = "nodata"
JprData.p6 = "nodata"

'- The necessary Gdal data
GdalData.ResolutionX = -1
GdalData.ResolutionY = -1
GdalData.ResolutionUnit = -1
GdalData.PixelSizeX = -1
GdalData.PixelSizeY = -1

```

```

'- The reference points
For Pointer = 1 To 5
    ReferencePoint(Pointer).x = 0
    ReferencePoint(Pointer).y = 0
    ReferencePoint(Pointer).lat = 0
    ReferencePoint(Pointer).lon = 0
Next

'- The vertex points (like Neatline)
For Pointer = 1 To 4
    VertexPoint(Pointer).x = 0
    VertexPoint(Pointer).y = 0
    VertexPoint(Pointer).lat = 0
    VertexPoint(Pointer).lon = 0
Next

End Sub

'Sub ExtractButton_Click is retrieving/extracting information to create a JPR file
Private Sub ExtractButton_Click(sender As Object, e As EventArgs) Handles ExtractButton.Click
    'Declarations
    Dim DataLine() As String = Split(GdalContent, EndLine)           'Splitting GdalContent in to lines
    Dim LineCount As Integer = DataLine.Length - 1                 'Determining number of lines
    Dim Counter As Integer
    Dim Pointer(1) As Integer
    Dim TempText As String
    Dim Answer As Integer
    Dim MsgText As String
    Dim TempCode As Integer

    'Clearing all data
    Call ClearData()
    JprDataBox.Text = ""
    JprFileBox.Text = ""
    DpiBox.Enabled = False
    DpiBox.Text = ""

    'Retrieving Nm (name of map) and graphic type (it)
    JprData.nm = ChartName
    JprDataBox.Text = JprDataBox.Text & "nm (chartname) = " & JprData.nm & EndLine
    JprData.it = Mid(ChartFile, Len(ChartName) + 2, 3)
    JprDataBox.Text = JprDataBox.Text & "it (graphics type) = " & JprData.it & EndLine

    'Retrieving/extracting data with a "more or less" one on one relation
    For Counter = 0 To LineCount

        'Retrieving/extracting dm (geographic datum)
        If InStr(DataLine(Counter), "DATUM", CompareMethod.Binary) > 0 Then
            TempText = Trim(LCase(UnderScoreToSpace(DataLine(Counter))))
            Pointer(0) = InStr(TempText, "[", CompareMethod.Binary)
        End If
    Next
End Sub

```

```

TempText = Mid(TempText, Pointer(0) + 2, Len(TempText) - Pointer(0) - 3)
JprData.dm = TempText
JprDataBox.Text = JprDataBox.Text & "dm (datum) = " & JprData.dm & EndLine
End If

'Retrieving/extracting pr (projection)
If InStr(DataLine(Counter), "PROJECTION", CompareMethod.Binary) > 0 Then
    Pointer(0) = InStr(DataLine(Counter), "[", CompareMethod.Binary)
    Pointer(1) = InStr(DataLine(Counter), "]", CompareMethod.Binary)
    TempText = UnderScoreToSpace(LCase(Mid(DataLine(Counter), Pointer(0) + 2, Pointer(1) - Pointer(0) - 3)))
    JprData.pr = TempText
    JprDataBox.Text = JprDataBox.Text & "pr (projection) = " & JprData.pr & EndLine
End If

'Retrieving/extracting pp (central meridian)
If InStr(LCase(DataLine(Counter)), "central_meridian", CompareMethod.Binary) > 0 Then
    Pointer(0) = InStr(DataLine(Counter), ",", CompareMethod.Binary)
    Pointer(1) = InStr(DataLine(Counter), "]", CompareMethod.Binary)
    JprData.pp = Trim(Mid(DataLine(Counter), Pointer(0) + 1, Pointer(1) - Pointer(0) - 1))
    JprDataBox.Text = JprDataBox.Text & "pp (central meridian) = " & JprData.pp & EndLine
End If

'Retrieving/extracting p1 (latitude of origin)
If InStr(LCase(DataLine(Counter)), "latitude_of_origin", CompareMethod.Binary) > 0 Then
    Pointer(0) = InStr(DataLine(Counter), ",", CompareMethod.Binary)
    Pointer(1) = InStr(DataLine(Counter), "]", CompareMethod.Binary)
    JprData.p1 = Trim(Mid(DataLine(Counter), Pointer(0) + 1, Pointer(1) - Pointer(0) - 1))
    JprDataBox.Text = JprDataBox.Text & "p1 (latitude of origin) = " & JprData.p1 & EndLine
End If

'Retrieving/extracting p2 (scale factor)
If InStr(LCase(DataLine(Counter)), "scale_factor", CompareMethod.Binary) > 0 Then
    Pointer(0) = InStr(DataLine(Counter), ",", CompareMethod.Binary)
    Pointer(1) = InStr(DataLine(Counter), "]", CompareMethod.Binary)
    JprData.p2 = Trim(Mid(DataLine(Counter), Pointer(0) + 1, Pointer(1) - Pointer(0) - 1))
    JprDataBox.Text = JprDataBox.Text & "p2 (scale factor) = " & JprData.p2 & EndLine
End If

'Retrieving/extracting p3 (false northing)
If InStr(LCase(DataLine(Counter)), "false_northing", CompareMethod.Binary) > 0 Then
    Pointer(0) = InStr(DataLine(Counter), ",", CompareMethod.Binary)
    Pointer(1) = InStr(DataLine(Counter), "]", CompareMethod.Binary)
    JprData.p3 = Trim(Mid(DataLine(Counter), Pointer(0) + 1, Pointer(1) - Pointer(0) - 1))
    JprDataBox.Text = JprDataBox.Text & "p3 (false northing) = " & JprData.p3 & EndLine
End If

'Retrieving/extracting p4 (false easting)
If InStr(LCase(DataLine(Counter)), "false_easting", CompareMethod.Binary) > 0 Then
    Pointer(0) = InStr(DataLine(Counter), ",", CompareMethod.Binary)
    Pointer(1) = InStr(DataLine(Counter), "]", CompareMethod.Binary)

```

```

JprData.p4 = Trim(Mid(DataLine(Counter), Pointer(0) + 1, Pointer(1) - Pointer(0) - 1))
JprDataBox.Text = JprDataBox.Text & "p4 (false easting) = " & JprData.p4 & EndLine
End If

'Retrieving/extracting p5 (standard parallel 1)
If InStr(LCase(DataLine(Counter)), "standard_parallel_1", CompareMethod.Binary) > 0 Then
    Pointer(0) = InStr(DataLine(Counter), ",", CompareMethod.Binary)
    Pointer(1) = InStr(DataLine(Counter), "]", CompareMethod.Binary)
    JprData.p5 = Trim(Mid(DataLine(Counter), Pointer(0) + 1, Pointer(1) - Pointer(0) - 1))
    JprDataBox.Text = JprDataBox.Text & "p5 (standard parallel 1) = " & JprData.p5 & EndLine
End If

'Retrieving/extracting p6 (standard parallel 2)
If InStr(LCase(DataLine(Counter)), "standard_parallel_2", CompareMethod.Binary) > 0 Then
    Pointer(0) = InStr(DataLine(Counter), ",", CompareMethod.Binary)
    Pointer(1) = InStr(DataLine(Counter), "]", CompareMethod.Binary)
    JprData.p6 = Trim(Mid(DataLine(Counter), Pointer(0) + 1, Pointer(1) - Pointer(0) - 1))
    JprDataBox.Text = JprDataBox.Text & "p6 (standard parallel 2) = " & JprData.p6 & EndLine
End If

Next

'Retrieving/extracting width, Height and center of the image in pixel and their geographical coordinates
'needed for Reference and vertex points
For Counter = 0 To LineCount

    '- Retrieving width, Height and center in pixel
    If InStr(LCase(DataLine(Counter)), "size is", CompareMethod.Binary) > 0 Then
        Dim Size() As String = Split(Mid(DataLine(Counter), 8, Len(DataLine(Counter)) - 7), ",", CompareMethod.Binary)
        ImageWidth = Val(Size(0))
        ImageHeight = Val(Size(1))
        '- Setting Pixel coordinates of the upper left, upper right, lower right, lower left and center
        ReferencePoint(1).x = 0
        ReferencePoint(1).y = 0
        ReferencePoint(2).x = ImageWidth - 1
        ReferencePoint(2).y = 0
        ReferencePoint(3).x = ImageWidth - 1
        ReferencePoint(3).y = ImageHeight - 1
        ReferencePoint(4).x = 0
        ReferencePoint(4).y = ImageHeight - 1
        ReferencePoint(5).x = ImageWidth * 0.5 - 1
        ReferencePoint(5).y = ImageHeight * 0.5 - 1
    End If

    '- Retrieving and setting the geographical coordinate of the upper left corner
    If InStr(DataLine(Counter), "Upper Left", CompareMethod.Binary) > 0 Then
        Dim TempString1() As String = Split(DataLine(Counter), "(" , CompareMethod.Binary)
        Dim TempString2() As String = Split(TempString1(1), ",", CompareMethod.Binary)
        ReferencePoint(1).lon = CvtLatLonStr(TempString2(0))
        ReferencePoint(1).lat = CvtLatLonStr(Mid(TempString2(1), 1, Len(TempString2(1)) - 1))
    End If
End For

```



```

End If

'- Retrieving and setting the geographical coordinate of the upper right corner
If InStr(DataLine(Counter), "Upper Right", CompareMethod.Binary) > 0 Then
    Dim TempString1() As String = Split(DataLine(Counter), ") (", CompareMethod.Binary)
    Dim TempString2() As String = Split(TempString1(1), ",", CompareMethod.Binary)
    ReferencePoint(2).lon = CvtLatLonStr(TempString2(0))
    ReferencePoint(2).lat = CvtLatLonStr(Mid(TempString2(1), 1, Len(TempString2(1)) - 1))
End If

'- Retrieving and setting the geographical coordinate of the lower right corner
If InStr(DataLine(Counter), "Lower Right", CompareMethod.Binary) > 0 Then
    Dim TempString1() As String = Split(DataLine(Counter), ") (", CompareMethod.Binary)
    Dim TempString2() As String = Split(TempString1(1), ",", CompareMethod.Binary)
    ReferencePoint(3).lon = CvtLatLonStr(TempString2(0))
    ReferencePoint(3).lat = CvtLatLonStr(Mid(TempString2(1), 1, Len(TempString2(1)) - 1))
End If

'- Retrieving and setting the geographical coordinate of the lower left corner
If InStr(DataLine(Counter), "Lower Left", CompareMethod.Binary) > 0 Then
    Dim TempString1() As String = Split(DataLine(Counter), ") (", CompareMethod.Binary)
    Dim TempString2() As String = Split(TempString1(1), ",", CompareMethod.Binary)
    ReferencePoint(4).lon = CvtLatLonStr(TempString2(0))
    ReferencePoint(4).lat = CvtLatLonStr(Mid(TempString2(1), 1, Len(TempString2(1)) - 1))
End If

'- Retrieving and setting the geographical coordinate of the center
If InStr(DataLine(Counter), "Center ", CompareMethod.Binary) > 0 Then
    Dim TempString1() As String = Split(DataLine(Counter), ") (", CompareMethod.Binary)
    Dim TempString2() As String = Split(TempString1(1), ",", CompareMethod.Binary)
    ReferencePoint(5).lon = CvtLatLonStr(TempString2(0))
    ReferencePoint(5).lat = CvtLatLonStr(Mid(TempString2(1), 1, Len(TempString2(1)) - 1))
End If

Next

'- Copying reference points to 'JprDataBox'
For Pointer(0) = 1 To 5
    JprDataBox.Text = JprDataBox.Text & "rp" & Pointer(0) & " (Reference point " & Pointer(0) & ") = " &
        CommaToDot(CStr(ReferencePoint(Pointer(0)).lat)) & ", " &
        CommaToDot(CStr(ReferencePoint(Pointer(0)).lon)) & ", " &
        CommaToDot(CStr(ReferencePoint(Pointer(0)).x)) & ", " &
        CommaToDot(CStr(ReferencePoint(Pointer(0)).y)) & EndLine
Next

'Asking if the map has a collar.
'- If not: Vertex polygon will be created based on the reference points (1 to 4) and copied to JprDataBox
' (Msgbox appearance: 4 = Yes/No, 32 = Question mark, 256 = second button default, 292 = total)
Answer = MsgBox("Does the map has a collar?", 292, "Question")
If Answer = 7 Then

```

```

For Pointer(0) = 1 To 4
    VertexPoint(Pointer(0)) = ReferencePoint(Pointer(0))
    JprDataBox.Text = JprDataBox.Text & "vp" & Pointer(0) & " (Vertex point " & Pointer(0) & ") = " &
    CommaToDot(CStr(VertexPoint(Pointer(0)).x)) & ", " &
    CommaToDot(CStr(VertexPoint(Pointer(0)).y)) & EndLine
Next
End If

'Handling UTM case (UTM is a special case Transverse Mercator)
'Retrieving/ extracting zn (UTM zone data)
For Counter = 0 To LineCount
    If InStr(LCase(DataLine(Counter)), "utm zone", CompareMethod.Binary) > 0 Then
        Pointer(0) = InStr(LCase(DataLine(Counter)), "utm zone", CompareMethod.Binary)
        TempText = Mid(DataLine(Counter), Pointer(0) + 9, 3)
        JprData.zn = TempText
        Select Case Mid(JprData.zn, 3, 1)
            Case "N", "n"
                JprData.zn = Mid(JprData.zn, 1, 2) & "t"
            Case "S", "s"
                JprData.zn = Mid(JprData.zn, 1, 2) & "j"
            Case Else
                'if no N or S: the t or j will be based on latitude of center map (reference point 5)
                If Val(CommaToDot(CStr(ReferencePoint(5).lat))) >= 0 Then
                    JprData.zn = Mid(JprData.zn, 1, 2) & "t"
                Else
                    JprData.zn = Mid(JprData.zn, 1, 2) & "j"
                End If
            End Select
        JprDataBox.Text = JprDataBox.Text & "zn (UTM zone) = " & JprData.zn & EndLine
        'Changing Projection to UTM
        JprData.pr = "utm"
        JprDataBox.Text = JprDataBox.Text & "pr is changed to 'utm'" & EndLine
    End If
Next

'Retrieving/extracting data to calculate the scale (sc) and the resolution (sr)
For Counter = 0 To LineCount
    If InStr(DataLine(Counter), "TIFFTAG_RESOLUTIONUNIT", CompareMethod.Binary) > 0 Then
        Dim TempString1() As String = Split(DataLine(Counter), "=", CompareMethod.Binary)
        GdalData.ResolutionUnit = Val(Mid(TempString1(1), 1, 1))
    End If

    If InStr(DataLine(Counter), "TIFFTAG_XRESOLUTION", CompareMethod.Binary) > 0 Then
        Dim TempString1() As String = Split(DataLine(Counter), "=", CompareMethod.Binary)
        GdalData.ResolutionX = Val(TempString1(1))
    End If

    If InStr(DataLine(Counter), "TIFFTAG_YRESOLUTION", CompareMethod.Binary) > 0 Then
        Dim TempString1() As String = Split(DataLine(Counter), "=", CompareMethod.Binary)
        GdalData.ResolutionY = Val(TempString1(1))
    End If

```

```

End If

If InStr(DataLine(Counter), "Pixel Size =", CompareMethod.Binary) > 0 Then
    Dim TempString1() As String = Split(DataLine(Counter), "(", CompareMethod.Binary)
    Dim TempString2() As String = Split(TempString1(1), ",", CompareMethod.Binary)
    GdalData.PixelSizeX = System.Math.Abs(Val(TempString2(0)))
    GdalData.PixelSizeY = System.Math.Abs(Val(Mid(TempString2(1), 1, Len(TempString2(1)) - 1)))
End If
Next

'Resetting 'ResolutionScaleCode'
ResolutionScaleCode = 0

'Handling the resolution (sr)

'Determining code for msg no resolution could be determined
If GdalData.ResolutionX <> GdalData.ResolutionY Then ResolutionScaleCode = ResolutionScaleCode + 1
If GdalData.ResolutionX <= 0 Or GdalData.ResolutionY <= 0 Then ResolutionScaleCode = ResolutionScaleCode + 2
If GdalData.ResolutionUnit < 2 Or GdalData.ResolutionUnit > 3 Then ResolutionScaleCode = ResolutionScaleCode + 4

'Retrieving resolution (sr)
If GdalData.ResolutionUnit = 3 Then
    JprData.sr = GdalData.ResolutionX * 2.54
    JprDataBox.Text = JprDataBox.Text & "sr (resolution) = " & JprData.sr & EndLine
End If
If GdalData.ResolutionUnit = 2 Then
    JprData.sr = GdalData.ResolutionX
    JprDataBox.Text = JprDataBox.Text & "sr (resolution) = " & JprData.sr & EndLine
End If

'Handling the scale
'- If no pixeldata: GdalData.PixelSizeX and GdalData.PixelSizeY has its default -1
If GdalData.PixelSizeX < 0 Then ResolutionScaleCode = ResolutionScaleCode + 8

'- The pixelsize X/Y can slidy differ. If less then 0.5% it is OK
If GdalData.PixelSizeX / GdalData.PixelSizeY > 1.005 Or GdalData.PixelSizeX / GdalData.PixelSizeY < 0.995 Then
    ResolutionScaleCode = ResolutionScaleCode + 16
End If

'- Calculating scale (sc) ("if everything when well")
If ResolutionScaleCode = 0 Then
    JprData.sd = (GdalData.PixelSizeX + GdalData.PixelSizeY) / 2
    JprData.sc = System.Math.Round(JprData.sd * 100 * (JprData.sr / 2.54))
    JprDataBox.Text = JprDataBox.Text & "sc (scale) = " & JprData.sc & EndLine
End If

'Creating the 'MsgText' based on ResolutionScaleCode
If ResolutionScaleCode > 0 Then
    TempCode = ResolutionScaleCode
    MsgText = "The resolution and/or scale couldn't determined! (" & ResolutionScaleCode & ")" & EndLine

```

```

    If TempCode >= 16 Then
        TempCode = TempCode - 16
        MsgText = MsgText & "- The height/width of a pixel differs to much" & EndLine
    End If
    If TempCode >= 8 Then
        TempCode = TempCode - 8
        MsgText = MsgText & "- The pixel size couldn't be determined" & EndLine
    End If
    If TempCode >= 4 Then
        TempCode = TempCode - 4
        MsgText = MsgText & "- The resolution unit couldn't be determined" & EndLine
    End If
    If TempCode >= 2 Then
        TempCode = TempCode - 2
        MsgText = MsgText & "- The height and/or width resolution couldn't be determined" & EndLine
    End If
    If TempCode = 1 Then
        MsgText = MsgText & "- The height and/or width resolution differ" & EndLine
    End If
    JprDataBox.Text = JprDataBox.Text & MsgText
End If

'Enabling next step of the application
BuildButton.Enabled = True
BuildButton.Select()

End Sub

'The subroutine BuildButton_Click Creates the contents of the JPR file.
Private Sub BuildButton_Click(sender As Object, e As EventArgs) Handles BuildButton.Click
    'Declarations
    Dim Pointer(2) As Integer

    'Cpyping Data to JprFileBox
    '- Standard values
    JprFileBox.Text = "//This file was created Using Gdal data To JPR (G2J) by www.hzns.nl" & EndLine
    JprFileBox.Text = JprFileBox.Text & "nm= " & JprData.nm & EndLine
    JprFileBox.Text = JprFileBox.Text & "it=" & JprData.it & EndLine
    If JprData.it <> "tif" Then
        JprFileBox.Text = JprFileBox.Text & "//The PDF-file has To be converted into a graphics format (256 colors/8 bit)." & EndLine &
            "//Change the value 'it' into the proper format (gif, jpg ,png, bmp or tif)." & EndLine
    End If

    '- Datum values
    If JprData.dm <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "dm=" & TranslateDatum(JprData.dm) & EndLine
    JprFileBox.Text = JprFileBox.Text & "st=" & JprData.st & EndLine
    JprFileBox.Text = JprFileBox.Text & "sn=" & JprData.sn & EndLine

    '- Projection values
    If JprData.pr <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "pr=" & TranslateProjection(JprData.pr) & EndLine

```

```

Select Case TranslateProjection(JprData.pr)
Case "mercator"
    If JprData.p1 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p1=" & JprData.p1 & EndLine
Case "transverse mercator"
    If JprData.pp <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "pp=" & JprData.pp & EndLine
    If JprData.p1 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p1=" & JprData.p1 & EndLine
    If JprData.p2 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p2=" & JprData.p2 & EndLine
    If JprData.p3 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p3=" & JprData.p3 & EndLine
    If JprData.p4 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p4=" & JprData.p4 & EndLine
Case "utm"
    If JprData.pp <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "pp=" & JprData.pp & EndLine
    If JprData.p1 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p1=" & JprData.p1 & EndLine
    If JprData.p2 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p2=" & JprData.p2 & EndLine
    If JprData.p3 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p3=" & JprData.p3 & EndLine
    If JprData.p4 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p4=" & JprData.p4 & EndLine
    If JprData.zn <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "zn=" & JprData.zn & EndLine
Case "lambert conformal conic"
    If JprData.pp <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "pp=" & JprData.pp & EndLine
    If JprData.p1 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p1=" & JprData.p1 & EndLine
    If JprData.p5 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p5=" & JprData.p5 & EndLine
    If JprData.p6 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p6=" & JprData.p6 & EndLine
Case Else
    JprFileBox.Text = JprFileBox.Text & "//Your map program may possibly not supports this projection." & EndLine
    If JprData.pp <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "pp=" & JprData.pp & EndLine
    If JprData.p1 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p1=" & JprData.p1 & EndLine
    If JprData.p2 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p2=" & JprData.p2 & EndLine
    If JprData.p3 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p3=" & JprData.p3 & EndLine
    If JprData.p4 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p4=" & JprData.p4 & EndLine
    If JprData.p5 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p5=" & JprData.p5 & EndLine
    If JprData.p6 <> "nodata" Then JprFileBox.Text = JprFileBox.Text & "p6=" & JprData.p6 & EndLine
End Select

'- Resolution and Scale values
If JprData.sr <> -1 Then JprFileBox.Text = JprFileBox.Text & "sr=" & JprData.sr & EndLine
If JprData.sc <> -1 Then JprFileBox.Text = JprFileBox.Text & "sc=" & JprData.sc & EndLine

'- Calibration values
For Pointer(0) = 1 To 5
    JprFileBox.Text = JprFileBox.Text & "rp" & Pointer(0) & "=" &
        CommaToDot(CStr(ReferencePoint(Pointer(0)).lat)) & "," &
        CommaToDot(CStr(ReferencePoint(Pointer(0)).lon)) & "," &
        CommaToDot(CStr(ReferencePoint(Pointer(0)).x)) & "," &
        CommaToDot(CStr(ReferencePoint(Pointer(0)).y)) & EndLine
Next

'- Vertex values
If VertexPoint(3).x <> 0 Then
    For Pointer(0) = 1 To 4
        JprFileBox.Text = JprFileBox.Text & "vp" & Pointer(0) & "=" &
            CommaToDot(CStr(VertexPoint(Pointer(0)).x)) & "," &

```

```

        CommaToDot(CStr(VertexPoint(Pointer(0)).y)) & EndLine
    Next

End If

'Enabling next step of the application
SaveJprFileButton.Enabled = True
SaveJprFileButton.Select()

'Conditional (only if relevant data are available) enabling DpiBox for next step of the application
If ResolutionScaleCode = 0 Then
    DpiBox.Enabled = True
    DpiBox.Text = JprData.sr
End If

End Sub

'The function 'TranslateProjection' translates the Gdal value for the projection
'to the proper JPR value.
Private Function TranslateProjection(GdalProjection As String) As String
    'Declarations
    Dim TempProjection As String

    Select Case GdalProjection
        Case "lcc", "lambert cc", "lambert conformal"
            TempProjection = "lambert conformal conic"
        Case "tm"
            TempProjection = "transverse mercator"
        Case "universal transverse mercator"
            TempProjection = "utm"
        Case "merc"
            TempProjection = "mercator"
        Case Else
            TempProjection = GdalProjection
    End Select

    Return TempProjection
End Function

'The function TranslateDatum translates the Gdal value for the datum
'to the proper JPR value. This function gives the flexibility to translate weird projection names into
'more common names (although the parameters of both datums must be more or less the same).
Private Function TranslateDatum(GdalDatum As String) As String
    'Declarations
    Dim TempDatum As String

    Select Case GdalDatum
        Case "north american datum 1983"
            TempDatum = "nad83"
        Case "north american datum 1927"

```

```

        TempDatum = "nad27"
    Case "wgs 1984", "european terrestrial reference system 1989", "world geodetic system 1984"
        TempDatum = "wgs84"
    Case Else
        TempDatum = GdalDatum
End Select

Return TempDatum
End Function

'Subroutine SaveJprFileButton_Click saves the JprFileBox.text to a JPR-file in the same directory
'as the original TIF/PDF file. If this file exists, a warning will be prompted.
Private Sub SaveJprFileButton_Click(sender As Object, e As EventArgs) Handles SaveJprFileButton.Click
    'Declararions
    Dim Answer As Integer

    'Checking for an existing JPR file
    If My.Computer.FileSystem.FileExists(JprFilePath) = True Then
        '(Msgbox appearance: 4 = Yes/No, 32 = Question mark, 256 = second button default, 292 = total)
        Answer = MsgBox("JPR-file allready exists. Overwrite the existing file?", 292, "Warning")
        If Answer = 6 Then
            'Overwriting existing JPR-file if answer on MsgBox is Yes (Answer = 6)
            My.Computer.FileSystem.WriteAllText(JprFilePath, JprFileBox.Text, False)
        End If
    Else
        'Creating JPR-file
        My.Computer.FileSystem.WriteAllText(JprFilePath, JprFileBox.Text, False)
    End If
End Sub

'Subroutne DisclaimButton shows the disclaimer foor hzns.nl and gdal.org
Private Sub DisclaimButton_Click(sender As Object, e As EventArgs) Handles DisclaimButton.Click
    'Declarations
    Dim MsgText As String

    'Disclaimer message
    MsgText = "This application is provided 'As is'. The use of the application is on your own risk. " &
        "Direct or indirect damage by using this application is users responsibility/risk, " &
        "not the application-builders. Redistribution by a third party (commercial of non-commercial) " &
        "is prohibit. Download the application direct from https://hzns.nl." & EndLine & EndLine &
        "This application uses gdalinfo, ad part of Gdal (GDAL/OGR contributors (2019). GDAL/OGR " &
        "Geospatial Data Abstraction software Library. Open Source Geospatial Foundation. (https://gdal.org))."
    MsgBox(MsgText, 48, "Disclaimer")
End Sub

'Unloading and closing application
Private Sub EndButton_Click(sender As Object, e As EventArgs) Handles EndButton.Click
    Me.Close()

```

```
End Sub
```

```
'Subroutine DipBox_LostFocus handles the cases LostFocus and SelectedValueChanged (with focus on DipBox)
'- recalculating the map scale
Private Sub DipBox_LostFocus(sender As Object, e As EventArgs) Handles DipBox.LostFocus, DipBox.SelectedValueChanged
    If JprDataBox.Text <> "" Then
        If JprData.sr <> Val(DipBox.Text) Then
            JprData.sr = Val(DipBox.Text)
            JprData.sc = System.Math.Round(JprData.sd * 100 * (JprData.sr / 2.54))
            BuildButton_Click(sender, e)
        End If
    End If
End Sub
```

```
End Sub
```

```
'Subroutine DipBox_PressKey handles the case "Enter"-key was used (with focus on DipBox)
'- recalculating the map scale
Private Sub DipBox_KeyPress(sender As Object, e As KeyPressEventArgs) Handles DipBox.KeyPress
    If JprDataBox.Text <> "" And e.KeyChar = Chr(13) Then
        If JprData.sr <> Val(DipBox.Text) Then
            JprData.sr = Val(DipBox.Text)
            JprData.sc = System.Math.Round(JprData.sd * 100 * (JprData.sr / 2.54))
            BuildButton_Click(sender, e)
        End If
    End If
End Sub
```

```
End Sub
```

```
End Class
```

Code for Module G2J.Designer.vb

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
```

```
Partial Class G2J
```

```
    Inherits System.Windows.Forms.Form
```

```
'Form overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()> _
Protected Overrides Sub Dispose(ByVal disposing As Boolean)
    Try
        If disposing AndAlso components IsNot Nothing Then
            components.Dispose()
        End If
    Finally
        MyBase.Dispose(disposing)
    End Try
End Sub
```

```
'Required by the Windows Form Designer
```

```
Private components As System.ComponentModel.IContainer
```

```
'NOTE: The following procedure is required by the Windows Form Designer
```

```
'It can be modified using the Windows Form Designer.
```

```
'Do not modify it using the code editor.
```

```
<System.Diagnostics.DebuggerStepThrough()> _
```

```
Private Sub InitializeComponent()
```

```
    Me.BatFileLabel = New System.Windows.Forms.Label()
```

```
    Me.BatFileValue = New System.Windows.Forms.Label()
```

```
    Me.OSBox = New System.Windows.Forms.GroupBox()
```

```
    Me.OSButton = New System.Windows.Forms.Button()
```

```
    Me.FindBox = New System.Windows.Forms.GroupBox()
```

```
    Me.GdalInfoBox = New System.Windows.Forms.TextBox()
```

```
    Me.ChartFileLabel = New System.Windows.Forms.Label()
```

```
    Me.ChartFileValue = New System.Windows.Forms.Label()
```

```
    Me.FindButton = New System.Windows.Forms.Button()
```

```
    Me.ExtractBox = New System.Windows.Forms.GroupBox()
```

```
    Me.JprDataBox = New System.Windows.Forms.TextBox()
```

```
    Me.ExtractButton = New System.Windows.Forms.Button()
```

```
    Me.BuildBox = New System.Windows.Forms.GroupBox()
```

```
    Me.DpiLabel = New System.Windows.Forms.Label()
```

```
    Me.DpiBox = New System.Windows.Forms.ComboBox()
```

```
    Me.SaveJprFileButton = New System.Windows.Forms.Button()
```

```
    Me.JprFileBox = New System.Windows.Forms.TextBox()
```

```
    Me.BuildButton = New System.Windows.Forms.Button()
```

```
    Me.DisclaimButton = New System.Windows.Forms.Button()
```

```
    Me.EndButton = New System.Windows.Forms.Button()
```

```
    Me.OSBox.SuspendLayout()
```

```
    Me.FindBox.SuspendLayout()
```

```
    Me.ExtractBox.SuspendLayout()
```

```
    Me.BuildBox.SuspendLayout()
```

```
    Me.SuspendLayout()
```

```
,
```

```
    'BatFileLabel
```

```
,
```

```
    Me.BatFileLabel.AutoSize = True
```

```
    Me.BatFileLabel.Location = New System.Drawing.Point(134, 19)
```

```
    Me.BatFileLabel.Name = "BatFileLabel"
```

```
    Me.BatFileLabel.Size = New System.Drawing.Size(117, 13)
```

```
    Me.BatFileLabel.TabIndex = 0
```

```
    Me.BatFileLabel.Text = "OSGeo4W.bat location"
```

```
,
```

```
    'BatFileValue
```

```
,
```

```
    Me.BatFileValue.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
```

```
    Me.BatFileValue.Location = New System.Drawing.Point(137, 47)
```

```
    Me.BatFileValue.Name = "BatFileValue"
```

```
    Me.BatFileValue.Size = New System.Drawing.Size(416, 15)
```

```
    Me.BatFileValue.TabIndex = 1
```

```
,
```

```
'OSBox
'
Me.OSBox.Controls.Add(Me.OSButton)
Me.OSBox.Controls.Add(Me.BatFileLabel)
Me.OSBox.Controls.Add(Me.BatFileValue)
Me.OSBox.Location = New System.Drawing.Point(12, 12)
Me.OSBox.Name = "OSBox"
Me.OSBox.Size = New System.Drawing.Size(568, 81)
Me.OSBox.TabIndex = 2
Me.OSBox.TabStop = False
Me.OSBox.Text = "OSGeo4W/GDAL Environment"
'
'OSButton
'
Me.OSButton.Location = New System.Drawing.Point(6, 19)
Me.OSButton.Name = "OSButton"
Me.OSButton.Size = New System.Drawing.Size(122, 43)
Me.OSButton.TabIndex = 0
Me.OSButton.Text = "Find/Change &Environment"
Me.OSButton.UseVisualStyleBackColor = True
'
'FindBox
'
Me.FindBox.Controls.Add(Me.GdalInfoBox)
Me.FindBox.Controls.Add(Me.ChartFileLabel)
Me.FindBox.Controls.Add(Me.ChartFileValue)
Me.FindBox.Controls.Add(Me.FindButton)
Me.FindBox.Location = New System.Drawing.Point(12, 104)
Me.FindBox.Name = "FindBox"
Me.FindBox.Size = New System.Drawing.Size(568, 185)
Me.FindBox.TabIndex = 3
Me.FindBox.TabStop = False
Me.FindBox.Text = "Find GeoTIFF or GeoPDF file"
'
'GdalInfoBox
'
Me.GdalInfoBox.Font = New System.Drawing.Font("Consolas", 8.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.GdalInfoBox.Location = New System.Drawing.Point(6, 68)
Me.GdalInfoBox.Multiline = True
Me.GdalInfoBox.Name = "GdalInfoBox"
Me.GdalInfoBox.ReadOnly = True
Me.GdalInfoBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
Me.GdalInfoBox.Size = New System.Drawing.Size(547, 104)
Me.GdalInfoBox.TabIndex = 2
'
'ChartFileLabel
'
Me.ChartFileLabel.AutoSize = True
Me.ChartFileLabel.Location = New System.Drawing.Point(134, 19)
Me.ChartFileLabel.Name = "ChartFileLabel"
```

```

Me.ChartFileLabel.Size = New System.Drawing.Size(98, 13)
Me.ChartFileLabel.TabIndex = 4
Me.ChartFileLabel.Text = "TIFF/PDF location"
'
'ChartFileValue
'
Me.ChartFileValue.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.ChartFileValue.Location = New System.Drawing.Point(137, 47)
Me.ChartFileValue.Name = "ChartFileValue"
Me.ChartFileValue.Size = New System.Drawing.Size(416, 15)
Me.ChartFileValue.TabIndex = 5
'
'FindButton
'
Me.FindButton.Location = New System.Drawing.Point(6, 19)
Me.FindButton.Name = "FindButton"
Me.FindButton.Size = New System.Drawing.Size(122, 43)
Me.FindButton.TabIndex = 1
Me.FindButton.Text = "&Find TIFF/PDF"
Me.FindButton.UseVisualStyleBackColor = True
'
'ExtractBox
'
Me.ExtractBox.Controls.Add(Me.JprDataBox)
Me.ExtractBox.Controls.Add(Me.ExtractButton)
Me.ExtractBox.Location = New System.Drawing.Point(12, 295)
Me.ExtractBox.Name = "ExtractBox"
Me.ExtractBox.Size = New System.Drawing.Size(568, 181)
Me.ExtractBox.TabIndex = 4
Me.ExtractBox.TabStop = False
Me.ExtractBox.Text = "Extract and evaluate JPR-data"
'
'JprDataBox
'
Me.JprDataBox.Font = New System.Drawing.Font("Consolas", 8.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.JprDataBox.Location = New System.Drawing.Point(4, 68)
Me.JprDataBox.Multiline = True
Me.JprDataBox.Name = "JprDataBox"
Me.JprDataBox.ReadOnly = True
Me.JprDataBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
Me.JprDataBox.Size = New System.Drawing.Size(549, 104)
Me.JprDataBox.TabIndex = 4
'
'ExtractButton
'
Me.ExtractButton.Location = New System.Drawing.Point(6, 19)
Me.ExtractButton.Name = "ExtractButton"
Me.ExtractButton.Size = New System.Drawing.Size(122, 43)
Me.ExtractButton.TabIndex = 3
Me.ExtractButton.Text = "&Extract JPR-data"

```

```
Me.ExtractButton.UseVisualStyleBackColor = True
'
'BuildBox
'
Me.BuildBox.Controls.Add(Me.DpiLabel)
Me.BuildBox.Controls.Add(Me.DpiBox)
Me.BuildBox.Controls.Add(Me.SaveJprFileButton)
Me.BuildBox.Controls.Add(Me.JprFileBox)
Me.BuildBox.Controls.Add(Me.BuildButton)
Me.BuildBox.Location = New System.Drawing.Point(586, 12)
Me.BuildBox.Name = "BuildBox"
Me.BuildBox.Size = New System.Drawing.Size(568, 415)
Me.BuildBox.TabIndex = 5
Me.BuildBox.TabStop = False
Me.BuildBox.Text = "Build JPR-file"
'
'DpiLabel
'
Me.DpiLabel.AutoSize = True
Me.DpiLabel.Location = New System.Drawing.Point(134, 19)
Me.DpiLabel.Name = "DpiLabel"
Me.DpiLabel.Size = New System.Drawing.Size(175, 13)
Me.DpiLabel.TabIndex = 9
Me.DpiLabel.Text = "Resolution in DPI to calculate scale"
'
'DpiBox
'
Me.DpiBox.FormattingEnabled = True
Me.DpiBox.Location = New System.Drawing.Point(134, 41)
Me.DpiBox.Name = "DpiBox"
Me.DpiBox.Size = New System.Drawing.Size(175, 21)
Me.DpiBox.TabIndex = 8
'
'SaveJprFileButton
'
Me.SaveJprFileButton.Location = New System.Drawing.Point(429, 19)
Me.SaveJprFileButton.Name = "SaveJprFileButton"
Me.SaveJprFileButton.Size = New System.Drawing.Size(122, 43)
Me.SaveJprFileButton.TabIndex = 7
Me.SaveJprFileButton.Text = "&Save JPR-file"
Me.SaveJprFileButton.UseVisualStyleBackColor = True
'
'JprFileBox
'
Me.JprFileBox.Font = New System.Drawing.Font("Consolas", 8.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.JprFileBox.Location = New System.Drawing.Point(4, 68)
Me.JprFileBox.Multiline = True
Me.JprFileBox.Name = "JprFileBox"
Me.JprFileBox.ReadOnly = True
Me.JprFileBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
```

```
Me.JprFileBox.Size = New System.Drawing.Size(547, 341)
Me.JprFileBox.TabIndex = 6
'
'BuildButton
'
Me.BuildButton.Location = New System.Drawing.Point(6, 19)
Me.BuildButton.Name = "BuildButton"
Me.BuildButton.Size = New System.Drawing.Size(122, 43)
Me.BuildButton.TabIndex = 5
Me.BuildButton.Text = "&Build JPR-file"
Me.BuildButton.UseVisualStyleBackColor = True
'
'DisclaimButton
'
Me.DisclaimButton.Location = New System.Drawing.Point(592, 433)
Me.DisclaimButton.Name = "DisclaimButton"
Me.DisclaimButton.Size = New System.Drawing.Size(122, 43)
Me.DisclaimButton.TabIndex = 8
Me.DisclaimButton.Text = "&Disclaimer"
Me.DisclaimButton.UseVisualStyleBackColor = True
'
'EndButton
'
Me.EndButton.Location = New System.Drawing.Point(1015, 433)
Me.EndButton.Name = "EndButton"
Me.EndButton.Size = New System.Drawing.Size(122, 43)
Me.EndButton.TabIndex = 9
Me.EndButton.Text = "End/Close &Program"
Me.EndButton.UseVisualStyleBackColor = True
'
'G2J
'
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.ClientSize = New System.Drawing.Size(1163, 483)
Me.Controls.Add(Me.EndButton)
Me.Controls.Add(Me.DisclaimButton)
Me.Controls.Add(Me.BuildBox)
Me.Controls.Add(Me.ExtractBox)
Me.Controls.Add(Me.FindBox)
Me.Controls.Add(Me.OSBox)
Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
Me.Name = "G2J"
Me.Text = "Gdal Data to JPR"
Me.OSBox.ResumeLayout(False)
Me.OSBox.PerformLayout()
Me.FindBox.ResumeLayout(False)
Me.FindBox.PerformLayout()
Me.ExtractBox.ResumeLayout(False)
Me.ExtractBox.PerformLayout()
```

```
Me.BuildBox.ResumeLayout(False)
Me.BuildBox.PerformLayout()
Me.ResumeLayout(False)
```

```
End Sub
```

```
Friend WithEvents BatFileLabel As Label
Friend WithEvents BatFileValue As Label
Friend WithEvents OSBox As GroupBox
Friend WithEvents OSButton As Button
Friend WithEvents FindBox As GroupBox
Friend WithEvents FindButton As Button
Friend WithEvents ChartFileLabel As Label
Friend WithEvents ChartFileValue As Label
Friend WithEvents GdalInfoBox As TextBox
Friend WithEvents ExtractBox As GroupBox
Friend WithEvents JprDataBox As TextBox
Friend WithEvents ExtractButton As Button
Friend WithEvents BuildBox As GroupBox
Friend WithEvents JprFileBox As TextBox
Friend WithEvents BuildButton As Button
Friend WithEvents SaveJprFileButton As Button
Friend WithEvents DisclaimButton As Button
Friend WithEvents EndButton As Button
Friend WithEvents DpiLabel As Label
Friend WithEvents DpiBox As ComboBox
End Class
```