

GPS_OS.bas

```

Attribute VB_Name = "GPS_OS"
Option Explicit
Rem
Rem With thanks to Chris Veness for the Latitude/Longitude to"
Rem OS coordinate conversions routines which have been translated from"
Rem his Javascript examples to this applications Visual Basic code."
Rem
Rem Ordnance Survey Grid Reference functions (c) Chris Veness 2005-2012"
Rem - www.movable-type.co.uk/scripts/gridref.js"
Rem - www.movable-type.co.uk/scripts/latlon-gridref.html"
Rem
Rem Converted to VB6 by R. J. Spriggs March 2013
Rem Programmers: The Visual Basic source code in this document may be copied
Rem and reused any for non commercial use.
Rem

Rem
Rem          Functions and Subroutines located in this Module.
Rem
Rem      Type          Name          Comment
Rem
Rem      Subroutine    CvtSG2NG      Convert Standard Grid to Numeric Grid
Rem      Subroutine    CvtNG2SG      Convert Numeric Grid to Standard Grid
Rem      Subroutine    CvtNG2LL      Convert Numeric Grid to Latitude and Longitude
Rem      Subroutine    CvtLL2NG      Convert Latitude and Longitude to Numeric Grid
Rem      Function      TidyOSCode     Will remove invalid characters from OS Code

Rem =====
Rem OS English Conversion Section
Rem =====

Rem Definitions of Common OS information stores

Rem Functions and Subroutines for OS Conversions

Public Sub CvtSG2NG(SG As String, NG As String, Offset As String, Status As String)
Rem Convert Standard Grid to Numeric Grid
Rem Valid input format is 2 Letters to Indicate Square followed by
Rem 6 8 or 10 numerics where first 3,4,5 numerics are east co-ordinate
Rem          and numerics 6,7,8 are the          north co-ordinate
Rem Offset = "5" when location is a position Reference or
Rem Offset = "0" when location is a Grid Reference

Dim Cnt As Integer          'General Counter
Dim Pnt As Integer          'General Pointer
Dim C1$, C2$, C3$          'General Strings
Dim CPnt1 As Integer        'Character Code Pointer
Dim CPnt2 As Integer        'Character Code Pointer
Dim ValE As Integer         'Easting Value
Dim ValN As Integer         'Northing Value

    'Convert Standard Grid Reference to fully Numeric Reference
    Cnt = Len(SG)           'Hold Length of Standard Grid
    C2$ = "ABCDEFGHJKLMNPOQRSTUVWXYZ" 'Hold Valid SG letters (No I)
    If Cnt = 8 Or Cnt = 10 Or Cnt = 12 Then 'Only allow 8,10,12 references
        Pnt = InStr(C2$, Mid$(SG, 1, 1)) 'Check First Code Letter
        If Pnt = 0 Then Cnt = -1 'Oh dear not a Character
        CPnt1 = Pnt - 1 'Hold 1st Char's Code
        Pnt = InStr(C2$, Mid$(SG, 2, 1)) 'Check Second Code Letter
        If Pnt = 0 Then Cnt = -1 'Oh dear not a Character
        CPnt2 = Pnt - 1 'Hold 2nd Char's Code
        C1$ = Mid$(SG, 3) 'Hold Numeric Code Section
        If IsNumeric(C1$) Then 'Check if all Numerics
            If Cnt > 0 Then
                Cnt = Cnt / 2
                C2$ = Mid$(C1$, Cnt)
                C1$ = Mid$(C1$, 1, Cnt - 1)
            End If
        Else
            Status = "Error with Digits part " + C1$
        End If
    End If
End Sub

```

```

                                GPS_OS.bas
        Exit Sub                                'Oh dear not all digits
    End If
Else
    Status = "Error Not 6,8 or 10 Digits"
    Exit Sub                                'Oh dear invalid digit count
End If

If Cnt < 0 Then
    Status = "Error problem with Grid Letters " + Mid$(SG, 1, 2)
    Exit Sub                                'Oh dear invalid grid letters
Else
    Status = ""
    'Assume all is OK
End If

Vale = ((CPnt1 - 2) Mod 5) * 5 + (CPnt2 Mod 5)
ValN = (19 - (Int(CPnt1 / 5) * 5)) - Int(CPnt2 / 5)
If Vale < 0 Or Vale > 6 Or ValN < 0 Or ValN > 12 Then
    Status = "Error Outside UK Grid Area"
    Exit Sub                                'Oh dear Square code
End If

'   Status = "Diag 1 " + Str$(CPnt1) + " - " + C1$ + " , " + Str$(CPnt2) + " - " + C2$
+ Chr$(10)
'If OSCentre.Value = vbChecked Then C3$ = "5" Else C3$ = "0"
C3$ = Offset                                'Distinguish between Reference and Grid position
If Len(C1$) = 3 Then C1$ = C1$ + C3$ + "0": C2$ = C2$ + C3$ + "0"
If Len(C1$) = 4 Then C1$ = C1$ + C3$: C2$ = C2$ + C3$

'   Status = Status + "Diag 2 " + Str$(Vale) + C1$ + " , " + Str$(ValN) + C2$ +
Chr$(10)
    NG = Mid$(Str$(Vale), 2) + C1$ + "," + Mid$(Str$(ValN), 2) + C2$

End Sub

Public Sub CvtNG2SG(SG As String, NG As String, Status As String)
Rem Convert Numeric Grid to Standard Grid
Rem Valid input format is 6 numerics <comma> 6+ numerics
Rem where first 6 numerics are east co-ordinate
Rem      and the rest      are north co-ordinate
'Dim Cnt As Integer                                'General Counter
Dim Pnt As Integer                                'General Pointer
Dim Vale As Long                                  'Easting Value
Dim ValN As Long                                  'Northing Value
Dim E100K As Single                               '100K East Value
Dim N100K As Single                               '100K North Value
Dim e$, n$, C1$, C2$                              'General Strings

    Pnt = InStr(NG, ",")                            'Locate the Comma Delimiter
    If Pnt < 2 Or Pnt = Len(NG) Then
        Status = "Error with Numeric Grid format"
        Exit Sub                                'Oh dear invalid delimiter
    Else
        Status = ""
        'Assume all is OK
    End If
    e$ = Mid$(NG, 1, Pnt - 1): Vale = Val(e$)        'Extract East reference
    n$ = Mid$(NG, Pnt + 1): ValN = Val(n$)          'Extract North reference
    E100K = Int(Vale / 100000)                       'Get 100K Grid Index
    N100K = Int(ValN / 100000)                       'Get 100K Grid Index
    If E100K < 0 Or E100K > 6 Or N100K < 0 Or N100K > 12 Then
        Status = "Error Outside UK Grid Area"
        Exit Sub                                'Oh dear Square code problem
    End If
    C2$ = "ABCDEFGHJKLMNPOQRSTUVWXYZ"              'Hold Valid SG letters (No I)
    Pnt = (19 - N100K) - ((19 - N100K) Mod 5) + Int((E100K + 10) / 5)
    C1$ = Mid$(C2$, Pnt + 1, 1)                    'Extract first SG letter
    Pnt = (19 - N100K) * 5 Mod 25 + E100K Mod 5
    C1$ = C1$ + Mid$(C2$, Pnt + 1, 1)              'Extract second SG letter
    If N100K <= 9 Then
        n$ = Mid$(n$ + "00000", 2, 5)              'Extract North Coordinate
    End If

```

GPS_OS.bas

```

Else
    n$ = Mid$(n$ + "00000", 3, 5)           'Extract North Coordinate
End If
e$ = Mid$(e$ + "00000", 2, 5)           'Extract East Coordinate
SG = C1$ + " " + e$ + " " + n$         'Hold Converted Coordinate

End Sub

Public Sub CvtNG2LL(NG As String, Lat As Double, Lng As Double, Status As String)
Rem Convert Numeric Grid to Latitude and Longitude (in degrees)
Dim Pnt As Integer                       'General Pointer
Dim e As Double                           'Easting Value
Dim n As Double                           'Northing Value
Dim A As Double                           'Airy 1830 major & minor semi-axes
Dim B As Double                           'Airy 1830 major & minor semi-axes
Dim F0 As Double                          'NatGrid scale factor on central meridian
Dim lat0 As Double                        'Work Variable
Dim lon0 As Double                        'Work Variable
Dim N0 As Double                          'Work Variable
Dim E0 As Double                          'Work Variable
Dim e2 As Double                          'Work Variable
Dim nn As Double                          'Work Variable to replace var n
Dim n2 As Double                          'Work Variable
Dim n3 As Double                          'Work Variable
Dim M As Double                           'Work Variable
Dim MA As Double                          'Work Variable
Dim Mb As Double                          'Work Variable
Dim Mc As Double                          'Work Variable
Dim Md As Double                          'Work Variable

    Pnt = InStr(NG, ",")                  'Locate the Comma Delimiter
    If Pnt < 2 Or Pnt = Len(NG) Then
        Status = "Error with Numeric Grid format"
        Exit Sub                          'Oh dear invalid delimiter
    Else
        Status = ""                       'Assume all is OK
    End If
    e = Val(Mid$(NG, 1, Pnt - 1))         'Extract East reference
    n = Val(Mid$(NG, Pnt + 1))           'Extract North reference

    A = 6377563.396                       'Airy 1830 major & minor semi-axes
    B = 6356256.91                        'Airy 1830 major & minor semi-axes
    F0 = 0.9996012717                     'NatGrid scale factor on central meridian

    lat0 = 49 * Pi / 180: lon0 = -2 * Pi / 180 'NatGrid true origin (in radians)
    N0 = -100000: E0 = 400000             'northing & easting of true origin, metres
    e2 = 1 - (B * B) / (A * A)            'eccentricity squared
    nn = (A - B) / (A + B): n2 = nn * nn: n3 = nn * nn * nn

    Lat = lat0: M = 0
    Do
        Lat = (n - N0 - M) / (A * F0) + Lat
        MA = (1 + nn + (5 / 4) * n2 + (5 / 4) * n3) * (Lat - lat0)
        Mb = (3 * nn + 3 * nn * nn + (21 / 8) * n3) * Sin(Lat - lat0) * Cos(Lat + lat0)
        Mc = ((15 / 8) * n2 + (15 / 8) * n3) * Sin(2 * (Lat - lat0)) * Cos(2 * (Lat +
lat0))
        Md = (35 / 24) * n3 * Sin(3 * (Lat - lat0)) * Cos(3 * (Lat + lat0))
        M = B * F0 * (MA - Mb + Mc - Md)   'meridional arc
        'DoEvents                          'Allow interrupts
    Loop While (n - N0 - M) >= 0.00001     'ie until < 0.01mm

Dim cosLat As Double                     'More Work Variables
Dim sinLat As Double                     'More Work Variables
Dim tanLat As Double                     'More Work Variables
Dim tan2Lat As Double                    'More Work Variables
Dim tan4Lat As Double                    'More Work Variables
Dim tan6Lat As Double                    'More Work Variables
Dim seclat As Double                     'More Work Variables
Dim nu As Double                         'More Work Variables
Dim nu3 As Double                        'More Work Variables

```

```

GPS_OS.bas
Dim nu5 As Double 'More Work Variables
Dim nu7 As Double 'More Work Variables
Dim rho As Double 'More Work Variables
Dim eta2 As Double 'More Work Variables

cosLat = Cos(Lat): sinLat = Sin(Lat)
nu = A * F0 / Sqr(1 - e2 * sinLat * sinLat) 'tranverse radius of curvature
' meridional radius of curvature

rho = A * F0 * (1 - e2) / ((1 - e2 * sinLat * sinLat) ^ 1.5)
eta2 = nu / rho - 1

tanLat = Tan(Lat)
tan2Lat = tanLat * tanLat: tan4Lat = tan2Lat * tan2Lat: tan6Lat = tan4Lat * tan2Lat
seclat = 1 / cosLat
nu3 = nu * nu * nu: nu5 = nu3 * nu * nu: nu7 = nu5 * nu * nu

Dim VII As Double 'More Work Variables
Dim VIII As Double 'More Work Variables
Dim IX As Double 'More Work Variables
Dim x As Double 'More Work Variables
Dim XI As Double 'More Work Variables
Dim XII As Double 'More Work Variables
Dim XIII As Double 'More Work Variables
Dim dE As Double 'More Work Variables
Dim dE2 As Double 'More Work Variables
Dim dE3 As Double 'More Work Variables
Dim dE4 As Double 'More Work Variables
Dim dE5 As Double 'More Work Variables
Dim dE6 As Double 'More Work Variables
Dim dE7 As Double 'More Work Variables

VII = tanLat / (2 * rho * nu)
VIII = tanLat / (24 * rho * nu3) * (5 + 3 * tan2Lat + eta2 - 9 * tan2Lat * eta2)
IX = tanLat / (720 * rho * nu5) * (61 + 90 * tan2Lat + 45 * tan4Lat)
x = seclat / nu
XI = seclat / (6 * nu3) * (nu / rho + 2 * tan2Lat)
XII = seclat / (120 * nu5) * (5 + 28 * tan2Lat + 24 * tan4Lat)
XIII = seclat / (5040 * nu7) * (61 + 662 * tan2Lat + 1320 * tan4Lat + 720 * tan6Lat)
dE = (e - E0): dE2 = dE * dE: dE3 = dE2 * dE: dE4 = dE2 * dE2
dE5 = dE3 * dE2: dE6 = dE4 * dE2: dE7 = dE5 * dE2
Lat = Lat - VII * dE2 + VIII * dE4 - IX * dE6
Lat = R2D(Lat)
Lng = lon0 + x * dE - XI * dE3 + XII * dE5 - XIII * dE7
Lng = R2D(Lng)
End Sub

Public Sub CvtLL2NG(NG As String, lt As Double, Lng As Double, Status As String)
Rem Convert Latitude and Longitude (in degrees) to Numeric Grid

Dim Lat As Double 'Work Variable
Dim lon As Double 'Work Variable
Dim lat0 As Double 'Work Variable
Dim lon0 As Double 'Work Variable
Dim A As Double 'Airy 1830 major & minor semi-axes
Dim B As Double 'Airy 1830 major & minor semi-axes
Dim F0 As Double 'NatGrid scale factor on central meridian
Dim N0 As Double 'Work Variable
Dim E0 As Double 'Work Variable
Dim e2 As Double 'Work Variable
Dim nn As Double 'Work Variable to replace var n
Dim n2 As Double 'Work Variable
Dim n3 As Double 'Work Variable

If lt < 49 Or lt > 61.465173 Or Lng < -7.56 Or Lng > 3.634722 Then
Status = "Error Outside OS Grid"
Exit Sub 'Oh dear invalid coordinate
Else
Status = "" 'Lat/Long should be OK
End If

```

GPS_OS.bas

```

Lat = lt * Pi / 180          'Hold Latitude in Radians
lon = Lng * Pi / 180       'Hold Longitude in Radians

A = 6377563.396             'Airy 1830 major & minor semi-axes
B = 6356256.91             'Airy 1830 major & minor semi-axes
F0 = 0.9996012717          'NatGrid scale factor on central meridian

lat0 = 49 * Pi / 180: lon0 = -2 * Pi / 180 'NatGrid true origin (in radians)
N0 = -100000: E0 = 400000 'northing & easting of true origin, metres
e2 = 1 - (B * B) / (A * A) 'eccentricity squared
nn = (A - B) / (A + B): n2 = nn * nn: n3 = nn * nn * nn

Dim cosLat As Double       'More Work Variables
Dim sinLat As Double       'More Work Variables
Dim nu As Double           'More Work Variables
Dim rho As Double         'More Work Variables
Dim eta2 As Double         'More Work Variables
Dim M As Double            'More Work Variable
Dim MA As Double           'More Work Variable
Dim Mb As Double           'More Work Variable
Dim Mc As Double           'More Work Variable
Dim Md As Double           'More Work Variable

cosLat = Cos(Lat): sinLat = Sin(Lat)
nu = A * F0 / Sqr(1 - e2 * sinLat * sinLat) 'transverse radius of curvature
rho = A * F0 * (1 - e2) / ((1 - e2 * sinLat * sinLat) ^ 1.5) 'meridional radius of curvature
eta2 = nu / rho - 1

MA = (1 + nn + (5 / 4) * n2 + (5 / 4) * n3) * (Lat - lat0)
Mb = (3 * nn + 3 * nn * nn + (21 / 8) * n3) * Sin(Lat - lat0) * Cos(Lat + lat0)
Mc = ((15 / 8) * n2 + (15 / 8) * n3) * Sin(2 * (Lat - lat0)) * Cos(2 * (Lat + lat0))
Md = (35 / 24) * n3 * Sin(3 * (Lat - lat0)) * Cos(3 * (Lat + lat0))
M = B * F0 * (MA - Mb + Mc - Md) 'meridional arc

Dim cos3Lat As Double      'More Work Variables
Dim cos5Lat As Double      'More Work Variables
Dim tan2Lat As Double      'More Work Variables
Dim tan4Lat As Double      'More Work Variables

cos3Lat = cosLat * cosLat * cosLat
cos5Lat = cos3Lat * cosLat * cosLat
tan2Lat = Tan(Lat) * Tan(Lat)
tan4Lat = tan2Lat * tan2Lat

Dim I As Double            'More Work Variables
Dim II As Double           'More Work Variables
Dim III As Double          'More Work Variables
Dim IIIA As Double         'More Work Variables
Dim IV As Double           'More Work Variables
Dim V As Double            'More Work Variables
Dim VI As Double           'More Work Variables
I = M + N0
II = (nu / 2) * sinLat * cosLat
III = (nu / 24) * sinLat * cos3Lat * (5 - tan2Lat + 9 * eta2)
IIIA = (nu / 720) * sinLat * cos5Lat * (61 - 58 * tan2Lat + tan4Lat)
IV = nu * cosLat
V = (nu / 6) * cos3Lat * (nu / rho - tan2Lat)
VI = (nu / 120) * cos5Lat * (5 - 18 * tan2Lat + tan4Lat + 14 * eta2 - 58 * tan2Lat *
eta2)

Dim dLon As Double         'More Work Variables
Dim dLon2 As Double        'More Work Variables
Dim dLon3 As Double        'More Work Variables
Dim dLon4 As Double        'More Work Variables
Dim dLon5 As Double        'More Work Variables
Dim dLon6 As Double        'More Work Variables
dLon = lon - lon0
dLon2 = dLon * dLon: dLon3 = dLon2 * dLon: dLon4 = dLon3 * dLon
dLon5 = dLon4 * dLon: dLon6 = dLon5 * dLon

```

GPS_OS.bas

```

Dim n As Double           'More Work Variables
Dim e As Double           'More Work Variables
n = I + II * dLon2 + III * dLon4 + IIIA * dLon6
e = E0 + IV * dLon + V * dLon3 + VI * dLon5
Dim Tmp$                  'More Work Variables
Tmp$ = Mid$(Str$(Int(e + 0.5)), 2) 'Format NG to (e*6+),(n*6+)
While Len(Tmp$) < 6: Tmp$ = "0" + Tmp$: Wend
NG = Tmp$
Tmp$ = Mid$(Str$(Int(n + 0.5)), 2)
While Len(Tmp$) < 6: Tmp$ = "0" + Tmp$: Wend
NG = NG + "," + Tmp$

End Sub

Rem =====
Rem Utility Routines
Rem =====

Public Function TidyOSCode(Disp As String)
Rem This routine will remove any spaces and commas from OS Style Code field
Rem Initial Design Mar/2013 Author R. J. Spriggs
Rem Mod 23/06/13 RJS Converted from Subroutine to Function for more universal use.
Dim Pnt
Dim A$, B$, C$, D$

A$ = "": B$ = UCase$(Disp)
D$ = "0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ"
For Pnt = 1 To Len(B$)
    C$ = Mid$(B$, Pnt, 1)
    If InStr(D$, C$) <> 0 Then A$ = A$ + C$ 'Hold Current Character
                                         'Keep Character when valid
Next Pnt
TidyOSCode = A$ 'Reply with corrected OSCode
End Function

```