

The binary raster file format is a simple format that can be used to transfer raster data between various applications. It consists of two files, the IEEE floating-point file and a supporting ASCII header file. The header file must have the same name as the data file, but with a .hdr file extension. The header data includes the following keywords and values:

- ncols - number of columns in the data set.
- nrows - number of rows in the data set.
- xllcenter or xllcorner - x-coordinate of the center or lower-left corner of the lower-left cell.
- yllcenter or yllcorner - y-coordinate of the center or lower-left corner of the lower-left cell.
- cellsize - cell size for the data set.
- nodata\_value - value in the file assigned to cells whose value is unknown. This keyword and value is optional. The nodata\_value defaults to -9999.
- 
- byteorder - the byte order of the binary cell values. You can choose between two keywords, msbfirst or lsbfirst. Msbfirst is used for cell values written with the most significant bit first. Lsbfirst is used for cell values written with the least significant bit first.

For example,

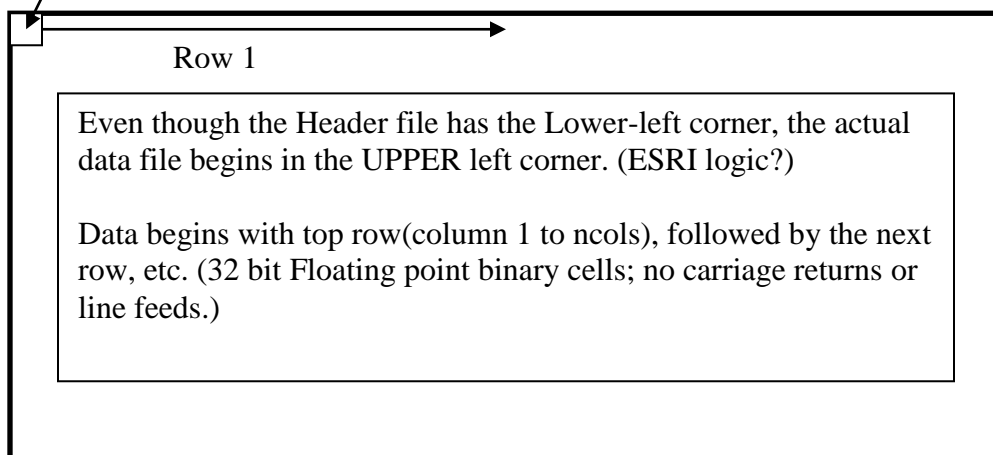
```
ncols      43200
nrows      20880
xllcorner  -180
yllcorner  -90
cellsize   0.008333333333333333
NODATA_value -9999
byteorder  LSBFIRST
```

The data file will be a matrix of 32 bit signed IEEE floating-point values. The file will have a line of binary numbers for each row in the data set. The first line of data is the top row of the data set, moving from left to right.

---

The above is the ESRI Help file for the binary grid format. The following is a short interpretation of what this really means:

(1,1) First data point at Lon, Lat: -179.991666, 83.991666



Each cell value represents the Population count at the cell center. The (xllcorner,yllcorner) is the **outside** lower left corner of the grid; must move a half cell to the right and up to get to the cell center at the lower left corner of the grid. But since the first data value is the upper left corner of the grid, its cell center would be:

$$(xllcorner + (cellsize/2), yllcorner + ((cellsize * (nrows-1)) + cellsize/2))$$

### Sample Visual Basic Program to read the Grid:

```
Dim xllcorner, yllcorner, NoData, CellSize, Half
Dim Npts As Long, Ncols As Long, Nrows As Long
Dim grd() As Single, Lat() As Single, Lon() As Single
.....
```

```
Public Sub ReadGrid(fil, grd() As Long, Ncols, Nrows)
    headr = Left(fil, Len(fil) - 3) + "hdr"
```

```
    Open headr For Input As #1
```

```
    Line Input #1, a
```

```
    Var = StringExtract(a, 1)
```

```
    v = StringExtract(a, 2)
```

```
    Ncols = Val(v)
```

```
    Line Input #1, a
```

```
    Var = StringExtract(a, 1)
```

```
    v = StringExtract(a, 2)
```

```
    Nrows = Val(v)
```

```
    Line Input #1, a
```

```
    Var = StringExtract(a, 1)
```

```
    v = StringExtract(a, 2)
```

```
    xllcorner = Val(v)
```

```
    Line Input #1, a
```

```
    Var = StringExtract(a, 1)
```

```
    v = StringExtract(a, 2)
```

```
    yllcorner = Val(v)
```

```
    Line Input #1, a
```

```
    Var = StringExtract(a, 1)
```

```
    v = StringExtract(a, 2)
```

```
    CellSize = Val(v)
```

```
    Line Input #1, a
```

```
    Var = StringExtract(a, 1)
```

```
    v = StringExtract(a, 2)
```

```
    NoData = Val(v)
```

```
    Close #1
```

```
    Open fil For Binary Access Read As #1
```

```
    ReDim grd(Nrows, Ncols)
```

```
    ReDim Lat(Nrows), Lon(Ncols)
```

```
    ' Loop for illustration of data order
```

```
    Half=cellsize/2
```

```
    For r = 1 To Nrows
```

```

Lat (r)= yllcorner + (cellsize * (Nrows-(r-1))) - Half
For c = 1 To Ncols
    Lon(c )= xllcorner +(cellsize * (c-1)) + Half
    Get #1, , grd(r, c)
Next
Next
Close #1
End Sub

```

```

Public Function StringExtract(str, iplace As Integer) As String
' Extracts the String # iplace in s; Blanks are separators
Dim s As String, i As Integer, blnk As Integer
s = str
i = 0
s = Trim(s)
blnk = InStr(s, " ")
Do While blnk > 0
    i = i + 1
    If i = iplace Then
        StringExtract = Left(s, blnk - 1)
        Exit Function
    Else
        s = LTrim(Mid(s, blnk))
    End If
    blnk = InStr(s, " ")
Loop
i = i + 1
If i = iplace Then
    StringExtract = s
Else
    StringExtract = ""
End If

End Function

```