Place File Specification

Place files are user-supplied text files containing drawing statements. They can be on a local file system or given as a URL. When a URL is requested, the latitude and longitude of the radar site being viewed are included as parameters (e.g. ?lat=nnn.nnn&lon=nnn.nnn) along with the placefile version supported (e.g. &version=1.2).

Threshold: ***nautical_miles***

nautical_miles
Sets the view threshold in miles for subsequent Place statements. The default threshold is 999.

Color: ***red green blue***

The **Color** statement sets the default color of subsequent **Place** statements. The default color is white (255 255 255).

Refresh: ***minutes***

minutes
Specifies the amount of time in minutes until the placefile is reloaded.

RefreshSeconds: ***seconds***

seconds
Specifies the amount of time in seconds until the placefile is reloaded.

IconFile: ***fileNumber, iconWidth, iconHeight, hotX, hotY, fileName***

fileNumber:
This is a number from 1 to 8 used to reference the file in subsequent Icon statements. The number cannot be used in other IconFile statements.

iconWidth and iconHeight:
width and height of each icon in the file (pixels)

hotX, hotY:
zero-based center of the icon, used for placement and as the center of rotation. Must be less than iconWidth, iconHeight

fileName:
local, network, or URL filename (PNG, JPG, TIF supported) if no alpha channel is in the file, solid black (0,0,0) is made transparent

---

Icon: ***lat, lon, angle, fileNumber, iconNumber, hoverText***

lat, lon:

coordinates where the icon hotspot is placed

angle:
clockwise angle of rotation of the icon (degrees)

fileNumber:
fileNumber from earlier IconFile statement

iconNumber:
number of the icon to draw from 1 to the number of icons in the icon file. Icons in an icon file are assigned by row from left to right, top to bottom.

hoverText (optional):
text displayed when the mouse cursor hovers over the lat, lon of the icon

---

Font: ***fontNumber, pixels, flags, "face"***

fontNumber:
number to assign to the font, from 1 to 8. Font numbers cannot be used in any other Font statement

pixels:
pixel height of the font

flags:
font attribute flags: 1 means bold, 2 means italic

face:
face name of the font to use.

Text: ***lat, lon, fontNumber, "string", "hover"***

lat, lon:
coordinates for the center of the text string

fontNumber:
fontNumber from earlier Font statement

string:
text to display

hoverText (optional):
text displayed when the mouse cursor hovers over the lat, lon of the string

---

Object: ***lat, lon***
...
End:

lat, lon:
center of composite object location.

An Object statement must be paired with an End statement. All

other statements can appear in between the pair. When inside, the lat, lon values are interpreted as x, y offsets in pixels from the center of the object. Positive offsets are to the right and towards the top of the screen.

Line: ***width, flags [, hover_text]***
lat, lon
...
End:

width
width of line in pixels

flags
unused currently

hover_text
optional text to display when user hovers mouse cursor over the line

lat, lon
coordinates of point on line

The Line statement draws a line along the list of lat,lon points. There can be up to 10,000 points in each line. The color of the line is set by a previous Color: statement.

Triangles:
lat, lon [, r, g, b [,a]]
...
End:

lat, lon
coordinates of a vertex of a triangle
r, g, b, a
optional red, green, blue, alpha color for the vertex, 0 to 255

The Triangles statement draws triangles from groups of three color vertices. Vertex colors are smoothly interpolated the across the triangle faces.

Image: ***image_file***
lat, lon, Tu [, Tv]
...
End:

image_file
image file for texture: PNG, JPG, TGA supported

lat, lon

coordinates of a vertex of a triangle

Tu, Tv
coordinates of image for the vertex where
0.0, 0.0 is the upper-left corner of the image
1.0, 1.0 is the lower-right corner of the image

The Image statement draws triangles from groups of three textured vertices. Each vertex has texture coordinates that specify where in the image_file to retrieve the color. GR3 smoothly interpolates Tu,Tv across the triangle faces.

Polygon:
lat1, lon1 [, r, g, b [,a]] ; start of the first contour
...
lat1, lon1 ; repeating the first point closes the contour

lat2, lon2 ; next point starts a new contour
...
lat2, lon2 ; and repeating it ends the contour
End:

lat, lon
coordinates of a vertex of a triangle

The Polygon statement specifies outer and inner lines that make the boundary of a filled shape. The polygon is filled in ODD winding mode. More information on contours and winding modes can be found here. Each point on the contour can have its own RGBA color and the colors are smoothly interpolated across the polygon interior.

Old Place File specification

The Place file is a simple text file containing single line statements using the following syntax:

```
;
; Everything after ';' is a comment
;

Threshold: miles
Color:  red  green  blue
Place: latitude, longitude, string with spaces
```

Color: ***red green blue***

The **Color** statement sets the display color of subsequent **Place** statements. The default color is white (255 255 255).

Place: ***latitude, longitude, string***

The **Place** statement gives the location and string to display. Commas separate each field. The string can contain spaces and is trimmed of leading and trailing spaces before display.

Example

Here is a Place file containing some radar sites and METAR stations:

```
;
; Show some radar sites at all zoom levels in white
;

Threshold: 999
Color: 255 255 255

Place: 33.172, -86.770, KBMX
Place: 31.460, -85.459, KEOX
Place: 34.931, -86.084, KHTX
Place: 32.537, -85.790, KMXX
Place: 30.679, -88.240, KMOB

;
; And some METAR stations at 100 miles in green
;

Threshold: 100
Color: 0 255 0

Place: 35.93,  -95.00, KTQH
Place: 35.94,  -89.83, KHKA
Place: 35.95, -112.15, KGCN
Place: 35.95,  -85.08, KCSV
Place: 35.97,  -89.95, KBYH
```