

Mapfile Reference

Mapfile Reference

Document Actions

- 
- 

Note: This is the print view with all the Reference Manual pages on one page. [The paginated version is available here](#), if you prefer that.

MapFiles are the basic configuration mechanism for MapServer. This document contains a rundown of the keys and values that describe a mapfile.

1. Introduction

Important notes on using mapfiles.

The Mapfile is the heart of MapServer. It defines the relationships between objects, points MapServer to where data are located and defines how things are to be drawn.

There are some important concepts that you must understand before you can reliably use mapfiles to configure MapServer. First is the concept of a layer. A layer is the combination of data plus styling. Data, in the form of attributes plus geometry, are given styling using CLASS and STYLE directives.

Notes

- The Mapfile is NOT case-sensitive.
- Strings containing non-alphanumeric characters or a MapServer keyword MUST be quoted. It is recommended to put ALL strings in double-quotes.
- There is a maximum of 50 layers per mapfile. This can be changed by editing the map.h file to change the value of MS_MAXLAYERS to the desired number and recompiling.
- File paths may be given as absolute paths, or as paths relative to the location of the mapfile. In addition, data files may be specified relative to the SHAPEPATH.
- The mapfile has a hierarchical structure, with the Map object being the "root". All other objects fall under this one.
- Comments are designated with a #.
- Attributes are named using the following syntax: [ATTRIBUTENAME] ... Note that the name of the attribute included between the square brackets *IS CASE SENSITIVE*. Generally ESRI generated shapefiles have their attributes (.dbf column names) all in upper-case for instance, and for PostGIS, *ALWAYS* use lower-case.
- MapServer Regular Expressions are used through the operating system's C Library. For information on how to use and write Regular Expressions on your system, you should read the documentation provided with your C Library. On Linux, this is GLibC, and you can read "man 7 regex" ... This man page is also available on most UNIX's. Since these RegEx's are POSIX compliant, they should be the same on Windows as well, so windows users can try searching the web for "man 7 regex" since man pages are available all over the web.

2. Class

Defines thematic classes for a given layer and each layer must have at least one class. In cases with more than one class, membership is determined using attribute values and expressions. Starts with the keyword CLASS and terminates with the keyword END.

BACKGROUNDCOLOR [r] [g] [b]

Color to use for non-transparent symbols.

COLOR [r] [g] [b]

Color to use for drawing features.

DEBUG [onoff]

Enables debugging of the class object. Verbose output is generated and sent to the standard error output (STDERR) or the MapServer r logfile if one is set using the LOG parameter in the WEB object.

EXPRESSION [string]

Four types of expressions are now supported to define class membership. String comparisons, regular expressions, simple logical expressions, and string functions. If no expression is given, then all features are said to belong to this class.

- ◊ String comparisons are case sensitive and are the fastest to evaluate. No special delimiters are necessary although string must be quoted if they contain special characters. (As a matter of good habit, it is recommended you quote all strings).
- ◊ Regular expressions function just like previous versions of MapServer. However, you must now delimit a regular expression using /regex/. No quotes should be used.
- ◊ Logical expressions allow you to build fairly complex tests based on one or more attributes and therefore are only available with shapefiles. Logical expressions are delimited by parentheses "(expression)". Attribute names are delimited by square brackets "[ATTRIBUTE]". These names are case sensitive and must match the items in the shapefile. For example: EXPRESSION ([POPULATION] > 50000 AND '[LANGUAGE]' eq 'FRENCH') ... The following logical operators are supported:
=,>,<,<=,>=,=,or,and,lt,gt,ge,le,eq,ne. As you might expect this level of complexity is slower to process.
- ◊ One string function exists: length(). This obviously computes the length of a string. An example follows:

```
EXPRESSION (length(' [NAME_E] ') < 8)
```

String comparisons and regular expressions work from the classitem defined at the layer level. You may mix expression types within the different classes of a layer.

JOIN

Signals the start of a JOIN object.

KEYIMAGE [filename]

Full filename of the legend image for the CLASS. This image is used when building a legend (or requesting a legend icon via MapScript or the CGI application).

LABEL

Signals the start of a LABEL object.

MAXSCALE [double]

Maximum scale at which this CLASS is drawn.

MAXSIZE [integer]

Maximum size in pixels to draw a symbol. Default is 50.

MINSCALE [double]
Minimum scale at which this CLASS is drawn.

MINSIZE [integer]
Minimum size in pixels to draw a symbol. Default is 0.

NAME [string]
Name to use in legends for this class. If not set class won't show up in legend.

OUTLINECOLOR [r] [g] [b]
Color to use for outlining polygons and certain marker symbols. Line symbols do not support outline colors.

SIZE [integer]
Height, in pixels, of the symbol/pattern to be used. Only useful with scalable symbols. For vector (and ellipse) symbol types the default size is based on the range of Y values in the POINTS defining the symbol. For pixmaps, the default is the vertical size of the image. Default size is 1 for TTF symbols.

STYLE
Signals the start of a STYLE object. A class can contain multiple styles.

SYMBOL [integer|string|filename]
The symbol name or number to use for all features if attribute tables are not used. The number is the index of the symbol in the symbol file, starting at 1, the 5th symbol in the file is therefore symbol number 5. You can also give your symbols names using the NAME keyword in the symbol definition file, and use those to refer to them. Default is 0, which results in a single pixel, single width line, or solid polygon fill, depending on layer type.

You can also specify a gif or png filename. The path is relative to the location of the mapfile.

TEMPLATE [filename]
Template file or URL to use in presenting query results to the user.

TEXT [string]
Static text to label features in this class with. This overrides values obtained from the LABELITEM. The string may be given as an expression delimited using the ()'s. This allows you to concatenate multiple attributes into a single label. For example: ([FIRSTNAME],[LASTNAME]).

You can also "stack" 2 symbols to achieve interesting effects. You define the second symbol, which effectively sits "on top" of the symbol normally defined above.

The following parameters allow you to define the symbol, and they are equivalent to their non-overlay counterparts:

- OVERLAYBACKGROUND COLOR
- OVERLAY COLOR
- OVERLAY OUTLINE COLOR
- OVERLAY SIZE
- OVERLAY MIN SIZE
- OVERLAY MAX SIZE
- OVERLAY SYMBOL

3. Feature

Defines inline features. You can use inline features when it's not possible (or too much trouble) to create a shapefile. Inline features can also be built via urls or forms. Starts with the keyword FEATURE and terminates with the keyword END.

POINTS

A set of xy pairs terminated with an END, for example:

```
POINTS 1 1 50 50 1 50 1 1 END
```

Note that with POLYGON/POLYLINE layers POINTS must start and end with the same point (i.e. close the feature).

TEXT [string]

String to use for labeling this feature.

WKT [string]

A geometry expressed in OpenGIS Well Known Text geometry format. This feature is only supported if MapServer is built with OGR or GEOS support.

```
WKT "POLYGON((500 500, 3500 500, 3500 2500, 500 2500, 500 500))"  
WKT "POINT(2000 2500)"
```

Note

Inline features should be defined as their own layers in the mapfile. If another CONNECTIONTYPE is specified in the same layer, MapServer will always use the inline features to draw the layer and ignore the other CONNECTIONTYPES.

4. Grid

The GRID object defines a map graticule as a LAYER. Starts with the keyword GRID and terminates with the keyword END.

LABELFORMAT [DDMM|DDMMSS]

Format of the label. "DDMM" for degrees minutes, and "DDMMSS" for degrees, minutes, seconds.

The default is decimal display of whatever SRS you're rendering the GRID with.

MINARCS [double]

The minimum number of arcs to draw. Use this parameter to get more lines. Optional.

MAXARCS [double]

The maximum number of arcs to draw. Use this parameter to get fewer lines. Optional.

MININTERVAL [double]

The minimum number of intervals to try to use. The distance between the grid lines, in the units of the grid's coordinate system. Optional.

MAXINTERVAL [double]

The maximum number of intervals to try to use. The distance between the grid lines, in the units of the grid's coordinate system. Optional.

MINSUBDIVIDE [double]

The minimum number of segments to use when rendering an arc. If the lines should be very curved, use this to smooth the lines by adding more segments. Optional.

MAXSUBDIVIDE [double]

The maximum number of segments to use when rendering an arc. If the graticule should be very straight, use this to minimize the number of points for faster rendering. Optional, default 256.

The following is an example of a GRID object in use:

```
LAYER  
  NAME "grid"  
  METADATA  
    "DESCRIPTION" "Grid"  
  END
```

```

TYPE LINE
STATUS ON
CLASS
  NAME "Graticule"
  COLOR 0 0 0
LABEL
  COLOR 255 0 0
  FONT fritqat
  TYPE truetype
  SIZE 8
  POSITION AUTO
  PARTIALS FALSE
  BUFFER 5
  OUTLINECOLOR 255 255 255
END
END
PROJECTION
  "init=epsg:4326"
END
GRID
LABELFORMAT DDM
MAXARCS 10
MAXINTERVAL 10
MAXSUBDIVIDE 2
END
END # Layer

```

5. Include

Defines a file to be included in the mapfile parsing.

When this directive is encountered parsing switches to the included file immediately. As a result the included file can be comprised of any valid mapfile syntax. For example:

```
INCLUDE 'myLayer.map'
```

Performance does not seem to be seriously impacted with limited use, however in high performance instances you may want to use includes in a pre-processing step to build a production mapfile. The C pre-processor can also be used (albeit with a different syntax) and is far more powerful.

Notes

- Supported in versions 4.10 and higher.
- The name of the file to be included **MUST be quoted** (single or double quotes).
- Includes may be nested, up to 5 deep.
- File location is relative to the location of the mapfile. You may, of course specify a full path.
Windows users must always specify full paths however (4.10.0-beta2).
- Debugging can be problematic since: 1) the file an error occurs in does not get output to the user and 2) the line number counter is not reset for each file. Here is one possible error that is thrown when the include file cannot be found:

```
msyylex(): Unable to access file. Error opening included file "parks_include.map"
```

Example

```

MAP
  NAME 'include'
  EXTENT 0 0 500 500

```

```

SIZE 250 250

INCLUDE "test_include_symbols.map"
INCLUDE "test_include_layer.map"
END

```

where test_include_symbols.map contains:

```

SYMBOL
  NAME 'square'
  TYPE VECTOR
  FILLED TRUE
  POINTS 0 0 0 1 1 1 0 0 0 END
END

```

and test_include_layer.map contains:

```

LAYER
  TYPE POINT
  STATUS DEFAULT
  FEATURE
    POINTS 10 10 40 20 300 300 400 10 10 400 END
  END
  CLASS
    NAME 'Church'
    COLOR 0 0 0
    SYMBOL 'square'
    SIZE 7
    STYLE
      SYMBOL "square"
      SIZE 5
      COLOR 255 255 255
    END
    STYLE
      SYMBOL "square"
      SIZE 3
      COLOR 0 0 255
    END
  END
END

```

6. Join

Defines how a specific join is handled. Starts with the keyword JOIN and terminates with the keyword END. Joins are defined within a **LAYER** object. It is important to understand that JOINS are *ONLY* available once a query has been processed. You cannot use joins to affect the look of a map. The primary purpose is to enable lookup tables for coded data (e.g. 1 => Forest) but there are other possible uses.

Please note this particular documentation is in a state of flux while I do a bit of source code review... Steve Lime

```

CONNECTION [string]
  Todo...
CONNECTIONTYPE [string]
  Todo...
FROM [item]
  Join item in the dataset.

```

NAME [string]

Unique name for this join. Required.

TABLE [filename|tablename]

For file-based joins this is the name of XBase or comma delimited file (relative to the location of the mapfile) to join TO. For MySQL support this is the name of the MySQL table to join TO.

TEMPLATE [filename]

Template to use with one-to-many joins. The template is processed once for each record and can only contain substitutions for items in the joined table.

TO [item]

Join item in the table to be joined.

TYPE [ONE-TO-ONE|ONE-TO-MANY]

The type of join. Default is one-to-one.

7. Label

This object is used to define a label, which is in turn usually used to annotate a feature with a piece of text. Labels can however also be used as symbols through the use of various TrueType fonts.

ANGLE [double|auto|follow]

Angle, given in degrees, to draw the label or AUTO to allow the software to compute the angle, AUTO is valid for LINE layers only. FOLLOW was introduced in version 4.10 and tells map server to compute a curved label for appropriate linear features (see RFC 11 for specifics).

ANTIALIAS [true|false]

Should text be antialiased? Note that this requires more available colors, decreased drawing performance, and results in slightly larger output images.

BACKGROUNDCOLOR [r] [g] [b]

Color to draw a background rectangle (i.e. billboard). Off by default.

BACKGROUNDSHADOWCOLOR [r] [g] [b]

Color to draw a background rectangle (i.e. billboard) shadow. Off by default.

BACKGROUNDSHADOWSIZE [x][y]

How far should the background rectangle be offset? Default is 1.

BUFFER [integer]

Padding, in pixels, around labels. Useful for maintaining spacing around text to enhance readability. Available only for cached labels. Default is 0.

COLOR [r] [g] [b]

Color to draw text with.

ENCODING [string]

Supported encoding format to be used for labels. If the format is not supported, the label will not be drawn. Requires the iconv library (present on most systems). The library is always detected if present on the system, but if not the label will not be drawn.

Required for displaying international characters in MapServer. More information can be found at:

<http://www.foss4g.org/FOSS4G/MAPSERVER/mpsnf-i18n-en.html>.

FONT [name]

Font alias (as defined in the FONTSET) to use for labeling.

FORCE [true|false]

Forces labels for a particular class on, regardless of collisions. Available only for cached labels. Default is false.

MAXSIZE [integer]

Maximum font size to use when scaling text (pixels). Default is 256.

MINDISTANCE [integer]

Minimum distance between duplicate labels. Given in pixels.

MINFEATURESIZE [integer|auto]

Minimum size a feature must be to be labeled. Given in pixels. For line data the overall length of the displayed line is used, for polygons features the smallest dimension of the bounding box is used.

"Auto" keyword tells MapServer to only label features that are larger than their corresponding label.
Available for cached labels only.

MINSIZE [integer]

Minimum font size to use when scaling text (pixels). Default is 4.

OFFSET [x][y]

Offset values for labels, relative to the lower left hand corner of the label and the label point. Given in pixels. In the case of rotated text specify the values as if all labels are horizontal and any rotation will be compensated for.

OUTLINECOLOR [r] [g] [b]

Color to draw a one pixel outline around the text.

PARTIALS [true|false]

Can text run off the edge of the map? Default is true.

POSITION [ull|uclrlc|lcclcr|lll|lclrl|auto]

Position of the label relative to the labeling point (layers only). First letter is "Y" position, second letter is "X" position. "Auto" tells MapServer to calculate a label position that will not interfere with other labels. With points and polygons, MapServer selects from the 8 outer positions (i.e. excluding cc). With lines, it only uses lc or uc, until it finds a position that doesn't collide with labels that have already been drawn. If all positions cause a conflict, then the label is not drawn (Unless the label's **FORCE** a parameter is set to "true"). "Auto" placement is only available with cached labels.

SHADOWCOLOR [r] [g] [b]

Color of drop shadow.

SHADOWSIZE [x][y]

Shadow offset in pixels.

SIZE [integer]||[tiny|small|medium|large|giant]

Text size. Use "integer" to give the size in pixels of your TrueType font based label, or any of the other 5 listed keywords to bitmap fonts.

TYPE [bitmap|true-type]

Type of font to use. Generally bitmap fonts are faster to draw than TrueType fonts. However, TrueType fonts are scalable and available in a variety of faces. Be sure to set the **FONT** parameter if you select TrueType.

WRAP [character]

Character that represents an end-of-line condition in label text, thus resulting in a multi-line label.

8. Layer

The most used object in a MapFile, this one describes layers used to make up a map. Layers are drawn in their order of appearance in the MapFile (first layer is at the bottom, last is on top).

CLASS

Signals the start of a **CLASS** object.

CLASSITEM [attribute]

Item name in attribute table to use for class lookups.

CONNECTION [string]

Database connection string to retrieve remote data.

An SDE connection string consists of a hostname, instance name, database name, username and password separated by commas.

A PostGIS connection string is basically a regular PostgreSQL connection string, it takes the form of
"user=nobody password=***** dbname=dbname host=localhost port=5432"

An Oracle connection string: user/pass[@db]

CONNECTIONTYPE [local|sdelogr|postgis|oracle|spatial|wms]

Type of connection. Default is local. See additional documentation for any other type.

DATA [filename][[sde parameters][postgis table/column][oracle table/column]]

Full filename of the spatial data to process. No file extension is necessary for shapefiles. Can be specified relative to the SHAPEPATH option from the Map Object.

If this is an SDE layer, the parameter should include the name of the layer as well as the geometry column, i.e. "mylayer,shape,myversion".

If this is a PostGIS layer, the parameter should be in the form of "<columnname> from <tablename>", where "columnname" is the name of the column containing the geometry objects and "tablename" is the name of the table from which the geometry data will be read.

For Oracle, use "shape FROM table" or "shape FROM (SELECT statement)" or even more complex Oracle compliant queries! Note that there are important performance impacts when using spatial subqueries however. Try using MapServer's FILTER whenever possible instead. You can also see the SQL submitted by forcing an error, for instance by submitting a DATA parameter you know won't work, using for example a bad column name.

DEBUG [on|off]

Enables debugging of the layer object. Verbose output is generated and sent to the standard error output (STDERR) or the MapServer logfile if one is set using the LOG parameter in the WEB object.

DUMP [true|false]

Switch to allow mapserver to return data in GML format. Usefull when used with WMS GetFeatureInfo operations. "false" by default.

FEATURE

Signals the start of a FEATURE object.

FILTER [string]

This parameter allows for data specific attribute filtering that is done at the same time spatial filtering is done, but before any CLASS expressions are evaluated. For OGR and shapefiles the string is simply a mapserver regular expression. For spatial databases the string is a SQL WHERE clause that is valid with respect to the underlying database.

For example: FILTER "type='road' and size <2"

FILTERITEM [attribute]

Item to use with simple FILTER expressions. OGR and shapefiles only.

FOOTER [filename]

Template to use *after* a layer's set of results have been sent. Multiresult query modes only.

GRID

Signals the start of a GRID object.

GROUP [name]

Name of a group that this layer belongs to. The group name can then be reference as a regular layer name in the template files, allowing to do things like turning on and off a group of layers at once.

HEADER [filename]

Template to use *before* a layer's set of results have been sent. Multiresult query modes only.

LABELANGLEITEM [attribute]

Item name in attribute table to use for class annotation angles. Values should be in degrees.

LABELCACHE [on|off]

Specifies whether labels should be drawn as the features for this layer are drawn, or whether they should be cached and drawn after all layers have been drawn. Default is on. Label overlap removal, auto placement etc... are only available when the label cache is active.

LABELITEM [attribute]

Item name in attribute table to use for class annotation (i.e. labeling).

LABELMAXSCALE [double]

Maximum scale at which the layer is labeled.

LABELMINSCALE [double]

Minimum scale at which the layer is labeled.

LABELREQUIRES [expression]

Sets context for labeling this layer, for example:

```
LABELREQUIRES " ! [orthoquads] "
```

means that this layer would NOT be labeled if a layer named "orthoquads" is on. The expression consists of a boolean expression based on the status of other layers, each [layer name] substring is replaced by a 0 or a 1 depending on that layer's STATUS and then evaluated as normal. Logical operators AND and OR can be used.

LABELSIZEITEM [attribute]

Item name in attribute table to use for class annotation sizes. Values should be in pixels.

MAXFEATURES [integer]

Specifies the number of features that should be drawn for this layer in the CURRENT window. Has some interesting uses with annotation and with sorted data (i.e. lakes by area).

MAXSCALE [double]

Maximum scale at which this layer is drawn.

METADATA

```
METADATA
    title "My layer title"
    author "Me!"
END
```

MINSCALE [double]

Minimum scale at which this layer is drawn.

NAME [string]

Short name for this layer. Limit is 20 characters. This name is the link between the mapfile and web interfaces that refer to this name. They must be identical. The name should be unique, unless one layer replaces another at different scales. Use the GROUP option to associate layers with each other.

OFFSITE [r] [g] [b]

Sets the color index to treat as transparent for raster layers.

POSTLABELCACHE [true|false]

Tells MapServer to render this layer after all labels in the cache have been drawn. Useful for adding neatlines and similar elements. Default is false.

PROCESSING [string]

Passes a processing directive to be used with this layer. The supported processing directives vary by layer type, and the underlying driver that processes them. Currently the SCALE,BANDS and

DITHER directives are supported for raster layers processed via the GDAL driver. This is further explained in the Raster HOWTO.

```
PROCESSING "SCALE_1=AUTO"
PROCESSING "SCALE_2=AUTO"
PROCESSING "SCALE_3=AUTO"
PROCESSING "BANDS=3,2,1,4"
```

This is also where you can enable connection pooling for certain layer types. Connection pooling will allow MapServer to share the handle to an open database or layer connection throughout a single map draw process. Additionally, if you have FastCGI enabled, the connection handle will stay open indefinitely, or according to the options specified in the FastCGI configuration. Oracle, ArcSDE, OGR and PostGIS currently support this approach.

```
PROCESSING "CLOSE_CONNECTION=DEFER"
```

PROJECTION

Signals the start of a PROJECTION object.

REQUIRES [expression]

Sets context for displaying this layer (see LABELREQUIRES).

SIZEUNITS [pixels|feet|inches|kilometers|meters|miles]

Sets the unit of CLASS object SIZE values (default is pixels). Useful for simulating buffering.

STATUS [on|off|default]

Sets the current status of the layer. Often modified by MapServer itself. Default turns the layer on permanently.

Some notes regarding the STATUS values:

- ◊ In CGI mode, layers with STATUS DEFAULT cannot be turned off using normal mechanisms. It is recommended to set layers to STATUS DEFAULT while debugging a problem, but set them back to ON/OFF in normal use.
- ◊ For WMS, layers in the server mapfile with STATUS DEFAULT are always sent to the client.

STYLEITEM [attribute]

Item to use for feature specific styling. This is *very* experimental and OGR only at the moment.

SYMBOLSCALE [double]

The scale at which symbols and/or text appear full size. This allows for dynamic scaling of objects based on the scale of the map. If not set then this layer will always appear at the same size. Scaling only takes place within the limits of MINSIZE and MAXSIZE as described above.

TEMPLATE [file|url]

Used as a global alternative to CLASS TEMPLATE.

TILEINDEX [filename|layername]

Name of the tileindex file or layer. A tileindex is similar to an ArcInfo library index. The tileindex contains polygon features for each tile. The item that contains the location of the tiled data is given using the TILEITEM parameter. When a file is used as the tileindex for shapefile or raster layers, the tileindex should be a shapefile. For CONNECTIONTYPE OGR layers, any OGR supported datasource can be a tileindex. Normally the location should contain the path to the tile file relative to the shapepath, not relative to the tileindex itself. If the DATA parameter contains a value then it is added to the end of the location. When a tileindex layer is used, it works similarly to directly referring to a file, but any supported feature source can be used (ie. postgres, oracle).

NOTE: All files in the tileindex should have the same coordinate system, and for vector files the same set of attributes in the same order.

TILEITEM [attribute]

Item that contains the location of an individual tile, default is "location".

TOLERANCE [double]

Sensitivity for point based queries (i.e. via mouse and/or map coordinates). Given in **TOLERANCEUNITS**. If the layer is a POINT or a LINE, the default is 3. For all other layer types, the default is 0. To restrict polygon searches so that the point must occur in the polygon set the tolerance to zero.

TOLERANCEUNITS [pixels|feet|inches|kilometers|meters|miles|dd]

Units of the **TOLERANCE** value. Default is pixels.

TRANSPARENCY [integer|alpha]

Sets the transparency level of all classed pixels for a given layer. The value can either be an integer in the range (0-100) or the named symbol "ALPHA". Although this parameter is named "transparency", the integer values actually parameterize layer opacity. A value of 100 is opaque and 0 is fully transparent.

The "ALPHA" symbol directs the mapserver rendering code to honor the indexed or alpha transparency of pixmap symbols used to style a layer. This is only needed in the case of RGB output formats, and should be used only when necessary as it is expensive to render transparent pixmap symbols onto an RGB map image.

TRANSFORM [true|false|ulluclrlcclrlrlrlcclrlr]

Tells MapServer whether or not a particular layer needs to be transformed from some coordinate system to image coordinates. Default is true. This allows you to create shapefiles in image/graphics coordinates and therefore have features that will always be displayed in the same location on every map. Ideal for placing logos or text in maps. Remember that the graphics coordinate system has an origin in the upper left hand corner of the image, contrary to most map coordinate systems.

Version 4.10 introduces the ability to define features with coordinates given in pixels (or percentages, see **UNITS**), most often inline features, relative to something other than the UL corner of an image. That is what 'TRANSFORM FALSE' means. By setting an alternative origin it allows you to anchor something like a copyright statement to another portion of the image in a way that is independent of image size.

TYPE [point|line|polygon|circle|annotation|raster|query]

Specifies how the data should be drawn. Need not be the same as the shapefile type. For example, a polygon shapefile may be drawn as a point layer, but a point shapefile may not be drawn as a polygon layer. Common sense rules. Annotation means that a label point will be calculated for the features, but the feature itself will not be drawn although a marker symbol can be optionally drawn. this allows for advanced labeling like numbered highway shields. Points are labeled at that point. Polygons are labeled first using a centroid, and if that doesn't fall in the polygon a scanline approach is used to guarantee the label falls within the feature. Lines are labeled at the middle of the longest arc in the visible portion of the line. Query only means the layer can be queried but not drawn.

In order to differentiate between POLYGONS and POLYLINES (which do not exist as a type), simply respectively use or omit the COLOR keyword when classifying. If you use it, it's a polygon with a fill color, otherwise it's a polyline with only an OUTLINECOLOR.

A circle must be defined by a minimum bounding rectangle. That is, 2 points that define the smallest square that can contain it. These 2 points are the two opposite corners of said box.

The following is an example using inline points to draw a circle:

```

LAYER
  NAME 'inline_circles'
  TYPE CIRCLE
  STATUS ON
  FEATURE
    POINTS
      74.01 -53.8
      110.7 -22.16
    END
  END
  CLASS
    STYLE
      COLOR 0 0 255
    END
  END
END

```

9. Legend

Defines how a legend is to be built. Legend components are built automatically from class objects from individual layers. Starts with the keyword LEGEND and terminates with the keyword END.
The size of the legend image is NOT known prior to creation so be careful not to hard-code width and height in the tag in the template file.

IMAGECOLOR [r] [g] [b]

Color to initialize the legend with (i.e. the background).

INTERLACE [on|off]

Deprecated Default is [on]. This keyword is now deprecated in favour of using the FORMATOPTION "INTERLACE=ON" line in the [OUTPUTFORMAT](#) declaration.

LABEL

Signals the start of a [LABEL](#) object

OUTLINECOLOR [r] [g] [b]

Color to use for outlining symbol key boxes.

POSITION [ull|uclur|lll|lc|lr]

Where to place an embedded legend in the map. Default is lr.

KEYSIZE [x][y]

Size of symbol key boxes in pixels. Default is 20 by 10.

KEYSPACING [x][y]

Spacing between symbol key boxes ([y]) and labels ([x]) in pixels. Default is 5 by 5.

POSTLABELCACHE [true|false]

Tells MapServer to render this legend after all labels in the cache have been drawn. Useful for adding neatlines and similar elements. Default is false.

STATUS [on|off|embeded]

Is the legend image to be created.

TEMPLATE [filename]

HTML legend template file. Refer to the [HTML Legend howto](#).

TRANSPARENT [on|off]

Deprecated Should the background color for the legend be transparent. This flag is now deprecated in favour of declaring transparency within [OUTPUTFORMAT](#) declarations. Default is off.

10. Map

A MAP object defines the master object of the mapfile. It defines application/map wide parameters.

ANGLE [double]

Angle, given in degrees, to rotate the map. Default is 0. The rendered map will rotate in a clockwise direction. The following are important notes:

- ◊ Requires a PROJECTION object specified at the MAP level and for each LAYER object (even if all layers are in the same projection).
- ◊ Requires MapScript (SWIG, PHPMapscript). Does not work with CGI mode.
- ◊ If using the LABEL object's ANGLE or the LAYER object's LABELANGLEITEM parameters as well, these parameters are relative to the map's orientation (i.e. they are computed after the MAP object's ANGLE). For example, if you have specified an ANGLE for the map of 45, and then have a layer LABELANGLEITEM value of 45, the resulting label will not appear rotated (because the resulting map is rotated clockwise 45 degrees and the label is rotated counter-clockwise 45 degrees).
- ◊ More information can be found on the [MapRotation Wiki Page](#).

CONFIG [key] [value]

This can be used to define the location of your EPSG files for the PROJ.4 library. Setting the [key] to PROJ_LIB and the [value] to the location of your EPSG files will force PROJ.4 to use this value. Using CONFIG allows you to avoid setting environment variables to point to your PROJ_LIB directory. Here are some examples:

1. Unix

```
CONFIG "PROJ_LIB" "/usr/local/share/proj/"
```

2. Windows

```
CONFIG "PROJ_LIB" "C:/somedir/proj/nad/"
```

Any other value will be passed on to CPLSetConfigOption(). This provides customized control of some GDAL and OGR driver behaviours. Details on such options would be found in specific GDAL driver documentation.

DATAPATTERN [regular expression]

This defines a regular expression to be applied to requests to change DATA parameters via URL requests (i.e. map_layername_data=...). If a pattern doesn't exist then web users can't monkey with support files via URLs. This allows you to isolate one application from another if you desire, with the default operation being very conservative. See also [TEMPLATEPATTERN](#).

DEBUG [on/off]

Enables debugging of the map object. Verbose output is generated and sent to the standard error output (STDERR) or the MapServer logfile if one is set using the LOG parameter in the WEB object.

EXTENT [minx] [miny] [maxx] [maxy]

The spatial extent of the map to be created. In most cases you will need to specify this, although mapserver can sometimes (expensively) calculate one if it is not specified.

FONTSET [filename]

Full (absolute) filename of fontset file to use.

IMAGECOLOR [r] [g] [b]

Color to initialize the map with (i.e. background color). When transparency is enabled (TRANSPARENT ON) for the typical case of 8-bit pseudocolored map generation, this color will be marked as transparent in the output file palette. Any other map components drawn in this color will

also be transparent, so for map generation with transparency it is best to use an otherwise unused color as the background color.

IMAGEQUALITY [int]

Deprecated Use FORMATOPTION "QUALITY=n" in the OUTPUTFORMAT declaration to specify compression quality for JPEG output.

IMAGETYPE [gif|png|jpeg|wbmp|gtiff|swf|userdefined]

Output format to generate. See details in the [OUTPUTFORMAT](#) section for available formats. The name here must match the 'NAME' of a user defined or internally generated OUTPUTFORMAT section.

INTERLACE [on|off]

Deprecated Use FORMATOPTION "INTERLACE=ON" in the [OUTPUTFORMAT](#) declaration to specify if the output images should be interlaced.

LAYER

Signals the start of a [LAYER](#) object.

LEGEND

Signals the start of a [LEGEND](#) object.

MAXSIZE [integer]

Sets the maximum size of the map image. This will override the default value. For example, setting this to 2048 means that you can have up to 2048 pixels in both dimensions (i.e. max of 2048x2048).

NAME [name]

Prefix attached to map, scalebar and legend GIF filenames created using this MapFile. It should be kept short.

PROJECTION

Signals the start of a [PROJECTION](#) object.

QUERYMAP

Signals the start of a [QUERYMAP](#) object.

REFERENCE

Signals the start of a [REFERENCE MAP](#) object.

RESOLUTION [int]

Sets the pixels per inch for output, only affects scale computations and nothing else, default is 72.

SCALE [double]

Computed scale of the map. Set most often by the application.

SCALEBAR

Signals the start of a [SCALEBAR](#) object.

SHAPEPATH [filename]

Path to the directory holding the shapefiles or tiles. There can be further subdirectories under SHAPEPATH.

SIZE [x][y]

Size in pixels of the output image (i.e. the map).

STATUS [on|off]

Is the map active? Sometimes you may wish to turn this off to use only the reference map or scale bar.

SYMBOLSET [filename]

Full (absolute) filename of the symbolset to use.

SYMBOL

Signals the start of a [SYMBOL](#) object.

TEMPLATEPATTERN [regular expression]

This defines a regular expression to be applied to requests to change TEMPLATE parameters via URL requests (i.e. map_layername_template=...). If a pattern doesn't exist then web users can't monkey with support files via URLs. This allows you to isolate one application from another if you desire, with the default operation being very conservative. See also [DATAPATTERN](#).

TRANSPARENT [on|off]

Deprecated Use FORMATOPTION "TRANSPARENT=ON" in the OUTPUTFORMAT declaration to specify if the output images should be transparent.

UNITS [feet|inches|kilometers|meters|miles|dd]

Units of the map coordinates. Used for scalebar and scale computations.

WEB

Signals the start of a [WEB](#) object.

11. OutputFormat

This section discusses how output formats are defined and selected.

A map file may have zero, one or more OUTPUTFORMAT object declarations, defining available output formats supported including formats like PNG, GIF, JPEG, GeoTIFF and Flash (SWF).

If OUTPUTFORMAT sections declarations are not found in the map file, the following implicit declarations will be made. Only those for which support is compiled in will actually be available. The GeoTIFF depends on building with GDAL support, and the Flash (SWF) depends on compiling with support for the MING library.

```
OUTPUTFORMAT
  NAME gif
  DRIVER "GD/GIF"
  MIMETYPE "image/gif"
  IMAGEMODE PC256
  EXTENSION "gif"
END
OUTPUTFORMAT
  NAME png
  DRIVER "GD/PNG"
  MIMETYPE "image/png"
  IMAGEMODE PC256
  EXTENSION "png"
END
OUTPUTFORMAT
  NAME jpeg
  DRIVER "GD/JPEG"
  MIMETYPE "image/jpeg"
  IMAGEMODE RGB
  EXTENSION "jpg"
END
OUTPUTFORMAT
  NAME wbmp
  DRIVER "GD/WBMP"
  MIMETYPE "image/wbmp"
  IMAGEMODE PC256
  EXTENSION "wbmp"
END
OUTPUTFORMAT
  NAME swf
  DRIVER "SWF"
  MIMETYPE "application/x-shockwave-flash"
  EXTENSION "swf"
  IMAGEMODE PC256
```

```

FORMATOPTION "OUTPUT_MOVIE=SINGLE"
END
OUTPUTFORMAT
  NAME GTiff
  DRIVER "GDAL/GTiff"
  MIMETYPE "image/tiff"
  IMAGEMODE RGB
  EXTENSION "tif"
END

```

NAME [name]

The name to use in the IMAGETYPE keyword of the map file to select this output format.(optional)

DRIVER [name]

The name of the driver to use to generate this output format. Some driver names include the definition of the format if the driver supports multiple formats. For GD the possible driver names are "GD/Gif", "GD/PNG", "GD/WBMP" and "GD/JPEG". For flash the driver is just called "SWF". For output through GDAL the GDAL shortname for the format is appended, such as "GDAL/GTiff". Note that PNG, JPEG and GIF output can be generated with either GDAL or GD (GD is generally more efficient).(mandatory)

IMAGEMODE [PC256/RGB/RGBA/INT16/FLOAT32]

Selects the imaging mode in which the output is generated. Does matter for non-raster formats like Flash. Not all formats support all combinations. For instance GD/GIF supports only PC256. (optional)

- ◊ PC256: Produced a pseudocolored result with up to 256 colors in the palette (traditional MapServer mode)
- ◊ RGB: Render in 24bit Red/Green/Blue mode. Supports all colors but does not support transparency.
- ◊ RGBA: Render in 32bit Red/Green/Blue/Alpha mode. Supports all colors, and alpha based transparency.
- ◊ BYTE: Render raw 8bit pixel values (no presentation). Only works for RASTER layers (through GDAL) and WMS layers currently.
- ◊ INT16: Render raw 16bit signed pixel values (no presentation). Only works for RASTER layers (through GDAL) and WMS layers currently.
- ◊ FLOAT32: Render raw 32bit floating point pixel values (no presentation). Only works for RASTER layers (through GDAL) and WMS layers currently.

MIMETYPE [type]

Provide the mime type to be used when returning results over the web. (optional)

EXTENSION [type]

Provide the extension to use when creating files of this type. (optional)

TRANSPARENT [ON/OFF]

Indicates whether transparency should be enabled for this format. Note that transparency does not work for IMAGEMODE RGB output. Not all formats support transparency (optional). When transparency is enabled for the typical case of 8-bit pseudocolored map generation, the IMAGECOLOR color will be marked as transparent in the output file palette. Any other map components drawn in this color will also be transparent, so for map generation with transparency it is best to use an otherwise unused color as the background color.

FORMATOPTION [option]

Provides a driver or format specific option. Zero or more FORMATOPTION statement may be present within a OUTPUTFORMAT declaration. (optional)

- ◊ GD/JPEG: The "QUALITY=n" option may be used to set the quality of jpeg produced (value from 0-100).
- ◊ GD/PNG: The "INTERLACE=[ON/OFF]" option may be used to turn interlacing on or off.

- ◊ GD/GIF: The "INTERLACE=[ON/OFF]" option may be used to turn interlacing on or off.
- ◊ GDAL/GTiff: Supports the TILED=YES, BLOCKXSIZE=n, BLOCKYSIZE=n, INTERLEAVE=[PIXEL/BAND] and COMPRESS=[NONE,PACKBITS,JPEG,LZW,DEFLATE] format specific options.
- ◊ GDAL/*: All FORMATOPTIONS are passed onto the GDAL create function. Options supported by GDAL are described in the detailed documentation for each GDAL format

12. Projection

This object defines the coordinate system to display your data.

To set up projections you must define two projection objects: one for the output image (in the MAP object) and one for each layer (in the LAYER objects) to be projected. MapServer relies on the [PROJ.4](#) library for projections. Projection objects therefore consist of a series of PROJ.4 keywords, which are either specified within the object directly or referred to in an *epsg* file. An *epsg* file is a lookup file containing projection parameters, and is part of the PROJ.4 library.

The following two examples both define the same projection (UTM zone 15, NAD83), but use 2 different methods:

Example1: Inline Projection Parameters

```
PROJECTION
  "proj=utm"
  "ellps=GRS80"
  "datum=NAD83"
  "zone=15"
  "units=m"
  "north"
  "no_defs"
END
```

Example2: epsg Projection Use

```
PROJECTION
  "init=epsg:26915"
END
```

(this refers to an *epsg* lookup file that contains a '26915' code with the full projection parameters)

The next two examples both display how to possibly define unprojected lat/longs ("geographic"):

Example3: Inline Projection Parameters

```
PROJECTION
  "proj=latlong"
  "ellps=WGS84"
  "datum=WGS84"
END
```

Example4: epsg Projection Use

```
PROJECTION
  "init=epsg:4326"
END
```

Important Notes

- If all of your data in the mapfile is in the same projection, you DO NOT have to specify any projection objects. MapServer will assume that all of the data is in the same projection.
- If you specify a MAP-level projection, and then only one other LAYER projection object, MapServer will assume that all of the other layers are in the specified MAP-level projection.
- Always refer to the epsg file in lowercase, because it is a lowercase filename and on Linux/Unix systems this parameter is case sensitive.

For More Information

- If you get projection errors, refer to the [MapServer Error page](#) to check if your exact error has been discussed.
 - ◆ Search the MapServer-users [email list archives](#), odds are that someone has faced your exact issue before.
- See the [PROJ.4](#) user guides for complete descriptions of supported projections and coordinate systems.
- Refer to the [Cartographical Map Projections](#) page for background information on projections.

13. QueryMap

Defines a mechanism to map the results of a query. Starts with the keyword QUERYMAP and terminates with the keyword END.

COLOR [r] [g] [b]

Color in which features are highlighted. Default is yellow.

SIZE [x][y]

Size of the map in pixels. Defaults to the size defined in the map object.

STATUS [on|off]

Is the query map to be drawn?

STYLE [normal|hiliteselected]

Sets how selected features are to be handled. Layers not queried are drawn as usual.

◊ Normal: Draws all features according to the settings for that layer.

◊ Hilite: Draws selected features using COLOR. Non-selected features are drawn normally.

◊ Selected: draws only the selected features normally.

14. Reference Map

Defines how reference maps are to be created. Starts with the keyword REFERENCE and terminates with the keyword END.

Three types of reference maps are supported. The most common would be one showing the extent of a map in an interactive interface. It is also possible to request reference maps as part of a query. Point queries will generate an image with a marker (see below) placed at the query point. Region based queries will depict the extent of the area of interest. Finally, feature based queries will display the selection feature(s) used.

COLOR [r] [g] [b]

Color in which the reference box is drawn. Set any component to -1 for no fill. Default is red.

EXTENT [minx][miny][maxx][maxy]

The spatial extent of the base reference image.

IMAGE [filename]

Full filename of the base reference image. Must be a GIF image.

MARKER [integer]string

Defines a symbol (from the symbol file) to use when the box becomes too small (see MINBOXSIZE and MAXBOXSIZE below). Uses a crosshair by default.

MARKERSIZE [integer]

Defines the size of the symbol to use instead of a box (see MARKER above).

MINBOXSIZE [integer]

If box is smaller than MINBOXSIZE (use box width or height) then use the symbol defined by MARKER and MARKERSIZE.

MAXBOXSIZE [integer]

If box is greater than MAXBOXSIZE (use box width or height) then draw nothing (Often the whole map gets covered when zoomed way out and it's perfectly obvious where you are).

OUTLINECOLOR [r] [g] [b]

Color to use for outlining the reference box. Set any component to -1 for no outline.

SIZE [x][y]

Size, in pixels, of the base reference image.

STATUS [on|off]

Is the reference map to be created? Default it off.

15. Scalebar

Defines how a scalebar should be built. Starts with the keyword SCALEBAR and terminates with the keyword END.

Scalebars currently do not make use of TrueType fonts. The size of the scalebar image is NOT known prior to rendering, so be careful not to hard-code width and height in the tag in the template file. Future versions will make the image size available.

BACKGROUNDCOLOR [r] [g] [b]

Color to use for scalebar background, not the image background.

COLOR [r] [g] [b]

Color to use for drawing all features if attribute tables are not used.

IMAGECOLOR [r] [g] [b]

Color to initialize the scalebar with (i.e. background).

INTERLACE [true|false]

Should output images be interlaced? Default is [on]. This keyword is now deprecated in favour of using the FORMATOPTION "INTERLACE=ON" line in the OUTPUTFORMAT declaration.

INTERVALS [integer]

Number of intervals to break the scalebar into. Default is 4.

LABEL

Signals the start of a LABEL object

OUTLINECOLOR [r] [g] [b]

Color to use for outlining individual intervals. Set any component to -1 for no outline which is the default.

POSITION [ulluclurllllcrlr]

Where to place an embedded scalebar in the image. Default is lr.

POSTLABELCACHE [true|false]

For use with embedded scalebars only. Tells the MapServer to embed the scalebar after all labels in the cache have been drawn. Default is false.

SIZE [x][y]

Size in pixels of the scalebar. Labeling is not taken into account.

STATUS [on|off|embed]

Is the scalebar image to be created, and if so should it be embedded into the image? Default is off.

(Please note that embedding scalebars require that you define a marker set. In essence the scalebar becomes a custom marker that is handled just like any other annotation.)

STYLE [integer]

Chooses the scalebar style. Valid styles are 0 and 1.

TRANSPARENT [on|off]

Should the background color for the scalebar be transparent. This flag is now deprecated in favour of declaring transparency within OUTPUTFORMAT declarations. Default is off.

UNITS [feet|inches|kilometers|meters|miles]

Output scalebar units, default is miles. Used in conjunction with the map's units to develop the actual graphic. Note that decimal degrees are not valid scalebar units.

16. Style

This object holds parameters for symbolization. Multiple styles may be applied within a class.

This object is new in 4.0 and is intended to separate logic from looks. The final intent is to have named styles (Not yet supported) that will be re-usable through the mapfile. This is the new, preferred way of defining the appearance of an object, notably a class.

ANGLE [double]

Angle, given in degrees, to draw the line work. Default is 0. For symbols of Type HATCH, this is the angle of the hatched lines. For its use with hatched lines, see Example#8 in the [SYMBOL examples](#).

ANGLEITEM [string]

Attribute/field that stores the angle to be used in rendering. Angle is given in degrees with 0 meaning no rotation.

ANTIALIAS [true|false]

Should TrueType fonts and Cartoline symbols be antialiased.

BACKGROUNDCOLOR [r] [g] [b]

Color to use for non-transparent symbols.

COLOR [r] [g] [b]

Color to use for drawing features.

MAXSIZE [integer]

Maximum size in pixels to draw a symbol. Default is 50.

MINSIZE [integer]

Minimum size in pixels to draw a symbol. Default is 0.

MINWIDTH [integer]

Minimum width in pixels to draw the line work.

OFFSET [x][y]

Offset values for shadows, hollow symbols, etc ...

OUTLINECOLOR [r] [g] [b]

Color to use for outlining polygons and certain marker symbols. Line symbols do not support outline colors.

SIZE [integer]

Height, in pixels, of the symbol/pattern to be used. Only useful with scalable symbols. Default is 1.

For symbols of Type HATCH, the SIZE is the distance between hatched lines. For its use with hatched lines, see Example#8 in the [SYMBOL examples](#).

SIZEITEM [string]

Attribute/field that stores the size to be used in rendering. Value is given in pixels.

SYMBOL [integer]string|filename]

The symbol name or number to use for all features if attribute tables are not used. The number is the index of the symbol in the symbol file, starting at 1, the 5th symbol in the file is therefore symbol number 5. You can also give your symbols names using the NAME keyword in the symbol definition file, and use those to refer to them. Default is 0, which results in a single pixel, single width line, or solid polygon fill, depending on layer type.

You can also specify a gif or png filename. The path is relative to the location of the mapfile.

WIDTH [integer]

Width refers to the thickness of line work drawn, in pixels. Default is 1. For symbols of Type HATCH, the WIDTH is how thick the hatched lines are. For its use with hatched lines, see Example#8 in the [SYMBOL examples](#).

17. Variable Substitution

New in MapServer 4.0, variables can be substituted within mapfile parameter values. At this time, cookie and CGI parameter values are supported. This allows mapserver mapfiles to be aware of a user's cookies (Good for implementing security), or non-mapserver request parameters (Good for integrating with other systems). Syntax: '%' + variable name + '%'

Example 1. Connecting securely to a Spatial Database

You want to map some sensitive data held in a PostGIS database. The username and password to be used for the database connection are held in 2 cookies previously set by a separate authentication mechanism, "uid" and "passwd".

```
CONNECTION "user=%uid% password=%passwd% dbname=postgis"
```

Example 2. Handling temporary files

You have a user based discovery application that generates shapefiles and stores them in a user's home directory on the server. The "username" comes from a cookie, the "filename" comes from a request parameter.

```
DATA "/home/%username%/tempshp/%filename%"
```

This feature is only available in the CGI version of MapServer through a mapfile pre-processor. If you are using MapScript, you will have to code the substitution logic into your application yourself (By writing your own pre-processor).

18. Web

Defines how a web interface will operate. Starts with the keyword WEB and terminates with the keyword END.

EMPTY [url]

URL to forward users to if a query fails. If not defined the value for ERROR is used.

ERROR [url]

URL to forward users to if an error occurs. Ugly old MapServer error messages will appear if this is not defined

FOOTER [filename]

Template to use AFTER anything else is sent. Multiresult query modes only.

HEADER [filename]

Template to use BEFORE everything else has been sent. Multiresult query modes only.

IMAGEPATH [path]

Path to the temporary directory for writing temporary files and images. Must be writable by the user the web server is running as. Must end with a / or depending on your platform.

IMAGEURL [path]

Base URL for IMAGEPATH. This is the URL that will take the web browser to IMAGEPATH to get the images.

LOG [filename]

File to log MapServer activity in. Must be writable by the user the web server is running as.

MAXSCALE [double]

Maximum scale at which this interface is valid. When a user requests a map at a bigger scale, MapServer automatically returns the map at this scale. This effectively prevents user from zooming too far out.

MAXTEMPLATE [fileurl]

Template to be used if above the maximum scale for the app, useful for nesting apps.

METADATA

```
METADATA
    title "My layer title"
    author "Me!"
END
```

MINSCALE [double]

Minimum scale at which this interface is valid. When a user requests a map at a smaller scale, MapServer automatically returns the map at this scale. This effectively prevents the user from zooming in too far.

MINTEMPLATE

Template to be used if above the minimum scale for the app, useful for nesting apps.

OUTPUTFORMAT [mime-type]

Format of the query output. Default is "text/html". This is experimental, the use of the OUTPUTFORMAT object is recommended instead.

TEMPLATE [filenameurl]

Template file or URL to use in presenting the results to the user in an interactive mode (i.e. map generates map and so on ...)

19. Article Info

Document information (version, author, etc.)

Release: 4.6

Last Updated: 2005/12/15

Author: Jean-FranÃ§ois Doyon

Contact: jdoyon(at)ccrs.nrcan.gc.ca

Author: Jeff McKenna

Contact: jmckenna(at)dmsolutions.ca

Author: Steve Lime

Contact: steve.lime(at)dnr.state.mn.us