

map_server_faq : AEROPGLG-FAQ

GLG Map Server FAQs

FAQs

- [Map Server Setup FAQ](#)
 - › [Map Server Image Gallery](#)
 - › [Online Demos \(integrated with the Toolkit\)](#)
- [Map Server GIS Data FAQ](#)
 - › [Map Server Info](#)
 - › [Documentation](#)
- [Map Server Features FAQ](#)



[Map Server Setup FAQ](#)

[How do I edit setup files?](#)

[Where can I find sample setup files?](#)

[Can I reuse layer setup files \(LIFs\) between different datasets?](#)

[Can I use the Map Server to generate map images from a command line?](#)

[Can I test the Map Server output without setting it up on a web server?](#)

[How do I setup the GLG Map Server on a web server?](#)

[How does a sample http map request look like?](#)

[Are there any tools for interactive testing and prototyping?](#)

[Map Server GIS Data FAQ](#)

[What GIS data is provided with the Map Server?](#)

[What kind of data can I use with the GLG Map Server?](#)

[Where can I find more GIS data?](#)

[How do I use shape files with the GLG Map Server?](#)

[How do I use XYZ raster format with the GLG Map Server?](#)

[How do I use elevation data \(DEM/DTED\)?](#)

[How do I use shaded relief images?](#)

[Where can I find documentation for the GVF vector format?](#)

[How do I tile large datasets?](#)

[How do I setup data sets with large number of data files?](#)

[What if data may change on the fly, such as weather maps?](#)

[Does Generic Logic provide consulting services for handling custom requests?](#)

[**Map Server Features HOWTO**](#)

[Does GLG Map Server comply with the OpenGIS WMS standard?](#)

[Does it support 24-bit color?](#)

[Does it support TrueType fonts and Unicode?](#)

[How do I add a new font?](#)

[How do I define semi-transparent layers?](#)

[How do I obtain lat/lon coordinates of a mouse click?](#)

[How do I query elevation of a point?](#)

[How do I set up layers to show more details when the map is zoomed in?](#)

[How do I define a default set of layers to display?](#)

[How do I control grid spacing and labeling?](#)

[How do I alter display of vector features depending on their attribute values?](#)

[How do I show only some vector features based on some criteria?](#)

[How do I define custom icons for point features?](#)

[How do I disable display of vector features which are too small for the selected zoom level?](#)

[How do I make labels visible on top of a background of the same color?](#)

[How do I prevent layer's labels obscuring each other and labels in other layers?](#)

Map Server Setup FAQ

How do I edit setup files?

The Map Server uses ASCII setup files which use simple "*TOKEN= value*" syntax and may be edited using any text editor. All information for one layer is kept in the layer information file (LIF). The LIF files for all layers are listed in the server data file (SDF). The following lists a content of a simple LIF file:

```
TYPE=IMAGE
```

```
IMAGE TYPE=JPEG
```

```
TRANS TYPE=OPAQUE
```

```
FILENAME=earth.jpg
```

```
# Image extent in lat/lon: whole earth. Image size in pixels is auto-detected.
```

```
MIN LON=-180
```

```
MAX LON=180
```

MIN LAT=-90

MAX LAT=90

Where can I find sample setup files?

The Map Server provides sample datasets with all setup files, allowing to use it out of the box. The server SDF setup files for the sample datasets are located in the *map_data* directory of the Map Server installation, and the layer LIF files are located in *map_data/layers* directory.

Can I reuse layer setup files (LIFs) between different datasets?

Yes. The map server setup files use modular design with all layer information encapsulated in the layer's LIF file. Once defined, the layer's LIF file may be used in multiple datasets. For example, the sample datasets provided with the map server reuse the *earth* and *grid* layers.

The Map Server setup files also support "INCLUDE=filename" directive that allows to share portions of the setup information between different layer LIF files.

Can I use the Map Server to generate map images from a command line?

Yes, the Map Server executable provides a command-line interface, which can be used for testing, as well as generating images without web server setup. Refer to the *Command Line Options* chapter of the [GLG Map Server Reference Manual](#) for more information.

[› Examples of command-line options](#)

Can I test the Map Server output without setting it up on a web server?

Yes, the command line options may be used as described in the previous answer.

[› Examples of command-line options](#)

How do I setup the GLG Map Server on a web server?

The GLG Map Server is setup using either the *CGI* or *FastCGI* methods. The run-time environment is very compact, with no dependencies on external libraries or drivers. The only things that are required in the *cgi-bin* directory of the web server are:

- the map server's executable, *GlmMap*
- the setup script, *GlmScript*
- the map data directory containing GIS datasets

The map data directory may be moved outside the *cgi-bin* directory and pointed to by the setup script. For complete setup instructions, refer to the *Appendix A: Web Server Installation Notes* of the [GLG Map Server Reference Manual](#).

How does a sample http map request look like?

Sample http map requests

The GLG Map Server uses OpenGIS WMS standard for the http map server requests.

The following lists a few sample http requests for the **Unix/Linux** environment:

// World map in orthographic projection with elevation and political boundaries

https://www.myserver.com/cgi-

bin/GlmScript?VERSION=1.3.0&REQUEST=GetMap&SRS=AUTO2:42003,1,10,40&WIDTH=700&HEIGHT=600&BBOX=-6500000,-6500000,6500000,6500000.&BGCOLOR=0x0&STYLES=default&FORMAT=image/jpeg&LAYERS=earth,shadow,political&STRETCH=0

// World map in rectangular projection with elevation and political boundaries

https://www.myserver.com/cgi-

bin/GlmScript?VERSION=1.1.0&REQUEST=GetMap&SRS=EPSG:4326&WIDTH=800&HEIGHT=600&BBOX=-180,-90,180,90&BGCOLOR=0x0&STYLES=default&FORMAT=image/jpeg&LAYERS=earth,shadow,political&STRETCH=1

// USA map in orthographic projection with elevation, state outlines and grid

https://www.myserver.com/cgi-

bin/GlmScript?VERSION=1.3.0&REQUEST=GetMap&SRS=AUTO2:42003,1,-97,38&WIDTH=700&HEIGHT=700&BBOX=-2500000,-2500000,2500000,2500000.&BGCOLOR=0x0&STYLES=default&FORMAT=image/jpeg&LAYERS=earth,shadow,grid,states&STRETCH=0

On Windows, the script name will have .pl extension added, and cgi-bin directory will be named *Scripts* if IIS is used instead of apache:

// World map in orthographic projection with elevation and political boundaries

https://www.myserver.com/Scripts/GlmScript.pl?VERSION=1.3.0&REQUEST=GetMap&SRS=AUTO2:42003,1,10,40&WIDTH=700&HEIGHT=600&BBOX=-6500000,-6500000,6500000,6500000.&BGCOLOR=0x0&STYLES=default&FORMAT=image/jpeg&LAYERS=earth,shadow,political&STRETCH=0

These examples use a sample dataset provided with the Map Server.

The *GlmScript* may be renamed as required; multiple copies of the script (i.e. *MapScript1*, *MapScript2*, etc.) may be used to generate map images using different datasets from one web server.

Command-line examples

The following are the commands to produce the same map images from the command line, without the web server setup. The images will be saved in the *sample.jpg* file. A proper path to the *GlmMap* executable and *sample.sdf* file may be required.

```
# World map in orthographic projection with elevation and political boundaries
GlmMap -generate -dataset sample.sdf -output sample.jpg -oGISreq
VERSION=1.3.0&REQUEST=GetMap&SRS=AUTO2:42003,1,10,40&WIDTH=700&HEIGHT
=600&BBOX=-6500000,-
6500000,6500000,6500000.&BGCOLOR=0x0&STYLES=default&FORMAT=image/jpeg
&LAYERS=earth,shadow,political&STRETCH=0
```

```
# World map in rectangular projection with elevation and political boundaries
GlmMap -generate -dataset sample.sdf -output sample.jpg -oGISreq
VERSION=1.1.0&REQUEST=GetMap&SRS=EPSG:4326&WIDTH=800&HEIGHT=600&BB
OX=-180,-
90,180,90&BGCOLOR=0x0&STYLES=default&FORMAT=image/jpeg&LAYERS=earth,sha
dow,political&STRETCH=1
```

```
# USA map in orthographic projection with elevation, state outlines and grid
GlmMap -generate -dataset sample.sdf -output sample.jpg -oGISreq
VERSION=1.3.0&REQUEST=GetMap&SRS=AUTO2:42003,1,-
97,38&WIDTH=700&HEIGHT=700&BBOX=-2500000,-
2500000,2500000,2500000&BGCOLOR=0x0&STYLES=default&FORMAT=image/jpeg&L
AYERS=earth,shadow,grid,states&STRETCH=0
```

Are there any tools for interactive testing and prototyping?

A **sample HTML file** located in the *map_server* directory, *glg_map_display.html*, provides an example of an HTML and JavaScript-based **GIS viewer**. It allows the user to zoom and pan the map, change projection, select map layers to be displayed and change various map parameters. When the user clicks on the map, the script uses AJAX to query the map server and display the lat/lon coordinates of the mouse click. The HTML viewer requires setting up the GLG Map Server on a web server. To use the viewer, edit the *map_server_url* and *map_script* parameters in the HTML file to point to your web server setup.

The **GLG Graphics Builder** may be used to prototype, test and debug Map Server setup interactively **without setting up a web server**. The Graphics Builder enables you to create a GIS object to display a map and change all parameters of the map request

interactively. The Builder's interactive zooming and panning capabilities allow you to test the map setup and automatic layer switching. The Builder also provides interactive features for querying elevation and lat/lon coordinates of a point at the cursor position. The [Community Edition](#) of the GLG Toolkit includes the Community Edition of the GLG Graphics Builder which may be used to prototype the map server and test the map server setup.

› [Download the Community Edition of the GLG Toolkit](#)

To **prototype**, start the Graphics Builder and follow the following steps:

- Select *File, New Widget* from the main menu.
- Create a GIS Object by selecting a map icon  from the Object Palette on the left (use tooltips to find GIS Object).
- Define the GIS Object's size by selecting two points in the drawing (click on the left mouse button to select the first point, move the mouse, then select the second point).
- Press *OK* to select the default dataset: the GIS Object will appear with a map display.
- To resize the map, move the control points at the corners of the GIS Object with the mouse.
- Press the right mouse button and select *Properties*, then try changing the *GIS Projection* attribute.
- Change the value of the *GIS Layers* attribute from "*default*" to "*default,grid*" to display an additional grid layer.
- Try changing the values of the *GIS Center* and *GIS Extent* attributes.
- Try increasing the value of the *GISVerbosity* attribute. Notice an increase in the debugging information output when you reset the drawing using *File, ResetDrawing* option from the main menu.
- With the GIS Object selected, use the *Set as Parent's Viewport GIS Object* option of the *Arrange* menu to activate the **GIS Zoom mode** for the viewport. In this mode, the zoom and pan controls of the Builder will zoom and pan the map instead of zooming and panning the viewport's drawing. In the RECTANGULAR projection, the *RotateZ* slider in the control panel will also rotate the map. Try using the controls to pan and scroll the map.
- Try scrolling the map with the mouse by Ctrl-clicking in the drawing outside the map and dragging the mouse.

- Go down into the GIS Object using the Hierarchy Down  button to activate the **GIS Editing mode**. In this mode, you can draw objects in the GIS lat/lon coordinates on top of the map. Try creating a marker object by selecting the Marker Object  button and clicking in the drawing to position the marker. *Shift-click* on the marker's control point to show its longitude and latitude stored as X and Y coordinates respectively. The objects created on top of the map are stored in the drawing and may be used in applications that use the GIS Object. An application can also create and position dynamic GLG objects on top of the map at run-time via the GLG API.
 - Save the drawing for future use by selecting *File, Save* from the main menu.
- The Builder's *Run Mode* can also be used for prototyping the map. Refer to the [GLG Map Server Tutorial](#) for more information and details of testing interactive zooming and panning in the Builder's *Run Mode*.

Map Server GIS Data FAQ

What GIS data are provided with the Map Server?

The GLG Map Server comes with a few datasets, so that you can start using the map server right away. The following datasets are provided:

- NASA's Visible Earth image (tiled 8192x4096 image and a smaller fallback image)
- The World Political Boundaries layer of the Digital Chart of the World (VMAP0) dataset
- The US States layer of the US Census Tiger dataset
- US Cities dataset

These datasets may be used as a background or high-level overview, while more detailed datasets of particular areas of interest may be overlayed on top, becoming visible when the map is zoomed in to view these areas.

What kind of data can I use with the GLG Map Server?

The GLG Map Server supports both vector and raster data. TIFF, GeoTIFF and JPEG formats are supported natively. All other raster formats (including NITF, CADRG, DEM, DTED, etc.) can be converted to one of the native formats using [gdal](#). The conversion is done at the setup time, to avoid performance penalties and dependencies on numerous gdal support drivers at the map generation run-time. A gdal-based raster conversion utility is provided for the Linux platform.

For vector data, the Map Server's open vector format ([GVF](#)) is used natively. The [shape files](#) are also supported and may be used either directly or converted to the GVF format using a supplied conversion utility. Other vector formats are supported using filters. Filters for most common GIS vector formats, such as shorelines, CIA World DataBank (WDB), and others, are provided, and the rest of the vector formats may be imported using [gdal](#) converters.

While the vector data filters can be used by the Map Server at run-time, the data conversion may be performed at the setup time to optimize the run-time performance. The source code for the filters is provided in the *map_server/convert* directory of the map server installation.

The Map Server uses unprojected data in lat/lon and projects them to a requested projection on the fly. Map rotation is performed on the fly as well, if the ANGLE parameter of the map request is different from zero.

Where can I find more GIS data?

There are numerous sources of free GIS data. One of them is GIS Data Depot (<https://data.geocomm.com>), but Google search for "free GIS data" yields many more. There are also numerous companies that sell various GIS datasets on commercial terms.

The **Digital Chart of the World**, **Census 2000 TIGER/Line** datasets, as well as a preprocessed **OpenStreetMap (OSM)** dataset data may be purchased from Generic Logic:

- The **Digital Chart of the World** contains 2GB of the world coverage data, such as political, state and provincial boundaries, major roads, railroads, airports, lakes and rivers, populated areas, vegetation, elevation contours and other coverage.
- The **Census 2000 TIGER/Line** dataset contains 10GB of data containing the street-level details for the entire United States, as well as states, countries, subcounties, highways, water features and populated areas coverages.
- The preprocessed **Open Street Map (OSM)** dataset contains 100 GB of vector data for the whole Earth down to the street level and buildings details.

All datasets are preprocessed into a hierarchical tiled layout for optimum performance and includes a complete set of setup files, ready to be used with the GLG Map Server.

[› View Map Server Image Gallery](#)

How do I use shape files with the GLG Map Server?

The shape files may be used directly by specifying them as the source of data in the FILENAME parameter of the LIF file:

```
FILENAME=/usr/gis_data/shorelines_shp/gshhs_land.shp
```

```
VECTOR FORMAT=SHP
```

The *VECTOR FORMAT* parameter is optional. If it is omitted, the map server will try to determine the data format from the filename's extension.

For maximum performance, shape files may be converted to an easier to read GVF format at the setup time, to avoid costly shape file parsing at the map generation run-time. To convert shape files to GVF format, use the *shp2gvf* shape file conversion utility described in the *Tools and Utilities* chapter of the [GLG Map Server Reference Manual](#). The shape file conversion utility also provides options for processing large number of shape file located in a directory tree.

How do I use XYZ raster format with the GLG Map Server?

The raster formats that are not supported natively are imported using gdal as described [in the following section](#).

How do I use elevation data (DEM/DTED)?

The provided gdal-based raster setup utility converts DEM, DTED and other elevation data files to 16-bit single-channel TIFF file, which is then used in the layer's LIF file the same way as other TIFF files. The LIF's *ELEVATION MODE* attribute is set to the desired elevation mode: 1 to be used as data for elevation queries, or 2 to display elevation using a selected color threshold table. The color threshold table for color display mode is defined in LIF with the *PIXEL MAP* attribute. Refer to [GLG Map Server Reference Manual](#) for more information.

For tiled elevation datasets with large number of data files in a directory tree, the raster setup utility supports recursive mode for processing all files at once. The LIF generation utility described in [the following section](#) may be used to automatically generate setup files for such datasets.

Raster elevation data may also be converted into shaded relief images which are used to show elevation as a three-dimensional shadow.

How do I use shaded relief images?

Shaded relief images are generated from elevation data and used as bump maps textures overlayed on top of the map image. The LIF setup is the same as for other image files, with the addition of the *TRANS TYPE* and *ALPHA* attribute:

```
TRANS TYPE="BUMP MAP"
```

```
# With BUMP MAPs, ALPHA >1. is acceptable
```

```
ALPHA=5.
```

The TRANS TYPE attribute defines the blending function for combining the layer with the background: in this case, the image is used as a bump map applied to the layers beneath it. The ALPHA attribute controls the contrast of the shadows.

Where can I find documentation for the GVF vector format?

GVF is an open ASCII vector format described in the *GVF Filters and Data Converters* chapter of the [GLG Map Server Reference Manual](#). The source code of simple utility functions to write GLM format is provided in the Gvf.c file as a resource for writing custom conversion filters.

The ASCII version of the GVF format is cross-platform. The binary version of the GVF format may also be used for increasing map generation performance. The Map Server includes utility for converting between ASCII to BINARY versions of the GVF files. The conversion utility automatically handles hierarchical trees for tiled vector datasets. Refer to the *Tools and Utilities* chapter of the [GLG Map Server Reference Manual](#) for more information on the conversion utility.

How do I tile large datasets?

Large datasets may be split into smaller tiles to optimize rendering performance. When the map is zoomed on a particular area, only the tiles that are required for rendering the area will be loaded, yielding huge savings in map generation time and memory consumption. The Map Server maintains a tile cache, reusing the tiles from one request to another, eliminating the need to repeatedly read data when the same tile is used in several map generation requests. The maximum size of the tile cache can be controlled on a by-layer basis to limit the Map Server's memory footprint in order to avoid swapping. The Map Server also supports hierarchical tiling for datasets that use directory structure to encapsulate its subsets and increase the speed of the tile search. The Map Server also provides a fallback layer mechanism for displaying a less-detailed layer when a maximum number of tiles is exceeded.

The GLG Map Server provides **tiling utilities** for both raster and vector data. Raster images may be split into smaller rectangular tiles. For vector datasets, the data may also be split into subsets based on a value of a particular attribute. For example, a dataset of cities and towns may be split into several datasets based on the population, so that large cities are saved in a one layer and small towns in another. Another option is splitting large shorelines into smaller segments to optimize performance when only part of the shoreline is visible.

Refer to the *Tools and Utilities* chapter of the [GLG Map Server Reference Manual](#) for more information on the Map Server's tiling utilities.

How do I setup data sets with large number of data files?

Many large GIS datasets contain a big number of data files, located in one directory or kept in subdirectories of a directory tree. The Digital Chart of the World and Census 2000 TIGER/Line datasets are good examples of directory-based datasets.

It would be a quite tedious and time consuming process to generate setup files for such datasets by hand. GLG Map Server provides utilities that allow to combine a collection of such data files into one map server layer and automatically generate setup files.

For **vector data**, convert the data files to GVF format as described in the previous sections for [shape files](#) or [other vector formats](#) first. For the shape files, the *shp2glm* utility provides recursive options for converting all files in a directory tree that match a specified pattern. After converting the data files to the GVF format, use the LIF generation utility described in the *Tools and Utilities* chapter of the [GLG Map Server Reference Manual](#).

The LIF generation utility traverses a directory tree and process all files matching a specified filename pattern, extracting GIS extent information and generating setup file for each data file. The utility also combines GIS extents of all files in each subdirectory and generates an aggregate setup file for each subdirectory. The setup file generated for the top level directory of the dataset is then listed in the layer's setup file (LIF):

```
# Include generated top level sub-LIF (SLF) into layer's LIF
TILES SUBLIF=dataset_name.slf
```

For **raster data**, a gdal-based raster setup utility (described in the *Tools and Utilities* chapter of the [GLG Map Server Reference Manual](#)) is provided to simplify the conversion process when large number of raster files needs to be converted. It traverses a directory tree and converts all files matching a specified filename pattern. It also extracts georeferencing information from the raster files. After that, the LIF generation utility described above is used to automatically generate setup files. The top level generated setup file is then included into the layer's LIF file.

For both vector and raster data, the LIF file contains only layer attributes that do not depend on the actual GIS data provided in the data files. If GIS data is changed or updated, the LIF generation utility may be re-run at any time without a need to change the content of the LIF file.

What if data may change on the fly, such as weather maps?

The GLG Map Server provides redirection and locking features which guarantee data consistency for volatile data. The Map Server will be using the old version of the data

until the new version is completely updated, after which the old version may be safely deleted. The locking mechanism locks the map server for a brief moment while the data link is switched from old to new data, preventing the read errors.

Does Generic Logic provide consulting services for handling custom requests?

Yes, absolutely. Our consulting services division will be happy to assist you with any custom requests.

[› Contact information](#)

Map Server Features HOWTO

Does GLG Map Server comply with the OpenGIS WMS standard?

Yes. The GLG Map Server complies with the following OpenGIS WMS standards:

- [WMS 1.0](#)
- [WMS 1.1](#)
- [WMS 1.1.1](#)
- [WMS 1.3.0](#)

Does it support 24-bit color?

Yes. The GLG Map Server engine is designed for 24 bit TrueColor systems and does not impose any performance penalties for using 24 bit (3 x 8 bit RGB) color.

Does it support TrueType fonts and Unicode?

The answer to both questions is yes. TrueType fonts and Unicode are supported using FreeType font engine. In addition to TrueType fonts, vector fonts are also supported.

How do I add a new font?

By default, all fonts are located in the *map_data/fonts* directory of the Map Server installation. To add a new font, copy the font file into the *fonts* directory and add a line into the SDF file. For example:

```
IMAGE FONT=moderna MgOpenModernaRegular.ttf
```

defines *moderna* font name alias for the *MgOpenModernaRegular.ttf* font file.

How do I define semi-transparent layers?

Just set the *ALPHA* attribute in the LIF file to a value between 0 and 1 to render the layer as semi-transparent. Place the transparent layer on top of other layers.

Both raster and vector layers may be semi-transparent. However, detailed vector layers may require lower ALPHA settings due to multiple vector features rendering at the same image pixels.

Various OpenGL-style blending functions (*TRANS TYPE* LIF attribute) are provided for blending weather maps, shaded relief images and images with transparent colors. Refer

to the *Configuration File Format: TRANS TYPE* chapter of the [GLG Map Server Reference Manual](#) for more information.

How do I obtain lat/lon coordinates of a mouse click?

The GLG Map Server provides coordinate conversion capabilities that are often used to convert x/y image coordinates of the mouse click to lat/lon coordinates on the map. To convert screen coordinates of the point to lat/lon, use the *GetFeatureInfo* request type with *INFO_TYPE=lat_lon*. The *I* and *J* parameters of the request supply the X and Y image coordinates of the point of interest, and an optional *INFO_FORMAT* parameter may be set to either *text/xml* or *text/plain* to specify a desired output format. The map image parameters (i.e. width and height, projection, extent, etc.) needs to be supplied as well in order to convert point's X/Y coordinates to lat/lon location. The map server will respond with the lat/lon coordinates of the point.

For example, the following sample coordinate conversion request:

```
https://www.myserver.com/cgi-  
bin/GlmScript?VERSION=1.3.0&REQUEST=GetFeatureInfo&SRS=AUTO:42003,9001,-  
97.,38.&WIDTH=400&HEIGHT=400&BBOX=-2500000,-  
2500000,2500000,2500000&STRETCH=0&QUERY_LAYERS=default&INFO_TYPE=lat_lon  
&INFO_FORMAT=text/xml&I=200&J=200
```

will produce the following output:

Content-Type: text/xml

```
<GlmLatLonData>  
  <QueryException>None</QueryException>  
  <Lat>38.000000</Lat>  
  <Lon>-97.000000</Lon>  
</GlmLatLonData>
```

The following command will produce the same output from the command line (a proper path to the *GlmMap* executable and *sample.sdf* file may be required):

```
GlmMap -generate -dataset sample.sdf -oGISreq  
"VERSION=1.3.0&REQUEST=GetFeatureInfo&SRS=AUTO:42003,9001,-  
97.,38.&WIDTH=400&HEIGHT=400&BBOX=-2500000,-  
2500000,2500000,2500000&STRETCH=0&QUERY_LAYERS=default&INFO_TYPE=lat_lon  
&INFO_FORMAT=text/xml&I=200&J=200"
```

Note: A query request is used with the *CGI/FastCGI* version of the map server. Refer to the *Map Server API* chapter of the [GLG Map Server Reference Manual](#) for information on the Map Server library methods for querying elevation.

How do I query elevation of a point?

Elevation Query for a Lat/Lon location

To obtain elevation of a point defined by its lat/lon coordinates, the *GetLocationInfo* request is used with *INFO_TYPE=elevation*. The *LON_LAT* parameter provides coordinates of the point of interest. An optional *INFO_FORMAT* parameter may be set to either *text/xml* or *text/plain* to specify a desired output format.

For example, the following request for a point at the sea level:

https://www.myserver.com/cgi-

bin/GlmScript?VERSION=1.3.0&REQUEST=GetLocationInfo&QUERY_LAYERS=elevation&INFO_TYPE=elevation&INFO_FORMAT=text/xml&LON_LAT=10.,-30.

will produce the following output (in the same units as defined in the data file):

Content-Type: text/xml

```
<GlmElevationData>
  <QueryException>None</QueryException>
  <Elevation>0.000000</Elevation>
</GlmElevationData>
```

Note: This example uses *elevation* layer which is not defined in the sample dataset.

The following command will produce the same output from the command line (a proper path to the *GlmMap* executable and *sample.sdf* file may be required):

GlmMap -generate -dataset sample.sdf -oGISreq

"VERSION=1.3.0&REQUEST=GetLocationInfo&QUERY_LAYERS=elevation&INFO_TYPE=elevation&INFO_FORMAT=text/xml&LON_LAT=10.,-30."

Note: A query request is used with the *CGI/FastCGI* version of the map server. Refer to the *Map Server API* chapter of the [GLG Map Server Reference Manual](#) for information on the Map Server library methods for querying elevation.

Elevation Query for an X/Y image point

To obtain elevation of a point defined by a pair of x/y image coordinates, the *GetFeatureInfo* request is used instead of *GetLocationInfo*. The point of interests is defined by its X/Y image coordinates (*I* and *J* requests parameters). The map image parameters (i.e. with and height, projection, extent, etc.) needs to be supplied as well in order to convert point's X/Y coordinates to lat/lon location:

https://www.myserver.com/cgi-

bin/GlmScript?VERSION=1.3.0&REQUEST=GetFeatureInfo&SRS=AUTO:42003,9001,-

97.,38.&WIDTH=400&HEIGHT=400&BBOX=-2500000,-

2500000,2500000,2500000&STRETCH=0&QUERY_LAYERS=default&INFO_TYPE=elevati

on&INFO_FORMAT=text/xml&I=200&J=200

or from the command line (a proper path to the *GlmMap* executable and *sample.sdf* file may be required):

GlmMap -generate -dataset sample.sdf -oGISreq

"VERSION=1.3.0&REQUEST=GetFeatureInfo&SRS=AUTO:42003,9001,-

97.,38.&WIDTH=400&HEIGHT=400&BBOX=-2500000,-

2500000,2500000,2500000&STRETCH=0&QUERY_LAYERS=default&INFO_TYPE=elevati

on&INFO_FORMAT=text/xml&I=200&J=200"

Note: A query request is used with the CGI/*FastCGI* version of the map server. Refer to the *Map Server API* chapter of the [GLG Map Server Reference Manual](#) for information on the Map Server library methods for querying elevation.

How do I set up layers to show more details when the map is zoomed in?

The *MIN_ZOOM* and *MAX_ZOOM* layer attributes in the LIF file control if layer is active for the current zoom level. The layer is switched off if the current zoom is outside of the defined zoom limits. By setting matching zoom limits, several layers may be arranged in such a way that less detailed layer is displayed as an overview, and more detailed layers are turned on as the map is zoomed in. The *MIN_ZOOM* and *MAX_ZOOM* may be set to -1 (default) to disable the limit.

For tiled layers, the *MAX_TILES* attribute can be used to switch to a *FALLBACK_LAYER* when the number of used tiles exceeds *MAX_TILES*. This may be use to set up a chain of fallback layers and provides an alternative that does not require figuring out appropriate zoom factor values.

The *-verbosity* command line option helps in figuring out zoom limits and the maximum number of used tiles. To display a current zoom factor, change *GlmScript* to add "*-verbosity 1*" command line option. To display the number of used tiles, use "*-verbosity 2*".

How do I define a default set of layers to display?

The *LAYERS* parameter of the query string is set to *LAYERS=default* to display a default set of layers. To include a layer into the default set, the "*DEFAULT ON*" parameter may be set to 1 in the layer's LIF file.

To display layers in addition to the default layers, add them to *LAYERS* using comma as a separator. To exclude a layer, use the "-" prefix. For example, *LAYERS="default,states,-grid"* adds a *states* layer and disables the *grid* layer if it was enabled by default.

In addition to the default set of layers, aliases may be used to define a set of layers to be displayed. For example, the following alias:

ALIAS=default2 "political_filled,states,grid"

defines "*default2*" as an alias for a set of the *political_filled*, *states*, and *grid50* layers.

When an alias is used in the *LAYERS* parameter or a map request, all layers listed in the alias will be displayed. The alias value may also contain attribute conditions.

How do I control grid spacing and labeling?

The attributes of a grid layer are defined in its LIF file. The value of *ADAPTIVE* attribute (default is 5) defines an approximate number of grid lines to display. The actual number of grid lines may slightly differ to accommodate rounded grid interval.

Alternatively, *GRID LAT INTERVAL* and *GRID LON INTERVAL* may be set to define the exact grid interval.

Grid labels may be switched on and off by setting *GRID LABELS* attribute to 1 or 0. If the attribute is set to 2, the labels will be displayed on both sides of the image.

The *DECIMAL LABELS* may be set to 0 or 1 to display labels using minutes and seconds or decimal fractions. The rest of the grid label attributes are controlled using standard text label attributes.

How do I alter display of vector features depending on their attribute values?

The attribute threshold tables may be used to alter attribute of a vector feature depending on some other attribute. For example, attribute threshold tables may be used to change the size of the city marker depending on its population, or change the line width and color of the road depending on its FCC code: interstate highway, state highway, etc.

The threshold table may be controlled by a value of the feature attribute as well as the value of the zoom factor. The threshold table may define an absolute value; for numerical attributes it may also define a relative adjustment factor.

The *us_cities.lif* file in the *map_data/layers* directory of the GLG Map Server Installation provides an example of using attribute threshold tables. Refer to the *Configuration Variables: ATTR MAP* chapter of the [GLG Map Server Reference Manual](#) for more information.

How do I show only some vector features based on some criteria?

There are two ways of selecting features to be displayed. **At setup time**, an attribute table can be attached to the layer's visibility attribute in the LIF file to display only some features based on the value of their attributes. This may be used to set up several layers that use the same data file but display different subsets of features. However, for large data files it is more efficient to use Map Server's split utility to split the data file into several smaller files, one for each layer.

To dynamically define a subset of the features to display **at run time**, a list of attribute conditions may be added to a layer name in the *LAYERS* list of the map query to display only the features that match. For example, "*LAYERS=cities@1>=300000*" will display only cities whose population (attribute index 1) is greater or equal 300K. To disable selected feature, add the "-" prefix before the layer name.

Refer to the description of the *LAYERS* parameter in the *Map Query String* chapter of the [GLG Map Server Reference Manual](#) for more information.

How do I define custom icons for point features?

The *ICON FILE* attribute of the LIF file may be used to define a custom icon for a point feature. An attribute threshold may be attached to the IconFile attribute to use different images depending on value of some attribute of the feature. Refer to the *Configuration Variables: ATTR MAP* chapter of the [GLG Map Server Reference Manual](#) for more information.

How do I disable display of vector features which are too small for the selected zoom level?

Set the *MIN PIXEL SIZE* LIF file attribute to the desired size to activate decluttering based on the pixel size. This is often used to eliminate a large number of small lakes in the water coverage layer.

The default value of the attribute is 1. If a layer (such as shorelines or political boundaries) has lines that are split into small segments, the segments may not be drawn if they are smaller than the defined minimum pixel size. If that results in gap in the layer's lines, set the minimum pixel size to 0 to disable small segments elimination.

How do I make labels visible on top of a background of the same color?

Set the *LABEL STYLE* attribute to *OUTLINE* and define *LABEL OUTLINE COLOR* that provides a good contrast with the *LABEL COLOR*. For example, set label outline color to white if label color is black. The outline will be drawn around each character of the text to improve the label's visibility without obscuring the map behind the label. The outline works best with the TrueType fonts (defined by the *IMAGE FONT* attribute).

If obscuring the map behind the label is not a concern, the label style may be set to "BG_FILL,BG_EDGE" to draw a box around the label. The *BOX FILL COLOR* and *BOX EDGE COLOR* attributes define the fill and edge colors of the text box.

How do I prevent layer's labels obscuring each other and labels in other layers?

Set the *LAYOUT TYPE* attribute of the LIF file to *GLOBAL* to enable layout negotiation, which makes sure the labels do not obscure each other. The global layout negotiation is performed for labels in all layers that have *LAYOUT TYPE* set to *GLOBAL*.

The *LAYER* layout type performs layout only for labels within the layer, and may be used for less important labels, such as grid labels. The labels will not obscure each other, but may be obscured by labels in other layers.

The *LABEL PRIORITY* attribute of the LIF file may be used to define priority for the layer's labels: 0, 1, 2 and so on. The labels with the priority value of 0 have the highest priority and will never be obscured by labels with lower priority.

If one layer contains labels with different priority values, an attribute threshold may be attached to the layer's priority resource to vary it depending on the label's type. For example, the *us_cities.lif* LIF file in the sample dataset uses an attribute threshold to assign priority to city labels depending on the city's population.

OS & HARDWARE SUPPORT	INDUSTRY APPLICATION

Note: ** Specifications are copied from manufacturer data. Please contact before buying.