



# OmniMap API Configuration Guide

Version 2.0.3

Last updated: August 15, 2007

IMMERSIVE PROJECTION DESIGN  
**THE ELUMENATI**  
www.elumenati.com

---

# Contents

.....	1
OmniMap API .....	1
Configuration Guide.....	1
Overview.....	3
Intended Audience.....	3
Assumptions.....	3
Setting the Screen Shape.....	4
Advanced Settings.....	5
Dome Radius.....	5
Render Target.....	5
Graphics Rendering Library.....	6
Heads Up Display.....	6
Whether or Not to Consolidate the View parameters Into the Projection Matrix.....	6
Whether or Not the OmniMap Library Sets the Viewing and Projection Matrices.....	7
Multisampling Quality (Direct3D only).....	7
Channel Resolution.....	7
Near and Far Clipping Planes.....	8

© 2007 The Elumenati, LLC. All rights reserved. The Elumenati and the Elumenati logo are registered trademarks of The Elumenati, LLC. Other trademarks and brands are the property of their respective owners. The information in this document belongs to The Elumenati, LLC.

## Notice of non-liability:

The Elumenati, LLC is providing the information in this document to you AS-IS with all faults. The Elumenati, LLC makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. The Elumenati, LLC assumes no liability for damages (whether direct or indirect), caused by errors or omissions, or resulting from the use of this document or the information contained in this document or resulting from the application or use of the product or service described herein. The Elumenati, LLC reserves the right to make changes to any information herein without further notice.

Please send any questions or comments to [support@elumenati.com](mailto:support@elumenati.com).

---

## Overview

This document is a guide for configuring the OmniMap Geometry Correction Library API into existing OpenGL applications. The OmniMap library is designed to allow great flexibility in the way it renders the final image to the screen. That flexibility is provided through Lua scripts (<http://www.lua.org>). These scripts are located in a directory called OmniMapConfig that is required to be located in the working directory of your application. This document describes the configuration parameters that can be set in these scripts. The first section describes the most basic settings. The second section describes more advanced settings that require a deeper knowledge of how OmniMap works.

For more information, see the *OmniMap API OpenGL Integration Guide* and the *OmniMap API Class Documentation* included with the OmniMap installer.

## Intended Audience

This document is written for programmers interested in implementing geometry correction within their real-time OpenGL application for use with the Elumenati OmniFocus range of products. It is also meant as a guide for users that are configuring the OmniMap-enabled applications.

## Assumptions

This document assumes the developer has installed the OmniMap API from the installer program *OmniMap.msi* (available at <http://www.elumenati.com>), and used the default location for the installation. The default installation location for the OmniMap API is:

*C:\Program Files\Elumenati\OmniMap*

In this document, we will refer to this folder as *<InstallationDir>*. So if you have chosen to install the OmniMap API in a location other than the default location, *<InstallationDir>* refers to that location.

---

## Setting the Screen Shape

There are two different ways to set the screen shape to be used. OmniMap first checks to see if the Windows registry variable :

SOFTWARE\Elumenati\Elumenati OmniMap API\OmniMapAPI\DefaultConfig

The value of that registry variable is determines the configuration file used. This value can be set using *OmniConfig™*, Elumenati's dome configuration tool. If this registry variable is not set, the default configuration is set in the file

*OmniMapConfig/omnimap\_user\_edit.lua*. This file enables users to set the OmniMap dome configuration that is being used. The file contains one command that sets the "Screenshape":

Screenshape = "HorizontalDome"

There are other lines that are comment lines(preceded by "--") that show the other possible settings for Screenshape. As of this writing, supported Elumenati screen shapes include:

- HorizontalDome
- VerticalDome
- FullDome
- Cylinder
- Panoramic

Support for other screen shapes is available from Elumenati. Send email to [support@elumenati.com](mailto:support@elumenati.com) for more information.

---

## Advanced Settings

The advanced settings are intended for use by developers of OmniMap enabled applications. These settings can be found in *OmniMapConfig/omnimap\_dome\_wiz\_ai.lua*. This file sets up some global OmniMap parameters, and then calls a function that sets up parameters for the screen configuration specified in *omnimap\_user\_edit.lua*. The configuration parameter in this file include :

1. Dome Radius
2. Render Target
3. The graphics rendering library used.
4. Heads Up Display
5. Whether or not to consolidate the view parameters into the projection matrix.
6. Whether or not the OmniMap library sets the viewing and projection matrices.
7. Multi Sample Quality for Direct3D
8. The near and far clipping plane for channel rendering.
9. Which channels to enable
10. The fields of view for each channel
11. The projector orientation
12. The starting view orientation
13. The dome center
14. The dome orientation

### Dome Radius

The dome radius set with the command:

```
GLOBAL_domeRadius = 3
```

The parameter is in meters. So to change the dome radius to 6 meters:

```
GLOBAL_domeRadius = 6
```

### Render Target

Set by the line :

```
GLOBAL_renderTarget = RT_FRAME_BUFFER_OBJECT
```

The render target for channel rendering can one of three values:

- 1) `RT_FRAME_BUFFER_OBJECT` In OpenGL this causes offscreen rendered channels to use framebuffer objects. In Direct3D, it causes offscreen rendered channels to use a `ID3DXRenderToSurface`. In Direct3D, Applications cannot call

---

BeginScene/EndScene if using a RenderToSurface for channel rendering. In this case, BeginScene/EndScene are called in

OmniMapChannelBase::beginRenderToChannel and

OmniMapChannelBase::endRenderToChannel respectively.

- 2) RT\_PBUFFER In OpenGL this causes offscreen rendered channels to use pbuffer's. In Direct3D 9, this option uses IDirect3DSurface9, and does not call BeginScene/EndScene in .OmniMapChannelBase::beginRenderToChannel and OmniMapChannelBase::endRenderToChannel respectively. So the application is responsible for making those calls. This option is ignored in Direct3D 10.
- 3) RT\_BACK\_BUFFER In both OpenGL and Direct3D 9, this causes offscreen rendered channels to use the back buffer. If you set multisampling to greater than 0.0, back buffers are used in Direct3D. This option is ignored in Direct3D 10.

## Graphics Rendering Library

Set by the line:

*GLOBAL\_Renderer = "D3D" -- for Direct3D 9, and*

*GLOBAL\_Renderer = "D3D10" -- for Direct3D 10, and*

*GLOBAL\_Renderer = "OGL" -- for OpenGL*

## Heads Up Display

The OmniMap API heads up display displays the contents of the render channels that make up the final image in squares in front of the final image. The heads up display is turned on with the line:

*GLOBAL\_DisplayHUD = true*

To turn off the heads up display, just replace *true* with *false*.

## Whether or Not to Consolidate the View parameters Into the Projection Matrix.

To maintain SPI compatibility, the OmniMap library can put both the channel specific camera rotation and the projection matrix into the projection matrix. Otherwise, OmniMap puts the projection changes into the projection matrix, and the camera viewing rotation into the model view matrix. The default is the latter. If the programmer wants to consolidate both matrices into the projection matrix, use the parameter, set with the line:

*GLOBAL\_ConsolidateCameraIntoProjectionMatrix = true*

If the programmer wants to set have the camera rotation in the modelview matrix, use the line:

*GLOBAL\_ConsolidateCameraIntoProjectionMatrix = true*

---

## Whether or Not the OmniMap Library Sets the Viewing and Projection Matrices

Whether or not the OmniMap library sets the viewing and projection matrices is set with the line:

```
GLOBAL_UseOmniMapMatrix = true
```

If the programmer wants complete control over the camera rotation and projection for all channels, use the line:

```
GLOBAL_UseOmniMapMatrix = false
```

## Multisampling Quality (Direct3D only)

The multisampling quality for Direct3D is set with the line:

```
GOBAL_multiSamplingQuality = 0.0
```

In Direct3D 9, the value set here will be used for the Direct3D call `CreateRenderTarget` when the render targets for the channels are set up. Be careful not to set this number too high. Read the Microsoft documentation on `CreateRenderTarget` for more information. If multisampling is set to greater than 0.0, then the back buffer is used for rendering channels. The application must call `BeginScene/EndScene` if using the back buffer for channel rendering.

In Direct3D 10, this value is used to set the sample count in the `DXGI_SAMPLE_DESC` structure when creating the off screen render targets for the channels. The quality value in `DXGI_SAMPLE_DESC` for the channels is always set to 0. See the Microsoft documentation for more information about multisampling. The maximum allowable number of samples can be determined by calling `ID3D10Device::CheckMultisampleQualityLevels`.

## Channel Resolution

Channel resolution is set with the 2 lines:

```
GLOBAL_channelResX = 1024
```

```
GLOBAL_channelResY = 1024
```

This is the resolution of the off screen rendering surface used for rendering each channel. The number should always be a power of 2.

---

## Near and Far Clipping Planes

The near and far clipping planes for channel rendering are set with the lines:

```
GLOBAL_clipNear = .1  
GLOBAL_clipFar = 1000
```