

4.4.1 Documentation

Introduction

Google Earth Enterprise (GEE) 4.4.1 is a network-based, rich 3D and 2D mapping solution that makes vast amounts of data easily accessible from a desktop application. With GEE, you can create a central geographic information system (GIS) database that can be distributed simultaneously to thousands of users.

Documentation

For offline use, documentation is available on the server where you installed GEE. You can find it in the directory `/opt/google/share/doc/manual/` (by default).

The GEE documentation provided with your software includes:

Google Earth Enterprise documentation

Installation Guide - Guides you through preparation for and installation of GEE.

Administration Guide - Covers configuration of Fusion, GEE Server software, and the GEE tutorial work space. Also includes a command reference and a list of common error messages and their resolutions.

Fusion Tutorial - Guides you through the basic steps of creating and publishing a Fusion database from raw source data, and provides a series of advanced lessons for specific data preparation tasks.

Reference Guide - Provides comprehensive reference material about Fusion and how you can use it to create graphically-rich GIS databases for distribution to your in-house users.

Portable Globe and Server - Describes the portable globe and server: installation of the server, cutting globes, serving and broadcasting, and command line tools.

Search Framework Developer's Guide - Explains how to use the Search Framework Java API to write a plug-in application for Google Earth or Google Maps to search a custom data source. The plug-in creates requests and responses in the framework's Common Search Format, a generic format that can be adapted to deal with data from any data store.

Release Notes - Describes the latest feature enhancements and improvements for GEE.

Google Earth Client documentation

Google Earth User Guide - Describes how users can view published databases in Google Earth Client (EC), as well as common functions in the Earth Client.

Earth Enterprise Client and Earth Plugin Release Notes - Describes updates and known issues in EC and the Earth Plugin.

API documentation

Google Maps API

Google Maps API Developer's Guide and Reference - The Google Maps API lets you embed Google Maps in your own web pages with JavaScript. The API provides a number of utilities for manipulating maps and adding content to the map through a variety of services, allowing you to create robust maps applications on your website.

Google Maps API extensions for GEE - A special extension class called **GFusionMap** is

needed to view 2D Maps hosted from a GEE Server. Additional information about building 2D Maps is available in the **Google Earth Enterprise Reference Guide**.

Google Earth API

Google Earth API Developer's Guide and Reference - The Google Earth API allows you to embed three-dimensional maps into web pages.

Google Earth API extensions for Google Earth Enterprise - Additional information about connecting the Google Earth API to a local GEE Server.

Requesting technical support

Google Earth Enterprise Technical Support - Technical Support is available to all current GEE customers from a dedicated, web-based [Support Portal](#). An account is required to access the portal. Please [complete this form](#) if you do not have an account yet.

[Legal Notices and License Agreements](#)

[Back to top](#)

[4.4.1 Documentation](#)**Administration Guide**[Home](#)[Configuring Server](#)[Configuring Fusion](#)[Configuring Tutorial Work Space](#)[Command Reference](#)[Common Server Error Messages](#)[Multiple Database Support](#)

Legal Notices

Legal Notices

Copyright © 2008, 2009, 2010, 2011, 2012, 2013 Google, Inc.
All rights reserved.

Google, Google Earth, and Keyhole are trademarks or registered trademarks of Google, Inc. in the United States and elsewhere. Other trademarks not owned by Google, Inc. are the property of their respective owners.

All Google and Keyhole logos and graphics are copyrighted. The content of this publication is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Google, Inc.

This publication is provided pursuant to and subject to the terms of the Google Earth Enterprise [Purchase Agreement](#) and [Acceptable Use Policy](#). This publication and the information herein may not be reverse engineered, reproduced, duplicated, modified, altered, disclosed, published, printed, or distributed to any third party whatsoever without the express written permission of Google, Inc.

Google Earth Enterprise may utilize the following products as integrated components, linked libraries, or optional modules. No usage rights, expressed or implied, are provided for said products independent of Google Earth Enterprise, unless otherwise stated in the product documentation. All trademarks are the property of their respective owners.

- Apache
- eXpat
- GDAL
- GEOS
- Kakadu
- libcURL
- libgeotiff
- libjs
- LibTIFF
- MrSID
- OGD
- OpenLDAP
- OpenSSL
- PostGIS
- PostgreSQL
- PROJ
- Tomcat
- Tornado

Portions of this computer program are copyright © 1995-2008 Celartem, Inc., doing business as LizardTech. All rights reserved. MrSID is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

[4.4.1 Documentation](#)**Administration Guide**[Home](#)[Configuring GEE Server](#)[Configuring Fusion](#)[Configuring Tutorial Work Space](#)[Command Reference](#)[Common Server Error Messages](#)[Multiple Database Support](#)[Legal Notices](#)**About This Guide**

This document discusses system administration for the Google Earth Enterprise (GEE) system. This includes hardware and network preparation steps, as well as installation, configuration, and maintenance for all system components:

- [Fusion](#)
- [GEE Server](#)
- [Fusion Tutorial](#)

Audience

This guide is intended for individuals who are responsible for installing, configuring, and maintaining any part or all of the Google Earth Enterprise (GEE) system.

In This Guide

This guide includes the following chapters:

- [Configuring Google Earth Enterprise Server](#) explains how to configure GEE Server.
- [Configuring Google Earth Enterprise Fusion](#) explains how to configure Fusion.
- [Configuring Tutorial Work Space](#) explains how to set up a separate work space for each user to work on the Fusion Tutorial, as well as how to remove the tutorial data when each user is finished.
- [Command Reference](#) provides a list of all of the system administration commands you can use to configure the GEE system.
- [Common Server Error Messages](#) provides a list of common error messages that you might encounter when you install and configure the GEE system.
- [Multiple Database Support](#) describes the feature in Google Earth Client (EC) 6.0 that supports simultaneous connection to multiple databases.

[4.4.1 Documentation](#)**Administration Guide**[Home](#)[Configuring Server](#)[Configuring Fusion](#)[Configuring Tutorial Work Space](#)[Command Reference](#)[Common Server Error Messages](#)[Multiple Database Support](#)[Legal Notices](#)**Configuring Google Earth Enterprise Server**[Overview](#)[Verifying that Server is Running](#)**Configuring:**

- [Virtual Servers](#)
- [Server Associations](#)
- [Secure Servers](#)
- [The Publish Root](#)
- [Authentication](#)
- [Secure Remote Publishing \(Using an SSH Tunnel\)](#)

[Disconnected Publishing](#)**Overview**

Before you use many of the Google Earth Enterprise Server configuration tools, you must stop the Google Earth Enterprise Server and then start it after you change the configuration. To stop/start the Google Earth Enterprise Server, enter:

```
/etc/init.d/geserver [stop | start | restart]
```

In general, no additional configuration is required after you install the Google Earth Enterprise Server. However, there are a number of ways in which you can customize your Google Earth Enterprise Server configuration, if desired. This chapter provides information about those configuration options. (See the [Command Reference](#) chapter for a complete list of commands available to system administrators.)

The `geserveradmin` command handles many of the Google Earth Enterprise Server configuration changes you might want to make. All of the options and uses for `geserveradmin` are described in the [geserveradmin](#) section of the **Command Reference** doc.

Verifying That Google Earth Enterprise Server Is Running

Because Google Earth Enterprise Server is critical to many operations you perform with Google Earth Enterprise Fusion, verify that it is running.

1. Open a web browser (either on the machine where the server is installed or any other computer that has network access to that machine).
2. Enter the URL for the server host (such as `http://www.example.com`).

If Google Earth Enterprise Server is running, the browser displays the Google logo.

If Google Earth Enterprise Server is not running, the browser displays an error message informing you that the page cannot be found. On the command line, enter:

```
/etc/init.d/geserver start
```

If you again reach an error page, make sure that you have network access to the machine on which the server is installed and that the DNS entry for that machine is up-to-date.

Configuring Virtual Servers

Before publishing to virtual servers other than the default virtual servers, please familiarize yourself with the Apache Virtual Host documentation (<http://httpd.apache.org/docs/2.2/vhosts/>).

Caution: Publishing to virtual servers other than the default server is supported only in version 4.2 or later of Google Earth EC. If you are using version 4.0 or earlier, only databases that you publish to the default server can be accessed by Google Earth EC.

Defining the virtual server

To define a virtual server, use the `geserveradmin` command (see [geserveradmin](#) for more details):

```
geserveradmin --addvs name --vstype type --vsurl url
```

Where:

- *name* is the name of the virtual server
- *type* is either `ge` (for 3D streaming) or `map` (for maps)
- *url* is the location of the virtual server. This will usually look like `/somename_ge`; refer to [geserveradmin](#) in the **Command Reference** chapter for more information on the required format.

Creating the configuration file

After defining a virtual server, create a configuration file that specifies required information for that particular virtual server. To do so:

1. Log in to the server you want to configure.
2. Change to the directory where the sample configuration files are stored:

```
cd /opt/google/gehttpd/conf.d/examples
```

3. Copy a sample configuration file to `/opt/google/gehttpd/conf.d/virtual_servers` and rename it. For example:

```
cp /opt/google/share/gehttpd/examples/port_based_ge_vs_example.vhost  
/opt/google/gehttpd/conf.d/virtual_servers/private_ge.vhost
```

When choosing a configuration file to copy, use a file that:

- contains `_ge_` if you are creating a Google Earth Enterprise Fusion virtual server.
- contains `_map_` if you are creating a Google Maps virtual server.
- starts with `location_based` if you are creating a location-based virtual server.
- starts with `name_based` if you are creating a name/domain-based virtual server.
- starts with `port_based` if you are creating a port-based virtual server.

You must maintain the `.location` or `.vhost` extension on the file name.

4. Edit the new file with the text editor of your choice, making the following changes:

In a location-based configuration file:

- a. Change `<LOCATION>` to the `vsurl` value specified in the `geserveradmin` command you used to register the virtual server.

Note: For a Google Maps virtual server, you must make this change in five places.

- b. Change `<VS_NAME>` to the `addvs` name specified in the `geserveradmin` command you used to register the virtual server.

In a name-based configuration file:

- a. Change `<HOST.DOMAIN>` to the desired server name.
- b. Change `<VS_NAME>` to the `addvs` name specified in the `geserveradmin` command you used to register the virtual server.

In a port-based configuration file:

- a. Change `<IP_ADDRESS>` to the IP address of the host.

- b. Change <PORT> to the number of the port you want to use.
- c. Change <VS_NAME> to the `addvs` name specified in the `geserveradmin` command you used to register the virtual server.

Note: You can optionally create combinations of name-based or port-based and location-based virtual servers. If you do, you must create an Apache configuration file with the `.vhost` extension that references multiple virtual servers. Make sure you create the corresponding virtual servers using the `geserveradmin --addvs` command. (See [geserveradmin](#) for more information.)

5. Save the file.

Next, you must configure a server association for the new virtual server.

Configuring Server Associations

The Google Earth Enterprise Server consists of a stream server and a search server. You can publish a Google Earth Enterprise Fusion database to stream and search virtual servers running on your local workstation or on a remote server. Associated stream and search virtual servers can be located on different physical servers or on the same server.

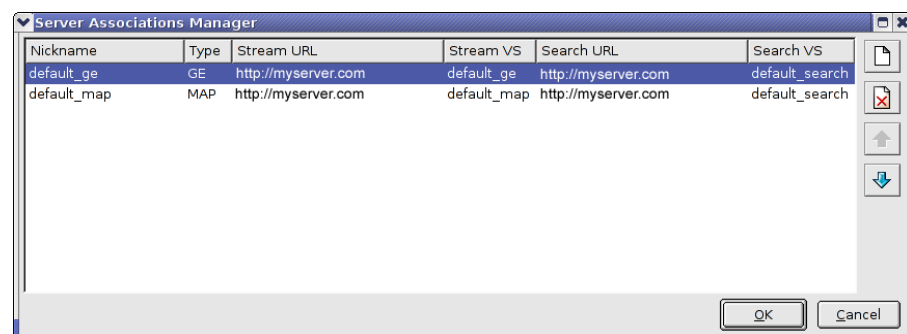
The **Server Associations Manager** allows you to specify server associations in preparation for publishing your databases. You can specify any number of additional server associations in the Server Associations Manager. Then you can select the desired server association in the **Publish Database** dialog. (Refer to **Publishing a Database** in the *Google Earth Enterprise Fusion Reference Guide* for more information.)


Caution: If you're working with multiple Google Earth Enterprise Fusion users on multiple workstations, it is important to remember that all managers on the **Tools** menu can be accessed by all users at the same time. If multiple users are working with the same manager at the same time, when one user closes the manager, that user's changes overwrite all previous data for that manager. So if you are working in a multi-user environment, be sure to coordinate with the other users to be sure that only one user has this manager open at a time.

To specify server associations:

1. Select **Tools > Server Associations Manager**.

The Server Associations Manager window appears with two default server associations listed--one for publishing to Google Earth and the other for publishing to Google Maps. The default server associations assume that you have Google Earth Enterprise Server installed on the same workstation as Google Earth Enterprise Fusion.



2. To add a server association, click the page icon . The Edit Server Association dialog appears.

Edit Server Association

Nickname: Type: **GE**

Stream Server

URL:

Virtual Server:

Query

Search Server

URL:

Virtual Server:

Query

OK **Cancel**

3. Enter a unique name for the server association in the **Nickname** field. (When you're ready to publish a database, the nickname appears on the list of server associations to which you can publish.)
4. For **Type**, select **GE** for Google Earth databases or **MAP** for Google Maps databases.
5. Enter the URL for the Stream Server (and a port number, if other than 80; for example, `http://www.example.com:9090`).
6. Click **Query** to populate the **Virtual Server** drop-down list with all of the virtual servers available at that URL (and port, if specified).
7. Repeat for the Search Server.
8. Click **OK**. The new server association appears on the list in the **Server Associations Manager**.

Server Associations Manager

Nickname	Type	Stream URL	Stream VS	Search URL	Search VS
default_ge	GE	http://myserver.com	default_ge	http://myserver.com	default_search
default_map	MAP	http://myserver.com	default_map	http://myserver.com	default_search
Home Serv	GE	http://another.server.com	private_ge	http://another.server.com	private_search

OK **Cancel**

Repeat the steps above for each server association you want to add. The server associations appear on this list in the order in which you added them. They also appear on the **Server Associations** drop-down list in the **Publish Database** dialog in the same order. The first server association listed here is the default in the **Publish Database** dialog.

If you want to reorder the server associations on the list (particularly if you publish to a certain server most of the time and want it to be the default in the Publish Database dialog), select the server association that you want to move, and click the up or down arrows to change that server association's position in the list.

9. When you have finished adding and sorting server associations, click **OK**. The server associations are ready for you to select when you publish a database. (Refer to **Publishing a Database** in the *Google Earth Enterprise Fusion Reference Guide* for more information.)

Configuring Secure Servers

You can configure your Google Earth Enterprise Server to use HTTPS connections by editing the virtual server's configuration file. Each time you restart Google Earth Enterprise Server, publish a database, or start Apache, Apache reads this file.

To configure a secure server:

1. Enable the main Apache server to run HTTPS by activating the SSL engine. (If you need assistance with this step, read the Apache HTTP server documentation at <http://httpd.apache.org/docs/2.2/>.)

2. In a text editor, open the configuration file (`gehttpd.conf`) for the virtual server you want to make secure.

The configuration file is stored in `/opt/google/gehttpd/conf.d/virtual_servers`. (See [Configuring Virtual Servers](#) for details.) The contents of the configuration file look like:

```
<LocationMatch "/private_ge/*">
    Include conf.d/virtual_servers/runtime/private_ge_runtime
</LocationMatch>
```

3. Add the two bold lines below to the `LocationMatch` directive exactly as they appear here:

```
<LocationMatch "/private_ge/*">
    Include conf.d/virtual_servers/runtime/private_ge_runtime
    SSLRequireSSL
    SSLVerifyClient none
</LocationMatch>
```

4. Save and close the configuration file.
5. Restart the Apache service.

This procedure gets you to a point where you have “bare bones” security on your server. It requires that users use the HTTPS protocol instead of HTTP when they specify a location. However, if you want to add more security, you can do so by following the instructions provided in the Apache 2.2 documentation at <http://httpd.apache.org>.

Configuring the Publish Root

When you install the Google Earth Enterprise Server, you specify a location for your publish root, the directory in which all of your published databases are stored.

To add a publish root:

1. At the command line prompt, log in as root.
2. Stop the Google Earth Enterprise Server:

```
/etc/init.d/geserver stop
```

3. Specify the location of the new publish root:

```
geconfigurepublishroot
```

When prompted, enter the path where you want to create the publish root. For example, if you have designated a separate volume for publishing called `/gepub`, enter `/gepub/published_dbs`.

4. When you see the success message, start your Google Earth Enterprise Server:

```
/etc/init.d/geserver start
```

5. Log out as root.

When you create a new publish root, it is automatically selected. The next time you publish a database, Google Earth Enterprise Fusion stores it in the selected publish root.

If you have multiple publish roots, you can go back and forth among them, using different publish roots for different purposes.

Note: If you have multiple publish roots, be sure to select the corresponding *asset root* for the intended publish root before you publish. Otherwise, it could end up with an incompletely published database. (See [geselectpublishroot](#) in the **Command Reference** chapter for more information.)

To select a different publish root:

1. At the command line prompt, log in as root.
2. Stop the Google Earth Enterprise Server:

```
/etc/init.d/geserver stop
```

3. Select the desired publish root:

```
geselectpublishroot path
```

where *path* is the full path to the publish root, such as `/gepub/published_dbs`.

4. When you see the success message, start your Google Earth Enterprise Server:

```
/etc/init.d/geserver start
```

5. Log out as root.

Configuring Authentication

Authentication is the process of verifying the identity of a user, a system, or a process. *Controlled access* refers to any content or process that requires authentication. The following sections describe how to configure:

- Publishing authentication - To restrict the users allowed to publish databases
- Client authentication - To restrict the users allowed to view published databases

Configuring Publishing Authentication

By default, Google Earth Enterprise Server does not require authentication for publishing. However, you can configure your server to control access for publishing using standard Apache authentication methods. This section is primarily for system administrators who are unfamiliar with configuring Apache authentication.

First, decide which authentication method you want to use.

- HTTP Digest authentication (enable the `mod_auth_digest` Apache module)
- HTTP Basic authentication (enable the `mod_auth_basic` Apache module)

Both methods are essentially the same, except that `mod_auth_digest` encrypts the users' password, and `mod_auth_basic` does not. However, `mod_auth_digest` is not supported by all browsers. See http://httpd.apache.org/docs/2.2/mod/mod_auth_digest.html and http://httpd.apache.org/docs/2.2/mod/mod_auth_basic.html for complete details on these two authentication methods.

Google strongly encourages you to use HTTP Digest authentication (`mod_auth_digest`) instead of HTTP Basic authentication. The following procedure describes configuring publishing authentication using HTTP Digest.

To configure publishing authentication on your server:

1. Create a password file by entering on the command line:

```
/opt/google/gehttpd/bin/htdigest -c path_to_pwd_file Publisher user_name
```

The first part is the path to the Apache utility you use to create the password file, `htdigest`. The `-c` option indicates that you are creating a new file. The *path_to_pwd_file* is the location where you want to store the password file and the name of that file; it can be any name and location you choose. *Publisher* is the realm, and you must enter it exactly as it appears above. The *user_name* is the first user name you want to add to the password file.

When you press **Enter**, the utility prompts you to enter a password for the specified user.

2. Enter the password, and retype it when prompted.

The `htdigest` utility creates the password file, which contains only that one user name, the realm, and the encrypted password. For example, if you view the contents of the password file, it might look like this:

```
jsmith:Publisher:f9d3c803aeb515f35568b37ec2d65399
```

3. Add each user name and password to the same password file by entering:

```
/opt/google/gehttpd/bin/htdigest path_to_pwd_file Publisher user_name
```

Use the same *path_to_pwd_file* as you used for the first user. Do not include the `-c` option, since you are not creating a new file this time.

4. When you finish adding users to the password file, copy the `publisher_digest_auth.conf` file from `/opt/google/gehttpd/conf.d/examples` to `/opt/google/gehttpd/conf.d`.
5. Open the copy (`/opt/google/gehttpd/conf.d/publisher_digest_auth.conf`) in an editor and:
 - Uncomment the `<Location>` directive(s) for stream publisher and/or search publisher by removing the `#` at the beginning of each line. (If you want to configure authentication for one type of server only, uncomment only its directive. If you want to configure both the search server and the stream server, uncomment both directives.)
 - Change `<<PATH_TO_USER_FILE>>` to the password file path (the value of *path_to_pwd_file* in step 1).
6. Save and close the `publisher_digest_auth.conf` file.
7. Restart the server by entering:

```
/etc/init.d/geserver restart
```

When a server is configured for publishing authentication, Google Earth Enterprise Fusion users who try to publish to that server must enter their user name and password. Refer to **Publishing a Database** in the *Google Earth Enterprise Fusion Reference Guide* for more information.

Configuring Client Authentication

Configuring client authentication is very similar to configuring publishing authentication, except that client authentication is specific to each virtual server, whereas publishing authentication is not.

To configure client authentication for a virtual server:

1. Create a password file by entering on the command line:

```
/opt/google/gehttpd/bin/htdigest -c path_to_pwd_file Client user_name
```

The first part is the path to the Apache utility you use to create the password file, `htdigest`. The `-c` option indicates that you are creating a new file. The *path_to_pwd_file* is the location where you want to store the password file and the name of that file; it can be any name and location you choose. `Client` is the realm. You can use any realm you want; however, you must use the same realm in step 4. The *user_name* is the first user name you want to add to the password file.

Note: Set the umask to `0022` before using `htdigest` to create the `htdigest_password_file` or change permissions to `rw-r--r--` (644) after the `htdigest_password_file` is created. Check the permissions for the `htdigest_password_file` and if the permissions are set to `rw-r-----`, change the group ownership to `gegrou`.

2. Press **Enter**. The utility prompts you to enter a password for the specified user. Enter the password, and retype it when prompted.

The `htdigest` utility creates the password file, which contains only that one user name, the realm, and the encrypted password. For example, if you view the contents of the password file, it might look like this:

```
jsmith:Client:f9d3c803aeb515f35568b37ec2d65399
```

3. Add each user name and password to the same password file by entering:

```
/opt/google/gehttpd/bin/htdigest path_to_pwd_file Client user_name
```

Use the same *path_to_pwd_file* as you used for the first user. Do not include the `-c` option, since you are not creating a new file this time.

- When you finish adding users to the password file, open the `/opt/google/gehttpd/conf.d/virtual_servers/config_file` file (where *config_file* is the configuration file for the virtual server you want to protect) in an editor, and add the following lines at the beginning of the `<Location>` directive:

```
AuthType Digest
AuthName "Client"
AuthDigestProvider file
AuthUserFile path_to_pwd_file
Require valid-user
```

Where `Client` is the exact name of the realm and *path_to_pwd_file* is the same path you used in the `htdigest` command in step 1.

If this is a `_ge_` virtual server, add:

```
BrowserMatch "GoogleEarth" AuthDigestEnableQueryStringHack=On
```

If this is a `_map_` virtual server, add:

```
BrowserMatch "MSIE" AuthDigestEnableQueryStringHack=On
```

For more information about `AuthDigestEnableQueryStringHack`, refer to the Apache documentation at http://httpd.apache.org/docs/2.2/mod/mod_auth_digest.html#msie.

The final `<Location>` directive looks like:

```
<LocationMatch /default_ge/*>
    AuthType Digest
    AuthName "Client"
    AuthDigestProvider file
    AuthUserFile path_to_pwd_file
    Require valid-user
    Include conf.d/virtual_servers/runtime/default_ge_runtime
    BrowserMatch "GoogleEarth" AuthDigestEnableQueryStringHack=On
</LocationMatch>
```

- Save and close the virtual server configuration file.
- Restart the server by entering:

```
/etc/init.d/geserver restart
```

Configuring LDAP Authentication For Publishing

If you use LDAP for user authentication on your network, you can configure LDAP authentication for publishing with Google Earth Enterprise Fusion. Since Basic authentication is used for password challenge, Google suggests that you publish through an SSH tunnel. (See [Configuring Secure Remote Publishing \(Using an SSH Tunnel\)](#) for more information about configuring an SSH tunnel.)

To securely configure LDAP authentication for publishing:

- Copy the `publisher_ldap_auth.conf` file from `/opt/google/gehttpd/conf.d/examples` to `/opt/google/gehttpd/conf.d`.
- Open `/opt/google/gehttpd/conf.d/publisher_ldap_auth.conf` in an editor.
- Uncomment the following lines to enable LDAP authentication on the stream server:

```
<Location /StreamPublisher/>
    AuthType Basic
    AuthName "Publisher"
    AuthBasicProvider ldap
    AuthzLDAPAuthoritative off
    AuthLDAPURL "<<LDAP_SERVER_URL_and_LDAP_SEARCH_PARAMETERS>>"
    Require valid-user
</Location>
```

- Change `<<LDAP_SERVER_URL_and_LDAP_SEARCH_PARAMETERS>>` to the actual path to your LDAP server and any parameters you want to set, for example:

```
ldap://ent-test-ldap.mycompany.com/dc=mycompany,dc=com?uid?sub?(objectClass=*)
```

5. Uncomment the following lines to enable LDAP authentication on the search server:

```
<Location /SearchPublisher/>
  AuthType Basic
  AuthName "Publisher"
  AuthBasicProvider ldap
  AuthzLDAPAuthoritative off
  AuthLDAPURL "<<LDAP_SERVER_URL_and_LDAP_SEARCH_PARAMETERS>>"
  Require valid-user
</Location>
```

6. Change <<LDAP_SERVER_URL_and_LDAP_SEARCH_PARAMETERS>> to the actual path to your LDAP server and any parameters you want to set.
7. Save the file.

Configuring LDAP Authentication For Virtual Servers

If you use LDAP for user authentication on your network, you can configure LDAP authentication for any virtual servers you configure. As with LDAP authentication for publishing, since the AuthDigestProvider directive does not support LDAP and AuthBasicProvider does not encrypt passwords, please contact Google technical support for help configuring secure LDAP authentication for virtual servers.

Securing Your Search Server

If you publish a database that you want to keep private and allow access only to certain users, configuring client authentication provides that security with regard to the stream server. However, if you host a searchable private database on the same server with a publicly accessible database, users who are able to log in to the public database can, theoretically, search the private database on that server. Such a user would need to be very knowledgeable about Google Earth Enterprise Fusion and would need to know the exact URL of the private database, so the likelihood of this breach is small; however, it is possible.

Therefore, to ensure that a database is completely secure (that is, no unauthorized users can view or search the data), you must publish that database on a separate search server host that requires authentication and allows access to only approved users.

Note: If you are short of server space, you can publish the private database to a shared stream server (that is, a mix of databases that do and do not require authentication) and just publish the private database to a search server on a separate (authentication only) host.

Configuring Secure Remote Publishing (Using an SSH Tunnel)

When Google Earth Enterprise Fusion publishes a database, it does not encrypt the data. If you want to encrypt the data during the publishing process to a remote server (or even on separate remote stream and search servers), you can use an SSH tunnel.

To publish securely to a remote server:

1. Open an SSH tunnel by entering on the command line:

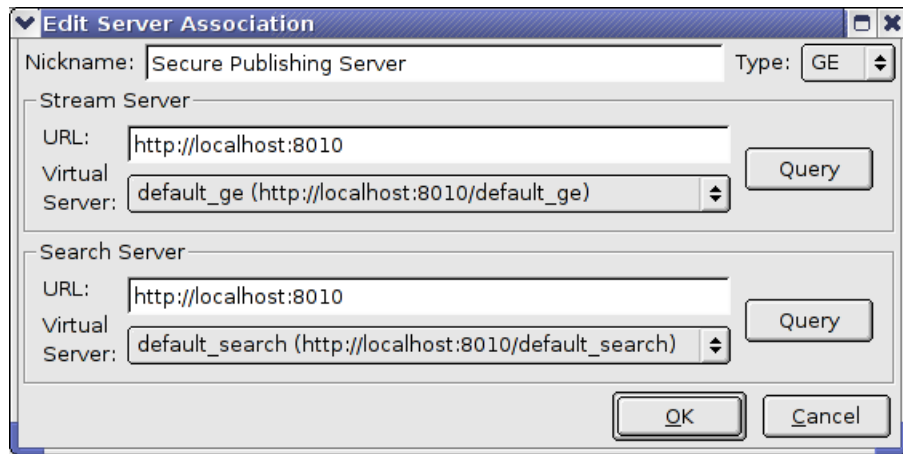
```
ssh -L port_num:host_name:80 host_name
```

For example:

```
ssh -L 8010:remoteServer:80 remoteServer
```

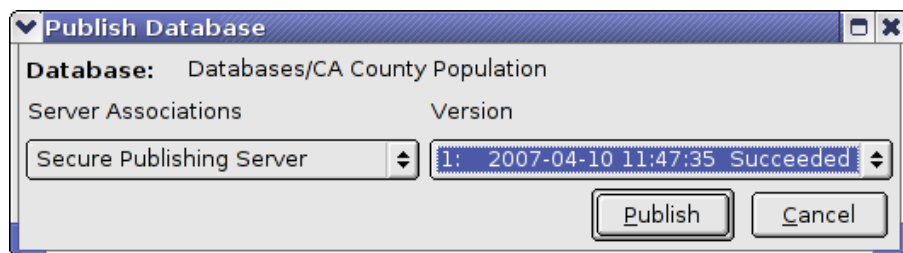
Note: You can use any port that is available, but Google recommends a high number.

2. Follow the instructions in [Configuring Server Associations](#) to configure a server association using the local port number you designated in the `ssh` command, for example:



Note: If the remote server requires authentication, it prompts you to log in when you click **Query**.

3. Publish the database to the server association you just created, for example:



4. Close the SSH tunnel by entering **Ctrl-D** or `exit` in the terminal window.

The next time you want to publish securely to that same remote server, you can skip step [2](#).

To publish securely to separate remote search and stream servers:

1. In one terminal window, open an SSH tunnel to the stream server by entering on the command line:

```
ssh -L port_num:host_name:80 host_name
```

For example:

```
ssh -L 8010:remoteServer:80 remoteServer
```

Note: You can use any port that is available, but Google recommends a high number.

2. In a different terminal window, open an SSH tunnel to the search server by entering the same command with a different port number, for example:

```
ssh -L 8020:remoteServer:80 remoteServer
```

3. Follow the instructions in [Configuring Server Associations](#) to configure a server association:
 - Use the local port number you designated for the stream server in the Stream Server's URL field.
 - Use the local port number you designated for the search server in the Search Server's URL field.
4. Publish the database to the server association you just created.
5. Close both SSH tunnels by entering **Ctrl-D** or `exit` in the terminal windows.

The next time you want to publish securely to that same remote server association, you can skip step [3](#).

Disconnected Publishing

If you want to publish databases on a server that does not have a network connection to your Google Earth Enterprise Fusion workstations, you can configure a server with the disconnected server add-on.

New in Google Earth Enterprise 4.0 is the ability to combine disconnected publishing and connected publishing together. Prior to 4.0, only one type was supported for a database. With this change, for example, a large database can be copied to portable media and loaded onto the server using disconnected publish, saving the time and bandwidth required to send terabytes of data over the network. Later, deltas for the same databases can be sent over the wire.

Note: Disconnected production servers require X11 libraries, including Mesa. Google Earth Enterprise Server does not require an X server to be running or installed, however. Only the libraries are required.

Prepare a disconnected database

To create the disconnected database, use `gedisconnectedsend` from the command line. In the example below, the `--report_size_only` flag is turned on; the size of the database should be compared to the free space on your removable media.

```
gedisconnectedsend --report_size_only --sendpath \  
/gevol/assets/Databases/igain.kdatabase/geodb.kda/ver001/geodb  
  
rm -rf /portable_media/new_dir  
mkdir -p /portable_media/new_dir  
  
gedisconnectedsend --output /portable_media/new_dir \  
--sendpath /gevol/assets/Databases/igain.kdatabase/geodb.kda/ver001/geodb
```

Making delta disconnected databases

Use `--havepath` or `--havepathfile` to create a delta database.

When publishing from `machine_one` to `machine_two`, for example:

On `machine_two`:

```
geserveradmin --fusion_host machine_one --listdbs > /portable_media/dblist
```

On `machine_one`:

```
gedisconnectedsend --output /portable_media/new_dir \  
--havepathfile /portable_media/dblist \  
--sendpath /gevol/assets/Databases/igain.kdatabase/geodb.kda/ver001/geodb
```

Publish on the remote Server host

Connect the portable media to the Server host and mount the drive. Then:

1. Add the database manifest to the Server:

```
geserveradmin --stream_server_url http://your_stream_server \  
--search_server_url http://your_search_server \  
--adddb \  
/portable_media/new_dir/gevol/assets/Databases/igain.kdatabase/geodb.kda/ver001/geodb
```

2. Push the files to the Server:

```
geserveradmin --stream_server_url http://your_stream_server \  
--search_server_url http://your_search_server \  
--pushdb \  
/portable_media/new_dir/gevol/assets/Databases/igain.kdatabase/geodb.kda/ver001/geodb
```

3. Publish the database on the Server:

```
geserveradmin --stream_server_url http://your_stream_server \  
--search_server_url http://your_search_server \  
--publishdb \  
/portable_media/new_dir/gevol/assets/Databases/igain.kdatabase/geodb.kda/ver001/geodb
```

```
/portable_media/new_dir/gevol/assets/Databases/igain.kdatabase/geadb.kda/ver001/geadb
```

4. Verify the database manifests:

```
geserveradmin --stream_server_url http://your_stream_server \  
--search_server_url http://your_search_server \  
--listdbs  
  
geserveradmin --stream_server_url http://your_stream_server \  
--search_server_url http://your_search_server \  
--dbdetails \  
/portable_media/new_dir/gevol/assets/Databases/igain.kdatabase/geadb.kda/ver001/geadb
```

5. Verify the integrity of files in the published database:

```
ssh your_server /opt/google/bin/geindexcheck --database \  
/gevol/published_dbs/stream_space/your_server/gevol/assets/  
Databases/igain.kdatabase/geadb.kda/ver001/geadb  
  
ssh your_server /opt/google/bin/geindexcheck --mode all --database \  
/gevol/published_dbs/stream_space/your_server/gevol/assets/  
Databases/igain.kdatabase/geadb.kda/ver001/geadb
```

Note: The paths in the examples above were split onto two lines for documentation. They should be entered on one line when completing your commands.

The steps to delete DBs at end of life, for disconnected servers, is same as what was described in over the wire publishing section.

[Back to top](#)

4.4.1 Documentation

Administration Guide

[Home](#)

[Configuring Server](#)

[Configuring Fusion](#)

[Configuring Tutorial Work Space](#)

[Command Reference](#)

[Common Server Error Messages](#)

[Multiple Database Support](#)

[Legal Notices](#)

Configuring Google Earth Enterprise Fusion

[Before You Begin](#)

[Configuring Your Data Locations](#)

[Configuring Font Size in the GUI](#)

[Configuring Fonts for the Text Style Dialog](#)

[Customizing Your Google Maps Landing Page](#)

[Using the dbRoot Snippet Manager](#)

[Monitoring System Processes](#)

[Backing Up Your Data](#)

[Running Fusion on a Machine with Multiple CPU Cores](#)

Before You Begin

Before you use many of the Fusion configuration tools, you must stop the system manager and then start it after you change the configuration. To stop/start the system manager, enter:

```
/etc/init.d/gefusion [stop | start | restart]
```

Configuring Your Data Locations

The Google Earth Enterprise installation script prompts you to enter information about your system and then properly configures your primary asset root and source volume. After you install Fusion, use the `geconfigureassetroot` command to:

- **Add asset roots.**

If you use the *Google Earth Enterprise Fusion Tutorial*, Google recommends that you specify separate asset roots for each user's tutorial data and a completely different asset root for your real working data.

- **Identify source volumes.**

You must identify each directory that contains your source data files (or subdirectories of your source data files). Fusion is not able to read files located in other directories.

- **Modify the current volume.**

You can modify volume definitions when migrating from a single- to a multiple-workstation configuration or when modifying the local path of a network-mounted source volume (for example, when adding a larger drive for source data).

For more information on these commands, see [geconfigureassetroot](#) in the **Command Reference** chapter.

Caution: Do not modify the volume definition after you save data to that volume. Because the local name and the network path definitions are used by Fusion for resources, projects, and database definitions, any change to a volume definition invalidates the data already processed on that volume.

If you add multiple asset roots, you can use the `geselectassetroot` to specify the volume in which you want to work and to switch back and forth among the available asset roots, if desired. For more information on these commands, see [geselectassetroot](#) in the **Command Reference** chapter.

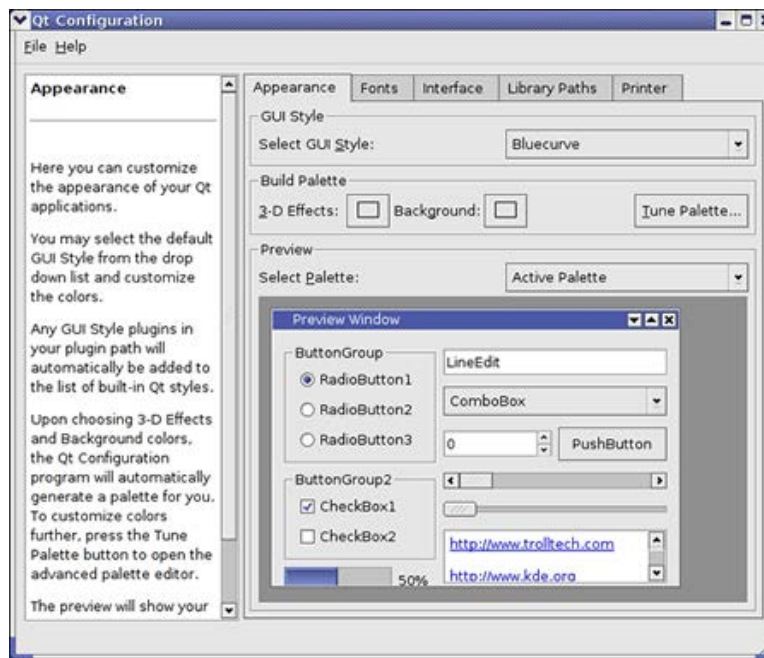
Configuring Font Size in the GUI

If you want to customize certain aspects of the Fusion GUI (such as menus, button labels, tables, lists, and so on), you can use a graphical configuration tool called Qt Configuration to customize the font size and other GUI features.

To customize the appearance of the Fusion GUI:

1. On the command line, enter `qtconfig`.

The Qt Configuration dialog appears.



2. Click the tabs at the top of the right side of the window to view the available options, and use the help text on the left side for help with each tab.

Note: The Qt Configuration tool is not a Google product. For more information about Qt Configuration, select either of the options on the Help menu.

Configuring Fonts For the Text Style Dialog

Fusion provides the *Sans regular* font for map vector labels. If you want to let users create labels in additional fonts, you can define a "font list" file so that the Text Style dialog displays a list of the fonts you support. For example, you can display international fonts such as Chinese, Japanese, or Hebrew character fonts. (For more information about the Text Style dialog, see the *Google Earth Enterprise Fusion Reference Guide*.)

To create a font list file:

1. Create an ASCII text file and save it as `/opt/google/share/fonts/fontlist`.
2. For each font you want to support, enter one line in the file that contains the four following pieces of information:
 - **Font name** - the name of the font, which appears in the drop-down list of fonts available in the Text Style dialog (for example, `TimesRoman`). No spaces are allowed in the font face name. Multiple variations of the font, such as regular, bold, italic, are automatically grouped under the same name in the drop-down list.
 - **File Path** - the full path to the TrueType (`.ttf`) font file. No spaces are allowed in this path.
 - **Bold** - 1 if the font is bold; 0 if not.
 - **Italic** - 1 if the font is italic (or oblique); 0 if not.
3. Save the file.

For example, your font list file might contain:

```
LucidaBrightDemi /usr/local/lib/fonts/LucidaBrightDemiBold.ttf 1 0
LucidaBrightDemi /usr/local/lib/fonts/LucidaBrightDemiItalic.ttf 0 1
LucidaBright /usr/local/lib/fonts/LucidaBrightItalic.ttf 0 1
LucidaBright /usr/local/lib/fonts/LucidaBrightRegular.ttf 0 0
LucidaSansDemi /usr/local/lib/fonts/LucidaSansDemiBold.ttf 1 0
LucidaSans /usr/local/lib/fonts/LucidaSansRegular.ttf 0 0
LucidaTypewriter /usr/local/lib/fonts/LucidaTypewriterBold.ttf 1 0
LucidaTypewriter /usr/local/lib/fonts/LucidaTypewriterRegular.ttf 0 0
LucidaTypewriter /usr/local/lib/oblique-fonts/LucidaTypewriterBoldOblique.ttf 1 1
LucidaSansDemi /usr/local/lib/oblique-fonts/LucidaSansDemiOblique.ttf 0 1
LucidaSans /usr/local/lib/oblique-fonts/LucidaSansOblique.ttf 0 1
LucidaTypewriter /usr/local/lib/oblique-fonts/LucidaTypewriterOblique.ttf 0 1
```

Customizing Your Google Maps Landing Page

Google used the Google Maps API to create a sample web application that provides a very basic way to display your Google Maps database output in a browser. The sample web application and the Google Maps API are automatically installed with Fusion.

Google recommends that you use the sample application to display your Google Maps data at first. After you see how it looks, you can create your own Google Maps web application that looks more your other web applications, using the provided sample web application as a guide. Documentation for the Google Maps API is at

<http://www.google.com/apis/maps/>.

To get started, make a copy of the sample application files (`maps_local.html`, `maps_google.html`, `fusionmaps.js`, and `fusionmaps.css`). Then configure a virtual server on which you can experiment, and move the copied files to that virtual server. (See [Configuring Virtual Servers](#) for help, if you need it.) When you create the Apache configuration file for the new virtual server, change `"maps/maps_google.html"` or `"maps/maps_local.html"` in the following line to point to your copy of the example files on the new virtual server:

```
RewriteRule ^/default_map/+$ /maps/maps_google.html [PT]
```

or

```
RewriteRule ^/default_map/+$ /maps/maps_local.html [PT]
```

You can edit the rest of the sample application in whatever ways you like, adding your own logo, branding, and so on.

Configuring the Google Maps API License Key

Google Maps supports only specific browser/operating system combinations. Even if you are using a supported browser, there are some features in Fusion that are not supported by some browsers on certain operating systems. (See **Google Maps Browser Support and Incompatibilities** in the *Google Earth Enterprise Fusion Reference Guide*.)

As long as you are connected to the Internet and have a license key for the Google Maps API, there is no problem (regardless of your platform), since your server contacts Google's servers for functions that are not supported in the browser.

Note: The following procedure assumes that you have received an email from Google that contains your license key for the Google Maps API.

To configure your Google Maps API license key:

1. Open `/opt/google/gehttpd/htdocs/maps/maps_google.html`.
2. Locate the following line in the script:

```
<script src="http://maps.google.com/maps?file=api&v=2&key=abcdefg" type="text/javascript"></script>
```

3. Replace the `key` placeholder (`'abcdefg'`) with your Google Maps API license key. Your key is contained in an email sent from Google.
4. Save the `maps_google.html` file.
5. Restart the Google Earth Enterprise Server (as root):

```
sudo /etc/init.d/geserver restart
```

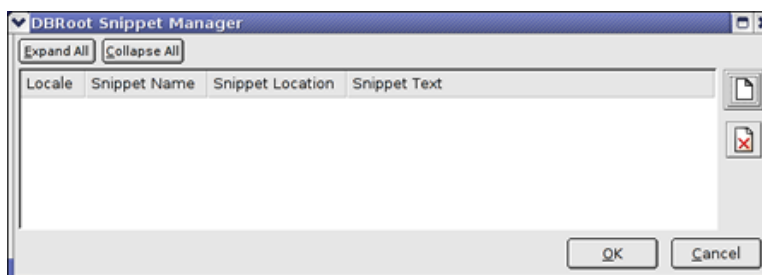
Using the dbRoot Snippet Manager


The dbRoot is a file that contains information that Fusion passes to Google Earth EC when you publish a database. The dbRoot Snippet Manager allows you to add your preferences to the dbRoot. Many of these preferences (or *snippets*) apply to display characteristics, such as showing or hiding the Google logo in Google Earth EC. The snippets you specify in the dbRoot Snippet Manager apply to all databases you publish from Fusion.

Caution: If you are working with multiple Fusion users on multiple workstations, it is important to remember that all managers on the Tools menu can be accessed by all users at the same time. If multiple users are working with the same manager at the same time, when one user closes the manager, that user's changes overwrite all previous data for that manager. So if you are working in a multi-user environment, be sure to coordinate with the other users to be sure that only one user has this manager open at a time.

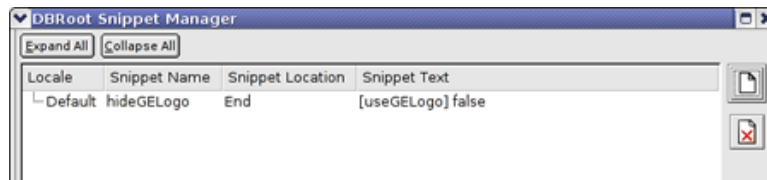
To add a snippet:

1. Select **Tools > dbRoot Snippet Manager**. The dbRoot Snippet Manager appears.



2. Click the page icon . The **Add New Snippet** dialog appears.
3. Enter a name for the new snippet, and click **OK**. The snippet appears on the list in the dbRoot Snippet Manager.

- Click in the **Snippet Text** column to enter the appropriate snippet text. The next section contains a list of allowable snippets. Snippet text syntax is as follows: [snippetName] value



- If you want to specify different snippet text for each supported locale, right-click the name of the snippet, and select a locale.
- When finished, click **OK**.

dbRoot Snippet Syntax

The following table lists the syntax for all of the possible dbRoot snippets.

Snippet Name	Purpose	Syntax - Default Value	Notes
userGuideURL	Help menu > User Guide	../userguide/v4/userGuideURL	Defaults to local PDF file for Google Earth EC. Can be set to an internal IP or hostname address.
supportCenterURL	Help menu > Support Center	../userguide/v4/support	Hidden in Google Earth EC if dbRoot setting does not exist. Can be set to an internal IP or hostname address.
tutorialGuideURL	Help menu > Tutorial	../userguide/v4/tutorials/index.htm	Can be set to an internal IP or hostname address.
keyboardShortcutURL	Help menu > Keyboard shortcuts link	../userguide/v4/ug_keyboard.html	Can be set to an internal IP or hostname address.
defaultWebPageURL	Tools menu > Web browser home page	http://www.google.com	Can be set to an internal IP or hostname address.
useGELogo	Shows/hides Google Earth logo in lower right corner of display	boolean	Default is true.
bbsServer.host	Host name	string	bbs.keyhole.com
bbsServer.port	Port	int	80
bbsServer.fileSubmitPath	URL to post KMZ content to	string	/ubb/postbouncer.php
bbsServer.postWizardPath	URL where posting wizard is located	string	/ubb/postcatcher.php
bbsServer.timeout	Timeout in seconds	double	15.0
bbsServer.retries	Number of retries before failing	int	2
bbsServer.secureSS	true = use HTTPS connection to	boolean	Default is false.

	hostfalse = use HTTP connection		
cobrandInfo	Custom co- branded logo	[export.cobrandInfo] { <etCobrandInfo> [top-center] { " http://your_server.com/your_logo.png " 0.5 0.95 true true top-center 0.25 } }	
earthIntlURL	Localize international URL for Google Earth	http://earth.google.com/intl/%1/	
supportAnswerIntlURL	Localize international URL for support answers	http://earth.google.com/support/ bin/answer.py?answer=%4&hl=%1	
supportTopicIntlURL	Localize international URL for support topics	http://earth.google.com/support/ bin/topic.py?topic=%4&hl=%1	
supportRequestIntlURL	Localize international URL for support requests	http://earth.google.com/support/ bin/request.py?hl=%1	
startupTipsURL	Localize international URL for startup tips	http://earth.google.com/ intl/%1/tips/v4/	
userGuideIntlURL	Localize international URL for documentation	http://earth.google.com/intl/%1/ userguide/v4/	
supportCenterIntlURL	Localize international URL for the support center	http://earth.google.com/support/?hl=%1	
businessListingIntlURL	Localize international URL for business listings	http://www.google.com/local/ add/login?hl=%3&gl=%2	
defaultWebPageIntlURL	Localize international URL for the default web page	http://www.google.com (English) http://www.google.com/intl/pt-BR (Portuguese) http://www.google.com/intl/%1 (all other languages)	
numStartupTips	Set the number of startup tips	13	
hideUserData	true = Suppress user name and license key information in the Help -> About windowfalse = Display user name and license key	boolean	Default is false.

	information		
export.options. disableDiskCache	true = Disable disk caching, so databases can ask Google Earth EC not to use disk cache at all.false = Enable disk caching.	boolean	Default is false.
reverseGeocodingServer...	Defines the reverse geocoding server.	<p>The default uses Google's geocoding server:</p> <pre>[googleMFReverseGeocoder] http://maps.google.com/ maps/api/earth/ GeocodeService.Search [reverseGeocodingServerVersion] 3 [reverseGeocodingServer.host] geo.keyhole.com [reverseGeocodingServer.path] /maps</pre> <p>To use your own geocoding server:</p> <pre>[reverseGeocodingServerVersion] 2 [reverseGeocodingServer.host] your_geocoding_server [reverseGeocodingServer.path] /path_for_queries</pre>	

Note: Advanced users of previous versions of Fusion were able to hand edit the dbRoot preamble. That is no longer possible. Most of the hand edits that users made in the past are now features built in to Fusion (such as the Search Tab Manager), and other changes are possible using the dbRoot Snippet Manager. However, there are some edits that advanced users made in dbRoot in previous versions of the software that are no longer allowed at all, including:

- Manually editing streamed layers combined for hybrid service.
- Appending a KML file to dbRoot.
- Manually adding a nested layer.

If you have any questions about these functions, please contact Google Earth Enterprise Fusion technical support.

To modify a snippet:


1. If you want to change the location of the snippet in dbRoot, click in the Snippet Location field for the snippet you want to move.

This field toggles between the values **End** and **Beginning** to move the snippet to the end or beginning of the dbRoot. This setting is useful when you want to change the order in which the layers appear in Google Earth EC by reordering them in dbRoot.

2. If you want to change the snippet text, click in the Snippet Text field for the snippet you want to modify, and edit the text.

Note: If you modify a dbRoot snippet, it does not force you to rebuild project affected by that change. If that dbRoot snippet affects a project that has already been built, when you publish a database that includes that project, the previous dbRoot snippet remains in effect. If something else in that project subsequently triggers a build, the modified dbRoot snippet will take effect in the resulting version.

To delete a snippet:

1. Select the snippet you want to delete, and click .

A message prompts you to confirm that you want to delete the selected snippet.

2. Click **OK**.

The snippet disappears from the dbRoot Snippet Manager.

Monitoring System Processes

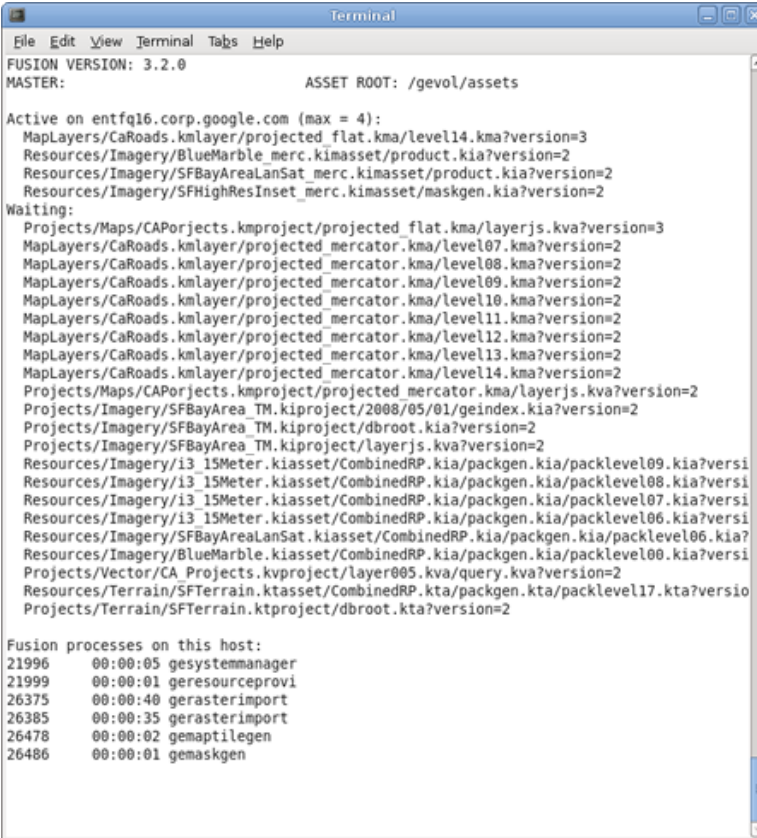
You can watch all active Fusion system tasks and processes using the `getop` command. This command is similar to the Unix command `top`, but is specific to Fusion processing. It is the command line equivalent to the System Manager in the

Fusion GUI. See [getop](#) in the **Command Line Reference** chapter for more information.

The `getop` command outputs the following data for the workstation to the console:

- Version of Fusion running
- Host name of the master
- Currently selected asset root
- The active process and maximum number of jobs allowed (max = *n*)
- List of processes waiting to run
- Process IDs, running time, and names of all Fusion processes running

For example, the output might look something like this:



```
Terminal
File Edit View Terminal Tabs Help
FUSION VERSION: 3.2.0
MASTER:                               ASSET ROOT: /gevol/assets

Active on entfq16.corp.google.com (max = 4):
MapLayers/CaRoads.kmlayer/projected_flat.kma/level14.kma?version=3
Resources/Imagery/BlueMarble_merc.kimasset/product.kia?version=2
Resources/Imagery/SFBayAreaLanSat_merc.kimasset/product.kia?version=2
Resources/Imagery/SFHighResInset_merc.kimasset/maskgen.kia?version=2
Waiting:
Projects/Maps/CAPorjects.kmproject/projected_flat.kma/layerjs.kva?version=3
MapLayers/CaRoads.kmlayer/projected_mercator.kma/level07.kma?version=2
MapLayers/CaRoads.kmlayer/projected_mercator.kma/level08.kma?version=2
MapLayers/CaRoads.kmlayer/projected_mercator.kma/level09.kma?version=2
MapLayers/CaRoads.kmlayer/projected_mercator.kma/level10.kma?version=2
MapLayers/CaRoads.kmlayer/projected_mercator.kma/level11.kma?version=2
MapLayers/CaRoads.kmlayer/projected_mercator.kma/level12.kma?version=2
MapLayers/CaRoads.kmlayer/projected_mercator.kma/level13.kma?version=2
MapLayers/CaRoads.kmlayer/projected_mercator.kma/level14.kma?version=2
Projects/Maps/CAPorjects.kmproject/projected_mercator.kma/layerjs.kva?version=2
Projects/Imagery/SFBayArea_TM.kiproject/2008/05/01/geindex.kia?version=2
Projects/Imagery/SFBayArea_TM.kiproject/dbroot.kia?version=2
Projects/Imagery/SFBayArea_TM.kiproject/layerjs.kva?version=2
Resources/Imagery/i3_15Meter.kiasset/CombinedRP.kia/packgen.kia/packlevel09.kia?versi
Resources/Imagery/i3_15Meter.kiasset/CombinedRP.kia/packgen.kia/packlevel08.kia?versi
Resources/Imagery/i3_15Meter.kiasset/CombinedRP.kia/packgen.kia/packlevel07.kia?versi
Resources/Imagery/i3_15Meter.kiasset/CombinedRP.kia/packgen.kia/packlevel06.kia?versi
Resources/Imagery/SFBayAreaLanSat.kiasset/CombinedRP.kia/packgen.kia/packlevel06.kia?
Resources/Imagery/BlueMarble.kiasset/CombinedRP.kia/packgen.kia/packlevel00.kia?versi
Projects/Vector/CA_Projects.kvproject/layer005.kva/query.kva?version=2
Resources/Terrain/SFTerrain.ktasset/CombinedRP.kta/packgen.kta/packlevel17.kta?versio
Projects/Terrain/SFTerrain.ktproject/dbroot.kta?version=2

Fusion processes on this host:
21996  00:00:05 gesystemmanager
21999  00:00:01 geresourceprovi
26375  00:00:40 gerasterimport
26385  00:00:35 gerasterimport
26478  00:00:02 gemaptiligen
26486  00:00:01 gemaskgen
```

Backing Up Your Data

Generally, the important Google Earth Enterprise Server data is all within the `/var` directory. If you perform regular back-ups of `/var`, that covers most of the server data. Google recommends that you also back up the following directories:

```
/etc
/opt/google/gehttpd/conf
/opt/google/gehttpd/conf.d
/opt/google/getomcat/conf
/opt/google/gehttpd/htdocs
```

With regard to Fusion data, Google strongly recommends that you back up all of your original source data (vector and raster). In addition, Google encourages you to back up all of the `.xml` files within your asset root(s). In theory, if you have the source data and the asset root `.xml` files, everything else can be reconstructed. Google Earth Enterprise does not include any tools for doing such a reconstruction, but the data is there and it could be done, if necessary.

You do not need to back up your publish root(s), since there is nothing there that cannot be reproduced. If you have the space and want to back them up, however, recovery will be faster, but it is not necessary.

Many Google Earth Enterprise users also back up their product files. These are the low-level files that result from building resources (for example, `.kip`, `.ktp`, `.kmp`, `.kvp`). That way, if you do need to reconstruct, you can import the product files (rather than raw source files) to recreate the resources. This will save you some build time.

Running Fusion on a Machine with Multiple CPU Cores

Fusion can be configured to use a maximum number of CPU cores on a machine with the command:


```
/opt/google/bin/geselectassetroot --assetroot /my/assetroot --numcpus X
```

This number will show up as the `maxjobs` entry within `getop` output and corresponds to how many concurrent jobs Fusion may spawn at any one time. Note that the value for `--numcpus` should be equal to, or less than, the total number of CPUs permitted by your Fusion license; it should never exceed the number of physical CPU cores on the machine.

Each task within Fusion is configured to use 1 CPU by default. Changing the maximum number of CPUs using the above command will *not* affect the number of CPUs assigned to each individual task. Some tasks in Fusion are capable of multithreaded support including:

- `gepackgen` (imagery and terrain projects)
- `gemaptilegen` (2D vector-based map tiles)
- `gecombineterrain` (3D databases)

These tasks may be multithreaded by enabling 'Taskrules' within Fusion. Implementation of Taskrules is described in: `/opt/google/share/taskrules/README`.

The `PacketLevel.taskrule`, `MapLayerLevel.taskrule`, and `CombinedTerrain.taskrule` taskrules will enable multiple CPUs to work on each individual task. For example, configuring a `minNumCPUs` and `maxNumCPUs` of 2 for `gepackgen` means that up to 4 `gepackgen` processes may run on Fusion, with 2 CPUs assigned each, when 8 CPU cores are allocated.

As a best practice, `PacketLevel.taskrule` and `MapLayerLevel.taskrule` should be configured such that multiple `gepackgen` and `gemaptilegen` processes can run concurrently. Since `gecombineterrain` is CPU-intensive and can be an operational bottleneck, more CPUs should be assigned to that task to expedite processing.

So on a machine with 8 CPU cores available for Fusion processing:

1. Set `--numcpus=7`. It's recommended to set the maximum number of CPUs allocated to Fusion to (N-1), such that one CPU core is reserved for system operations
2. Set `minNumCPU=2` and `maxNumCPU=2` in `PacketLevel.taskrule` for imagery projects (3 concurrent `gepackgen` jobs possible)**
3. Set `minNumCPU=3` and `maxNumCPU=4` in `PacketLevel.taskrule` for terrain projects (2 concurrent `gepackgen` jobs possible)**
4. Set `minNumCPU=3` and `maxNumCPU=4` in `MapLayerLevel.taskrule` (2 concurrent `gemaptilegen` jobs possible)
5. Set `minNumCPU=7` and `maxNumCPU=7` in `CombinedTerrain.taskrule` (1 `gecombineterrain` job)

****Note:** Fusion will use up to 200% CPU processing for imagery projects per `gepackgen` process, and up to 600% CPU processing for terrain projects. There is fundamentally a balance between assigning sufficient numbers of CPUs to each individual process for `gepackgen` while still enabling multiple concurrent `gepackgen` processes for parallel processing.

All other tasks in Fusion will continue to operate with min/max 1 CPU.

[Back to top](#)

[4.4.1 Documentation](#)**Administration Guide**[Home](#)[Configuring Server](#)[Configuring Fusion](#)

Configuring Tutorial Work Space

[Command Reference](#)[Common Server Error Messages](#)[Multiple Database Support](#)[Legal Notices](#)**Configuring the Tutorial Work Space**

In order to accommodate users who are working through the tutorial at the same time other users (or even those same users) are working on production data, Google recommends that you configure a tutorial work space on each workstation where you install the tutorial data. That way, you can keep the tutorial source volume and asset root separate from your production source volume(s) and asset root.

In addition, for any user who plans to work through the tutorial lessons on search tabs and search layers, you must register the search plug-in.

This chapter describes how to configure the tutorial work space and register the search plug-in. It also provides instructions for users who need to switch back and forth between the tutorial asset root and the production asset root, and instructions for cleaning up the tutorial files when they are no longer needed.

Configuring the Tutorial Asset Root and Source Volume

When you install Google Earth Enterprise, you configure a source volume and asset root for your production data. If you accepted the default values, they are `/gevol/src` and `/gevol/assets`, respectively.

The installation script installs the tutorial files in `/opt/google/share/tutorials/fusion`, so you must add that path as the source volume for the tutorial. In addition, Google recommends that you add a tutorial asset root for users to store the data they create with Google Earth Enterprise Fusion while working through the tutorial.

The following procedure describes how to configure an asset root and the tutorial source volume for each tutorial user. You must configure them for *each* user on *each* workstation where you install the Google Earth Enterprise Fusion tutorial files.

To configure a tutorial asset root and source volume:

1. On the command line, log in as root.
2. Stop the system manager by entering:

```
/etc/init.d/gefusion stop
```

3. Enter:

```
geconfigureassetroot --new --assetroot /username/assets
```

where *username* is the name of the user.

The *username* does not have to be the user's official account name. It can be anything that distinguishes that user from other users on that particular workstation, such as `edaniels` or just `Emily`.

Note: Google recommends that you place the tutorial asset root on the same partition as the publish root, so the publisher uses hard links instead of making copies of the tutorial databases. (See *Planning the Location of Your Publish Root* for more information.)

If you place the tutorial asset root on the root partition, ensure that there is enough disk space for the data created by each user working through the tutorial. If a user completes all lessons in the tutorial, allow 1.5 GB of disk space.

The tool asks if you want to create a new source volume.

4. Enter `y`, and press **Enter**.

You are prompted to enter a directory for the source volume.

5. Enter `/opt/google/share/tutorials/fusion`, and press **Enter**.

The tool asks if you want to add more volumes.

6. If you want to create tutorial work spaces for more users, enter `y`, press **Enter**, and repeat steps 3 through 5. If not, skip this step. Otherwise, enter `n`, and press **Enter**.

The tool displays the message “Configured `/username/assets`” and returns you to the command line prompt.

7. Log out as root.

8. Start the system manager by entering:

```
/etc/init.d/gefusion start
```

Note: The new source volume and asset root are automatically selected for you. You do not have to run the `geselectassetroot` command. (See the next section, [Selecting the Tutorial Asset Root](#), for more information about selecting different asset roots.)

Selecting the Tutorial Asset Root

There are two occasions when users must select a different asset root:

- When multiple users share a single workstation, each user must select his or her own tutorial asset root.
- When a user switches from the tutorial data to real production data, he or she must select the appropriate asset root.

This section explains how to select a different asset root.

Caution: Note that even though your source volumes and asset roots are separate for each user or for the tutorial and production data, there is only one publish root on each virtual server for Earth databases and one for Map databases.

When two users are sharing a single workstation, they are both publishing to the same publish root. When one user publishes a database on that workstation, it overwrites any database that might have been published previously by another user on that same workstation. Likewise, if a user is switching back and forth between tutorial and production data on the same workstation, it is possible to overwrite a production database with a tutorial database and vice versa. Of course, the user can simply republish the desired database to make it available to Google Earth EC again.

To select the tutorial asset root:

1. On the command line, log in as root.
2. Stop the system manager by entering:

```
/etc/init.d/gefusion stop
```

3. Enter:

```
geselectassetroot --assetroot /username/assets
```

where *username* is the name you used in step 3 of [Configuring the Tutorial Asset Root and Source Volume](#).

4. Log out as root.
5. Start the system manager by entering:

```
/etc/init.d/gefusion start
```

Registering the Search Plug-In

When you install the Google Earth Enterprise software, it installs a sample plug-in (called

ExamplePlugin) in /opt/google/share/searchexample. Users are instructed to reference that plug-in in the **Defining Search Tabs** lesson in the tutorial.

You must register the plug-in before users can reference it. (See [Search Plug-in](#) in the **Command Reference** chapter for more information about the search plug-in commands for the geserveradmin tool.)

To register the sample plug-in for tutorial users, on the command line, enter:

```
geserveradmin --addplugin ExamplePlugin --jar_path \  
/opt/google/search/plugins/ExamplePlugin.jar \  
--class com.google.earth.search.plugin.ExamplePlugin
```

This command registers the plug-in and makes it available for users to reference on search tabs.

Cleaning Up the Tutorial Work Space

When a user completes the tutorial or no longer needs the tutorial data, you can clean up that tutorial work space, if desired, by removing the tutorial source files, asset root, and published databases. This section describes the best way to perform that clean-up.

Note: Google recommends that you keep the tutorial files intact, since they use very little space and can come in handy for users to practice, even after they have quite a bit of experience with Google Earth Enterprise Fusion.

To remove the tutorial source files, asset root, and databases:

1. At the command line prompt, log in as root.
2. Stop the system manager by entering:

```
/etc/init.d/gefusion stop
```

3. Select the production asset root by entering:

```
geselectassetroot --assetroot /gevol/assets
```

Substitute the appropriate asset root path, if necessary.

4. Stop the Google Earth Enterprise Server:

```
/etc/init.d/geserver stop
```

Note: If you are upgrading from version 2.4 to 3.x for the first time, the command to stop the server is /etc/init.d/khhttpd stop.

5. Insert the distribution DVD, and mount the DVD drive.
6. Run the installer from any directory.

Command-line installer:

```
/media/cdrom/InstallGEFusion.sh
```

GUI installer:

```
/media/cdrom/InstallGEFusionGUI.sh
```

Substitute the designation for your DVD drive for `cdrom`, if necessary.

The installation script starts running, and the first prompt asks what product(s) you want to install.

7. Enter **3** for the Google Earth Enterprise Fusion Tutorial.

The **X** next to Google Earth Enterprise Fusion Tutorial disappears. Removing the **X** causes the installation script to remove the tutorial files.

8. Enter **F** to finish selecting products.

9. Follow the rest of the prompts.

The installer removes the tutorial files but does not change any of your other installed products. When it completes the installation, it returns you to the prompt.

10. Delete the user's tutorial asset root by entering:

```
rm -Rf /username/assets
```

where *username* is the name of the user you specified when you configured the tutorial work space.

Caution: Make sure you are removing the tutorial asset root, not the production root. If you delete the production root, there is no way to recover it (other than from back-ups, if available).

11. Start the Google Earth Enterprise Fusion system manager and Google Earth Enterprise Server:

```
/etc/init.d/gefusion start  
/etc/init.d/geserver start
```

The order in which you start them does not matter.

12. List the databases on the current server by entering:

```
geserveradmin --listdbs
```

The tool displays a list of all databases ever published (except any deleted databases) on the server. If the server type is omitted, the server type defaults to *stream*.

13. Select the database you want to remove by entering:

```
geserveradmin --deletedb db_name...
```

where *db_name* is the name of the database you want to delete.

Note: If you want to delete a currently published database, you can either publish a different database to the same virtual server or disable the virtual server on which it is published. Then you can delete the database.

This tool does not delete the actual files. It is similar to putting files in the trash on a Windows or Macintosh desktop.

14. Permanently delete the selected databases by entering:

```
geserveradmin --garbagecollect
```

Command Reference

This appendix describes all of the command line tools for system administrations in alphabetical order. If you prefer, you can find each tool’s syntax by entering the name of the tool with the --help option, for example:

```
geconfigureassetroot --help
```

This appendix uses the following typographic conventions:

<i>Italic</i>	Information that the user must supply
Bold	Text that the user must type exactly as shown
Ellipsis ...	Argument that can be repeated several times in a command
Square brackets []	Optional commands or arguments
Curly braces { } with options separated by pipes ; for example: {even odd}	Lists a set of choices from which the user can select only one
Parentheses ()	Grouped items that function together
Courier font	Code or program output

geconfigureassetroot

```
geconfigureassetroot {--new --assetroot path | --repair | --editvolumes | --fixmasterhost | --noprompt} [--srcvol path]
```

Purpose

To add volume definitions or edit existing volume definitions.

Note: You must stop the system manager before using this command and then start it again after you are done. You must also run this command as root.

Example

```
geconfigureassetroot --new --assetroot /gevol/assets
geconfigureassetroot --new --assetroot /gevol/assets --srcvol /data1/src
geconfigureassetroot --repair
geconfigureassetroot --editvolumes
```

Commands

```
--new --assetroot path
```

Optional. Specify the path to the new asset root.

Note: If you omit all options, the system creates /gevol/assets.

```
--repair
```

Optional. Repairs various inconsistencies in the asset root (such as permissions, ownership, missing ID files, and so on). When you run this command, the tool auto-detects the problems that need to be repaired and fixes them.

Note: Do not use this command unless you see a system message instructing you to do so.

```
--editvolumes
```

Optional. Follow the prompts to add a volume to the selected asset root or, modify the `localpath` definition for an existing volume, or to add a volume definition.

```
--fixmasterhost
```

Optional. Change the `assetroot host` entry to match the current host name. (This command corrects cases where a host name is changed after installing and configuring Fusion.)

```
--noprompt
```

Optional. Perform the command without prompting the user for any input. This option requires that some commands have arguments specified on the command line.

```
--srcvol path
```

Optional. Specify the path to the source volume.

geconfigurepublishroot

```
geconfigurepublishroot [--path=path] [--allow_symlinks] [--noprompt]
```

Purpose

To specify the path where you want to publish databases for the current GEE Server. Follow the prompts.

Note: You must run this command as root.

Example

```
geconfigurepublishroot --path /gevol/published_dbs --allow_symlinks
```

Commands

```
--path=path
```

Optional. The path to the publish root. Default value is `/gevol/published_dbs`.

```
--allow_symlinks
```

Optional. Configures the publisher to accept symbolic links. Useful when the publish root is on a separate logical volume from the asset root. Default is no.

```
--noprompt
```

Optional. Perform the command without prompting the user for any input. This option requires that some commands have arguments specified on the command line. If the arguments are insufficient or the configuration fails, the program will return -1 (0 is returned on success).

Caution: Do not create more than one publish root for a single asset root. That configuration produces unpredictable or undesirable results, including the inability to publish at all from that asset root.

gedisconnectedclean

```
gedisconnectedclean [--dbpath dbpath] [--list assetroot]
```

Purpose

To clean a disconnected database from a disconnected mock asset root.

Example

```
gedisconnectedclean --dbpath /gevol/assets/Databases/MyPOIs.kdatabase
```

Commands

```
--dbpath dbpath
```

Required. Specify the database path to clean. This must be a low-level path to a database directory (one of the entries in the `assetroot/dbpaths.list` file). See `--list` command option to find databases stored within the mock asset root.

```
--list assetroot
```

Optional. List all dbpaths currently in disconnected asset root

gedisconnectedpublish

```
gedisconnectedpublish [db_alias] db_name
```

Purpose

To publish a database on a disconnected server.

Example

```
gedisconnectedpublish MyPOIs
```

Commands

```
db_alias
```

Optional. Since `db_name` is the “low-level” name of the database, `db_alias` allows you to enter a name that is easier to remember, for example, `Databases/SF Highways.kdatabase?ver=1`.

```
db_name
```

Required. The full, “low-level” name of the database you want to publish.

gedisconnectedreceive

Deprecated in version 4.0

`gedisconnectedreceive` is required only when the disconnected database was sent with an older (pre 4.0) version of Fusion.

```
gedisconnectedreceive --input dirname
```

Purpose

To copy a disconnected database from either detachable media or local storage into the mock asset root.

Example

For detachable media:

```
gedisconnectedreceive --input /mnt/usbdrive/SFHighways_3dDatabase_v20
```

For local storage:

```
gedisconnectedreceive --input  
/gevol/src/disconnected_databases/SFHighways_3dDatabase_v20
```

Commands

```
--input dirname
```

Required. Specify the directory that contains the files to be copied. This is typically the mount point of a hard drive.

Notes:

The `gedisconnectedreceive` command will create an asset tree that mirrors the asset tree of the Fusion system that built the database.

The `gedisconnectedreceive` command will copy data to the mock asset root if the input folder is on a separate volume than the mock asset root. Links to the input folder to the mock asset root will be created if both the input and mock asset root folders on the same volume.

gedisconnectedsend

```
gedisconnectedsend [--extra filename] [--havepath dbpath] [--havepathfile file]  
--output dirname [--sendpath dbpath] [--sendver dbver]
```

Purpose

To gather all the files from a Fusion asset root necessary for a disconnected publish, for either publishing new databases or publishing "delta" updates.

Example

```
gedisconnectedsend --sendver Databases/SFHighways.kdatabase?version=2  
--output /gevol/src/disconnected_databases/SFHighways_3dDatabase_v2
```

Commands

```
--extra filename
```

Optional. Specify an extra file to package. This is typically used to repair broken files.

```
--havepath dbpath
```

Optional. Specify which database path already exists on the target server. This must be a low-level path to a database directory and may be specified more than once.

```
--havepathfile file
```

Optional. Specify the file that contains the list of existing database paths (copy of

assetroot/dbpaths.list from the remote server).

```
--output dirname
```

Required. Specify where to gather the files. The directory must already exist and be empty. This is typically the mount point of a hard drive.

```
--sendpath dbpath
```

Optional. Specify which database path to send. This must be a low-level path to a database directory. You can determine this path by entering `gequery --outfiles dbver` on the source server.

```
--sendver dbver
```

Optional. Specify which database version to send. Use the `?version=...` syntax. Available database versions may be found with the `gequery --versions` command.

Note: The `gedisconnectedpublish` command may be used for publishing new databases to a remote server, or for "delta" publishes (which include only changes between versions) to a remote server.

geindexcheck

```
geindexcheck [--mode= {index | all}] [--verbose= n] [--progress n]
              [--help] { --index=index_path | --database=database_name }
```

Purpose

To diagnose problems with Fusion databases. You can run `geindexcheck` on either the unified index or intermediate indexes. The most common errors `geindexcheck` detects are CRC errors and missing files. For missing files, it reports the full file path. For CRC errors, it reports the full file path and the position within the file where the error occurred. After reporting an error, `geindexcheck` stops searching. This is because searching a corrupt index usually generates large numbers of redundant or irrelevant error messages caused by the initial error.

Example

```
geindexcheck --mode all --database
/gevol/published_dbs/stream_space/your_server/gevol/assets/Databases/igain.kdatabase/
gedb.kda/ver001/geadb
```

Commands

```
--mode= {index | all}
```

Optional. Search the index only (default) or specify "all" to search both the index and packet files (in index order).

```
--verbose= n
```

Optional. Specify the level of error reporting from 0 (fatal error) to 7 (all messages). The default is 3. This overrides `KH_NFY_LEVEL`.

```
--progress n
```

Optional. Request progress messages every `n` index locations. The default is 100000. To disable, set to 0.

Note: If enabled, this requires a verbose level of 3 or greater.

```
--help
```

Optional. Print the error message.

```
--index=index_path | --database=database_name
```

Required. Specify either the index or the database parameter, but not both. For index, specify the path to the index to search. The index can be of any type. For database, specify the path from the asset root to the database. The unified index for the database is searched. Defaults to the latest version. To override the default, append "?version=n", where "n" is the version number.

gepublishdatabase

Deprecated in version 4.0

You can now publish databases using either the geserveradmin tool or the Fusion GUI.

```
gepublishdatabase [--delete db_name [--serverurl url]][--listdbs [--serverurl url]][--publish dbname [--server nickname]][--publisheddbs [--serverurl url]]
```

Purpose

To publish a database.

Examples

```
gepublishdatabase --publish msDelta
gepublishdatabase --publish msDelta --server mainServer
gepublishdatabase --listdbs --serverurl http://private.company.com
```

Commands

```
--delete db_name
```

Delete a registered database from the server.

```
--serverurl url
```

Optional. Use a specific server URL, such as http://private.company.com.

```
--listdbs
```

List all databases registered on the server.

```
--serverurl url
```

Optional. Use a specific server URL, such as http://private.company.com.

```
--publish dbname
```

Specify the location and name of the database that you want to publish relative to the asset root, for example, Databases/databaseA.

```
--server nickname
```

Optional. Specify the nickname of the server association to which you want to publish. If you omit the server specification, Fusion publishes to the default server. (Consult your

administrator or refer to the *Google Earth Enterprise Administration Guide* for information about setting up server associations.)

```
--publisheddbs
```

List all of the published databases on the server.

```
--serverurl url
```

Optional. Use a specific server URL, such as `http://private.company.com`.

geselectassetroot

```
geselectassetroot [--lock] [--noprompt] [--unlock]  
  ( [--assetroot path [--role {master | slave}] [--numcpus num]] )
```

Purpose

To perform a variety of operations related to existing asset roots on the current machine.

Note: You must stop the system manager before using this command and then start it again after you are done. You must also run this command as root.

Example

```
geselectassetroot --lock  
geselectassetroot --unlock  
geselectassetroot --assetroot /gevol/assets  
geselectassetroot --assetroot /gevol/assets --role slave --numcpus 3
```

Commands

```
--lock
```

Optional. Disables the ability to select a different asset root on this machine.

```
--noprompt
```

Optional. Perform the command without prompting the user for any input. This option requires that some commands have arguments specified on the command line.

```
--unlock
```

Optional. Enables the ability to select a different asset root on this machine. (Use only if `--lock` is enabled.)

```
--assetroot path
```

Optional. Specify the path to the asset root for this machine.

```
--role {master | slave}
```

Optional. Specify this machine's role in the asset root (master or slave). The default role is master. This command is available only in combination with `--assetroot`.

```
--numcpus num
```

Optional. Specify the number of CPUs on this machine to use for processing. The default will be the maximum number of CPUs detected on the machine during installation. This command is available only in combination with `--assetroot`.

geselectpublishroot

```
geselectpublishroot path [--noprompt]
```

Purpose

To specify a different publish root. The specified path must exist. If you want to create a publish root, see [geconfigurepublishroot](#).

Example

```
geselectpublishroot /gevol/published_dbs
```

Arguments

path

Required. Specify the path to the desired publish root.

Options

```
--noprompt
```

Optional. Perform the command without prompting the user for any input. This option requires that some commands have arguments specified on the command line.

geserveradmin

```
geserveradmin [options] commands
```

Purpose

To configure your GEE Server. This section breaks down the `geserveradmin` commands into the following categories:

- Database
- Virtual server
- Search plug-in
- Admin
- Miscellaneous

All of the commands of each type are described below. At least one command is required.

Examples

```
geserveradmin --listdbs
geserveradmin --listplugins
geserveradmin --server_type map --dbdetails
/gevol/assets/Databases/USCounties.kdatabase/gedb.kda/ver001/gedb
geserveradmin --server_type map --dbdetails "/gevol/assets/Databases/SF
Neighborhoods.kdatabase/gedb.kda/ver001/gedb"
geserveradmin --addvs private --vstype ge
geserveradmin --disablevs private
geserveradmin --addplugin ExamplePlugin --jar_path /your_path/ExamplePlugin.jar--
class com.google.earth.search.plugin.ExamplePlugin
```

Options

```
--stream_server_url url
```

Optional. Specify a stream server other than the default. The default is the current server.

```
--search_server_url url
```

Optional. Specify a search server other than the default. The default is the current server.

```
--server_type {stream | search}
```

Optional. Specify whether the servers in question are stream or search servers. The default is stream. If you want to reference a search server, this option is required with the `listdbs`, `dbdetails`, `listvss`, `vssdetails`, `addvs`, `deletevs`, `disablevs`, and `garbagecollect` commands.

Commands

Database

Each of the database commands is listed below, along with its syntax and description. If the name of the database contains one or more spaces, double quote the entire path. (See the examples above.)

```
--adddb db_name [--dbalias alias]
```

Register a new database with the specified name.

```
--dbalias alias
```

Optional. Specifies a user-friendly name for the database.

```
--dbdetails db_name
```

Provides a list of all of the files required by the specified database. If omitted, the server type defaults to `stream`.

```
--deletedb db_name
```

Deletes the specified database entry from the server. Does not delete the actual files. (This command is similar to putting files in the trash on a Windows or Macintosh desktop. See also `--garbagecollect`.)

Note: If you want to delete a currently published database, you can either publish a different database to the same virtual server or disable the virtual server on which it is published. Then you can delete the database. (See also `--disablevs`.)

```
--listdbs
```

Lists all databases ever published (except any deleted databases) on the server. If the server type is omitted, the server type defaults to `stream`.

```
--listplugins
```

Lists all plug-ins used on the server.

```
--publishdb db_name [--push] [--streamvs vs_name] [--searchvs vs_name]
```

Publish one or more databases on the specified virtual server. If the virtual server name is omitted, it publishes to the default server.

```
--push
```

Optional. Push the specified database to the virtual server.

```
--streamvs vs_name
```

Optional. Specify the name of the stream server.

```
--searchvs vs_name
```

Optional. Specify the name of the search server.

```
--publishedbbs
```

Lists the databases currently published on the server. If the server type is omitted, the server type defaults to `stream`.

```
--pushdb db_name...
```

Push one or more databases to the server.

Virtual Server

Each of the virtual server commands is listed below, along with its syntax and description.

Caution: Publishing to virtual servers other than the default server is supported only in version 4.2 or later of Google EC. If you are using version 4.0 or earlier, only databases that you publish to the default server can be accessed by Google EC.

```
--addvs vs_name [--vstype {ge | map}] --vsurl url [--vscachelevel num]
```

Registers a new virtual server with the specified name. Spaces are not allowed in the virtual server name. If the server type is omitted, the server type defaults to `stream`.

```
--vstype {ge | map}
```

Optional for stream servers. Specify the type of databases to be published to the new virtual server. If omitted, the virtual server type defaults to `ge`.

```
--vsurl url
```

Required. The `vsurl` specifies the location of the virtual server. It must match the corresponding server-side virtual server configuration. There are three ways to specify the `vsurl`:

- **Location-based URL**, such as `/private_ge`. For example, if the entire URL is `http://www.company.com/private_ge`, you enter `/private_ge`.

Note: Google recommends that you use the `_ge` and `_map` naming convention to make it easier to distinguish between virtual server types.

- **Port-based URL**, such as:

```
http://www.company.com:1234
```

The entire URL, including protocol, server name, path (if applicable), and port are required.

- **Name-based URL**, such as:

```
http://corp.company.com
```

For this type of specification, you must modify your DNS appropriately for the virtual server.

After you use this command, you must create a configuration file for the new virtual server. (See [Configuring Virtual Servers](#) for details.)

```
--vscachelevel num
```

Optional. Specify a cache level (1, 2, or 3) for the virtual server. The default is 2.

This cache is different than the client cache. This option caches only the index nodes at display levels 4, 8, and 12 (not data packets). If you increase this setting, Fusion caches more of the index in RAM, thereby decreasing server latency at the cost of server RAM. Level 3 uses approximately 1 GB of RAM. Level 2 uses approximately 4 MB of RAM. Level 1 uses approximately 16 KB of RAM. Each additional cache level consumes 256 times the RAM as the previous level and saves one disk read per packet served.

The server makes no checks that the RAM needed for caching does not exceed the total RAM on the machine. For example, if you have three virtual servers set to cache at level 3 on a machine that has only 2 GB of RAM, the machine will thrash memory. The default is Level 2, so you should be able to create as many virtual servers as you want at the default cache level without worrying about running out of RAM.

Typically, users increase only a small number of virtual servers to cache level 3 on production servers and leave the rest of them at level 2. On servers that share a machine with Fusion, do not increase the level to 3. Google Earth Enterprise Fusion needs more RAM than the server does.

```
--deletevs vs_name
```

Permanently delete the specified virtual server. If the server type is omitted, the server type defaults to `stream`.

```
--disablevs vs_name
```

Temporarily disable the specified virtual server, primarily to prevent Google EC users from connecting to that server temporarily. If the server type is omitted, the server type defaults to `stream`.

Note: To re-enable a disabled virtual server, simply publish a database to it.

```
--listvss
```

Provides a list of all virtual servers configured for the current machine. If the server type is omitted, the server type defaults to `stream`.

```
--vsdetails vs_name
```

Displays the name and URL of the specified virtual server. If the server type is omitted, the server type defaults to `stream`.

Search Plug-in

Each of the search plug-in commands is listed below, along with its syntax and description.

Note: These commands are available on search servers only.

```
--addplugin plugin_name --jar_path path --class class_name
```

Registers a new search plug-in.

```
--jar_path path
```

Required. Specify the path to the `.jar` file for the plug-in.

```
--class class_name
```

Required. Specify a fully qualified class name for the plug-in.

```
--deleteplugin plugin_name
```

Deletes the specified search plug-in.

```
--listplugins
```

Lists all registered plugins.

Admin

Each of the admin commands is listed below, along with its syntax and description.

```
--garbagecollect
```

Permanently deletes the files for databases that have been selected for deletion. Generally, you run this command nightly to remove the files for databases that users have deleted to free up space on the storage device. (This command is similar to emptying the trash on a Windows or Macintosh desktop. See also `--deletedb`.)

Note: Deletes only those files that are not used by other databases on that server.

Miscellaneous

Each of the miscellaneous commands is listed below, along with its syntax and description.

```
--enable_cutter
```

Enables cutting of portable globes. For more information about portable globes, please refer to the Portable Globe and Server guide.

```
--disable_cutter
```

Disables cutting of portable globes.

getop

```
getop --delay seconds
```

Purpose

To display a list of what Fusion is currently working on and whether `gesystemmanager` and `geresourceprovider` are currently running.

Enter Ctrl-C to exit and return to the prompt.

Example

```
getop --delay 30
```

Commands

```
--delay seconds
```

Optional. Specify the number of seconds delay between refreshes. For example, if you specify 30, `getop` runs every 30 seconds. If you do not specify the delay, it does not refresh.

geupgradeassetroot


```
geupgradeassetroot --assetroot path [--noprompt]
```

Purpose

To upgrade an existing asset root after installing a later version of the software.

Note: You must stop the system manager before using this command and then start it again after you are done. You must also run this command as root.

Example

```
geupgradeassetroot --assetroot /data1/assets
```

Commands

```
--assetroot path
```

Required. Specify the path to the asset root. If omitted, the asset root defaults to /gevol/assets.

```
--noprompt
```

Optional. Perform the upgrade without prompting the user for any input. This option requires that some commands have arguments specified on the command line.

[Back to top](#)

[4.4.1 Documentation](#)**Administration Guide**[Home](#)[Configuring Server](#)[Configuring Fusion](#)[Configuring Tutorial Work Space](#)[Command Reference](#)

Common Server Error Messages

[Multiple Database Support](#)[Legal Notices](#)**Common Server Error Messages**

Server error messages appear mainly when:

- The server (or part of it) is not running.
- The network connection to the server is down.

To determine whether the server is running, enter the server's URL or IP address in a browser window. If it is running, the browser displays the Google Earth logo. If it is not running, the browser displays a Page Not Found error message. To start the server, if necessary, enter:

```
/etc/init.d/geserver start
```

The remainder of this appendix provides additional information about the most common error messages you might see in Google Earth Enterprise Server, including:

- [Authentication required](#)
- [Database is currently published. Please disable the stream virtual server first.](#)
- [Either the virtual server vs_name does not exist or is already disabled.](#)
- [Incorrect cache level](#)
- [Invalid database](#)
- [Plugin already exists.](#)
- [Plugin plug-in_name is not registered.](#)
- [Search server push failed](#)
- [Stream server push failed](#)
- [Unable to contact server](#)
- [Unable to contact search server](#)
- [Unable to contact stream server](#)
- [Upload failed](#)
- [Virtual server already exists](#)
- [Virtual server URL missing](#)

If you encounter any errors that include the text “internal error”, contact Google technical support.

Authentication required**Where It Appears**

When you try to publish a database on a server that requires authentication.

Why the Error Occurred

You entered an invalid user name or password three times.

Resolution

Check with your system administrator to be sure you have permission to publish a database and have a valid user name and password. Then try entering the user name and password.

Note: Be sure that CapsLock is not on.

Database is currently published. Please disable the stream virtual server first.**Where It Appears**

When you try to delete a published database with `geserveradmin --deletedb db_name`. (See

[geserveradmin](#).)

Why the Error Occurred

You cannot delete a published database.

Resolution

Either publish a different database, or disable the virtual stream server, and then try deleting the database again.

Either the virtual server *vs_name* does not exist or is already disabled.

Where It Appears

When you are trying to publish to a virtual server.

Why the Error Occurred

Either the virtual server has been disabled, or you made a typographical error when you entered the command.

Resolution

Use `geserveradmin --listdbs` to list all databases registered on the server. (See [geserveradmin](#).)

Incorrect cache level

Where It Appears

When you add a virtual server.

Why the Error Occurred

You set the cache level to a number other than 1, 2, or 3.

Resolution

Set the cache level to 1, 2, or 3. (See [“geserveradmin” on page A-11](#) for details.)

Invalid database

Where It Appears

When you try to publish a database using the `geserveradmin --publishdb` command. (See [geserveradmin](#).)

Why the Error Occurred

The database path is incorrect, the database has not yet been built, or the database has been corrupted.

Resolution

Enter the correct path or build/rebuild the database.

Plugin already exists.

Where It Appears

When you are trying to register a plug-in.

Why the Error Occurred

The plug-in is already registered.

Resolution

No action is required.

Plugin *plug-in_name* is not registered.

Where It Appears

When trying to create a search tab.

Why the Error Occurred

You tried to create a search tab with a plug-in that is not registered with the server.

Resolution

Register the plug-in using `geserveradmin`. (See [“geserveradmin” on page A-11.](#))

Search server push failed Stream server push failed

Where It Appears

When you try to publish a database.

Why the Error Occurred

The network connection to the server is down, or permissions are set incorrectly where the publisher is attempting to write data.

Resolution

Check your network connections and ensure that the targeted server is running and accessible. Check the access permissions in the publish root directory.

Unable to contact server Unable to contact search server Unable to contact stream server

Where It Appears

When you try to publish a database.

Why the Error Occurred

The specified server is not running, or a service that it requires is not running.

Resolution

Enter the following command as root to restart the server:

```
/etc/init.d/geserver restart
```

Upload failed

Where It Appears

When you try to publish a database.

Why the Error Occurred

The network connection to the server is down, or permissions are set incorrectly where the publisher is attempting to write data.

Resolution

Check your network connections and ensure that the targeted server is running and accessible. Check the access permissions in the publish root directory.

Virtual server already exists

Where It Appears

When you are creating a virtual server with `geserveradmin --addvs`. (See [“geserveradmin” on page A-11](#).)

Why the Error Occurred

You tried to use a virtual server name that has already been used for another server.

Resolution

Always use a unique name for a new virtual server.

Virtual server URL missing

Where It Appears

When you add a virtual server.

Why the Error Occurred

You did not provide a URL for the virtual server you are trying to add.

Resolution

Include a URL in the command. (See [“geserveradmin” on page A-11](#) for details.)

Additional Support

For the most up-to-date information about Google Earth Enterprise, please email the Google Enterprise Support team at enterprise-support@google.com.

[4.4.1 Documentation](#)**Administration Guide**[Home](#)[Configuring Server](#)[Configuring Fusion](#)[Configuring Tutorial Work Space](#)[Command Reference](#)[Common Server Error Messages](#)

Multiple Database Support

[Legal Notices](#)

Multiple Database Support

Google Earth EC 4.0+ supports simultaneous connections to multiple databases. This feature allows users to view extra data and imagery layers drawn on top of each other.

The first database the user connects to is identified as the primary database. After logging in, the user can connect to other databases (which are identified as secondary databases) with the **Add Database** option from the **File** menu.

There is no hard limit to the number of databases that a user can connect to simultaneously. However the disk cache system supports a maximum number of eight databases. For all practical purposes, that means that a user should connect to a maximum of seven secondary databases in addition to the primary database.

Layers

The layers of each database are grouped in the **Layers** panel under the name of the database.

Imagery Layers

Google Earth EC draws the imagery of the primary database first. It draws other imagery layers from secondary databases (if present) on top.

Users can make these additional imagery layers partially or totally transparent using the transparency slider that appears at the bottom of the Layers panel when a secondary database is selected. The transparency slider may not appear if the selected database does not contain any imagery layers. This feature facilitates comparison of imagery layers.

Vector Layers

Google Earth EC draws vector layers independently of each other. Users can enable any number of vector layers from any connected databases (primary or secondary).

Search Tabs

Custom search tabs can be configured for secondary databases, as well as for the primary database. The customization takes effect only after Google Earth EC connects to that database by either logging in to it as the primary database, or by adding it as a secondary database. (See the *Google Earth Enterprise Fusion Reference Guide* for instructions for configuring search tabs and adding them to a database.)

If you log in to a primary database for which custom search tabs are configured, those search tabs appear in Google Earth EC. If you add a secondary database that also has search tabs configured for it, the secondary database's search tabs do NOT appear in Google Earth EC; only the primary database's search tabs appear. The search tabs configured for the secondary database appear in Google Earth EC only if the primary database has no custom search tabs configured for it.

Authentication Considerations

The prevailing authentication mode in any Google Earth EC session (that is, while logged in) is the mode used by the primary databases. For instance, if the primary database has authentication turned off, it is not possible to successfully download data from a database such as `kh.google.com`, because it requires session cookie authentication (`khauth`). However, Google Earth EC does not report any errors while trying to connect to that database; it just fails to download the imagery data.

Authentication Modes

The following list includes all possible authentication modes and the modes with which each is compatible:

Authentication Modes

Mode	Description	Compatible Modes
khauth	Regular authentication used by <code>kh.google.com</code> and other Google Earth servers	khauth, LDAP, LDAP+SSL, noauth
LDAP	HTTP 1.1 authentication with user name/password	LDAP, LDAP+SSL, noauth
LDAP + SSL	HTTPS authentication with user name/password	LDAP, LDAP+SSL, noauth
noauth	Servers that do not require any authentication	LDAP, LDAP+SSL, noauth

Sample Scenarios

A user wants to connect to the following databases:

- `kh.google.com` (khauth authentication mode)
- `publicDB`, which requires no authentication (noauth authentication mode)
- `internalDB`, which uses LDAP authentication (LDAP authentication mode)

The user connects to `kh.google.com` as the primary database, and then adds `publicDB` and `internalDB` as secondary databases.

First, Google Earth EC authenticates itself with `kh.google.com`. Then, when it downloads the first packet from `internalDB`, it prompts the user to enter credentials on that server (the username/password combination configured by the administrator of that database). The primary authentication mode in this case is **khauth**, which is compatible with both **noauth** and LDAP.

Failure Scenarios

Connecting to `publicDB` first then to `kh.google.com` as secondary database fails. Google Earth EC downloads no imagery from `kh.google.com` (although it reports no error), because **noauth**, as noted in the table above, is not compatible with **khauth**.

The same holds true if `internalDB` is the primary database and the user adds `kh.google.com` as a secondary database. In that case, the primary database uses LDAP authentication, which is not compatible with **khauth**.

Another unsupported mode is connecting to a **khauth** primary database and a **miniauth** secondary database. In that case, the user cannot successfully log in to the secondary database (although, as with the first scenario, it reports no error).

[Back to top](#)