Google

Google Earth Enterprise Documentation

For the latest version of this documentation, go to the Google Earth Enterprise help center.

About This Tutorial

Google Earth Enterprise Fusion is designed for organizations that want to display their own geospatial data in Google Earth and Google Maps. You can use it to create graphically rich geographic information system (GIS) databases for distribution to your customers or in-house end users. With Google Earth Enterprise Fusion, you can integrate your own geospatial data, publish it to the Google Earth Enterprise Server, and view it using Google Earth Enterprise Client (EC), the Google Maps API, or the Google Earth API.

Before you begin working with real data in Google Earth Enterprise Fusion, it is important for you to familiarize yourself with the process of defining and building different types of data. This tutorial provides a series of exercises that you can perform in an environment where it is all right to make mistakes. Once you start working with real data, it will be more difficult to correct mistakes, so please take your time working through the tutorial until you feel comfortable with Google Earth Enterprise Fusion.

The Basic Tasks section of this tutorial guides you through building a Google Earth Enterprise Fusion database from raw source material, and publishing it using the Google Earth Enterprise Server, using the tutorial files provided on the Google Earth Enterprise Fusion installation DVD.

Note: Work through the first five lessons in order.

The Advanced Tasks section of this tutorial guides you through more advanced topics. You can work through the advanced lessons in any order you wish. Alternatively, you can skip the Advanced Tasks section until you want to learn about something new. Then, you can come back to the tutorial and work through the lessons of your choice.

Audience

This guide is intended for individuals who are new to Google Earth Enterprise Fusion 4.0. It assumes that you have some familiarity with GIS data and have used a Google Earth client, but have no familiarity with this application. It also assumes that Google Earth Enterprise Fusion, the Google Earth Enterprise Server, and Google Earth EC have been installed either on a network drive accessible to you or on your local workstation. If any of those three applications are not installed, contact your system administrator or refer to *Installation overview*.

In This Guide

This guide provides five basic lessons and nine advanced lessons.

The **Basic Tasks** section includes:

1. Setting Up the Tutorial
2. Defining and Building Resources
3. Defining and Building Projects
4. Defining and Building Databases
5. Publishing and Viewing a Database

The **Advanced Tasks** section includes:

1. Configuring Display Rules for Point Data
2. Configuring Display Rules for Polygon Data
3. Importing and Exporting Style Templates
4. Configuring a Searchable Database
5. Specifying Search Fields for Individual Layers
6. Creating a Map Database
7. Creating Imagery Mosaics
8. Segmenting Large Imagery Files
9. Creating a Historical Imagery Project

Appendix A: Sample Data Files provides a list of all of the sample data files included when you install the tutorial data.

Back to top

Setting Up the Tutorial

The Google Earth Enterprise Fusion tutorial data is provided on the Google Earth Enterprise Fusion installation DVD and should have been installed in `/opt/google/share/tutorials/fusion/` on your workstation or network. In addition, your system administrator should have configured a tutorial environment for you to work on the tutorial lessons, keeping your practice data separate from your live production data.

If Google Earth Enterprise Fusion or the tutorial files are not installed or you encounter an error message that tells you that a tutorial source file is not readable or you cannot save a resource, contact your system administrator or refer to Configure tutorial workspace to install the files and configure the tutorial environment before saving any practice data.

As you follow the steps in this tutorial, you'll begin to learn how to create 2D and 3D databases, as well as becoming familiar with the Google Earth Enterprise Fusion user interface.

- Select the Tutorial Asset Root
- Launch Google Earth Enterprise Fusion
- Build the Asset Navigation Tree
- Define Source Providers

Select the Tutorial Asset Root

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

- When multiple users share a single workstation, you must select your own tutorial asset root.
- When you switch from the tutorial data to real production data, you must select the appropriate asset root.

To configure asset roots for multiple users, see Configure tutorial workspace.

This section explains how to select a different asset root.

Caution: 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, both users are 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 you are 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, you can 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 or your system administrator used when configuring the tutorial asset root. (If you do not know the path of your tutorial asset root, contact your system administrator.)

4. When you return to the prompt, log out as root.
5. Start the system manager by entering:

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

Launch Google Earth Enterprise Fusion

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

To launch Google Earth Enterprise Fusion:

1. Open a Linux terminal window.
2. Enter `fusion` (or `fusion &` to return to the Linux prompt).

When the application starts, the Google Earth Enterprise Fusion graphical user interface (GUI) appears.

Build the Asset Navigation Tree

The components of a Google Earth Enterprise Fusion database are called *assets*. The main location where you store all of your Google Earth Enterprise Fusion assets is called the *asset root*. The asset root is located in the main Google Earth Enterprise Fusion volume--in most cases, `/gevol/assets`. For the tutorial lessons, this guide refers to the *tutorial asset root* to differentiate it from the asset root that contains your live data.

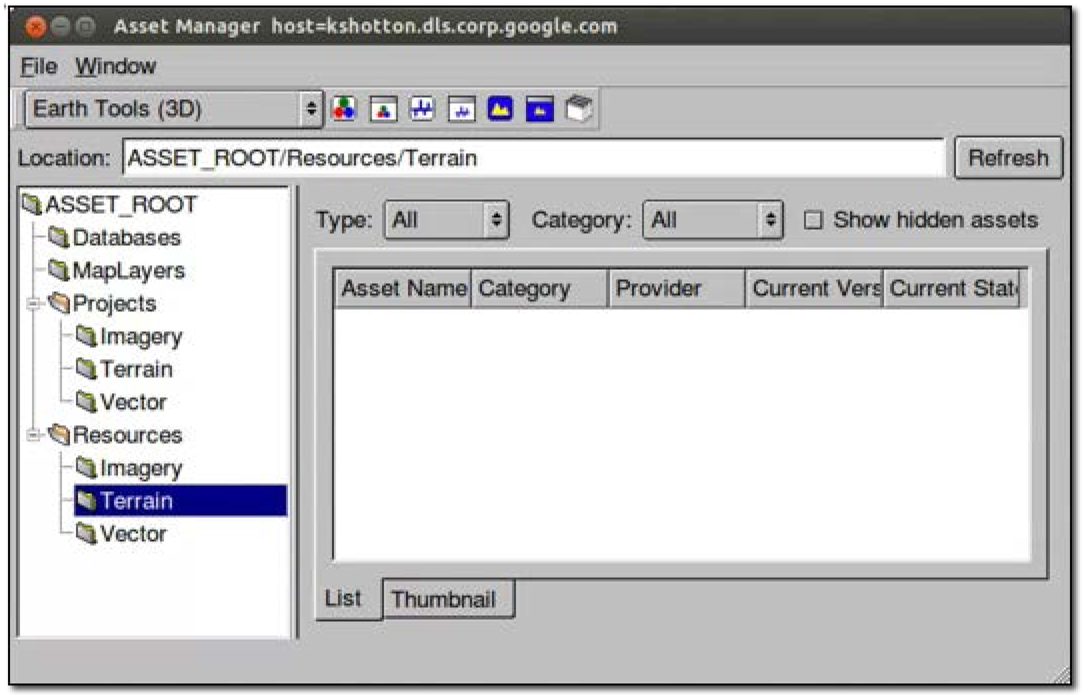
In this exercise, you add subfolders to the tutorial asset root, so you can store your assets in an organized way.

Caution: Once you create subfolders in your asset root, you cannot delete them. This makes it especially important for you to plan out and organize your subfolders before you or anyone else begins working with real data in Google Earth Enterprise Fusion.

To add a subfolder:

1. Open the Asset Manager by selecting **Tools > Asset Manager**.

   The Asset Manager appears:



The asset navigation tree appears on the left side of the Asset Manager, and `ASSET_ROOT` is the only folder on the tree.

2. Right-click **ASSET_ROOT**, and select **New Subfolder** from the context menu.

   The New Subfolder dialog appears.

3. Enter **Resources** as the name of the new subfolder, and click **OK**.

   The new subfolder appears in the asset navigation tree.

4. In the same way, add three more subfolders:

   - **MapLayers**
   - **Projects**
   - **Databases**

   All four subfolders appear in alphabetical order in the asset navigation tree.

5. For the Projects and Resources folders you just created, right-click to create the following subfolders:
    - **Vector**
    - **Imagery**
    - **Terrain**

   You do not need to create Vector, Imagery, and Terrain subfolders for the MapLayers or Databases folders.
6. Close the Asset Manager by clicking the close box (**X**) in the top right corner.

Define Source Providers

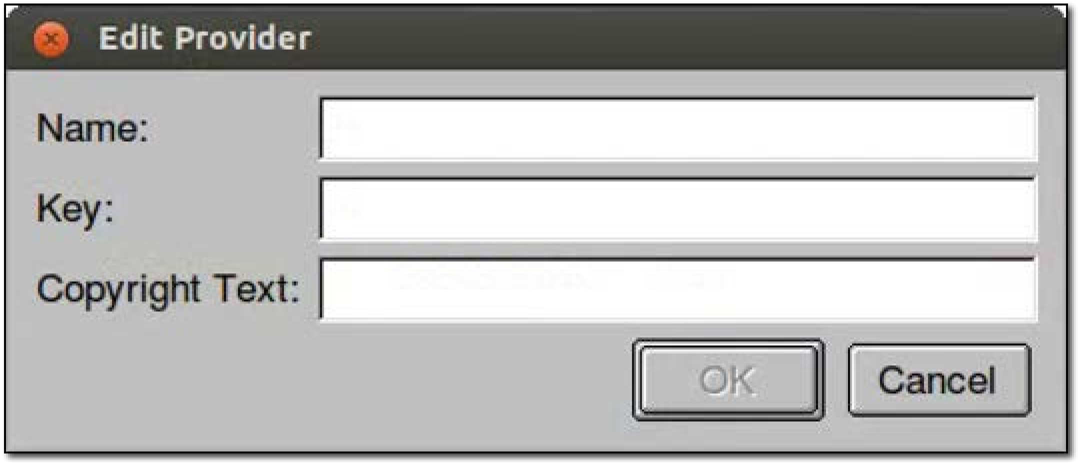
The **Provider Manager** allows you to create a list of organizations that provide the source data you use in Google Earth. For each provider, you specify a unique lookup key (any unique abbreviation you choose) and copyright information. When you create a resource, a data provider is associated with it, so that when the resource is displayed in Google Earth EC, the appropriate copyright information is also displayed.

To create a source provider:

1. Open the Provider Manager by selecting **Tools > Provider Manager**.

2. Click the page icon:

The **Edit Provider** dialog appears.



3. Enter the following information:

   Name: USGS Imagery
   Key: USGS-I
   Copyright Text: Imagery © 2005 USGS

   To enter the copyright symbol, open a text editor, press CTRL-SHIFT-U, then enter the Unicode 00A9. Copy and paste the displayed symbol into the dialog.

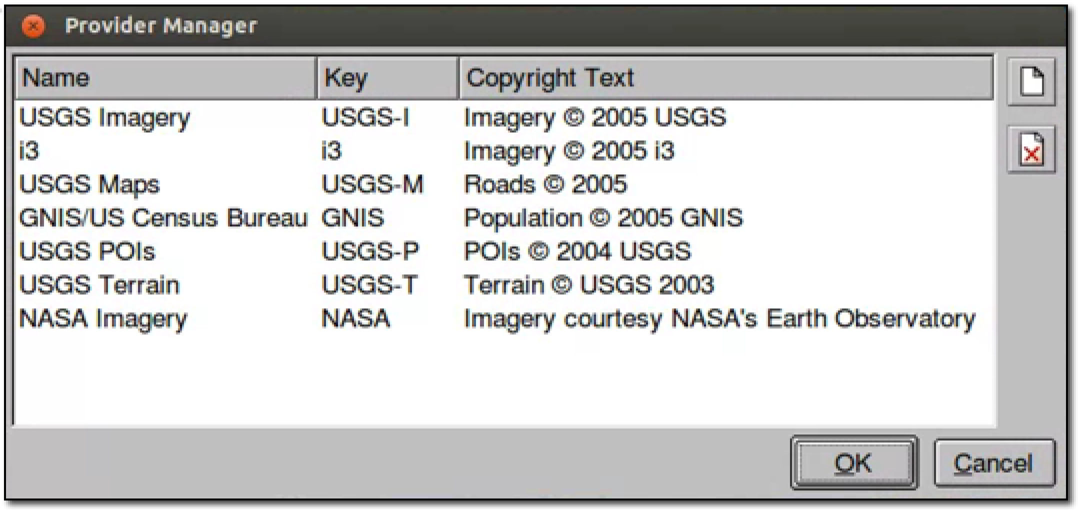
4. Click **OK**.

5. Repeat steps **2** through **4** to add the following source providers:

Name	Key	Copyright Text
i3	i3	Imagery © 2005 i3
USGS Maps	USGS-M	Roads © 2005 USGS

GNIS/US Census Bureau	GNIS	Population © 2005 GNIS
USGS POIs	USGS-P	POIs © 2004 USGS
USGS Terrain	USGS-T	Terrain © USGS 2003
NASA Imagery	NASA	Imagery courtesy NASA's Earth Observatory

The new providers appear in the Provider Manager dialog in the order in which you added them:



6. Click **OK** to close the Provider Manager, and go on to the next lesson.

Back to top

Defining and Building Resources

The first step in preparing data is to define resources by importing the source data into Google Earth Enterprise Fusion. You import imagery, terrain, and vector data separately.

This lesson guides you through defining and building imagery, terrain, and vector resources.

About This Tutorial

Setting Up the Tutorial

Defining and Building Resources

Defining and Building Projects

Defining and Building Databases

Pushing, Publishing, and Viewing a Database

Configuring Display Rules for Point Data

Configuring Display Rules for Polygon Data

Importing and Exporting Style Templates

Specifying Search Fields for Individual Layers

Configuring a Searchable Database

Creating a Map Database

Creating Imagery Mosaics

Segmenting Large Imagery Files

Define Imagery Resources

The following exercises guide you through exploring your source data and defining and building imagery resources. Resources comprise the most basic components of a Google Earth Enterprise Fusion database.

Explore Imagery Source Files

In this exercise, you learn to use the Preview panes in the Google Earth Enterprise Fusion GUI to investigate imagery source files to be sure they cover the desired area before you import them into Google Earth Enterprise Fusion.

The following limitations apply to previewing data.

- Certain display rule settings are ignored:
    - Simplification method
    - Suppress duplicates
    - Elevation/height
    - Highlight style
    - Road label
    - Road shield
- Lines and polygons are drawn as lines only (not filled), so use Line Color or Outline Color, if you want to see the colors of lines or polygons in the Preview pane.
- Labels appear in the assigned color, but scaling and centering are ignored for labels.
- Icons do appear in the Preview pane, but any style settings (color, scale) are ignored.
- Icons are not selectable, and their pop-up text is not displayed.

To explore imagery source files:

1. Click the **Open** icon: .

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

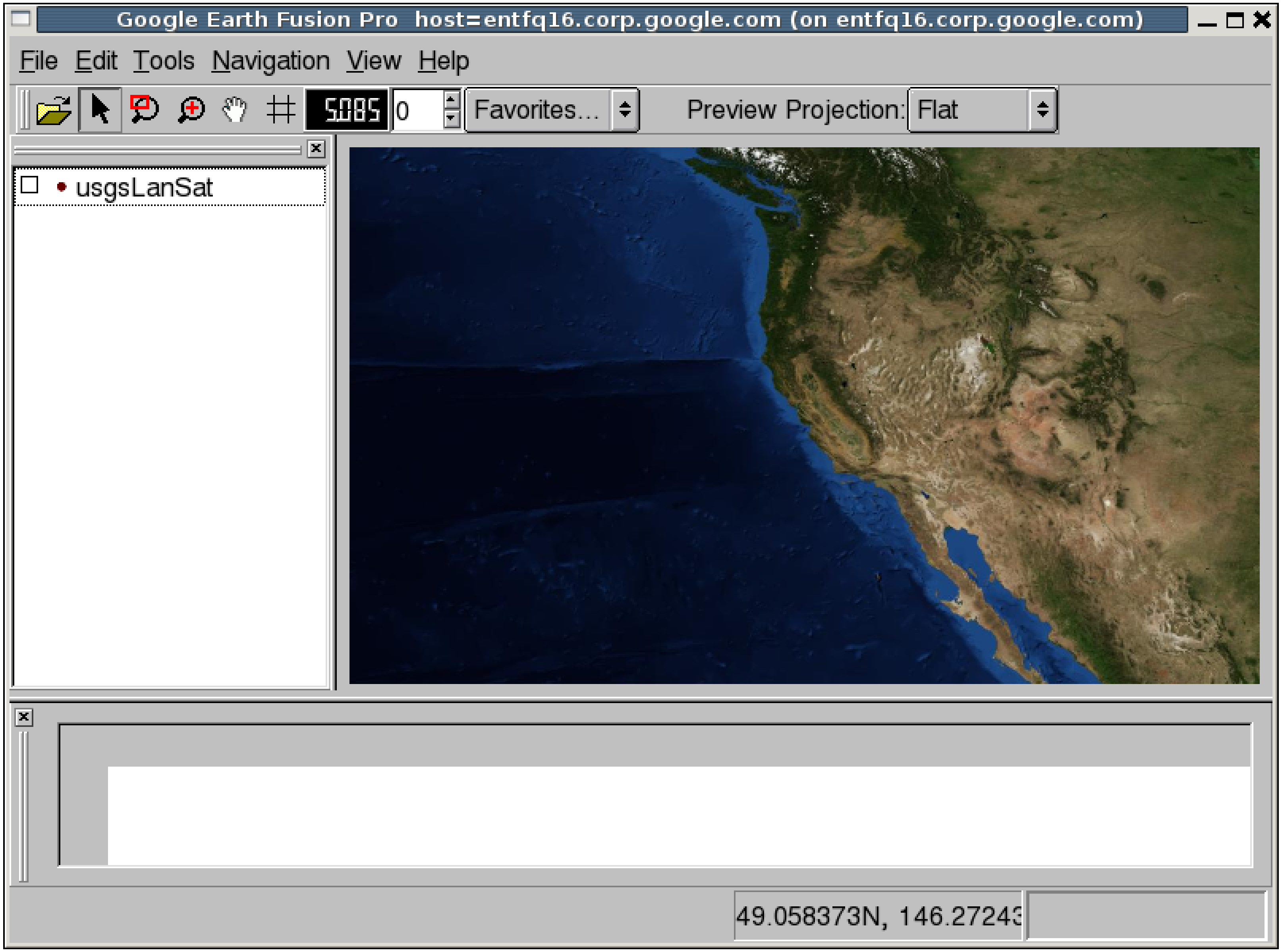
Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

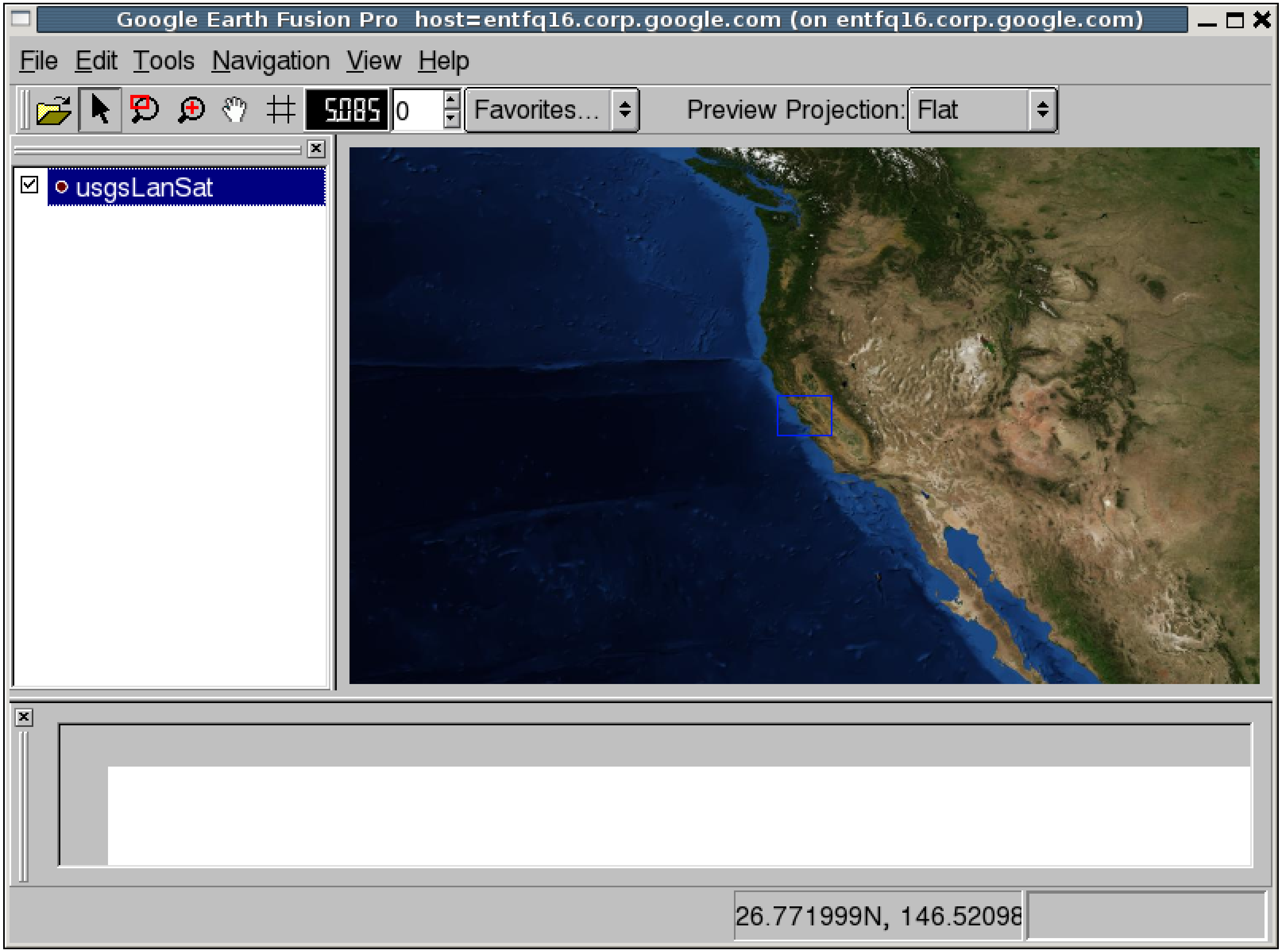
2. Select `/opt/google/share/tutorials/fusion/Imagery/usgsLanSat.jp2` and click **Open**.

   The image name appears in the Preview List pane. This is called a *layer* (an individual source file or resource).



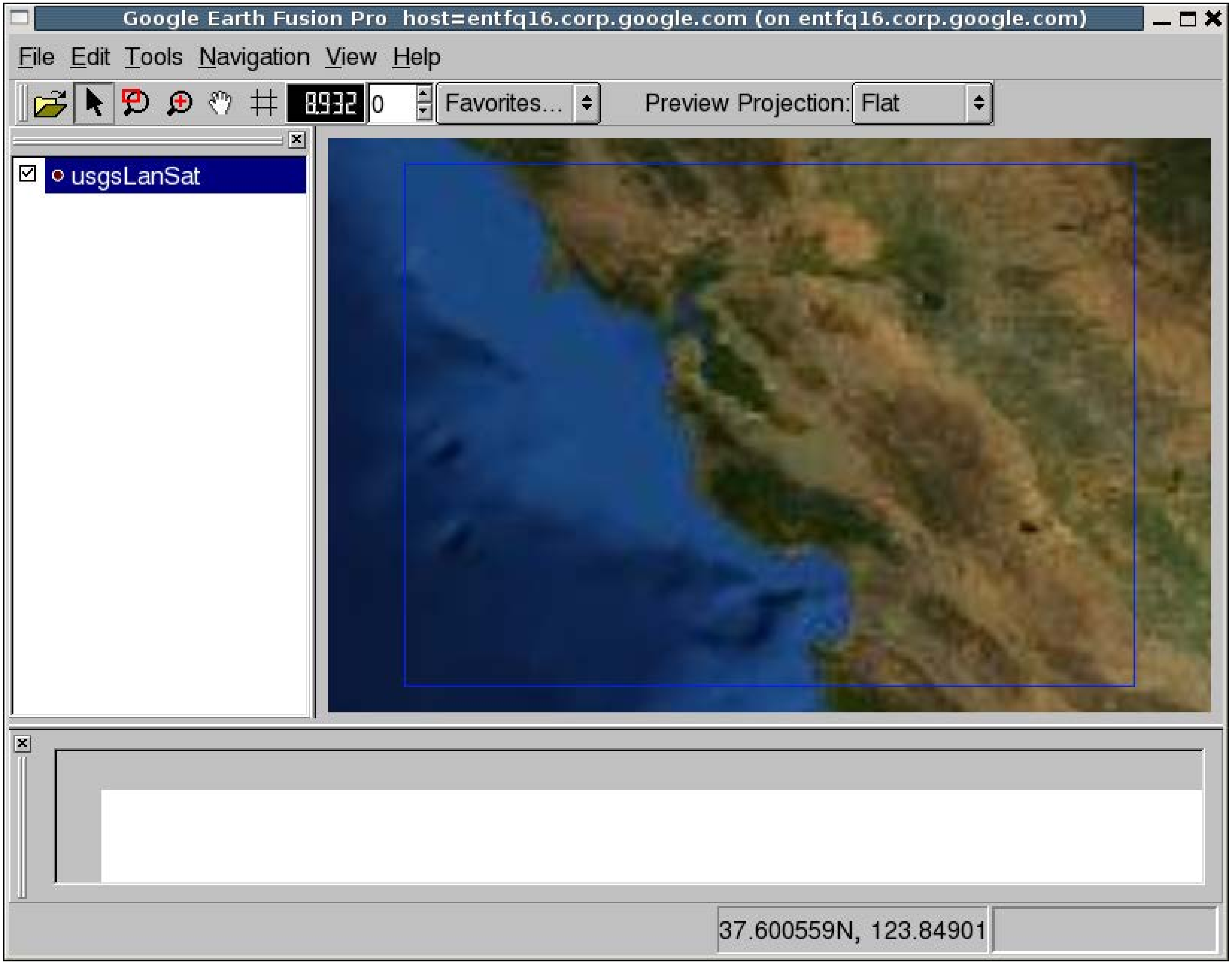
3. Select the layer's checkbox.

A bounding box in the Preview pane indicates the extents of the imagery in the selected file.



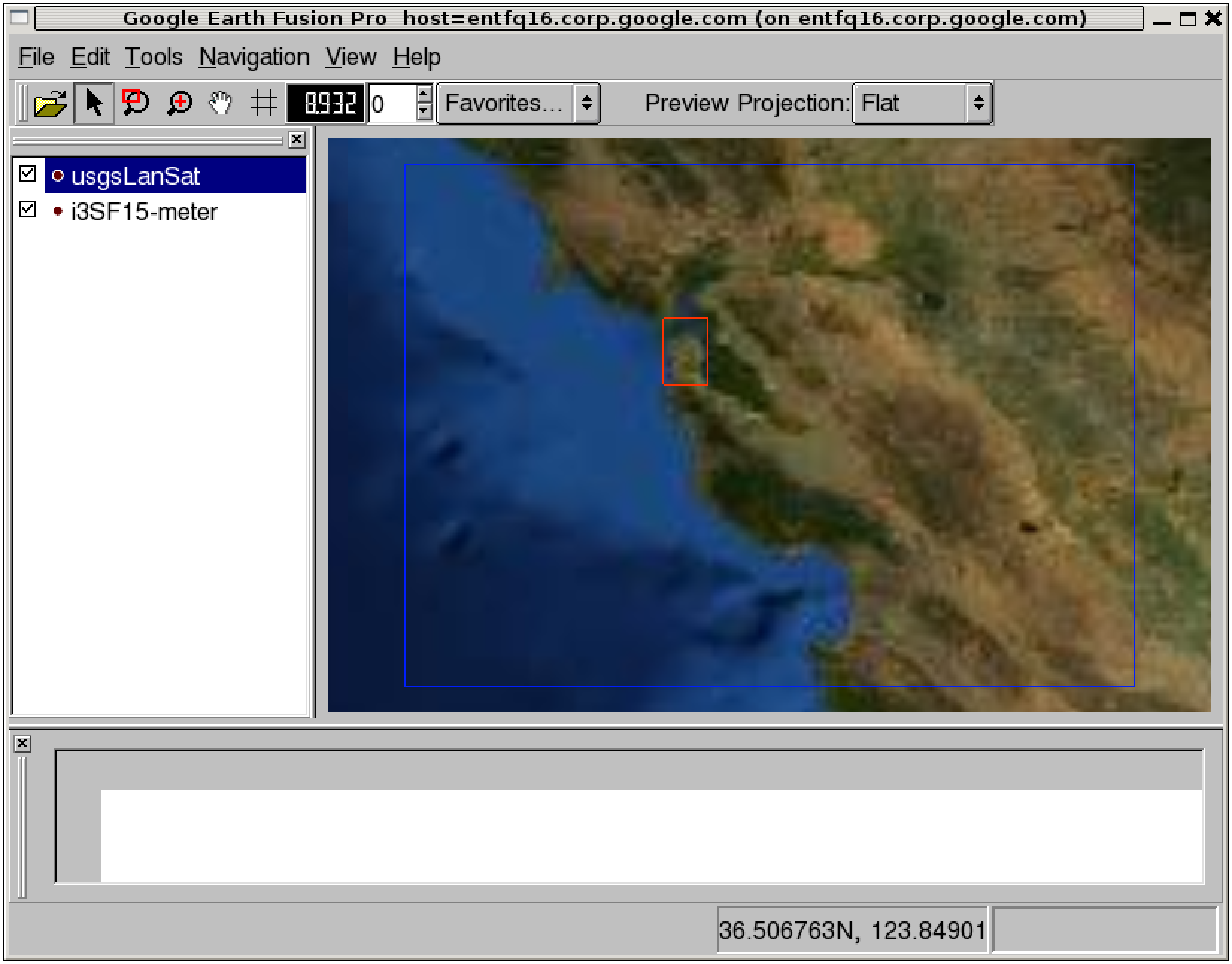
4. Right-click the layer in the Preview List pane, and select **Zoom to Layer** from the context menu.

   The Preview pane zooms to show only the region in the selected imagery file.



5. Repeat the steps above, this time opening the **i3SF15-meter.tif** file.

   When you select the checkbox next to this second layer, a bounding box indicates the extents of the second layer within the first layer. You can see the relationship of each area to the other in the Preview pane.



6. When you finish viewing the imagery layers, right-click either layer in the Preview List pane, and select

Remove All Layers from the context menu.

A message prompts you to confirm that you want to remove all layers from the Preview panes.

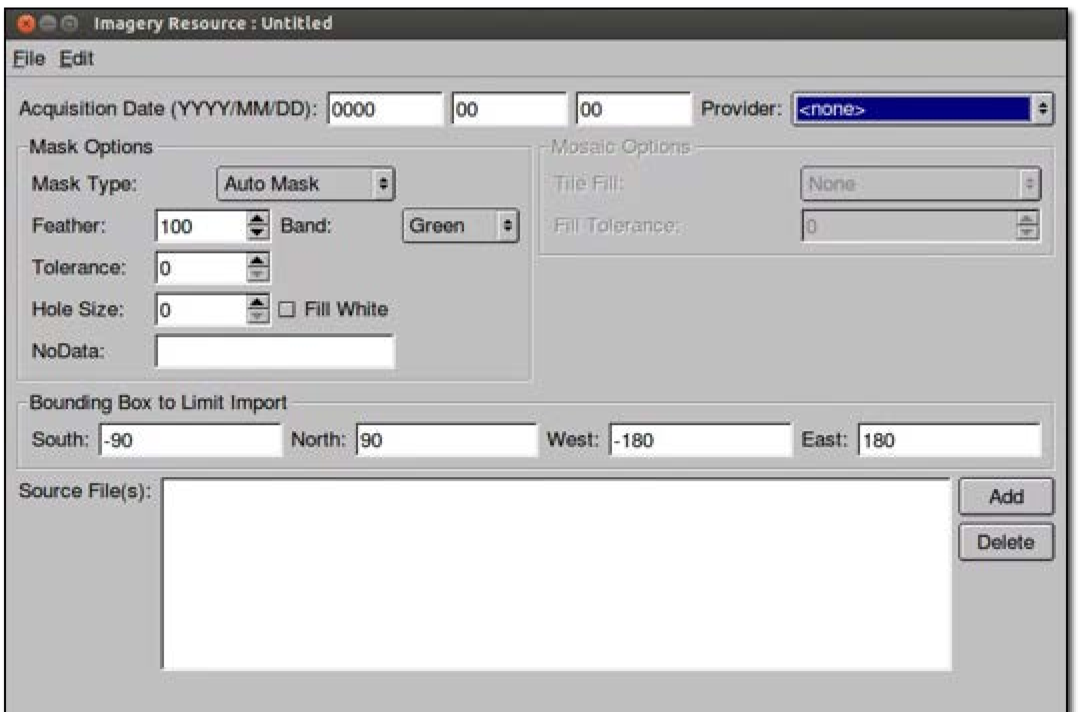
7. Click **OK**.
8. Press **Ctrl-R** to reset the view to the whole Earth.

Define Imagery Resources

In this exercise, you create imagery resources from the imagery data provided for this tutorial.

To define an imagery resource:

1. Select **Tools > Asset Manager**.
2. In the Asset Manager, click  on the toolbar. The Imagery Resource window appears.



3. Set the **Acquisition Date** to today’s date. This is required if you will be creating a historical imagery project later.

The date you set for the imagery **Acquisition Date** is visible in the Google Earth Client when hovering the cursor over a tile. If you add this information to the imagery at a later point, a rebuild of the imagery project is required (as the date needs to be encoded in the JPEG tiles).

When entering date information, the day or month values can be left blank. For example:

- `2008-01-00` indicates January 2008
- `2008-00-00` indicates 2008
- `0000-00-00` indicates undefined

Leading zeros are not required when entering dates.

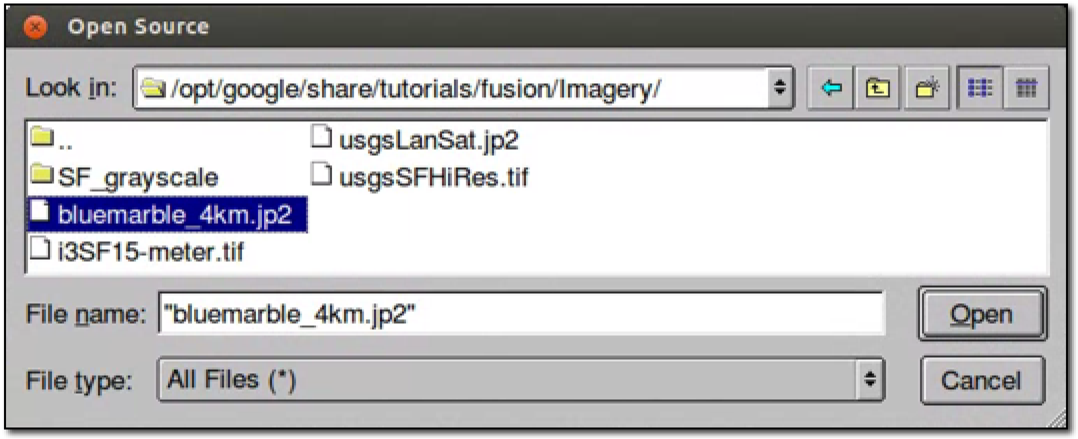
Note: When you work with real data, the acquisition date should reflect the date the data was released. You can obtain this information from the provider. However, for the purpose of simplifying this tutorial, use the current date for all **Acquisition Date** fields.

---

4. Select **NASA Imagery** from the **Provider** drop-down list. Notice that the Provider drop-down list contains all of the providers you added earlier.
5. Set the **Mask Type** to **No Mask**.

   You are about to import the BlueMarble source data. Since that data covers the entire world, there is no need for a mask. That is, there is no fill data to mask out in the imagery.
6. Click **Add**.

   The Open Source dialog opens to the `/opt/google/share/tutorials/fusion/Imagery` folder.



7. Select the `bluemarble_4km.jp2` file, and click **Open**.
8. Select **File > Save** and navigate to the `/ASSET_ROOT/Resources/Imagery` folder you created in the previous chapter.
9. Enter the name **BlueMarble** for the resource, and click **Save**.

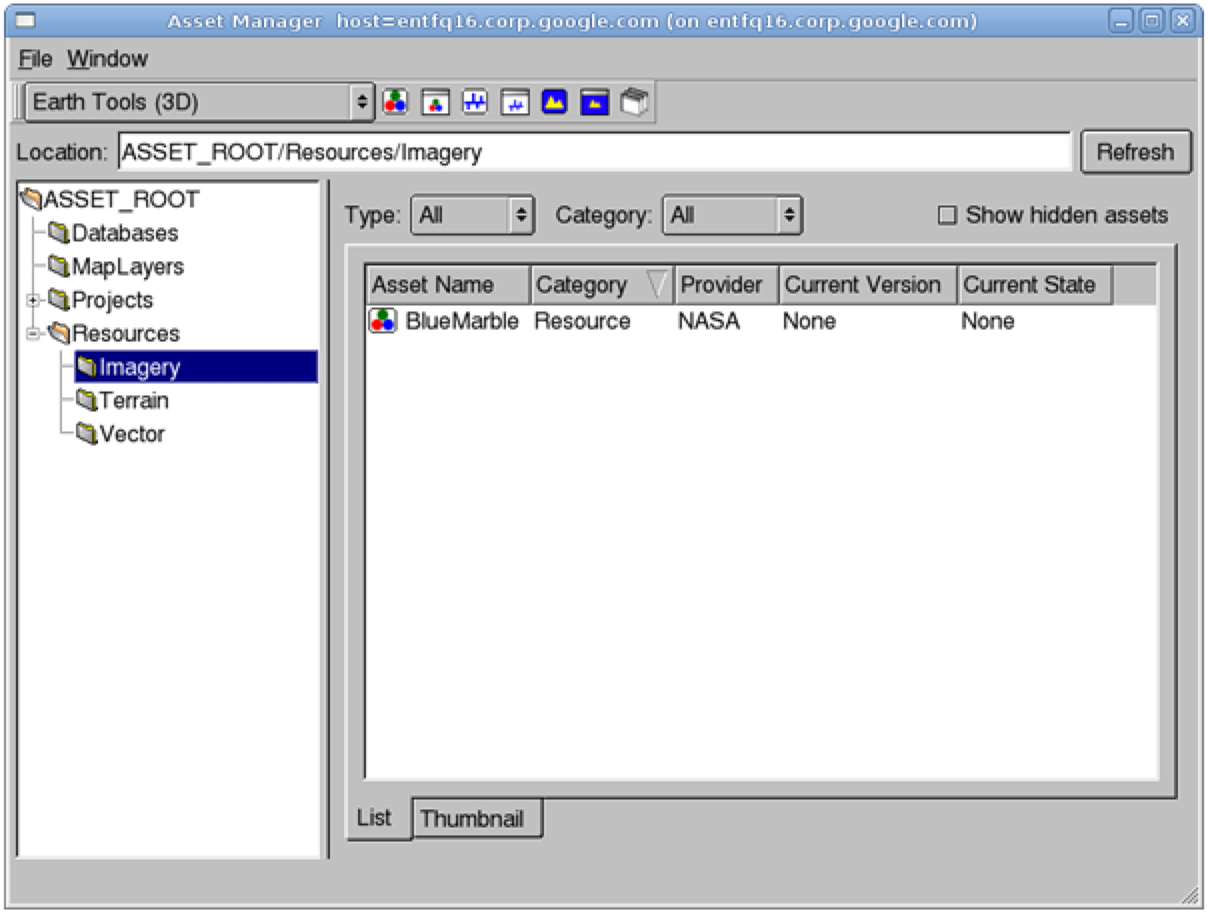
Notes: Your system administrator should have configured a tutorial environment for you to work on the tutorial lessons, keeping your practice data separate from your live production data. If you encounter an error message that tells you that a tutorial source file is not readable or you cannot save a resource, contact your system administrator or refer to the **Google Earth Enterprise Administration Guide** and configure the tutorial environment yourself before saving any practice data.

When you finish using any of the asset editors (such as the Imagery Resource Editor used in this exercise), you can either leave it open and move it to the side or close it. Generally, if you know you have more work to do on a given asset, you leave the editor open. If you know you are done with an asset for now, you can close it and get it out of the way.

Caution: When you are defining assets for your live production system, it is important to remember that you cannot delete or edit asset names after you save them.

In the Asset Manager, the name of the resource appears on the right when you select the

`/ASSET_ROOT/Resources/Imagery` folder in the asset navigation tree.



Build an Imagery Resource

Before you can view the imagery resource in the Preview pane or include it in a project, you must build it. You do not have to build each resource right away, however. You can define several resources and then build them all at the same time, if you prefer. There are advantages and disadvantages to both approaches. You can develop your own routine as you get more comfortable with Google Earth Enterprise Fusion.

In this exercise, you build the first resource right away.

To build an imagery resource:

1. In the Asset Manager, select the `/ASSET_ROOT/Resources/Imagery` folder.

BlueMarble appears on the right with the **Current Version** and the **Current State** set to **None**, indicating that the resource has not yet been built.

2. Right-click **BlueMarble**, and select **Build** from the context menu. The status of the resource immediately changes to **Queued** and then to **In Progress**.

   Note: Because imagery files are data intensive, it can take some time to build imagery resources.

3. Double-click the **Current Version** or **Current State** column for the resource to view the progress of the build.
4. Right-click **BlueMarble**, and select **Current Version Properties**.

   The **Version Properties** dialog displays the most recent version of that resource. You can expand the version tree to view the status of the build in real time by clicking the **+** signs.

5. When you are done reviewing the information in the **Version Properties** dialog, close that window.

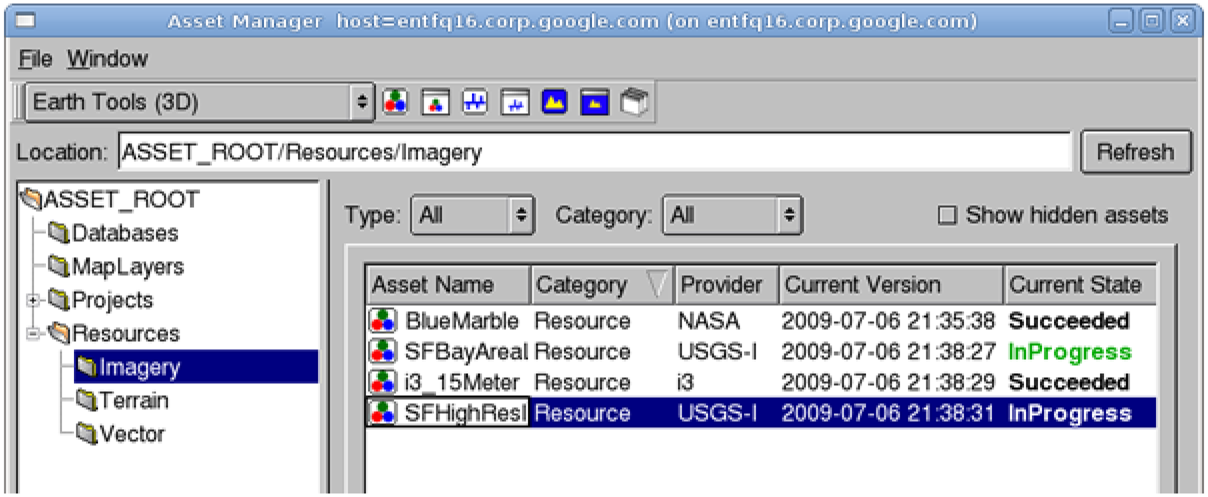
When the BlueMarble resource finishes building, its **Current State** column in the Asset Manager changes to **Succeeded**, and its **Current Version** column changes to the date and time the most recent build was started.

Define and Build the Remaining Imagery Resources

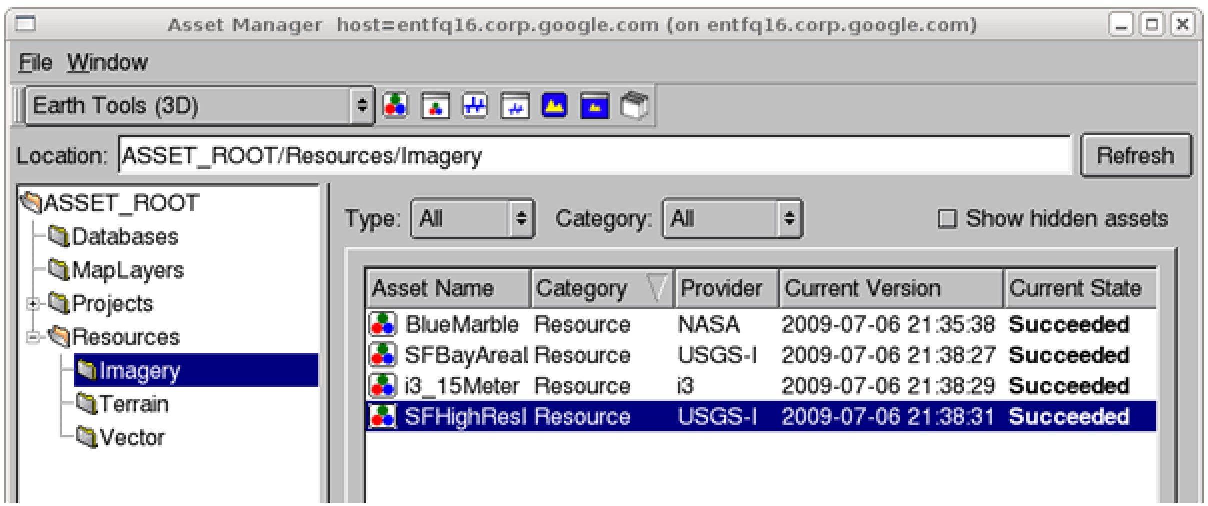
Now that you have defined and built one imagery resource, you can define and build the remaining imagery resources provided for this tutorial:

Name (Resources/Imagery/...)	Acquisition Date	Provider	Mask	Source file
SFBayAreaLanSat_20021010	Today's date	USGS Imagery	**Auto Mask** **Tolerance**: 2 Default for all other mask values.	`usgsLanSat.jp2`
i3_15Meter_20041010	Today's date	i3	**Auto Mask** Default values.	`i3SF15-meter.tif`
SFHighResInset_20061010	Today's date	USGS Imagery	**Auto Mask** Default values.	`usgsSFHiRes.tif`

While the resources are building, the Asset Manager list looks something like this:



It could take several minutes to build all of the imagery resources. When the builds are all complete, it looks like this:



Preview the Imagery Resources

After you successfully build the imagery resources, you can view some of them in the Preview pane of the Google Earth Enterprise Fusion GUI.

To preview imagery resources:

1. In the Asset Manager, drag and drop the **SFHighResInset** resource from the Asset Name column onto the Preview List pane.

2. Drag and drop the **SFBayAreaLanSat** resource onto the Preview List pane, and then close the Asset Manager.

Note: Google Earth Enterprise Fusion displays the resources in the order in which they are listed in the Preview List pane with the last asset on the list at the bottom of the stack and the first asset on the list on top. In this case, the SFHighResInset resource provides much higher resolution imagery for a small area of the SFBayAreaLanSat resource, so you want SFHighResInset to appear on top.

---

3. Check the box next to each resource in the Preview List pane to display the associated imagery.

   Bounding boxes appear where the imagery is located on the base imagery; however, they appear to be very small because the display level is so high.

4. Right-click **SFBayAreaLanSat** in the Preview List pane, and select **Zoom to Layer** from the context menu.

   The Preview pane zooms to the outermost edges of the selected layer. Notice the bounding box for the other resource.

5. Right-click **SFHighResInset** in the Preview List pane, and select **Zoom to Layer** from the context menu.

   The Preview pane zooms to the outermost edges of the selected layer.

6. To prepare for the next exercise, zoom out to a display level between 11 and 12  to view more of the San Francisco bay area, as shown in the following graphic.

   To zoom out, you can either select  and then click in the Preview pane and push the mouse away from you, or roll the mouse wheel away from you.

Define Terrain Resources

Defining terrain resources is very similar to defining imagery resources. The following exercises guide you through defining and building terrain resources.

Explore Terrain Source Files

As with imagery files, you can preview terrain source files to be sure they cover the correct area before you convert them to resources. Although you learned about previewing source files in a previous exercise, this exercise gives you an opportunity to learn about more about the preview tools.

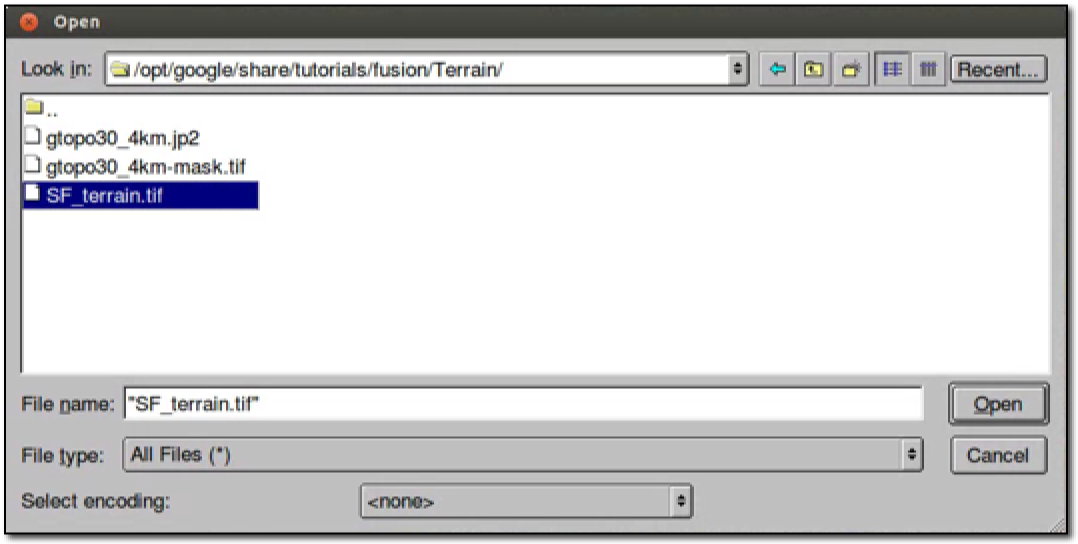
To explore terrain source files:

1. Click .

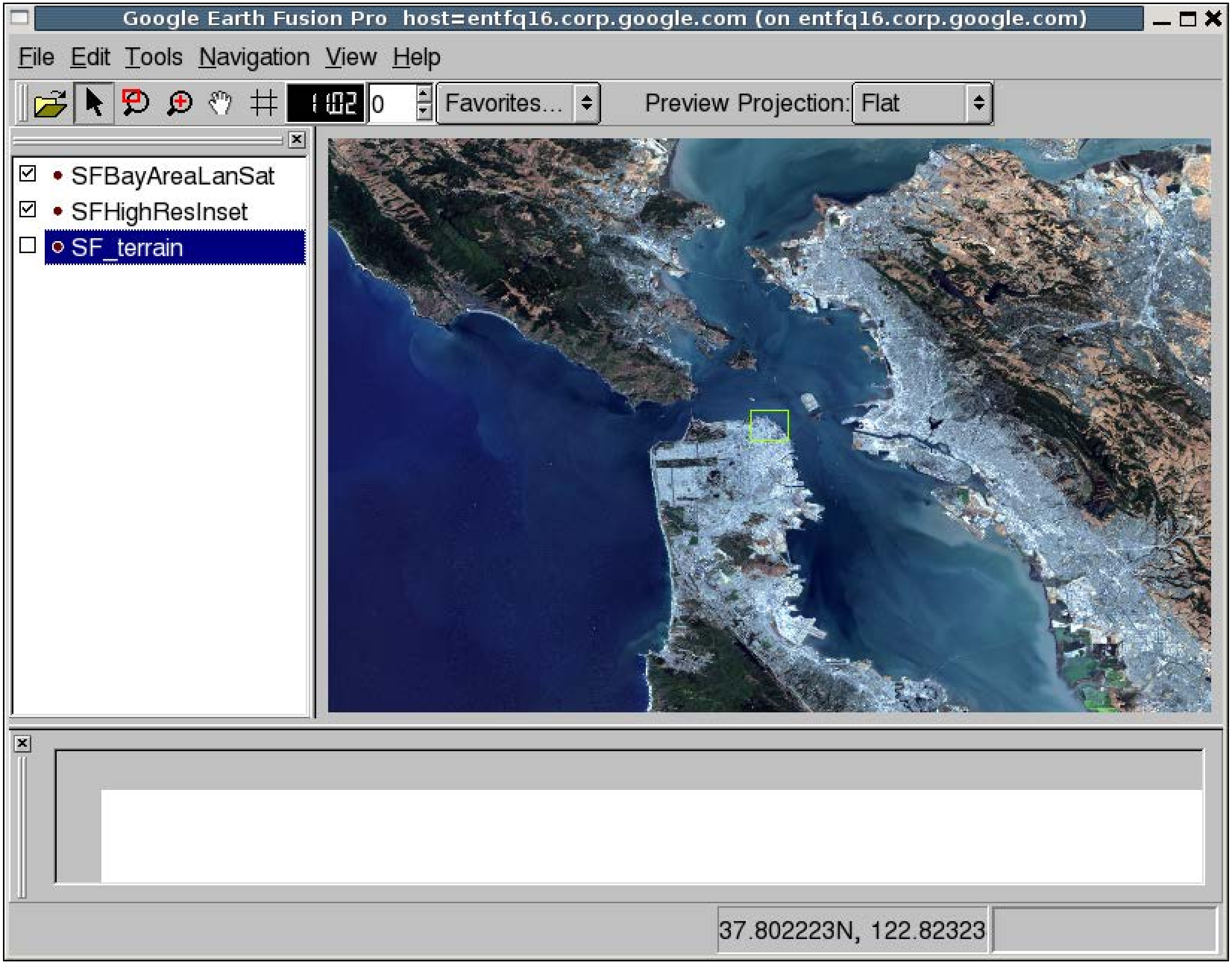
   The Open dialog appears.

2. Navigate to the `/opt/google/share/tutorials/fusion/Terrain` folder.

3. Select `SF_terrain.tif`, and click **Open**.



The new layer name appears in the Preview List pane unchecked.

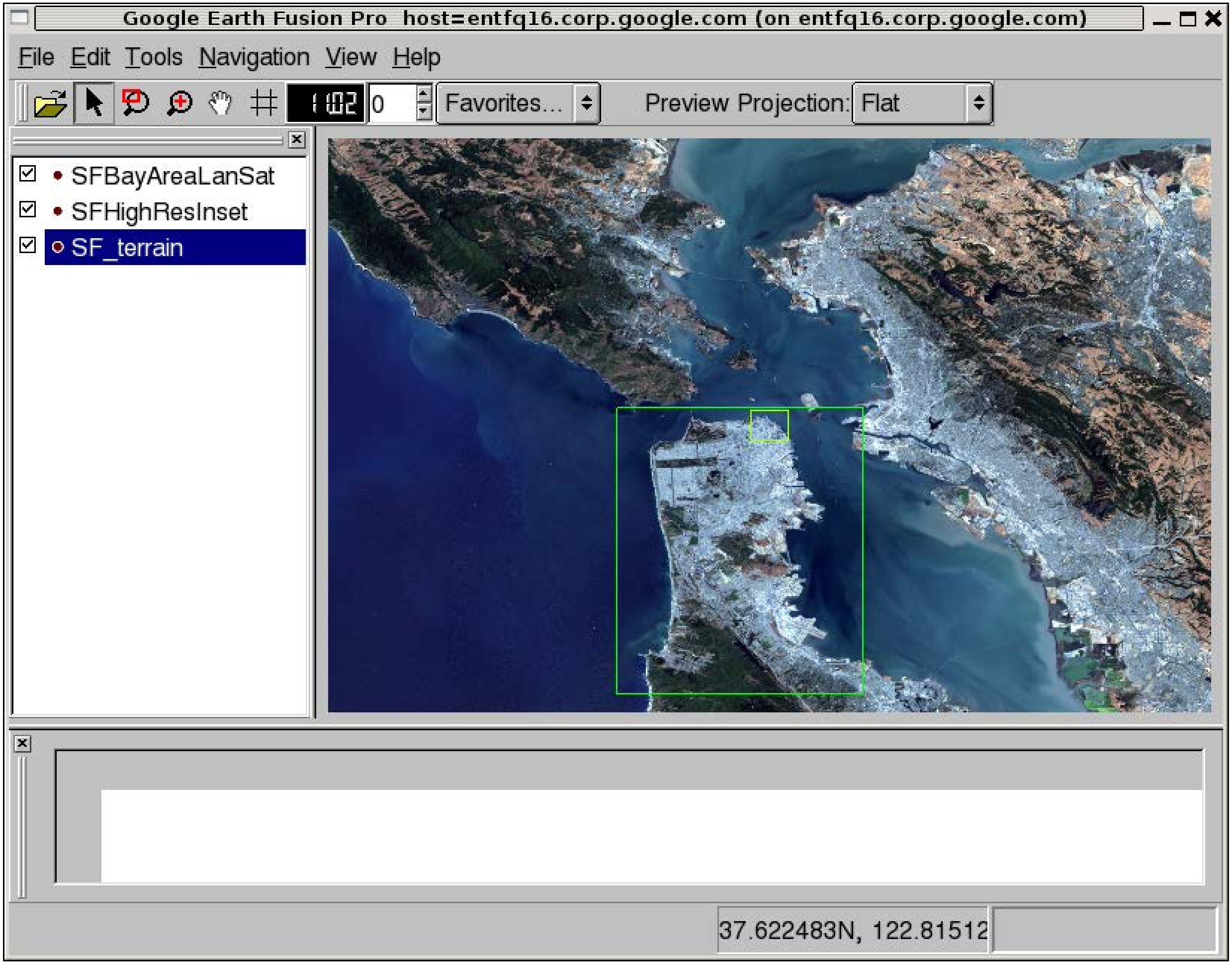


4. Select **Enable All Layers** from the Edit menu to check the boxes for all layers.

   A bounding box appears for the terrain layer.

5. Zoom out a bit to see the entire bounding box.

Note: The Preview pane displays a bounding box for terrain source data, not the actual terrain imagery. You must define and build terrain resources to be able to see a preview of the actual terrain.



6. Right-click **SF_terrain**, and select **Zoom to Layer** from the context menu.

   This fills the Preview pane with the city of San Francisco. The high-resolution imagery inset, SFHighResInset, is still part of this view, and the SFBayAreaLanSat provides the background imagery.

7. Use the toolbar buttons and/or your mouse wheel to zoom in and out to explore the terrain extents on the underlying imagery.
8. Right-click any layer and select **Remove All Layers**.

   A message prompts you to confirm that you want to remove all layers.

9. Click **OK**.

   All of the layers disappear from the Preview panes. You can leave the preview pane zoomed in to prepare for an upcoming exercise, even though the imagery is too close to make out any details at this point.

Define Terrain Resources

In this exercise, you create terrain resources from the terrain data provided for this tutorial.

To define a terrain resource:

1. Open the Asset Manager, and click  on the toolbar.

   The Terrain Resource Editor appears.

2. Set the acquisition date to today's date in year-month-day format by clicking each section of the date and enter the values.

3. Select **USGS Terrain** from the Provider drop-down list.

4. Set the Mask Type (under Mask Options) to **Have Mask**.

   The mask file for your import must be located in the same folder as the source file, and the file name must match the name of the source file with `-mask` appended. For example, in the tutorial files provided, the source file is called `gtopo30_4km.jp2`, and its mask file is named `gtopo30_4km-mask.tif`. Google Earth Enterprise Fusion automatically applies the mask file by reference to the source file.

   Note: You can use the **Have Mask** option for resources that contain one source file only.

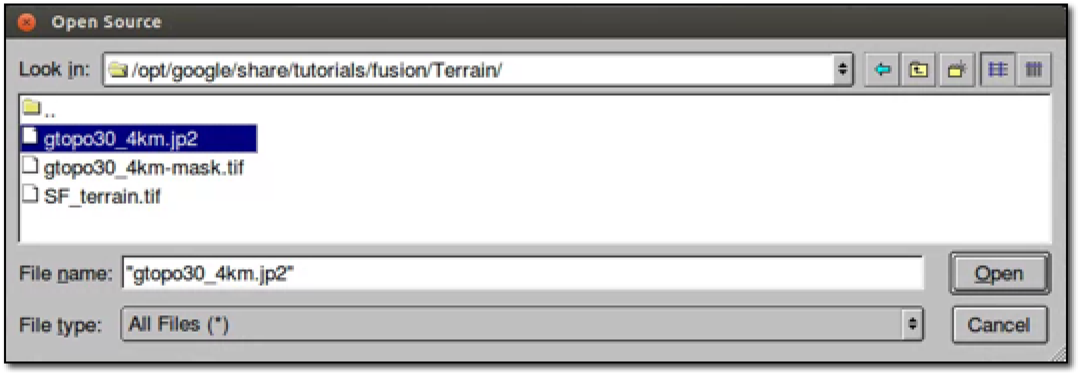
   ---

5. Accept the Elevation Units default, **Meters**.

6. Click **Add**.

   The Open Source dialog opens to the `/opt/google/share/tutorials/fusion/Terrain` folder.

7. Select **gtopo30_4km.jp2**, and click **Open**.



The selected file appears on the Source File(s) list.

8. Select **Save** from the File menu.

   The Save dialog appears.

9. Navigate to the `/ASSET_ROOT/Resources/Terrain` folder you created in *Setting Up the Tutorial*.
10. Enter the name **WorldTopography** for the resource, and click **Save**.
11. Close the Terrain Resource Editor dialog.

    The name of the resource appears on the right when you select the `/ASSET_ROOT/Resources/Terrain` folder in the asset navigation tree in **Asset Manager**.

To define another terrain resource:

1. In the **Asset Manager**, click  on the toolbar.

   The Terrain Resource Editor appears.

2. Set the acquisition date to today’s date in year-month-day format by clicking each section of the date and enter the values.
3. Select **USGS Terrain** from the Provider drop-down list.
4. Set the Mask Type (under Mask Options) to **Auto Mask**, and accept the default mask settings:
   - Feather: **100**
   - Hole Size: **0**

- NoData: **-99999:0**

(For details about masking, see the **Mask Options** section in the **Defining Resources** chapter of the **Reference Guide**.)

5. Accept the Elevation Units default, **Meters**.
6. Click **Add**.

   The Open Source dialog appears.
7. From the `/opt/google/share/tutorials/fusion/Terrain` folder, select **SF_terrain.tif**, and click **Open**.

   The selected file appears on the Source File(s) list.
8. Select **Save** from the File menu.

   The Save dialog appears.
9. Navigate to the `/ASSET_ROOT/Resources/Terrain` folder you created in *Setting Up the Tutorial*.
10. Enter the name **SFTerrain** for the resource, and click **Save**.
11. Close the Terrain Resource Editor dialog.

    The name of the resource appears on the right when you select the `/ASSET_ROOT/Resources/Terrain` folder in the asset navigation tree in **Asset Manager**.

Build and Modify Terrain Resources

As with imagery resources, in this exercise, you build the terrain resources right away.

Note: The WorldTopography terrain resource is quite large and could take up to 30 minutes to build, depending on the speed of your CPU. It is a good idea to start this exercise close to lunch time or just before you attend a meeting, so it can be building while you are busy doing something else.

To build and modify a terrain resource:

1. In the Asset Manager, select the `/ASSET_ROOT/Resources/Terrain` folder.

The terrain resources appear on the right with the Current Version and the Current State set to **None**, indicating that the resources have not yet been built.

2. Right-click **WorldTopography**, and select **Build** from the context menu.

   The status of the resource changes to **Queued** and then to **In Progress**.

3. Right-click **SFTerrain**, and select **Build** from the context menu.

   The status of the SFTerrain resource changes to **Queued** until the WorldTopography resource finishes building and then to **In Progress**.

   When each resource finishes building, the Current State column in the Asset Manager changes to **Succeeded**, and its Current Version column changes to the date and time the most recent build was started.

   Drag the **SFTerrain** resource into the Preview List pane, and check the box next to it.

4. Right-click **SFTerrain**, and select **Zoom to Layer**.

   The bounding box for the terrain resource appears in the Preview pane, and the grayscale terrain imagery appears in the bounding box.

Note: Since the Preview pane in Google Earth Enterprise Fusion is meant for preview purposes only, it does not render terrain in 3D like Google Earth EC. Instead, it renders a grayscale interpretation of the terrain. The lighter pixels represent the higher elevations, and the darker pixels represent lower elevations. For this reason, the Preview pane is not useful for comparing elevation values from different resources.

Because the background is a very low-resolution image, it is hard to determine what you are looking at. The solution to this problem is to add high-resolution imagery to the Preview pane to give you a frame of reference.

5. In the Asset Manager, navigate to `ASSET_ROOT/Resources/Imagery`, and drag **SFBayAreaLanSat** to the Preview List pane, and check the box next to it.

   The higher-resolution imagery appears under the terrain imagery, so you can get a better idea of where the terrain is located.

   The mask automatically generated by Google Earth Enterprise Fusion removes all of the fill data in the terrain resource, using the feather value specified in the Terrain Resource Editor.

The preview shows that the default feather of 100 pixels is far too aggressive, removing much of the terrain data around the coastline. In a real-world situation, you can provide your own mask for the data to be sure you can see every detail around the coastline. For this tutorial, however, simply adjust the feather value for the mask that Google Earth Enterprise Fusion generates automatically.

6. Double-click the for the **SFTerrain** resource in the Asset Manager.

   The Terrain Resource Editor appears with all of the SFTerrain resource's settings.

7. Change the Feather value to **5**.
8. Select **Save** from the File menu.

   Google Earth Enterprise Fusion saves the terrain resource with the same name.

9. Build the SFTerrain resource again.
10. Drag and drop the **SFTerrain** resource onto the Preview List pane.

    The new version of the resource appears at the bottom of the list with a number after the resource name to distinguish it from other versions of the same resource.

11. Uncheck the box next to the original version of the resource, and check the box next to the modified version.

    The Preview pane displays the modified version of the resource.

    Because the second version of the resource is listed below the imagery resource in the Preview List pane, it appears below the imagery in the Preview pane. Because of the list order, the imagery resource is actually obscuring the terrain resource. All you can see is the mask.

12. Right-click any layer in the Preview List pane, and select **Remove All Layers** from the context menu.

    A message prompts you to confirm that you want to remove all layers.

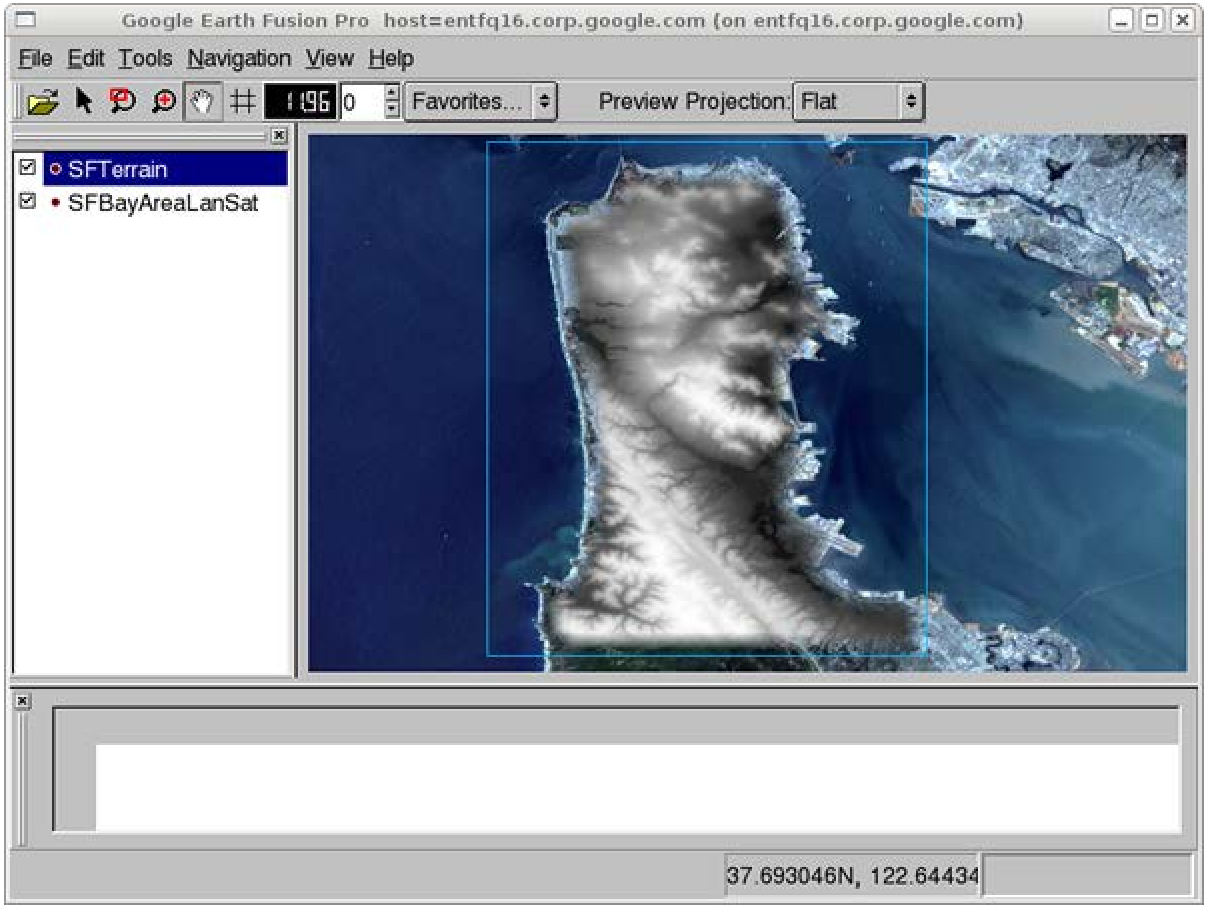
13. Click **OK**.

    All of the layers disappear from the Preview panes.

14. In the Asset Manager, navigate to `ASSET_ROOT/Resources/Terrain`, and drag and drop the new version of the **SFTerrain** resource onto the Preview List pane, and check the box next to it.

15. Navigate to `ASSET_ROOT/Resources/Imagery`, and drag **SFBayAreaLanSat** to the Preview List pane, and check the box next to it.

    Now you can see the preview of the terrain over the imagery. With a feather value of 5, the mask removes the fill data but removes much less of the real data, allowing the actual terrain data to be visible out to the edges of the coastline.

16. When you finish examining the preview, right-click either layer in the Preview List pane, and select **Remove All Layers** from the context menu.

    A message prompts you to confirm that you want to remove all layers.

17. Click **OK**.

All of the layers disappear from the Preview panes.

18. Press **Ctrl-R** to reset the view to the whole Earth.

Define Vector Resources

The following exercises guide you through the process of defining and building a vector resource for California highway data.

Explore Vector Source Files

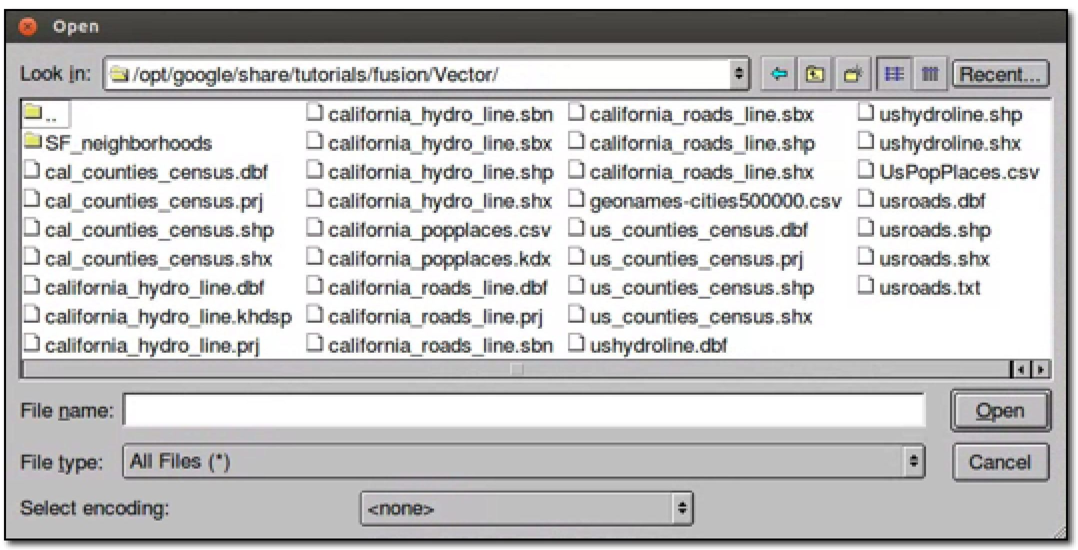
As with imagery and terrain files, you can preview vector source files to be sure they provide the data you want before you convert them to resources. This exercise provides an opportunity for you to use some additional preview tools.

To explore vector source files:

1. Click .

   The Open dialog appears.

2. Navigate to the `/opt/google/share/tutorials/fusion/Vector` folder.

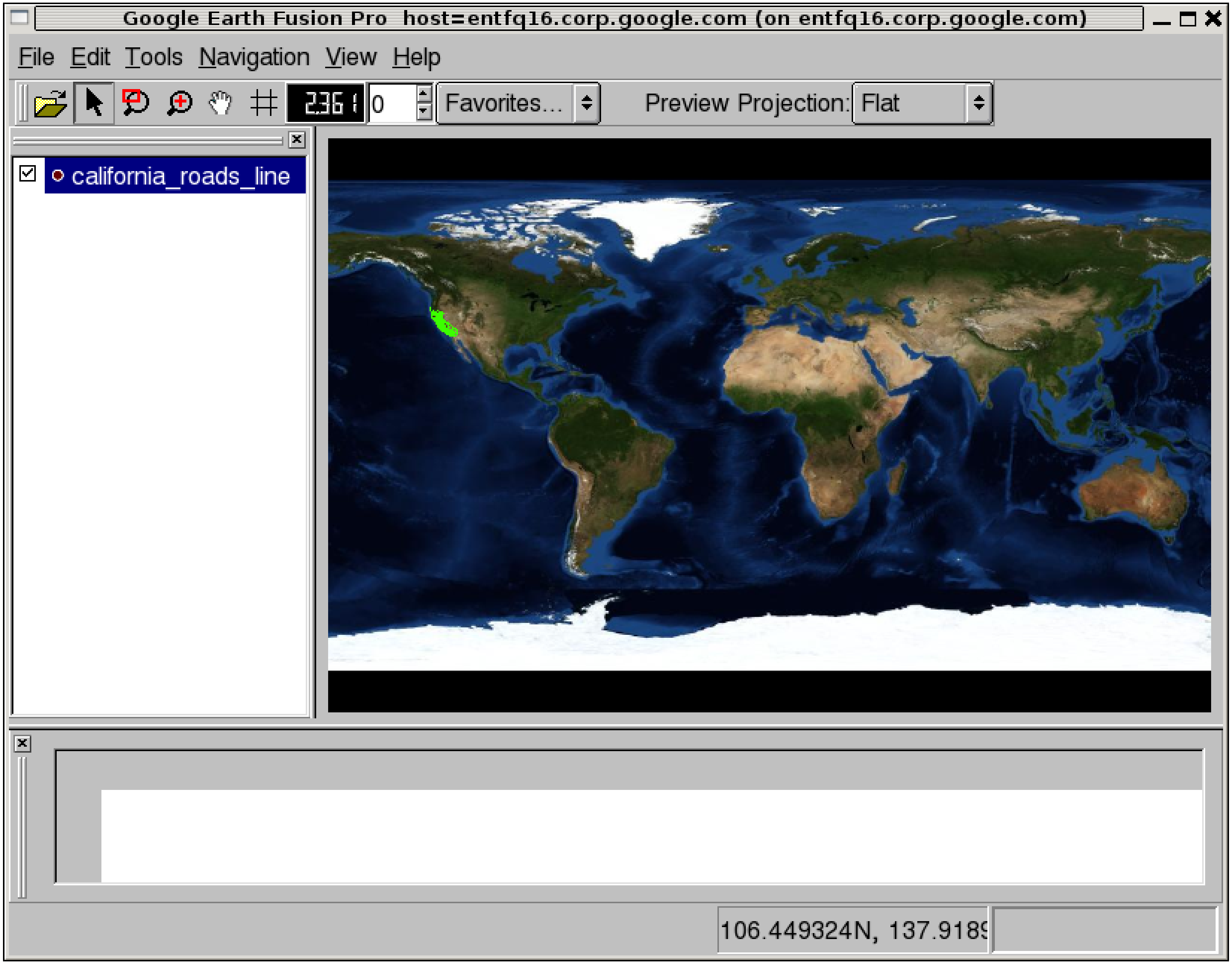


3. Select `california_roads_line.shp`, and click **Open**.

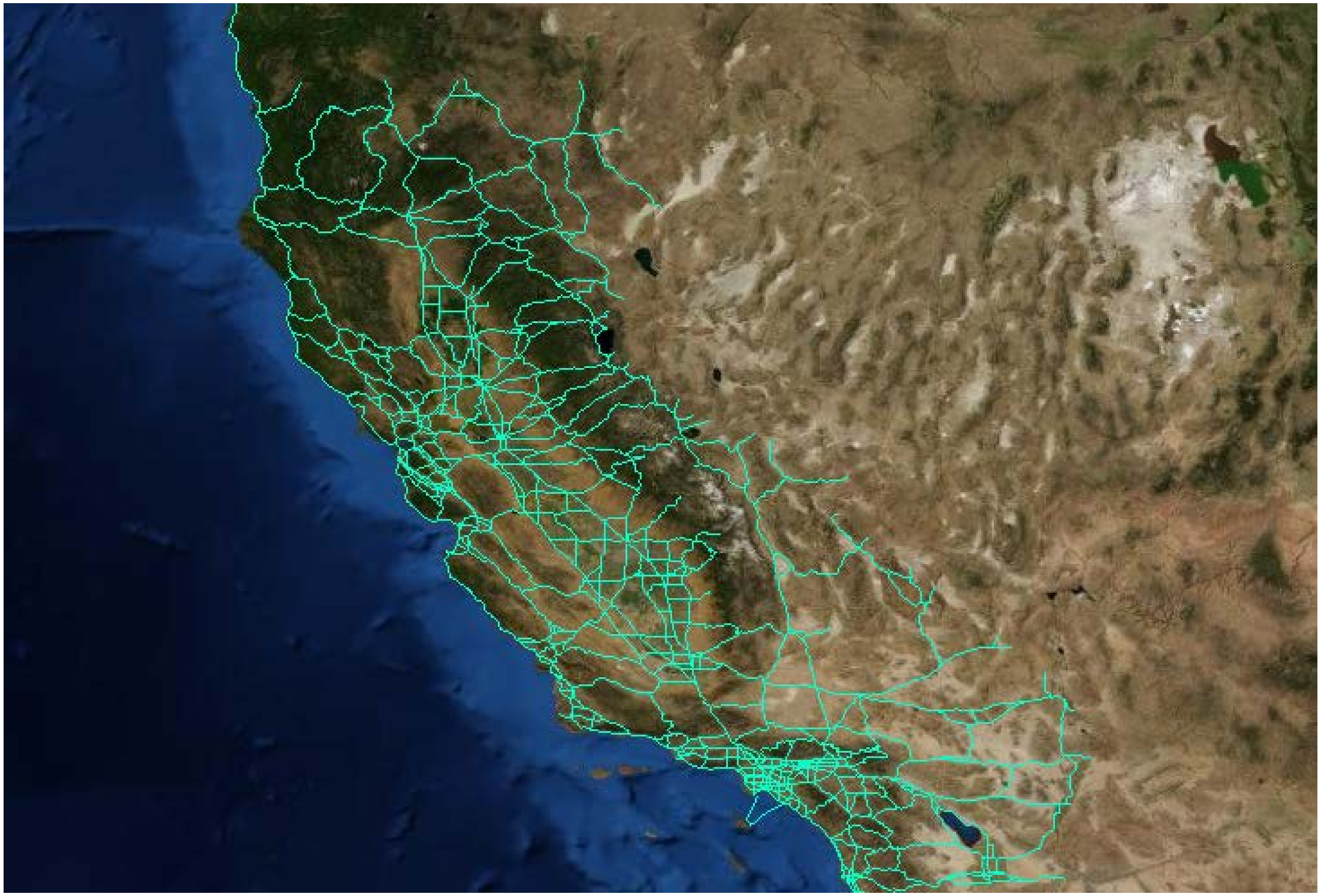
The layer name appears in the Preview List pane.

4. Check the box next to `california_roads_line`.

   The highway and road lines appear on California. (It appears below in bright green; it might appear in a different color on your screen.)



5. Right-click **california_roads_line**, and select **Zoom to Layer** to zoom into the San Francisco Bay area to view the road features.



6. Make sure the layer is selected (highlighted) in the Preview List pane. Then, with  selected on the toolbar, drag a selector rectangle around the City of San Francisco.

   The selected area is highlighted (yellow), and the data fields that correspond to the selected area appear in the Data List pane.

   You can scroll through this data and sort it by columns to explore the values of each field to determine the potential attributes to use in the filters you set up in *Defining and Building Projects*.

7. When you finish exploring the data, right-click the **california_roads_line**, and select **Remove Layer** to clear the Preview panes.

A message prompts you to confirm that you want to remove the layer.

8. Click **OK**.

   The layer disappears from the Preview panes.

9. Press **Ctrl-R** to reset the view to the whole Earth.

Define a Vector Resource

In this exercise, you define vector resources from the vector data provided for this tutorial.

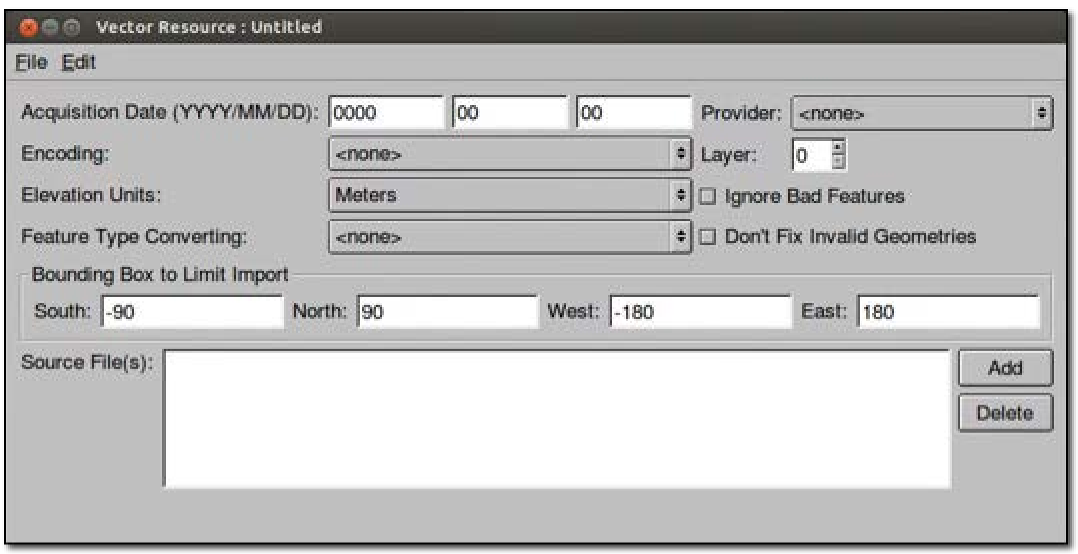
To define a vector resource:

1. Select **Asset Manager** from the Tools menu.

   The Asset Manager appears.

2. Click  on the toolbar.

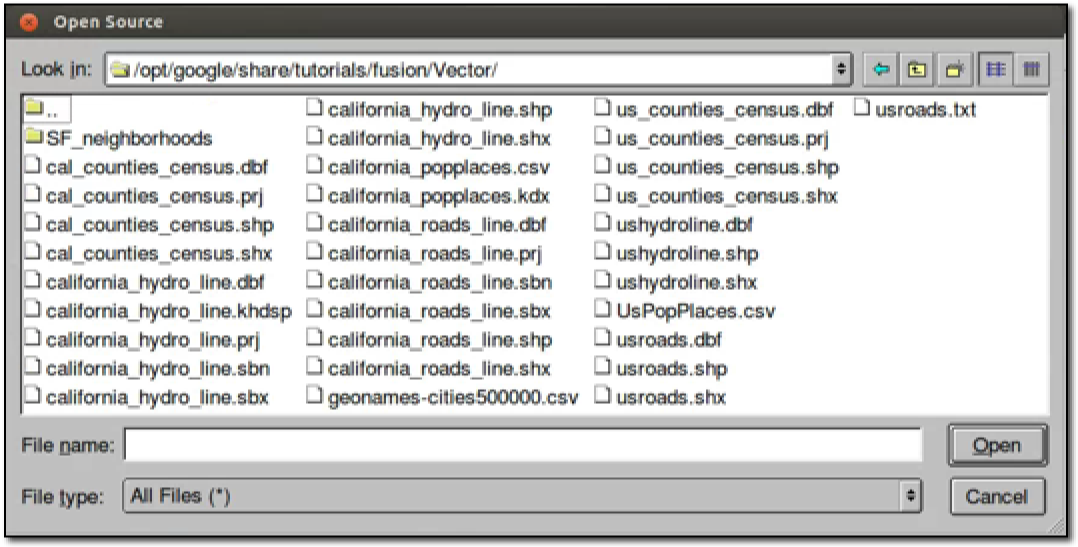
   The Vector Resource Editor appears.

3. Set the acquisition date to today's date in year-month-day format by clicking each section of the date and enter the values.

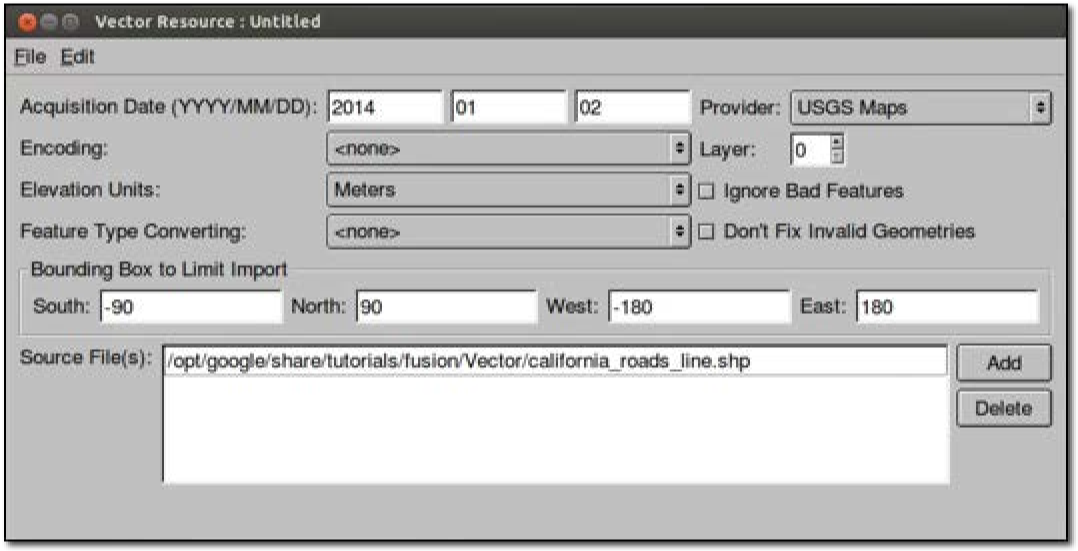
4. Select **USGS Maps** from the Provider drop-down list.
5. Click **Add**.

   The Open Source dialog opens to the `/opt/google/share/tutorials/fusion/Vector` folder.



6. Select the `california_roads_line.shp` file, and click **Open**.

   The selected file appears on the Source File(s) list.



7. Select **Save** from the File menu.

   The Save dialog appears.

8. Navigate to the `/ASSET_ROOT/Resources/Vector` folder you created in *Setting Up the Tutorial*.

9. Enter the name **CAHighways** for the resource, and click **Save**.

10. Close the Vector Resource Editor dialog.

    The name of the resource appears on the right when you select the `/ASSET_ROOT/Resources/Vector` folder in the asset navigation tree in **Asset Manager**.

Build a Vector Resource

As with the imagery and terrain resources, in this exercise, you build the vector resource right away. In fact, you must build vector resources before you can include them in projects.

To build a vector resource:

1. In the Asset Manager, select the `/ASSET_ROOT/Resources/Vector` folder.

   CAHighways appears on the right with the Current Version and the Current State set to **None**, indicating that the resource has not yet been built.

2. Right-click **CAHighways**, and select **Build** from the context menu.

   The status of the resource changes to **Queued** and then to **In Progress**. When the CAHighways resource finishes building, its Current State column in the Asset Manager changes to **Succeeded**, and its Current Version column changes to the date and time the most recent build was started.

Define and Build the Remaining Vector Resources

Now that you have defined and built one vector resource, you can define and build the remaining vector resources provided for this tutorial.

Follow the steps in Define a Vector Resource to define resources for the following vector source files:

- **CAPopPlaces**
  - Acquisition Date: today's date
  - Provider: **USGS POIs**
  - Source File: `california_popplaces.csv`
  - Name: **CA _POIs**
- **USCensusbyCounty**
  - Acquisition Date: today's date
  - Provider: **GNIS/US Census Bureau**
  - Source File: `us_counties_census.shp`
  - Name: **US_Population**

After defining each resource, right-click it, and select **Build** from the context menu. By the time you finish defining the last resource, the other builds should all be complete.

When Google Earth Enterprise Fusion finishes building the last resource, close the Asset Manager by clicking the close box (**X**) in the top right corner, and go on to the next lesson.

Back to top

Defining and Building Projects

The following exercises guide you through defining and building imagery, terrain, and vector projects, using the resources you created in the previous lesson.

About This Tutorial

Setting Up the Tutorial

Defining and Building Resources

Defining and Building Projects

Defining and Building Databases

Pushing, Publishing, and Viewing a Database

Configuring Display Rules for Point Data

Configuring Display Rules for Polygon Data

Importing and Exporting Style Templates

Specifying Search Fields for Individual Layers

Configuring a Searchable Database

Creating a Map Database

Creating Imagery Mosaics

Segmenting Large Imagery Files

Define an Imagery Project

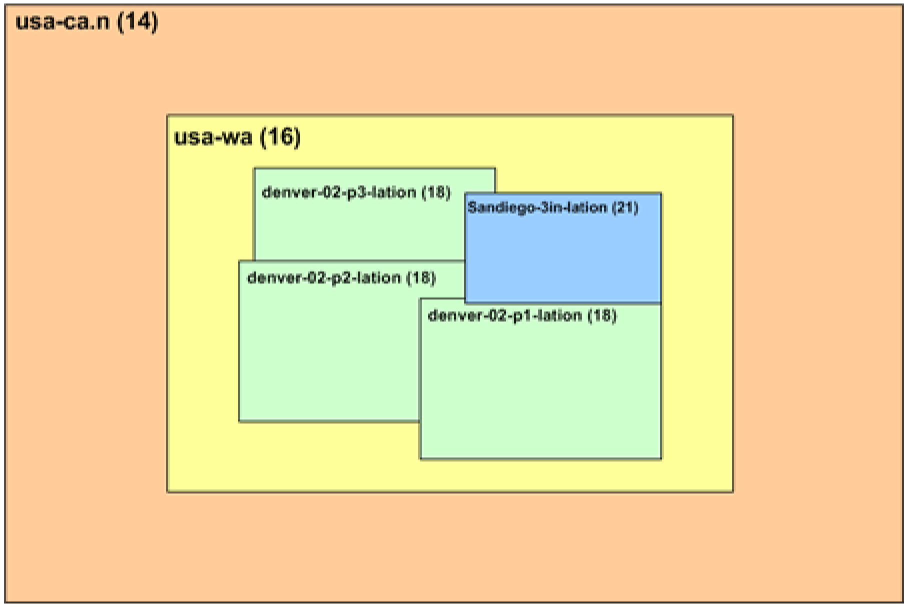
If you intend to use your own imagery data, it makes sense to define and build the imagery project before adding vector data. That way, you can use the imagery project as the base image map in the Preview pane in Google Earth Enterprise Fusion when you develop your vector project, making it easier to visualize how your vector data will appear over your actual imagery.

Add Resources to an Imagery Project

Although you can change the display order of imagery and terrain resources within a project, the order is ultimately determined by the resolution of the source files. That is, lower-resolution insets are automatically ordered below higher-resolution insets. So in reality, you can change the order of resources with the same resolution only.

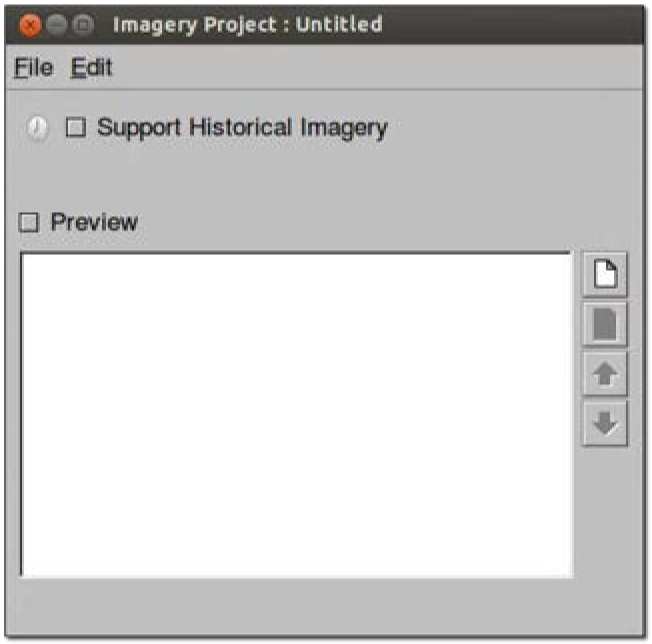
The following example shows a number of imagery resources in a project ordered by resolution. The resolution of each resource appears in parentheses after the resource name.

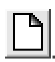
The order in which the imagery or terrain resource data appears in the Imagery and Terrain Project Editors is the same as the stacking order of the insets in Google Earth EC. That is, higher-resolution insets appear above lower-resolution insets, so that viewing preference is given to the higher-quality imagery. The stacking order of same-resolution insets follows the order you define in the project. The following graphic illustrates this concept.



To add resources to an imagery project:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Click  on the toolbar. The Imagery Project Editor appears.



3. Click . The Open dialog appears.

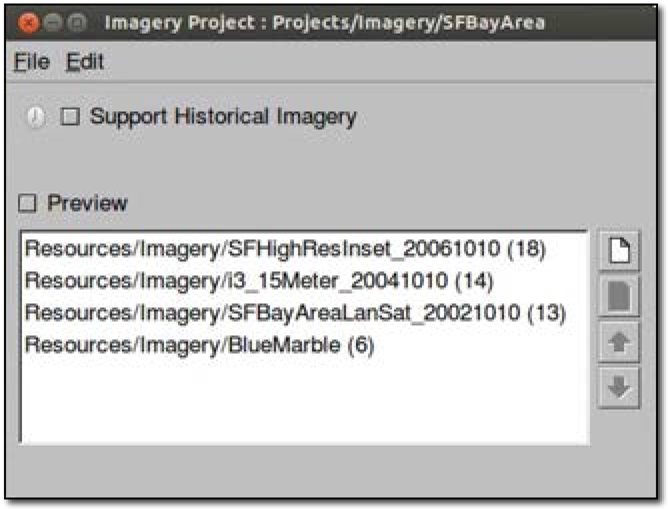
4. Navigate to the `ASSET_ROOT/Resources/Imagery` folder.

5. Select **BlueMarble**, and click **Open**. The BlueMarble resource appears in the Imagery Project Editor.

6. Repeat steps 4 through 6 for each of the following resources:

   - **SFBayAreaLanSat**
   - **i3_15Meter**
   - **SFHighResInset**

   The resources appear in order by resolution, with the higher resolution imagery at the top of the list.



7. Select **File > Save**.

8. Navigate to the `ASSET_ROOT/Projects/Imagery` folder.

9. Enter **SFBayArea** as the name of your project, and click **Save**. The new project appears in the Asset Manager when you select **ASSET_ROOT/Projects/Imagery** in the asset navigation tree.

Build an Imagery Project

As with resources, you do not have to build projects right away. You can define several projects and then build them all, or you can wait until you include them in a database and build the entire data hierarchy at the same time. However, in this lesson, you build the imagery project right away, so you can use it as a base map in the Preview pane.

Note: It could take up to 10 minutes for the project to build. Do this exercise when you can spend some time away from your computer while the project builds.

---

1. In the Asset Manager, navigate to **ASSET_ROOT/Projects/Imagery** in the asset navigation tree.

2. Right-click **SFBayArea** and select **Build** from the context menu.

   The status of the project immediately changes to **Queued** and then **In Progress**. (Sometimes the status changes so rapidly that it appears to change directly to **In Progress**.)

As with building imagery resources, you can view the progress of the build by double-clicking the Current Version or Current State column for the project. The Version Properties dialog displays the most recent version of that project. You can expand the version tree to view the status of the build in real time by clicking the + signs.

When the status of the imagery project build is **Succeeded**, go on to the next exercise.

Preview an Imagery Project

You can preview an imagery project after you build it.

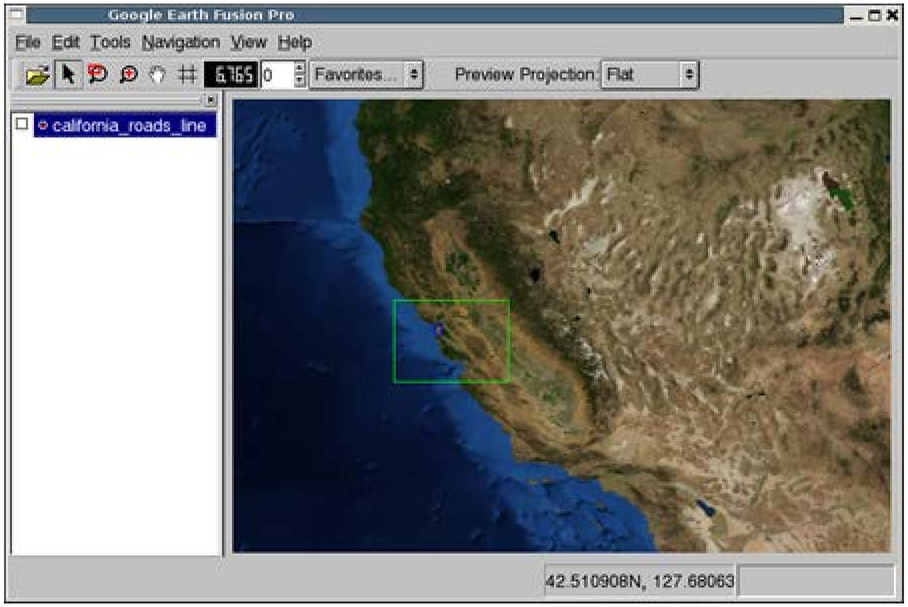
Note: Only bounding boxes appear in the Preview pane, not the actual imagery.

To preview an imagery project:

1. Double-click the name of the imagery project you built in the last exercise, **SFBayArea**. The Imagery Project Editor for that project opens.
2. Check the box next to **Preview** above the list of source files, and then switch to the main Google Earth Enterprise Fusion window.

   Bounding boxes indicate the extents of each of the individual imagery resources. The largest bounding box is around the entire Earth, which is the BlueMarble imagery. (You might have to zoom out a bit to see it.) The other three bounding boxes are in the San Francisco Bay Area. You can zoom in to see them.

   Note: The name of the project does not appear in the Preview List pane. When you close the Imagery Project Editor or uncheck the box next to **Preview**, the bounding boxes disappear. You can reset the view by pressing **Ctrl-R**.



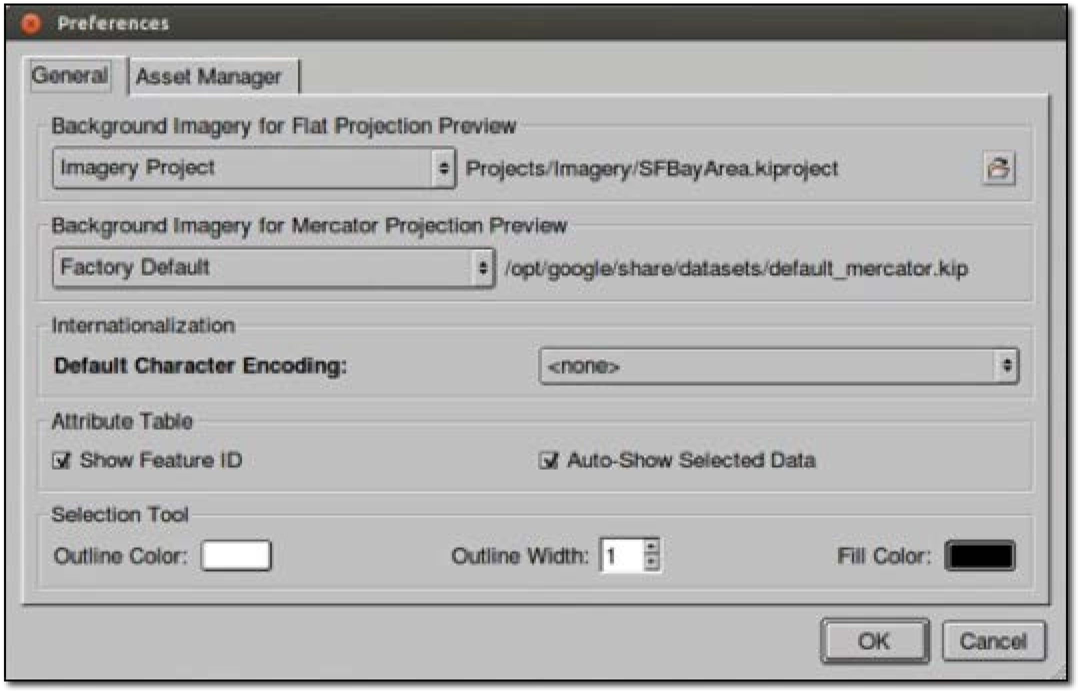
Specify an Imagery Project as Your Base Map

When the imagery project build is done, you can specify it as your base map.

To specify an imagery project as your base map:

1. Select **Preferences** from the Edit menu. The Preferences dialog appears.
2. Under Background Imagery, select **Imagery Project** from the drop-down list.
3. Click  to the right of the drop-down list. The Open dialog appears.
4. Navigate to `ASSET_ROOT/Projects/Imagery`, and select **SFBayArea**.

   The full path within the tutorial asset root appears to the right of the drop-down list.



5. Click **OK**. The Preview pane displays the specified imagery. The specified imagery will remain the background imagery until you change it to another image or return it to the default imagery in the Preferences dialog.

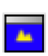
Define a Terrain Project

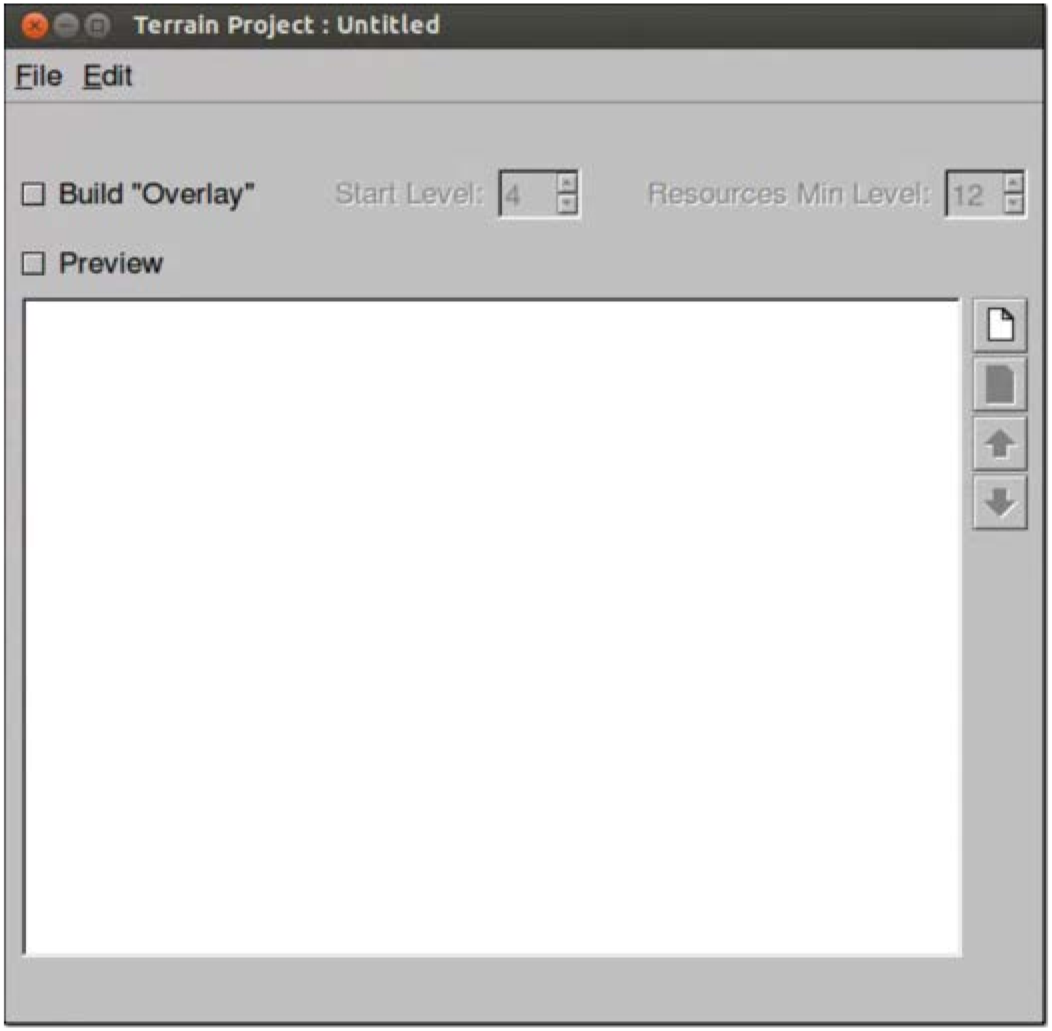
The terrain project for this tutorial is very simple. It includes one of the resources you built in the previous lesson.

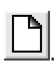
Add a Resource to a Terrain Project

The following procedure provides the steps to add resources to a terrain project.

To add resources to a terrain project:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Click  on the toolbar. The **Terrain Project** editor appears.



3. Click . The Open dialog appears.

4. Navigate to the `ASSET_ROOT/Resources/Terrain` folder.

5. Select **SFTerrain**, and click **Open**. The SFTerrain resource appears in the Terrain Project Editor.

Note: For a real terrain project, you should have worldwide coverage. However, a project with worldwide coverage would take too long to build, so for the purpose of this lesson, you use a smaller terrain image.

---

6. Check the box next to **Preview**.
7. Right-click the name of the resource, and select **Zoom to Layer** from the context menu.

   The Preview pane zooms in to the bounding box that indicates the extents of the terrain resource.

8. In the Terrain Project Editor, select **File > Save**.
9. Navigate to the `ASSET_ROOT/Projects/Terrain` folder. Enter **SFTerrain** as the name of your project, and click **Save**.
10. The new project appears in the Asset Manager when you select **ASSET_ROOT/Projects/Terrain** in the asset navigation tree.

Build a Terrain Project

As with the imagery project, in this exercise, you build the terrain project right away.

1. In the Asset Manager, right-click **SF Terrain**.
2. Select **Build** from the context menu.

   The status of the project immediately changes to **Queued** and then **In Progress**.

Note: It could take a while for this project to build, depending on the speed of your workstation.

When the status of the terrain project build is **Succeeded**, go on to the next exercise.

Define a Vector Project

The following exercises cover how to define, configure, and build a vector project using the resources you created in the previous lesson.

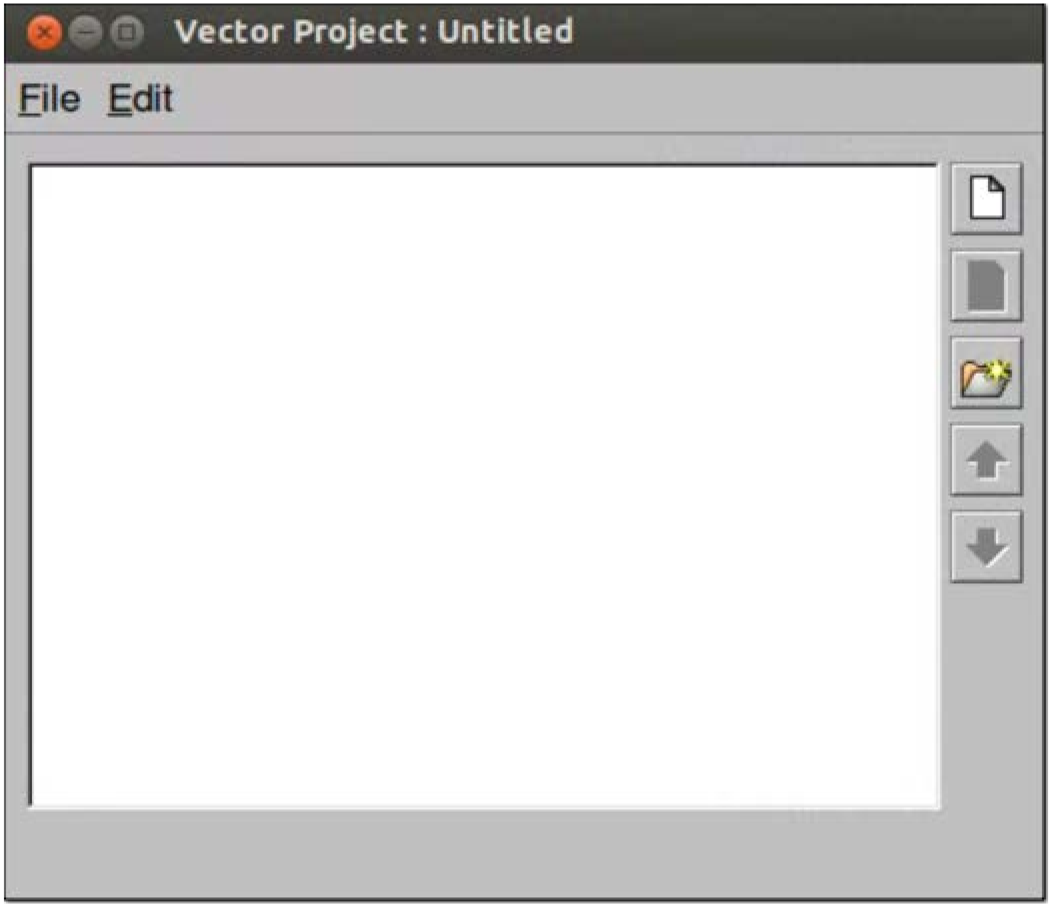
You spend the majority of your time in Google Earth Enterprise Fusion configuring display rules for vector projects, determining how they look in Google Earth EC. Using the data in your vector source files, you can designate specific data for a variety of display purposes, such as road labels, features lines, and icons at viewing altitudes that are most appropriate for each feature. For example:

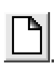
Add Resources to a Vector Project

Before specifying the display rules for this vector project, you must add the resource you created in *Defining and Building Resources*.

To add resources to a vector project:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Click  on the toolbar. The Vector Project Editor appears.



3. Click . The Open dialog appears.
4. Navigate to the `ASSET_ROOT/Resources/Vector` folder.
5. Select **CAHighways**, and click **Open**. The CAHighways resource appears in the Vector Project Editor.

6. Repeat steps 3 through 5 to add **USPopulation** to the project. Notice that a check box appears next to each resource/layer in the project.
7. Check the box next to **CAHighways**.
8. Right-click **CAHighways**, and select **Zoom to Layer** from the context menu.

   The roads in the CAHighways resource appear in the Preview pane.

9. Check the box next to **USPopulation**, switch to the Preview pane, and zoom out a bit.

   Since this resource contains US census data by county, the outlines of counties across the US appear in the Preview pane as well as the roads in California.

Note: The name of the project does not appear in the Preview List pane. When you close the Vector Project Editor or uncheck the boxes next to the resources, the vector data disappears. You can reset the view, if desired, by pressing **Ctrl-R**.

10. In the Vector Project Editor, select (highlight) **USPopulation**, and click  to remove the US Population resource from the project. A message prompts you to confirm the deletion.
11. Click **OK**. The **USPopulation** disappears.

Note: Removing the resource from the project does not delete the resource. The resource remains intact and available for use by other projects. It is just not part of this particular project.

12. Select **File > Save**.
13. Navigate to the `ASSET_ROOT/Projects/Vector` folder.
14. Enter **CARoads** as the name of your project, and click **Save**. The new project appears in the Asset Manager when you select **ASSET_ROOT/Projects/Vector** in the asset navigation tree.

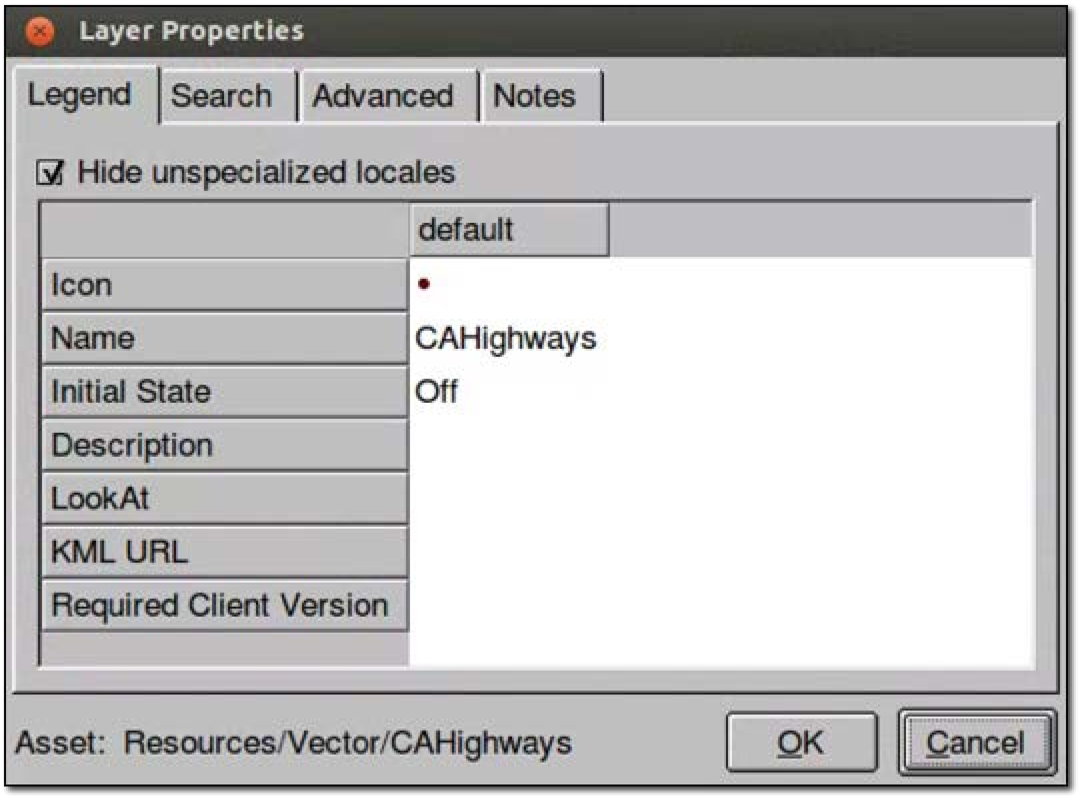
Now you are ready to begin configuring the vector layer.

Configuring Layer Properties for a Vector Project

This exercise covers how to configure layer properties for your vector project. Layer properties determine a number of aspects of how your data appears and is accessed in Google Earth EC.

To configure layer properties:

1. In the Asset Manager, double-click **CARoads** in the `/ASSET_ROOT/Projects/Vector` folder. The Vector Project Editor appears and displays the resource you added in the previous lesson.
2. Right-click **CAHighways**, and select **Layer Properties** from the context menu. The Layer Properties dialog appears.



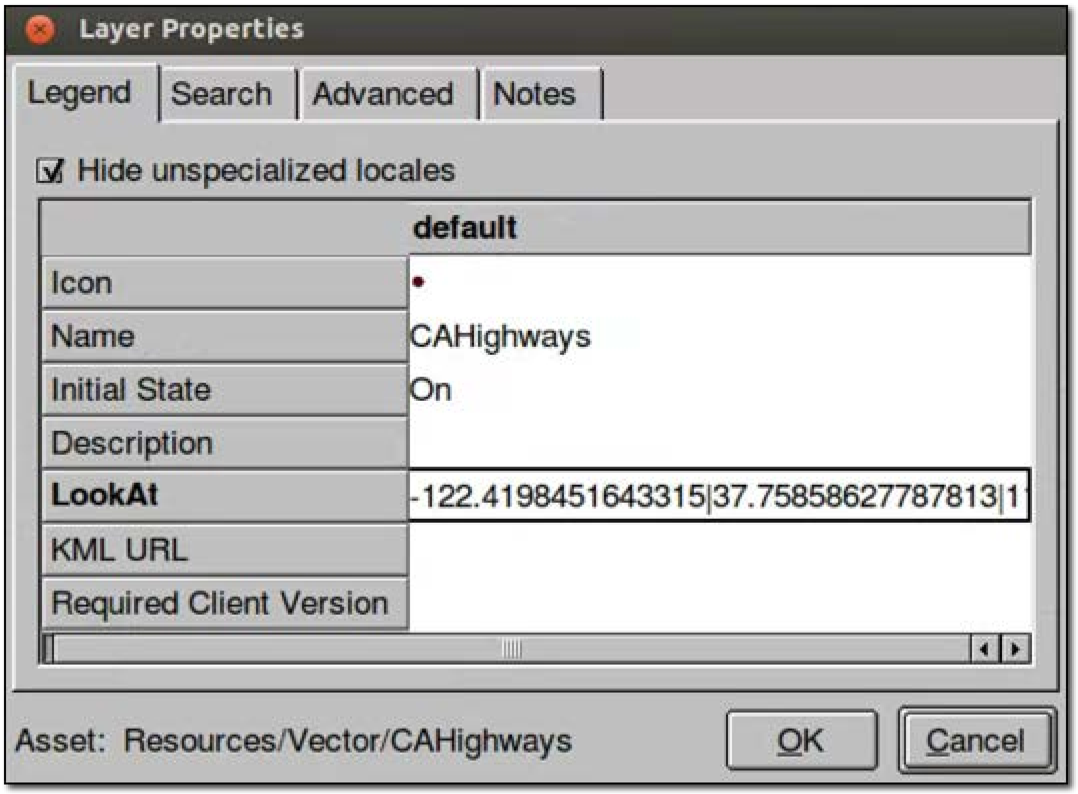
3. Click **Off** next to Initial State to change it to **On**. Changing the initial state to **On** results in the CAHighways layer being automatically checked (turned on) in Google Earth EC.

4. Click the blank field next to **LookAt**. The Open dialog appears.

5. Navigate to `/opt/google/share/tutorials/fusion/KML`, select `San Francisco View Oblique.kmz`, and click **Open**.

   When you specify a KML/KMZ file in this field, Google Earth EC users can fly directly to the specified camera view by double-clicking the layer.

   The latitude and longitude of the selected KMZ file appear in the LookAt field. (You can only see the beginning of the latitude unless you expand the default column width.)



6. Click **OK**. You return to the Vector Project Editor.

7. Select **File > Save**. Google Earth Enterprise Fusion saves the vector project with the same name.

Configure Display Rules for a Vector Project

This exercise covers how to specify display rules for your vector project. Display rules determine how your data looks in Google Earth EC.

Each resource in a project is known as a *layer*. Each layer can have one set of display rules. Each display rule includes feature and label formatting that you specify and one or more filters for the selected layer. The filters for each display rule determine which data in the layer to apply that formatting to. For example, in this exercise, you work with road data that includes major and minor highways. You might want the major highways to appear as blue lines and the minor highways as yellow lines. To accomplish that, you create two display rules for the layer--one for major highways and one for minor highways. The filter(s) for each rule determine which data are affected by that rule's formatting specifications.

In this exercise, you create and modify a number of display rules for the CAHighways layer, so Google Earth EC displays the road information at the desired display levels with appropriate labels and coloring to distinguish between major freeways and other roads.

A key part of knowing how to configure display rules for vector data requires familiarity with the data fields in the source data. For the CAHighways layer of your vector project in this tutorial, you can refer to the description of the data fields at this web site:

`http://www.nationalatlas.gov/mld/roadtrl.html`

To configure a vector layer:

1. In the Vector Project Editor, right-click **CAHighways**, and select **Configure Display Rules** from the context menu.

   The Display Rules dialog appears with the Feature tab in the foreground and the **default select all** rule highlighted.

When you first create a vector project, the default display rule--**default select all--**is the only rule listed for each layer. The filter for the default rule has no matching criteria, so it matches all data. This rule is considered the *catch-all* rule. It is designed to catch all of the data that does not match any other rules you create.

Tip: Google Earth Enterprise Fusion executes the display rules sequentially, based on the order in which they are listed on the Rules list. So you should always make the **default select all** rule the last one on the list.

First, Google Earth Enterprise Fusion attempts to match the filter specified for the first rule to the data in the resource. Then it applies the formatting specified for that rule to any matching data.

Next, it attempts to match the filter specified for the second rule to the remaining data in the resource (that is, data not selected for the first filter). It applies the formatting specified for that rule to any matching data.

Then, it applies the formatting you specify for the **default select all** rule (or the default formatting, if you do not change it) to any data that does not match the previous rules on the list, if any. This ensures that all vector data for the layer is displayed.

Configure the Default Select All Rule

In this part of the exercise, you define the **default select all** rule. This rule applies to all of the surface streets that are left after the highways and freeways are covered by other rules. You define the additional rules for highways and freeways later in this exercise, because they are based on this default rule.

To define the default select all rule:

1. Specify the geometry characteristics of the lines:
    a. For Draw Features As, select **Lines**.
    b. For Visibility, set the lower end of the range to **9**, and the upper end of the range to **18**.
    c. For Maximum Simplification Error, accept the default setting, **0.5**.
    d. For Maximum Resolution Level, select **18**. This sets visibility level and the point where the Google Earth Enterprise Fusion stops building the resource.
2. Check the box next to **Draw As Roads**, and specify the options related to road labels:
    a. For Road Label Type, accept the default setting, **Label**.
    b. Under Draw Style, for Line Color, set it to bright red.
    c. For Line Width, enter **2.0**.
    d. For Elevation/Height Mode, accept the default setting, **Clamp to Ground**.

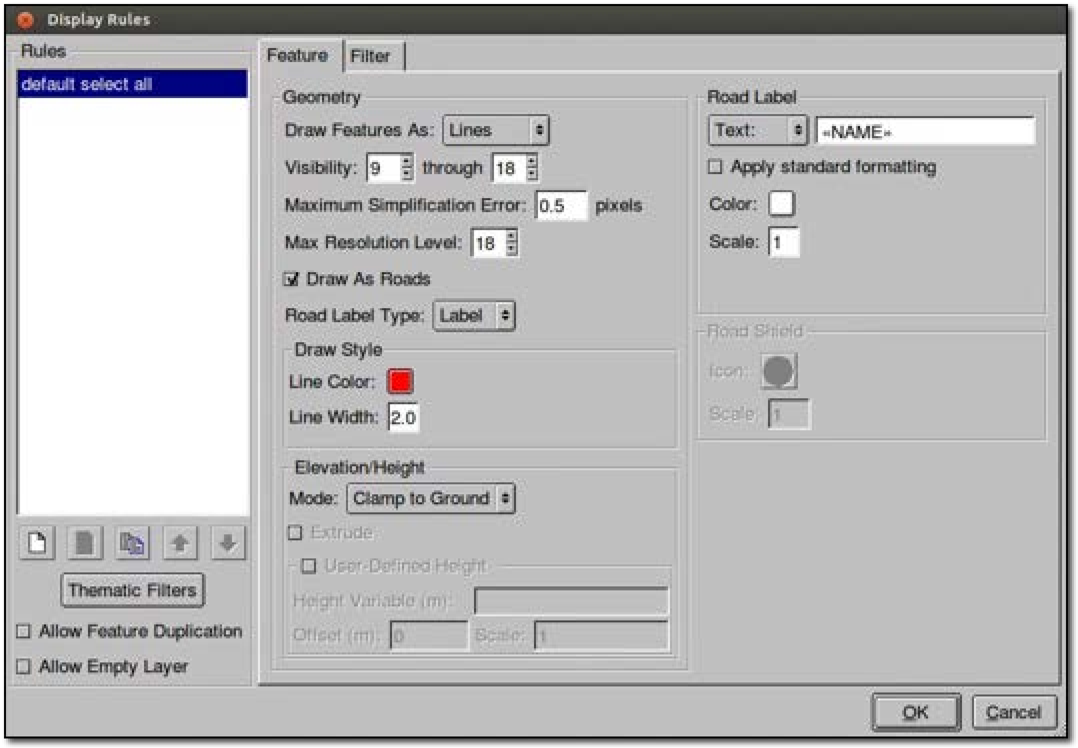
    These settings display the roads as red lines when Google Earth EC is zoomed in fairly close.
3. In the **Road Label** section on the right:

a. Click the empty text field. The Label Format dialog appears. This option allows you to specify the text that appears on the label.

b. Click the Insert Field drop-down list to display the names of all of the fields in your source data.

c. Select **NAME** from the list. The string «**NAME**» appears in the text field.

d. Click **OK**. The string «**NAME**» appears in the Text field. The Display Rule dialog shows all of your selections. These settings result in Google Earth EC displaying the value of the NAME column for each road in your source data.



e. Click **OK** to save your display rule.

4. Verify that your display rule does what you intend:

   a. Ensure that nothing is listed in the Preview List pane. If one or more assets are listed, right-click any asset, and select **Remove All Layers** from the context menu; then click **OK** to confirm the removal.

   b. In the Vector Project Editor, check the box next to **CAHighways**.

c. Right-click **CAHighways**, and select **Zoom to Layer** from the context menu.

   No roads appear because your display level is approximately 7, and you set the visibility level to between **9** and **18** in the Display Rules dialog.

d. Zoom in to a display level of just over **9**. Red lines appear for the roads in the Preview pane.

   Note: Labels for vector projects do not appear in the Preview pane.

   ---

e. Zoom out to a display level less than **9**. The roads disappear from the Preview pane.

5. Save the vector project by selecting **Save** from the File menu in the Vector Project Editor.

   This saves the project with the same name. If you want to save a project you create outside this tutorial with a different name, you can select **Save As** from the File menu, and follow the instructions in Add Resources to a Vector Project.

   Note: Whenever you modify display rules or filters for your data, it is a good idea to save the project.

Display Rules for Major Freeways

This exercise guides you through creating the display rules necessary to achieve the desired appearance for the major freeways in the San Francisco Bay Area. When you finish this exercise, you should have a good understanding of the use of filters in managing complex data.

When setting display rules for vector data, it is critical that you are familiar with the source data you are working with and have an understanding of the fields used to classify different types of vector data. In the source data for this tutorial, the FEATURE column sorts the roads and highways into the following types:

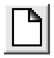
- Principal Highway
- Other Through Highway
- Other Highway
- Limited Access Highway

In this exercise, you create a second filter and use the FEATURE column values to distinguish the limited access highways (major freeways) from other types of highways and roads, and then display them appropriately.

To define a display rule for major freeways:

1. In the Vector Project Editor, right-click **CAHighways**, and select **Configure Display Rules** from the context menu.

   The Display Rules dialog appears with the Feature tab in the foreground and the **default select all** rule highlighted.

2. Click  at the bottom-left of the dialog. The New Rule dialog appears.

3. Enter **Major Freeways** in the New Rule Name field, and click **OK**. The new rule name appears on the Rules list below the **default select all** rule.

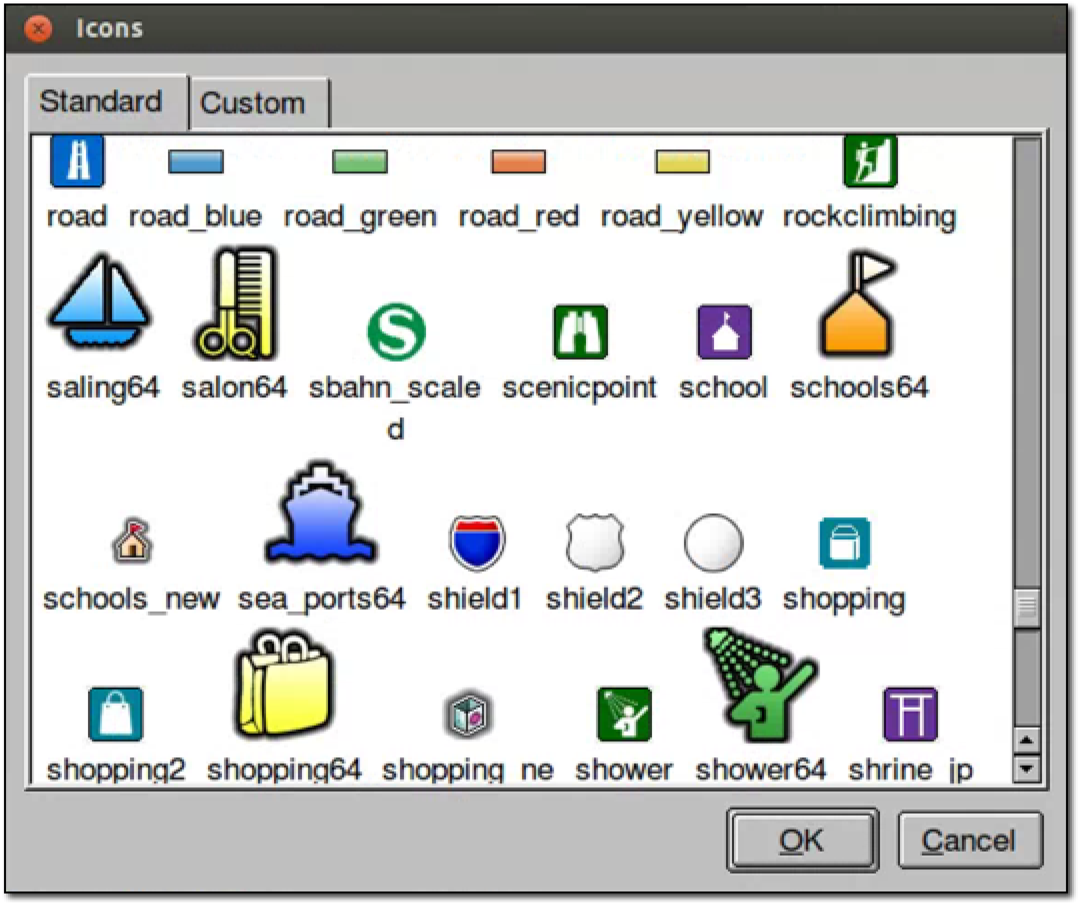
   

4. Click  to move the new rule up, so it appears before the **default select all** rule.

   Because Google Earth Enterprise Fusion applies display rules in the order in which they appear on this list, the MajorFreeways rule must appear before the *catch-all* rule (**default select all**), which covers all data not covered by other rules.

5. On the Feature tab, specify the geometry characteristics for the Major Freeways rule:
    a. For Draw Features As, select **Lines**.
    b. For Visibility, set the low end of the range to **9** and the high end of the range to **18**.
    c. For Maximum Simplification Error, accept the default setting, **0.5**.
6. Check the box next to **Draw as Roads** (if it is not already checked).
    a. For Road Label Type, select **Shield**.
    b. Under Draw Style, for Line Color, set it to royal blue.
    c. For Line Width, enter **3.0**.
    d. Under Elevation/Height, for Mode, accept the default, **Clamp to Ground**.

        These settings display the roads as thick blue lines when Google Earth EC is zoomed in fairly close.
7. In the Road Label section, accept the current settings.
8. In the Road Shield section on the right, click the button next to **Icon**. The Icons dialog appears.



9. Scroll down, and select **shield1**, and click **OK**.

   The Display Rule dialog shows all of your selections.

10. Click the **Filter** tab, and specify the filter for the Major Freeways rule:

    a. Accept the default selection, **Match all of the following**, at the top of the tab.

    b. Click **More** at the bottom of the tab.

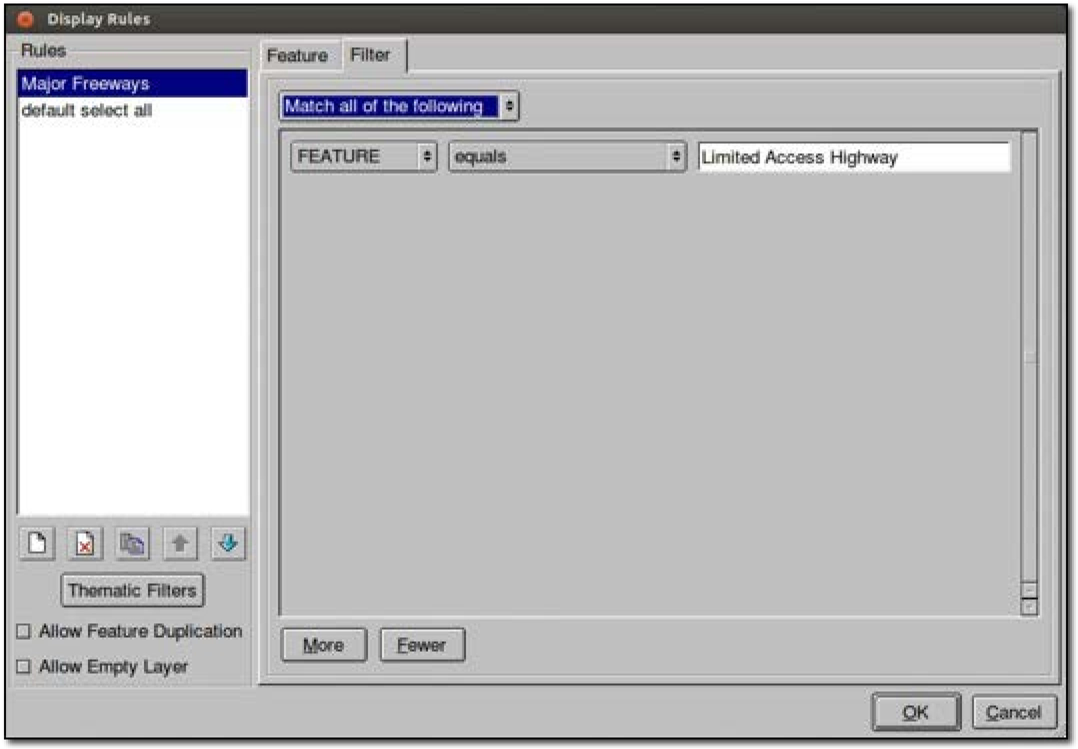
       Two drop-down lists and a text box appear on the list of filters.

    c. Select **FEATURE** from the left drop-down list.

    d. Select **equals** from the other drop-down list.

e. Enter **Limited Access Highway** in the text field.

The Filter tab shows your selections.



f. Click **OK** to save your changes to the display rule.

When this data is displayed in Google Earth EC, this filter causes the display settings on the Feature tab to be applied only to the road segments in your source data with the value **Limited Access Highway** in their FEATURE column.

11. Verify that both filters and rules are working correctly:

a. In the Vector Project Editor, check the box next to **CAHighways**, if it is not already checked.

b. Zoom in to a display level between **9** and **18**, if necessary.

Both thin red roads and thicker blue roads appear in the Preview pane. The thicker blue roads are the roads defined in the MajorFreeways filter, and the thinner red roads are the rest of the roads in the source data, which are defined by the **default select all** filter.

12. Save the vector project by selecting **Save** from the File menu in the Vector Project Editor.

    This saves the project with the same name.

Build a Vector Project

As with imagery and terrain projects, in this exercise, you build the vector project as soon as you finish configuring display rules.

1. In the Asset Manager, right-click **CARoads**.
2. Select **Build** from the context menu.

   The status of the project immediately changes to **Queued** and then **In Progress**.

When the status of the vector project build is **Succeeded**, close the Asset Manager by clicking the close box (**X**) in the top right corner, and go on to the next lesson.

Back to top

Defining and Building Databases

The following exercises guide you through defining and building a database, using the projects you created in the previous lesson.

- Define a Database
- Build a Database

Define a Database

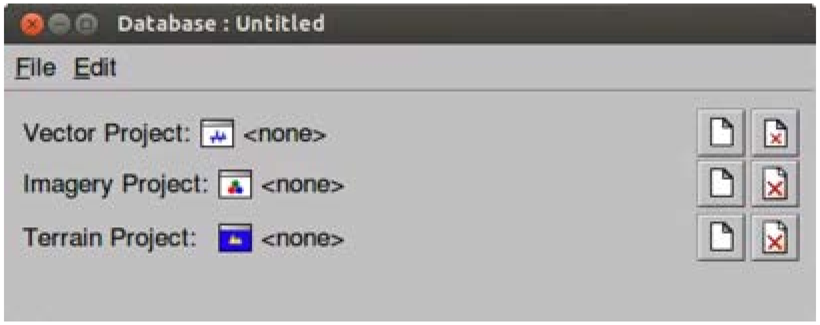
When you define a database, you specify one or more Google Earth Enterprise Fusion projects that Google Earth Enterprise Fusion combines into a single self-contained world that is flyable in Google Earth EC. You can select up to three projects for a database--one of each type:

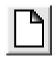
- Vector
- Imagery
- Terrain

Because the majority of your efforts involve defining and configuring different projects for inclusion in your database, it is relatively fast to define a database once the projects are created. You simply select the projects that comprise the database and give it a name.

To define a database:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Click the page icon on the toolbar:  The Database Editor appears with no projects selected.



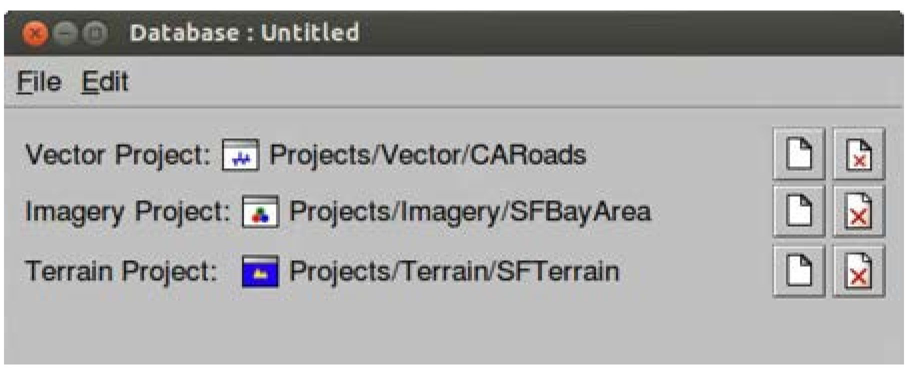
3. Click the page icon in the **Vector Project** row: 
4. Navigate to `ASSET_ROOT/Projects/Vector`.

Note: The selection in the **Type** drop-down list near the bottom of this dialog determines the type of projects that appear on the list. Vector Project is automatically selected, so only vector resources appear on the list.

---

5. Select the **CARoads** project and click **Open**. The CA Roads project appears in the Database Editor next to Vector Project.
6. Click the page icon in the **Imagery Project** row and select `Projects/Imagery/SFBayArea.`
7. Click the page icon in the **Terrain Project** row and select `Projects/Terrain/SFTerrain.`

All three projects are now listed in the Database Editor window:



8. Select **File > Save** and navigate to the `ASSET_ROOT/Databases` folder.

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

9. Enter **SFHighways** as the name of your database, and click **Save**.

   The name of the database is displayed when you select the `/ASSET_ROOT/Databases` folder in the asset navigation tree.

Build a Database

Once the database has been defined, it is ready to be built.

Note: If you had not built the projects included in this database previously, the following process would build them in the course of building the database. As you might expect, the database build process takes much longer if it is building all of the projects in the database at once. However, since Google Earth Enterprise Fusion gives you the option to build each project as soon as you finish making modifications or when you build the database, you can determine the best work style for yourself.

To build a database:

1. In the Asset Manager, select the `/ASSET_ROOT/Databases` folder.

   SFHighways appears on the right with the Current Version and the Current State set to **None**, indicating that the database has not yet been built.

2. Right-click **SFHighways** and select **Build**. The status of the database immediately changes to **Queued** and then to **In Progress**.
3. Double-click the Current Version or Current State column of the **SFHighways** to view the progress of the build.

   The Version Properties dialog displays the most recent version of that database. You can expand the version tree to view the status of the build in real time by clicking the **+** signs.

When Google Earth Enterprise Fusion finishes building the database, its Current State column in the Asset Manager changes to **Succeeded**, and its Current Version column changes to the date and time the most recent build was started.

Close the Asset Manager by clicking the close box (**X**) in the top right corner, and go on to the next lesson.

Back to top

Pushing, Publishing, and Viewing a Database

Pushing a database is the process of making a Google Earth Enterprise Fusion database available on a Google Earth Enterprise Server so that it can then be published for viewing with Google Earth EC.

The *push* operation copies all the necessary files associated with a given 2D/3D Fusion database version to Google Earth Enterprise Server and registers the database. Pushing is performed from Fusion.

The *publish* operation makes a previously pushed database available for serving at a specified publish point. Publishing is performed on Google Earth Enterprise Server.

Note: This lesson assumes that Google Earth Enterprise Fusion, the Google Earth Enterprise Server, and Google Earth EC are all installed on your local workstation. If any of these applications are installed elsewhere on your network, adjust the instructions in this lesson accordingly.

Note: This lesson describes the steps for pushing a 3D Fusion database. The same workflow is used to push a 2D Mercator database, but note that snippet profiles don't apply and that the published map can be viewed in a web browser, not Google Earth EC. See Creating a Map Database for more information.

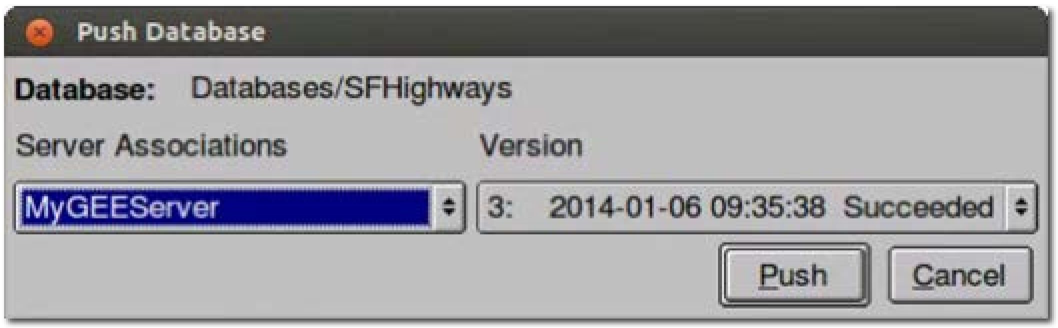
Push Your Database to Google Earth Enterprise Server

After you define and build your database, you “push” your database to Google Earth Enterprise Server. Pushed databases appear on the Databases page of the Google Earth Enterprise Server Admin console, from where you can then publish them.

To push a database:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Navigate to the **SFHighways** database you built in *Defining and Building Databases*.
3. Right-click the name of the database, and select **Push** from the context menu.

   The Push Database dialog appears. The default server association is the only option on the Server Associations drop-down list. The most recent version of the selected database is the default selection on the Version drop-down list.

4. Click **Push**. Google Earth Enterprise Fusion pushes the database to Google Earth Enterprise Server, and displays a success message when it is done.

   Note: If you get an error message, contact your Google Earth Enterprise Server administrator for help, or check the *Google Earth Enterprise Administration Guide* for more information.

Publish Your Database in Google Earth Enterprise Server

After you push your database to Google Earth Enterprise Server, you can publish it for viewing with Google Earth EC.

To publish a database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to

myserver.mydomainname.com/admin, replacing *myserver* and *mydomainname* with your server and domain.

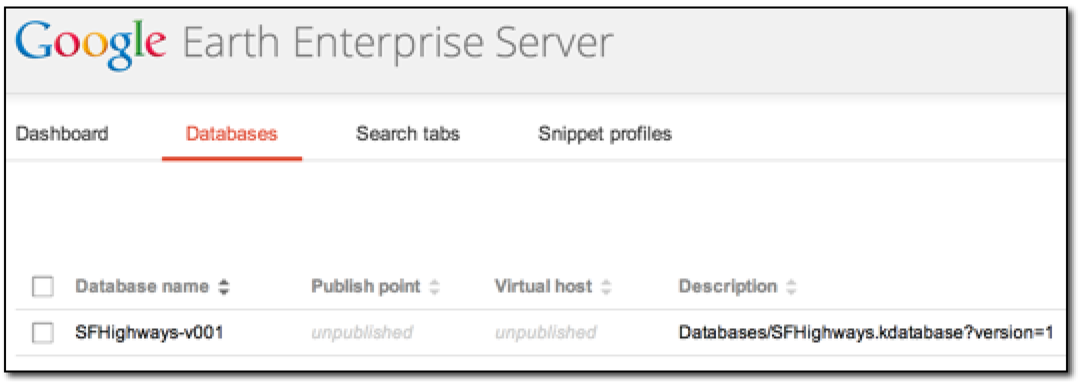
2. Sign in with the default credentials or the username and password assigned to you:
    - Default username: geapacheuser
    - Default password:geeadmin

Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.

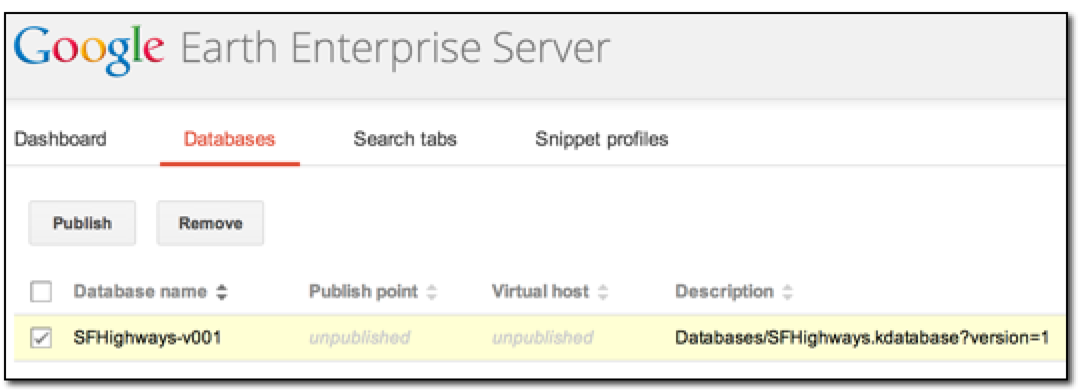
---

3. Click **Databases** to display the list of databases pushed to the Server.

   SFHighways-v001 is listed. The Description, Type, Date, and Size of the database are also noted.

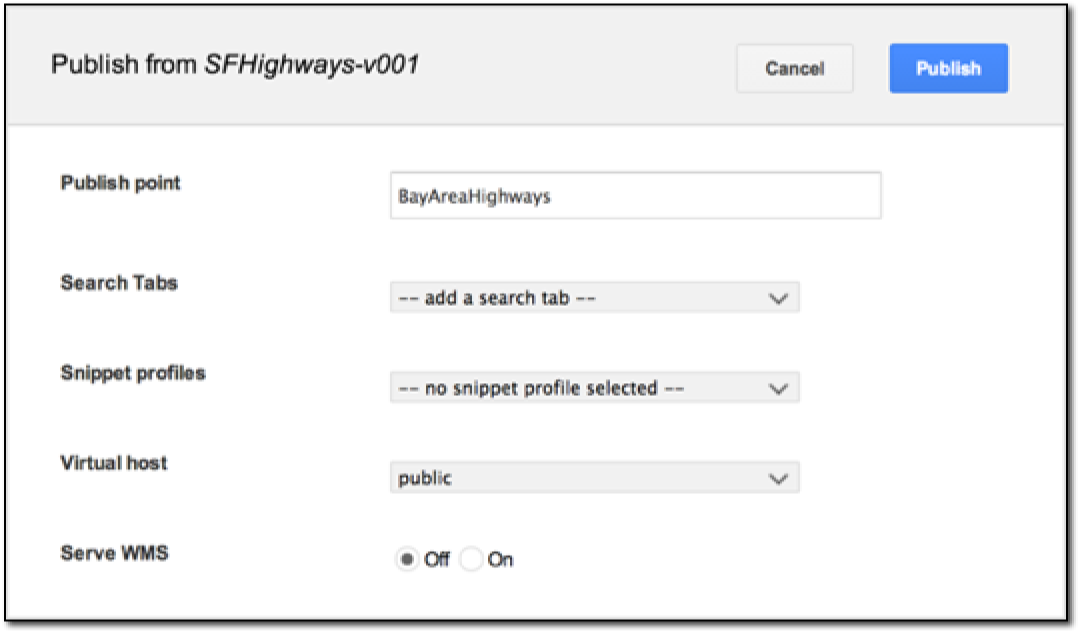


4. Check the box next to SFHighways-v001. The **Publish** button on the Databases page appears.



5. Click **Publish**. The Publish dialog appears.

6. Change the default Publish point from SFHighways-v001 to **BayAreaHighways**.

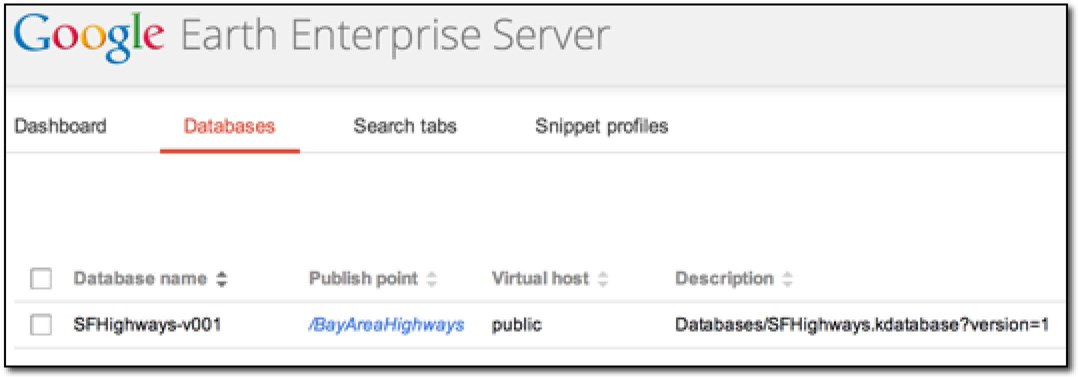


Note: You can publish a database to *multiple publish points*, useful when you want to apply different search tabs, snippet profiles, and virtual host settings for different viewing needs.

Note: When publishing a database, the publish point you specify is case *insensitive*. Upper and lower case are not differentiated. Make sure each publish point path name you specify is unique.

To learn more about the options available in the Publish dialog, see *Create Search Tabs*, *Snippet profiles*, and *WMS*.

7. Click **Publish**. The Databases page updates to indicate the published status of your database.



View Your Database

After you publish your database, you can view it in Google Earth EC.

To view your database:

1. Launch Google Earth EC.

   The Login dialog appears.

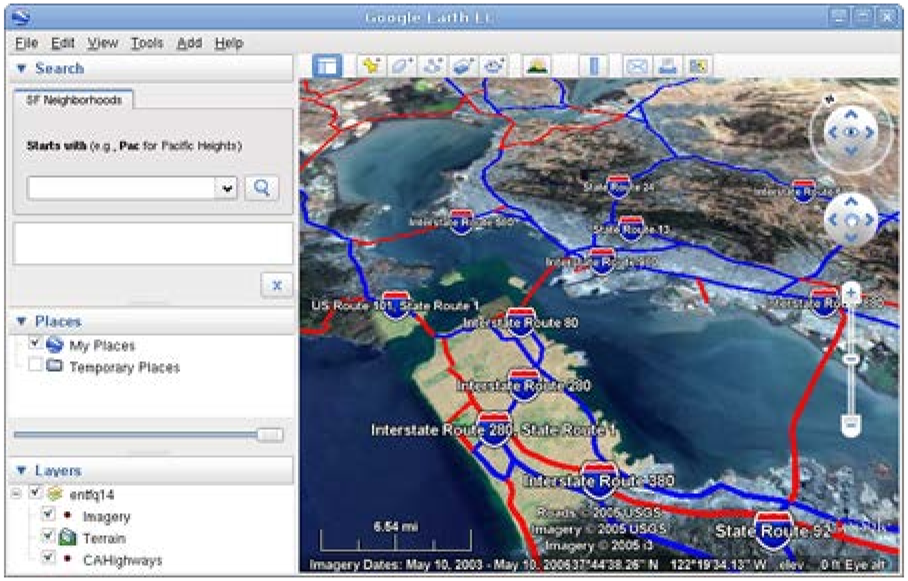
2. Specify the Publish point by entering or selecting the URL or IP address of your server and database in the Server field. For this tutorial, the Publish point is *myserver.mydomainname*.com/BayAreaHighways, where *myserver* and *mydomainname* are specific to your server. An example URL would be http://myhostname/BayAreaHighways.

3. Click **Sign In**.

   Caution: If you have logged in to this server with Google Earth EC previously, log out, clear your cache, and log back in. See clearing your cache.

   Google Earth EC displays your database. The Layers panel shows the terrain, imagery, and vector layers in the database you published in the previous exercise.

4. Double-click **CAHighways** to zoom in to your road data.



This concludes the basic lessons in this tutorial. If you want to continue with the advanced lessons, you can work through them in any order you want. Alternatively, you can start working with real data now and come back and work through the advanced lessons as you need them.

Back to top

Configuring Display Rules for Point Data

Point data can provide useful information about points of interest (POIs) around the globe or even in your home town. Google Earth Enterprise Fusion allows you to configure the display rules for point data to distinguish different types of information. This lesson guides you through the process of graphically distinguishing the popular places in California using display rules.

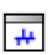
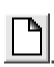
- Define, Configure, and Build a Vector Project
- Define and Build an Imagery Project
- Define, Build, Push, and Publish a Database for the Point Data
- View Your Database in Google Earth EC

Define, Configure, and Build a Vector Project

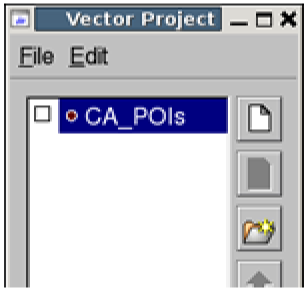
This exercise walks you through the process of defining, configuring, and building a vector project using a resource you created in *Defining and Building Resources*.

To define and build a vector project and configure display rules for points:

1. Select **Asset Manager** from the Tools menu.

   The Asset Manager appears.
2. Click  on the toolbar. The Vector Project Editor appears.
3. Click . The Open dialog appears.
4. Navigate to the `ASSET ROOT/Resources/Vector` folder.

5. Select **CA_POIs** (a resource that you built in *Defining and Building Resources*), and click **Open**. The CA_POIs resource appears in the Vector Project Editor.



6. Right-click **CA_POIs**, and select **Configure Display Rules** from the context menu.

   The Display Rules dialog appears with the Feature tab in the foreground and the **default select all** rule highlighted.

7. Specify the geometry characteristics of the POIs:

   a. For Draw Features As, select **Points**.

   b. For Simplification Method, select **Representative Subset Per Tile**.

   c. For Choose, accept the default, **75**% points.

   d. For Min Points, enter **50**.

   e. For Max Points, enter **200**.

   f. Check the box next to **Suppress Duplicate Points**.

   g. Under Elevation/Height, set Mode to **Clamp to Ground.**

      These settings depend on the type of data you are working with. You can try settings you think will work, publish the data, review the result in Google Earth EC, and then go back and adjust the settings to make the data look the way you want. (Refer to the *Google Earth Enterprise Fusion Reference Guide* for complete details about each of the settings in this dialog.)

8. On the right side of the dialog, check the box next to **Draw Label**.

9. For Visibility, accept the default range, **5** through **24**.

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

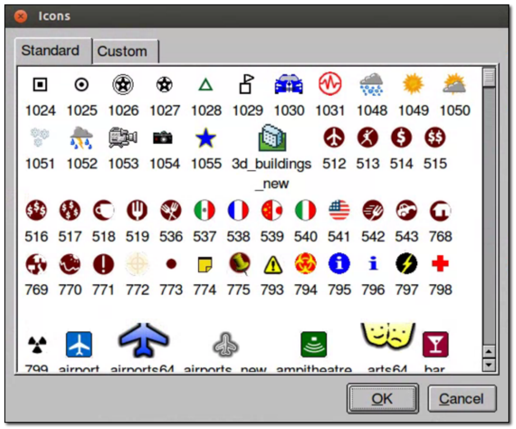
Manage mosaics with virtual rasters

10. Do not make any changes under Label Properties,

    This area allows you to specify the content and appearance of a text label for each POI. However, there are so many POIs in California that the view in Google Earth EC would be too crowded.

11. Check the box next to **Draw Icon** and then:

    a. Click the icon pair. The Icons dialog appears.



    b. Scroll down, if necessary, and select **blue_star**, and click **OK**.

    A yellow star appears next to Highlight, and a blue star appears next to Normal. The result of this setting is that Google Earth EC displays a blue star for each POI, and when you mouse over the star (highlight it), it turns yellow.

    c. For both Highlight and Normal, accept the default color, white.

    d. For Highlight, accept the default scale, **1**.

    e. For Normal, change the scale to **0.89**.

    The result of these settings is that when you mouse over a star, Google Earth EC displays the yellow (highlighted) star a little larger than the blue star.

12. In the Balloon section, select **Text**, and then click the empty text field. The Label Format dialog appears.

    a. Select **COUNTY** from the **Insert Field** drop-down list.

    «**COUNTY**» appears in the text box under Popup Text.

Note: You can use some basic HTML tags to format the Label text, if desired. (See Appendix C, "HTML Tags Allowed" in the *Google Earth Enterprise Fusion Reference Guide* for details.)

b. To the left of «**COUNTY**» enter ****.

c. To the right of «**COUNTY**» enter **

, and press **Enter.

Adding this HTML code formats the name of the county as bold, followed by two line breaks to separate it from the text that follows (next step).

d. Select **NAME** from the Insert Field drop-down list. «**NAME**» appears on the next line of the text box.

e. To the right of «**NAME**» enter **

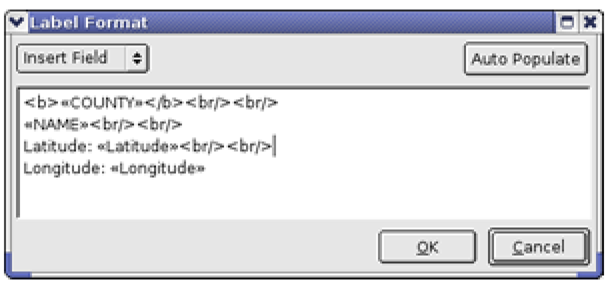
, and press **Enter.

f. On the next line, enter **Latitude:** , and then select **LATITUDE** from the Insert Field drop-down menu.

g. To the right of «**LATITUDE**» enter **

, and press **Enter.

h. On the next line, enter **Longitude:** , and then select **LONGITUDE** from the Insert Field drop-down menu.



i. Click **OK**

13. For Style, select **Default** from the drop-down list.

14. Check the box next to **Directions** to include "To here" and "From here" links in the description balloon in Google Earth EC.

15. Click the **Text Color** button, and select a medium blue.

16. Click the **Background Color** button, and select a light yellow.

17. Click **OK**.

    You return to the Vector Project Editor.

18. Select **File > Save**.

    The Save dialog appears.

19. Navigate to the `ASSET_ROOT/Projects/Vector` folder.

20. Enter **CA_POIs** as the name of your project, and click **Save**.

    The new project appears in the Asset Manager when you select **ASSET_ROOT/Projects/Vector** in the asset navigation tree.

21. Right-click **CA_POIs**, and select **Build** from the context menu.

    Google Earth Enterprise Fusion builds the project.

Define and Build an Imagery Project

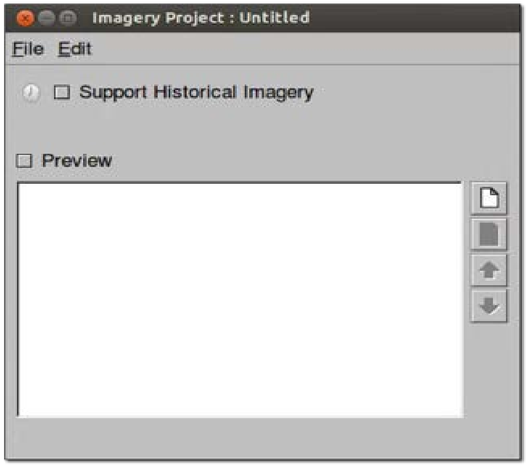
Although you can build and publish a database that includes a vector project only, you cannot connect directly to your server with Google Earth EC to view that data unless your database also includes an imagery project. If you publish a database that includes a vector project only, you must use the **Add Database** command on the File menu in Google Earth EC to add the database to the to the base imagery after logging in to `kh.google.com` or some other server that serves a database that includes imagery.

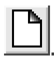
This exercise walks you through the process of defining and building an imagery project using a resource you created in *Defining and Building Resources*. Since the focus of this lesson is on polygon display rules, it is not desirable to spend a lot of time building a large imagery project. So this exercise walks you through creating a small and simple imagery project.

Note: If you have already completed this exercise in *Configuring Display Rules for Polygon Data*, skip this exercise and move on to Define, Build, Push, and Publish a Database for the Point Data.

To define and build an imagery project:

1. In the Asset Manager, click  on the toolbar. The Imagery Project Editor appears.



2. Click .

   The Open dialog appears.

3. Navigate to the `ASSET_ROOT/Resources/Imagery` folder.
4. Select **BlueMarble**, and click **Open**.

   The BlueMarble resource appears in the Imagery Project Editor.

5. Select **File > Save**.

   The Save dialog appears.

6. Navigate to the `ASSET_ROOT/Projects/Imagery` folder.
7. Enter **BlueMarble** as the name of your project, and click **Save**.

   The new project appears in the Asset Manager when you select **ASSET_ROOT/Projects/Imagery** in the asset navigation tree.

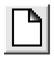
8. Right-click **BlueMarble**, and select **Build** from the context menu.

   Google Earth Enterprise Fusion builds the project.

Define, Build, Push, and Publish a Database for the Point Data

This exercise walks you through the process of defining, building, pushing, and publishing a database using the projects you created in the previous exercise.

To define, build, and push a database:

1. Click  on the toolbar. The Database Editor appears with no projects selected.
2. Click  next to Vector Project. The Open dialog appears.
3. Navigate to `ASSET_ROOT/Projects/Vector`.
4. Select the **CA_POIs** project, and click **Open**. The CA_POIs project appears in the Database Editor next to Vector Project.
5. Repeat steps **2** through **4** to add **BlueMarble** as the imagery project.

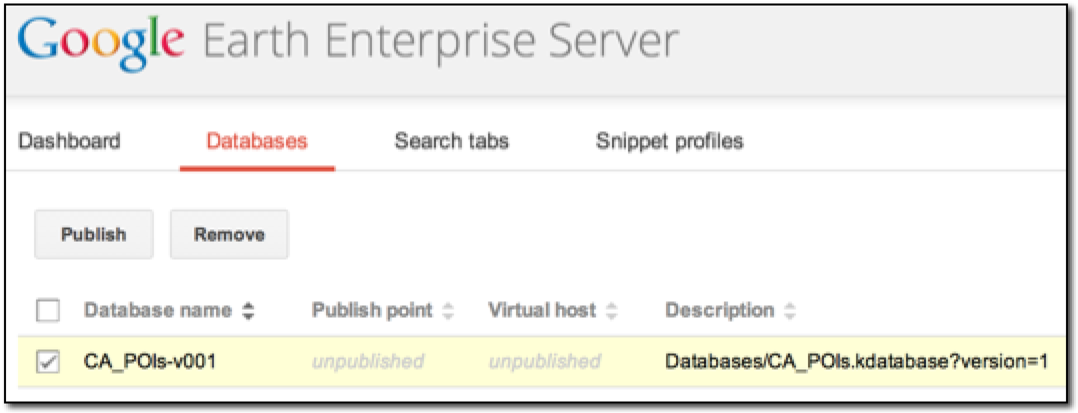
   You do not need to add a terrain project for this exercise. Both projects appear on the list.
6. Select **File > Save**.
7. Navigate to the `ASSET_ROOT/Databases` folder.
8. Enter **CA_POIs** for the name of your database, and click **Save**. The name of the database appears on the right when you select the `/ASSET_ROOT/Databases` folder in the asset navigation tree.
9. Right-click **CA_POIs**, and select **Build** from the context menu. Google Earth Enterprise Fusion builds the database.
10. Right-click **CA_POIs**, and select **Push** from the context menu. The Push Database dialog appears.
11. Select the server association on the Server Associations drop-down list. The most recent version of the selected database is the default selection on the Version drop-down list.
12. Click **Push**. Google Earth Enterprise Fusion runs the process of pushing the database to the Google Earth Enterprise Server, and displays a success message when it is done.

To publish a database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials:

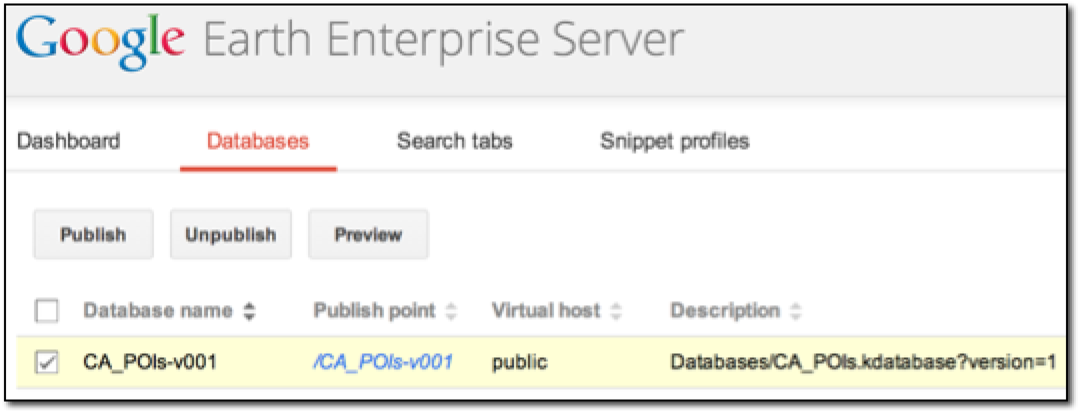
- Username: geapacheuser
- Password:geeadmin

3. Click **Databases** to display the list of databases pushed to the Server.
4. Check the box next to the database you want to publish. The **Publish** button on the Databases page appears.



5. Click **Publish**. The Publish dialog appears.
6. Specify a Publish point, where the database will be accessible from. For example, if you specify CA_POIs-v001, it will be accessible from *myserver.mydomainname*.com/CA_POIs-v001.

   To learn more about the options available in the Publish dialog, see *Create Search Tabs*, *Snippet profiles*, and *WMS*.
7. Click **Publish**. The Databases page updates to indicate the published status of your database.



View Your Database in Google Earth EC

This exercise walks you through the process of viewing your database in Google Earth EC.

To view your database:

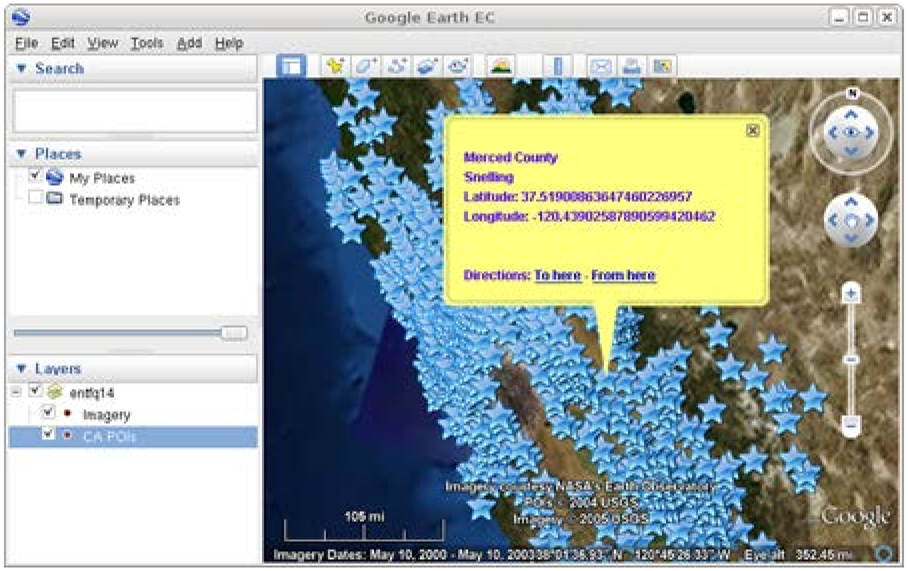
1. Launch Google Earth EC. The Select Server dialog appears.
2. Enter or select the host name or IP address of your server in the Server field, and specify the Publish point that you selected when you published your map database. For example, if you specify CA_POIs-v001, it will be accessible from *myserver.mydomainname*.com/CA_POIs-v001.
3. Click **Sign In**.

   Caution: If you have logged in to this server with Google Earth EC previously, log out, clear your cache, and log back in. See clearing your cache.

4. Zoom in to the west coast of the US.

   When you zoom in enough, Google Earth EC displays blue stars all over California. The closer you zoom in, the more spread out the stars are and the more stars you can see. When you mouse over a star, it turns yellow. When you click a star, a description balloon displays the information you specified in the format you defined in the Display Rules dialog.

Note: If you cannot see the vector data, make sure the check box next to **CA_POIs** in the Layers panel is checked.



[Back to top](#)

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Configuring Display Rules for Polygon Data

You can use elevation, color, and text labels to distinguish the characteristics of polygon shapes in your data. This lesson guides you through the process of graphically distinguishing the population in each county in California using display rules.

- Define and Build a Vector Resource Using Polygon Data
- Define, Configure, and Build a Vector Project
- Define and Build an Imagery Project
- Define, Build, and Publish a Database for the Polygon Data
- View Your Database in Google Earth EC

Define and Build a Vector Resource Using Polygon Data

This exercise walks you through the process of defining and building a vector resource using US census data for the state of California organized by county.

To define and build a vector resource using polygon data:

1. Select **Asset Manager** from the Tools menu.

   The Asset Manager appears.

2. Click  on the toolbar.

   The Vector Resource Editor appears.

3. Set the acquisition date to today’s date in year-month-day format by clicking each section of the date and enter the values.
4. Select **GNIS/US Census Bureau** from the Provider drop-down list.
5. Click **Add**.

   The Open Source dialog appears.
6. Navigate to the `/opt/google/share/tutorials/fusion/Vector` folder.
7. Select the `cal_counties_census.shp` file, and click **Open**.

   The selected file appears on the Source File(s) list.
8. Select **File > Save**.

   The Save dialog appears.
9. Navigate to the `/ASSET_ROOT/Resources/Vector` folder you created in *Setting Up the Tutorial*.
10. Enter the name **CACountyPopulation** for the resource, and click **Save**.

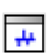
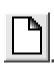
    The name of the resource appears on the right when you select the `/ASSET_ROOT/Resources/Vector` folder in the asset navigation tree.
11. Right-click **CACountyPopulation**, and select **Build** from the context menu.

    Google Earth Enterprise Fusion builds the resource.

Define, Configure, and Build a Vector Project

This exercise walks you through the process of defining, configuring, and building a vector project using the resource you created in the previous exercise.

To define and build a vector project and configure display rules for polygons:

1. Click  on the toolbar. The Vector Project Editor appears.
2. Click .

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

The Open dialog appears.

3. Navigate to the `ASSET_ROOT/Resources/Vector` folder.
4. Select **CACountyPopulation**, and click **Open**.

   The CACountyPopulation resource appears in the Vector Project Editor.

5. Right-click **CACountyPopulation**, and select **Configure Display Rules** from the context menu.

   The Display Rules dialog appears with the Feature tab in the foreground and the **default select all** rule highlighted.

6. Specify the geometry characteristics of the polygons (in this case, the counties):
   a. For Draw Features As, select **Polygons**.
   b. For Visibility, accept the default range, **4** through **24**.
   c. For Maximum Simplification Error, accept the default setting, **0.5**.
   d. Under Draw Style, set:
      - Mode to **Outlined and Filled**.
      - Fill Color to medium green.

        Tip: When you set fill color, specify a value of 125 for the alpha channel instead of 255. This makes the fill color semi-transparent instead of opaque.

      - Outline Color to black.
      - Outline Width to **1.**

      The result of these settings is that Google Earth EC displays each county as a green polygon with a black outline.
   e. Under Elevation/Height, set:
      - Mode to **Relative**.
      - Check the box next to **Extrude**.

- Check the box next to **User-Defined Height**.
- Height Variable to **POP2000**. (Click the empty text field, select **POP2000** from Insert Field the drop-down list, and click **OK**.)
- Offset to **0**.
- Scale to **0.05**.

The result of these settings is that Google Earth EC displays each county at an elevation relative to its population. That is, counties with higher population appear *taller*. Counties with lower population appear *shorter*. (Refer to the *Google Earth Enterprise Fusion Reference Guide* for complete details about each of the settings in this dialog.)

7. On the right side of the dialog, check the box next to **Draw Label**.

   This allows you to specify the content of a text label for each county.

8. For Visibility, set the range to **4** through **24**.
9. Under Label Properties, select **Text** from the drop-down list, and then click the empty text field.

   The Format Label dialog appears.

   a. Select **COUNTY** from the Insert Field drop-down list, and click **OK**.

   b. For Highlight, set:

      - Color to pale yellow.
      - Scale to **1**.

      The result of these settings is that Google Earth EC displays each county's name in bright yellow text. Each text label is centered within its county

   c. For Normal, set:

      - Color to bright yellow.
      - Scale to **1**.

10. Click **OK**.

You return to the Vector Project Editor.

11. Select **File > Save**.

    The Save dialog appears.

12. Navigate to the `ASSET_ROOT/Projects/Vector` folder.

13. Enter **CACountyPopulation** as the name of your project, and click **Save**.

    Note: Although this is the same name as the resource, Google Earth Enterprise Fusion allows it, because they are different asset types and, therefore, have different file name extensions. In addition, they are being stored in different folders.

    The new project appears in the Asset Manager when you select **ASSET_ROOT/Projects/Vector** in the asset navigation tree.

14. Right-click **CACountyPopulation**, and select **Build** from the context menu.

    Google Earth Enterprise Fusion builds the project.

Define and Build an Imagery Project

Although you can build, push, and publish a database that includes a vector project only, you cannot connect directly to your server with Google Earth EC to view that data unless your database also includes an imagery project. If you publish a database that includes a vector project only, you must use the **Add Database** command on the File menu in Google Earth EC to add the database to the base imagery after logging in to `kh.google.com` or some other server that serves a database that includes imagery.

This exercise walks you through the process of defining and building an imagery project using a resource you created in *Defining and Building Resources*. Since the focus of this lesson is on polygon display rules, it is not desirable to spend a lot of time building a large imagery project. So this exercise walks you through creating a small and simple imagery project.

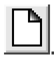
Note: If you have already completed this exercise in *Configuring Display Rules for Point Data*, skip this exercise and move on to Define, Build, Push, and Publish a Database for the Polygon Data.

To define and build an imagery project:

1. Click  on the toolbar.

   The Imagery Project Editor appears.

2. Accept all of the default values in the Legend area. (Refer to the *Google Earth Enterprise Fusion Reference Guide* for details about these settings.)

3. Click .

   The Open dialog appears.

4. Navigate to the `ASSET_ROOT/Resources/Imagery` folder.

5. Select **BlueMarble**, and click **Open**.

   The BlueMarble resource appears in the Imagery Project Editor.

6. Select **File > Save**.

   The Save dialog appears.

7. Navigate to the `ASSET_ROOT/Projects/Imagery` folder.

8. Enter **BlueMarble** as the name of your project, and click **Save**.

   The new project appears in the Asset Manager when you select **ASSET_ROOT/Projects/Imagery** in the asset navigation tree.

9. Right-click **BlueMarble**, and select **Build** from the context menu.

   Google Earth Enterprise Fusion builds the project.

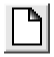
Define, Build, and Publish a Database for the Polygon Data

This exercise walks you through the process of defining, building, and publishing a database using the projects you created in the previous exercise.

To define, build, and push a database:

1. Click  on the toolbar.

   The Database Editor appears with no projects selected.

2. Click  next to Vector Project.

   The Open dialog appears.

3. Navigate to `ASSET_ROOT/Projects/Vector`.

4. Select the **CACountyPopulation** project, and click **Open**.

   The CACountyPopulation project appears in the Database Editor next to Vector Project.

5. Repeat steps 2 through 4 to add **BlueMarble** as the imagery project.

   You do not need to add a terrain project for this exercise. Both projects appear on the list.

6. Select **File > Save**.

   The Save dialog appears.

7. Navigate to the `ASSET_ROOT/Databases` folder.

8. Enter **CACountyPopulation** for the name of your database, and click **Save**.

   The name of the database appears on the right when you select the `/ASSET_ROOT/Databases` folder in the asset navigation tree.

9. Right-click **CACountyPopulation**, and select **Build** from the context menu.

   Google Earth Enterprise Fusion builds the database.

10. Right-click **CACountyPopulation**, and select **Push** from the context menu.

    The Push Database dialog appears.

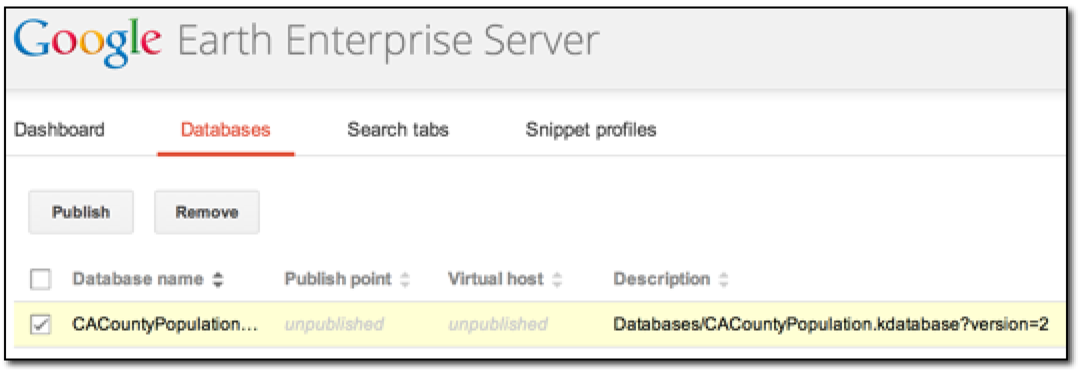
11. Select the server association on the Server Associations drop-down list. The most recent version of the selected database is the default selection on the Version drop-down list.

12. Click **Push**.

Google Earth Enterprise Fusion runs the process of pushing the database to the Google Earth Enterprise Server, and displays a success message when it is done.

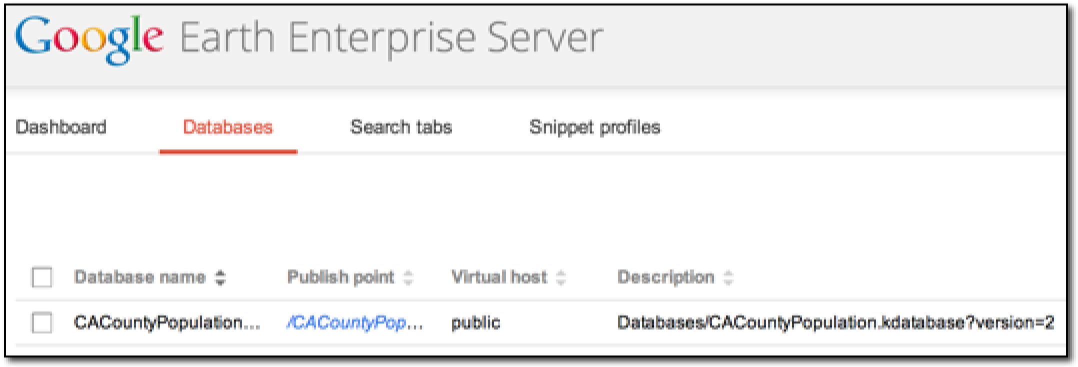
To publish a database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials:
    - Username: geapacheuser
    - Password:geeadmin
3. Click **Databases** to display the list of databases pushed to the Server.
4. Check the box next to the database you want to publish. The **Publish** button on the Databases page appears.



5. Click **Publish**. The Publish dialog appears.
6. Specify a Publish point, where the database will be accessible from. For example, if you specify CACountyPopulation-v001, it will be accessible from *myserver.mydomainname*.com/CACountyPopulation-v001.

   To learn more about the options available in the Publish dialog, see *Create Search Tabs*, *Snippet profiles*, and *WMS*.
7. Click **Publish**. The Databases page updates to indicate the published status of your database.



View Your Database in Google Earth EC

This exercise walks you through the process of viewing your database in Google Earth EC.

To view your database:

1. Launch Google Earth EC. The Select Server dialog appears.
2. Enter or select the host name or IP address of your server in the Server field, and specify the Publish point that you selected when you published your map database. For example, if you specify CACountyPopulation-v001, it will be accessible from *myserver.mydomainname*.com/CACountyPopulation-v001.
3. Click **Sign In**.

Caution: If you have logged in to this server with Google Earth EC previously, log out, clear your cache, and log back in. See clearing your cache.

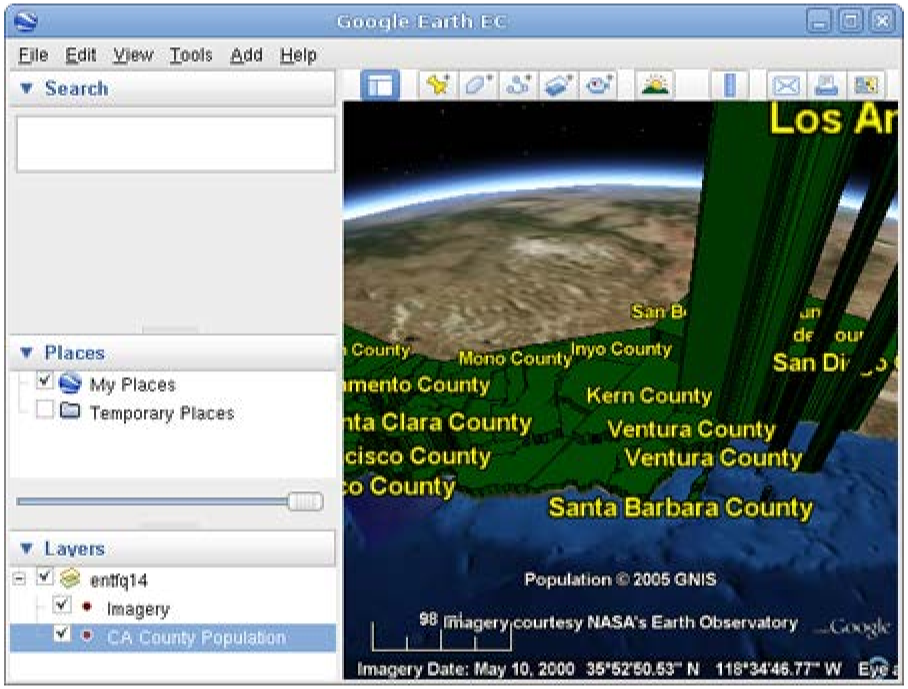
4. Zoom in to the west coast of the US.

Google Earth EC displays all of the counties in California labeled with the names of the counties. The closer you zoom in, the more spread out the counties look and the more county names you can see.

Note: If you cannot see the vector data, make sure the checkbox next to **CACountyPopulation** in the Layers panel is checked.

5. Turn and tilt the Earth so you can see the *heights* of the counties.

   Each county's height indicates its relative population. Counties with higher populations are *taller*, and counties with lower populations are *shorter*.

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Importing and Exporting Style Templates

Often when you define the display rules for one set of data, you can use the same display rules (sometimes with minor tweaking) for other similar sets of data. In fact, to make your data look consistent across different layers, you probably want to use the same display rules. To do so, you can simply export as a style template the display rules you want to reuse and then import that style template into other projects.

For example, if a project contains two resources with different providers, and you want the data to look the same except for the copyright information, you can define the display rules for one resource, export the display rules as a template, and then import the template for the other resource. Another example is that you might want to define display rules for a small set of data and then use those same rules on a larger set of data. There are many more circumstances in which you can save time by using style templates.

This lesson guides you through the process of defining display rules for a small set of data (the state of California) and then exporting those rules as a style template. In a subsequent exercise, you import that same style template for a larger set of data (the entire US). Since you already defined display rules for population by county in the state of California in the last chapter, most of the work is already done.

Note: If you have not completed *Configuring Display Rules for Polygon Data*, you must complete at least the first three exercises in that lesson (Define and Build a Vector Resource Using Polygon Data, Define, Configure, and Build a Vector Project, and Define and Build an Imagery Project) before continuing with this lesson.

- Export the Display Rules
- Import the Template
- Define, Build, Push, and Publish a Database for Your Data
- View Your Data in Google Earth EC

Export the Display Rules

1. Select **Asset Manager** from the Tools menu.

   The Asset Manager appears.

2. Navigate to the `/ASSET_ROOT/Projects/Vector` folder you created in *Configuring Display Rules for Polygon Data*.
3. Double-click **CACountyPopulation** to open it in the Vector Project Editor.
4. Right-click **CACountyPopulation**, and select **Export Configuration as Template** from the context menu.

   The Export Template dialog appears.

5. Navigate to your home folder, and click  to create a new folder; name the new folder **templates**.
6. Open the templates folder, enter **Counties** in the File name field, and click **Save**.

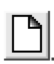
   You return to the Vector Project Editor.

7. Close the Vector Project Editor.

Import the Template

This exercise walks you through importing the template you created in the previous exercise into a new project.

To define a project and import a template:

1. Click  on the toolbar. The Vector Project Editor appears.
2. Click . The Open dialog appears.
3. Navigate to the `ASSET_ROOT/Resources/Vector` folder.
4. Select **USPopulation**, and click **Open**.

   The US Population resource appears in the Vector Project Editor.

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

5. Right-click **USPopulation**, and select **Import Configuration from Template** from the context menu.

   The Import Template dialog appears.

6. Navigate to the folder where you saved the template in the previous exercise.

7. Select **Counties.khdsp**.

8. Check the box next to **Apply display rules** but not **Apply legend settings** at the bottom of the dialog.

   If you check the box next to **Apply legend settings**, this step would apply all of the legend settings for the project from which the template was created to this project, including changing the name of the layer. (See Chapter 5 in the *Google Earth Enterprise Fusion Reference Guide* for details about these settings.)

9. Click **Open.**

   You return to the Vector Project Editor.

10. Right-click **USPopulation**, and select **Layer Properties** from the context menu.

    The Layer Properties dialog appears.

11. Click **Off** next to Initial State to change it to **On**.

    Changing the initial state to on results in the US Population layer being automatically checked (turned on) in Google Earth EC.

12. Right-click **USPopulation**, and select **Configure Display Rules** from the context menu.

    The Configure Display Rules dialog appears.

13. Set **Max Resolution Level** to **14** (or to 12 if you want to save some time).

    Tip: Leaving the level set at the default, **18**, results in unduly long processing time. To determine the recommended maximum resolution level, use the `gevectorimport-tool` command. This command prints out the recommended maximum resolution level in `gevectorimport-log`. For example, for USPopulation, the log reads:

    - Fusion Notice: Recommended levels for vector project build:
    - Fusion Notice: Min Resolution Level (max feature has diameter 1/8 of tile): 1
    - Fusion Notice: Max Resolution Level (min feature has diameter 1/8 of tile): 14

- Fusion Notice: Efficient Resolution Level (average feature has diameter 1/8 of tile): 7
- Fusion Notice: Efficient Resolution Level can be used as Min/Max Resolution Level for data sets with big variance of feature diameters distribution.

So in this example, the maximum resolution level is 14.

14. Click **OK**.

    You return to the Vector Project Editor.

15. Select **File > Save**.

    The Save dialog appears.

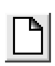
16. Navigate to the `ASSET_ROOT/Projects/Vector` folder.
17. Enter **USPopulation** as the name of your project.
18. Click **Save**.

    The new project appears in the Asset Manager when you select **ASSET_ROOT/Projects/Vector** in the asset navigation tree.

Define, Build, Push, and Publish a Database for Your Data

This exercise walks you through the process of defining, building, pushing, and publishing a database using the projects you created in the previous exercise.

To define, build, and push a database:

1. Click  on the toolbar. The Database Editor appears with no projects selected.
2. Click  next to Vector Project. The Open dialog appears.
3. Navigate to `ASSET_ROOT/Projects/Vector`.
4. Select the **USPopulation** project, and click **Open**. The US Population project appears in the Database Editor next to Vector Project.

5. Repeat steps **2** through **4** to add **BlueMarble** as the imagery project. You do not need to add a terrain project for this exercise. Both projects appear on the list.
6. Select **File > Save**.
7. Navigate to the `ASSET_ROOT/Databases` folder.
8. Enter **USPopulation** for the name of your database, and click **Save**.

   The name of the database appears on the right when you select the `/ASSET_ROOT/Databases` folder in the asset navigation tree.
9. Right-click **USPopulation**, and select **Build** from the context menu.

   Google Earth Enterprise Fusion builds the database. The build process might take a little longer than it has for previous databases, because it is building the project as well, since you did not build the project at the end of the previous exercise.
10. Right-click **USPopulation**, and select **Push** from the context menu. The Push Database dialog appears.
11. Select the server association on the Server Associations drop-down list. The most recent version of the selected database is the default selection on the Version drop-down list.
12. Click **Push**.

    Google Earth Enterprise Fusion runs the process of pushing the database to the Google Earth Enterprise Server, and displays a success message when it is done.

To publish a database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials:
   - Username: geapacheuser
   - Password:geeadmin
3. Click **Databases** to display the list of databases pushed to the Server.
4. Check the box next to the database you want to publish. The **Publish** button on the Databases page appears.

5. Click **Publish**. The Publish dialog appears.
6. Specify a Publish point, where the database will be accessible from. For example, if you specify USPopulation-v001, it will be accessible from *myserver.mydomainname*.com/USPopulation-v001.

   To learn more about the options available in the Publish dialog, see *Create Search Tabs*, *Snippet profiles*, and *WMS*.
7. Click **Publish**. The Databases page updates to indicate the published status of your database.

View Your Data in Google Earth EC

This exercise walks you through the process of building, publishing, and viewing your database in Google Earth EC.

To build, publish, and view your database:

1. Launch Google Earth EC.

   The Select Server dialog appears.
2. Enter or select the host name or IP address of your server in the Server field, and specify the Publish point that you selected when you published your map database. For example, if you specify USPopulation-v001, it will be accessible from *myserver.mydomainname*.com/USPopulation-v001.
3. Click **Sign In**.

Caution: If you have logged in to this server with Google Earth EC previously, log out, clear your cache, and log back in. See clearing your cache.

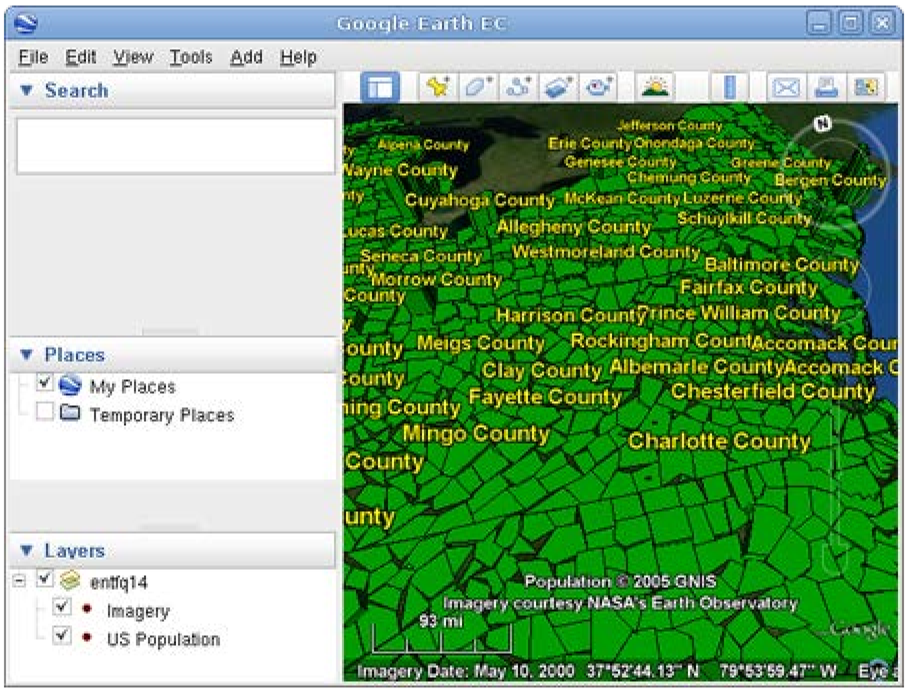
---

1. Zoom in until you can see part of the US.
2. Turn and tilt the Earth so you can see the *heights* of the counties.

   Google Earth EC displays all of the counties in the US labeled with the names of the counties. The closer you zoom in, the more spread out the counties look and the more county names you can see. As with the database you created in *Configuring Display Rules for Polygon Data*, each county's height indicates its relative population. Counties with higher populations are *taller*, and counties with lower populations are *shorter*.

Note: If you cannot see the vector data, make sure the check box next to **US Population** in the Layers panel is checked.

If this were a real project, you might decide that displaying the names of the counties makes this view too cluttered. You can go back into the vector project, modify the display rules, rebuild and publish the database, and then view your data in Google Earth EC again. Repeat this process as many times as required to get the result you want.



Back to top

Specifying Search Fields for Individual Layers

You can specify one or more fields by configuring layer properties in your source data on which users of Google Earth EC can search. This lesson guides you through the process of specifying one field on one layer as searchable, using the vector project you created in *Configuring Display Rules for Polygon Data*. If you have not yet completed that lesson, complete at least the first two sections, Define and Build a Vector Resource Using Polygon Data and Define, Configure, and Build a Vector Project, before continuing with this lesson.

- Specify Search Fields in Layer Properties
- Build and Push Your Database
- Specify Your Search Tab and Publish Your Database in Google Earth Enterprise Server
- View your Database with the Search Tab

Specify Search Fields in Layer Properties

To specify the search fields in your data:

1. Select **Asset Manager** from the Tools menu.

   The Asset Manager appears.

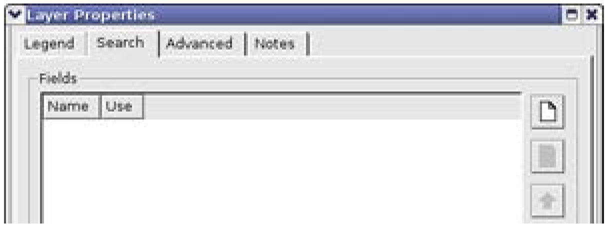
2. Navigate to the `/ASSET_ROOT/Projects/Vector` folder.
3. Double-click **CACountyPopulation**.

   The **CACountyPopulation** resource appears in the Vector Project Editor dialog.

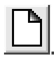
4. Right-click **CACountyPopulation**, and select **Layer Properties** from the context menu.

The Layer Properties dialog appears.

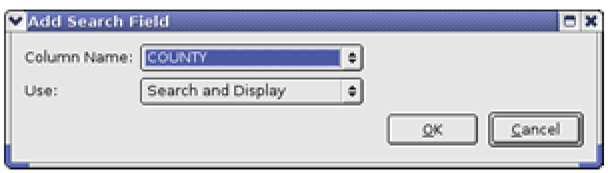
5. Click the **Search** tab.



When you specify one or more search fields on this tab, a search tab appears in Google Earth EC that allows users to search for data in the selected field(s).

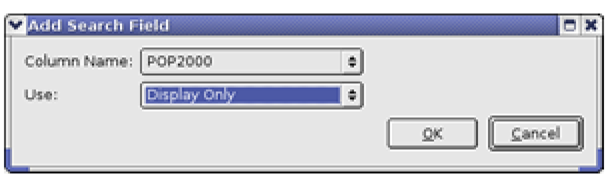
6. Click . The **Add Search Field** dialog appears.

7. Select **COUNTY** from the drop-down list next to Column Name, select **Search and Display** from the drop-down list next to Use, and then click **OK**.



These selections allow users to search on the COUNTY field and display the county name in the description label.

8. Select **POP2000** from the drop-down list next to Column Name, select **Display Only** from the drop-down list next to Use, and then click **OK**.



Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

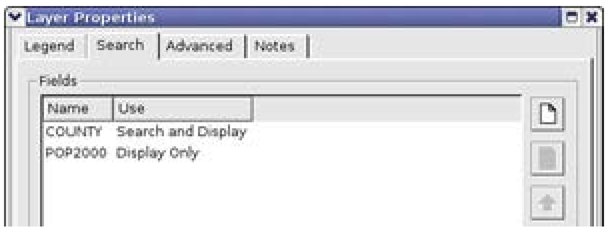
Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

These selections result in the county population being displayed in the description label for each county that matches the user's search criteria. The first field selected for display becomes the placemark name. The second field selected for display becomes the snippet. Any additional fields selected for display become the description.

Note: This search performs an exact match, which means that the Google Earth EC user must enter the value exactly as it appears in the database. It can also perform a wildcard search, so users can enter part of the text they want to match. For example, if a user enters "dakota" in a state search, the search results match "North Dakota" and "South Dakota".

Your selections appear on the list of search fields on the Search tab in the order in which you added them. Use the Up or Down arrows to reorder the list.



The order in which the fields appear on this list dictates the order in which they appear in the description label in Google Earth EC.

9. Click **OK**.
10. Save the project.

Push To Your Database

1. In the Asset Manager, navigate to the `/ASSET_ROOT/Databases` folder.
2. Right-click the **CACountyPopulation** database and select **Build** from the context menu.

   The status of the database immediately changes to **Queued** and then to **In Progress**.
3. Double-click the Current Version or Current State column of the **CACountyPopulation** to view the progress

of the build.

The Version Properties dialog displays the most recent version of that database. You can expand the version tree to view the status of the build in real time by clicking the **+** signs.

When Google Earth Enterprise Fusion finishes building the database, its Current State column in the Asset Manager changes to **Succeeded**, and its Current Version column changes to the date and time the most recent build was started.

4. Right-click the **CACountyPopulation** database and select **Push** from the context menu.

   The Push Database dialog appears. Choose your server association from the Server Associations drop-down list. The most recent version of the selected database is the default selection on the Version drop-down list.

5. Click **Push**. Google Earth Enterprise Fusion pushes the database to Google Earth Enterprise Server, and displays a success message when it is done.

   Note: If you get an error message, contact your Google Earth Enterprise Server administrator for help, or check the *Google Earth Enterprise Administration Guide* for more information.

Specify Your Search Tab and Publish Your Database in Google Earth Enterprise Server

After you push your database to Google Earth Enterprise Server, you can publish it for viewing with Google Earth EC.

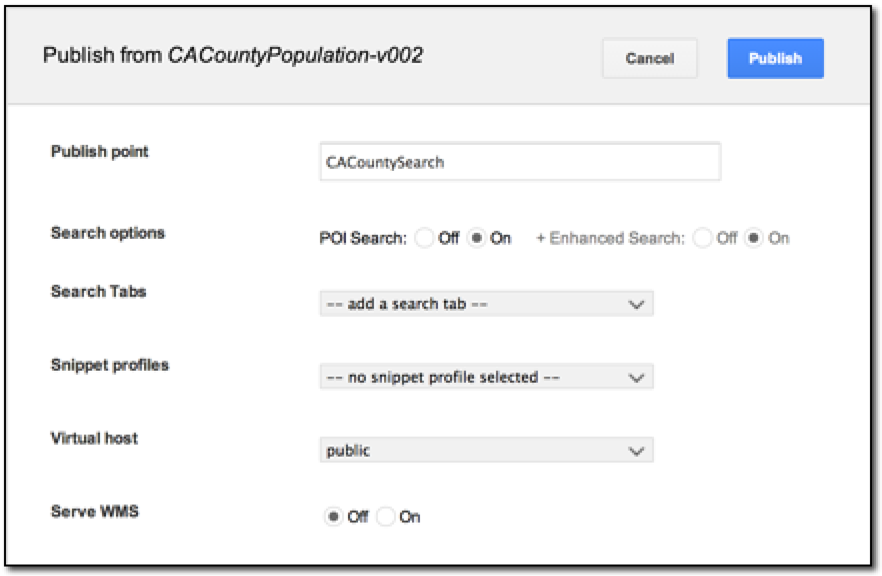
To publish a database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials or the username and password assigned to you:
   - Default username: *geapacheuser*
   - Default password: *geeadmin*

   Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.
3. Click **Databases** to display the list of databases pushed to the Server.

CACountyPopulation-v001 is listed. The Description, Type, Date, and Size of the database are also noted.

4. Check the box next to CACountyPopulation-v001. The **Publish** button on the Databases page appears.
5. Click **Publish**. The Publish dialog appears.
6. Select the POI Search radio button labeled **On**. The **+ Enhanced Search** option appears. Leave its radio button setting as **On**. Doing so will first search against the *POI database*, and if no results are found, the *GeocodingFederated database* is automatically searched.
7. Change the default Publish point from CACountyPopulation-v001 to **CACountySearch**.



To learn more about the options available in the Publish dialog, see *Create Search Tabs*, *Snippet profiles*, and *WMS*.

8. Click **Publish**. The Databases page updates to indicate the published status of your database.

View your Database with the Search Tab

Now you can view the POISearch search tab in Google Earth EC and use it to search on the county information parameters you specified in your database.

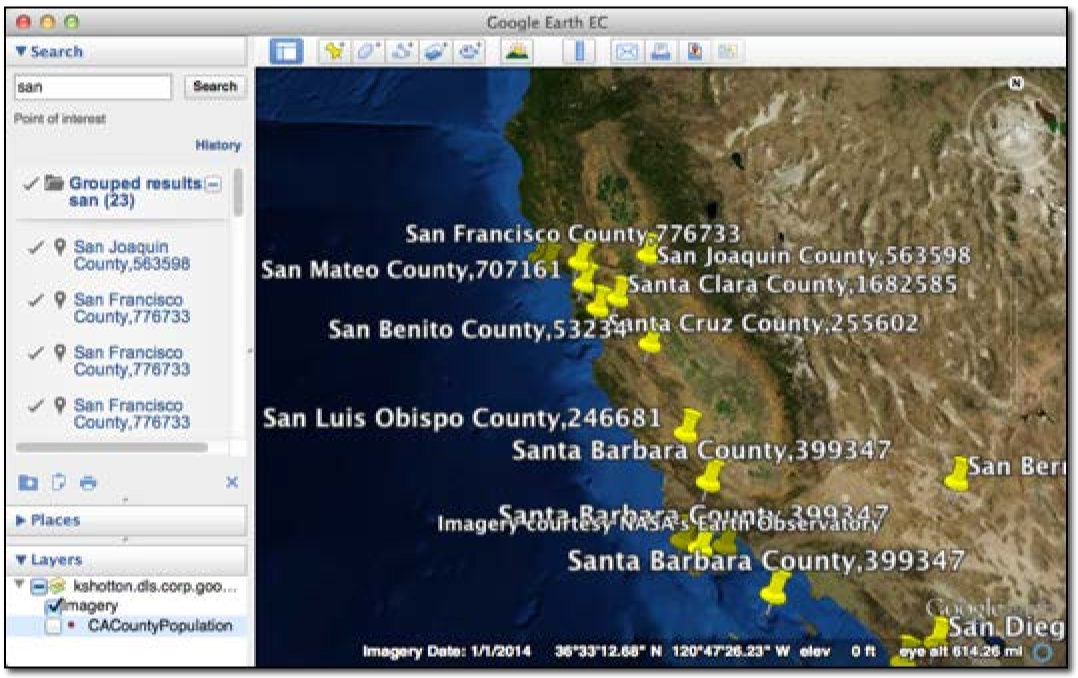
1. Launch Google Earth EC, and log in to your server.

2. Specify the Publish point by entering or selecting the URL or IP address of your server and database in the Server field. For this tutorial, the Publish point is *myserver.mydomainname*.com/CACountySearch, where *myserver* and *mydomainname* are specific to your server. An example URL would be http://myhostname/CACountySearch.

3. Click **Sign In**.

Caution: If you have logged in to this server with Google Earth EC previously, log out, clear your cache, and log back in. See clearing your cache.

Google Earth EC displays your database with the search tab you specified. When you enter a search string, such as “San”, all of the counties containing that string appear in the search results, and POI icons indicate the location of each of the search results on the map.



Back to top

Configuring a Searchable Database

Search tabs allow Google Earth EC and Google Maps users to search external databases on Google Earth Enterprise Servers as well as non-Google servers. For example, if you have a database of San Francisco neighborhoods that contains specific information that your users need about each neighborhood, you can add a search tab called SF Neighborhoods and configure it to search for neighborhoods in your database, even if that database is stored on another server. Google distributes some sample databases with the Google Earth Enterprise Server.

The first thing you need to do is create a search tab definition using the Search tabs page of Google Earth Enterprise Server. Then you can associate the search tab with a published database. The exercises in this lesson show you how to perform both steps.

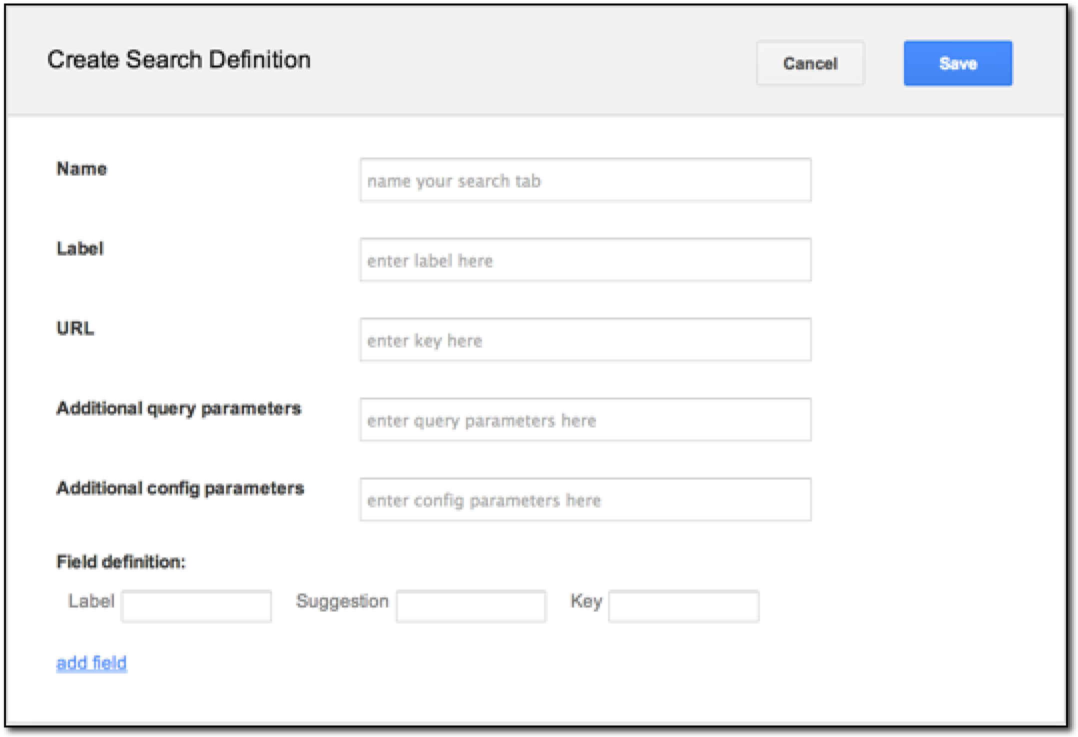
- Define Search Tab
- Add Search Tab To Your Database

Define Search Tab

To define a new search tab:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials:
   - Username: *geapacheuser*
   - Password: *geeadmin*
3. In the GEE Server Admin console, click **Search tabs**.

4. Click **Create new**. The Create Search Definition dialog appears.



5. Enter **SFNeighborhoods** in the **Name** field as the unique name for your search tab.
6. Enter **San Francisco Neighborhoods** in the Label field.

   This label will appear on the search tab that displays with your database in Google Earth EC.
7. Enter **/gesearch/ExampleSearch** in the **URL** field.

   The **ExampleSearch** is one of the search plug-ins that are available with Google Earth Enterprise Server. **ExampleSearch** provides a searchable database of information about San Francisco neighborhoods.
8. Enter **flyToFirstElement=true&displayKeys=location** in the **Additional query parameters** field.

   These query parameters specify that Google Earth EC flies to the first element of your search results and displays the location key in the results.

   Tip: When a Google Earth EC user specifies a search value, the Key value is associated with the value

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

specified by the user and added to the `URL GET` request submitted to the example plug-in. It does not necessarily correlate to a field in the search database.

9. For **Additional config parameters**, leave the field blank.
10. For **Field Definitions**, enter **San Francisco Neighborhood** in the **Suggestion** field and **q** in the **Key** field.

    This is one of the plug-in examples supplied with Google Earth Enterprise Server. For more information about plug-ins for search tabs and the other fields in this dialog, see *Create Search Tabs*.

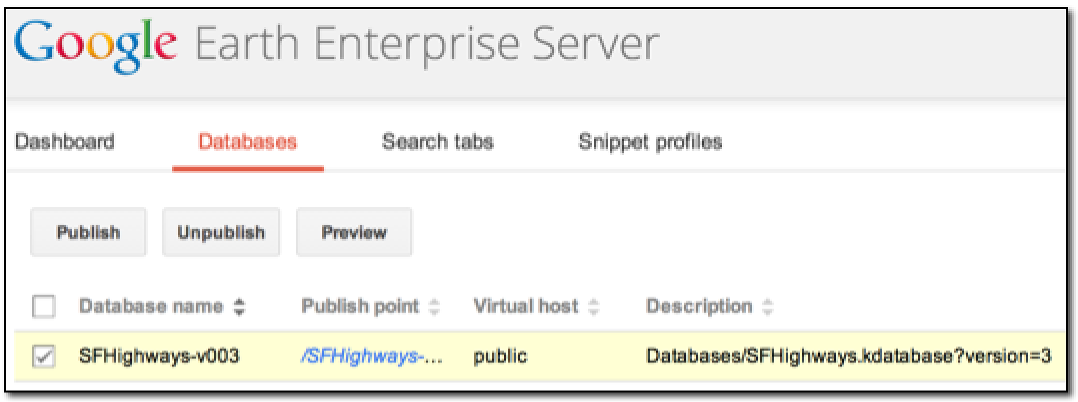
11. Click **Save**. Now the search tab is ready for you to add to your databases.

Add Search Tab To Your Database

To add search tabs to your database:

1. In Google Earth Enterprise Server, click **Databases** on the Admin console.

   The Databases page appears.

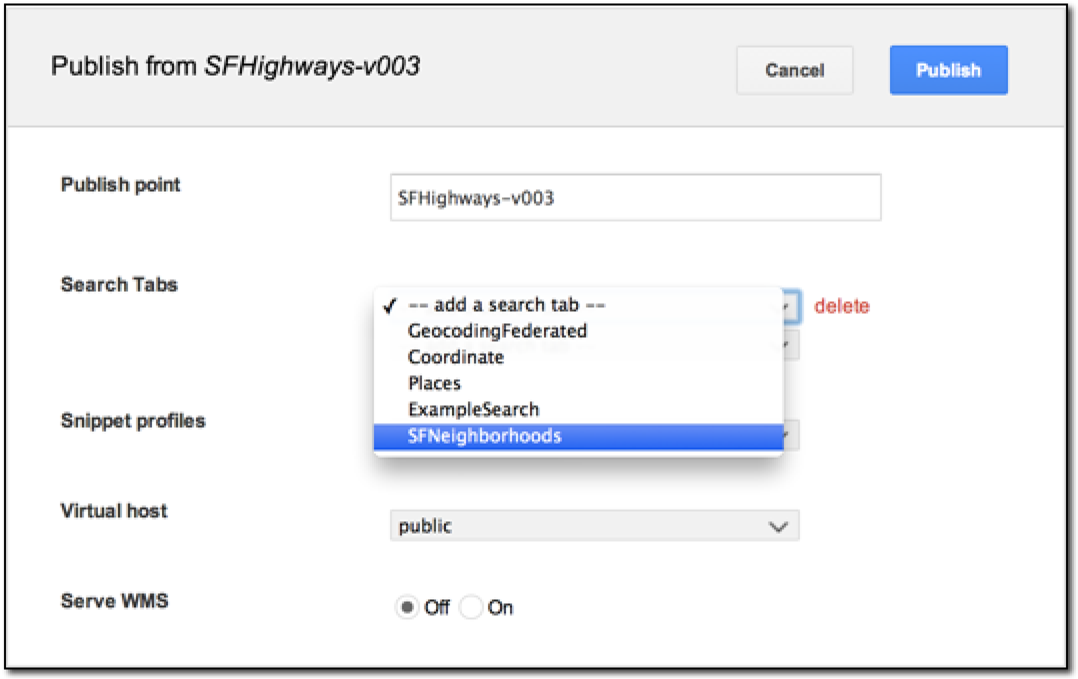


2. Check the box next to the **SFHighways** database that you built in *Defining and Building Databases* and click **Unpublish**.

   The **Publish point** and **Virtual host** fields update to **unpublished**.

   Even though you published the database before, you must publish it again to include the new search tab.

3. Click **Publish**. The Publish dialog appears.



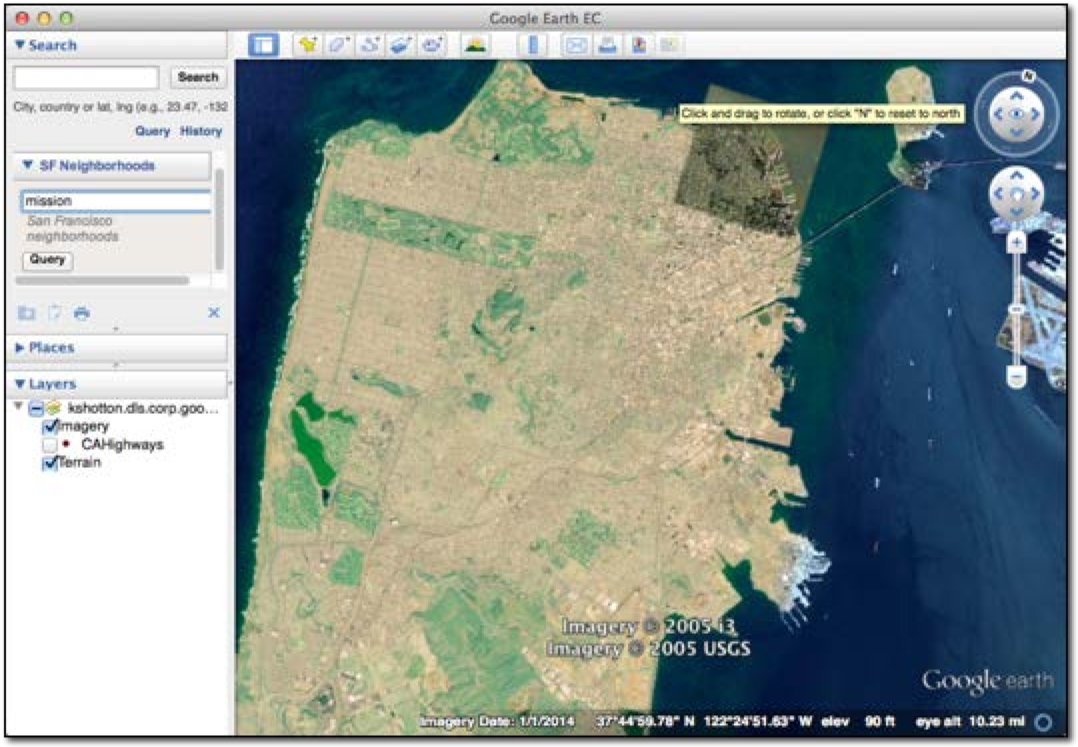
4. Accept the default Publish point of the database name, **SFHighways** and the version number.
5. Select **SFNeighborhoods** from the **Search Tabs** drop-down list.
6. Click **Publish**.

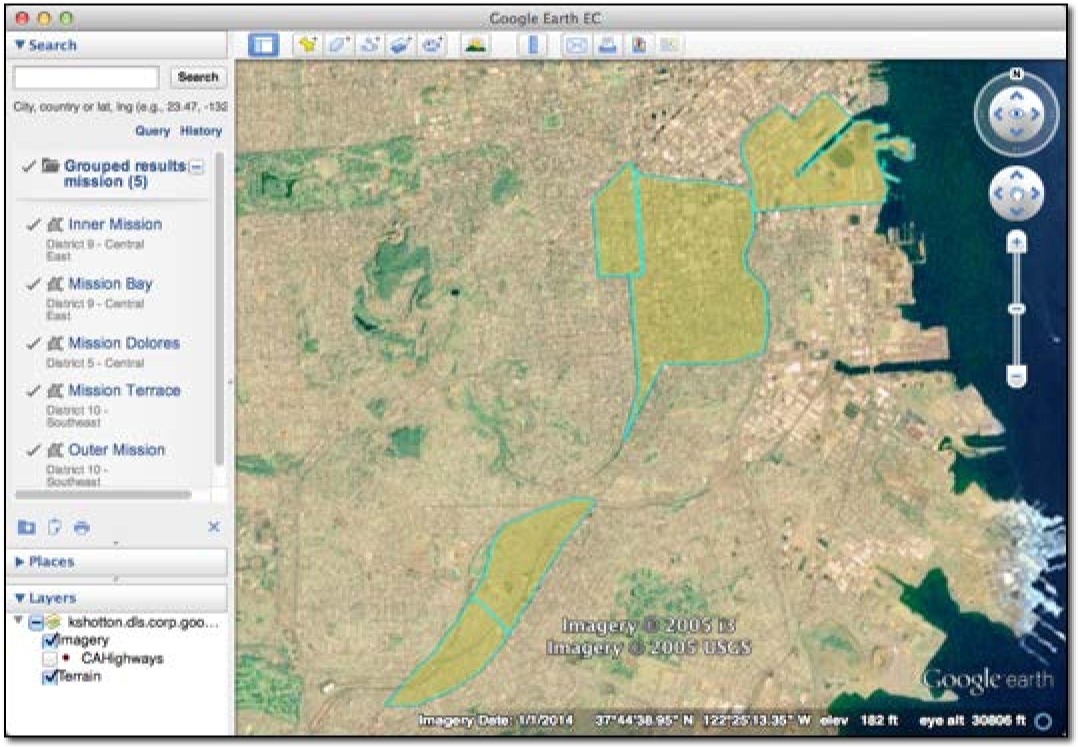
   Google Earth Enterprise Server publishes the **SFHighways** database with the associated **SFNeighborhoods** search tab.

7. Launch Google Earth EC, and log in to your server.

Caution: If you have logged in to this server with Google Earth EC previously, log out, clear your cache, and log back in. See clearing your cache.

Google Earth EC displays your database with the **SF Neighborhoods** search tab you specified.


Google Earth EC
Search
Search
City, country or lat, lng (e.g., 23.47, -132
Query History
SF Neighborhoods
mission
San Francisco neighborhoods
Query
Places
Layers
kshotton.dls.corp.goo...
Imagery
CAHighways
Terrain
Click and drag to rotate, or click "N" to reset to north
Imagery © 2005 i3
Imagery © 2005 USGS
Google earth
Imagery Date: 1/1/2014 37°44'59.78" N 122°24'51.63" W elev 90 ft eye alt 10.23 mi



Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Creating a Map Database

In addition to using Google Earth Enterprise Fusion to prepare and publish data in Google Earth, you can use it to prepare and publish map data in Google Maps. Map data can be published with Google Earth Enterprise Fusion using Plate Carrée (EPSG: 4326) and Mercator (EPSG: 3857) projections, and is referred to hereafter as a *map layer*, *map project*, and *map database*.

This lesson defines, builds, and publishes a Mercator map database and it also demonstrates how you can add an imagery project of *flat or Plate Carrée-based* imagery resources. When you publish the Mercator map database, the flat imagery is then projected as *Mercator on the Fly*. This feature converts a Plate Carrée-based imagery resource into a Mercator-based imagery resource when you define your Mercator map database. When hosted on GEE Server, the server requests the *Mercator on the Fly* imagery resource in the database and serves it with a Mercator projection.

The option to project *Mercator on the Fly* enables you to re-use existing flat imagery resources for all your database types, including Mercator map databases, thereby saving you the task of creating and storing Mercator imagery resources. For more information, see Add flat imagery to Mercator map databases.

Note: As an alternative to your own imagery resources, you can use the Google Base Map in place of an imagery project. See Using the Google Base Map for more details.

The first step in preparing any data for publication is to import the source data as Google Earth Enterprise Fusion resources. You can use one of the vector resources you defined in *Defining and Building Resources* for Google Maps.

After you define your vector resources, you must define and build at least one map layer for each map project. The first exercise in this lesson, Define a Map Layer, describes how to do so.

The remaining steps are similar to defining, building, pushing, and publishing other databases that you have also created for viewing in Google Earth EC.

- Define a Map Layer

Define a Map Layer

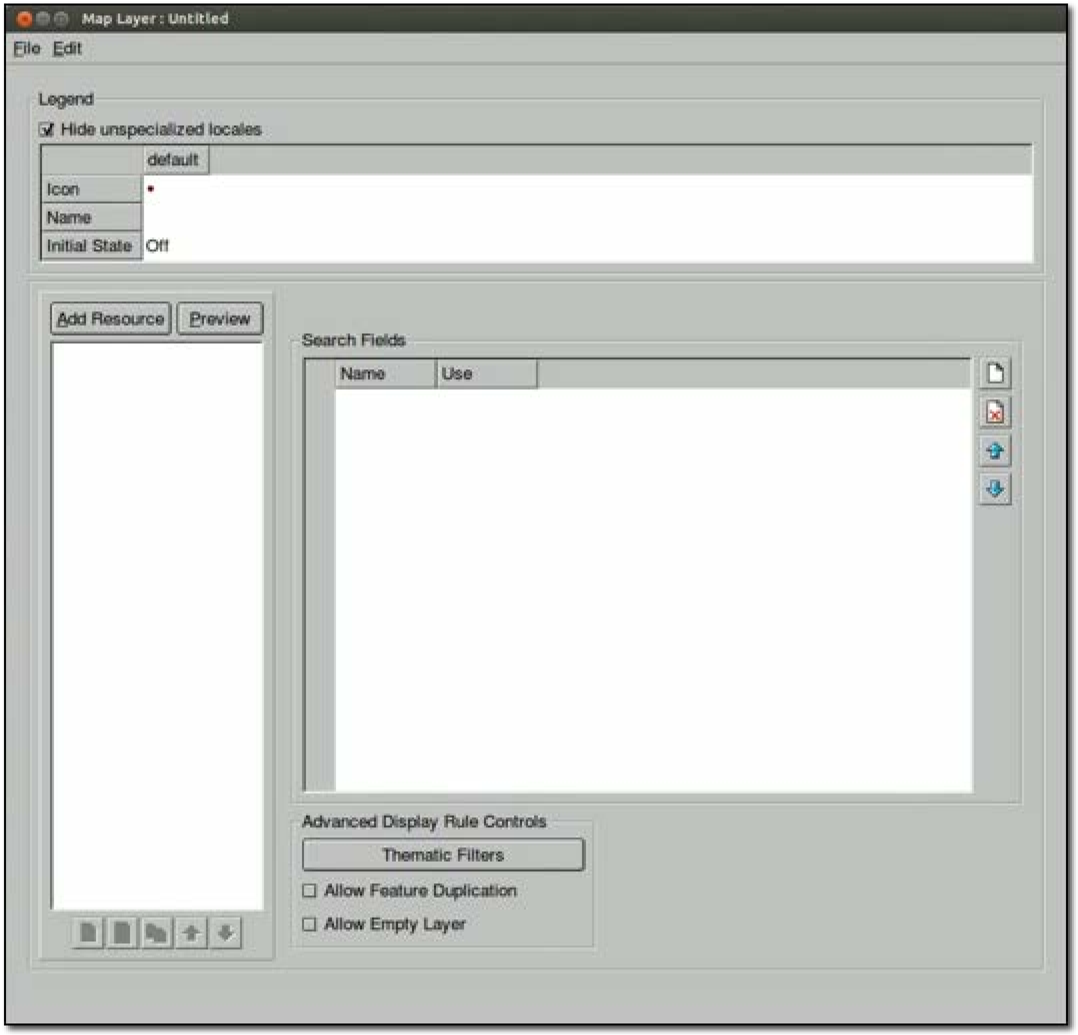
Defining a map layer consists of adding resources to the layer and defining the display rules and filters for the layer. This is similar to the process of defining display rules and filters for a vector project destined for Google Earth EC.

To define a map layer:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Select **Mercator Map Tools (2D)** from the drop-down menu.

> **Note:**The maps layer includes Google Maps layers and uses the Google Maps API from google.com. Because the maps layer uses Mercator projection, it requires a separate imagery database. For more details, see Define and Build Mercator Imagery Resources.

3. Click . The Map Layer Editor appears.



The legend area displays the current value of each property you can set for the map layer. The top row lists the names of the locales you support. When you first open this dialog for a new map layer, only the default locale and its values appear, and the box next to **Hide unspecialized locales** is checked. Google Maps uses the default locale settings when you do not specify different settings for a user's locale in this tab.

4. Set the property values for the default locale as follows:

   a. Click the icon in the Icon field, select **places3_new**, and click **OK**. This is the icon that appears next to the name of the map layer in the Layers panel of Google Maps.

   b. Click in the Name field, and enter **CAfreeways**, and press **Enter** to save the change. This is the text label that appears in the Layers panel of Google Maps.

   c. For Initial State, accept the default setting, **Off**. This setting determines whether the map layer is turned on or off in Google Maps.

5. Click **Add Resource**.
6. Navigate to the `ASSET_ROOT/Resources/Vector` folder.
7. Select **CAHighways**, and click **Open**. The CAHighways resource appears on the resource list.
8. Select **default select all** under the resource name.

   The **Feature** and **Filter** tabs appear on the right. The first option on the **Feature** tab is **Draw Features As**. This option allows you to specify the display rules for the selected resource. The value you set for **Draw Features As** determines the options available for you to specify.

   In this lesson, you define display rules for label only.

9. For **Draw Features As**, select **Lines**. The **Lines** options appear.
10. Under **Line**, accept the default visibility range for the lines, **8** and **14**.
11. For **Color**, set it to orange.
12. For **Line Width**, accept the default, **2**.
13. Check the box next to **Label**.
14. Under **Label**, accept the default setting on the drop-down list, **Text**.
15. Click . The **Label Format** dialog appears.

    The **Insert Field** drop-down list contains the names of all fields in your source data.

16. Select **NAME**, and click **OK**.

    In the source file for the selected resource, the **Name** column lists the names of the roads, so when you select the **Name** field here, it results in the names of the roads appearing in Google Maps.

17. For the label, change the visibility range to **10** through **14**.

    The visibility range refers to the zoom level at which your labels are visible in Google Maps.

18. The default text style is black on white in the Sans 12 font. Click the text style button (labeled **Sans/12**) to specify a different text style for the labels.

    The **Text Style** dialog appears.

19. For **Size**, change the value to **10**, and accept the default setting for Color, black.

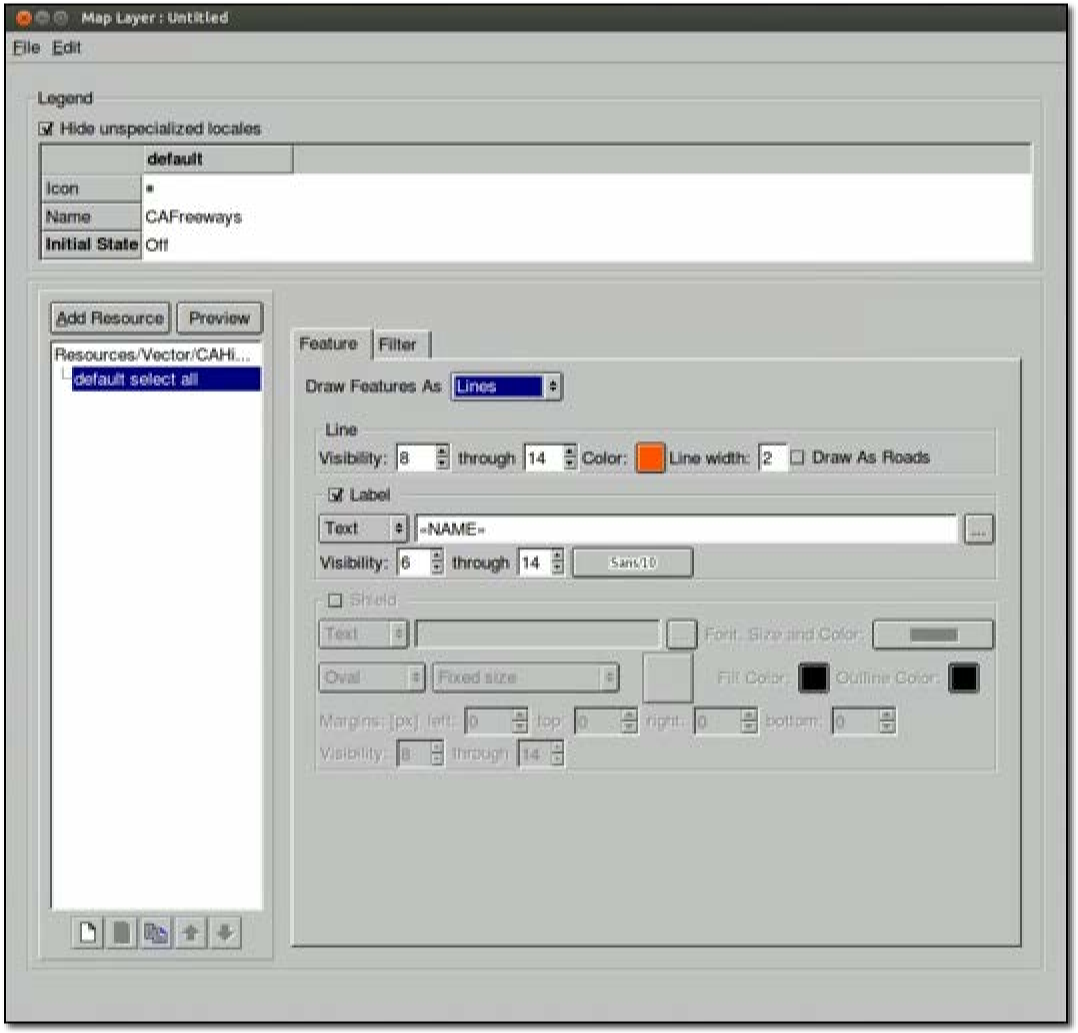
    Note: In version 3.0 and beyond, only one font and style is provided, Sans regular. However, you can create a configuration file in which you can specify additional fonts, if desired. Refer to the **Administration Guide** for details.

    A preview of the label appears in the Preview box.

20. Drag the preview over any button on the left. The new style is stored on that button. The button name reflects the font face and size of the style, and it appears with the selected color and outline attributes.

    After you save a style to a button, you can simply click that button to automatically select its text style settings for another label in the future.

21. Click **OK**. You return to the Map Layer Editor, and the Feature tab shows all of your selections.



22. Select **Save** from the File menu**.**

23. Navigate to the `ASSET_ROOT/MapLayers` folder.

24. In the Name field, enter **SFMapLayer**, and click **Save**. The new map layer name appears in the Asset Manager's asset list.

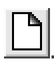
Define a Map Project

The first step in defining a map project is to specify which map layers to include and give the project a name. Before you define a map project, however, add a subfolder to the asset navigation tree in the Asset Manager in which you can store your map projects.

To add a subfolder for map projects:

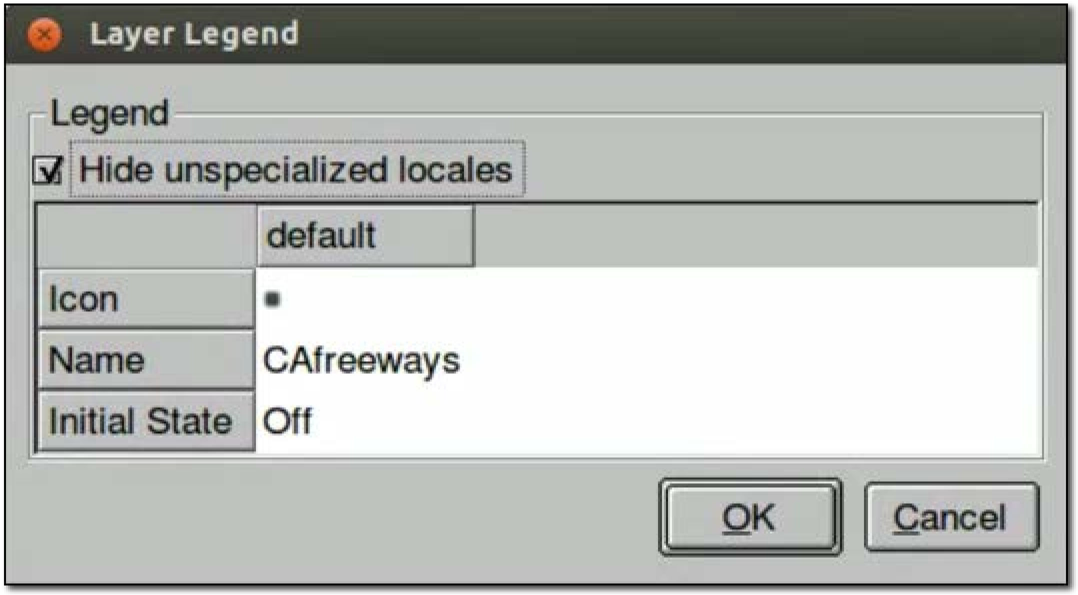
1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Right-click **Projects** in the asset navigation tree, and select **New Subfolder**. The New Subfolder dialog appears.
3. Enter **Map** in the Folder Name field, and click **OK**. The new Map subfolder appears under Projects in the asset navigation tree.

To create a map project:

1. In the Asset Manager, click . The Map Project Editor appears.
2. Click . The Open dialog appears.
3. Navigate to the `ASSET_ROOT/MapLayers` folder.
4. Select **SFMapLayer**, the map layer you created in the previous exercise, and click **Open**. SF Map Layer appears in the Map Project Editor.

   The default legend name of the map layer, **CAFreeways**, appears in the Legend Name column, followed by **<DEFAULT>**, which indicates that this is the name you specified for the legend when you created the layer. The name and path of the map layer appears in the Layer column.

5. Double-click **CAFreeways**. The Layer Legend dialog displays the current values for the map layer.



6. Click the name field, change the name to **California Freeways**, press **Enter** to save the change, and click **OK**.

   The new legend name appears in the Map Project Editor. Notice that **<DEFAULT>** no longer appears after the legend name. (If you want to return to the default name, right-click **California Freeways** in the Legend Name field, and select **Use Layer Defaults** from the context menu.)

7. Select **File > Save**.

8. Navigate to the `ASSET_ROOT/Projects/Map` folder.

9. In the Name field, enter **SFMapProject**, and click **Save**. The new map project name appears in the Asset Manager's asset list.

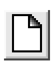
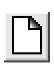
Define a Flat Imagery Project

The maps layer includes Google Maps layers and uses the Google Maps API from google.com. Map databases use either Plate Carrée (EPSG: 4326) and Mercator (EPSG: 3857) projections and normally require corresponding imagery resource types. In this lesson you add a flat imagery project from a previous tutorial lesson to a Mercator map database, which can then be projected as *Mercator on the Fly*. For more information, see Add flat imagery to Mercator map databases.

If you did not complete the tutorial lesson to create an imagery project, see Define an Imagery Project.

Define, Push, and Publish a Map Database

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Select **Mercator Map Tools (2D)**. The tools for Mercator maps appear in the toolbar.
3. Click . The Map Database Editor appears with no projects selected.

4. Click  next to **Map Project**. The Open Asset dialog appears.
5. Navigate to the `ASSET_ROOT/Projects/Map` folder.
6. Select **SFMapProject**, and click **Open**. The SF Map Project appears in the Map Database Editor next to Map Project.
7. Click  next to **Imagery Project**. The Open Asset dialog appears.
8. Navigate to the `ASSET_ROOT/Projects/Imagery` folder.

   Notice that **Type** is set to **Mercator Imagery Project** and only the Mercator imagery project(s) you have built are listed.
9. Select **SFBayAreaMercator**, and click **Open**. The SF Bay Area Mercator imagery project appears in the Map

Database Editor next to Imagery Project.

Both map and Mercator imagery projects appear on the list.

10. Select **File > Save**.
11. Navigate to the `ASSET_ROOT/Databases` folder.
12. In the Name field, enter **SFMapDatabase**, and click **Save**.

    In the Asset Manager, the new map database appears, along with the other databases you have created. Notice that the Category column distinguishes between the Google Earth databases and the Google Maps database.

13. Right-click **SFMapDatabase**, and select **Build** from the context menu. The status of the database immediately changes to **Waiting** or **Queued** and then to **In Progress**.

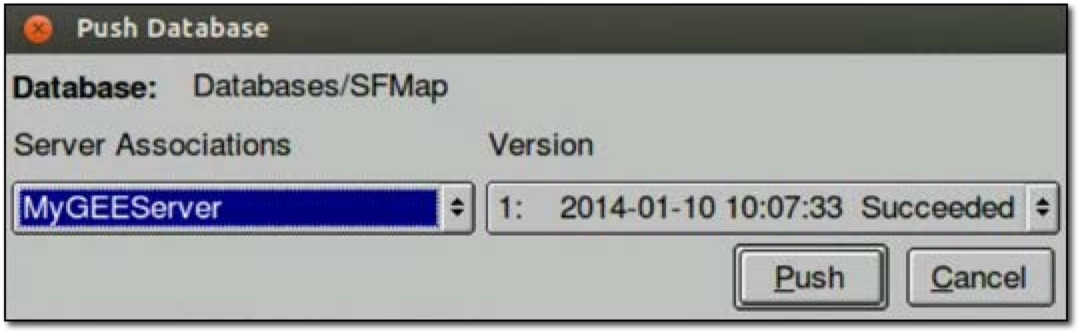
    Note: It might take a while to build the map database, because it is also building the project, since you did not build it.

    When Google Earth Enterprise Fusion finishes building the database, its Current State column in the Asset Manager changes to **Succeeded**, and its Current Version column changes to the date and time the most recent build was started.

Push a Map Database to Google Earth Enterprise Server

To push a map database:

1. Right-click the name of the map database you built in the previous exercise, and select **Push** from the context menu. The Push Database dialog appears.



2. Click **Push**.

   Google Earth Enterprise Fusion pushes the database to the Google Earth Enterprise Server, and displays a success message when it is done.

   Note: If you get an error message, contact your Google Earth Enterprise Server administrator for help, or check the *Google Earth Enterprise Administration Guide* for more information.

Publish a Map Database on Google Earth Enterprise Server

To publish a database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials or the username and password assigned to you:
   - Default username: geapacheuser
   - Default password:geeadmin

   Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.

   ---
3. Click **Databases** to display the list of databases pushed to the Server.
4. Check the box next to the map database you want to publish. The **Publish** button on the Databases page appears.

5. Click **Publish**. The Publish dialog appears.
6. Specify a **Publish point** where the database will be accessible from. For example, if you specify MySFMap, it will be accessible from *myserver.mydomainname*/MySFMap.

   To learn more about the options available in the Publish dialog, see *Create Search Tabs*, *Snippet profiles*, and *WMS*.
7. Click **Publish**. The Databases page updates to indicate the published status of your database.

View Your Map Database

After you publish your map database, you can view it from Google Earth Enterprise Server or in any browser.

To view your map database from Google Earth Enterprise Server:

1. Log in to the Admin console of Google Earth Enterprise Server.

   The Admin console opens to the Databases page.
2. Check the box next to the map database that you want to view.
3. Click **Preview**.

   A new browser tab opens displaying your map database.

To view your map database in any browser:

1. Launch any web browser.
2. Point your browser to:

   ```
   http://myserver.mydomainname/publish_point
   ```

   where *myserver.mydomainname* is the host name or IP address of your server to which you published the map database, and *publish_point* is the publish point that you specified when the published the map database. For example:

   ```
   http://my_host_name/MySFMap
   ```

   If you are not sure which server you published to, contact your Google Earth Enterprise Server administrator for help.

Google Maps displays your database.

Using the Google Base Map

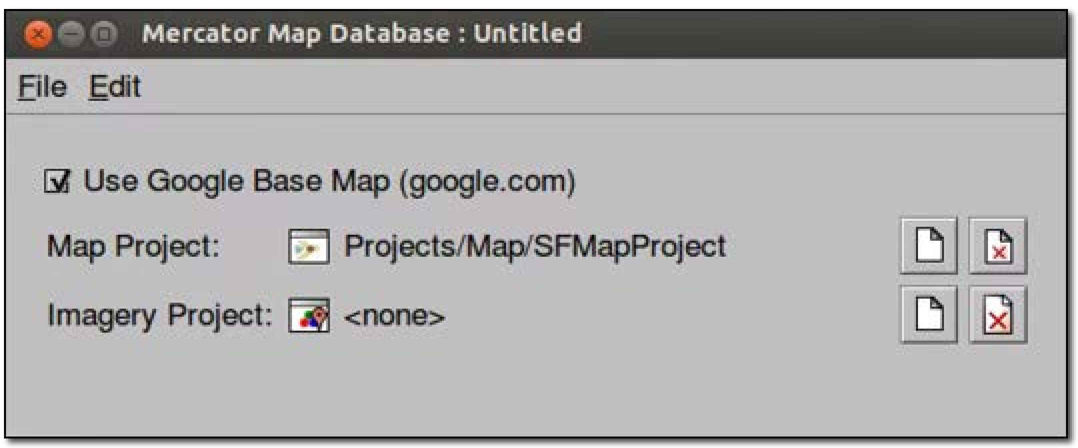
When you build a map database, you have the option to use the Google Base Map. Enabling this feature connects to www.google.com, loads Google base map tiles through the Google Maps Javascript API V3, and renders your map layers on top of them.

Note: As per Google's terms of service, you may need to purchase a Maps for Business license if you are deploying your application in an internal environment.

Deploying the Google Base Map involves adding it when you build your map database.

To build your map database to include the Google Base Map:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Select Mercator Map Tools (2D). The tools for Mercator maps appear in the toolbar.
3. Click . The Map Database Editor appears with no projects selected.



4. Select the **Use Google Base Map (google.com)** checkbox.

5. Add your map project.

   The map project appears in the Map Database Editor next to Map Project.

6. Optionally add an imagery project. This is useful for overlaying high-resolution imagery insets on the Google Base Map.

7. Select **File > Save**.

   Your map database is saved to the location you specified.

8. **Build**, **Push**, and **Publish** your map database as described in Define, Push, and Publish a Map Database.

Learn more

- Google Maps Javascript API V3 Reference

Back to top

Creating Imagery Mosaics

The simplest way to import source data is to create one resource from each data file. In certain cases, though, you might obtain multiple source files that comprise a single image. In that case, you can combine the source files into a single resource, called a *mosaic*. This lesson guides you through the process of creating an imagery mosaic.

Note: All of the steps in this lesson apply to terrain data as well.

Create an Imagery Resource

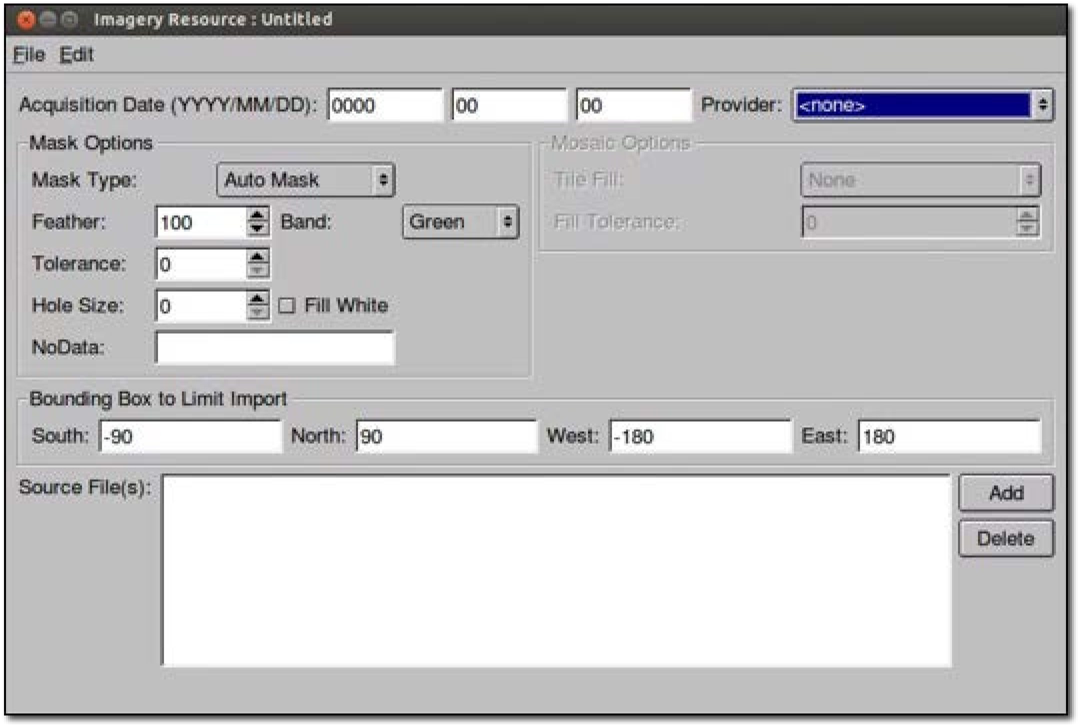
To create a mosaic, the source images must meet the following requirements:

- Source images must be in the same projection and use the same coordinate system
- Source images must have the same resolution
- Source images must have geographic proximity to each other
- Ideally, the individual source images are perfectly abutting (no overlap, no gaps)

To define an imagery mosaic:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.

2. Click  on the toolbar. The Imagery Resource Editor appears.



3. Set the acquisition date to today's date in year-month-day format by clicking each section of the date and enter the values.

4. Select **USGS Imagery** from the Provider drop-down list.

5. Set the Mask Type (under Mask Options) to **No Mask**.

6. Click **Add**. The Open Source dialog appears.

7. Navigate to the `/opt/google/share/tutorials/fusion/Imagery/SF_grayscale/SFnorth` folder.

8. Select all four TIFF files (only the four files with the `.tif` extension), and click **Open**.

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

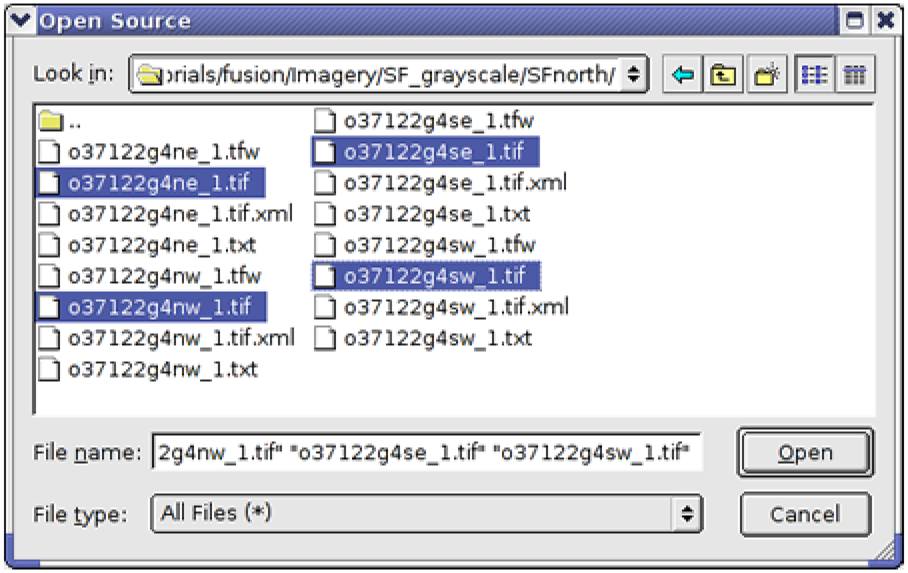
Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters



The selected files appear on the Source File(s) list.

Notice that the Mosaic Options area is enabled when you import more than one imagery file. (Refer to the **Reference Guide** for detailed explanations of these settings.)

9. Under Mosaic Options, for Tile Fill, select **Black**. Now the Fill Tolerance option is enabled.
10. For Fill Tolerance, select **1**.
11. Select **File > Save**.
12. Navigate to the `/ASSET_ROOT/Resources/Imagery` folder you created in *Setting Up the Tutorial*.
13. Enter the name **ImageryMosaic** for the resource, and click **Save**.

    The name of the resource appears on the right when you select the `/ASSET_ROOT/Resources/Imagery` folder in the asset navigation tree.
14. Right-click **ImageryMosaic**, and select **Build** from the context menu. Google Earth Enterprise Fusion builds the resource.

Note: Building this resource could take several minutes, since it contains four images.

---

15. When the status of the resource changes to **Succeeded** in the Asset Manager, drag it to the Preview List pane.

16. Check the box next to **ImageryMosaic** in the Preview List pane, and then right-click it and select **Zoom to Layer** from the context menu.

    The grayscale image of San Francisco appears in the Preview pane, surrounded by a bounding box. This is the single mosaic resource created from four separate source files.

Notice the black border and uneven edges of the mosaic. These are called *fill pixels*. The fill pixels fill the frame around the imagery where its borders are uneven.

If you want to add more imagery around this image and have all of the imagery blend together seamlessly, you need to *mask* the fill pixels. Masking blocks the pixels that you do not want users to see. The auto mask feature in Google Earth Enterprise Fusion does a very good job of creating masks for most situations. The *Google Earth Enterprise*

Fusion Reference Guide provides instructions for doing so.

In some cases, however, you might want to use a custom mask that you create outside of Google Earth Enterprise Fusion so you have more control over the pixels that are blocked and the pixels that are allowed to show through. However, since Google Earth Enterprise Fusion allows you to use custom masks with resources that include a single source file only, you cannot apply a custom mask to a resource that includes more than one source file.

There is a way to resolve this problem. You can create a *virtual mosaic* and apply your custom mask to it. A virtual mosaic is a single source file created from multiple source files. The following exercises walk you through the process of creating a virtual mosaic and applying a custom mask to it.

Create a Virtual Mosaic

If you have multiple source files that you want to combine into a single resource to which you want to apply a custom mask, you can create a virtual mosaic using command line tools and then apply the custom mask within the GUI.

To create a virtual mosaic:

1. At the command prompt, change to the folder that contains the imagery source files by entering:

   ```
   cd /opt/google/share/tutorials/fusion/Imagery/SF_grayscale/SFnorth
   ```

2. Change the permissions to `read/write` for the directory and files that you are using to create the virtual mosaic:

   ```
   chmod 777 /path/mydirectory
   ```

   ```
   chmod 777 /opt/google/share/tutorials/fusion/Imagery/SF_grayscale/SFnorth/*
   ```

3. Create a virtual raster file by entering (all on one line):

   ```
   gevirtualraster --fill 0,0,0 -o path/all_files.khvr *.tif
   ```

   where *path* is the path to the folder where you want to save the resulting `.khvr` file.

   The `gevirtualraster` command specifies the fill value to be used for the mosaic (`0,0,0` = black), the name of the file to be generated (`all_files.khvr`), and includes all of the `.tif` files in the specified folder as input.

4. Change to the folder where you stored the all_files.khvr file (the value of *path* in step **2**) by entering:

   ```
   cd path
   ```

5. Enter the following command to be sure the file was created successfully:

   ```
   ls a*
   ```

   The file `all_files.khvr` should be listed.

6. Change the permissions for the newly created all_files.khvr file:

   ```
   sudo chmod 777 all_files.khvr
   ```

7. Preview the new virtual mosaic:

   a. In the Google Earth Enterprise Fusion GUI, click .

      The Open dialog appears.

   b. Navigate to the `path` folder.

   c. Select **all_files.khvr**, and click **OK**.

      Two entries appear in the Preview List pane: **all_files:0** and **all_files:1**. (If Imagery Mosaic is still listed, you can leave the box next to it checked.)

      When you check the box next to **all_files:0**, a bounding box indicates the position of the entire virtual mosaic. When you check the box next to **all_files:1**, four bounding boxes indicate the position of each of the individual source files.

      Note: Because you have not yet built this resource, only the bounding boxes appear, not the actual imagery.

      ---

8. You can check and uncheck the boxes to see the different views. When you finish, right-click any layer in the Preview List pane, and select **Remove All Layers** from the context menu. A message prompts you to confirm that you want to remove all layers from the Preview panes.

9. Click **OK**. All of the layers disappear from the Preview panes.

10. Back at the command prompt, enter the following command:

    ```
    cp /opt/google/share/tutorials/fusion/Imagery/SF_grayscale/all_files-mask.tif
    path
    ```

    This command copies the mask file provided with the tutorial data into the same folder as `all_files.khvr` you created in step 3.

11. Change to the `path` folder, if necessary, and enter the following command:

    ```
    ls a*
    ```

    Both files, `all_files.khvr` and `all_files-mask.tif`, should be listed.

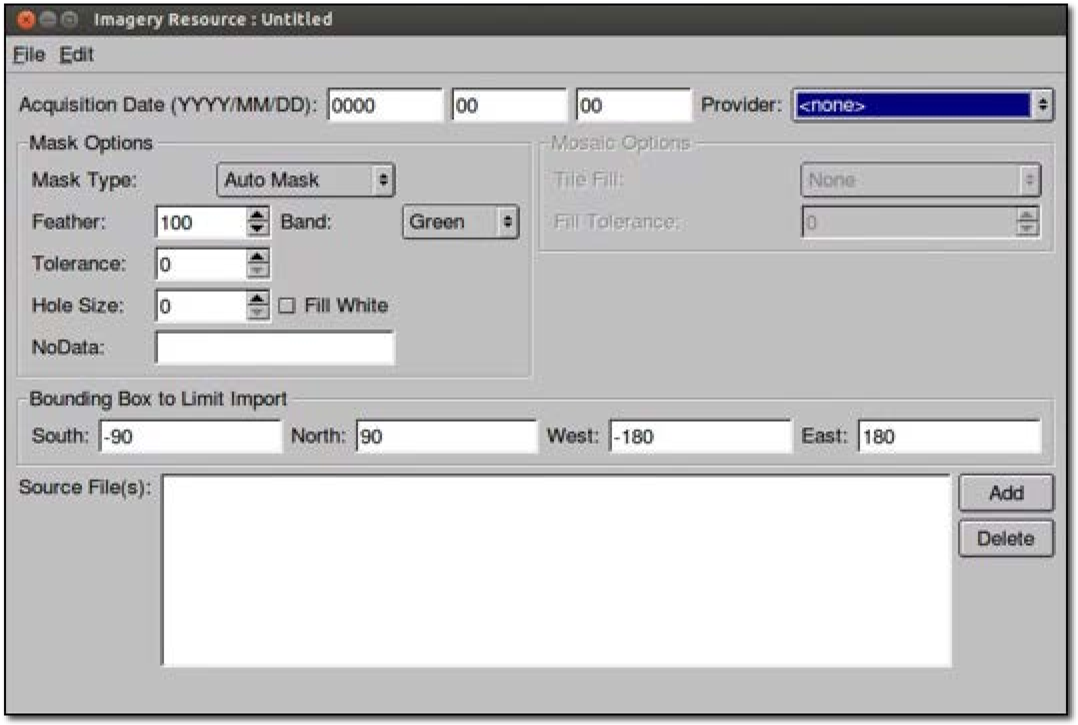
The mask for your input must be located in the same folder as the source file, and the file name must match the name of the source file with `-mask` appended. In this case, the mask file is named `all_files-mask.tif`. When you select **Have Mask** in the Imagery Resource Editor in the next exercise, Google Earth Enterprise Fusion automatically applies the mask file by reference to the source file.

Apply a Custom Mask to the Virtual Mosaic

Now that you have a virtual mosaic and a custom mask for it, you can import the virtual mosaic into a resource and apply the mask. This exercise guides you through that process.

To apply a custom mask to the virtual mosaic:

1. Select **Asset Manager** from the Tools menu. The Asset Manager appears.
2. Click  on the toolbar. The Imagery Resource Editor appears.



3. Set the acquisition date to today's date in year-month-day format by clicking each section of the date and enter the values.
4. Select **USGS Imagery** from the Provider drop-down list.
5. Set the Mask Type (under Mask Options) to **Have Mask**.
6. Click **Add**. The Open Source dialog appears.
7. Navigate to the `path` folder.
8. Select `all_files.khvr`, and click **Open**. The selected file appears on the Source File(s) list.
9. Select **File > Save**.
10. Navigate to the `/ASSET_ROOT/Resources/Imagery` folder you created in *Setting Up the Tutorial*.
11. Enter the name **VirtualMosaic** for the resource, and click **Save**. The name of the resource appears on the right when you select the `/ASSET_ROOT/Resources/Imagery` folder in the asset navigation tree.
12. Right-click **VirtualMosaic**, and select **Build** from the context menu. Google Earth Enterprise Fusion builds

the resource.

Note: Building this resource could take several minutes.

13. When the status of the resource changes to **Succeeded** in the Asset Manager, drag it to the Preview List pane.
14. Check the box next to **VirtualMosaic** in the Preview List pane, and then right-click it and select **Zoom to Layer** from the context menu.

    The grayscale image of San Francisco appears in the Preview pane, surrounded by a bounding box. This is the mosaic created from four separate tiles.

    Notice that the black border and uneven edges of the mosaic are gone. The mask has done its job.

Back to top

Segmenting Large Imagery Files

Google Earth Enterprise Fusion does not allow you to import raw imagery source files larger than 80 GB. (Raw size = number of pixels width * number of pixels height * 3.) Therefore, if you have an imagery source file that is larger than 80GB, you can split it into two or more source files using the `gesplitkhvr` command.

The `gesplitkhvr` tool produces a grid of image files designated as rows and columns. For example, if you specify 2 rows and 2 columns, the image is split into four smaller files. If you create a separate resource for each resulting source file, it would look like this in the Preview pane:

However, you can reconnect the split images by adding them all to a single resource, which would look like this in the Preview pane:

To split a large source file into multiple smaller files:

1. At the command prompt, change to the folder that contains the tutorial imagery source files by entering:

   ```
   cd /opt/google/share/tutorials/fusion/Imagery
   ```

2. Convert a `.tif` file to a `.khvr` file by entering:

   ```
   gevirtualraster -o path/virtual_raster.khvr usgsSFHiRes.tif
   ```

   where *path* is the path to the folder where you want to save the resulting `.khvr` file. (See *Creating Imagery Mosaics* for more information about the `gevirtualraster` tool.)

   Note: The path where you save the resulting `.khvr` file must be on a known volume. (See “`geconfigureassetroot --editvolumes`” in the Command reference.)

3. Change to the folder where you stored the virtual_raster.khvr (the value of *path* in step 2) by entering:

   ```
   cd path
   ```

4. Split the `.khvr` file into a 4x4 grid (16 image files) with a 10-pixel overlap among all of the files by entering:

   ```
   gesplitkhvr --rows 4 --cols 4 --overlap 10 virtual_raster.khvr
   ```

   The names of the resulting files appear on the screen as they are created. The file names are constructed from the name of the original file (`virtual_raster`), a row and column designation for each file (starting with `-R1C3`), and the `.khvr` extension.

Now you can import the 16 resulting image files into a resource.

Back to top

Google

Building a Historical Imagery Project

To build a historical imagery project:

1. Follow the steps in *Defining and Building Resources* to define and build the following:

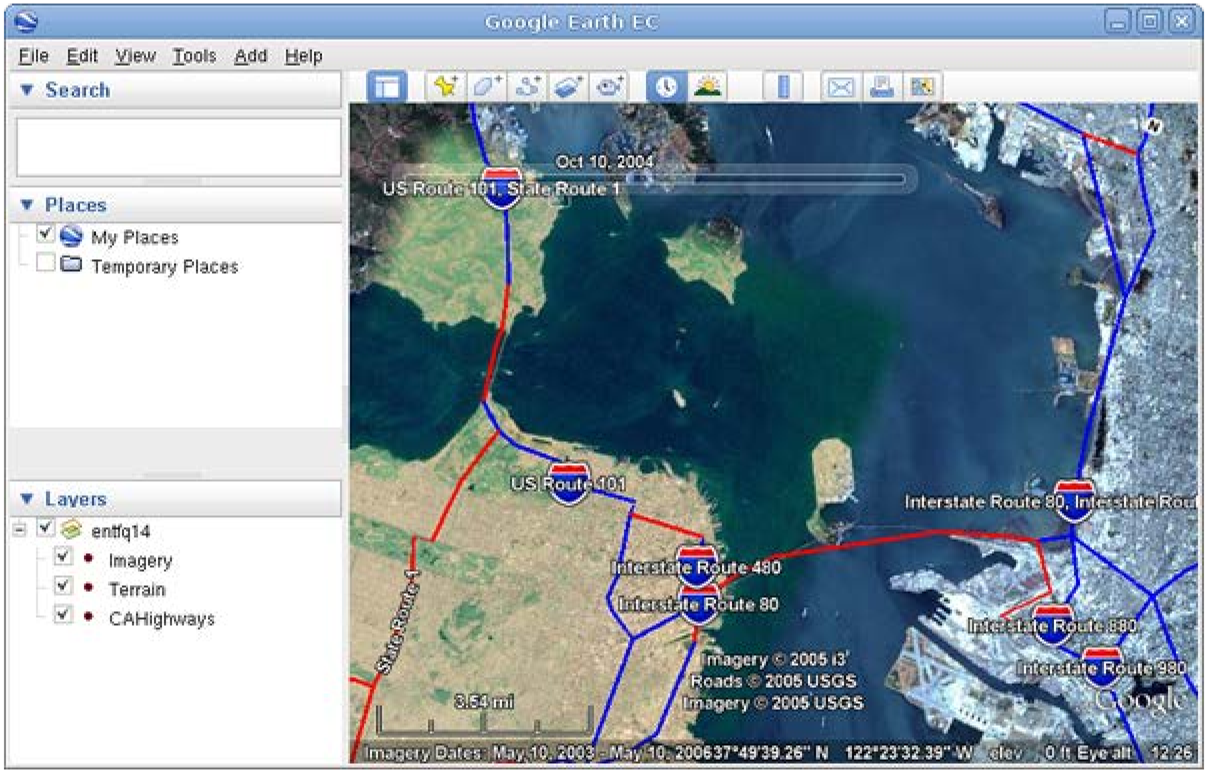
Name (Resources/Imagery/...)	Acquisition Date	Provider	Mask	Source file
BlueMarble	Any date. The date of the base image does not affect historical imagery browsing.	NASA Imagery	**No Mask**	`bluemarble_4km.jp2`
SFBayAreaLanSat_20021010	2002-10-10	USGS Imagery	**Auto Mask Tolerance**: 2 Default for all other mask values.	`usgsLanSat.jp2`
i3_15Meter_20041010	2004-10-10	i3	**Auto Mask** Default	`i3SF15-meter.tif`

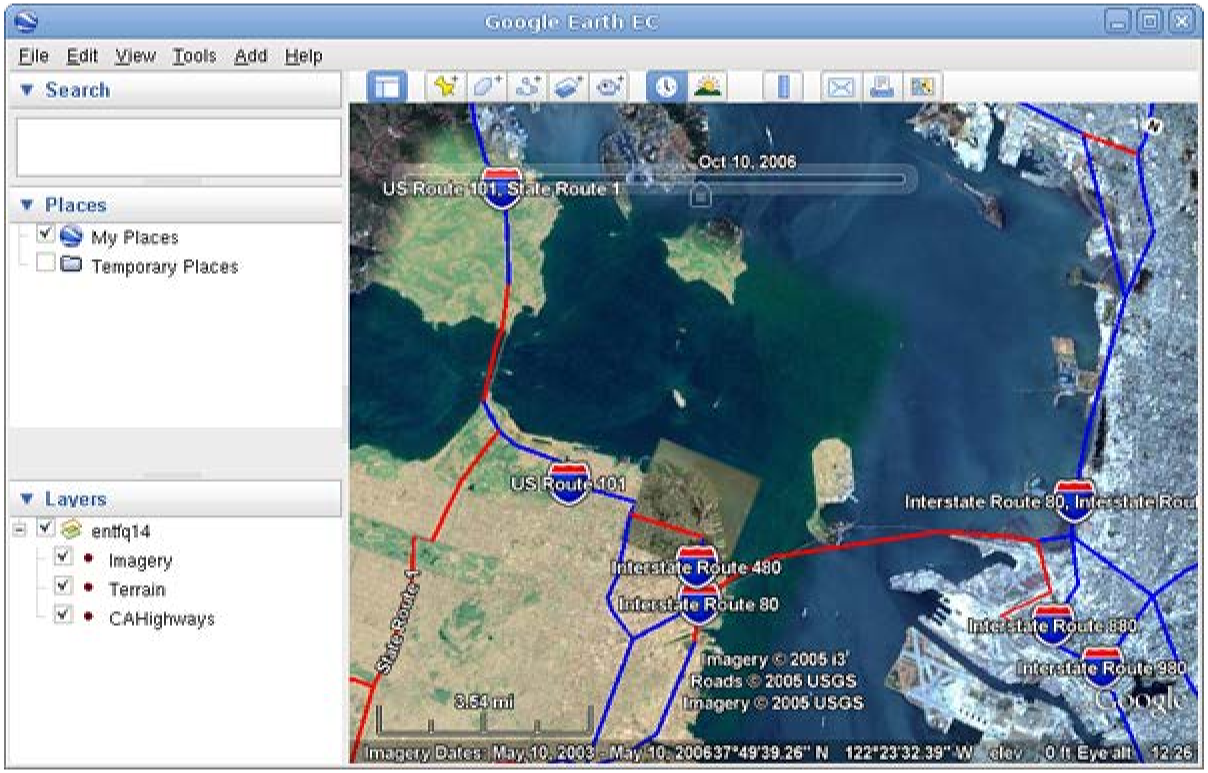
			values.	
SFHighResInset_20061010	2006-10-10	USGS Imagery	**Auto Mask** Default values.	`usgsSFHiRes.tif`

2. Open a new Imagery Project and add each of the image resources to the project:
    - **Resources/Imagery/BlueMarble**
    - **Resources/Imagery/SFBayAreaLanSat_20021010**
    - **Resources/Imagery/i3_15Meter_20041010**
    - **Resources/Imagery/SFHighResInset_20061010**
3. Select the **Support Historical Imagery** checkbox.
4. Select **File > Save**. Enter **Projects/Imagery/SFBayAreaHistorical** as the name of your project, and click **Save**.
5. Build the **SFBayAreaHistorical** project.
6. Create a new Earth Database and add the **SFBayAreaHistorical** project.
7. Save the new database as **Databases/SFBayAreaHistorical**.
8. Build and push the database to GEE Server, then publish it on GEE Server to the default **Publish point**.
9. Launch Google Earth EC.
10. Enter or select the host name or IP address of your server in the Server field, and specify the **Publish point** that you selected when you published your map database.

    For example, if you specify SFBayAreaHistorical-v001, it will be accessible from *myserver.mydomainname*.com/SFBayAreaHistorical-v001.
11. Zoom in to the San Francisco Bay Area.
12. Select the **Historical Imagery** toolbar button  to display a time slider. The time slider allows you to move the view through time.

The following two screens show how this project is displayed in the Google Earth EC client. Notice the timeslider is visible and the differences between the imagery dates and content.





Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Sample Data Files

This appendix provides a list of all of the sample data files included when you install the tutorial data. The files are listed hierarchically with folder names in bold. Blue font designates files that are not used in the tutorial lessons. You can use those files as you choose to explore the features of Google Earth Enterprise Fusion.

- Imagery
- Terrain
- Vector

Imagery

```
bluemarble_4km.jp2
i3SF15-meter.tif
usgsLanSat.jp2
usgsSFHiRes.tif

SF_grayscale
  all_files-mask.tif

  SFnorth
    o37122g4ne_1.tfw
    o37122g4ne_1.tif
    o37122g4ne_1.tif.xml
    o37122g4ne_1.tfw
```

o37122g4nw_1.tif
o37122g4nw_1.tif.xml
o37122g4nw_1.txt
o37122g4nw_1.tfw
o37122g4se_1.tif
o37122g4se_1.tif.xml
o37122g4se_1.txt
o37122g4se_1.tfw
o37122g4sw_1.tif
o37122g4sw_1.tif.xml
o37122g4sw_1.txt
o37122g4sw_1.tfw

SFsouth

o37122f4ne_1.tfw
o37122f4ne_1.tif
o37122f4ne_1.tif.xml
o37122f4ne_1.tfw
o37122f4nw_1.tif
o37122f4nw_1.tif.xml
o37122f4nw_1.txt
o37122f4nw_1.tfw
o37122f4se_1.tif
o37122f4se_1.tif.xml
o37122f4se_1.txt
o37122f4se_1.tfw
o37122f4sw_1.tif
o37122f4sw_1.tif.xml
o37122f4sw_1.txt
o37122f4sw_1.tfw

Terrain

gtopo30_4km.jp2

gtopo30_4km-mask.tif
SF_terrain.tif

Vector

cal_counties_census.dbf
cal_counties_census.prj
cal_counties_census.shp
cal_counties_census.shx
california_hydro_line.dbf
california_hydro_line.khdsp
california_hydro_line.prj
california_hydro_line.sbn
california_hydro_line.sbx
california_hydro_line.shp
california_hydro_line.shx
california_popplaces.csv
california_popplaces.kdx
california_roads_line.dbf
california_roads_line.prj
california_roads_line.sbn
california_roads_line.sbx
california_roads_line.shp
california_roads_line.shx
us_counties_census.dbf
us_counties_census.prj
us_counties_census.shp
us_counties_census.shx
ushydroline.dbf
ushydroline.shp
ushydroline.shx
UsPopPlaces.csv
usroads.dbf

```
usroads.shp
usroads.shx
usroads.txt
geonames-cities500000.csv (courtesy of geonames.org)

SF_neighborhoods
 realtor_neighborhoods0.bmp
  realtor_neighborhoods.dbf
  realtor_neighborhoods.htm
  realtor_neighborhoods.html
  realtor_neighborhoods.prj
  realtor_neighborhoods.shp
  realtor_neighborhoods.shp.xml
  realtor_neighborhoods.shx
```

Back to top

Configure tutorial workspace

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.

This article describes how to configure the tutorial work space and how to switch back and forth between the tutorial asset root and the production asset root. You also learn how to clean up the tutorial files when they are no longer needed.

- Configure the tutorial asset root and source volume
- Select the tutorial asset root
- Clean up the tutorial workspace

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.

   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, Select the Tutorial Asset Root, for more information about selecting different asset roots.)

Select 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 Configure the tutorial asset root and source volume.

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

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

Clean 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
   ```

5. Run the GEE Fusion installer:

   Command-line installer:

   ```
   /path/install_directory/InstallGEFusion.sh
   ```

GUI installer:

```
/path/install_directory/InstallGEFusionGUI.sh
```

In the GUI installer, select **Customize** from the **Choose Products** window and then uncheck Google Earth Enterprise Tutorial, setting the installation process to install Google Earth Enterprise Fusion *only*.

In the command line installer at the **Choose Products** prompt, Enter **3** to select the **Customized** installation. Both Google Earth Enterprise Fusion and Google Earth Enterprise Fusion Tutorial are selected for installation, indicated by the **X** next to each entry.

6. To remove the tutorial from the installation, enter **2** to deselect it.
7. 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.

8. 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).

9. 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.

10. 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`.

11. 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.

12. Permanently delete the selected databases by entering:

```
geserveradmin --garbagecollect
```

Back to top

Create terrain overlays

Introduction

A terrain overlay is a secondary terrain database that you can use to “overlay” on top of another database in Google Earth EC. For example, you may want to add a terrain overlay as an inset of terrain data at a higher resolution. Another great benefit is how easy it is to add new terrain resources for just a specific area of an existing database, requiring less storage space and less time; instead of having to rebuild the entire terrain project, you can now just build a new database for a terrain overlay, based on the new resource, then add it as an additional database (terrain layer) in Google Earth EC.

A terrain database is defined by the resolution tiles that contain it, but the actual terrain can have an irregular shape. For example, a geo-spatially smaller database of terrain that contains the land mass for a country includes areas of the resolution tiles that are outside the country's borders. These areas are called fill, and Google Earth Enterprise Fusion lets you create an overlay that uses lower resolution data for the fill area, which can then “blend” as you add your secondary terrain database on top.

Base terrain database:

About This Tutorial

Setting Up the Tutorial

Defining and Building Resources

Defining and Building Projects

Defining and Building Databases

Pushing, Publishing, and Viewing a Database

Configuring Display Rules for Point Data

Configuring Display Rules for Polygon Data

Importing and Exporting Style Templates

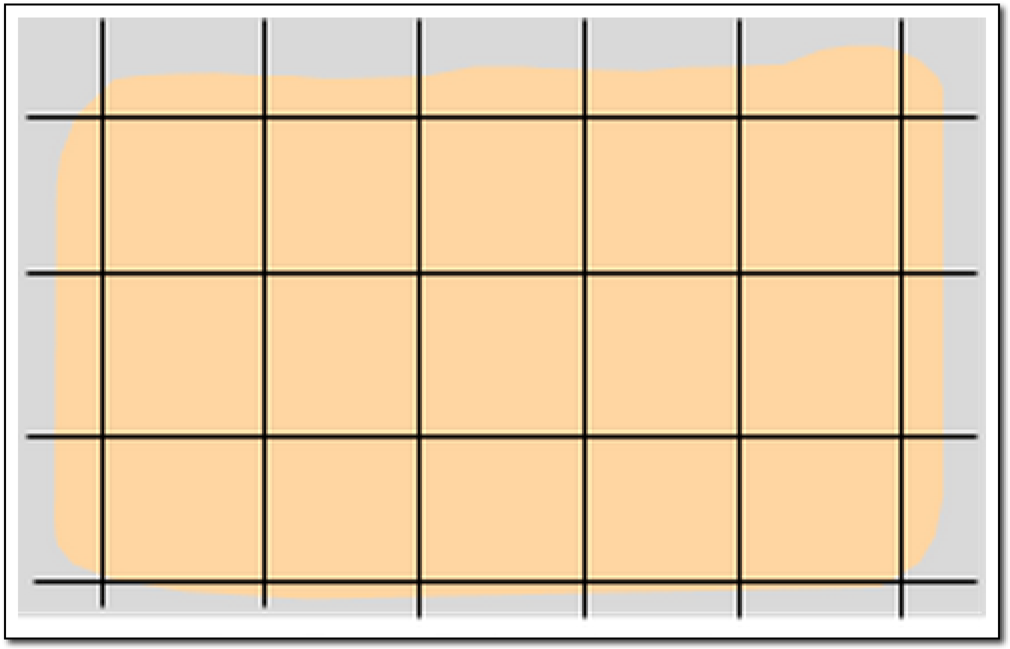
Specifying Search Fields for Individual Layers

Configuring a Searchable Database

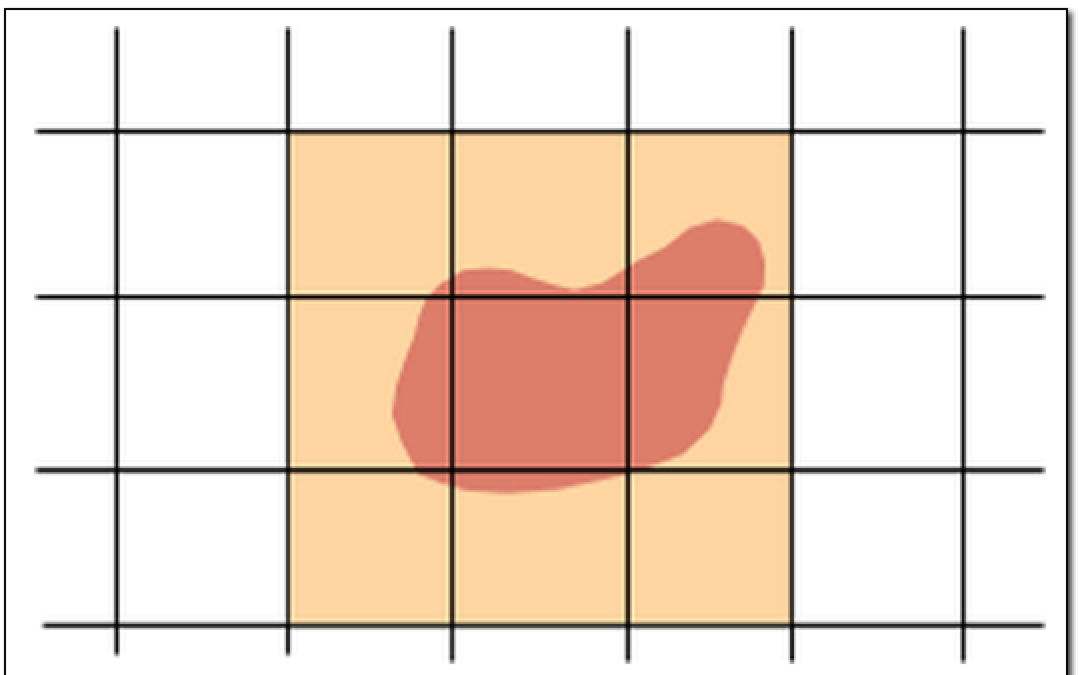
Creating a Map Database

Creating Imagery Mosaics

Segmenting Large Imagery Files

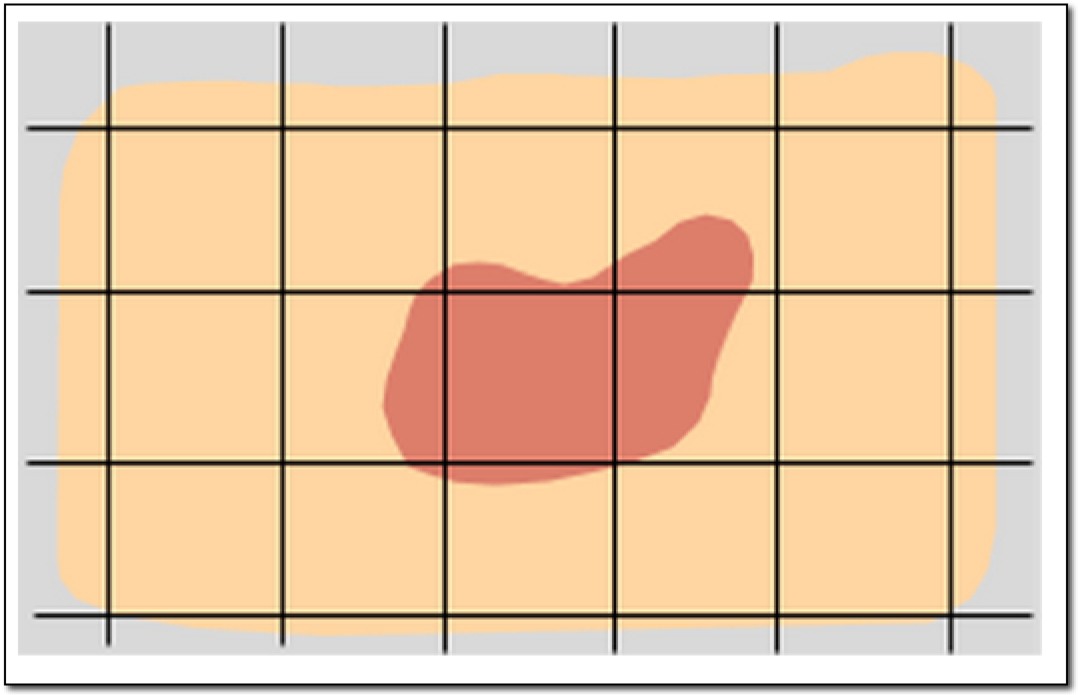


Secondary terrain overlay database with low resolution "fill":



In Google Earth EC, you overlay the secondary database using Add Database, blending the two terrain databases together.

Base terrain database and secondary terrain database blend together with low resolution "fill" created by the terrain overlay:

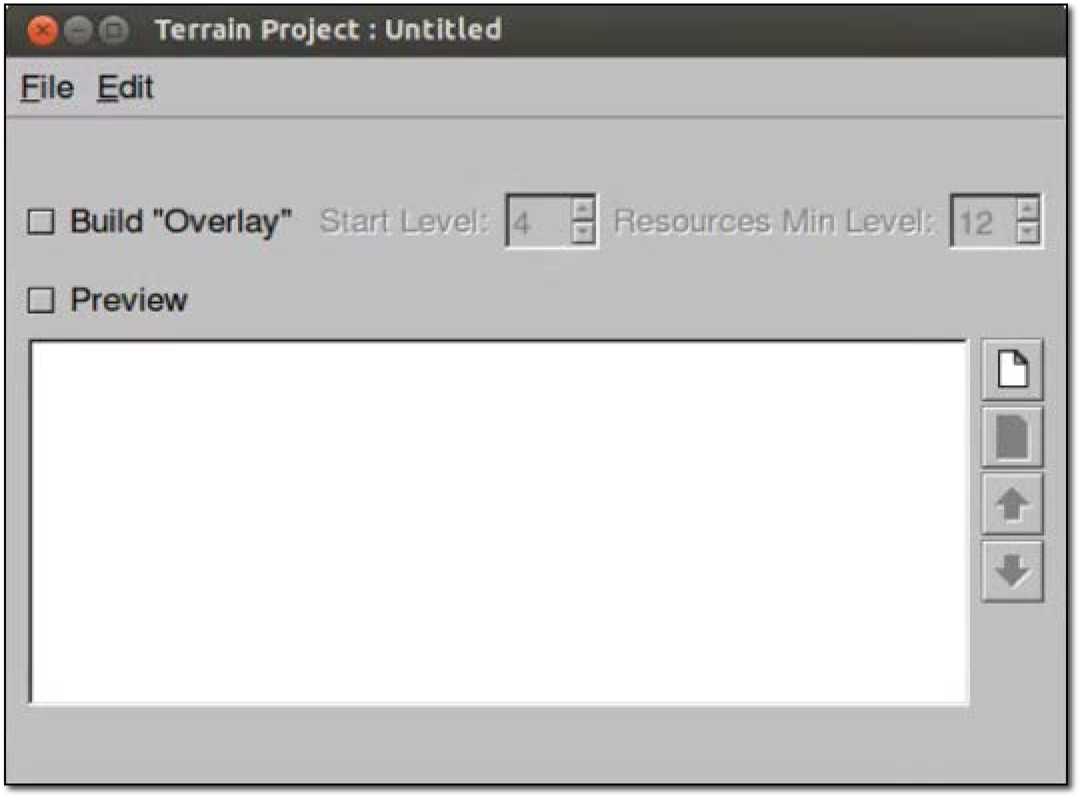


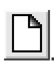
Using the base terrain as the fill for the secondary terrain ensures smooth viewing transitions at the overlay borders.

Creating Terrain Overlays

You create the terrain overlay in the Terrain Project Editor:

1. Select **Tools > Asset Manager**. The **Asset Manager** appears.
2. Click . The **Terrain Project Editor** opens.



3. Click . The **Open** dialog appears.

Note: The selection in the Type drop-down list near the bottom of this dialog determines the type of resources that appear on the list. Terrain Resource is automatically selected when you open this dialog from the Terrain Project Editor, so only terrain resources appear on the list.

4. Navigate to the folder that contains your terrain resources.
5. Select the resource that you want to add to the project and click **Open**. The selected resource appears in the Terrain Project Editor.
6. Repeat steps **3** through **5** for each additional resource that you want to include in your terrain project.
7. Check the **Build "Overlay"** box and specify the **Start Level** and **Resources Min Level** values.

Start Level determines the resolution level from which the terrain overlay will be built. For example, specifying a value of 4 denotes the initial resolution level of 4 is used to build the overlay. You should specify a start level to a value where the terrain inset becomes significant based on the minimal resolution level of the terrain resources used to build the terrain overlay.

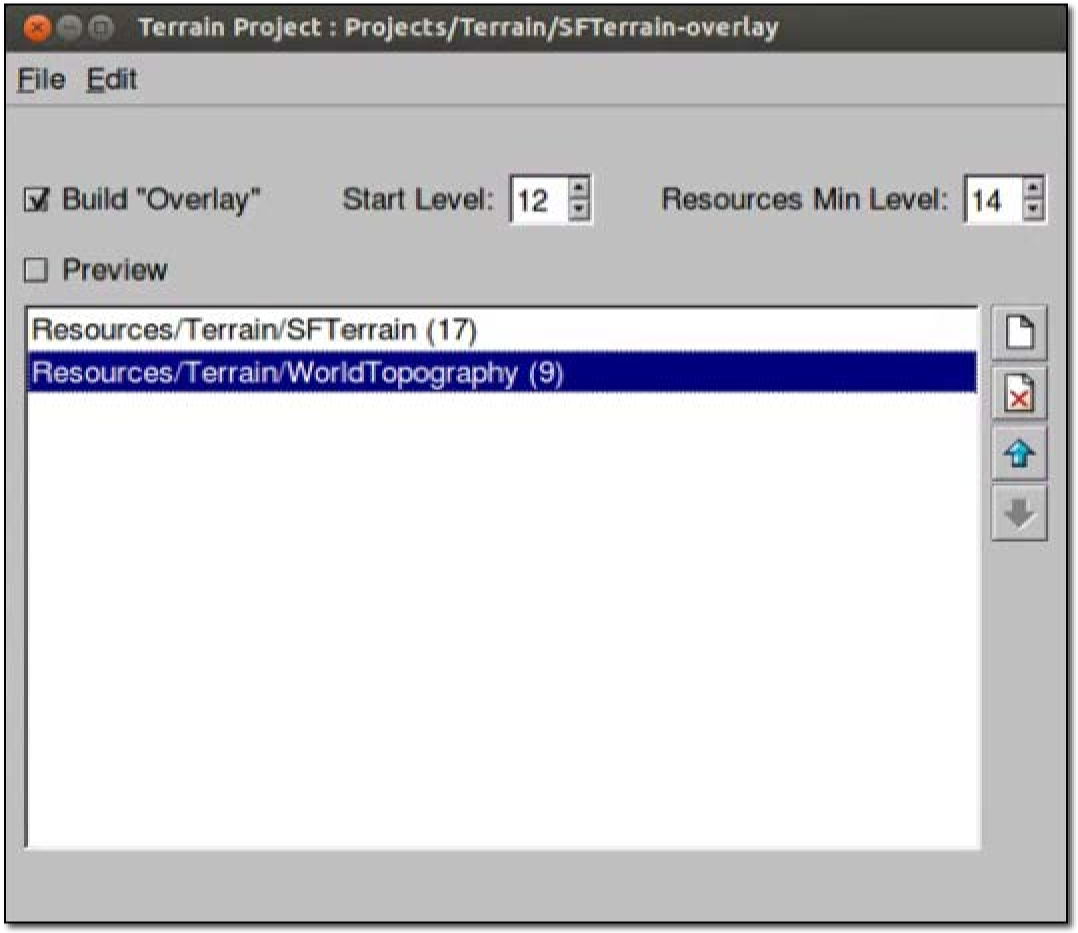
For example, if the resolution level for terrain insets used for building the terrain overlay is 17 while the resolution level for filling terrain is 9, as in the example below, then the **Start Level** might be set to 14 or 16. This would build a terrain overlay from a resolution level of 14 or 16 where the higher resolution terrain will typically begin to be relevant.

In some cases, where smoother transitions across zoom levels are desired, a lower **Start Level** might be used, and in very rare cases where the overlay data differs radically from the base terrain data, the **Start Level** might be set even lower than the fill terrain resolution.

Resources Min Level defines a threshold level that separates the resources into either fill terrain or the actual overlay terrain. Resources with a resolution level that are less than the **Resources Min Level** level are used to create the "fill" for the terrain overlay. Resources with a resolution level at or above the threshold level are used to create the overlay terrain itself. In this example, using the WorldTopography resource with a resolution level of 9 and setting the **Resources Min Level** to 14 specifies WorldTopography as fill in the terrain overlay. Likewise in this example, the SFTerrain resource with a resolution level of 17, higher than the **Resources Min Level** of 14, is used as the actual overlay terrain.

Note: It is the overlay resources and not the fill resources that determine the extent of the overlay.

Note: If every resource is at or above the **Resources Min Level**, then 0 (sea level) is used for the fill terrain. Equally, if you set the **Start Level** to a resolution beyond the extent of the base terrain resource resolution, then 0 (sea level) is used for fill terrain.



8. When you have finished setting the terrain overlay options, click **File > Save**.
9. Navigate to the folder where you want to save your project or click  to create a folder in the desired location.
10. Enter the name of your project, and click **Save**. The new project name appears in the Asset Manager’s asset list.
11. Right-click the project name and select **Build** from the context menu.

    The status of the project immediately changes to **Queued** and then **In Progress**.

    When the status of the build changes to **Succeeded**, go on to add the project to a new database.

Define, Build, and Push a Database for the Terrain Overlay

This exercise walks you through the process of defining, building, and pushing a database using the projects you created in the previous exercise.

To define, build, and push a database:

1. In the Asset Manager click . The Database Editor appears with no projects selected.
2. Click next to Terrain Project. The Open dialog appears.
3. Navigate to the folder where you saved your terrain overlay project.
4. Select the name of your terrain overlay project and click **Open**. The terrain overlay project appears next to Terrain Project in the Database Editor.
5. Select **File > Save**.
6. Navigate to the `/ASSET_ROOT/Databases` folder.
7. Enter the name for your database, and click **Save**. The name of the database appears on the right when you select the `/ASSET_ROOT/Databases` folder in the asset navigation tree.
8. Right-click the name of your database and select **Build** from the context menu. Google Earth Enterprise Fusion builds the database.
9. Right-click the name of your database and select **Push** from the context menu. The Push Database dialog appears.
10. Click **Push**. Google Earth Enterprise Fusion runs the process of pushing the database to the Google Earth Enterprise Server, and displays a success message when it is done.

Publish a Database for the Terrain Overlay

This exercise walks you through the process of publishing your terrain overlay database.

To publish a database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to `myserver.mydomainname.com/admin`, replacing `myserver` and `mydomainname` with your server and domain.
2. Sign in with the default credentials or the username and password assigned to you:
   - Default username: geapacheuser
   - Default password: geeadmin

   Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.

   ---

3. Click **Databases** to display the list of databases pushed to the Server.

   The name of your terrain overlay database is listed but not yet published.

   A pushed database is appended to the list so you may need to click through to the last page of your database list to display it.
4. Check the box next to the name of your database. The **Publish** button appears.
5. Click **Publish**. The Publish dialog appears.
6. Accept the default Publish point and click **Publish**.
7. The Databases page now indicates the published status of your database.

   Now you are ready to add your terrain overlay as a secondary database in Google Earth EC.

View Your Database

After you publish your database, you can add your terrain overlay as a secondary database in Google Earth EC. You first connect to the primary database you want to add the terrain overlay to.

To add your terrain overlay database:

1. Launch Google Earth EC.
2. The Select Server dialog appears.

3. Specify the **Publish point** of the primary database you want to add your terrain overlay to by entering or selecting the URL or IP address of your server and database in the Server field. For example, the Publish point is `myserver.mydomainname.com/mydatabase`, where `myserver` and `mydomainname` are specific to your server.
4. Click **Sign In**.

> **Caution:** If you have logged in to this server with Google Earth EC previously, log out, clear your cache, and log back in. For help with clearing your cache, refer to the Google Earth User Guide.

5. Google Earth EC displays your database. The Layers panel shows any terrain, imagery, and vector layers in the database.
6. Select **File > Add Database**. The Select Server dialog appears.
7. Specify the Publish point of your terrain overlay database by entering or selecting the URL or IP address of your server and database in the Server field. For example, the Publish point is `myserver.mydomainname.com/myterrainoverlay`, where `myserver` and `mydomainname` are specific to your server.
8. Click **Sign In**.
9. Google Earth EC adds your database. The Layers panel shows the secondary database.
10. Check the box next to Terrain to add your terrain overlay. Google Earth EC displays your terrain overlay on top of your existing database.

Learn more

- Defining and Building Projects

Back to top

Apply alpha masking to imagery

Image masking is a method for making areas of an image transparent. Images are defined as rectangular, but their contents can have irregular shapes. For example, an image that contains the land mass for a country has areas that are outside the country's borders. These areas are called fill, or no-data regions, and they are usually black or white. When the image is overlaid on a base map, these areas should be transparent, to allow the base map to show through.

An *alpha mask* is a set of pixels that specifies where to make an image transparent. By default, Google Earth Enterprise Fusion automatically applies a mask when it processes an image but you can also provide custom mask data or prevent masking altogether. The Google Earth Enterprise Fusion **Automask tool** scans an imagery resource, detects fill data from imagery, and then creates a mask to only show the imagery.

GEE 5.x enables this *overlay* approach, making it easy to add high resolution imagery insets without the need to include imagery for the entire globe. In addition to use as an overlay, alpha masking can be used to *blend* imagery layers together.

Prior to GEE 5.x, each imagery layer—built by an imagery project—would comprise of an opaque foundation from which all imagery resources would be added. In **GEE 4.4** and earlier, the following example of high resolution imagery inset with a custom mask includes the *mask* imagery covered with black pixels.

GEE 4.4 and earlier versions applied opaque masks:

As of **GEE 5.x**, each imagery layer—built by an imagery project—is comprised of a *transparent* foundation from which all imagery resources are added. The transparent alpha mask enables you to have imagery resources for specific areas, and to view these as *overlays* or *blends* on other imagery via Google Earth EC. The following image shows the same high resolution inset, this time with an alpha mask applied, so that it can easily be used as an overlay of a secondary database of worldwide imagery.

GEE 5.x applies alpha masking:

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

You can apply an alpha mask in different ways when you build an imagery asset:

- Use **Auto mask** to automatically scan an imagery resource, detect fill data from the imagery, then create a mask to only show the imagery.

> Auto mask works well to mask fill data though it cannot, for example, mask imagery to a coastline since the water would still be considered imagery and not fill. In such cases, create a custom mask.

- Create your own custom mask and use **Have mask** to apply your custom mask.

Learn More

- Defining imagery resources
- Creating custom masks

Back to top

Map projection types in GEE 5.1.0

In addition to using Google Earth Enterprise Fusion to prepare and publish data in Google Earth EC, you can use it to prepare and publish map data in Google Maps.

Google Earth Enterprise 5.1.0 supports the following projection types for map databases:

- **Mercator map**—The Mercator map database includes the Google Maps layer and optionally uses the Google Base Map from google.com. It also uses a local copy of the Google Maps API that requires no communication with google.com.
- **Flat Projection map**—The Flat Projection map database uses the local copy of the Google Maps API that requires no communication with google.com.

Preparing map data

When you prepare map data, there are four components:

- Resources (vector and imagery)
- Map Layers (Flat projection and Mercator maps supported)
- Map Projects (map and imagery)
- Map Database

Vector and imagery resources

For **Flat Projection** maps, you can share the same vector and imagery resources for both Google Earth EC and Google Maps databases.

For **Mercator** maps, you can create Mercator-based imagery projects or your Mercator maps can share the same imagery resources as a Flat Projection map database by projecting the flat imagery as *Mercator on the Fly*. Now that your Mercator map databases can share the same imagery projects as a Flat Projection map databases, you can save on both storage space and database building time by eliminating the need for Mercator-based imagery.

For more information, see Add flat imagery to Mercator map databases.

Access map tools

With map databases, in addition to vector and imagery projects, you have the extra step of creating *map layers*.

Defining a map layer includes adding resources and defining the display rules and filters for the layer.

You access the map tools from the toolbar of the **Asset Manager** dialog:

- Open Fusion and click **Tools > Asset Manager** to open the Asset Manager dialog.
- Select either **Mercator Map Tools (2D)** or **Flat Projection Map Tools (2D)** from the Asset Manager toolbar.

For information about creating map projects and databases, read the tutorial lesson, Creating a map database.

Back to top

Add flat imagery to Mercator map databases in GEE 5.1.0

When you build a Mercator-based map database, you must build imagery projects specifically for use with Mercator-based projection (EPSG: 3857). However, with GEE 5.1.0, if you already have database imagery projects in the Plate Carrée projection (EPSG: 4326), you can re-use them for your Mercator-based map database using *Mercator on the Fly*. When disk space is a key consideration, you only need imagery projects in Plate Carrée projection for 2D flat, 2D Mercator, and 3D databases, reducing the storage requirements for your assets.

Mercator on the Fly converts a Plate Carrée-based imagery project into a Mercator-based imagery project when you create your map database in Fusion. When GEE Server hosts the published Mercator database, the server requests the *Mercator on the Fly* imagery project in the database and serves it with a Mercator projection. With some projections, you may see a reduction in serving speed when projecting a *Mercator on the Fly* imagery project. For more information about performance, see Mercator on the Fly Performance Data.

As you decide which type of imagery project to build, you'll need to consider the projection types you need and whether disk space or serving speed is important. For 2D flat, 2D Mercator, and 3D databases, you can use flat imagery with *Mercator on the Fly* if disk space is a key consideration. However if serving speed is a key consideration, you should build both flat and Mercator imagery projects. The following table shows the decision matrix to guide you:

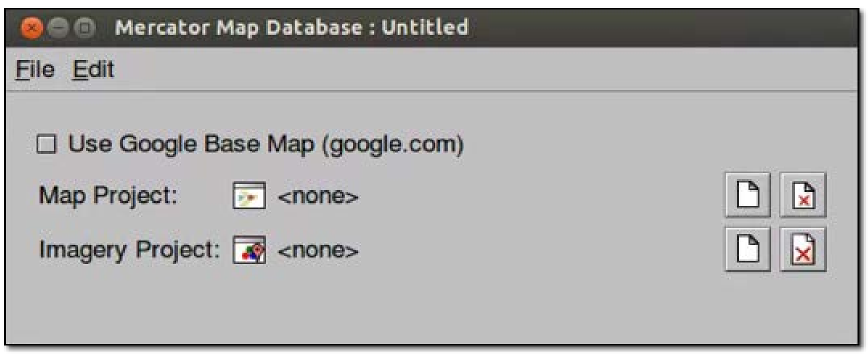
Build projects/serve databases	Mercator	2D flat and 3D	2D flat, Mercator, and 3D
Flat imagery project	X	✔	✔ Disk space is most important. Flat imagery projects used for all database types, with *Mercator on the Fly* used to serve flat imagery for Mercator map databases.
Mercator imagery project	✔	X	

Flat and Mercator imagery projects	✓	✓	✓ Serving speed is most important. Flat imagery projects used for 3D and 2D flat databases; Mercator imagery projects for Mercator map databases.

Of course, if you’re serving imagery *only* as Mercator map databases, you should only build Mercator imagery projects.

To add a flat imagery resource to a Mercator-based map database:

1. Select **Asset Manager** from the **Tools** menu. The Asset Manager dialog appears.
2. Select **Mercator Map Tools (2D)** from the Asset Manager toolbar. The tools for Mercator maps appear in the toolbar.
3. Click . The Map Database Editor appears with no projects selected.



4. Click next to **Map Project**. The Open Asset dialog appears.
5. Navigate to the `ASSET_ROOT/Projects/Map` folder.
6. Select the map project that you want to add to the map database and click **Open**. The map project appears in the Map Database Editor next to Map Project.
7. Click next to **Imagery Project**. The Open Asset dialog appears.
8. Navigate to the `ASSET_ROOT/Projects/Imagery` folder.
9. Select **Flat Imagery Project** from the **Type** drop-down list, then select the imagery project to add and click

Building a Historical Imagery Project

Sample Data Files

Configure tutorial workspace

Create terrain overlays

Apply alpha masking to imagery

Map projection types in GEE 5.1.0

Add flat imagery to Mercator map databases in GEE 5.1.0

Manage mosaics with virtual rasters

Open. The imagery project appears in the Map Database Editor next to Imagery Project.

10. Select **File > Save**.
11. Navigate to the `ASSET_ROOT/Databases` folder.
12. In the **Name** field, enter a name, and click **Save**.
13. In the Asset Manager, the new map database appears, along with the other databases you have created. Notice that the **Category** column distinguishes between the Google Earth databases and the Google Maps database.
14. Right-click the name of your map database, and select **Build** from the context menu. The status of the database immediately changes to **Waiting** or **Queued** and then to **In Progress**.

    When Google Earth Enterprise Fusion finishes building the database, its **Current State** column in the Asset Manager changes to **Succeeded**, and its **Current Version** column changes to the date and time the most recent build was started.

Mercator on the Fly performance data

Mercator resolution increases relative to Plate Carrée as latitude increases (or decreases) from the equator for any given map level . Therefore to obtain the correct Mercator resolution, tiles must be sampled from higher Plate Carrée levels (known as upsampling). The higher the latitude, the greater the number of tiles required and, as a result, processing time for *Mercator on the Fly* increases although not proportionally. The following table lists the number of Plate Carrée tiles required to produce a single *Mercator on the Fly* tile vs. latitude (north or south) and the approximate expected increase in processing time assuming cached tiles.

Approximate Latitude Range(degrees)	Level Up to sample	MotF P.C. tiles processed	Max Relative Clock Time Ratio (relative to 1 tile)
0 to 40	1	1-2	1 to 1.2 (2 tiles)
40 to 65	2	up to 4	2.1 (4 tiles)
56 to 75	3	up to 8	3.2 (8 tiles)
75 to 85	4	up to 16	5.1 (16 tiles)

Learn more

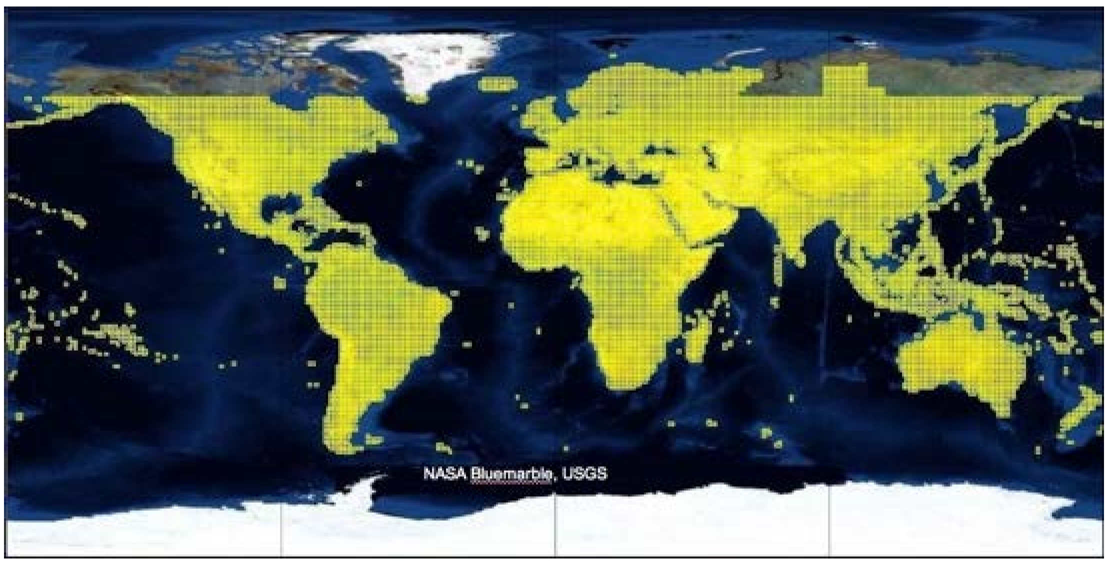
- Push and publish databases
- Make Web Map Service (WMS) requests

Back to top

Manage mosaics with virtual rasters

When you have large data sets of imagery or terrain resources, importing them into Fusion can be inefficient and lead to gaps between data tiles. GEE provides a tool, `gevirtualraster`, which is ideal for grouping blocks (or subsets) of contiguous imagery or terrain and creating a *virtual mosaic* from them.

In the following example, this SRTM 90meter terrain data set has over 14,476 source files.



This example of a large set of terrain data presents three potential ways to import all large data sets of source files:

- Create one resource for all source TIF tiles. This data processing is inefficient as Fusion must fill all missing data between source terrain tiles (oceans).
- Create one terrain resource per tif file (14, 476 in total). This data processing is inefficient and introduces

feather gaps between terrain resources due to no overlap between source files.

- Group the 14,476 files into virtual mosaics enabling you to select which files are grouped into mosaics.

The most efficient method of grouping tiles into virtual mosaics is to use `gevirtualraster`, a command line tool that lets you access the tile groups directly, making your mosaic management very flexible. `Gevirtualraster` validates that each source file has geographic coordinates and that all files are consistent projections, number of bands, and band formats.

Overview

A *virtual mosaic* or virtual raster file is a simple text file that describes a mosaic of peer raster files with the following elements:

- Projection parameters (all peers with same projection)
- Coordinate location of the overall raster (sum of parts)
- Pixel dimensions
- Band count and format (3 x 8 bit vs. 1 x 8 bit)
- File path of all included source files

Fusion can display the virtual raster file in **Preview** for simple visualization.

Fusion can also use the virtual raster file as source.

Note: `Gevirtualraster` is necessary if you are creating a custom mask file for the imagery or terrain resource as there is a software requirement of one source file for one custom mask file. See Create custom masks.

Example Usage

This end-to-end example uses `gevirtualraster` to create a new Fusion resource from a set of 800+ contiguous imagery resources (jpeg2000) with consistent pixel size, projection, etc.

1. List the jpeg2000 files that are to be used for the virtual mosaic:

```
$ ls -l *.jp2 | wc -l
```

```
863
```

2. Create a virtual raster file, specifiying the fill value to be used for the mosaic (`0,0,0 = black`), the name of the file to be generated (`my_virtualmosaic_20140624_1.khvr`), and include all of the `.jp2` files in the current folder as input.

   ```
   $ gevirtualraster --fill 0,0,0 --tolerance 3 -o my_virtualmosaic_20140624_1.khvr
    *.jp2
   ```

3. Import the virtual raster imagery file into Fusion, specifying the resource type, the virtual raster file to import and the output resource file, `$ my_virtualmosaic_20140624.kip`.

   ```
   gerasterimport --imagery my_virtualmosaic_20140624_1.khvr -o
    my_virtualmosaic_20140624.kip
   ```

4. Create a mask for the imagery resource, specifying band, fill, tolerance and feather values, and the imagery resource file and the name of the mask created.

   ```
   $ gemaskgen --mask --band 1 --fill 0 --tolerance 3 --feather 100
    my_virtualmosaic_20140624_1.kip my_virtualmosaic_20140624_1-mask.tif
   ```

5. Import the generated alpha mask, creating a `.kmp` folder. Corresponding `.kip` and `.kmp` folders should be stored in the same directory.

   ```
   $ gerasterimport --alphamask my_virtualmosaic_20140624_1-mask.tif --imagery
    my_virtualmosaic_20140624_1.kip --output my_virtualmosaic_20140624_1.kmp
   ```

6. Define a new imagery resource.

   ```
   $ genewimageryresource -o Resources/Imagery/candid_my_virtualmosaic_20140624_1
    /gevol/src/candid/my_virtualmosaic_20140624_1/my_virtualmosaic_20140624_1.kip
   ```

7. Build the new imagery resource.

   ```
   $ gebuild Resources/Imagery/candid_my_virtualmosaic_20140624_1
   ```

For more information about using virtual mosaics, see Create virtual mosaics.

Creating a virtual raster from a file list

You can create a virtual raster of all source files stored in a test file using the `gevirtualraster --filelist` option:

```
$ gevirtualraster --fill 0,0,0 -o /gevol/src/imagery/example/example-mosaic-list.khvr --filelist /gevol/src/imagery/example/source-files.txt
```

The file list must include one column of filenames and the full path of each source file must be included in the file list.

Gevirtualraster command line options

Various options are available to facilitate the creation of virtual raster files, including specifying the band values to be used as fill, tolerance values for the fill, and a file list of the source imagery or terrain files.

gevirtualraster [**--crop** *pixelx,pixely,pixelx,pixelh*] [**--fill** *a,b,...*] [**--src** *override_srs*] [**--tolerance** *num*] [**--validate**] **-o** *output*.khvr {**sourcefile** ... | **--filelist** file}

Example

```
gevirtualraster --fill 0,0,0 -o /gevol/src/imagery/usa/XX/new_location/all_files.khvr/gevol/src/imagery/usa/XX/new_location/*.tif
```

Parameters

Header	Header
--crop *pixelx,pixely,pixelx,pixelh*	*Optional*. Crop the image to the specified pixel extents.
--fill *a,b,...*	*Optional*. Specify band values to use as fill.
--src *override_srs*	*Optional*. Specify the SRS.
num	

--tolerance	*Optional*. Specify the tolerance to be applied to the fill. The default is 0.
--validate	*Optional*. Validate the inputs and exit.
-o *output*.khvr	*Required*. Specify the name of the output file, which must have the `.khvr` extension.
sourcefile	*Required*. Specify the path and file name of the source file for the resource. You can reference any network-available source file as this value. (Optional if you specify **--filelist** file.)
--filelist file	*Optional*. Specify the path and file name of a file that contains a list of source files that you want to include in the resource. You can use this option, list files individually, or use a combination of the two.

Splitting large virtual raster mosaics

If you create larger virtual raster mosaics, you may want to split them into smaller subsets. You can use the `gesplitkhvr` command line tool to automatically split large `.khvr` mosaics into subsets by specifying the number of rows and columns.

gesplitkhvr [**--rows** *num*] [**--cols** *num*] [**--overlap** *num*] [**--quiet**] ***input*.khvr**

The `gesplitkhvr` command creates two or more source files from a large imagery or terrain source file to reduce the size of each source file to under 80 GB in raw size. (Raw size = number of pixels width * number of pixels height * 3.) This tool produces a grid of image files designated as rows and columns.

Example

```
gesplitkhvr --rows 4 --cols 4 --overlap 10 image_file.khvr
```

Parameters

Header	Header

`--rows` *num*	*Required*. Specify the number of resulting image files across.
`--cols` *num*	*Required*. Specify the number of resulting image files high.
`--overlap` *num*	*Optional*. Specify the number of pixels of overlap between the resulting images. The default is 300.
`--quiet`	*Optional*. Do not display the progress messages in the terminal window.
input`.khvr`	*Required*. Specify the name of the input file. It must be a `.khvr` file.

Learn more

Create virtual mosaics

Segment large imagery files

Create custom masks

Back to top

Before you configure Google Earth Enterprise Fusion

Before you use many of the Google Earth Enterprise 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]
```

Back to top

Configure 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 Google Earth Enterprise 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. See Configure tutorial workspace to learn how to configure and select an asset root for the tutorial, as well as how to clean up tutorial files.

- **Identify source volumes.**

  You must identify each directory that contains your source data files (or subdirectories of your source data files). Google Earth Enterprise 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**.

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 Google Earth Enterprise 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**.

Back to top

Google

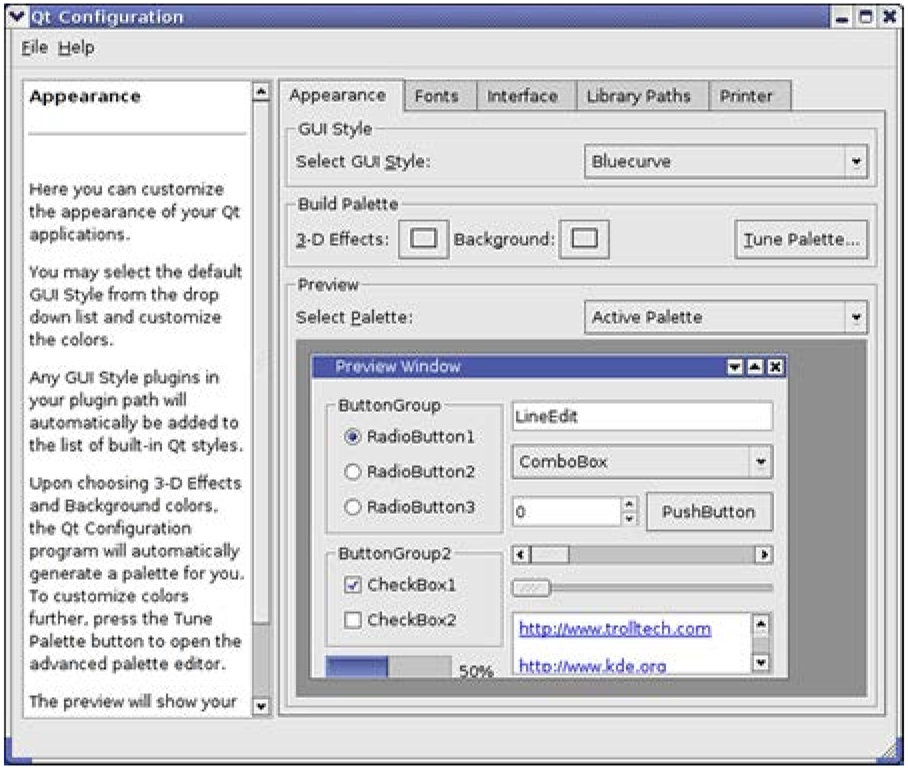
Configure font size in the GUI

If you want to customize certain aspects of the Google Earth Enterprise 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 Google Earth Enterprise Fusion GUI:

1. On the command line, change directory to `/opt/google/qt/bin/` and 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.

Back to top

Configure fonts for the Text Style dialog

When you are specifying a map layer in Google Earth Enterprise Fusion, the Text Style dialog allows users to select the text style for vector labels. (See Specifying Search Fields for Individual Layers.) Google Earth Enterprise Fusion supplies only one font and style, *Sans regular*. However, if you want to provide a choice of fonts for your users to select in the Text Style dialog, you can create a font list file that contains information about the fonts you want to support. This section explains how to create a font list file.

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 on 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
```

Back to top

Configure task rules

You can use *task rules* to change how Fusion builds data. As soon as you implement a task rule, it customizes how specific processes in Fusion operate. In other words, you can't apply task rules to individual assets or projects.

- Set the number of CPUs assigned to Fusion processing
- Task rule types
- Implement task rules

Set the number of CPUs assigned to Fusion processing

As some of the task rules described here rely on setting the number of CPUs assigned to Fusion processing, it's useful for you to know how many CPUs are available on your machine and how many are assigned for Fusion processing. The `getop` output lists the available CPUs on your machine as the `maxjobs` entry and corresponds to how many concurrent jobs Fusion may spawn at any one time.

Note: The standard Fusion installation includes a license that allows you to run eight concurrent tasks/jobs. To use more than eight CPUs for Fusion processing, purchase an additional license by contacting your Google Account Manager.

You can use the tool `geselectassetroot --assetroot path --numcpus` to assign more CPUs. For example:

```
geselectassetroot --assetroot /gevol/assets --numcpus 7
```

For information about assigning CPUs to Fusion processing, see Running Fusion on a Machine with Multiple CPUs.

Task rules impact Fusion processing in two different ways: by enabling/disabling multithreading, and by specifying output locations (KRP/KRMP), as shown in the following table.

Task rule types

Task rule type	Task rule	Description
Enable/Disable Multithreading	**ImageryPacketLevel**	Makes imagery project builds faster by assigning multiple CPUs to each task.
Enable/Disable Multithreading	**TerrainPacketLevel**	Makes terrain project builds faster by assigning multiple CPUs to each task.
Enable/Disable Multithreading	**CombinedTerrain**	Makes 3D database builds faster by assigning multiple CPUs to terrain completion tasks.
Enable/Disable Multithreading	**MapLayerLevel**	Makes 2D database builds faster by assigning multiple CPUs to each task.
Specify output locations	**KRP**	Lets you specify the location of where image resources will be written by Fusion for better storage management.
Specify output locations	**KRMP**	Lets you specify the location of where the mask files will be written by Fusion for better storage management. Note that the same storage volume must be specified for the KRMP as the KRP.

You can add one or more of these task rules to your asset root's `.config` directory.

Implement task rules

To implement a task rule:

1. Make sure you have superuser privileges, that is, root or sudo access.
2. Create the task rule using the XML sample code shown below for each rule.
3. Depending on the rule you want to use, name the task rule:
    - **ImageryPacketLevel.taskrule**
    - **TerrainPacketLevel.taskrule**
    - **CombinedTerrain.taskrule**
    - **MapLayerLevel.taskrule**
    - **KRP.taskrule**

- **KRMP.taskrule**

Note: Fusion uses these names when the tasks run, so don't change them.

4. For the KRP and KRMP task rules, define the volumes you want Fusion to write files to. See Defining volumes.
5. Set file permissions to read only for all (`chmod 444`).
6. Set file ownership to the `gefusionuser` user account and `gegroup` group (`chown gefusionuser:gegroup`).
7. Place the task rule files in the asset root's `.config` directory.
8. Restart the system manager using the `/etc/init.d/gefusion` **`stop`** and **`start`** commands.

ImageryPacketLevel task rule

During a project build, Fusion will have multiple concurrent imagery project tasks. The watchpoint is balancing the total number of CPUs assigned for Fusion, and the number of CPUs assigned for each imagery project process. For example, with 8 CPUs assigned for Fusion, with 2 CPUs per imagery project task, you can have 4 concurrent imagery project tasks with 2 CPUs per imagery project task.

Note: Assigning more than two CPUs for each imagery project task doesn't necessarily speed up processing, so you should keep the `maxNumCPU` value at 2.

Sample: ImageryPacketLevel.taskrule

```
<TaskRule>
  <taskname>ImageryPacketLevel</taskname>
  <inputConstraints/>
  <outputConstraints/>
  <cpuConstraint>
    <minNumCPU>2</minNumCPU>
    <maxNumCPU>2</maxNumCPU>
  </cpuConstraint>
 </TaskRule>
```

TerrainPacketLevel task rule

During a project build, Fusion will have multiple concurrent terrain project tasks. The watchpoint is balancing the total number of CPUs assigned for Fusion, and the number of CPUs assigned for each imagery project process. For example, with 8 CPUs assigned for Fusion, with 2 CPUs per terrain project task, you can have 4 concurrent terrain project tasks with 2 CPUs per terrain project task.

By default, Fusion assigns one CPU to work on each terrain project task. You can use this task rule to assign up to six CPUs instead. Set the minimum CPU value to a number from 1 to 6, and set the maximum to a number between 2 and 6.

Sample: TerrainPacketLevel.taskrule

```
<TaskRule>
  <taskname>TerrainPacketLevel</taskname>
  <inputConstraints/>
  <outputConstraints/>
  <cpuConstraint>
    <minNumCPU>1</minNumCPU>
    <maxNumCPU>6</maxNumCPU>
  </cpuConstraint>
 </TaskRule>
```

CombinedTerrain task rule

Since `gecombineterrain` is a serial process and must be completed before the other database-level building activities can start, you can use the CombinedTerrain task rule to assign all available CPUs to the build. For example, with 8 CPUs assigned for Fusion, assuming that one CPU is reserved for the system, you can assign 7 CPUs to the CombinedTerrain serial process.

Sample: CombinedTerrain.taskrule

```
<TaskRule>
  <taskname>CombinedTerrain</taskname>
  <inputConstraints/>
  <outputConstraints/>
  <cpuConstraint>
    <minNumCPU>7</minNumCPU>
    <maxNumCPU>7</maxNumCPU>
  </cpuConstraint>
 </TaskRule>
```

MapLayerLevel task rule

By default, Fusion assigns one CPU to work on map tile generation. As `maptilegen` can be run concurrently, you can use this task rule to assign multiple CPUs instead. Set the rule to min/max 4 on a 2 dual-core CPU system (3 may work well enough). On a 2 quad-core CPU system, set the rule to min/max 3.

Sample: MapLayerLevel.taskrule

```
<TaskRule>
  <taskname>MapLayerLevel</taskname>
  <inputConstraints/>
  <outputConstraints/>
  <cpuConstraint>
    <minNumCPU>4</minNumCPU>
    <maxNumCPU>4</maxNumCPU>
  </cpuConstraint>
 </TaskRule>
```

KRP task rule

When Fusion creates an asset, it writes large imagery files to the raster.kip directory in asset root by default. You can manage your disk space by using the KRP task rule to write these files to a different volume. See Example of File Locations.

In the sample below, the volume you've created is `products`. To learn how to create a new volume, see Defining volumes.

Sample: KRP.taskrule

```
<TaskRule>
  <taskname>KRP</taskname>
  <inputConstraints/>
  <outputConstraints>
    <outputConstraint>
      <num>0</num>
      <requiredVolume>products</requiredVolume>
      <pathPattern>${assetref:dirname:sansext}-
     ${vernum}${defaultpath:ext}</pathPattern>
    </outputConstraint>
  </outputConstraints>
  <cpuConstraint>
    <minNumCPU>1</minNumCPU>
    <maxNumCPU>1</maxNumCPU>
  </cpuConstraint>
 </TaskRule>
```

You can also enter a second `requiredvolume` tag in your task rule to specify the next available volume in case the first volume fills up. However, you'll need to restart the system manager to trigger Fusion to write files to the next volume.

KRMP task rule

When Fusion creates an asset, it writes the mask files to the `mask.kmp` directory in asset root by default. You can manage your disk space by using the KRMP task rule to write these files to a different volume. See Example of File Locations. In the sample below, the volume you've created is `products`. To learn how to create a new volume, see Defining volumes.

Sample: KRMP.taskrule

```
<TaskRule>
  <taskname>KRMP</taskname>
  <inputConstraints/>
  <outputConstraints>
    <outputConstraint>
      <num>0</num>
      <requiredVolume>products</requiredVolume>
      <pathPattern>${assetref:dirname:sansext}-
     ${vernum}${defaultpath:ext}</pathPattern>
    </outputConstraint>
  </outputConstraints>
  <cpuConstraint>
    <minNumCPU>1</minNumCPU>
    <maxNumCPU>1</maxNumCPU>
  </cpuConstraint>
 </TaskRule>
```

You can also enter a second `requiredvolume` tag in your task rule to specify the next available volume in case the first volume fills up. However, you'll need to restart the system manager to trigger Fusion to write files to the next volume.

Defining volumes

The KRP and KRMP task rules include a `requiredvolume` tag that indicates where to write files. You need to make sure that this volume is defined and has enough space to hold the files. To define the volume, use the `geconfigureassetroot -- editvolumes` command, then edit the XML as shown in the sample below. In this sample, the volume you're creating is called `products`.

Sample: volumes.xml

```
<products>
  <netpath>/gevol/products</netpath>
  <host>linux</host>
  <localpath>/gevol/products</localpath>
  <reserveSpace>100000</reserveSpace>
  <isTmp>0</isTmp>
 </products>
```

Example of File Locations

The table below shows an example of the default file locations and the locations after applying the KMP and KRMP task rules. In this example, the asset root is `/gevol/assets`, the volume you've created is `products`, and you've built an asset called `TestImage` in the imagery subdirectory.

Task rule	File type	Default location	Location with task rule
KMP	large imagery files	`/gevol/assets/imagery/TestImage.kiasset/products.kia/ver001/raster.kip`	`/gevol/products/imagery/TestImage-001.kip`
KRMP	smaller mask files	`/gevol/assets/imagery/TestImage.kiasset/maskproducts.kia/ver001/mask.kmp`	`/gevol/products/imagery/TestImage-001.kmp`

Back to top

Benchmarking outcomes for applying task rules

This article describes the benchmarking outcomes for applying various task rules to a test set of data under a similar environment. The results are useful in providing guidance in how task rules can improve GEE performance of loading assets and build times.

This set of tests was designed to help assess how GEE performs in a specific environment. Although certain conclusions can be drawn from the test results, your results may differ. The figures presented here can be used as a reference for the GEE performance only. This testing was conducted with GEE 4.4 using publicly available geospatial data to build a 3D database.

GEE System Specifications

The following table lists the software and hardware specifications that were used for testing.

OS	Ubuntu 12.04.1 LTS
Processor	8x Intel(R) Xeon ® CPU @2.60GHz
CPU	2599.998 MHz
Cache size	20488 KB
Memory	30 GB

This set of tests used raster data, terrain data and vector data that are freely available online. Some of them provide global coverage and others provide local high resolution coverage.

Fusion Server Configurations - Task Rules

A number of task rules can be configured to increase the processing speed. You configure task rules settings to assign a number of CPUs to processes that can be run concurrently.

Fusion Server has the following two binaries which are multithreaded to increase processing speed when building a 3D database:

- `gepackgen`
- `gecombineterrain`

You can identify the activity of these binaries during the processing of your databases by checking the log files listed in the Version Properties dialog. (Right-click an asset in the Asset Manager dialog.)

`gepackgen` generates imagery or terrain project packets. Multiple `gepackgen` processes can run concurrently to speed up the building of the imagery projects. `gecombineterrain` processes 3D databases and merges terrain packets from the specified indices.

Task rules can be configured based on the number of concurrent processes required to be run on the Fusion server. Since `gecombineterrain` is a serial process and must be completed before the other database-level building activities can start, the **CombinedTerrain** task rule is used to put all available CPUs/cores onto the build, assuming that one core is reserved for the system.

The **ImageryPacketLevel** and **TerrainPacketLevel** task rules configure `gepackgen` for concurrent processing. Both assign more or less CPUs to the pack files that are created during project builds.

The task rules apply to all subsequent tasks that the Fusion server performs, and can't be applied to individual assets or projects.

To compare the effect of different task rule settings on the performance of the Fusion server when loading assets and building resources, projects, and databases, three different task rule configurations were used: *default taskrule*, *taskrule 2* and *taskrule 3*. *Default task rule* is the GEE default setting after installation.

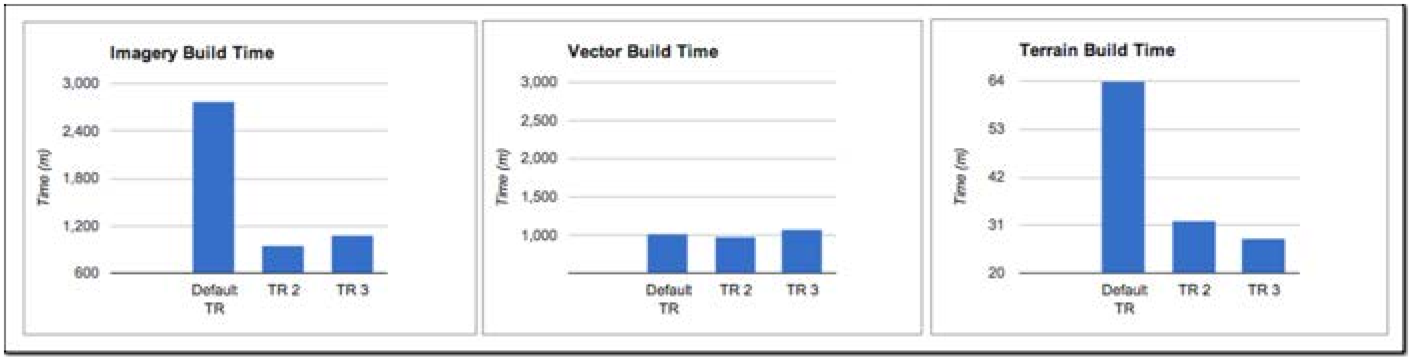
Detailed settings are in the following table.

	Default taskrule	**Default taskrule**	**Taskrule 2**	**Taskrule 2**	**Taskrule 3**	**Taskrule 3**
	Min CPU	Max CPU	Min CPU	Max CPU	Min CPU	Max CPU
CombinedTerrain	1	1	7	7	7	7
ImageryPacketLevel	1	1	3	3	2	2
TerrainPacketLevel	1	1	3	4	3	4

In this set of tests, the server was permitted to use 7 CPU cores for Fusion processing; take Taskrule 2 for example, there can be up to 2 gepackgen tasks running concurrently with 3 CPUs assigned for each task.

Project build time under different taskrules

Total raster, terrain and vector project build times are illustrated in the following charts. The vector project build time doesn't change much at all because the task rule settings have no impact on the `gevectorfuse` command. For the imagery project, *taskrule 2* provides the best build performance. For the terrain project, *taskrule 3* provides the best build performance.



Total database build time

The database build time with *task rule 3* was the fastest - reducing the time to 25.31 hours. Under the *default task rule* setting, the total database build time took the longest time to finish, 33.52 hours.

Recommendations

- Always set the task rule other than default to have better image project build performance.
- With this test dataset, *task rule 3* provides the shortest total database building time on this server - with 2 imagery `gepackgen` tasks running concurrently with 2 CPUs assigned for each task; with all available CPU assigned to **CombinedTerrain** and 2 terrain `gepackgen` tasks running concurrently with 3 or 4 CPUs assigned for each task.
- Set **CombinedTerrain** to put all available CPUs/cores onto the build.
- Select a machine with more memory if processing vector data as fusing vector data (`gevectorfuse`) can consume a lot of memory.

Learn more

- Configure task rules
- Running Fusion on a Machine with Multiple CPUs

Back to top

Customize your Google Maps landing page

Google uses 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 Google Earth Enterprise 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 Javascript API is at Google Maps Javascript API V3 Reference.

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 host on which you can experiment, and move the copied files to that virtual host. When you create the Apache configuration file for the new virtual host, 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 host:

```
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 Google Earth Enterprise Fusion that are not supported by some browsers on certain operating systems.

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
   ```

Back to top

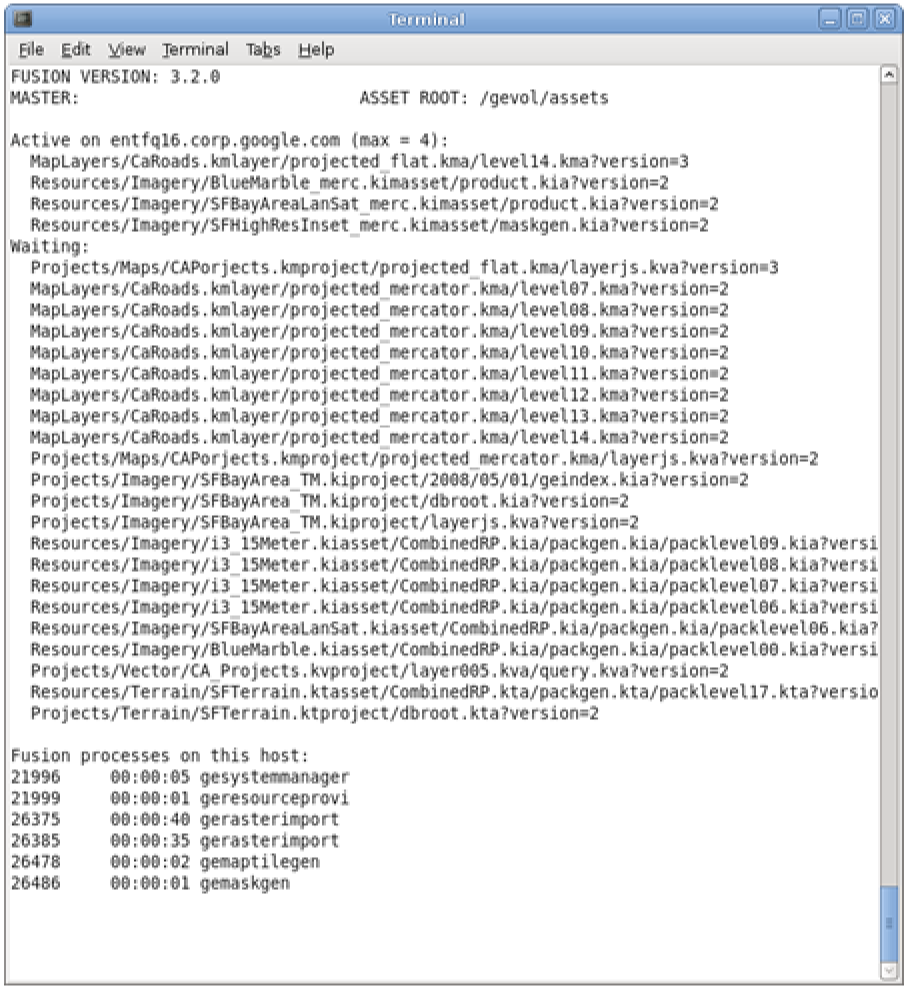
Monitor system processes

You can watch all active Google Earth Enterprise Fusion system tasks and processes using the `getop` command. This command is similar to the Unix command `top`, but is specific to Google Earth Enterprise Fusion processing. It is the command line equivalent to the System Manager in the Google Earth Enterprise Fusion GUI. See getop in the **Command Line Reference** for more information.

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

- Version of Google Earth Enterprise 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 Google Earth Enterprise 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 gemaptilegen
26486     00:00:01 gemaskgen
```

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

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 Google Earth Enterprise 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.

Back to top

Back up Fusion servers

The Fusion server typically stores multiple terabytes of source data and intermediate assets. If you have a tape backup system or enough disk space, you should perform a full back up of the server anytime a new database is built or a significant amount of new data is processed. If you don't have enough resources to do this, you can back up just your critical files and display rules instead.

Display rules

Typically, you'll spend quite a bit of time making your display rules exactly the way you want them, so you should back up the display rules each time you modify a vector layer.

To back up display rules:

1. In the Vector Layer dialog box, select **Export as a Template**.
2. Save the display rules on a volume different from the source volume and the asset root.

Critical Fusion files

If you have backups of the files listed below, you can rebuild your database with some patience and scripting. The rebuild will take a lot of CPU time, but should go smoothly with a minimum of administrator time.

The files that are critical to restoring your Fusion server are:

- **Your original source files**. These are typically stored on a network shared drive or on portable USB disks, depending on the environment. Make sure that you can find and re-load your original source files.
- **The XML files that Fusion generates during its processing**. These are critical because they contain the display rules for the vector data, the rules for building resources and projects, and other critical settings.

- **All the files under /opt/google/gehttpd in the cgi-bin, conf conf.d, and htdocs directory**. These files contain any user-customized scripts or web pages, and the definitions of any virtual servers you created on the Fusion server.
- **The files in the .config and .userdata directories under the asset root**. These contain custom icons and provider information as well as some basic Fusion configuration settings.

Backing up

The steps below show you how to back up your critical files. When you run these commands, replace the asset root and backup locations with the locations of your own asset root and backups.

1. Make sure that you can locate and re-load all the source data files if needed.
2. Create a directory to hold the backup files:

```
mkdir -p /data/backup/assets
mkdir -p /data/backup/gehttpd
```

3. Back up the gehttpd files:

```
cd /opt/google/gehttpd
rsync -r htdocs cgi-bin conf conf.d /data/backup/gehttpd
```

4. Back up the XML files and config files to a safe place on a disk that is not part of the asset root or source directories:

```
cd /gevol/assets
rsync -r .config .userdata /data/backup/assets/
find . -name "*.xml" > FileList.txt
rsync --files-from=FileList.txt /gevol/assets /data/backup/assets/
```

5. (Optional.) Store a compressed backup of the files somewhere else:

```
tar czf /home/backups/GevolXmlBackup.tgz /data/backups
```

Restoring imagery and terrain

1. Make sure that the Fusion software is installed correctly and is running.

2. Configure your asset root to be consistent with your old asset root. If you are not sure about this, manually run commands like:

   ```
   geconfigureassetroot --new --assetroot /gevol/assets
   geselectassetroot --assetroot /gevol/assets
   ```

3. Restore the backed-up XML and configuration files to their original location in the asset root.

4. Restore the source files to their original location on the Fusion server. If you can't remember the directory they were in, just place them on the Fusion server in the source directory you specified in the Fusion volume definitions (`/gevol/src` by default).

5. Use the `gequery` command and shell scripting to make sure your source files are in the right place. If the files are not found, find them and copy them to the right place. Below is an example of a script for reporting where all your imagery source files should be located. You can use the same script for terrain resources if you change `kiasset` to `ktasset`. Example script:

   ```
   cd /gevol/assets
   for resource in `find . -type d -name "*.kiasset"`
   do
       for file in `gequery --infiles $resource`
       do
           if [ ! -f $file ] && [ ! -d $file ]
           then
               echo "Can not find $file"
           else
               echo "Found $file"
           fi
       done
   done
   ```

6. After all the files are in place, generate a series of commands to rebuild the resources. For example:

   ```
   cd /gevol/assets
   for resource in `find . -type d -name "*.kiasset" | sed 's/.kiasset//'`
   ```

```
do
    gequery --infiles $resource > filelist
    genewimageryresource -o $resource --filelist filelist
    sleep 1
    gebuild $resource
done
```

7. There is a `khasset.xml` file under each `.kiasset` directory that contains information about the mask tolerance, feathering, white fill, and provider for each imagery and terrain resource. You can use scripting to extract this information from each resource and add it to the `genewimageryresource` command to apply the correct settings for each resource.

Restoring vectors

The vector restore process is different from the imagery and terrain process because of the display rules that each vector layer has. These display rules take a very long time to configure, and typically the Fusion administrator hasn't saved the display rules as a template. Without the XML files, the vector project can take many hours to re-create. Fortunately, all the display rules for each vector layer are contained in a single file called `khasset.xml`. This file is under the vector project's main directory, and is included in the backup script listed above. The `restore_vector_project.pl` file below reads the khasset.xml file, and creates the commands and display templates necessary to rebuild the entire vector project.

restore_vector_project.pl

```
#!/usr/bin/perl
 #
 # Do you want to back up your vector project or duplicate it as a new project?
 #
 #
 # This script generates the instructions and files necessary to completely rebuild your
 # vector project. It reads your current vector project and extracts the display rules
 # and determines which raw data files were used to create your resources. The results
 # of running this script is a series of four new scripts called Step_One thru Step_Four
 # Running the resulting four scripts in order rebuilds your vector project as it was
 before.

 use File::Path;
 use File::Copy;

 # Read in the vector project to be backed up and the directory to
 # write the scripts and files to.
```

```perl
if ($#ARGV != 1){ usage(); exit; }
$vector_project = $ARGV[0];
$output_dir = $ARGV[1]; chomp( $output_dir );

$assetroot = &get_assetroot();
chomp( $assetroot );
print "Using Asset Root $assetroot \n";

open(StepOne,  ">$output_dir/Step_One_Create_Resources.sh");
open(StepTwo,  ">$output_dir/Step_Two_Build_Resources.sh");
open(StepThree,">$output_dir/Step_Three_Create_Project.sh");
print StepThree "genewvectorproject -o $vector_project\n";
close(StepThree);
open(StepFour, ">$output_dir/Step_Four_Add_Resources.sh");

# Locate the khasset.xml file for this project.
$khasset = "$assetroot/$vector_project.kvproject/khasset.xml";

open(IN, "$khasset") || die "Can't open $khasset";

# Read in all of the header info on the khasset file and ignore it
while(<IN>){ last if $_ !~ /<layers>/; }


# Keep reading the khasset file and parse out the information for each individual
# layer.

while(<IN>){
    my(@current_layer);

    if ($_ =~ /<layer>/){ # Found a new layer
        $_ = <IN>;
        while ($_ !~ /<\/layer>/){ # Read all of the data on this layer
            push(@current_layer, $_);
            $_ = <IN>;
        }
        # Extract the important naming information from this layer including
        # the name of the layer, and the resource that was used for this layer.
        # The legend corresponds to the folder that contains the layer

        $assetref = get_value( grep(/assetRef/, @current_layer));
        $assetname = get_value(grep(/name useDefault/, @current_layer));
        $assetname =~ s/ /_/g;
        $assetname =~ s/\W/_/g;
        $legend   = get_value( grep(/<legend>/, @current_layer));
        $legend =~ s/\|/\//g;
        $legend =~ s/ /_/g;

        # Store all of the display rules in subdirectories that correspond to the
        # structure of the folders that the layers are grouped in. This way, duplicate
        # layer names will not overwrite eachother
```

```
        if (! -d "$output_dir/$legend"){mkpath("$output_dir/$legend")};

        # Begin writing each display rule to its own ".khdsp" file and simultaneously
create
        # the instructions for building the resource and adding the resource to the
project
        # using the correct ".khdsp" template.

        if (length($assetref) >= 1){ # Not just a folder

            open(OUT, ">$output_dir/$legend/$assetname.khdsp") ||
                die "Cant open $output_dir/$legend/$assetname.khdsp";
            print OUT '<?xml version="1.0" encoding="UTF-8" standalone="no" ?>' . "\n";
            print OUT "<LayerConfig>\n";
            print OUT @current_layer;
            print OUT "</LayerConfig>\n";
            close( OUT );

            $raw_files = `gequery --infiles "$assetref"`;
            $raw_files =~ s/\n/ /g;
            print StepOne "genewvectorresource -o $assetref $raw_files\n";
            print StepTwo "gebuild $assetref\n";
            print StepFour "geaddtovectorproject -o $vector_project --template
$output_dir/$legend/$assetname.khdsp $assetref\n";

        }
        undef @current_layer;
    }
    last if $_ =~ /<\/layers>/;
}

copy( $khasset, $output_dir);

close(IN); close(OUT);
close(StepOne);
close(StepTwo);
close(StepThree);
close(StepFour);

sub get_assetroot(){
    open(IN, "/opt/google/etc/systemrc") ||
    open(IN, "/usr/keyhole/etc/systemrc") ||
    die "Can not open /opt/google/etc/systemrc or /usr/keyhole/etc/systemrc\n";
    @sysrc = <IN>;
    $assetroot=&get_value(grep(/assetroot/, @sysrc));
    return $assetroot;
}

sub get_value(){
    $string=shift(@_);
```

```
    $string =~ s/\s*<[^<>]*>//;
    $string =~ s/<.*//;
    chomp($string);
    return $string;
}

sub usage(){
print<<EndOfUsage

rebuild_vector_layer.pl vector_project output_dir

where:
    vector_project is the name of your vector project relative to the asset
    root - for example, MyProjects/MyVectors

    output_dir is the directory for putting all of the output from this script.
    This needs to be an absolute directory path like /home/mydir/vectors

EndOfUsage
}
```

Back to top

Manage Fusion disk space

Fusion writes all assets, projects, and databases to the asset root directory (typically named `/gevol/assets`). As a result, the asset root can fill up quickly.

You can put source data, such as raw images, in a different directory like `/gevol/src`, but the asset root can still fill up from processing source files. Cleaning assets, projects, and databases helps reduce the disk use in the asset root, but eventually the volume will fill up completely.

A good way to create free space in your asset root is to move your pyramid files to auxiliary storage. When you create and build a new imagery asset, pyramid (`.pyr`) files are also created, and are saved to the `raster.kip` and `mask.kmp` directories. Google delivers most of its data to customers in pyramid format. The pyramid files must always be available to the Fusion server, but after they're built, they don't change size. That makes them good candidates for moving to a separate storage location.

The result of moving your `.pyr` files is an asset that stores configuration files in the asset root, and large pyramid files in auxiliary storage. Besides creating space in your asset root, there are other advantages to moving your pyramid files:

- You can share assets across Fusion installations, or copy assets from one system to another.
- You can create assets on stand-alone computers and copy them to the grid later.
- You can manage disk space more easily. Because pyramid files do not grow or shrink, you can keep filling a volume with the pyramid files until it is full, then add another volume.
- You can store metadata (like the security level, when the source file was acquired, the bounding box, the resolution, the sensor type, and any other relevant data) with each pyramid directory. If you want to use the metadata in a new imagery project, it's more easily accessible than storing it buried in the asset tree.

Moving pyramid files

There are a few different ways to move pyramid files out of an asset root that is filling up:

- Move files out of the asset root and create symbolic links to the files. To do this, copy the entire `raster.kip` directory to a different location and then symbolically link `raster.kip` to the new location with the `ln -s` command:

  ```
  ln -s /path/to/new/raster.kip
  ```

  This option is expedient and can provide significant short-term relief to a full asset root. However, extensive use of symbolic links can increase the potential risk of Fusion or publish errors, and isn't good administrative practice.

- Set up KRP and KRMP task rules to direct Fusion to write the pyramid files to a different storage location instead of to the asset root. You can implement the task rules yourself or you can ask technical support to do it for you.
- Copy the pyramid files from the asset root and to other storage, then re-create the asset. See the instructions below.

To copy the pyramid files and re-create the asset:

After the `raster.kip` and `mask.kmp` directories are created and the asset is finished building, all of the information needed to copy and re-use the asset is inside these two directories. There are other files in the asset directory, but they are auxiliary and not needed. The basic strategy for managing disk space is:

1. Locate the asset to be moved. It will be in a directory with a `.kiasset` extension.

2. Locate the `raster.kip` and the `mask.kmp` directories under the asset directory:

   ```
   find $asset.kiasset -type d -name raster.kip
    find $asset.kiasset -type d -name mask.kmp
   ```

3. Copy the `raster.kip` and `mask.kmp` directories to the auxiliary storage volume:

```
cp -av $kip $new_location/$asset.kip
 cp -av $kmp $new_location/$asset.kmp
```

Note: The auxiliary storage volume must be a defined Fusion volume. You can define the new volume with the `geconfigurefusionvolume` command. Volumes are presented to the server as NFS file systems, and those can't be nested. For example, NFS does not allow `Volume1` and `Volume2` to be mounted as `/Volume1/Volume2`.

4. Rename the `raster.kip` and `mask.kmp` files so that they have the same name. The name should be descriptive of the asset. For example, if the asset is `EastChicago.kiasset`, the directories should be called `EastChicago.kip` and `EastChicago.kmp`.

5. Modify the imagery asset with the `gemodifyimageryasset` command, using the same name that the asset was originally created with:

```
gemodifyimageryasset -o imagery/EastChicago -havemask
 /gevol/newvolume/imagery/EastChicago.kip
```

For more details about the commands to re-create the assets, see *Importing Preprocessed Resources* in the Google Earth Enterprise Reference Guide.

6. Rebuild the imagery project and any database that contains the imagery project:

```
gebuild imagery/EastChicago.
```

The `gemodifyimageryasset` and `gebuild` commands will complete in seconds, because the heavy processing took place when the pyramid files were generated.

7. Clean the imagery projects and databases using the Fusion UI:
    a. From the **Asset Manager**, right-click the project or database and select **Asset Versions**.
    b. Right-click the previous version and select **Clean**. Cleaning the project and database also cleans all the assets, removing the pyramid files from the asset root and freeing up quite a bit of space.
    c. Verify that each of the assets that were modified were cleaned, and that the pyramid files were removed from the asset root.

Note: The asset expects the pyramid files to remain in the same location you specified in the gemodifyimageryasset command. Don't move the pyramid files to a new location after you've copied them and then modified, built, and cleaned the asset.

Example script

The example below copies all the pyramid files from /gevol/assets/imagery to /gevol/volume1.

```
# The commands are echoed to the terminal so you can review them before executing.
 # To enable the commands, uncomment the following line:
 # do_command=true

 asset_root=/gevol/assets
 asset_directory=Resources/Imagery
 new_location=/gevol/volume1

 cd $asset_root/$asset_directory
 for asset in `ls | grep kiasset | sed 's/\.kiasset//'`
 do

 # Find the raster.kip and mask.kmp under the asset directory
 kip=`find $asset.kiasset -type d -name raster.kip`
 kmp=`find $asset.kiasset -type d -name mask.kmp`


 # Copy the raster and mask directories to the new volume
 echo "cp -av $kip $new_location/$asset.kip"
 if [ "$do_command" == "true" ]; then cp -av $kip $new_location/$asset.kip; fi
 echo "cp -av $kmp $new_location/$asset.kmp"
 if [ "$do_command" == "true" ]; then cp -av $kmp $new_location/$asset.kmp; fi


 # modify and build each of the imagery assets
 echo "gemodifyimageryresource --havemask -o $asset_directory/$asset
$new_location/$asset.kip"
 #if [ "$do_command" == "true" ]; then gemodifyimageryresource --havemask -o
$asset_directory/$asset $new_location/$asset.kip; fi
 echo "gebuild $asset_directory/$asset"
 if [ "$do_command" == "true" ]; then gebuild $asset_directory/$asset; fi
done

 # Rebuild, then clean the imagery project and the database,
 # then verify that all the assets were cleaned.
```

Back to top

Running Fusion on a Machine with Multiple CPUs

Fusion can be configured to use a maximum number of CPUs on a machine with the following 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.

Note: The standard Fusion installation includes a license that allows you to run eight concurrent tasks/jobs. To use more than eight CPUs for Fusion processing, purchase an additional license by contacting your Google Account Manager.

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 *task rules* within Fusion. Implementation of task rules is described in more detail in Configure task rules.

The **ImageryPacketLevel.taskrule**, **TerrainPacketLevel.taskrule**, **MapLayerLevel.taskrule**, and **CombinedTerrain.taskrule** task rules enable multiple CPUs to work on each individual task. For example, configuring a `minNumCPUs` and `maxNumCPUs` of 2 for the **ImageryPacketLevel.taskrule** 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, **ImageryPacketLevel.taskrule**, **TerrainPacketLevel.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 `/opt/google/bin/geselectassetroot --assetroot /my/assetroot --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 **ImageryPacketLevel.taskrule** for imagery projects (3 concurrent `gepackgen` tasks possible).**
3. Set `minNumCPU=3` and `maxNumCPU=4` in **TerrainPacketLevel.taskrule** for terrain projects (2 concurrent `gepackgen` tasks 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` task)

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.

Google

Command reference

This article describes all of the command line tools used for system administration in alphabetical order. If you prefer, you can find each tool's syntax by entering the name of the tool on the command line with the `--help` option, for example:

```
geserveradmin --help
```

This article uses the following typographic conventions:

Italic	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

geaddtoimageryproject

```
geaddtoimageryproject [--mercator | --flat] [--historical_imagery |
  --no_historical_imagery] -o projectname{[--maxlevel level] insetresource}...
```

Purpose

Creates a new imagery project. This tool is capable of building Mercator imagery projects for 2D databases, or Flat (Plate Carrée) imagery projects with or without Historic Imagery Support for 3D databases.

Commands

--mercator

Optional. Uses Mercator map projection for the imagery project.

--flat

Default. Uses Flat map (Plate Carrée) projection.

--historical_imagery

Optional. Uses historical imagery for the project..

--no_historical_imagery

Default. Uses normal imagery for the project.

geconfigureassetroot

geconfigureassetroot {**--new --assetroot** *path* [**--srcvol** *path*] | **--repair** | **--editvolumes** | **--addvolume** | **--fixmasterhost** | **--noprompt**} [--nochown]

Purpose

To add volume definitions or edit existing volume definitions.

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
```

Options

--assetroot *path*

Path to asset root. This option is mandatory or optional in the `geconfigureassetroot` commands. If optional, then the current asset root is used if it is not specified.

`--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.

`--chown`

Optional. Prevents attempts by this command to fix file/directory privileges. You may consider setting this option when you do not want any prvilege settings to change as a result of reconfiguring your asset root.

Commands

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

Optional. Creates a new asset root. Specify the path to the new asset root.

Note: If you omit the path, the system creates a new asset root in `/gevol/assets`.

`--srcvol` *`path`*

Optional. Specify the path to the source volume.

`--repair` `[--assetroot` *`path`*`]`

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` `[--assetroot` *`path`*`]`

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` `[--assetroot` *`path`*`]`

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 Google Earth Enterprise Fusion.)

`--addvolume volume_name:path]`

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 Google Earth Enterprise Fusion.)

geconfigurepublishroot

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

Purpose

To specify the path where you want to push databases for publishing and serving with the current Google Earth Enterprise 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).

Do not create more than one publish root for a single asset root. That configuration produces unpredictable or undesirable results, including the inability to push at all from that asset root. You cannot push the same database multiple times to different publish roots on the same server.

gecutter

gecutter {**enable** | **disable**}

Purpose

To enable and disable the Cutter tool. Once you have enabled the Cutter, you launch it from the Settings menu in the GEE Server admin console. You can also launch the Cutter directly from `http://myserver.com/cutter`.

Note: The default admin security does not apply to the Cutter, so although it provides security if you try to launch the Cutter from the Admin console Settings menu, it does not block direct access to the Cutter via the URL. If you need Cutter security, you will need to add it separately. See GEE Server security.

See Create portable globes and maps.

Example

```
gecutter enable
```

```
gecutter disable
```

gedisconnectedclean

Deprecated in release GEE 4.4 and higher.

---

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

Deprecated in release GEE 4.4 and higher. Use `geserveradmin --publishdb` instead.

`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.kdabase?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 push/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 exists 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.

gepublishdatabase

 Deprecated in GEE 4.0.

Use `geserveradmin` to push and publish databases or use the Fusion GUI and GEE Server.

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.

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

Options

--assetroot <dir>

Path to the asset root. `--assetroot` is shown in the commands below as mandatory or optional. If optional, then the current asset root is used if it is not specified.

--noprompt

Do not prompt for more information,returns -1 to indicate an error if command fails or has insufficient arguments.>/p>

Commands

--list

Optional. Displays a list of the known asset roots on this machine.

--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 Google Earth Enterprise Server. This section breaks down the `geserveradmin` commands into the following categories:

- Options
- Database
- Virtual host
- Admin

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

Examples

```
geserveradmin --listdbs
geserveradmin --server_type stream --dbdetails "/gevol/assets/Databases/SF
Neighborhoods.kdatabase/gedb.kda/ver001/gedb"
geserveradmin --addvh my_public_vh --vhurl http://myserver.com/public_vh
geserveradmin --deletevh my_public_vh
geserveradmin --deletedb
geserveradmin --garbagecollect
```

geserveradmin command options

Fusion host name

`--fusion_host`

Optional. Fusion host name. Defaults to the current host name.

Stream server URL

`--stream_server_url` *`url`*

Optional. Specify a stream server other than the default. Defaults to the current server.

`--search_server_url` *`url`*

Deprecated. Always specify a stream server.

Server type

`--server_type {stream | search}`

Optional. Specify whether the server(s) in question are `stream` or `search` server(s). The default is `stream`. This option is required with the `listdbs`, `dbdetails`, and `garbagecollect` commands.

geserveradmin Database Commands

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

List registered databases

`--listdbs`

Lists all databases registered on the server.

Database file list

`--dbdetails` *`db_name`*

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

List published databases

`--publisheddbs`

Lists the database(s) currently published on the server.

List target paths

`--listtgs`

Lists all the target paths currently serving databases on the server.

Add database

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

Registers a new database with the specified name.

--adddb option	Required/Optional	Description
`--dbalias` *`alias`*	Optional	Specifies a user-friendly name for the database.

Delete database

`--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 first need to unpublish.(See also `--unpublish`.) To list the currently published databases, use the `--publisheddbs` option. (See also `--deletevh`.)

Push databases

`--pushdb` *`db_name...`* `[`**`--force_copy`**`]`

Pushes one or more databases to the server. For example, *`--pushdb db1 --pushdb db2`*

--pushdb option	Required/Optional	Description
`--force_copy`	Optional	Copies database files while pushing/publishing, otherwise creates a hard/symbolic link when server settings allow. To allow symbolic links, specify using `geconfigurepublishroot`: `sudo /opt/google/bin/geconfigurepublishroot -path=/gevol/published_dbs -allow_symlinks`.

Publish database

`--publishdb` *`db_name`* **`--targetpath`** *`target_path`* `[`**`--vhname`** *`vh_name`*`]`

Publish the specified database on the specified target path. If the virtual host name is omitted, it publishes to the default virtual host: "public".

--publishdb Option	Required/Optional	Description
--targetpath *target_path*	Required	Specifies the target path on which to publish.
--vhname *vh_name*	Optional	Specify the name of the virtual host. If the virtual host name is omitted, it publishes to the default virtual host: "public".

Unpublish database

--unpublish *target_path*

Unpublish database served from specified target path. For example, to unpublih a target path `/test`: `geserveradmin --unpublish /test`

geserveradmin Virtual Host Commands

Each of the virtual host (VH) commands is listed below, along with its syntax, description, and options.

With GEE 5.0, you can now set up a virtual host that provides a secure publishing point for as many databases as you associate with it.

Caution: Publishing to virtual hosts 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.

List virtual hosts

--listvhs

Provides a list of all registered virtual hosts configured for the current machine.

List virtual host information

--vhdetails *vh_name*

Displays the name, URL, and cache level of the specified virtual host.

Add virtual hosts

--addvh *vh_name* [**--vhurl** *url*] [**--vhcachelevel** *level*] [**--ssl**]

Registers a new virtual host with the specified name. Spaces are not allowed in the virtual host name. For example:

```
geserveradmin --addvh public_vh --vhurl http://mysite.com/public_vh
```

--addvh option	Required/Optional	Description

`--vhurl` *`url`*	Optional	The `vhurl` specifies the location of the virtual host. It must match the corresponding server-side virtual host configuration. If `vhurl` is omitted, it will be set to `http://`*`yourserver.domain`*`/vh_name`.
		
		There are three ways to specify the `vhurl`:
		
		• 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 host 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 host.
		
		After you use this command, you must create a configuration file for the new virtual host.

`--vhcachelevel` *`num`*	Optional	Specify a cache level (`1`, `2`, or `3`) for the virtual host. 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, Google Earth Enterprise 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 hosts 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 hosts as you want at the default cache level without worrying about running out of RAM. Typically, users increase only a small number of virtual hosts to cache level 3 on production servers and leave the rest of them at level 2. On servers that share a machine with Google Earth Enterprise Fusion, do not increase the level to 3. Google Earth Enterprise Fusion needs more RAM than the server does.
`--ssl`	Optional	Create a location-based virtual host with SSL configuration with the naming convention `_host.location_ssl` located in the path `/conf.d/virtual_servers/`. For detailed information about ensuring your Apache HTTP server configuration files are set up correctly, see Configure GEE Server 5.1.0 for SSL/HTTPS.

Delete virtual hosts

`--deletevh` *`vh_name`*

Permanently deletes the specified virtual host.

Note: If you want to delete a virtual host, you must first unpublish all currently published databases associated with it. To list the currently published databases for the virtual host you want to delete, use the `--publisheddbs` option. (See also `--unpublish`.)

geserveradmin Admin Commands

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

Delete database files

`--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 OS X operating system. See also `--deletedb`.)

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

Clean up portable globes and maps registration

`--portable_cleanup`

Clean up portable globes registration information. The cleanup unregisters/unpublishes portable globes or maps that have been removed from your `/globes` directory. You should run `--portable_cleanup` to clean portable registration information when portable files, which are currently published/registered, have been removed from your `/globes` directory.

Note: The cleanup is not implemented when there are no portable globes or maps in the globes directory: `/opt/google/gehttpd/htdocs/cutter/globes`.

getop

`getop` `[`**`--delay`** *`seconds`*`]`

Purpose

To display a list of what Google Earth Enterprise 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, the display updates every five seconds.

geupgradeassetroot

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

Purpose

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

Note: You must run this command as root.

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

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

Push and publish databases

Pushing a database is the process of making a Google Earth Enterprise Fusion database available on a Google Earth Enterprise Server so that it can then be published for viewing with Google Earth EC.

The *push* operation copies all the necessary files associated with a given 2D/3D Fusion database version to Google Earth Enterprise Server and registers the database. Pushing is performed from Fusion.

The *publish* operation makes a previously pushed database available for serving at a specified publish point. Publishing is performed on Google Earth Enterprise Server.

Along with the GUI of both Fusion and GEE Server, you have the option of using the command line to push and publish databases.

Push Databases

After you define and build your database , you *push* your database to Google Earth Enterprise Server. Pushed databases appear on the Databases page of the Google Earth Enterprise Server Admin console, from where you can then publish them.

To push a database:

1. Select Asset Manager from the Tools menu. The Asset Manager appears.

Before you configure Google Earth Enterprise Fusion

Configure your data locations

Configure font size in the GUI

Configure fonts for the Text Style dialog

Configure task rules

Benchmarking outcomes for applying task rules

Customize your Google Maps landing page

Monitor system processes

Backing up your data

Back up Fusion servers

Manage Fusion disk space

Running Fusion on a Machine with Multiple CPUs

Command reference

Push and publish databases

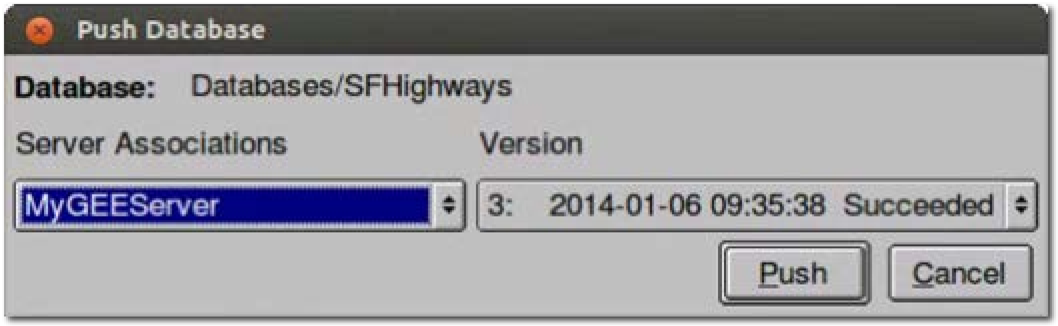
Publish databases using disconnected publishing

2. Navigate to the database that you want to publish.

3. Right-click the name of the database, and select Push from the context menu.

   If the Push option is not available, it may be because the database is still building.

4. The Push Database dialog appears.

   Select the server that you want to publish to from the Server Associations drop-down list. The most recent version of the selected database is the default selection on the Version drop-down list.

5. Click **Push**. Google Earth Enterprise Fusion pushes the database to Google Earth Enterprise Server, and displays a success message when it is done.

If you get an error message, contact your Google Earth Enterprise Server administrator for help, or check Troubleshooting Push issues for more information.

Publish databases

After you push your database to Google Earth Enterprise Server, you **publish** it in the GEE Server Admin console. You can then view your published databases with Google Earth EC or the Google Earth plugin for 3D databases or with the Google Maps API for 2D databases.

To publish a database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.

2. Sign in with the default credentials or the username and password assigned to you:
    - Default username: *geapacheuser*
    - Default password: *geeadmin*

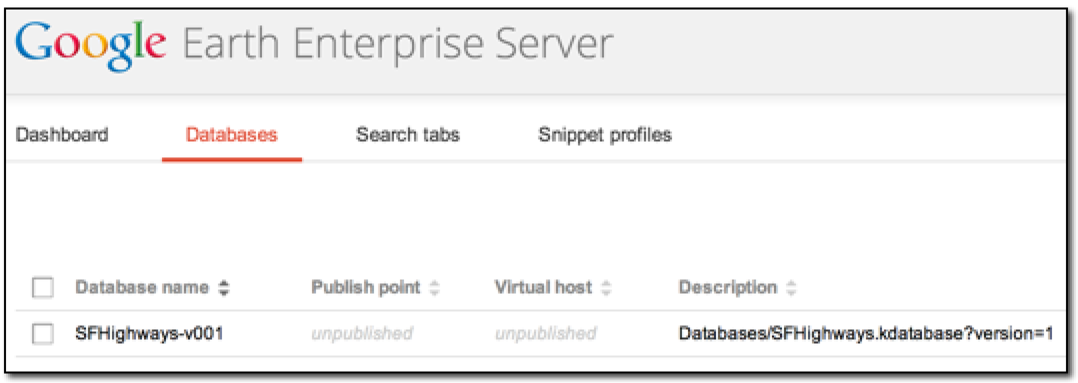
If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.

3. Click **Databases** to display the list of databases pushed to GEE Server.

   The list of databases appears *alphabetically*. Reverse the order by clicking the arrows at the top of the list:

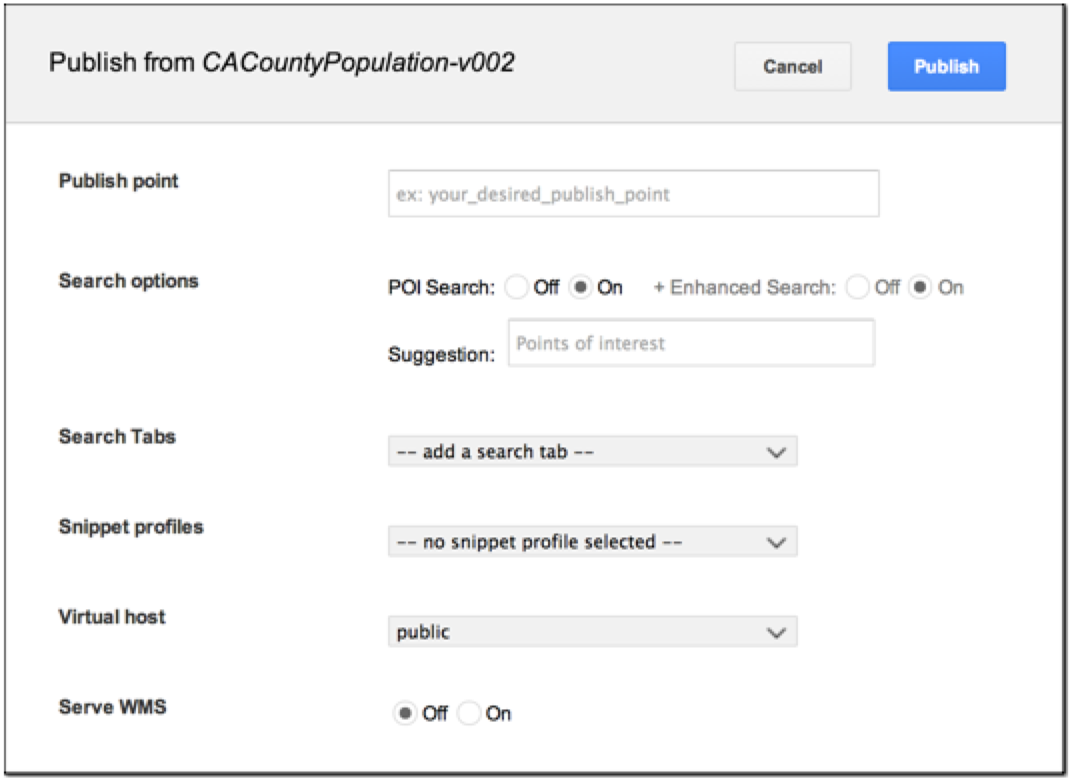
   

   Your most recently pushed database is added to the list.

The **Databases** page displays ten pushed databases per page so you may need to click the right arrow in the top-right corner of the window to see the next page of databases if your database is added to a longer list. To manage your list of databases, especially if you have multiple versions of the same database, you may want to remove unwanted versions by selecting the checkbox next to the database you want to remove and then click **Remove**. Note that this action removes the database from GEE Server and does not delete the database.

4. Check the box next to your pushed database. The **Publish** button appears.



5. Specify a **Publish point**, which is where you access the database or portable. For example, if you specify *sanfrancisco*, it will be accessible from *myserver.mydomainname*.com/*sanfrancisco*.
6. Specify the following options:
    - **POI Search** (Fusion databases with search data only)
    - **Search Tabs** (Fusion databases only)
    - **Snippet profile** (Fusion 3D databases only)
7. Select a **Virtual host**.
8. Optionally enable **Serve WMS**.
9. Click **Publish**. The **Databases** page updates to indicate the published status of your database.

Unpublish databases

If you no longer want to make a database available for serving on GEE Server, you can unpublish it and then remove it. If you want to publish it to a different server, or apply different settings, such as a secure virtual host, a search plug-in or a snippet profile, simply unpublish your database and republish using the new settings you want.

To unpublish a published database:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials or the username and password assigned to you:
   - Default username: *geapacheuser*
   - Default password: *geeadmin*

   If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.

   ---
3. Click **Databases** to display the list of databases pushed to GEE Server.
4. Check the box next to the database that you want to unpublish. The **Unpublish** button appears above the list of databases.
5. Click **Unpublish** to unpublish a database from its publish point.

   You can now either remove the database, which is to unpush an unpublished database, or you can republish to a different publish point using a different virtual host, for example.

To remove an unpublished database from GEE Server:

1. From the Databases page of GEE Server, check the box next to the database that you want to remove. The **Remove** button appears above the list of databases.
2. Click **Remove** to remove the database from GEE Server.

   Removing a database does not *delete* it. Remove reverses the *push* process so that the database is no longer pushed to GEE Server.

   ---
3. The checked database is removed from the list.

To restore the removed database, push it again from Fusion. See Push databases.

Learn more

- Create Search tabs
- Manage snippet profiles
- Manage virtual hosts
- Make Web Map Service (WMS) requests
- Google Earth EC

Publish using the command line

The procedure to publish a database using the command line requires you to add, push, then publish a database. You must specify a target path and may optionally specify a virtual host to publish to. Before you begin, you may need to identify the virtual hosts registered on your default or designated stream server, and you'll need to identify the full path and name of the database to publish.

To publish a database using the command line:

1. Query GEE Server for virtual stream servers and write down which virtual server you want to publish to. You can skip this step if you are using default servers only.

   `geserveradmin --stream_server_url http://myserver.org --server_type stream --listvhs`

2. Get the full folder path for the database to publish:

   `gequery --outfiles Databases/NameOf3DDatabase.kdatabase` for 3D

   For 3D databases, look for the path ending in `gedb`.

   `gequery --outfiles Databases/Nameof2DDatabase.kmmdatabase` for 2D

   For 2D databases, look for the path ending in `mapdb`.

3. Add the database to register it with GEE Server. In this example, we publish a 2D database, using the full

folder path as discovered in the previous steps. You use `geserveradmin --adddb db_name` to register the database.

```
geserveradmin --stream_server_url http://myserver.org --adddb
 /gevol/assets/Databases/Nameof2DDatabase.kmmdatabase/mapdb.kda/ver001/mapdb/
```

4. Push the database to GEE Server using `geserveradmin --pushdb db_name`. For example:

```
geserveradmin --stream_server_url http://myserver.org --pushdb
 /gevol/assets/Databases/Name of 2DDatabase.kmmdatabase/mapdb.kda/ver001/mapdb/
```

5. Publish the database to GEE Server using `geserveradmin --publishdb db_name --targetpath target_path`. For example:

```
geserveradmin --publishdb
 /gevol/assets/Databases/SFMapDatabase.kmmdatabase/mapdb.kda/ver001/mapdb/ --
targetpath http://myserver.org
```

For more information about the `geserveradmin` command options, see the Command Reference.

Back to top

Publish databases using 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 create a database that can be output to portable media, which can then be pushed and published to GEE Server.

A large database can be copied to portable media and loaded onto the server using `geserveradmin` commands, saving the time and bandwidth required to send terabytes of data over the network. In addition, you can also create deltas for the same databases, which can then be sent over the network.

Before you begin

To use the disconnected publishing service, you need to install the **GEE Server - Disconnected Add-on**, which you can add by running the installation process and choosing **Custom** as the installation type.

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

Before creating a disconnected database, you can estimate the space required to store the disconnected database by running `gedisconnectedsend` with the `--report_size_only` option.

To estimate the size of a disconnected database:

1. From the command line, specify the database that you want to publish with the `--report_size_only` flag:

```
# gedisconnectedsend --report_size_only --sendpath \
 /gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb
```

To create a disconnected database:

1. Create an output directory on your portable media, which you will specify when you create the disconnected database.

   ```
   rm -rf /media/ddb/test_ver001/
   
   mkdir -p /media/ddb/test_ver001/
   ```

2. Create the disconnected database and output the file to your portable media:

   ```
   gedisconnectedsend --output /media/ddb/test_ver001/ \
   
   --sendpath /gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb
   ```

   In this example, `/media/ddb/test_ver001/gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb/` is the path to the `gedb` directory of the disconnected database.

Prepare delta disconnected databases

The disconnected database command includes the option, `--havepath` or `--havepathfile`, to create a delta disconnected database, based on different versions of the same database. To create the delta, the assetroot of the GEE Fusion machine must store the versions of databases, as reported by geserveradmin `--listdbs` on the GEE Server machine, from which the delta disconnected database is created.

When publishing from a GEE Fusion machine, in this example, `machine_one`, to a disconnected GEE Server machine, in this example, `machine_two`, verify the list of versions of the disconnected database for which you want to create the delta:

For example, on `machine_two`:

```
geserveradmin --fusion_host machine_one --listdbs >
 /home/user_name/ddb/dblist_server_host
```

On `machine_one`, create the delta disconnected database using the `--havepathfile` option:

```
gedisconnectedsend --output /media/ddb/test_ver001/ \
--havepathfile /home/user_name/ddb/dblist_server_host \
--sendpath /gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb
```

Publish on the remote server host

Connect the portable media to the Server host and mount the drive. In this example, `/media/ddb/test_ver001/gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb/` is the path to the `gedb` directory of the disconnected database.

To publish on the remote server host:

1. Set ownership and permissions for the `gedb` directory:

   ```
   $ sudo chown -R gefusionuser:gegroup ddb/
   $ sudo chmod -R 755 ddb/
   ```

2. Register the database on the Server. Specifying the --stream_server_url is optional.

   ```
   geserveradmin --fusion_host fusion_host.company.com \
   --stream_server_url http://your_stream_server \
   --adddb \
   /media/ddb/test_ver001/gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb/
   ```

3. Push the files to the Server:

   ```
   geserveradmin --fusion_host fusion_host.company.com \
   --stream_server_url http://your_stream_server \
   --pushdb \
   /media/ddb/test_ver001/gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb/
   ```

4. Publish the database on the Server either using the GEE Server Admin console or on the command line. The GEE Server Admin console Publish dialog includes options to add search, snippet profiles, specify a virtual

host, and turn on WMS.

To publish on the command line:

```
geserveradmin --fusion_host fusion_host.company.com \

--stream_server_url http://your_stream_server \

--publishdb \

/media/ddb/test_ver001/gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb/
```

5. Verify that the database manifests:

```
geserveradmin --stream_server_url http://your_stream_server \

--listdbs

geserveradmin --stream_server_url http://your_stream_server \

--dbdetails \

/media/ddb/test_ver001/gevol/assets/Databases/test.kdatabase/gedb.kda/ver001/gedb
```

6. 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/test.kdatabase/gedb.kda/ver001/gedb

ssh your_server /opt/google/bin/geindexcheck --mode all --database \

/gevol/published_dbs/stream_space/your_server/gevol/assets/

Databases/test.kdatabase/gedb.kda/ver001/gedb
```

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

Symbolic links

Symbolic links are turned on by default in the published databases configuration file, `AllowSymLinks=Y` in the `/gevol/published_dbs/.config`, resulting in hard or soft symbolic links being created, depending on the logical volumes of the relevant directories:

- **hard-link**: the disconnected database folder and published_dbs folder are on the same logical volume.
- **soft-link**: the disconnected database folder and published_dbs folder are on different logical volumes.
- If the disconnected database folder is in `/tmp`, the directory files will be copied.

Delete disconnected databases

To delete disconnected databases at their end of life use `geserveradmin --deletedb`, the same procedure used for databases that are published normally.

Back to top

GEE Server Overview 5.0

GEE Server 5.0 introduces a flexible and dynamic approach to publishing maps and globes: When publishing, you specify unique options for a map or globe at a given URL; the map or globe can then be easily published multiple times at different URLs with different configurations. Likewise, particular configurations can quickly be unpublished or modified.

Virtual hosts (previously called virtual servers) now only specify a security protocol and can be associated with multiple published globes and maps. This change decouples the task of setting up security protocols from publishing maps and globes.

Associating search sets and snippet profiles at the time of publishing allows you to present different versions on the same underlying Fusion database, which can then be made available at different URLs and under different security protocols. For example,you can protect one set of searchable data but display it on the same globe as an unprotected set of searchable data. You can then publish the database privately by adding the sensitive search and a secured virtual host at one URL, then publish the database again, this time adding the non-sensitive search and public virtual host at different URL.

GEE Server now supports Web Map Service (WMS). One of the benefits of using the WMS standard is that supported clients can request images from multiple WMS servers and then combine those mapping images into a single view. Because the WMS standard is used to get all the images, they can easily be overlaid on one another.

Publishing can now be done in a matter of seconds from the GEE Server admin console, and no longer requires interaction with Fusion. We hope that this new publishing approach makes your life easier and maybe even a little more fun!

GEE Server also provides tools for cutting portable globes `.glb` and maps `.glc`. Composite globes and maps can be assembled from 2D or 3D portable files. A resulting offline globe or map is packed as a `.glc` file that may be downloaded for use with Portable Server.

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Sign in to GEE Server Admin console

To access the GEE Server Admin console for managing and publishing maps and globes:

1. Go to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials:
   - Username: geapacheuser
   - Password: geeadmin

Tip: To reset the username and password:

```
sudo /opt/google/gehttpd/bin/htpasswd -c
 /opt/google/gehttpd/conf.d/.htpasswd geapacheuser
```

Back to top

Publish databases and portables

When you *publish* a Fusion database, you select the database you want to publish, then apply publishing options in the **Publish** dialog. Before you publish a Fusion database, you must register it by *pushing it to GEE Server.*

*With portable files, the process is a little different. Instead, first you register your portable globes and maps in the **Manage Portables** dialog before publishing them. Registering a portable globe or map is similar to pushing a database from Fusion.*

Publish a Fusion database

To publish a Fusion database:

1. *In the GEE Server Admin console, click **Databases**.*
2. *Check the box next to the database or portable to publish.*
3. *Click **Publish** to open the Publish dialog.*

GEE Server Overview

Sign in to GEE Server Admin console

Publish databases and portables

Create Search tabs

Manage snippet profiles

Display custom logos

Manage virtual hosts

Configure GEE Server 5.1.0 for SSL/HTTPS

Configure a GEE virtual host for LDAP authentication

Make Web Map Service (WMS) requests

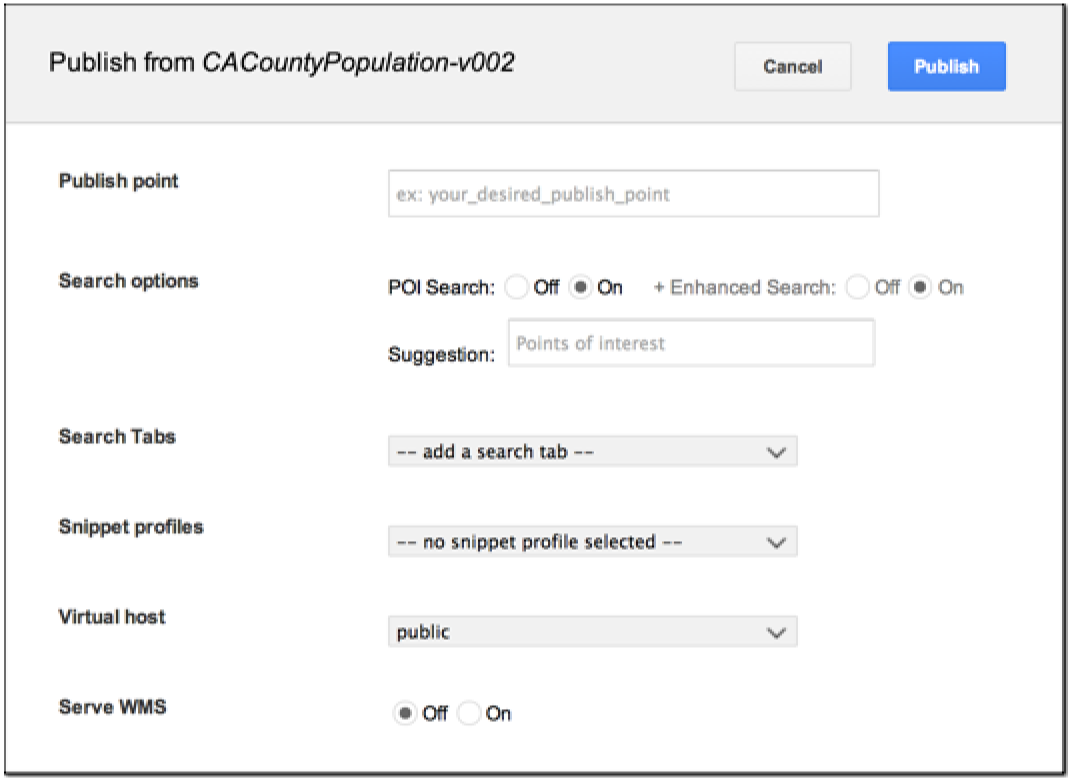
Create portable globes and maps

Create composite globes and maps

View 3D databases

Implement Earth Plugin

Provide Earth Plugin for your



4. *Specify a **Publish point**, which is where you access the database or portable. For example, if you specify sanfrancisco, it will be accessible from myserver.mydomainname.com/sanfrancisco.*

***Note:** When publishing a database, the publish point you specify is case insensitive. Upper and lower case are not differentiated. Make sure each publish point path name you specify is unique.*

---

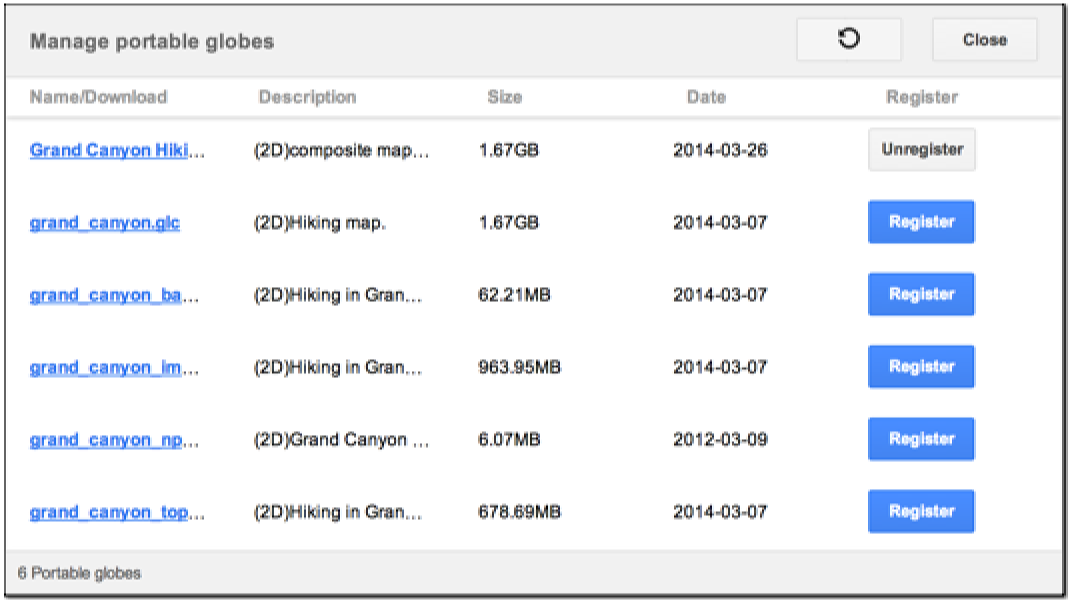
5. *Specify the following options:*
    1. ***POI Search** (Fusion databases with search data only)*
    2. ***Search Tabs** (Fusion databases only)*
    3. ***Snippet profile** (Fusion 3D databases only)*
6. *Select a **Virtual host**.*
7. *Optionally enable **Serve WMS.***

8. Click **Publish**. The **Databases** page updates to indicate the published status of your database.

Register and publish a portable globe or map

To register a portable globe or map:

1. In the GEE Server Admin console, click **Databases**.
2. Click **Manage Portables** to open the Manage portable globes dialog.



The list of globes and maps corresponds to the files located in your default `/opt/google/gehttpd/htdocs/cutter/globes` directory.

3. Click **Register** next to the globe or map that you want to make available for publishing on GEE Server. The registered globe or map appears in the list on the **Databases** page of the Admin console.

   You can register as many portable globes or maps as you have listed in the dialog. Click **Unregister** to make a globe or map unavailable on GEE Server.

4. On the **Databases** page of the Admin console, check the box next to the portable globe or map to publish.
5. Click **Publish** to open the Publish dialog.

6. *Specify a **Publish point**, where the portable will be accessible from. For example, if you specify grandcanyon, it will be accessible from myserver.mydomainname.com/grandcanyon.*

   ***Note:** When publishing a database, the publish point you specify is case insensitive. Upper and lower case are not differentiated. Make sure each publish point path name you specify is unique.*

7. *Select a **Virtual host**.*
8. *Click **Publish**. The **Databases** page updates to indicate the published status of your portable globe or map.*

Learn more

- *Manage snippet profiles*
- *Manage virtual hosts*
- *Make Web Map Service (WMS) requests*
- *Create portable globes and maps*

Back to top

Create Search tabs

You can create search tabs in GEE to add search services that let you query GEE Server using Google Earth Enterprise Client (EC) or from a Google Map (http://maps.google.com/).

Search tabs allow Google Earth EC or Google Maps users to:

- Replace the standard search tabs in Google Earth EC.
- Search 2D or 3D databases created with Google Earth Enterprise Fusion.
- Access other databases not related to Google Earth Enterprise Fusion (such as geocoders, Google Search Appliance, and so on).
- Access external search servers (such as a real estate search).

For example, if you have a database of property locations that contains specific information that your users need, you can create a search tab called *Property Search* and configure it to search for locations in your property database, even if that database is stored on another server.

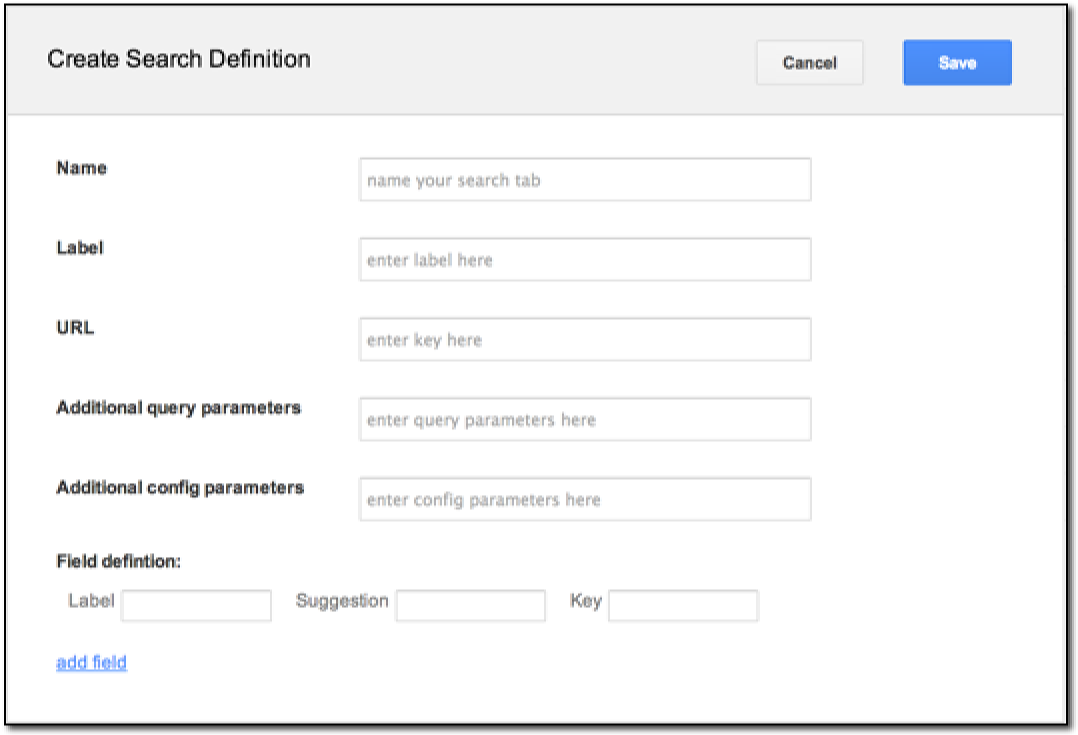
You can also create search tabs with *multiple fields*. For example, you might create a Driving Directions search tab with *From* and *To* labels in the Create Search Definition dialog. You populate the URL field with the custom service that returns the actual directions.

The **Search tabs** page of Google Earth Enterprise Server allows you to pre-configure the search tabs to be used for any Fusion database. You define all of the search tabs you need using the **Search tabs** page, and then you select one or more search tabs for a specific database when you publish using the **Databases** page. See Publish databases and portables for details.

Define a new search tab

To define a new search tab:

1. In the GEE Server Admin console, click **Search tabs**.
2. Click **Create new**. The Create Search Definition dialog appears.



3. Enter a unique **Name** for your search tab.
4. Enter a **Label** that will display as the name of the search tab in the client.
5. Provide a **URL**.

   if you want to reference a Google Earth Enterprise Server plug-in for the search tab, enter the name of the plug-in. The following plug-ins are available:

   - POISearch
   - /gesearch/FederatedSearch
   - /gesearch/CoordinateSearch

- /gesearch/PlacesSearch
- /gesearch/ExampleSearch

6. Optionally provide additional query parameters.

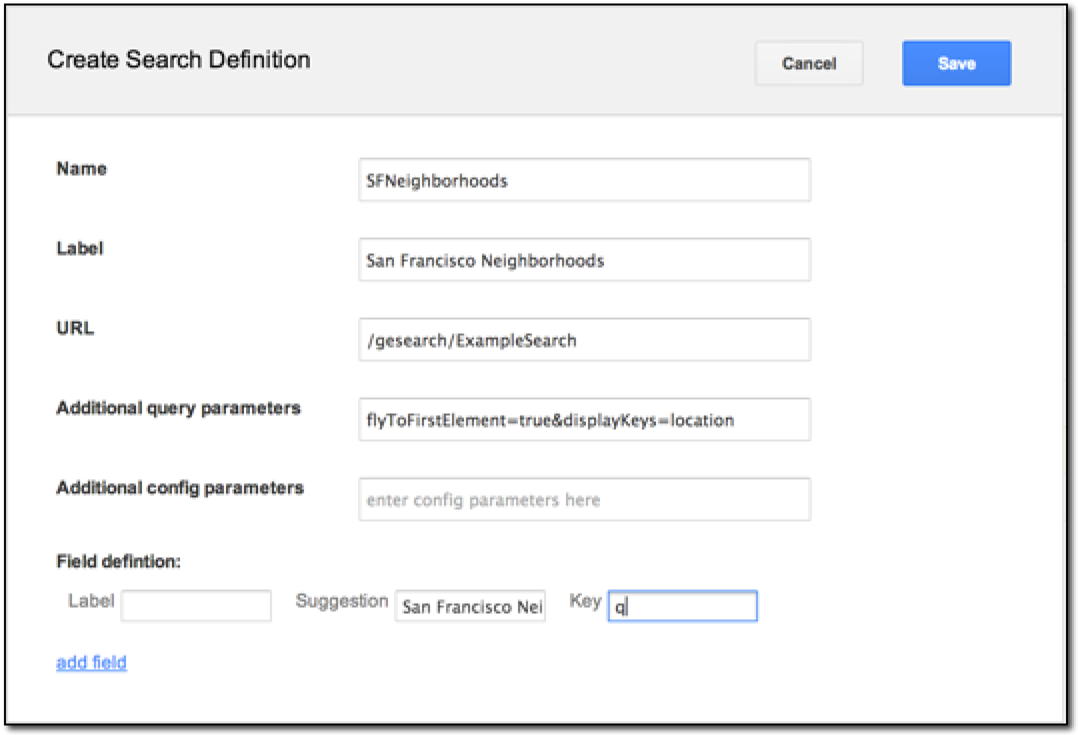
   The syntax is **key1=value1&key2=value2**. For example, **flyToFirstElement=true&displayKeys=location** flies to the first element of your search results and displays the Location key in the results. Google Earth EC must be able to understand and respond to these key/value pairs, so you must be very familiar with your search application to use this field.

   Caution: Any web service, servlet, or web application you configure the search tabs to query must return valid KML to Google Earth EC. For Google Maps, it must return valid JavaScript in the specified structure.

7. Optionally provide additional config parameters.
8. Specify the following field parameters:
   1. **Label**: Label of the search box.
   2. **Suggestion**: Suggestion to the user of what to search for.
   3. **Key**: The parameter name that the value in the text box will be assigned when the search is submitted.

   Optionally click **add field** to add another set of search field parameters.

9. Click **Save**.



The search tab is ready for you to add to your database. See Publish databases and portables for details.

Customize search tab style in Google Earth EC

You may want to customize the appearance of the search tabs as they display in Google Earth EC. You can edit `/opt/google/gehttpd/htdocs/earth/supplemental_ui.html` to change the style and various elements for the search form that appears. The changes you make are then included in the database at the time you publish.

To customize the style and elements of a search tab:

1. Make a backup copy of `/opt/google/gehttpd/htdocs/earth/supplemental_ui.html` and open your renamed file for edit.
2. In the HTML code, update the style and display elements to customize your search form.
3. Save the HTML file with the name you want to use for your search tabs, for example, `/opt/google/gehttpd/htdocs/earth/supplemental_ui_MySearchTab.html`.

4. Publish your database with the selected search tab(s).
5. View your database in Google Earth EC.

   Search tabs now appear with the style and element settings that you customized.

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Manage snippet profiles

You can customize your database view and available features in Google Earth Enterprise Client (EC) by setting up *snippet profiles* in the the GEE Server admin console.

Snippets are different properties and options that may be specified for a Fusion 3D database. When you connect to a database in Google Earth EC, snippets control its appearance and behavior. GEE Server lets you set your snippet preferences, then combine snippets into a *snippet profile*, which can then be applied to any database that you publish. The snippet profile settings then modify the database's behavior and appearance when you connect to it in Google Earth EC.

For example, many of these preferences (or *snippets*) apply to display characteristics, such as showing or hiding different elements such as the Google logo or your own co-brand logo. You can also apply other settings, such as caching data on disk, enabling authentication, hiding user data in the **About** dialog, and specifying a reverse geocoder server URL.

Google Earth EC recognizes only the settings made by the first database that you connect to in Google Earth EC. This applies when you are connecting to multiple databases.

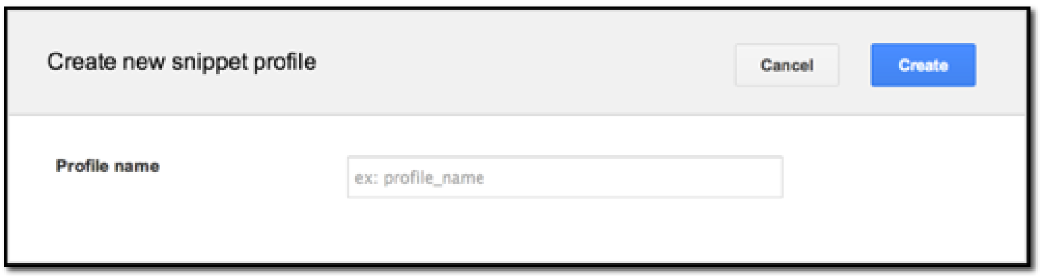
Caution: If you are working with multiple GEE Server users on multiple workstations, it is important to remember that snippet profiles can be accessed by all users at the same time. Be sure to coordinate with any other users to not overwrite snippet profile settings.

- Creating a snippet profile
- Editing snippet settings
- Table of snippet settings
- Modifying snippet profiles
- Deleting snippet profiles

To create a snippet profile:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials or the username and password assigned to you:
   - Default username: *geapacheuser*
   - Default password: *geeadmin*

   Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.
3. Click **Snippet profiles** to display your snippet profiles.
4. Click **Create New**. The **Create new snippet profile** dialog appears.



5. Enter a name for the new snippet profile and click **Create.** The snippet profile name appears in red in the **Existing snippet profile** list and the **Snippet editor** opens.

To edit snippet settings:

1. Click the **Add a new snippet set to the profile** drop-down to display the list of available snippets.

Clearing the Google Earth EC cache

Admin console pages

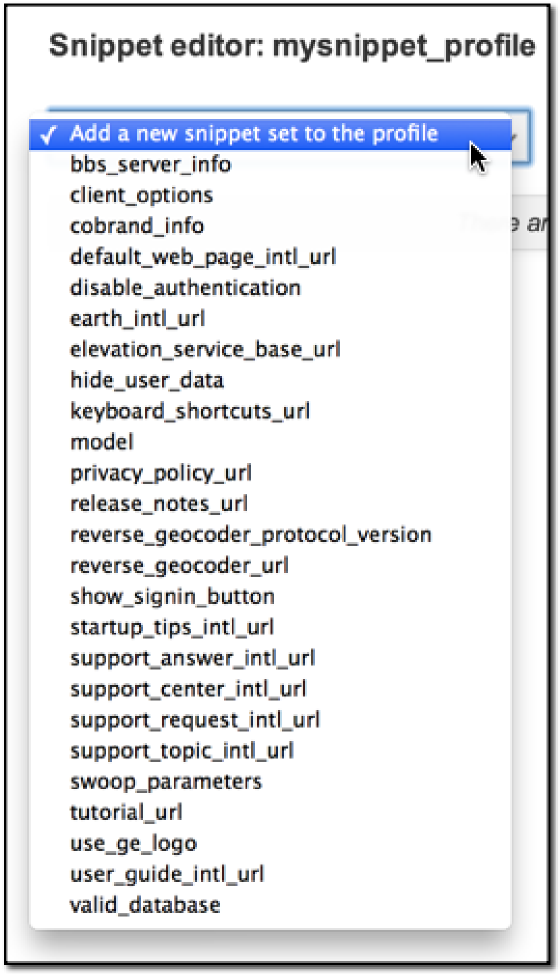
Snippet profiles page

Search tabs page

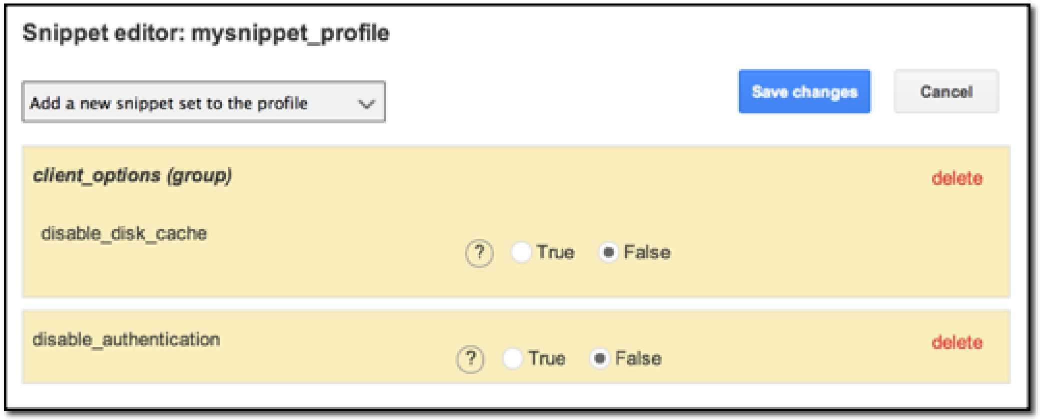
Settings page

Dashboard page

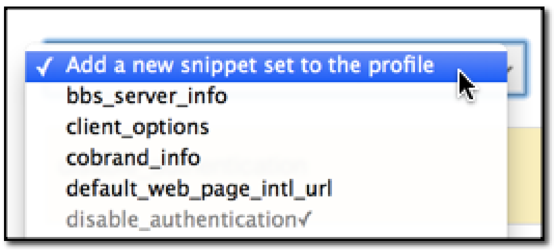
Databases page



2. Select a snippet from the list. The selected snippet options are added to the current snippet profile and appear in the profile list.



You can select some snippets multiple times per profile, For example, you may want to add more than one customized logo to be displayed in Google Earth EC. For those snippets that cannot be added to a profile more than once, such as disable authentication, the snippet name appears greyed out and a checkmark appears next to it in the snippet list.



3. Edit the snippet profile to specify values for the snippets. See the following **Snippet Settings** table for snippet settings.

When you add a snippet, you must enter values in the empty fields; if you want the snippet string to be empty, which is the default setting, you should not add the snippet and edit its values.

---

4. When you have finished adding snippets and specifying values for your snippet profile, click **Save changes**.

   Now you can apply your snippet profile to any 3D database you publish.

Snippet Settings

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

Snippet Name	Purpose	Syntax	Notes
`bbs_server_info`	Specify BBS Server info.	`base_url`—URL of the server, including protocol, domain name, and port. `file_submit_path`—Path on server where files can be submitted. `name`—Name that will be displayed in context menu to user. Must be translated. `post_wizard_path`—Path on server where wizard can be found.	
`client_options`	Disable disk caching in Google Earth EC.	`disable_disk_cache`	Default value is False
`cobrand_info`	Add custom logo to Google Earth EC display window.	`logo_url`—URL of image to use as logo `screen_size`—Positive value <=1 specifies scale with screen. `tie_point`—Controls the reference point in the overlay. `x_coord.is_relative`—If True, the coordinate is relative to the screen. `x_coord.value`—Coordinate value. Interpretation depends on value set in `x_coord.is_relative`. `y_coord.is_relative`—If True, the coordinate is relative to the screen. `y_coord.value`—Coordinate value. Interpretation depends on value set in `y_coord.is_relative`.	`logo_url` can be remote or local. `screen_size` makes logo scalable with screen by forcing its width to occupy a fraction of the screen. For example, a value of 0.25 sets the given logo to occupy 25% of the screen.
`default_web_page_intl_url`	Default location of web page in Google Earth EC.	`https://www.google.com/?hl=%251`	Can be set to an internal IP or host name address. Default web page value in GEE is an empty string.
`disable_authentication`	Disable session cookie-based authentication.	`boolean`	Indicates that this database does not require session cookie-based authentication.
`earth_intl_url`	Location of international page for Google Earth.	`http://earth.google.com`	
`elevation_service`	Terrain elevation service URL.		If field is

_base_url			empty, service is unavailable.
hide_user_data	True = Suppress user name and license key information in the Help -> About window. False = Display user name and license key information.	boolean	Default is False.
keyboard_shortcuts_url	URL for keyboard shortcuts page. If not specified, this URL is built from user_guide_intl_url as user_guide_intl_url"ug_keyboard.html"		Can be set to an internal IP or host name address.
model		compressed_negative_altitude_threshold—Threshold below which negative altitudes are compressed elevation_bias—Elevation bias flattening—Planet flattening. Default value is 1.0/298.257223563 (from WGS84) negative_altitude_exponent_bias—Bias for negative altitude so that ocean tiles can be streamed to older clients radius—Mean planet radius. Default value is the WGS84 model for earth	
privacy_policy_url	URL for privacy policy.	IP address or host name	Can be set to an internal IP or host name address.
release_notes_url	URL for release notes.	IP address or host name	Can be set to an internal IP or host name address.
reverse_geocoder _protocol_version	Reverse geocoder protocol version.	numeric value	Default is 3 which is the protocol supported by newer clients.
reverse_geocoder_url	Reverse geocoder server URL.		
show_signin_button	If True, shows the signin button in the top-right corner of the display window.	boolean	
startup_tips_intl_url	Localize international URL from which to load startup tips for Earth 7.0 or higher.		
support_answer_intl_url	Localize international URL for support answers	https://support.google.com/earth/#topic=4363013	
support_center_intl_url	Localize international URL for the support center	http://support.google.com/earth/	
support_request_intl_url	Localize international URL for support requests	https://support.google.com/earth/#topic=2364258	

support_topic_intl_url	Localize international URL for support topics	http://www.google.com/earth/learn/	
swoop_parameters	Controls how far from a target swooping should start.	start_dist_in_meters	
tutorial_url	URL for tutorial page. If URL is not specified, this URL is built from user_guide_intl_url as user_guide_intl_url + "tutorials/index.html".	http://www.google.com/earth/learn/	
use_ge_logo	Shows/hides Google Earth logo in lower right corner of display.	boolean	Default is True.
user_guide_intl_url	Localize international URL for documentation	http://www.google.com/earth/learn/	Defaults to local PDF file for Google Earth EC. Can be set to an internal IP or hostname address.
valid_database	Validates the database name and URL.	database_name—Human-readable name of database, for example, "Primary Database" or "Digital Globe Database" database_url—URL of server. This can include a path and a query, and must be a well-formed, absolute URL.	

To modify a snippet profile:

1. To change the snippets in a snippet profile, click the snippet profile name in the **Existing snippet profiles** list that you want to edit.

   The **Snippet editor** appears with the name of your selected snippet profile and the list of included snippets.

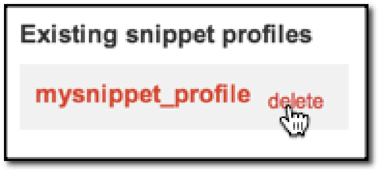
   All your previously selected snippets can be edited with new options and settings.

2. To add a snippet, click the drop-down list to select a new snippet.
3. To delete a snippet from your profile, click **delete** in the snippet settings.

4. Click **Save Changes**.

To delete a snippet profile:

1. Hover your cursor over the name of the snippet profile that you want to delete. The **delete** option appears.

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

2. Click **Yes**.

   The snippet profile disappears from the **Existing snippet profiles** list.

Learn more

Google Earth Enterprise Client (EC)

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Google

Display custom logos

Google Earth Enterprise Client (EC) 5.0 and later can display multiple custom logos with your globes and maps when connected to a GEE Server.

Custom logos are added as database preferences or *snippets*, which are managed using snippet profiles you create using GEE Server. Once you have saved a custom logo in a snippet profile, you assign it to any database you publish. You can apply a snippet profile to as many databases as you need.

Cobrand_info snippet

You can use the **cobrand_info** snippet within a snippet profile to place your logos anywhere in the Google Earth EC window.

The benefits of using **cobrand_info** are:

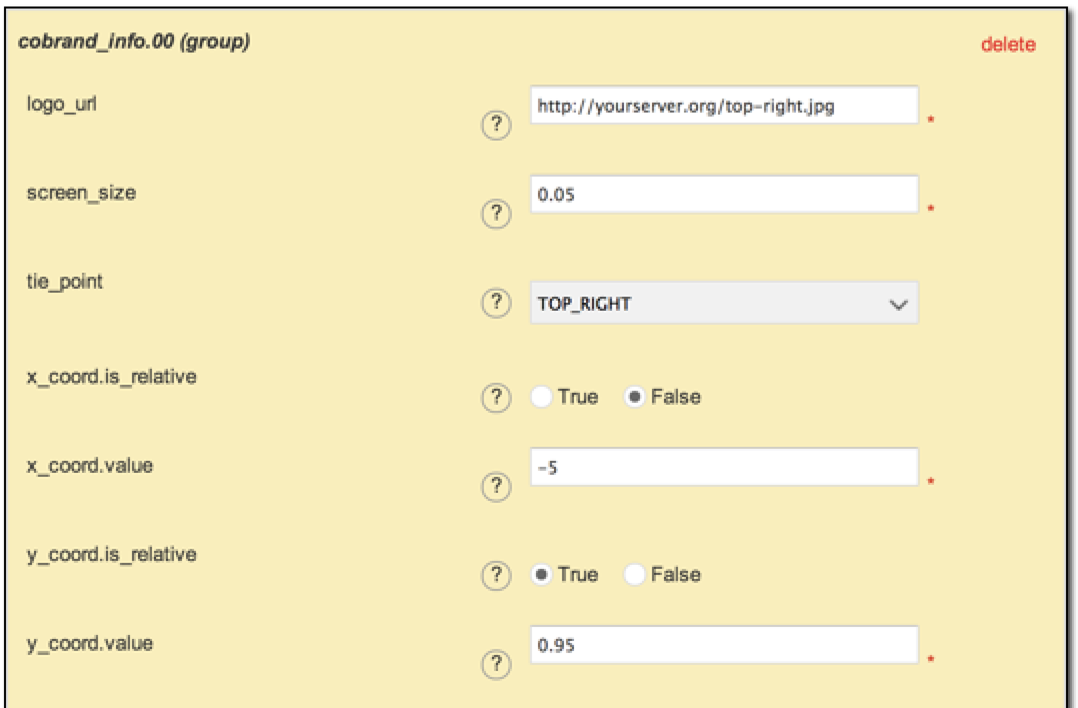
- You can access logos at both local and remote addresses (UNC, HTTP, and HTTPS) However, we recommend that you host your logos on GEE Server to ensure accessiblity to the server and Google Earth EC.
- PNG, GIF, and JPEG image formats are supported.
- Transparent logos are supported.
- You can place logos anywhere on the screen.
- You can display multiple logos at the same time.
- You can use either absolute pixel values or relative screen size values to place logos.
- Logos can either be a fixed size or scaled to the width of the screen.

- Logos are included with saved images and printouts from EC.

Cobrand_info snippet definition

You define the **cobrand_info** snippet by specifying the URL from which the logo will be served, the screen width size, the tie point, which controls the reference point used when overlaying the logo, and x and y coordinates, expressed as either absolute values in pixels, or relative values of percentage of screen width or height.

The following **cobrand_info** example snippet places the *top_right.jpg* logo in the upper-right corner of the window (the top right of the image is 5 pixels from the right edge of the window edge, using an absolute x coordinate value, and 95% from the bottom using a relative y coordinate value), and sets it to dynamically scale to 5% of the screen width as the Google Earth EC window is resized.



Cobrand_info parameters

The following table describes the cobrand_info snippet parameters, all of which are required to be set.

Name	Description
logoUrl	The location of the logo file. For example, "http://yourserver.org/top-left-logo.gif" Logos can be accessed locally or remotely via UNC, HTTP, or HTTPS. Supported logo formats are PNG, GIF, and JPEG.
screen_size	Specifies a fraction of window width the logo should be scaled to. Value must be between 0.0 and 1.0, where 0.0 disables the scaling feature.
tiePoint	Specifies the part of the logo file that is placed at the specified **x_coord.value** and **y_coord.value** values. The following screen positions are allowed: • **BOTTOM_LEFT** • **BOTTOM_CENTER** • **BOTTOM_RIGHT** • **TOP_LEFT** • **BOTTOM_LEFT** • **MID_LEFT** • **TOP_RIGHT** • **TOP_CENTER** • **MID_RIGHT** • **MID_CENTER**
x_coord.is_relative	Used in conjunction with **x_coord.value** to specify absolute or relative pixel placement of the **tiePoint**. • **False** means **x_coord.value** is a pixel value • **True** means **x_coord.value** is a window size fraction.
x_coord.value	

	The EC horizontal (X axis) coordinate for tiePoint. **x_coord.value** specifies a distance from the left edge of the Earth window for logo **tiePoint** placement. **x_coord.value** is used in conjunction with **x_coord.is_relative**, which specifies whether the distance is measured in pixel or relative values. • If x_coord.is_relative is **False** (default), x_coord.is_relative is a pixel value. You can use negative values to specify a distance from the right edge. • If x_coord.is_relative is **True**, allowed values for x_coord.is_relative are between 0.0 (left edge of window) and 1.0 (right edge).
y_coord.is_relative	Used in conjunction with **y_coord.value** to specify absolute or relative pixel placement of the **tiePoint**. • **False** means **y_coord.value** is a pixel value • **True** means **y_coord.value** is a window size fraction.
y_coord.value	The EC vertical (Y axis) coordinate position for tiePoint. **y_coord.value** specifies a distance from the bottom edge of the Earth window for logo tiePoint placement. **y_coord.value** is used in conjunction with **y_coord.is_relative**, which specifies whether the distance is measured in pixel or relative values. • If y_coord.is_relative is **False**, this is a pixel value. You can use negative values to specify a distance from the top edge. • If y_coord.is_relative is **True**, allowed values are between 0.0 (bottom edge of window) and 1.0 (top edge).

Add a custom logo

You can add any number of custom logos to your maps and globes, adding each one using a **cobrand_info** snippet. The ideal size for your logo is 64 x 64 pixels.

To add a custom logo:

1. Upload your logo file to a web server in the network. To host the file on your GEE Server (for example, `http://servername/filename.ext`), copy it to `/opt/google/gehttpd/htdocs`.
2. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
3. Sign in with the default credentials or the username and password assigned to you:
    - Default username: *geapacheuser*
    - Default password: *geeadmin*

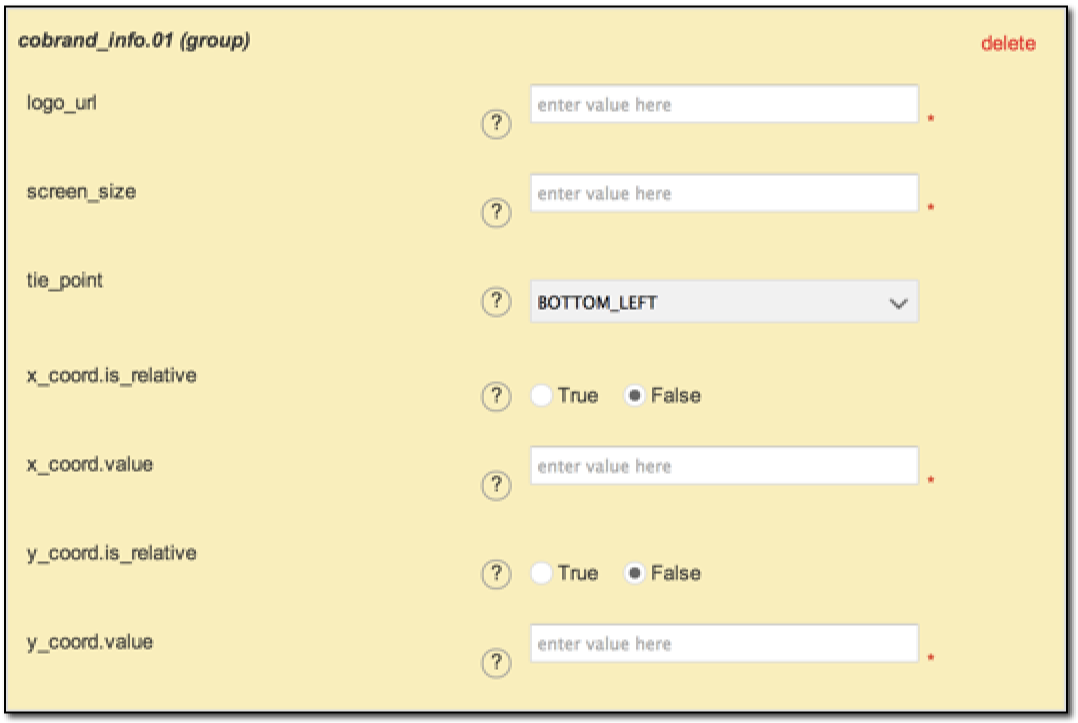
Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.

---

4. Click **Snippet profiles** to display your snippet profiles.
5. Click **Create New**. The **Create new snippet profile** dialog appears.

6. Enter a name for the new snippet profile and click **Create.** The snippet profile name appears in red in the **Existing snippet profile** list and the **Snippet editor** opens.
7. Click the **Add a new snippet set to the profile** drop-down to display the list of available snippets.
8. Select the **cobrand_info** snippet from the list.

9. Enter a URL for the path of the custom logo. Select whether your x and y coordinate values are expressed as relative or absolute and enter the parameter values. All fields must be completed.



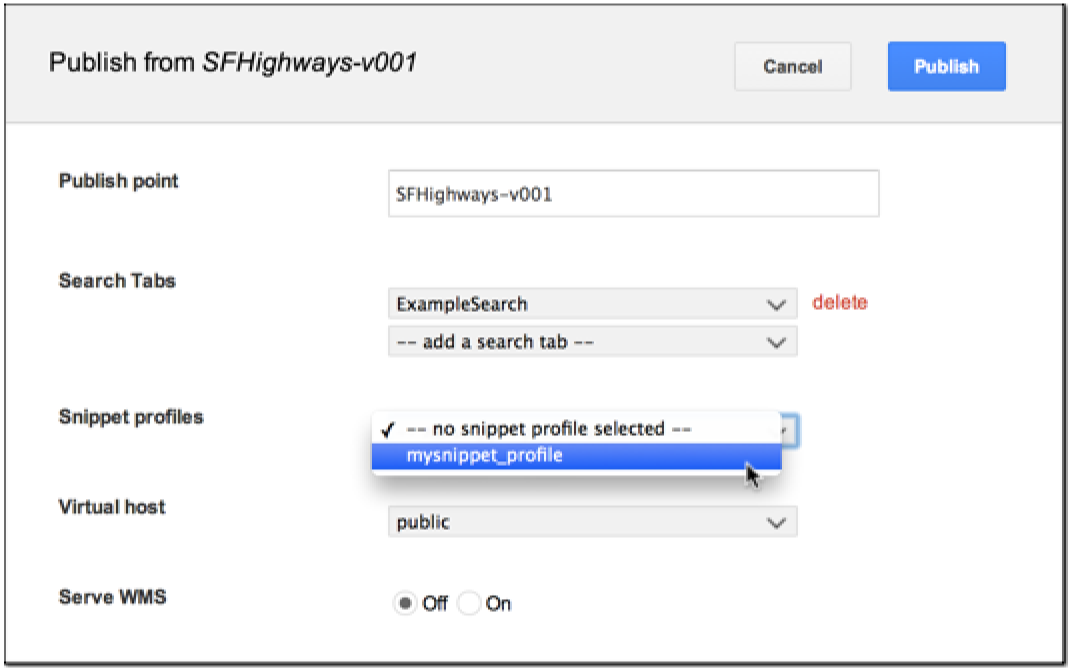
Click **Save changes** to save the snippet profile.

To apply a custom logo to a globe or map:

1. In the GEE Server Admin console, click **Databases**. The list of databases on GEE server appears.
2. Check the box next to the database to which you want to apply your custom logo. Click **Publish**. The Publish

dialog appears.

3. Select your snippet profile for your custom logo from the **Snippet profiles** drop-down list.



If you have already published your database, you need to **Unpublish** before publishing again, this time adding your snippet profile to apply your custom logo.

4. Click **Publish** to publish your database with the added snippet profile.

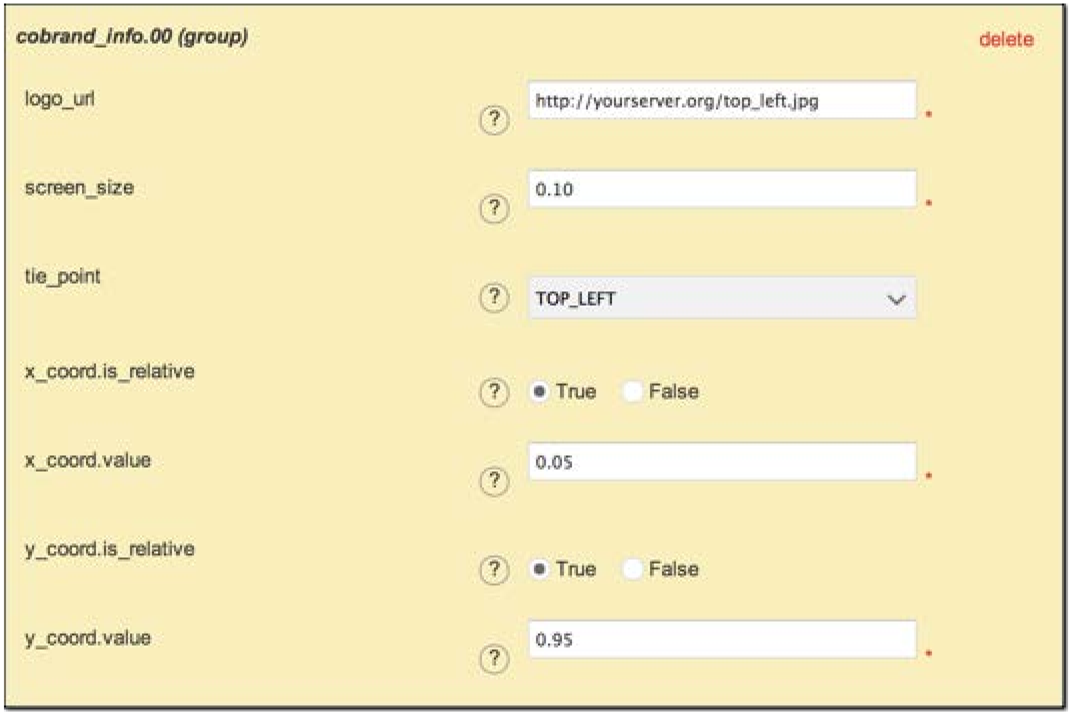
Now, when you view your database in Google Earth EC, your custom logo is displayed.

Google Earth EC recognizes only the snippet profile settings made by the first database that you connect to. This applies when you are connecting to multiple databases.

cobrand_info snippet definition examples

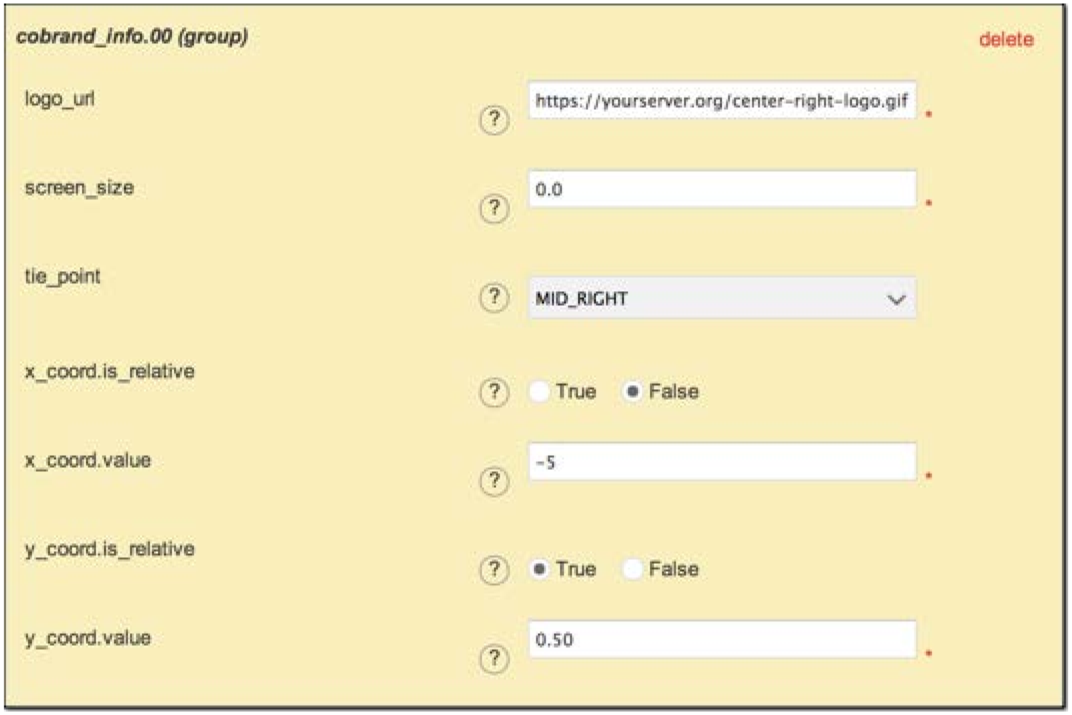
Single logo at top left

This example places one logo in the upper-left corner of the window (the top left of the image is 5% of the window width from the left side of the window, and 95% from the bottom), and sets it to dynamically scale to 10% of the window width as the window is resized.



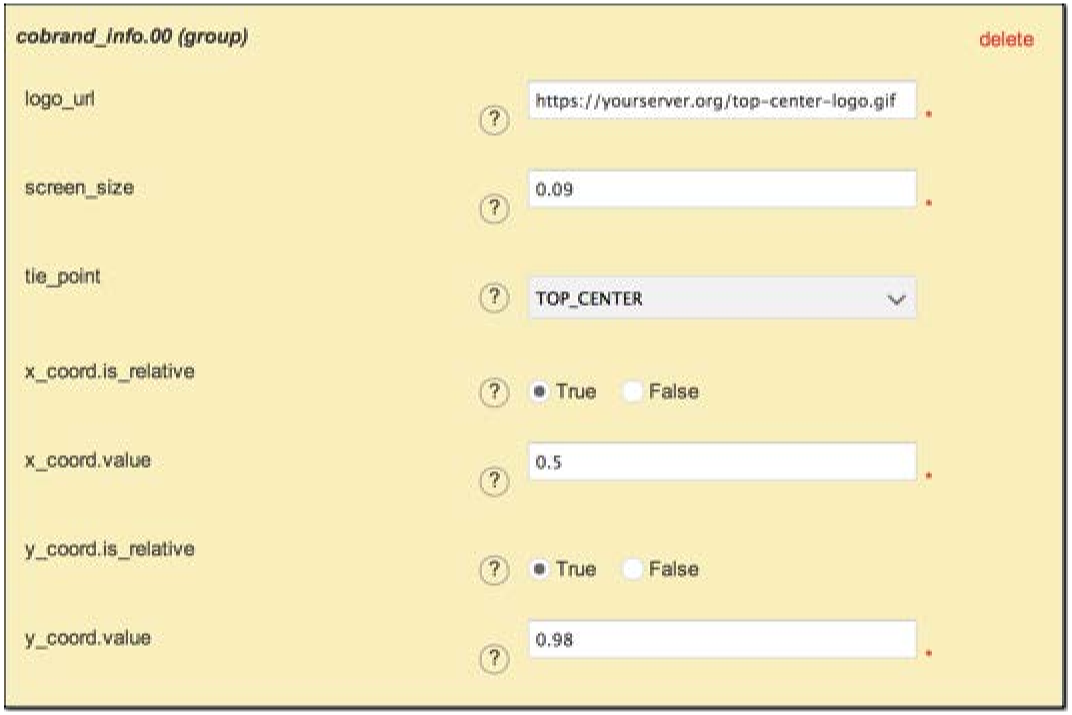
Single logo at mid right

This example places one logo at the vertical midpoint on the right side of the window using absolute and relative coordinate values. The center-right logo tiePoint is placed 5 pixels from the right window edge and at 50% of the relative window height. EC requests the file using HTTPS (as specified in **logo_url**) and doesn't scale the logo.



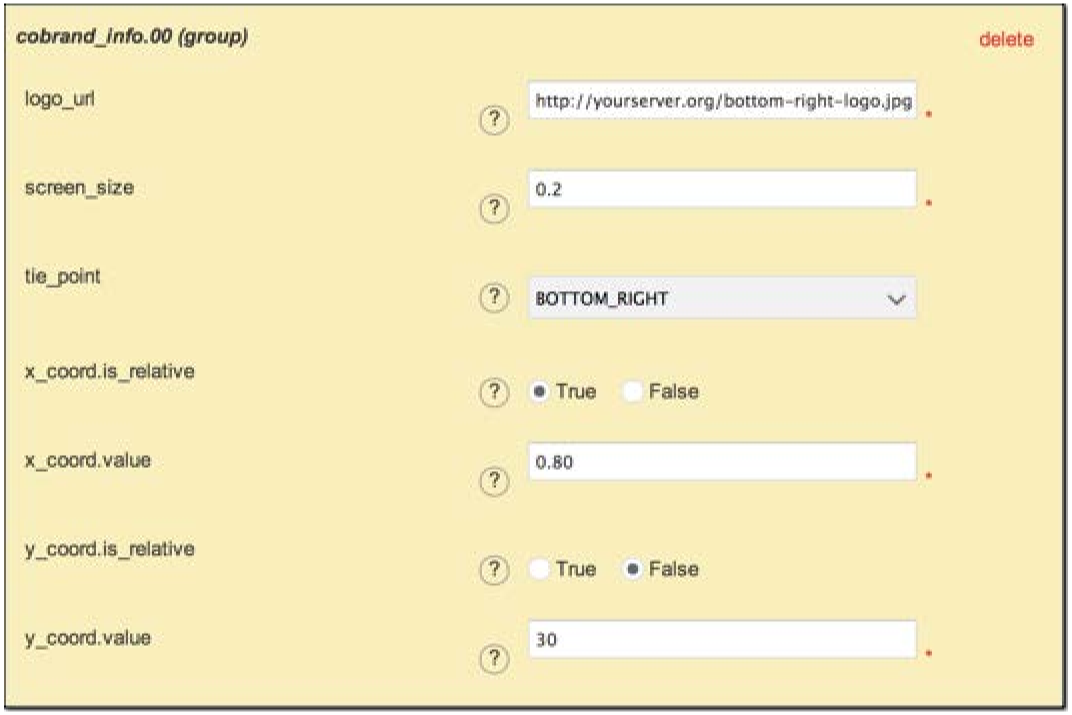
Single logo at top center

This example places one logo at the top center of the window. The top-center tiePoint of the logo is placed, by relative coordinates, at the 50% window width and 98% window height, and the logo dynamically scales to 9% of the window width.



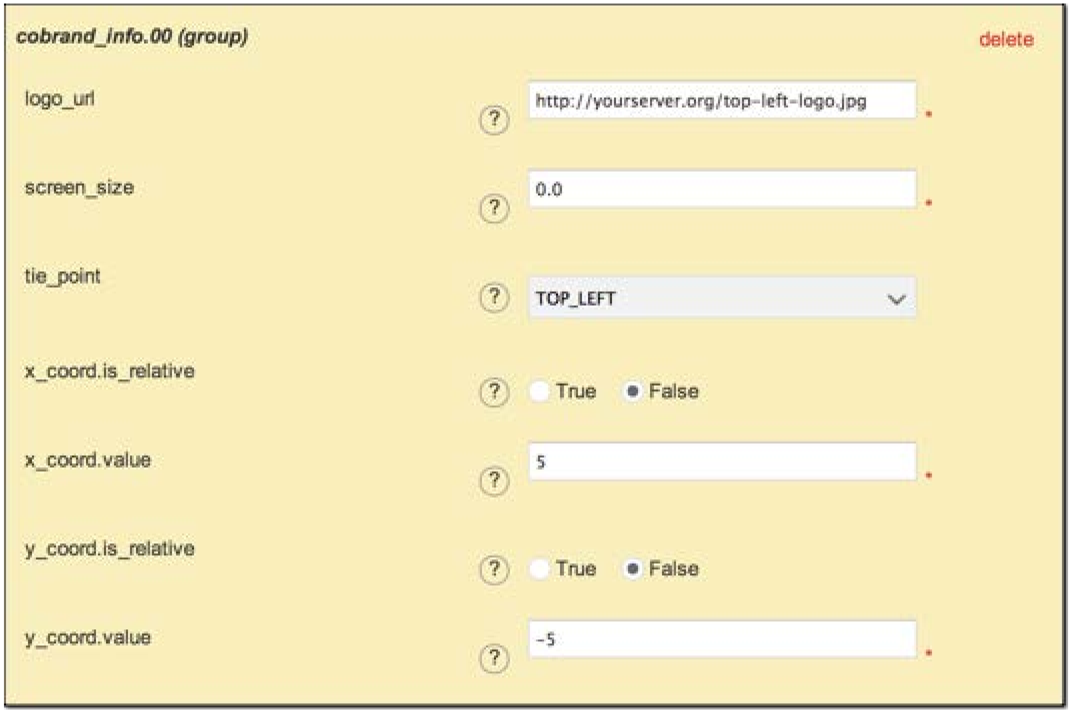
Single logo at lower right corner

This example places one logo in the bottom right corner of the window by combining relative and absolute coordinate values. The bottom-right logo tiePoint is placed at 80% of the window width from the left window edge, leaving a 20% margin at the right window edge, and 30 pixels from the bottom window edge. The logo is dynamically scaled to occupy 20% of the window width.



Single logo at upper left corner

This example places one logo in the upper left corner of the window using absolute pixel values. The top-left logo tiePoint is placed 5 pixels from the left window edge and 5 pixels from the top window edge, and the logo is displayed without scaling.



Learn more

- Managing snippet profiles

Back to top

Manage virtual hosts

A *virtual host* combines Apache virtual host functionality with publisher mapping within GEE Server so that you can publish your databases securely. You create and register a virtual host in GEE Server, assign a security protocol to it, and then associate multiple databases, including portable globes and maps, with that single virtual host. Add another virtual host when you want to use a different authentication and to associate different databases.

For example, you may have one or two security protocols but many databases and portable files to publish. Once you've created a virtual host for each security protocol and set up the appropriate authentication, you can associate different databases and portable globes and maps with the different virtual hosts, depending on the security protocols you need to apply.

You can also use the default virtual host configurations provided with GEE Server.

See Configure GEE Server 5.1.0 for SSL/HTTPS to add virtual hosts for HTTPS servers.

See geserveradmin in the Command Reference for information about all the virtual host commands.

Note: You may configure a virtual server in Apache using standard virtual host directives but it won't be capable of hosting a GEE database until it is registered using `geserveradmin`, as described in Create and register a virtual host in GEE Server.

- Default virtual hosts
- Create and register a virtual host in GEE Server
- Configure virtual hosts with a custom port number

- Associate a database with a virtual host
- Naming virtual hosts

Default virtual hosts

By default, GEE Server has two registered virtual hosts: *public* and *secure*. The secure virtual host is a protected virtual host with basic authentication. The default credentials are:

- Username: *geeuser*
- Password: *geeuser*

To reset the username and password:

```
sudo /opt/google/gehttpd/bin/htpasswd
 /opt/google/gehttpd/conf.d/virtual_servers/.vhpasswd geeuser
```

The public virtual host is unprotected and does not require any authentication.

Create and register a virtual host with digest authentication

You create and register a virtual host using the `geserveradmin --addvh` command. This action results in a location-based virtual host for which you can configure authentication and create a password for a given user name. The following example shows how to set up *digest authentication*, one of the standard authentication types (`AuthType`) provided by Apache server. See Apache HTTP Server 2.2 htdigest.

In GEE 4.4, virtual hosts were previously called *virtual servers*. Now, in GEE 5.0, virtual hosts specify a security protocol only and can also be associated with multiple published globes and maps. This change decouples the task of setting up security protocols from publishing maps and globes.

To create and register a virtual host with digest authentication:

1. Create and register virtual host in GEE Server using the `geserveradmin` command:

   ```
   /opt/google/bin/geserveradmin --addvh digest [--vhurl <url>] [--vhcachelevel
   ```

```
<level>]
```

`geserveradmin` creates the default location-based virtual host `digest_host.location` in `/opt/google/gehttpd/conf.d/virtual_servers` and registers it in GEE Server.

The `vhurl` specifies the location of the virtual host. It must match the corresponding server-side virtual host configuration.

2. Optionally, modify virtual host settings in `digest_host.location` to set up authentication.

To set up digest authentication:

1. Edit the `digest_host.location` file to set up the authentication configuration:

   ```
   cd /opt/google/gehttpd/conf.d/virtual_servers
   sudo vi digest_host.location
   ```

2. Add the following content to the `digest_host.location` file:

   ```
   # The digest virtual host.
   RewriteEngine on

   <Location /digest_host/>
     SetHandler fdb-handler
     AuthType Digest
     AuthName "Private"
     AuthDigestProvider file
     AuthUserFile /opt/google/gehttpd/conf.d/virtual_servers/.htdigest
     Require valid-user
   </Location>
   ```

3. Create the password with a given user name using the password path that you specified in the `digest_host.location` file:

   ```
   sudo htdigest -c /opt/google/gehttpd/conf.d/virtual_servers/.htdigest Private
   username

   sudo chmod 755 /opt/google/gehttpd/conf.d/virtual_servers/.htdigest
   ```

```
#Enter password twice at prompt
```

4. Restart geserver after virtual host settings have been modified.

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

A virtual host can also be configured for SSL/HTTPS. See Configure GEE Server 5.1.0 for SSL/HTTPS.

Configure virtual hosts with a custom port number

You may need to configure a virtual host to use a custom port number, commonly to differentiate from any default port (http:80, https:443) setting you may be using or when standard port numbers are being used for other purposes. With some edits to your Apache configuration files, you can set up custom port numbers for any location-based virtual host.

When creating a location-based virtual host with a custom port number, you need to rename and manually include the newly created location file, *.location or *.location_ssl in <Apache path>/conf.d/virtual_servers/ into the corresponding <Virtual Host> section of the Apache configuration file and restart GEE server.

To add a virtual host with a custom port number configured for SSL:

1. Run geserveradmin on the command line, in this example, adding a virtual host my_custom_port_ssl with a virtual host URL and custom port number:

   ```
   # geserveradmin –addvh –ssl my_custom_port_ssl –vhurl
   https://myservername.com:4343
   ```

2. Change the extension of the newly created my_custom_port_ssl.location_ssl configuration file to differentiate from any default port (443) SSL virtual hosts you may have configured:

   ```
   <Apache
   path>/conf.d/virtual_servers/my_custom_port_ssl.location_ssl_custom
   ```

3. Then manually add the Include directives for your custom port virtual host to the SSL configuration file for Apache server, /opt/google/gehttpd/conf/extra/httpd-ssl.conf:

   ```
   <VirtualHost  non default :4343>
   ```

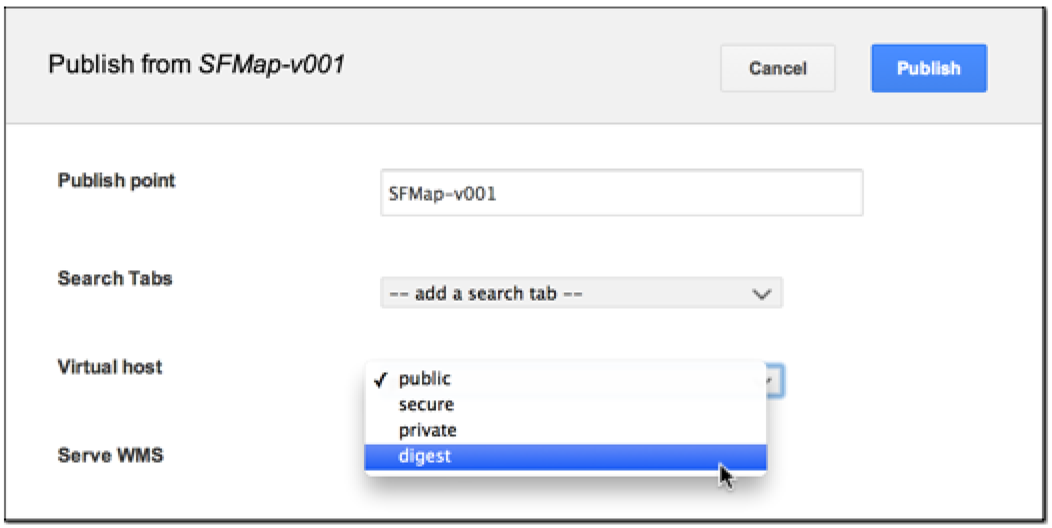
```
    # Include all SSL location-based virtual servers with custom port
4343.
    Include conf.d/virtual_servers/*.location_ssl_custom
</VirtualHost>
```

4. Restart GEE Server:

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

Associate a database with a virtual host

To associate a database or portable globe or map with a virtual host, you specify the virtual host when you publish in GEE Server.



See Publishing databases and portables.

Naming virtual hosts

Virtual hosts on GEE Server are “name-based,” meaning that you can have multiple “names” that serve assets running on a single IP address. With name-based virtual hosting, GEE Server relies on the client to report the host name as part of the HTTP headers. Using this method, many different hosts can share the same IP address.

Although usage of the virtual host names is hidden, reusing the name for other assets being served can cause conflicts. To that end, GEE Server checks the target path when you create a virtual host against reserved words and virtual host paths already registered in system, but does not check the virtual host name itself. Here is the list of reserved words that we check against: "fdb", "htdocs", "admin", "cutter", "earth", "icons", "js", "maps", "portable", "shared_assets."

Learn more

- Configure GEE Server for SSL/HTTPS
- LDAP authentication

Back to top

Google

Configure GEE Server 5.1.0 for SSL/HTTPS

Data transmission between Google Earth EC and GEE Server occurs on unencrypted HTTP by default. However, you may have strict requirements that secure HTTP (HTTPS) be used for all data communications. This article provides the steps to configure a GEE Server release 5.1.0 for use with HTTPS.

We also include the steps required to generate a self-signed SSL certificate for your server but we recommend you obtain a third-party certificate from a CA (Certificate Authority). Third-party certificates generally are trusted and do not lead to any issues with warning messages or exceptions. However, you may want to set up your own self-signed certificates to get up and running quickly.

Requirements

- Google Earth Enterprise Server 5.1.0
- A third-party or self-signed SSL certificate. Instructions for generating the latter are provided in the following setup procedure.

Generate self-signed SSL certificate and key

A self-signed server certificate is generated for demonstration purposes in the following steps. If you're using CA-verified server certificates and keys, see the following section, Apply third-party/CA-verified certificates and keys.

To generate a self-signed SSL certificate and key:

1. Change directory to the default certificate folder:

   ```
   cd /opt/google/gehttpd/conf
   ```

   The default SSL certificate and key files generated in the following steps and used in this example virtual host are `/opt/google/gehttpd/conf/server.crt` and `/opt/google/gehttpd/conf/server.key` respectively.

   Your certificate location and names may be different but make sure that they match the entries in the `httpd-ssl.conf` file, as shown in Set your virtual host as a SSL server.

2. Generate the server key:

   ```
   openssl genrsa -out server.key 1024
   ```

   It is recommended that you do not use the `-des3` option, which adds password protection when a key is created. While this adds an extra layer of security, it also requires manual input of the password should your system accidentally power down and restart, for example. Instead, generate the server key without a password or strip out the password with `openssl rsa -in server.key -out myservername_nopasswd.key` and use that instead.

3. Generate the server certificate based on the server key:

   ```
   openssl req -new -x509 -days 365 -key server.key -out server.crt
   ```

   Include as much information into the certificate as desired or accept the defaults, that is, Country, State, City, Company Name, Department, Server Name, Administrator email address.

4. Test the server certificate and verify all information is correct:

   ```
   openssl x509 -noout -text -in server.crt
   ```

Apply third-party/CA-verified certificates and keys

If you are using third-party/CA-verified certificates and keys, we recommend renaming them to use the default names for the virtual host configuration:

1. Change your third-party server certificate file name to `SSLCertificateFile /opt/google/gehttpd/conf/server.crt`
2. Change your third-party/CA verified key file name to `SSLCertificateKeyFile /opt/google/gehttpd/conf/server.key`

Optionally, if you choose not to use the default certificate and key names, you'll need to modify the entries in `/opt/google/gehttpd/conf/extra/httpd-ssl.conf` with the custom names accordingly, listed under `# Server Certificate` and `# Server Private Key` respectively.

Setting up SSL/HTTPS

In this example procedure, you perform the following steps:

- Add a virtual host `ssl`
- Set up the Apache server configuration to serve virtual hosts over HTTPS.
- Restart GEE Server

Note: The virtual host name "secure" is reserved for GEE Server use.

To add a virtual host for HTTPS serving:

1. Register your new virtual host using the `geserveradmin` command. See Manage virtual hosts.

   ```
   geserveradmin —-addvh --ssl
   ```

   The **--ssl** option registers the newly created virtual host by creating a configuration file with the naming convention: **_host.location_ssl** located in the path `<Apache path>/conf.d/virtual_servers/`.

   For example, to create a location-based virtual host with a configuration file that specifies SSL:

```
# /opt/google/bin$ ./geserveradmin --addvh test_ssl --ssl
Registering Virtual Host: test_ssl ...
Virtual Host registration successful.
Location-based Virtual Host created:

/conf.d/virtual_servers/test_ssl_host.location_ssl
```

2. The newly created virtual host configuration file in this example, `/opt/google/gehttpd/conf.d/virtual_servers/test_ssl_host.location_ssl`, includes the `<Location>` directives for SSL, in this case, `test_ssl`.

   ```
   <Location "/test_ssl_host/*">
       SetHandler fdb-handler
       SSLRequireSSL
       SSLVerifyClient none
   </Location>
   ```

   Use of the `SSLRequireSSL` directive prevents all HTTP requests that don't use SSL, thereby protecting your data from all but HTTPS requests. See Apache HTTP Server Version 2.2 Documentation for more information.

   Use of the `SSLVerifyClient` directive specifies the level of certificate verification required for the client. See Apache HTTP Server Version 2.2 Documentation for more information.

Set your virtual host as a SSL server

All commands must be executed as the root user unless otherwise specified.

To set your virtual host as a SSL server:

1. Edit the Apache HTTP server configuration file, `/opt/google/gehttpd/conf/gehttpd.conf` file, as

follows:

1. Uncomment and change `ServerName www.example.com` to `ServerName` *`MyServerName`*, where *`MyServerName`* is the real address users would enter in the network.
2. Check that `Include conf/extra/httpd-ssl.conf` appears and uncomment it. Note that this `Include` for the `httpd-ssl.conf` configuration is commented out by default as it should only be loaded if you serve a virtual host over HTTPS.
3. Save and close the `/opt/google/gehttpd/conf/gehttpd.conf` file.

2. Edit the Apache server configuration file, `/opt/google/gehttpd/conf/extra/httpd-ssl.conf` file, which provides SSL support. It contains the configuration directives to instruct the server how to serve pages over an HTTPS connection. For detailed information about these directives see Apache 2.2 documentation .
   1. Ensure the `ServerName www.example.com` is uncommented and matches that defined in the `/opt/google/gehttpd/conf/gehttpd.conf` file, that is, the alias or real address users would enter in the network.
   2. Check that the SSL virtual hosts configuration file location is already included in the `<VirtualHost _default_:443>` list of directives:

      ```
      <VirtualHost_default_:443>
           Include conf.d/virtual_servers/*.location_ssl
      ```

   3. Save and close the `/opt/google/gehttpd/conf/extra/httpd-ssl.conf` file.
3. Restart the Google Earth Enterprise Server software:

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

4. Publish a database to the SSL/HTTPS virtual host.
5. Test the connections with Google Earth Enterprise Client for HTTP and HTTPS-based virtual servers.

Back to top

Google

Configure a GEE virtual host for LDAP authentication

You can use LDAP authentication, which is based on Microsoft Active Directory, to authenticate users to access your GEE virtual hosts. GEE server software is based on Apache `httpd`.

It's easy to administer and manage user accounts. You can use a simple `.htpasswd` to secure a GEE virtual host, then you can assign the permissions levels via groups or user level. If you have an existing Active Directory user catalog, you can use that instead of implementing a redundant authentication system. For information about enabling your Active Directory for LDAP, see the OpenLDAP documentation.

This article explains how to configure a virtual host for LDAP authentication using GEE Server 5.x. For GEE Server 4.4.1, see *Authentication* in the *Google Earth Enterprise Administration Guide, Configuring GEE Server*.

To configure a GEE virtual host for LDAP authentication:

1. Make sure that the LDAP modules are loaded on your Apache server. For most GEE installations, these are loaded automatically.
2. Create a `geserver` Active Directory account that can access the user directory, and make sure to document the attributes.
3. Load the modules in your `/opt/google/gehttpd/conf.d/virtual_servers/private_host.location` file or your custom virtual host file.

   After the modules are loaded, you can control access by querying the directory for particular attributes.
4. Point Apache to the LDAP server `AuthLDAPUrl` key directive.

An example of an `AuthLDAPUrl` directive is:

```
AuthLDAPUrl ldap://ldap.company.com/ou=People,dc=company,dc=com?uid
```

The key directive format declares the LDAP server, the distinguished name (DN), and the attribute to use in the search (typically the `Uid` attribute in the `People` organizational unit). You can also customize these filters for additional security constraints.

5. Restart the GEE server to load the modules into the system:
   ```
   # /etc/init.d/geserver restart
   ```
6. Create a user account in Active Directory `geserver`.
   This can be any user account that has limited access. The user account only needs access to the global catalog. Make sure that your user has the full organizational unit name.

To bind to Active Directory GUI:

1. If you don't know your LDAP domain name, ask your system administrator. You can also find your domain name by searching for it in the format `Dn: CN=geserver,CN=Users,DC=location,DC=company,DC=com`. In this example, the domain name is *`location.company.com`*.
2. Use your domain name to configure the Apache module. Alternatively, you can use the command prompt:
   ```
   dsadd user UserDN [-samid SAMName] -pwd {Password|*}
   ```
3. Add the directives to your virtual server file at `/opt/google/gehttp/conf.d/virtual_servers`.
4. For testing purposes, add the lines below to the `private_host.location` file:
   ```
   <Location /private/*>
      AuthType Basic
      AuthName "LDAP LOGIN"
      AuthBasicProvider ldap
      AuthLDAPURL "ldap://server.name.local:389/cn=Users,dc=domain,dc=google,dc=com?
   uid"\
      AuthLDAPBindDN CN=geserver,OU=Users,OU=Yourorg,DC=DcNAME,DC=local
      AuthLDAPBindPassword localuserpassword
      AuthzLDAPAuthoritative Off
   ```

```
  Require valid-user
</Location>
```

5. Restart your server and try to access the virtual server via the client.
   A login screen appears. If you can't access Apache, open the file error log at `/opt/google/gehttpd/logs/error_log and access_log`.
   If `AuthzLDAPAuthoritative` is on, the client might continually prompt for the username and password even though you entered a valid username and password and the client is functioning. To correct this, turn `AuthzLDAPAuthoritative` off.

Directives

- `require valid-user`. Allows all users who log in with correct passwords.
- `AuthzLDAPAuthoritative`. Tells Apache whether a failed authentication request can be passed to other Apache modules.
- `AuthLDAPBindDN`. The distinguished name (DN) of the user account that Apache uses to connect to the directory system and authenticate the user.
- `AuthLDAPBindPassword`. The password for the user account configured with the `AuthLDAPBindDN` directive.
- `AuthLDAPURL`. The URL that tells where the directory server is, where to look for users at, which user attribute is used to identify a user, and other miscellaneous things that are specific to the LDAP query syntax.
- `AuthBasicProvider`. Tells Apache which authentication module to use for Basic Authentication.

For more information, see the Apache mod_ldap and mod_auth_basic documentation.

To troubleshoot any connection issues, begin by checking the logs on the LDAP server to make sure GEE server is correctly making authentication requests.

Back to top

Google

Make Web Map Service (WMS) requests

Google Earth Enterprise Server 5.0 supports the OpenGIS Web Map Service Interface Standard (WMS), which provides a standard HTTP interface to request map images from one or more published geospatial databases. Mapping images that are provided through a WMS over the Internet can show information such as weather formations and conditions, topographical maps, alternate high resolution satellite imagery and more.

One of the benefits of using the WMS standard is that supported clients can request images from multiple WMS servers and then combine those mapping images into a single view. Because the WMS standard is used to get all the images, they can easily be overlaid on one another. Supported clients include QGIS 2.0.1, ArcGIS/ArcGIS Explorer Desktop, and Google Earth Pro/EC.

Supported Database Types

The following database types are supported:

- Fusion databases: 2D Plate Carrée and Mercator
- Portable databases: 2D (.glm) and 3D (.glb)

WMS requests for 3D databases serve the imagery layers only.

Supported Projections

For imagery over WMS, GEE Server supports **Plate Carrée (EPSG code 4326)** and **Mercator (EPSG code 3857)**, as well as legacy Mercator codes **EPSG:3785**

GEE Server Overview

Sign in to GEE Server Admin console

Publish databases and portables

Create Search tabs

Manage snippet profiles

Display custom logos

Manage virtual hosts

Configure GEE Server 5.1.0 for SSL/HTTPS

Configure a GEE virtual host for LDAP authentication

Make Web Map Service (WMS) requests

Create portable globes and maps

Create composite globes and maps

View 3D databases

Implement Earth Plugin

Provide Earth Plugin for your users to install

View 2D databases

Clearing the Google Earth EC cache

Admin console pages

- Snippet profiles page
- Search tabs page
- Settings page
- Dashboard page
- Databases page

and **EPSG:900913**.

Working with Other Projection Types

To work with other projection types, you will need to perform coordinate transformation in your client using proj.4 or another library . Alternatively, you can use an implementation of the OGC WCTS to reproject your coordinates to one of the supported ones. It's recommended that you use one of the well-known libraries since you are only transforming two pairs of coordinates for a bounding box.

Another example of working with other projections is to use a third-party application such as MapServer to reproject your native WMS on-the-fly, for example, to change your projection from Mercator to UTM zone 35.

Supported Image Formats

The following image formats are supported:

- PNG
- JPEG

Setting up Google Earth Enterprise Server 5.0 Support for WMS

To enable WMS for a database that you want to publish:

1. In the GEE Server Admin console, click **Databases**.
2. Check the box next to the database you want to publish.
3. Next to **Serve WMS**, select **On**.
4. Click **Publish**.

   Publishing a 2D or 3D database with the **Serve WMS** setting turned on publishes both GEE and WMS; it's not necessary to publish twice.

Making WMS Requests to Google Earth Enterprise Server 5.0

The first step in creating maps via HTTP requests to Google Earth Enterprise Server 5.0 is to create a connection.

To connect to Google Earth Enterprise Server 5.0, use the following URL:

```
http://<gee_server_name or ip address>/<target_path_of_published_db>/wms
```

In this example, `http://localhost/merc/wms`, “merc” is the target path of the published database. Google Earth Enterprise Server 5.0 supports the **GetCapabilities** and **GetMap** requests for WMS versions 1.1.1 and 1.3.0. Use **GetCapabilities** to first request metadata from which you can then specify the map images that you want to request using **GetMap**.

GetCapabilities

The **GetCapabilities** operation requests information about the type of services and data (“capabilities”) that are provided by Google Earth Enterprise Server 5.0.

The URL the GIS client should use to make the **GetCapabilities** request is as follows:

```
http://<hostname>/<target_path>/wms?SERVICE=WMS&REQUEST=GetCapabilities&VERSION=x.y.z
```

The parameters for this operation are as follows:

GetCapabilities parameters

Parameter	Mandatory?	Description
SERVICE	Yes	Service name. Value is WMS
REQUEST	Yes	Operation name. Value is **GetCapabilities**
VERSION	No	Service version is one of 1.1.1 or 1.3.0. If no version is submitted, the WMS request defaults to the highest serving version, in this case 1.3.0.

An example **GetCapabilities** request is:

```
http://localhost/merc/wms?SERVICE=WMS&REQUEST=GetCapabilities&VERSION=1.1.1
```

The response to a **GetCapabilities** request includes the following information:

- Image formats supported on a published target path basis for the maps. Note that the target path database information is sent in the connection URL.
- A list of layers supported by the published target path database. Each layer provides the following information:
  - Name of the layer
  - Spatial projection supported
  - Bounding box limitations based on the projection type
- Formats supported for any exceptions, such as invalid requests (requests not as per the international standards), service not supported etc.

An example **GetCapabilities** request output is as follows:

```
<WMT_MS_Capabilities updateSequence="0" version="1.1.1">
   <Service>
     <Name>OGC:WMS</Name>
     <Title>Google Earth WMS service.</Title>
     <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
     xlink:type="simple" xlink:href="http://108.59.84.128/Earth/wms"/>
   </Service>
   <Capability>
     <Request>
       <GetCapabilities>
         <Format>text/xml</Format>
         <DCPType>
           <HTTP>
             <Get>
               <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
               xlink:type="simple" xlink:href="http://108.59.84.128/Earth/wms"/>
             </Get>
           </HTTP>
         </DCPType>
       </GetCapabilities>
       <GetMap>
         <Format>image/png</Format>
         <Format>image/jpeg</Format>
         <DCPType>
           <HTTP>
             <Get>
               <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
               xlink:type="simple" xlink:href="http://108.59.84.128/Earth/wms"/>
             </Get>
           </HTTP>
```

```
        </DCPType>
      </GetMap>
    </Request>
    <Exception>
      <Format>application/vnd.ogc.se_xml</Format>
    </Exception>
    <Layer opaque="0" noSubsets="0" queryable="0">
      <Title>Google Earth WMS service.</Title>
      <Layer opaque="1" cascaded="0" noSubsets="0" queryable="0">
        <Name>Earth</Name>
        <Title>Imagery</Title>
        <SRS>EPSG:4326</SRS>
        <LatLonBoundingBox minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"/>
        <BoundingBox maxx="180.0" maxy="90.0" miny="-90.0" minx="-180.0"
        SRS="EPSG:4326"/>
      </Layer>
    </Layer>
  </Capability>
</WMT_MS_Capabilities>
```

Once you have retrieved services and data information from Google Earth Enterprise Server 5.0 using **GetCapabilities**, you can then get the map image that you want using **GetMap**.

GetMap

The **GetMap** operation requests the map from Google Earth Enterprise Server 5.0, based on the layer data that was discovered in the **GetCapabilities** step.

The URL the GIS client should use to make the **GetMap** request is as follows:

```
http://<hostname>/<target_path>/wms?SERVICE=WMS&REQUEST=
GetMap&BBOX=<bounding_box_limits_of_the_requested_map>&SRS=<projection_type>&WIDTH=<width_of_requested_map>&HEIGHT=
<height_of_requested_map>&LAYERS=<layer_name>&STYLES=<style_info>&FORMAT=
<format_of_requested_map>&DPI=96&TRANSPARENT=TRUE
```

The parameters for this operation are as follows:

GetMap Parameters

Parameter	Mandatory?	Description
SERVICE	No	Service name. Default value is WMS.
REQUEST	Yes	Operation name. Value is **GetMap**.
VERSION	No	Service version is one of 1.1.1 or 1.3.0. If no version is submitted, the WMS request defaults to the highest serving version, in this case 1.3.0.
LAYERS	Yes	Layer name for the requested map/image.
BBOX	Yes	Bounding box for map extent. Value is minx, miny, maxx, maxy in units of the SRS or CRS, depending on the version. Use SRS for version 1.1.1; use CRS for version 1.3.0.
SRS or CRS	Yes	Spatial Reference System (SRS) of map output. Value is in form EPSG:nnn. Use SRS for version 1.1.1; use CRS as the parameter key for version 1.3.0.
WIDTH	Yes	Width of map output, in pixels.
HEIGHT	Yes	Height of map output, in pixels.
FORMAT	Yes	Format for the map output. PNG or JPEG are currently supported.
STYLE	No	Styles in which layers are to be rendered. Value is a list of required style names or empty if default styling is required.
DPI	No	Dots per inch. Value is client-dependent. For example, the QGIS client defaults to 96dpi.
TRANSPARENT	No	Determines if the map should be transparent. Values are TRUE and FALSE; default is FALSE. This parameter only applies when requesting PNG images.

An example **GetMap** request is:

- For version 1.1.1:

  ```
  http://localhost/merc/wms?
  VERSION=1.1.1&SERVICE=WMS&REQUEST=GetMap&BBOX=-0037508.342789,-20039414.861177,20037508.342789,20037508.342781&SRS=EPSG:900913&WIDTH=1002&HEIGHT=1001&LAYERS=
  [merc]:1002&STYLES=&FORMAT=image/png&DPI=96&TRANSPARENT=TRUE
  ```

- For version 1.3.0:

  ```
  http://localhost/merc/wms?
  VERSION=1.3.0&SERVICE=WMS&REQUEST=GetMap&BBOX=-20037508.342789,-20039414.861177,20037508.342789,20037508.342781&CRS=EPSG:900913&WIDTH=1002&HEIGHT=1001&LAYERS=
  [merc]:1002&STYLES=&FORMAT=image/png&DPI=96&TRANSPARENT=TRUE
  ```

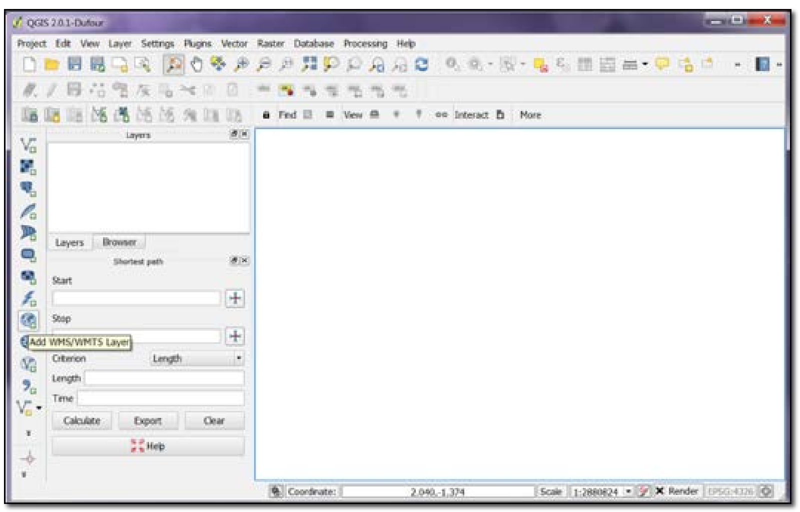
Making WMS Requests in GIS Clients

Any GIS client that supports WMS can query the Google Earth Server 5.0 using the URLs specified in this document. The following example shows how QGIS connects to Google Earth Server 5.0, requests the server's data, and then requests the maps to display.

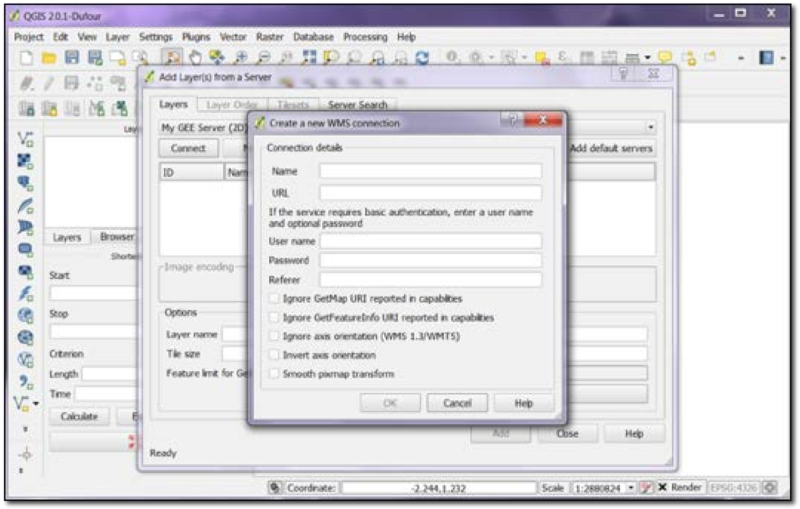
Connecting to Google Earth Server 5.0 WMS using QGIS

To connect the Google Earth Server 5.0 using the QGIS client:

1. Start the QGIS client application. At the top of the QGIS window, click the **Add WMS/WMTS Layer** toolbar button or from the **Layer** menu choose **Add > Add WMS/WMTS Layer...**.

2. Click **New** to establish a new WMS layer connection. A new connection window displays.



3. Enter the following details:

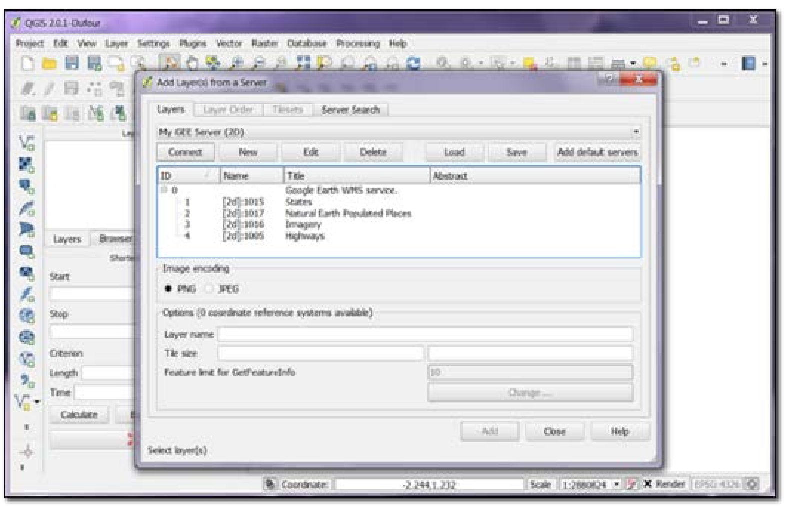
    - Name for the connection
    - Connection URL as `http://<server>/<target_path_of_published_database>/wms`, for example, `http://localhost/my_map/wms`

    Optionally, supply the username and password if the database has been published to a secure virtual host set up with basic authentication.

    Click **OK** to create a new connection.

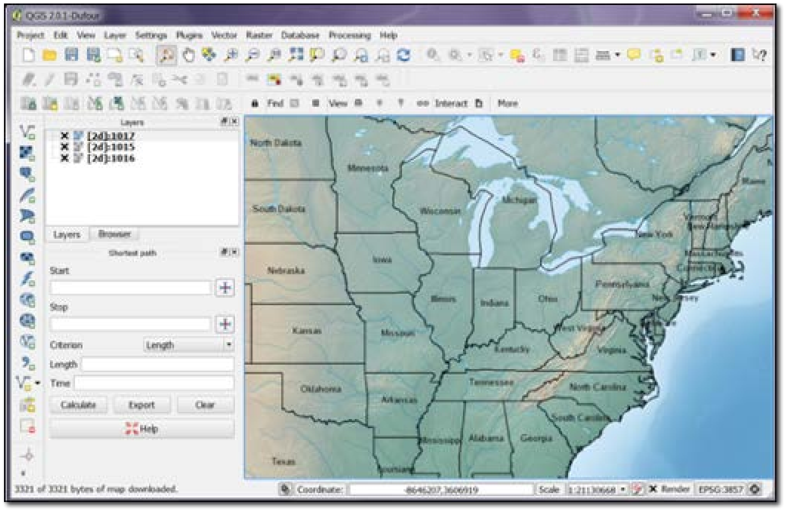
4. In the same window, click **Connect** to make a **GetCapabilities** request for a specific published target path database.

    An example response for a **GetCapabilities** request is shown in the following window.

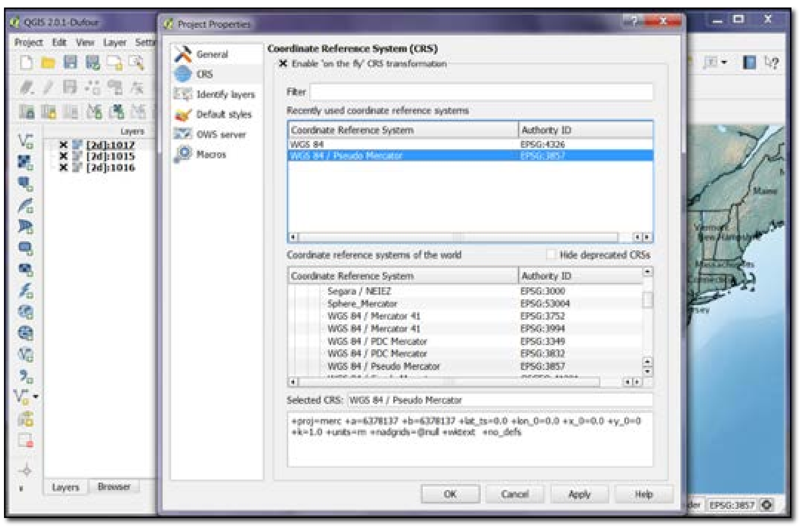


The **GetCapabilities** WMS request returns imagery and vector layers for 2D databases and only imagery for 3D databases.

5. Determine which layer you want to display, then double-click the layer name to send a **GetMap** request to Google Earth Enterprise Server 5.0. The following example shows a map retrieved from a **GetMap** request.



6. If you are adding layers from a 2D database to your QGIS project, be sure to first set the **Coordinate Reference System (CRS)** for the project to **EPSG:3857 (WGS 84 / Pseudo Mercator)** by clicking the icon in the lower-right corner of the display window.



Back to top

Create portable globes and maps

The portable globe cutter tool is installed with Google Earth Enterprise Server 4.2 and later. Globe cutting is disabled by default; to enable it:

- In GEE 5.0: `gecutter enable`
- In earlier versions: `geserveradmin --enable_cutter`

By default, the cut globes will be stored in the `/opt/google/gehttpd/htdocs/cutter/globes` directory. The cut globes can be large, and many servers do not have sufficient storage allocated in this directory for multiple globes. To designate a different directory as the storage area for the cut globes, see Custom folders.

Creating a portable globe

Cutting a globe is accomplished with a simple web interface. You'll use your mouse, or import KML, to define a polygon, which defines your 'area of interest.' This polygon not only defines the area that will display high-resolution imagery, but is also used by Fusion to create a localized search database.

Note: The globe cutting processes are CPU and disk intensive, as they are retrieving all data within the specified polygon from the Earth Enterprise Server. This can affect the overall performance of the Server, including slowing end-user access.

To mitigate performance impact to end users, you may consider:

- Limiting the number of users with access to globe cutting.
- Creating pre-cut portable globes to host as downloadable files for portable users.

- Operating a separate GEE Server specifically to support on-demand globe cutting needs.

Please contact the Google Earth Enterprise Support team for further information or questions about these procedures.

Before you begin

Enable the cutter

Before cutting a globe, you must enable the cutter from the command line:

- In GEE 5.0: `gecutter enable`
- In earlier versions: `geserveradmin --enable_cutter`

For more information about `geserveradmin`, see the Command Reference.

Note about authentication and SSL: Cutting is not currently supported on globes that require end-user authentication e.g. LDAP. One workaround is to allow unauthenticated access from localhost on your Earth Enterprise Server. (Contact Google Enterprise Support if you need sample Apache directives to enable such a configuration.) Cutting of a globe over HTTPS is supported; however the SSL certificate of the target server will not be verified during cutting.

Install the Google Earth Plugin

The Google Earth Plugin is used to display the globe from which maps are cut. To install the plugin, visit http://www.google.com/earth/explore/products/plugin.html and follow the prompts inside the Google Earth Plugin frame. If you require installers for offline installation, contact Google Enterprise Support.

The globe cutter interface

To create a portable globe:

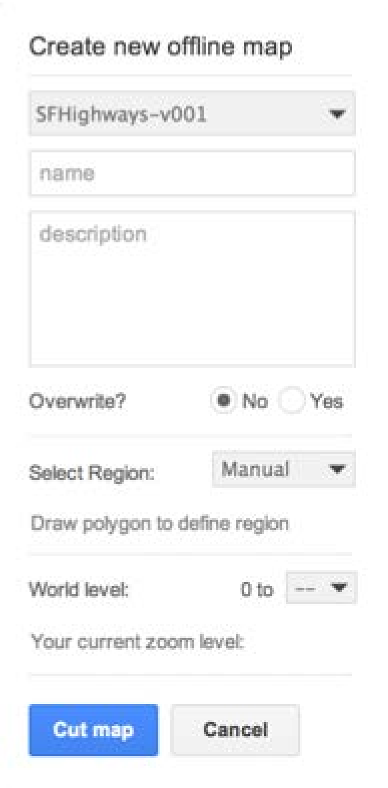
1. Access the Google Earth Enterprise Server Admin console in a browser window by going to *myserver.mydomainname*.com/admin, replacing *myserver* and *mydomainname* with your server and domain.
2. Sign in with the default credentials or the username and password assigned to you:
   - Default username: geapacheuser
   - Default password: geeadmin

Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.

3. Click **Databases** to display the list of databases pushed to the Server.

4. Click the **Settings** button in the top right of the window:  and choose **Launch Cutter** from the Tools menu.

   A new browser tab opens with the GEE Server - Cutting Tool and the **Create new offline map** window appears.



Map or globe name

1. Use the drop-down menu to select the database you'd like to cut to create your offline map or globe.

2. Enter a name for the offline map or globe.

   The name defines the filename for your offline map or globe. Offline maps are created as `.glm` files; offline globes are created as `.glb` files. Both file types are a single-file format for sharing Google Earth Enterprise maps and globes. Spaces, slashes, and double dots (..) will be converted to underscores in the saved globe name.

   Building an offline map or globe will overwrite any existing offline maps or globes with the same name. If multiple users are cutting maps or globes, we recommend assigning unique prefixes to each user for their globe names to ensure that files are not accidentally overwritten.

3. Enter a description to be associated with the offline map or globe.

   We recommend adding sufficient descriptive information for each offline map or globe, so that others will know what geographic area, or what mission, they were created for.

4. If you are overwriting an existing cut, click **Overwrite?** to **Yes**.

Drawing the polygon

Once the globe name has been specified, you can define the geographic region to be cut by drawing a polygon on the globe. There are two ways to draw the polygon.

Hand drawing the polygon

1. By default, you draw a polygon by hand so the Select Region drop-down list is set to **Manual**.

2. Use the Hand tool  to pan, then using the navigation controls in the plug-in, zoom in to the region of interest.

3. To use your mouse to define the polygon, click the polygon icon in the globe window:  .

4. Click on the map or globe to define each point. You can use the navigation controls on the right to move the globe or change zoom levels while drawing.

   Click the final point at the point of origin to complete the polygon selection.

If you need to draw the polygon again, click **Clear** to delete the polygon you just created.

Defining the polygon with KML

You can also use KML to define the polygon(s). The KML should be complete, and may contain single or multiple elements. To insert your KML:

1. From the Select Region drop-down list, select **Paste KML**. The Paste KML window appears.
2. Paste your KML into the text field, then click **Use KML**.

   GEE Server validates the KML and then draws the polygon using the KML data you provided. Your polygon appears on the map or globe.

   If you need to draw the polygon again, click **Clear** to delete the polygon you just created with pasted KML.

Globe resolution

The polygon you specify in the previous step defines your 'area of interest.' This area will contain high-resolution imagery and data, and search tabs will be created for information that lies within this zone. The maximum and minimum resolutions are specified as integers between 1 and 24. These correspond to the zoom levels that are used in the Fusion server. Setting a resolution of 24 results in a cut of the entire globe.

Caution: Setting a resolution of 24 to cut an entire globe may result in a very large file.

World level resolution

The area outside of the defined polygon is included in the globe at a lower resolution, which you set using **World level**. Areas near the polygon may be included at a higher resolution.

- To set the world level resolution, select a value from the **World level:** drop-down list.

A minimum zoom level of 5-7 will present a decent-looking world to the user and will most likely include vector layers such as international boundaries and state boundaries and main cities without affecting the size of the `.glb` file very much. For example:

- A cut globe with minimum and maximum resolution values set to 5 is 10MB.
- A cut globe with minimum and maximum resolution values set to 6 is 41MB.
- A cut globe with minimum and maximum resolution values set to 7 is 120MB.

These numbers are small in comparison to the overall size of your globe when a suitable maximum resolution has been selected. For example, a globe that contains all of the city of Atlanta, GA, USA in 1-foot resolution requires approximately 5GB of storage. Even level 7 imagery, at 120MB, is a small percentage of the overall globe size. You can also leave this field blank to use the highest available imagery.

Region level resolution

The zoom level for the polygon area is set using **Region level**.

The maximum resolution of the cut polygon area will be no higher than the maximum resolution of the source map or globe. For example, if the maximum resolution in the cutter is specified at 24, but the source imagery is at 18 (approximately 1-meter resolution), the cut map or globe will contain level 18 imagery. You can leave this field blank to use the highest available imagery.

You may enter a lower number to reduce the size of your map or globe, by not including the highest resolution imagery.

Advanced Polygon Resolution

The **Advanced** option provides an additional globe-cutting option, namely *Polygon Resolution*. This setting is useful when cutting with large polygons. For example, you may use 12 for a country-sized polygon or 18 for a city-sized polygon.

Note: Additional advanced settings may be offered in future versions. Use caution when changing them as they may dramatically increase build times and globe sizes.

To set the polygon resolution:

- Click **Advanced** to display the Polygon Resolution option.
- Click the drop-down list to set the resolution value you want.

Building the map or globe

Depending on the size of your polygon, building your cut map or globe can take from a few minutes to a few hours; likewise, file size will vary widely depending on the area selected and the desired resolution.

To build the map or globe:

- Click **Cut map** to start the build process.

  The progress of the build appears in the Build window

  When the build is finished, a `.glb` file is created in the default globes directory, `/opt/google/gehttpd/htdocs/cutter/globes`, and a download link appears to the file's location on GEE Server.

  KML files

  When a portable globe is cut from a source containing KML links in the Layer panel:

  - KML files that are stored locally on the primary Earth Server will be bundled into the portable globe. Only the main KML file will be copied, not any links or files that are embedded as links in the main KML file. The default copy is not recursive.
  - KML links that refer to servers other than the primary Earth Server are not copied. The layer will be visible in the client, but clicking the link will not cause any data to be displayed. If access to external

servers is needed, a small KML file should be stored locally on the primary Earth Server. This KML file should contain a link to the intended external server.

Historical Imagery

Historical Imagery is not supported in the portable globe as of Fusion 4.2. It is being investigated as a possible feature.

There are, however, two situations in which historic imagery will be displayed:

- When the computer running the portable globe has a connection to the Earth server from which the globe was cut. In this case, historic imagery can be streamed from the Earth server. Once in the field, however, and disconnected from the Earth server, no historic imagery will be displayed.
- If historic imagery has been cached on the portable globe machine.

Otherwise, the following error message will appear:

> **Google Earth can't contact the imagery server to download new images.**
> You'll be able to see areas that you've been to recently, but new image areas may appear blurry.

Learn more

- Serve a globe or map from GEE Portable
- Connect with GEE Portable for different ways you can connect to GEE Portable to view your offline maps and globes.
- Portable Developer Guide for ways to customize or extend GEE Portable, or create applications that work with it.

Back to top

Create composite globes and maps

Contents

Introduction

A composite globe (`.glc`) is an assembly of layers from existing portable globes or maps. A portable globe or map stores all the geospatial data available within your specified area of interest—your cut—including all high-resolution imagery, terrain, vector data, KML files, and searchable point of interest (POI) locations. Outside the specified area of interest, the globe or map stores only low-resolution imagery and terrain. You specify the levels of resolution when you cut the globe or map.

When you create a composite globe or map, you define your viewing area using a KML polygon, select layers, and in addition for globes, specify base globe options. One of the key benefits of using composite maps and globes is that you can apply layers that use different cuts or regions of interest, then assemble them to create one single portable map or globe.

For example, you may want to create a composite map for a hiking trip. You might cut a road map using one region of interest and then cut imagery and also a park map of the hiking trail using additional regions of interest. Ordinarily, cuts from map databases would result in separate `.glm` files. By assembling the `.glm` files into a composite `.glc` file, the result is a single portable map.

GEE Server Overview

Sign in to GEE Server Admin console

Publish databases and portables

Create Search tabs

Manage snippet profiles

Display custom logos

Manage virtual hosts

Configure GEE Server 5.1.0 for SSL/HTTPS

Configure a GEE virtual host for LDAP authentication

Make Web Map Service (WMS) requests

Create portable globes and maps

Create composite globes and maps

View 3D databases

Implement Earth Plugin

Provide Earth Plugin for your users to install

View 2D databases

Creating composite maps and globes

You can create composite files of portable globes and maps with the **Assembly tool** feature of the Google Earth Enterprise (GEE) Server. You provide a KML polygon definition to identify the ‘area of interest.’ This polygon not only defines the viewing area, but is also used by Fusion to create a localized search database.

You select 2D or 3D layers to add to the composite file from any of the cut maps or globes that you have available in your `/opt/google/gehttpd/htdocs/cutter/globes` directory. In addition, for globes, you specify a base globe.

New `.glc` files are added to the default `/opt/google/gehttpd/htdocs/cutter/globes` directory. If you want to change the location of your portable files directory, see Serving a globe or map from GEE Server.

If you want to extract the 2D or 3D layers from a composite .glc file, you use the Disassembly tool, specifying a target directory for the extracted files.

To assemble a composite map or globe:

1. Access the Google Earth Enterprise Server Admin console in a browser window by going to myserver.mydomainname.com/admin, replacing myserver and mydomainname with your server and domain.
2. Sign in with the default credentials or the username and password assigned to you:
    - Default username: *geapacheuser*
    - Default password: *geeadmin*

    Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator. Click Dashboard to view information about the portable files registered on and published to the Server.

    ---

3. Click the **Settings** button in the top right of the window and choose **Glc Assembly** from the Tools menu. The GLC Assembly tool window appears.
4. Enter a name for the offline map or globe.

    The name defines the filename for your composite map or globe. Composite maps and globes are created as .glc files. Spaces, slashes, and double dots (..) will be converted to underscores in the saved map or globe name.

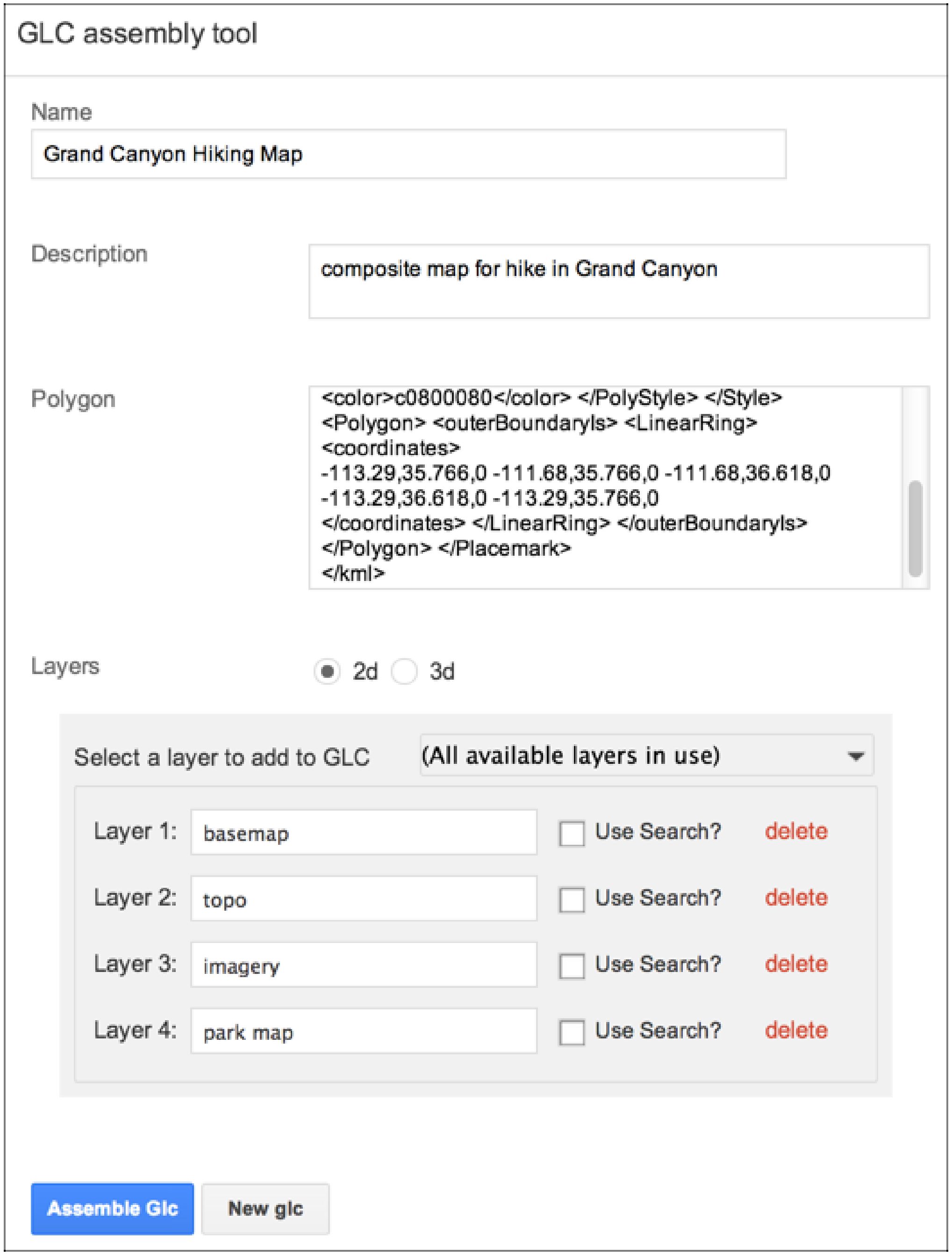
5. Enter a description to be associated with the composite file.

    The description appears in the list of portable files in Portable when you click the Folder icon and also on the Dashboard page in the GEE Server admin console.

6. Provide a KML polygon to define your ‘area of interest,’ including the definition of your localized search database.
7. Choose either 2D or 3D to see a list of the available layers from the maps and globes you have available in your /opt/google/gehttpd/htdocs/cutter/globes directory.
8. Click the **Layers** drop-down list to select a layer to add to your composite file. The selected layer appears as Layer 1. Check the **Use Search?** box to include a localized search database with this layer.

9. To add more layers, click the **Layers** drop-down list and continue to select from your available maps or globes.

   Select the layers in the order that you want them to appear in your portable globe or map. If you need to remove a layer, click **delete**.

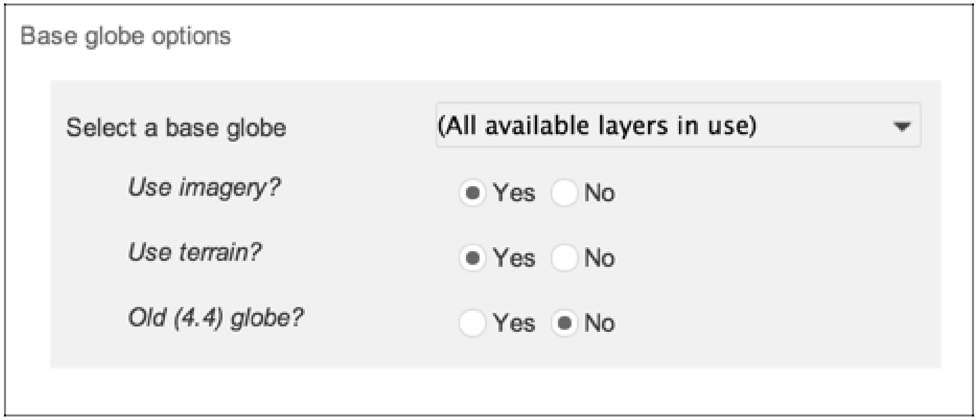


When you have no more layers available, the **Layers** drop-down list indicates that all available layers are in use.

For composite maps, once you have assembled the layers you want to include, click **Assemble Glc**.

For composite globes, you select a *base globe* and apply options to it.

10. To select a base globe option, click the **Base globe options** drop-down list and choose a portable globe, then click Yes to use the imagery and terrain data for the selected base globe.



11. If your base globe is a portable file created using Portable Server 4.4 or earlier, click **Yes** next to **Old (4.4) globe**.

    This option only affects the imagery layer within older `.glb` files.

12. Click **Assemble Glc**.

Extracting layers from a composite map or globe

If you want to extract the 2D or 3D layers from a composite `.glc` file, you use the **Glc Disassembly** tool, specifying a target directory for the extracted files.

To disassemble a composite file:

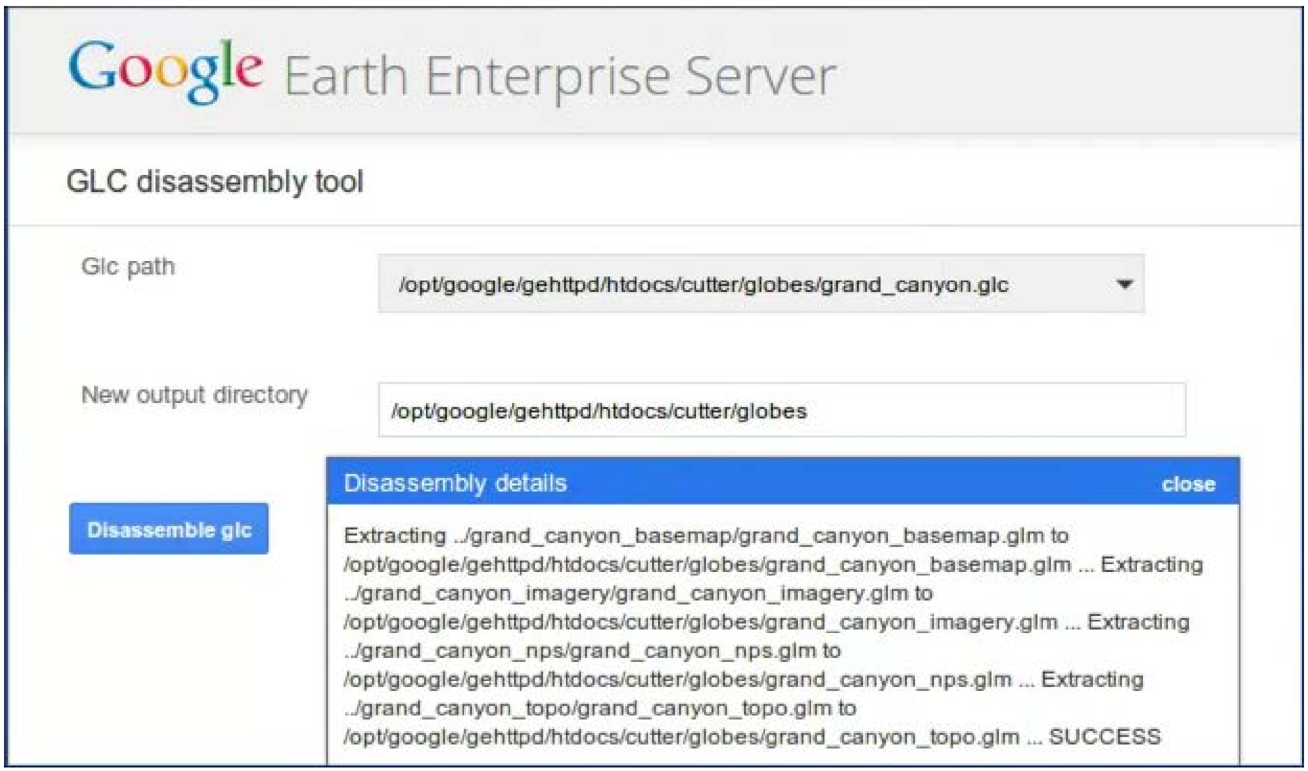
1. Access the Google Earth Enterprise Server Admin console in a browser window by going to myserver.mydomainname.com/admin, replacing myserver and mydomainname with your server and domain.
2. Sign in with the default credentials or the username and password assigned to you:
    - Default username: *geapacheuser*
    - Default password: *geeadmin*

Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator. Click Dashboard to view information about the portable files registered on and published to the Server.

3. Click the **Settings** button in the top right of the window and choose **Glc Disassembly** from the Tools menu. The GLC disassembly tool window appears.
4. Click open the Glc path drop-down list to select a composite file to disassemble.

   By default, the Glc path is the /opt/google/gehttpd/htdocs/cutter/globes directory, where all portable maps and globes are stored.
5. Enter a new output directory in the text box.
6. Click **Disassemble glc**.

   The layers in the composite file are extracted and are moved to the output directory that you specified. If you included a base globe, this portable file is also output to the same directory.



Back to top

View 3D databases

You can view your globes using Google Earth Enterprise Client (EC) or in a browser.

- View a globe in Google Earth EC
- View a globe using Google Earth Plugin

View a globe in Google Earth EC

Google Earth Enterprise Client (EC) lets you connect directly with your organization's private globes. You can sign in to as many databases as you want.

To view a globe in Google Earth EC:

1. Launch Google Earth EC. The Select Server dialog appears.
2. Specify the **Publish point** by entering or selecting the URL or IP address of your server and database in the Server field. For example, a Publish point of **BayAreaHighways** would be hosted at http://*myserver.mydomainname.com*/BayAreaHighways, where *myserver* and *mydomainname* are specific to your server.
3. Click **Sign In**.
4. If you have logged in to this server with Google Earth EC previously, log out, clear your cache, and log back in. For help with clearing your cache, see Clearing the Google Earth EC cache.
5. Google Earth EC displays your database. The Layers panel shows the terrain, imagery, and vector layers in the published database.

Learn more

- What is Google Earth EC?
- Use Google Earth EC features
- Clearing the Google Earth EC cache

Viewing 3D databases in a browser

You can view globes from Google Earth Enterprise directly using **Preview** or you can install Google Earth Plugin so that your 3D databases can be viewed from any browser. You can implement the Google Earth Plugin and also provide the installers so that users will be able to install it when they attempt to connect to GEE Server without the plugin loaded in their browser.

To view your map database from Google Earth Enterprise Server:

1. Log in to the Admin console of Google Earth Enterprise Server. The Admin console opens to the **Databases** page.
2. Check the box next to the 3D database that you want to view.
3. Click **Preview** or the **Publish point** of the database you selected. A new browser tab opens displaying your 3D database.

Learn more

- Implement Google Earth Plugin
- Provide Google Earth Plugin for your users to install

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Implement Google Earth Plugin

Included with Google Earth Enterprise is an example of how to implement the Google Earth Plugin in `/opt/google/gehttpd/htdocs/earth/earth_local.html`. By default, the virtual servers point to `earth_local.html` when accessing globes through a web browser.

The following HTML is the content of the `earth_local.html` file.

```
<html>
<head>
    <title>Google Earth Enterprise Earth Plug-in: Local Example</title>
    <link rel='stylesheet' type='text/css' href='/earth/earth.css' />

    <script type='text/javascript'>
      // To serve your static content from a different server, simply override
      // GEE_SERVER_URL to the URL of your GEE Server, e.g.:
      // "http://yourhost.com/default_ge/"       var GEE_SERVER_URL = "";
    </script>

    <!-- Load the required Javascript files. The loader for the Earth Plug-in API:
earth_plugin_loader.js. Utilities for this example UI: fusion_utils.js and
search_tabs.js The main routine: geeInit()
is found in fusion_earthplugin.js which defines the example UI and behaviors. -->
     <script type='text/javascript' src='/js/earth_plugin_loader.js'></script>
     <script type='text/javascript' src='/js/fusion_utils.js'></script>
     <script type='text/javascript' src='/js/search_tabs.js'></script>
     <script type='text/javascript' src='/js/fusion_earthplugin.js'></script>
</head>

<body onload='geeInit()' onresize='geeResizeDivs();'>

   <div id='header'>
     <div id='logo'> <img src="/earth/images/gee.gif" align="left" />
     </div>
     <div id='search_tabs'>
     </div>
  </div>
```

```
  <div style="clear: both;"></div>

    <table cellspacing="0" cellpadding="0" width="100%">
      <tr valign="top">
        <td id='left_panel_cell'>
           <div id='left_panel'></div> </td>
        <td>
          <div id='map'></div>
        </td>
      </tr>
    </table>
   </body>
 </html>
```

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Google

Provide Google Earth Plugin for your users to install

If your users want to connect to GEE Server and view your databases in a browser, they must install the Google Earth Plugin. If they don't already have the plugin, you can provide the installer so that the plugin downloads when they attempt to connect to GEE Server.

The plugin installer should be placed in the following directory on **GEE Server**:

```
/opt/google/gethttpd/htdocs/earth/plugin
```

The directory includes `install.html`, a landing page from which the installers are launched. When a user attempts to connect without the plugin installed, they are automatically directed here from earth_local.html's default error page. There are two installers, which should be renamed as follows:

- `/opt/google/gehttpd/htdocs/earth/plugin/GoogleEarth-Mac-Plugin-Latest.dmg`
- `/opt/google/gehttpd/htdocs/earth/plugin/GoogleEarth-Windows-Plugin-Latest.msi`

These installers do not require administrative rights on the user's machine. They also do not auto-update. If users are unable to download the installer from this location, they should contact their system administrator. If the user has internet access, they can download an auto-updating installer from http://www.google.com/earth/explore/products/plugin.html.

The Google Earth Plugin installer for Windows requires administrative rights when installing to Internet Explorer.

Back to top

users to install

View 2D databases

Clearing the Google Earth EC cache

Admin console pages

- Snippet profiles page
- Search tabs page
- Settings page
- Dashboard page
- Databases page

Google

View 2D databases

Once you publish a 2D database, you can view it from Google Earth Enterprise Server or in any browser. You can also interact with map layers you created in a 2D database using the `geeCreateFusionMap` class of Google Maps API v3 for GEE.

To view your map database from Google Earth Enterprise Server:

1. Log in to the Admin console of Google Earth Enterprise Server.

   The Admin console opens to the **Databases** page.

2. Check the box next to the map database that you want to view.
3. Click **Preview**.

   A new browser tab opens displaying your map database.

To view your map database in any browser:

1. Launch any web browser.
2. Point your browser to:

   http://myserver.mydomainname/publish_point

   where *myserver.mydomainname* is the host name or IP address of your server to which you published the map database, and *publish_point* is the publish point that you specified when the published the map database. For example: *http://my_host_name/mysanfranciscomap*

   If you are not sure which server you published to, contact your Google Earth Enterprise Server administrator for help.

3. The map database displays in your browser.

Using the Google Maps API

The Fusion Maps API is used to create and interact with map layers created in Google Earth Enterprise. The API is based on the Google Maps API but includes an additional `geeCreateFusionMap` class that makes it easier to interact with map layers generated by Google Earth Enterprise.

Learn more

- Google Maps API v3 for GEE
- Google Maps Javascript API V3 Reference

Back to top

Google Earth Enterprise Documentation Home | GEE Server

Clearing the Google Earth EC cache

If you have signed in to a database on GEE Server with Google Earth EC previously, you may be unable to sign in again to the same database. An effective way of resolving this issue is to sign out of your server, clear your cache, and log back in.

To clear your cache:

1. Select **File > Server Sign Out**.
2. Select **Google Earth EC > Preferences > Cache**.
3. Select **Delete Cache File** (only when logged out).

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Snippet profiles page

The **Snippet profiles** page of the GEE Server Admin console is where you manage Snippet profiles for use with 3D Fusion databases. Snippet profiles allow specific snippet needs to be managed and uniformly applied to all databases that require them. From this page, you can:

- Create new snippets:
  - Click **Create new**, add a name, and click **Create**.
  - Select one or more snippets.
  - Click **Save changes**.
- Edit snippet profiles: Click a snippet name and edit it, then click **Save changes.**
- Delete a snippet by clicking **Delete** next to a snippet name.

Learn more

- Manage snippet profiles

Back to top

Search tabs page

The **Search tabs** page of the GEE Server Admin console is where you can customize search services. From this page you can:

- View the default search tabs:
  - **POISearch**: Search from select vector and map layer fields.
  - **GeocodingFederated**: Search for places or coordinates.
  - **Places**: Search for places using the built-in locations database.
  - **Coordinate**: Search for latitude, longitude pairs. The following formats are supported:
    - Decimal Degrees (e.g. 39.507618° -84.168556°)
    - Degrees, Minutes, Seconds (e.g. 20°40'01.51" S 131°53'51.39" E)
    - Degrees, Decimal Minutes (e.g. 49° 32.876' N 110° 9.193' E)
    - Universal Transverse Mercator (e.g. 43 R 637072.95 m E 2825582.86 m N)
    - Military Grid Reference System (e.g. 36NTL8040632621)
- Create new search tabs: Click **Create New** and provide a search name, label, and parameters.
- Edit search tabs: Check the box next to a search tab and click **Edit**. Editing a default search tab creates a new search tab with your changes.
- Delete custom search tabs: Check the box next to a custom search tab and click **Delete**.

Learn more

- Create search tabs
- Specifying search fields for individual layers
- Configuring a searchable database

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Google

Settings page

Click **Settings** icon in the GEE Server Admin console to:

Perform GLC Assembly

Create new Glc globes from existing globes (Glm/Glb):

1. Enter a name and description for the new Glc.
2. Paste **Polygon** KML for use with the new globe.
3. Select either **2d** layers to build a Glc using selected layers of existing Glm files, or **3d** layers to build a Glc using Glb files at the layers.
4. Click **Assemble Glc**.

Glc files are added to the default globes directory when they are created.

Perform Glc disassembly

Extract Glm/Glb files:

1. Select a Glc to disassemble.
2. Enter a **New output directory** where the files will be placed.
3. Click **Disassemble Glc**.

Launch Cutter

Create a new Glm/Glb from published Fusion databases or Portable globes (Glm/Glb, not Glc):

1. Select a source map.

2. Enter a name and description for the offline map.
3. Choose to **Overwrite** existing globes with the same name.
4. Choose a **Region Selection** method:
    1. **Manual**: Manually draw the boundaries of your region using the polygon button in the top right of the map.
    2. **Paste KML**: Define your region by pasting a KML file.
5. Specify a **World level**; this represents the highest resolution at which the entire world will be available.
6. Optionally use the zoom buttons in the lower right of the map to change your zoom level. This represents the highest resolution at which the data within the region of interest will be available.
7. Click **Cut map**.

The cut progress will be displayed at the bottom of the screen. Upon successful completion, the globe is saved to the default globes directory.

View Apache logs

View Apache logs for troubleshooting errors or unexpected behavior with GEE Server.

Back to top

Dashboard page

The **Dashboard** page of the GEE Server Admin console provides an overview of GEE details, including:

- Number of Fusion databases
- Number of published Fusion databases
- Number of Portable databases
- Number of published Portable databases
- Number of search tabs
- Cutter status (on or off)
- Size of total Fusion data
- Size of Portable data
- Number of snippet profiles

Back to top

Google

Databases page

The **Databases** page of the GEE Server Admin console is where you manage Fusion databases and portables. From this page you can:

- View and sort your list of databases and portables. The list is populated by the items found in `/opt/google/gehttpd/htdocs/cutter/globes/`. Fusion databases must be pushed to this list from the Fusion interface. Portables appear in this list after they are registered using the **Manage Portables** button on the same page. Click the refresh button in the upper right corner to refresh the list.
- Manage databases and portables. Check the box next to a database, then click:
    - **Publish**: To publish a database. Specify a **Publish point**, where the database or portable will be accessible from. For example, if you specify *sanfrancisco*, it will be accessible from *myserver.mydomainname*.com/sanfrancisco.

      Note: When publishing a database, the publish point you specify is case *insensitive*. Upper and lower case are not differentiated. Make sure each publish point path name you specify is unique.

    - Choose a virtual host, and, depending on the type of item, the following options may also be available: Search Tabs, Snippet profiles, and Serve WMS.
    - **Unpublish**: To unpublish a database from its publish points.
    - **Preview**: To preview a published database.
    - **Remove**: To unpush an unpublished Fusion database, or unregister an unpublished portable. This removes the database or portable from the list, but doesn't delete them. Portables still appear in the master list of portables, which you can view by clicking **Manage Portables**.

- Register or unregister portables. Click the **Manage Portables** button, then click:
  - **Register**: To add a portable to the database list and make it available for publishing.
  - **Unregister**: To remove a portable from the list of databases. If the portable is published, it must be unpublished before being unregistered.
  - The name of a portable to download it.
  - The refresh button: To refresh the list of portables.
  - **Close**: To close the Manage Portables dialog.

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Portable User Guide for Mac OS, Windows, and Linux

Portable User Guide

Portable Developer Guide

Hello Earth! code sample

Hello Maps! code sample

Inspecting portable files

Contents

Introduction

Google Earth Enterprise Portable

Google Earth Enterprise Portable (GEE Portable) lets you view portable globes and maps on your laptop or desktop without requiring network access. This is useful for emergency responses to disasters like earthquakes or floods, or for maps that contain private information that you don’t want to share on the internet.

You can install GEE Portable on a Windows, Mac OS X, or Linux machine. It starts within seconds and automatically launches a web browser to display one of the portable globes or maps that's been saved to the GEE Portable `maps` directory.

Note: Instead of using GEE Portable, you can also connect to a portable globe or map with the Google Earth Enterprise Client (Google Earth EC) or the Google Earth Plug-in.

Portable globe and maps

A portable globe or map is a single file that stores all the geospatial data available within your specified area of interest — including all high-resolution imagery, terrain, vector data, KML files, and searchable point of interest (POI) locations.

Outside the specified area of interest, the globe or map stores only low-resolution imagery and terrain. You specify the levels of resolution when you cut the globe or map.

The following table describes the portable file types and their compatibility with GEE Portable.

Portable File Type	Description	Compatibility with Portable
.glb	Portable 3D globe.	All versions.
.glm	Portable 2D map.	All versions.
.glc	Composite map or globe file *assembled* from 2D or 3D layers of other portable files.	Versions 4.4 and higher.

Creating portable globes and maps

You can create portable globes and maps with the cutter tool feature of the Google Earth Enterprise (GEE) Server, or you can obtain them from third-party vendors. Depending on your area of coverage, it can take only a few minutes to specify and generate a globe or map and then save it to the GEE Portable `maps` directory.

Useful resources

- Creating portable globes and maps. Google Earth Enterprise users can learn how to cut globes or maps to serve from Portable.
- Portable Developer Guide. Software developers can create or customize applications for Portable.
- Google Enterprise Support Portal. To download the Google Earth Enterprise Portable installer, or for technical support 24 hours a day, 5 days a week in your time zone.

Install GEE Portable

GEE Portable is supported on:

- Windows XP and 7
- Windows 2008 Server
- Mac OS X 10.4 and later
- Red Hat Enterprise Linux versions 5.6 to 6.2, including the most recent security patches
- Ubuntu 10.04 LTS and 12.04 LTS

To install GEE Portable:

1. Download the GEE Portable Installer from the Google Enterprise Support Portal.
2. Install GEE Portable on your Windows, Mac OS X, or Linux machine. For details, see the appropriate section below.
3. Click the GEE Portable shortcut inside the `GEE Portable` folder to launch GEE Portable in a web browser.

Note: Before you install an upgrade, back up the contents of your `maps` directory.

Windows

For Windows 7, you can choose to install either **Per Machine** or **Per User** by selecting "Anyone who uses this computer (all users)" or "Only for me (username)". You need administration rights for **Per Machine** installation.

For Windows XP, your only option is **Per Machine**.

By default, the program files are typically installed to:

`C:\Program Files\Google\Google Earth Portable Server\Portable\`

The default path for the data files is typically:

`C:\Program Files\Google\Google Earth Portable Server\Portable\Data\`

The default paths might be different for different versions of Windows. You can specify your own paths when you install.

Mac OS X

1. Copy the `GEEPortableMacInstaller-5.0.dmg` file to the OS X machine.
2. Double-click the `.dmg` file, then open the image in Finder by double-clicking the file on your desktop.
3. Drag the `Google_Earth_Portable` folder into your `Applications` folder.
4. To launch the server, double-click the Google Earth Portable icon.

Linux

1. To install using the Ubuntu GUI, right-click and choose **Extract**. If you're using the command line instead, enter:

   ```
   tar -xvf GEEPortableLinux####Installer.tar.gz
   ```

   For *`GEEPortableLinux####Installer`*, substitute the exact name (including version number) of the Linux Installer you downloaded.

2. To launch the server, use the command line to enter:

   ```
   cd Google_Earth_Portable
    ./portable
   ```

   To stop the server, press `ctrl-c` from the command line.

3. To access the server, enter http://localhost:9335 in your web browser. If you've changed your hostname from 9335, use your new hostname.

Note: The Google Earth Plug-in isn't supported on Linux, so you can't view globes (`.glb` or `.glc` files) on your Linux browser. You can, however, view maps (`.glm` files) on Linux. You can also use your Linux machine to serve 3D globes that you can view from Mac or Windows machines that are connected to the Linux server.

Serve a globe or map from GEE Portable

Installing GEE Portable creates a `Google_Earth_Portable` directory on your machine. This directory contains a folder called `maps` (unless you renamed it in `portable.cfg`). Copy your globe or map to the `maps` folder. The GEE Portable interface lists all the globes and maps that you place in this folder. If you no longer want a globe or map to appear in the list, simply remove it from the `maps` folder.

Serve a globe or map from the Google Earth Enterprise Server

If you want to serve a globe or map to a large number of users, you can use a Google Earth Enterprise Server (GEE Server) on a Linux machine instead of buying a GEE Portable license for each user's machine. A GEE Server license allows 10,000 users. In contrast, each GEE Portable license allows either 1 or 10 users. Another reason to use a GEE Server is that it's capable of storing very large globes or maps. GEE Servers also let you serve globes and maps on your own private network so that only authorized users can connect.

To serve a map or globe from GEE Server:

1. Enable the GEE Server Cutter tool on the command line:
   - In GEE 5.0: `gecutter enable`
   - In earlier versions: `geserveradmin --enable_cutter`

   By default, the cut globes are stored in the `/opt/google/gehttpd/htdocs/cutter/globes` directory.

   To change the directory, create a symlink to point to another directory.
2. Access the Google Earth Enterprise Server Admin console in a browser window by going to `myserver.mydomainname.com/admin`, replacing *myserver* and *mydomainname* with your server and domain.
3. Sign in with the default credentials or the username and password assigned to you:
   - Default username: geapacheuser
   - Default password: geeadmin

   Note: If you do not know your username and password, contact your Google Earth Enterprise Server System Administrator.

   ---
4. Click **Manage Portable** to display the list of portable files in the `/opt/google/gehttpd/htdocs/cutter/globes` directory (by default).
5. Click **Register** next to the portable file you want to connect to. A message appears to indicate that your portable map or globe has been registered to GEE Server. Close the Manage portable globes window.

   If you want to download the file, click the file name.

The registered portable map or globe now appears in the Databases list of the GEE Server Admin console.

6. Check the box next to the portable file name, then click **Publish**. The Publish dialog appears.
7. Enter a Publish point or accept the default. For example, the Publish point **MyCutGlobe** would result in a serving URL *`myserver.mydomainname.com/`*`MyCutGlobe`, where *myserver* and *mydomainname* are specific to your server.
8. Specify a virtual host and optionally turn on WMS.
9. Click **Publish** to publish the portable file.

   A message appears to indicate that your portable map or globe has been published and the Publish point updates in the Databases list.
10. Click the Publish point link to view the portable map or globe in a new browser tab.

Broadcast a globe or map

To share a globe or map with others on your network:

1. When `disable_broadcasting` is set to `True` in `portable.cfg`, the default setting, you can enable broadcasting using either of the following methods:
   - `http://localhost:9335/?cmd=set_key&accept_all_requests=t`
   - Add `accept_all_requests True` to your `portable.cfg` file

Note: By default, broadcasting is off and cannot be turned on via an http call to the API. This feature is controlled by the `disable_broadcasting` flag, which is set to `True` in `portable.cfg`. However, if you set `accept_all_requests` to `True` in `portable.cfg`, then broadcasting is enabled, regardless of the `disable_broadcasting` state.

Connect using GEE Portable

For Windows, double-click the GEE Portable desktop shortcut created during installation. For Mac OS X, from the `Applications` folder, open the `Google_Earth_Portable` folder and double-click Google Earth Portable to launch it. After it launches, click the **Folder** (as shown outlined in red below), then select the globe or map you want to view. You can view only one globe or map at a time.

Globes and maps that are broadcast on your local network might require an access key. If prompted, enter the key to view the globe or map. You can obtain the key from the person who is broadcasting the globe or map.

Connect using the Google Earth Enterprise Client

Launch the Google Earth Enterprise Client (Google Earth EC). When prompted for a server address, enter `http://localhost:9335`. If you've changed the default port in `portable.cfg`, use the new port value instead.

Connect using the Google Earth Plug-in

GEE Portable comes with preconfigured HTML pages called `hello_earth.html` and `hello_maps.html` that display your globe or map using the Google Earth API or Google Map API.

If you'd like to make your own custom application, start by making a copy of either of these files and then add your own edits.

To access either of the files, enter the URL in your browser:

```
http://localhost:9335/local/preview/developers/hello_earth.html
http://localhost:9335/local/preview/developers/hello_maps.html
```

Note: Although it is possible to configure GEE Portable to display multiple globes or maps at the same time, this isn't recommended or supported. The additional configuration requires you to build the globe or map to reference a specific port number, which means you would have to rebuild it if you wanted to re-use it for any other ports.

Get info about your globe or map

Click the **Folder** (as shown outlined in red below) to view a list of the globes and maps you can access. Each globe or map is listed with its file name, description, creation date, and size.

Select a globe or map, then click the **Menu** (as shown outlined in red below) then select **Show layer list** to see all the layers for that globe or map. You can use the list to select the layers you want the globe or map to display. To hide the layer list, click the **Menu**, then select **Hide layer list**.

Change your configuration options

The Google Earth Portable directory contains the `portable.cfg` configuration file, either inside the `Tools` subdirectory (Windows) or the `.apps` subdirectory (OS X). For Linux, the configuration file is at `/gevol/portable/server/portable.cfg.`

The Portable configuration file defines the editable options listed below.

- **`port`**. The port on which to serve the globe or map. The default is 9335. If you change the port, you must re-cut your globes and maps with the new port number.
- **`globes_directory`**. The directory that contains the globe and map files.
- ***`map_name`***. The default globe or map to serve when GEE Portable launches.
- **`fill_missing_map_tiles`**. If set to `True`, enables pixel-filling from the ancestor map tile when there are no more tile descendants. Set this to `False` if you want to clearly indicate areas that are beyond their natural resolution, or if you just want to improve performance. You can also improve performance by lowering the value of the `max_missing_maps_tile_ancestor` option below.
- **`max_missing_maps_tile_ancestor`**. If the `fill_missing_map_tiles` option (above) is set to `True` and no tiles exist at your current display level, this option specifies the maximum number of tiles to create from samples of the ancestor tiles. Using a lower value can improve performance because the server creates fewer tiles. The value is written as 2^x by 2^x sized pixels. By default, $^x = 3$. (Or 23 by 23 pixels, which equals 8 x 8 pixels. This is Display Level 6, or 24 tiles). To lower the value, replace ^{x} with a number lower than 3.
- **`local_override`**. If set to `True`, GEE Portable looks for all the files on the server first before looking for them on your machine.
- **`disable_broadcasting`**. By default, this flag is set to True, preventing broadcasting from being turned on via an http call to the API. However, if you set `accept_all_requests` to `True` in `portable.cfg`, then broadcasting is enabled, regardless of the `disable_broadcasting` state.
- **`accept_all_requests`**. If set to True, GEE Portable accepts all requests to the server, and thus enables broadcasting, regardless of the state of the `disable_broadcasting` flag.

Back to top

Google

Portable User Guide

Portable Developer Guide

Hello Earth! code sample

Hello Maps! code sample

Inspecting portable files

Portable Developer Guide

Contents

Introduction

You can customize or extend GEE Portable, or create applications that work with it.

To help you get started, Google provides two `Hello World!` files that you can copy and use as templates. These files (`Hello_Maps.html` for 2D maps and `Hello_Earth.html` for 3D globes) call the required scripts, load the globe or map, define the `.css` file, and so on. The files include JSON and, if needed, KML polygons.

See the resources linked to below to learn how to edit and extend the `Hello_Maps.html` and `Hello_Earth.html` files.

Note: Although it is possible to configure GEE Portable to display multiple globes or maps at the same time, this isn't recommended or supported. The additional configuration requires you to build the globe or map to reference a specific port number, so you can no longer re-use it for any other ports.

Useful resources

Note: Clicking any of these links connects you to a site outside your network. If you're not sure if your organization allows outside connections, check with your administrator first.

`Hello_Maps.html` and `Hello_Earth.html` are based on the Google Maps API and Google Earth API. More information about these APIs is linked to below. Also included are links to some resources for learning how to parse your JSON data and KML coordinates.

- Google Maps API help at https://developers.google.com/maps/. Examples that use the Maps API.
- Google Earth API help at https://developers.google.com/earth/. Ways to extend the Earth Plug-in.
- jQuery help at http://api.jquery.com/jQuery.parseJSON/. A method for parsing JSON.
- geoxml3 at https://code.google.com/p/geoxml3/. A KML processor for use with the Google Maps API.

Hello Maps!

To write a script to display portable 2D maps, start with `Hello_Maps.html` and edit as needed. `Hello_Maps.html` is based on the Google Maps API.

To access the `Hello_Maps` file:

1. Launch GEE Portable on `localhost:9335`.
2. Enter `http://localhost:9335/local/preview/developers/hello_maps.html` in your browser.
3. View the page source. If you're not sure how to view the page source, see your browser's help.

Alternatively, click the link below:

Hello Maps! code sample

Hello Earth!

To write a script to display portable 3D maps (globes), start with `Hello_Earth.html` and edit as needed. `Hello_Earth.html` is based on the Google Earth API.

To access the file:

`Hello_Earth`

1. Launch GME Portable on `localhost:9335`.
2. Enter `http://localhost:9335/local/preview/developers/hello_earth.html` in your browser.
3. View the page source. If you're not sure how to view the page source, see your browser's help.

Alternatively, click the link below:

Hello Earth! code sample

JSON documents

Whenever your Portable instance is running, multiple JSON documents are serving. The JSON finds information about all your available globes and maps, as well as your current globe or map. Below are some code snippets that show how the JSON works.

Globes JSON example

This code snippet is from the Globes JSON document, which provides information about each available globe.

`Globes JSON: [yourhost]/?cmd=globes_info_json`

```
 [ {  "name":"test.glb",
  "timestamp":"2013-01-01 12:00:00",
  "size":"20.00MB",
  "description":"Some globe description.",
  "path":"../../../../globes/test.glb",
  "is_gee":true,
  "is_2d":false,
  "is_3d":true,
    "has_polygon":true,
 "is_mercator":false,
  "is_being_served":false
 },
 {
 some other globe
 },...
 ]
```

2D JSON and 3D JSON examples

The 2D and 3D JSON documents contain layer information as well as information that helps the Search feature categorize the results.

If you’re currently serving a map, 2D JSON will be serving:

```
2D JSON: [yourhost]/query?request=Json&vars=geeServerDefs&is2d=t
```

If you’re currently serving a globe, 3D JSON will be serving:

```
3D JSON: [yourhost]/query?request=Json&vars=geeServerDefs
```

Layer Definition examples

Below are examples of layer definitions. The first is an imagery layer and the second is a vector layer.

Imagery layer example

```
[
 ...{
  icon : "icons/1.png",
  id : 1001,
  initialState : true,
  isPng : false,
  label : "Imagery",
  lookAt : "none",
  opacity : 1,
  requestType : "ImageryMaps", // A layer of imagery. version : 8
 }...
 ]
```

Vector data example

```
[
 ...{
  icon : "icons/2.png",
  id : 1002,
  initialState : true,
  isPng : true,
  label : "Tokyo",
  lookAt : "none",
  opacity : 1,
  requestType : "VectorMapsRaster", // Vector data, such as roads, points, and borders.
 version : 4
 }...
 ]
```

Polygon KML document

Some globes also serve a KML file that contains polygon coordinates, which define the initial display when the globe or map loads. Polygon KML files also let Google Earth and Google Maps draw a polygon on the globe or map. If a globe doesn’t have any polygon data, the Polygon KML file is empty.

Polygon KML example

This is an example of a KML document that contains a set of coordinates that define the bounds of your globe's cut.

`Polygon KML: [your host]/earth/polygon.kml`

```
<?xml version="1.0"
encoding="UTF-8"?>
<kml>
<Document>
 <name>polygons</name>
<Placemark>
 <Polygon>
  <tessellate>1</tessellate>
  <outerBoundaryIs>
  <LinearRing>
  <coordinates>
-122.3185062675476,37.790043919799245,0
-122.3395950675476,37.84095511979925,0
-122.39050626754761,37.86204391979925,0
-122.44141746754761,37.84095511979925,0
-122.46250626754761,37.790043919799245,0
-122.44141746754761,37.739132719799244,0
-122.39050626754761,37.71804391979924,0
-122.3395950675476,37.739132719799244,0
-122.3185062675476,37.790043919799245,0
</coordinates>
  </LinearRing>
 </outerBoundaryIs>
 </Polygon>
 </Placemark>
<Placemark>
 <Polygon>
 <tessellate>1</tessellate>
 <outerBoundaryIs>
 <LinearRing>
 <coordinates>
-122.01254132080078,37.42221919299647,0
-122.03363012080078,37.473130392996474,0
-122.08454132080078,37.494219192996475,0
-122.13545252080078,37.473130392996474,0
-122.15654132080078,37.42221919299647,0
-122.13545252080078,37.37130799299647,0
-122.08454132080078,37.35021919299647,0
-122.03363012080078,37.37130799299647,0
-122.01254132080078,37.42221919299647,0
</coordinates>
 </LinearRing>
 </outerBoundaryIs>
 </Polygon>
</Placemark>
```

```
</Document>
</kml>
```

Back to top

Portable User Guide
Portable Developer Guide
Hello Earth! code sample
Hello Maps! code sample
Inspecting portable files

Hello Earth! code sample

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
 Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

 <html>

  <head>

   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
   <meta http-equiv="cache-control" content="max-age=0" />
   <meta http-equiv="cache-control" content="no-cache" />
   <meta http-equiv="expires" content="0" />
   <meta http-equiv="expires" content="Tue, 01 Jan 1980 1:00:00 GMT" />
   <meta http-equiv="pragma" content="no-cache" />
   <META HTTP-EQUIV="Pragma" CONTENT="no-cache">

    <!--This sample loads the necessary scripts for jQuery, Google Earth Plug-in,
 geeServerDefs, and the Maps API Fusion extension-->
    <title>Google Maps Engine Portable - Hello Earth Example</title>

     <!--Loads jQuery from the local source.-->
     <script src="/local/js/jquery-1.8.3.js"></script>

     <!--Loads the Google Earth Plug-in from the local source.-->
```

```
    <script src="/local/maps/api/earth_plugin_loader.js"></script>

    <!--Loads the server database definitions (geeServerDefs) from the local source. To
 view your server database definitions, navigate to http://localhost:9335/query?
request=Json-->
    <script type="text/javascript">

     $.ajax({
      url: "http://localhost:9335/query?request=Json&var=geeServerDefs",
      statusCode: {
       500: function() {
        document.getElementById('not_serving').style.display = 'block';
        document.getElementById('earth').style.display = 'none';
        document.getElementById('not_serving').innerHTML =
        'Error: To view this example, you need to run Google Maps Engine Portable\
        and serve a 3D globe on http://localhost:9335.'
       }
      }
     });

     function initEarth() {
       google.earth.createInstance(
        'earth', initCB, failureCB, {database: 'http://localhost:9335'});
     }

     // The Earth callback function sets the navigation controls for your 3D maps.
     // Unlike the Hello Maps code, Hello Earth requires the three functions below for
 initialization.
     // Some additional control options are listed in the comments below.
     // For even more options, see the Google Earth API Reference at:
 https://developers.google.com/earth/documentation/reference/.

     function initCB(earth) {
```

```
    ge = earth;
    ge.getWindow().setVisibility(true);
    ge.getNavigationControl().setVisibility(ge.VISIBILITY_AUTO);

    //Other visibility options for Controls
    //ge.getNavigationControl().setVisibility(ge.VISIBILITY_HIDE);
    //ge.getNavigationControl().setVisibility(ge.VISIBILITY_SHOW);

    //Set Controls in Bottom Right
    //ge.getNavigationControl().getScreenXY().setXUnits(ge.UNITS_INSET_PIXELS);
    //ge.getNavigationControl().getScreenXY().setYUnits(ge.UNITS_PIXELS);

    //Set Controls in Top Left
    //ge.getNavigationControl().getScreenXY().setXUnits(ge.UNITS_PIXELS);
    //ge.getNavigationControl().getScreenXY().setYUnits(ge.UNITS_INSET_PIXELS);

   }

   // Error handling in case the map doesn't load.
   function failureCB(earth) {
     alert('ALERT! The Google Earth Plug-in failed to load!');

   }

  </script>

 <style type="text/css">
 #head {
  margin: 0 auto;
  margin-top: 16px;
  height: 20px;
  width: 600px;
```

```
   font-family: "Arial", sans-serif;
   font-size: 16px;
   font-weight: normal;
  }
  #head span {
   background-color: #F1F1F1;
   color: #333;
  }
  #not_serving {
   background-color: #CB392A;
   color: #FFF;
   font-weight: bold;
  }
  #head span,
  #not_serving {
   display: block;
   padding: 16px;
   border: 1px solid #DCDCDC;
   font-size: 13px;
   margin: 8px 0 8px 0;
   box-shadow: 1px 1px 3px #F1F1F1;
  }
  #earth {
   margin: 0 auto;
   margin-top: 120px;
   width: 600px;
   height: 600px;
  }

  </style>

</head>
```

```
 <body onload='initEarth()' id='body'>

   <div id="head">Hello Earth! &#8212; Google Earth Plug-in &amp; Portable

     <span>
     To see how to initialize the Google Earth Plug-in and make a call to Portable, view
the source for this page. To view this example, you need to run Google Maps Engine
Portable and serve a 3D globe on http://localhost:9335.
     </span>

     <div id="not_serving" style="display:none;"></div>

   </div>

  <div id="earth"></div>

 </body>

</html>
```

Back to top

Hello Maps! code sample

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
 Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

 <html>

  <head>

    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="cache-control" content="max-age=0" />
    <meta http-equiv="cache-control" content="no-cache" />
    <meta http-equiv="expires" content="0" />
    <meta http-equiv="expires" content="Tue, 01 Jan 1980 1:00:00 GMT" />
    <meta http-equiv="pragma" content="no-cache" />
    <META HTTP-EQUIV="Pragma" CONTENT="no-cache">

     <!--This sample loads the necessary scripts for jQuery, Maps API, geeServerDefs,
 and the Maps API Fusion extension.-->

     <title>Google Maps Engine Portable - Hello Maps Example</title>

       <!--Loads jQuery from the local source.-->
        <script src="/local/js/jquery-1.8.3.js"></script>
```

```
        <!--Loads the Maps API from the local source.-->
        <script type='text/javascript' src='/local/maps/api/pbootstrap_loader.js'>
</script>

        <!--Loads the server database definitions (geeServerDefs) from the local
 source. To view your server database definitions, navigate
 to http://localhost:9335/query?request=Json&is2d=t-->
        <script src="http://localhost:9335/query?
request=Json&amp;var=geeServerDefs&amp;is2d=t"></script>

        <!--Loads the Maps API Fusion extension from the local source.-->
        <script type='text/javascript' src='/local/maps/api/fusion_extended_map.js'>
</script>

       <!--Error handling in case the map doesn't load.-->
     <script type="text/javascript">

      $.ajax({
       url: "http://localhost:9335/query?request=Json&var=geeServerDefs&is2d=t",
       statusCode: {
        500: function() {
         document.getElementById('not_serving').style.display = 'block';
         document.getElementById('not_serving').innerHTML =
         'Error: To view this example, you need to run Google Maps Engine Portable and
 serve a 2D globe on http://localhost:9335.'
       }
          }
      });

      function loadMap() {

 // Initializes a variable for your map and defines its settings.
 var mapOpts = {
```

```
                // Sets the default Zoom, Center, and other settings for the map's
initial  display.
                zoom: 2,
                center: new google.maps.LatLng(0, -22),
                navigationControl: false,
                mapTypeControl: false,
                streetViewControl: false,
                scaleControl: false
              };
                  geemap = geeCreateFusionMap('map', geeServerDefs, mapOpts);
     }

   </script>

   <style type="text/css">
   #head {
    margin: 0 auto;
    margin-top: 16px;
    height: 20px;
    width: 600px;
    font-family: "Arial", sans-serif;
    font-size: 16px;
    font-weight: normal;
   }
   #head span {
    background-color: #F1F1F1;
    color: #333;
   }
   #not_serving {
    background-color: #CB392A;
    color: #FFF;
```

```
    font-weight: bold;
   }
   #head span,
   #not_serving {
    display: block;
    padding: 16px;
    border: 1px solid #DCDCDC
    font-size: 13px;
    margin: 8px 0 8px 0;
    box-shadow: 1px 1px 3px #F1F1F1;
   }
   #map {
     margin: 0 auto;
     margin-top: 120px

     /*The 2D map won't load unless you set a height value. This code sample uses the
CSS, but you can use a div element instead. For example, <div id="map2d" style="border:
1px solid silver; height: 600px; width: 800px;"> </div>*/
      height: 600px;
      width: 600px;
   }
   </style>
</head>

<body onload='loadMap()' id='body'>

<div id="head">

Hello Maps! &#8212; Google Maps API &amp; Portable

<span>
To see how to initialize the Google Maps API and make a call to Portable, view the
source for this page. To view this example, you need to run Google Maps Engine Portable
```

```
and serve a 2D globe on http://localhost:9335.
</span>

<div id="not_serving" style="display:none;"></div>

</div>

<div id="map"></div>

</body>

</html>
```

Back to top

For the latest version of this documentation, go to the Google Earth Enterprise help center.

Google

Portable User Guide

Portable Developer Guide

Hello Earth! code sample

Hello Maps! code sample

Inspecting portable files

Inspecting portable files

The `geglxinfo` command provides different tools to analyze the composition of .glm, .glb, and .glc portable files. For example, you can determine the files included in a portable, extract those files, and test whether the portable file appears to be valid.

geglxinfo

```
geglxinfo [--glx glb_file_path]
```

Purpose

To inspect the .glc, .glb, .glm, or .glc files, the `geglxinfo` tool analyzes the composition and provides valuable information when you need to troubleshoot any issues that may arise when serving your portable files.

Examples

```
$ geglxinfo --glx NaturalView-US.glm --is_gee
IsGee: 1

$ geglxinfo --glx NaturalView-US.glm --check_crc

File crc: 0x7e6fbbfc
Calculated crc :0x7e6fbbfc
Good crc!

$ geglxinfo --glx NaturalView-US.glm --list_files
```

```
Index has 11 files.
0: earth/earth_local.html
  offset: 233330751
    size: 1731
1: earth/info.txt
  offset: 233332482
    size: 1849
2: earth/polygon.kml
  offset: 233334331
    size: 124947
3: icons/541_l.png
  offset: 233463282
    size: 363
4: icons/773_l.png
  offset: 233463645
    size: 226
5: icons/shield1_l.png
  offset: 233462733
    size: 549
6: mapdata/index
  offset: 226176927
    size: 7153824
7: mapdata/pbundle_0000
  offset: 0
    size: 226176927
8: maps/map.json
  offset: 233461561
    size: 1172
9: maps/map_v3.html
  offset: 233459278
    size: 2283
10: search_db/gepoi_14
  offset: 233463871
    size: 250
```

```
$ geglxinfo --glx NaturalView-US.glm --number_of_packets
298076 packets
```

Commands

`--is_gee`

Optional. Checks whether the glx file appears to be a valid globe or map. Returns a value of 1 is globe is valid; 0 if found to have errors. Use this validity check before testing your globe using `--crc`, especially on larger files, as it will catch almost all integrity issues.

`--glx` *`glb_file_path`*

The path and file name of the portable globe or map that you want to analyze.

`--list_files`

Optional. Lists all of the files in the glx.

`--id`

Optional. Unused.

`--check_crc`

Optional. Checks the crc of the glx.

`--extract_file` *`relative_file_path`*

Optional. File to be extracted from the glx. Use this option when you want to extract a .glm or .glb layer from one .glc in order to add it to a second .glc file.

`--extract_all_files`

Optional. Extract all files from the glx. Use this option when you want to extract .glm or .glb layers from one .glc in order to add them to a second .glc file.

`--number_of_packets`

Optional. Returns the number of data packets in the glx.

`--extract_packet` *`quadtree_address`*

Optional. Extracts a packet at a given quadtree address, for example, 310.

`--packet_type` *`type_string`*

Optional. Type of packet to extract: `dbroot, qtp, img, ter, or vec`.

`--packet_channel` *`channel_int`*

Optional. Channel of packet to extract.

`--output` *`dest_file_path`*

Optional. Destination file path where extracted file(s) should be written.

See Settings to learn about globe assembly and disassembly tools in the GEE Server Admin console.

Back to top

Google

Google Maps API v3 for GEE 5.0

Google Maps API v3 for GEE 5.0

Add Python libraries to GEE Python

For GEE release 4.4, see Google Maps API v3 for GEE 4.4.

Overview

The Google Maps Javascript API V3 is bundled with Google Earth Enterprise (GEE) Server and, for offline use, with Portable Server. As of June 2014 Google Maps JavaScript API v3.16 is bundled with GEE v5.0, but the API is updated periodically to align with the Release version.

The bundled offline Google Maps JavaScript API is the same core Javascript V3 API supporting:

- **Events**: Listening to click, mouseover
- **Controls**: Built-in zoom, pan, scale + custom controls
- **Overlays**: Symbols, markers, polylines

However there are some differences:

Method of invoking the API from a GEE or Portable Server to load in a web page.

- `geeCreateFusionMap` is a factory used to create instances of the `google.maps.Map` class.
- No support for objects that require access to online Google services:
    - Geocoding (`google.maps.Geocoder`)
    - Directions (`google.maps.Directions`)
    - StreetView (`google.maps.StreetViewPanorama`)
    - Traffic (`google.maps.TrafficLayer`)
    - Places API, Distance Matrix, Wikipedia, Panaramio, Weather layer, etc.

Using the Fusion Maps API

The Fusion Maps API is used to create and interact with map layers created in Google Earth Enterprise. The API is based on the Google Maps API but includes `geeCreateFusionMap`, which makes it easier to interact with map layers generated by Google Earth Enterprise. Essentially, `geeCreateFusionMap` is a factory used to create instances of the `google.maps.Map` class in which the GEE-specific configuration options can be added as Method extensions.

The following code snippet shows how `geeCreateFusionMap` is used to load a map:

```
<html>

  // Start by defining GEE_BASE_URL.
  <script type="text/javascript">
    var GEE_BASE_URL = window.location.protocol + '//' + window.location.host;
  </script>

  // Include the provided Maps API v3 files.  Located in
/opt/google/gehttpd/htdocs/maps/api/
  <script type="text/javascript" src="/maps/api/bootstrap.js"></script>
  <script type="text/javascript" src="/maps/api/fusion_extended_map.js"></script>
```

```
  // geeServerDefs must be defined with the following script.
  <script type="text/javascript" src="http://yourhost.com/YourPublishedDatabase/query?
request=Json&var=geeServerDefs"></script>

  <script type="text/javascript">
    // Using the geeServerDefs defined above, create a map instance.
    var geemap = new geeCreateFusionMap("map_canvas", geeServerDefs, myOptions);
  </script>

  </head>
  <body onload="initialize()">
    // Be sure there is an element on the page with the id you specified above
(map_canvas in this case).
    <div id="map_canvas" style="width:100%; height:100%"></div>
  </body>

</html>
```

Factory geeCreateFusionMap

Use of the factory `geeCreateFusionMap` is distinct from the standard Google Maps JavaScript API usage in the following ways:

- Instantiate `geeCreateFusionMap` in order to create a Fusion map. This is a factory for an extended version of the Maps API v3 `google.maps.Map` class, and the other methods are the extensions to that class.
- Use `geeCreateFusionMap` class instead of `google.maps.Map` to create applications that use layers from GME or GEE.
- `geeCreateFusionMap(container, opts?)` creates a new map inside the given HTML container, typically a `DIV` element. Options are the same as those of `google.maps.Map`
- If a map type is passed into the options, GEE Server will override these if an imagery layer is included in the Fusion Maps Database.
- The path and location of the new map is `serverURL/default_map`, for example, *http://my_host_name/default_map*.

Factory geeCreateFusionMap Description

The `Factory` and `Method extensions` are described in the following tables.

Factory

Factoryclass	Description
`geeCreateFusionMap(container, opts?)`	Creates a new Fusion map inside of the given HTML container, which is typically a `DIV` element. The options are the same as the options for `google.maps.Map`. However, if a map type is passed in to the options, the Fusion maps server will override these if an imagery layer is included in the Fusion Maps Database. After this constructor is invoked, the `setCenter()` method should be called before any methods that display Fusion layers on top of the map.

Method extensions

`geeCreateFusionMap` Method extensions are extensions of the Google Maps API v3 `google.maps.Map` class.

Method extensions	Return Value	Description
`showInitialFusionLayers()`	None	Show all layers that are enabled by default. This method should only be invoked after the `setCenter()`
`getFusionLayerCount()`	Number	Returns the number of Fusion map layers. This does not include the base imagery layer, which is built as a custom map type and serves as the background of the map.
`isFusionLayerVisible(index)`	Boolean	Returns true if the layer is currently shown on the map and false if it is hidden.
`showFusionLayer(index)`	None	Shows a previously hidden Fusion map layer. This method should only be invoked after the `setCenter()` method has been called to initially draw the map.
`hideFusionLayer(index)`	None	Hides the specified Fusion map layer.
`getFusionLayerName(index)`	String	Returns the name of the specified layer.
`getFusionLayerIcon(index)`	String	Returns the URL of the icon associated with the specified layer.

Examples

Several examples that illustrate how to use the Fusion Maps API are installed by default. You can find these examples in:

`/opt/google/gehttpd/htdocs/maps`

The following files are used by default for a published database. For example, when serving `http://your-host.com/YourPublishedDatabase`, this code is used to render the page.

- `maps_local.html`
- `maps_google.html`

The following files provide basic examples of initializing a Fusion Map. You can use them as templates but you'll need to edit them with the target path of a published 2D database to correctly initialize a map in the browser.

- `example_google.html`
- `example_local.html`

Back to top

Add Python Libraries to GEE Python

Python v2.7.5 is bundled with GEE, providing functionality such as publishing and custom search plug-ins.

The standard Python v.2.7.5 libraries are included with GEE. Additional modules specific to GEE are installed in:

`/opt/google/gepython/Python-2.7.5/lib/python2.7/site-packages`

- `psycopg2` is a PostgreSQL database adapter
- `google/protobuf` is Google's data storage mechanism
- `mgrs` provides conversion to and from MGRS
- `PIL` provides image processing and graphics capabilities, supporting many different file formats
- `GDAL` (Geospatial Data Abstraction Library) Python library

Add a Python library

You add a Python library to `/opt/google/gepython/Python-2.7.5` for it to be enabled with GEE. One of the easiest ways to install a Python library is to use the Python package tool, `pip`. You may also need to consider any other dependencies for your newly installed library.

The following example includes steps to test for and to install dependency libraries and to install the `lxml` library using pip.

Example: install lxml

`lxml` is a library for processing XML and HTML in the Python language. It has been developed as a new Python binding for `libxml2` and `libxslt`, completely independent from these existing Python bindings.

Requirements to install lxml

You need to install `libxml2` and `libxslt`, in particular: `libxml2 2.6.21` or later. It can be found here: http://xmlsoft.org/downloads.html.

- We recommend `libxml2 2.7.8` or a later version
- If you want to use `XPath`, do not use `libxml2 2.6.27`
- If you want to use the feed parser interface, especially when parsing from unicode strings, do not use `libxml2 2.7.4` through `2.7.6.`
- `libxslt 1.1.15` or later. It can be found here: http://xmlsoft.org/XSLT/downloads.html. We recommend libxslt 1.1.26 or later.

Newer versions generally contain fewer bugs and are therefore recommended. XML Schema support is also still being updated in `libxml2`, so newer versions will give you better compliance with the *W3C* spec.

Install libxml2, libxml2-dev, libxslt1-dev, and libxslt1.1

To check if `libxml2`, `libxml2-dev` are installed:

```
$ dpkg --list *xml2*
```

To install `libxml2-dev` and `libxml` if they are not already installed:

```
$ sudo apt-get install libxml2-dev
 $ sudo apt-get install libxml2
```

To check if `libxslt1.1`, `libxslt1-dev` are installed:

```
$ dpkg --list *xslt*
```

To install `libxslt1-dev`, and `libxslt1.1` if they are not already installed:

```
$ sudo apt-get install libxslt1-dev
 $ sudo apt-get install libxslt1.1
```

Install lxml

The best way to install `lxml` is to get the pip package management tool, `python-pip:`

To install python-pip, run the following command:

```
$ sudo apt-get install python-pip
```

To install lxml:

Run the following command, specifying the directory where the GEE Python version is installed:

```
$ sudo pip install --install-option="--prefix=/opt/google/gepython/Python-2.7.5" --
ignore-installed --verbose lxml==3.3.3
```

See the *pip install 1.5.4 Reference Guide* for information about install command options.

Back to top