Liquid Galaxy Manual

# Introduction

This manual will cover administration of Liquid Galaxy stand with Ubuntu machines.

The stand consist of:

* Minimum 2 Linux Machines with Liquid Galaxy configured
* Space Navigator Joystick
* Liquid Galaxy Controller



# Setting up Liquid Galaxy

Requirements

Installation system with command-ine install of Ubuntu (<https://help.ubuntu.com/community/Installation/LowMemorySystems>)

Or

Distributed squid proxy (link to page)

* User account name: lg
* Machine name: lgX (X is machine number, 1 is master)
* Configure LABELs for each of your filesystems so that the disk partitions will mount after cloning the disk to the other machines. Configure your /etc/fstab to mount by LABEL=\* instead of by UUID=\*.

If system is successfully installed and running, add these packages to allow installation of their dependencies:

sudo apt-get install xinput lxdm xorg rxvt-unicode fvwm fvwm-icons nitrogen graphicsmagick-imagemagick-compat librsvg2-bin subversion build-essential apache2 php5 php5-cgi xdotool squid3 squid3-cgi

## Network

We recommend using gigabit ethernet to connect the machines, but it isn’t really necessary, messages are small, and Google Earth’s bandwidth is limited to your upstream internet.

## Graphical Display

* Configure Xorg with the touchscreen as its "**CorePointer**".
* If you're using a podium with a [touch screen](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/TouchScreen.md), and your video card supports multi-headed displays, configure the second video card output for use with the touch screen.

The /etc/X11/xorg.conf configuration in the source control repository of this project is just one example and may need to be adjusted dramatically to fit your systems.

## Input device

(link to Setting SpaceNavigator)

In the source repository: the conf file, gnu\_linux/home/lg/etc/shell.conf and a helper script, earth/scripts/write-drivers-ini.sh work together at launch of Google Earth to configure it to use the specified device.

## Graphical Display

* Configure Xorg with the touchscreen as its "**CorePointer**".
* If you're using a podium with a [touch screen](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/TouchScreen.md), and your video card supports multi-headed displays, configure the second video card output for use with the touch screen.

The /etc/X11/xorg.conf configuration in the source control repository of this project is just one example and may need to be adjusted dramatically to fit your systems.

## User Account

* Create a user called "lg" and rsync home/lg from your git checkout to /home/lg.
* Rename the files and directories within the home/lg/dotfiles subdirectory to contain a "." (dot) as the first character and place them directly into the user's home directory. (e.g. "home/lg/dotfiles/fvwm2rc" becomes "/home/lg/.fvwm2rc")
* Play close attention to home/lg/dotfiles/xsession, as this is how the window manager is executed upon auto-login by the Display Manager.

## Google Earth

ViewSync functionality has been in [Google Earth](http://www.google.com/earth/index.html) since version 5.2. Be sure to download the most recent version and look over the license agreement when you download Google Earth if you have any questions about how you can use it.

* Install Google Earth into a directory called /home/lg/earth/builds/5.2.X.XXXX.
* Create a symlink in /home/lg/earth/builds directory:

$ cd /home/lg/earth/builds

$ ln -s ./5.2.X.XXXX latest

This will let our scripts find your Google Earth install.

The earth/config/master and earth/config/slave directories are used by run-earth-bin.sh to overwrite the files in .config/Google/ before starting Google Earth.

See the [Google Earth ViewSync Document](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/GoogleEarth_ViewSync.md) for more information on the specific workings of the ViewSync Feature in Google Earth.

## SSH Key setup

We use SSH extensively to communicate between galaxy nodes. The source repository contains a fairly dangerous [clean-ssh.sh](http://code.google.com/p/liquid-galaxy/source/browse/trunk/gnu_linux/home/lg/tools/clean-ssh.sh) file which can be executed with root privileges to initialize various SSH key items including:

* generating a keypair for the lg user.
* adding the generated keypair to authorized\_keys files for root and lg users.
* generating all-new host keys for sshd.
* adding the host key to known\_hosts to squelch warnings.

DO NOT use this script on a server or your own desktop system.  
DO use this script BEFORE cloning to make things easy on yourself.

## Machine cloning

Clone the disk of the machine you've completed setting up BEFORE you perform personality assignment in the next section.

Try using dd over netcat, or a g4u boot disk.

Note that udev likes to remember your network hardware's MAC address, so you may need to edit /etc/udev/rules.d/70-persistent-net.rules just before cloning, so that the new machines don't end up with, eg., eth1 instead of eth0.

## personality.sh

For Galaxy systems fitting the current MechanicalDesign, we've settled upon a clockwise numbering scheme beginning with the "**master**" system as **1**. Connect any input devices, such as a mouse or [multi-axis controller](https://github.com/liquidgalaxy/liquid-galaxy/wiki/LinuxSpaceNavigator.md), to "**master**" system.

Setting a personality sets the system hostname and prepares an alias interface ( eth0:0 ) via a new script dropped into /etc/network/if-up.d. Scripts in that directory are executed whenever an interface is brought up.

Run this on each machine, where "X" is the screen number (clockwise from the master):

$sudo /home/lg/bin/personality.sh X

# Setting up SpaceNavigator

## Make sure your multi-axis device is readable

First of all, make sure you **don't** have 3Dconnexion's drivers installed. The Linux kernel drivers work out of the box, and are what you need for Google Earth.

Recent linux kernels will automatically detect your multi-axis device and create a /dev/input/eventNdevice file, for some value of N. To avoid having to figure out which file corresponds to your device, create a udev rule that instructs the system to automatically create a /dev/input/spacenavigator file whenever your multi-axis device is plugged in.

As root, create a new file called "/etc/udev/rules.d/90-spacenavigator.rules" (for example, by typing "sudo gedit /etc/udev/rules.d/90-spacenavigator.rules"), containing this single line of text:

KERNEL=="event[0-9]\*", ATTRS{idVendor}=="046d", ATTRS{idProduct}=="c62[68]", MODE="0664", GROUP="plugdev", SYMLINK+="input/spacenavigator"

Unplug your multi-axis device and plug it back in, and see if a file called "/dev/input/spacenavigator" appears (for example, by typing "ls -l /dev/input/spacenavigator").

If not, you might want to reboot to make sure your udev rule gets read. If it still doesn't work, run "lsusb" and see if your multi-axis device is listed with "ID 046d:c626" (or 046d:c628 for some variants of the SpaceNavigator), matching the idVendor and idProduct fields in the 90-spacenavigator.rules file. If they don't match, you might have to change the idVendor or idProductfield in "/etc/udev/rules.d/90-spacenavigator.rules" to match. You might also try changing 'ATTRS' to 'SYSFS' in both places in the rule.

## (Optionally) disable the device as a mouse

Recent versions of xorg automatically detect multi-axis devices and try to use them as mice. The xinput command lets us turn that behavior on and off.

First, install xinput if it's not already installed, using your package manager. (For example, "sudo apt-get install xinput").

If you're using a SpaceNavigator, this should make it stop behaving like a mouse until you logout or until you unplug it:

xinput set-int-prop "3Dconnexion SpaceNavigator" "Device Enabled" 8 0

And this should turn the mouse behavior back on:

xinput set-int-prop "3Dconnexion SpaceNavigator" "Device Enabled" 8 1

If you're using some other multi-axis device, try running "xinput list" to find the name of your device.

## Enable multi-axis support in Google Earth

First we'll enable support for the SpaceNavigator by adding some settings to Google Earth's drivers.ini file. Make sure you exit Google Earth before editing your drivers.ini.

In Linux, the default location is /opt/google-earth if you installed as root, or $HOME/google-earth if you installed as yourself.

Open up the drivers.ini using your favorite text editor (for example, "gedit $HOME/google-earth/drivers.ini" or "sudo gedit /opt/google-earth/drivers.ini"), and add the following lines to the first stanza marked "SETTINGS" (between the { and }).

; Settings for multi-axis controllers

SpaceNavigator/sensitivityX = 0.125

SpaceNavigator/sensitivityY = 0.125

SpaceNavigator/sensitivityZ = 0.030

SpaceNavigator/sensitivityPitch = 0.01

SpaceNavigator/sensitivityYaw = 0.004

SpaceNavigator/sensitivityRoll = 0.007

SpaceNavigator/device = /dev/input/spacenavigator

SpaceNavigator/zeroX = 0.0

SpaceNavigator/zeroY = 0.0

SpaceNavigator/zeroZ = 0.0

SpaceNavigator/zeroPitch = 0.0

SpaceNavigator/zeroYaw = 0.0

SpaceNavigator/zeroRoll = 0.0

SpaceNavigator/gutterValue = 0.1

Close Google Earth and start it up again, and see if it works!

## Fine-tune the settings

Six-axis devices like the SpaceNavigator let you move forward, sideways or up and down (by shifting the puck without tilting), or rotate about each axis to create pitch, yaw and roll (by tilting or twisting the puck without shifting it).

The "sensitivity" parameters let you control how sharply Google Earth will react to your movements in each of those dimensions.

The "zero" parameters let you adjust for a device that doesn't quite rest in the center of its scale. You shouldn't normally need to change them.

The "gutter" parameter, however, makes a big difference in how responsive the device feels, by creating a "dead zone" near the center of travel. A larger dead zone means you'll have to move the device farther before Google Earth starts to respond. If you make the gutter value too low, you may find that Google Earth moves even when you aren't touching the device, due to slight variations in the device's position sensors.

# TouchScreen

Some of our Liquid Galaxy installations use an [Acer T230H](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/AcerT230H.md) touch screen to display a menu of places to go and an on-screen keyboard for searching:

In our [git repository](http://code.google.com/p/liquid-galaxy/source/checkout), you'll find a [php-interface directory](http://code.google.com/p/liquid-galaxy/source/browse/#svn/trunk/php-interface), with a few simple PHP scripts and HTML pages. These live on the "master" computer, and create [query.txt](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/QueryTxt.md) files that tell Google Earth where to go. Make sure you read up on the SecurityConsiderations, since anybody who can access the HTML and PHP will be able to use the interface.

The computer driving the touch screen (on the second display connector of the graphics card - see our [xorg.conf](http://code.google.com/p/liquid-galaxy/source/browse/trunk/gnu_linux/etc/X11/xorg.conf)) then displays a web browser (we use [chrome](http://www.google.com/chrome)) in full-screen mode.

The astute reader might think, "Hey! Google Earth and all of the helper scripts are running as the lguser, how do you handle permissions issues for the [QueryTxt](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/QueryTxt.md) file?". Well, both the www-data and lgusers can read and execute items within the /lg directory. So we've placed a small setuid-root binary ([chown\_tmp\_query.c source](http://code.google.com/p/liquid-galaxy/source/browse/trunk/gnu_linux/home/lg/bin/chown_tmp_query.c)) within /lg which sets ownership and permissions of the query file written by the PHP interface before renaming the query file to be consumed by Google Earth.

We mount the touch screen on a podium. Here's a [sketchup model of a podium we built from cabinet plywood](http://code.google.com/p/liquid-galaxy/downloads/detail?name=galaxy-podium-parts.skp&can=2&q=) and painted black:



# **MechanicalDesign**

Note that our frames are designed specifically to fit the 55" Samsung UN55B6000 LCD screens we use. You may need to modify the frame design if you plan to use something else.

In the [Downloads](http://code.google.com/p/liquid-galaxy/downloads/list), you'll find:

* *Google Liquid Galaxy 3-1 Mechanicals-External Panels.rar*: CAD files and drawings of the external panels that surround the Liquid Galaxy
* *Liquid Galaxy 3-1 External Panel Image 1 to 4.rar*and *Liquid Galaxy 3-1 External Panel Images 4 to 8.rar*: the images we used for the external panels that surround the Liquid Galaxy
* *Google Liquid Galaxy Mechanical Specification.pdf*: a summary of the critical mechanical components you need to build a Liquid Galaxy
* *Google Liquid Galaxy 3-1 Mechanicals-Frames.rar*: all the 3D CAD files (in STP file format) for the Liquid Galaxy frames
* *Google Liquid Galaxy 3-1 Mechanical Installation Guide.pdf*: step-by-step instructions for how to assemble a set of Liquid Galaxy frames

## Building your own Google custom frame

### 1. Find a manufacturer

You will need to find two different vendors:

1. A metal manufacturer who can fabricate the metal frames
2. A signage vendor who can print the external panels.

Make sure the signage vendor has the capability of cutting a material called Dibond using a [CNC process](http://en.wikipedia.org/wiki/Numerical_control/).

### 2. Get a quote and fabricate

Once you've selected a manufacturer, you have to get a quote from them that indicates an estimated turnaround time, and an estimated unit cost.

You can start the quoting process by sending the manufacturer a quote request form with CAD files and drawings. You can find these files in the archive *Google Liquid Galaxy 3-1 Mechanicals-Frames.rar*in the [Downloads](http://code.google.com/p/liquid-galaxy/downloads/list) section.

This folder also includes a Bill of Materials of all the hardware (such as bolts, nuts, etc.) needed to complete the assembly. The BOM can also be passed onto the manufacturer to have them purchase the parts for you and use them to fully assemble each frame segment.

Once you get a quote from the manufacturer, you should review it and make any revisions or negotiate the unit cost if necessary. After finalizing the quote and issuing a purchase order, all you will need to do is to wait for the finished frames. You may want to check in between to see if the quality of the frames are up to your standards. You may want to request a midpoint review to examine the frames in person if the manufacturer is local.

The same process applies for the external signage. To get a quote, send the vendor the panel images and the CAD files with a quote request form found in the archive *Google Liquid Galaxy 3-1 Mechanicals-External Panels.rar*, *Liquid Galaxy 3-1 External Panel Image 1 to 4.rar* and *Liquid Galaxy 3-1 External Panel Images 4 to 8.rar*.

### 3. Source and purchase other components

The list of hardware required is listed in the document *Google Liquid Galaxy Mechanical Specifications.pdf*.

### 4. Assemble

In the document *Google Liquid Galaxy 3-1 Mechanical Installation Guide*, you will find detailed step-by-step instructions of how to put the frames, panels and TVs together.

## Alternatives to building custom frames

You can use the 3D CAD files provided to modify the design or use them as a reference for measurements to design your own Liquid Galaxy from scratch.

Other options would be to mount the TVs onto a circular wall or use floor TV mounting stands. There are many options available online:

* [Luxor Universal Plasma Floor Mount](http://www.bhphotovideo.com/c/product/461507-REG/Luxor_LUPMF44_Universal_Plasma_Floor_Mount_.html/)
* [LDDFS68 Universal Rotating Mount](http://www.luxorfurn.com/education/lddfs68.php/)
* [Peerless Industries FPZ-600 TV Mount](http://www.homefurnitureshowroom.com/tv-mounts/peerless-industries-fpz-600-flat-panel-display-lcd-plasma-screens-tv-mount-white_g520091.html/)

# Introduction to Liquid Galaxy Controller

## User features:

User has ability to switch planet (Earth, Moon and Mars), choose between different places and pick tours.

## Administrative features:

Admin can add categories, POIs, Tours, Stand managing tools, LG tasks and Scan beacon.

### Categories and POIs:

Admin has ability to add POI (Point of Intrest) to Liquid Galaxy stand, when user chooses POI on Controller, Liquid Galaxy “flies” to given position and angle.

Example POI file ‘link to POIset file’

### Tours:

Tours are basically presentation of POIs with interval for each.

You can add a tour and edit existing.

### Tools:

There are 5 tool options

* Import POIS – Import POIs from file or Physical web beacon
* Reboot LG – Reboots system
* Shut down LG – Shuts down system
* Relaunch LG – Relaunches/refreshes system
* Clean KMLS – Cleans KML files from your Liquid Galaxy, KML file is file readable by Google Earth that contains information of placemark

### Scan beacon

Scans for physical beacons.

# Liquid Galaxy Controller configuration

In Administration panel, in upper right corner, there are menu with bunch of configuration options.

## Help

Help option contains a lot useful information about app and features.

## Settings

Settings contain important configuration options, used to connect to your Liquid Galaxy.

### Liquid Galaxy connection User

lg

### Liquid Galaxy connection Password

Password to your Liquid Galaxy that was set while installation.

### Liquid Galaxy connection Port

Port to your Liquid Galaxy that was set while installation, can be found in

ifconfig

as X in 10.42.X.1

### Administration Password

lg

## Reset database

This option deletes all the data inside the application.

## Export Database to local storage

Exports application database to device storgate.

## Log out

Logs out from Administration tools

# QueryTxt

QueryTxt file is used for controlling Liquid Galaxy, it is sent by controller to master in order to do following things:

* Search
* Switching Planets
* Fly to view
* Controlling Tours

### Search

If contents of the file are of the form "search=some search string", it performs a search on "some search string" just as if you had typed it into the Search box in Google Earth, then flies to nearest point/result

### Switching Planets

Switching planets are simple, content of file should be of form planet=planetname (Earth, Moon or Mars).

The contents “search=lat,long still work as expected for that planet.

### FlyTo

To send view to lat, long (place) the KML for a LookAt can be used with “flyview=”

Example with San Francisco

flytoview=<LookAt><longitude>-122.4017881321627</longitude><latitude>37.79152911640639</latitude><altitude>0</altitude><heading>167.0211046386626</heading><tilt>68.68179673613697</tilt><range>774.4323347622752</range><altitudeMode>relativeToGround</altitudeMode><gx:altitudeMode>relativeToSeaFloor</gx:altitudeMode></LookAt>

Just as one of POIs from Introduction to Liquid Galaxy Controller.

### Controlling Tours

Tours can be started and stopeed with “playtour=” and “exittour=”

playtour= expects the name of a tour, i.e. the contents of the "name" element of a "gx:Tour" in a currently loaded kml:

playtour=My World Tour

Where the tour kml looks like:

...

<gx:Tour>

<name>My World Tour</name>

...

exittour= accepts "true", and will always stop a currently running tour:

exittour=true

# DistributedSquid

## Why Cache?

Since Liquid Galaxy involves sending multiple computers to the same place at the same time, it's handy to have a web cache to reduce network load.

We use a distributed cache, in which each of the Galaxy machines also runs [squid](http://www.squid-cache.org/). They're set up to peer with each other, so our 8-machine Galaxy also ends up with an 8-machine squid cluster.

There are several ways to do it. Here's how we did it on Ubuntu GNU/Linux machines.

We provide links to the individual configuration files you'll need, but you may find it easier to just [check out everything](http://code.google.com/p/liquid-galaxy/source/checkout) using git.

## Make sure you're not running a web server on port 80

In the next section, you'll see how we run squid on port 80 as a simple hack to avoid setting firewall rules.

If you have a web server running on your machine, squid won't be able to bind to port 80. We moved apache over to port 81. You can see how we did that in [the gnu\_linux/etc/apache2 directory in our source repository](http://code.google.com/p/liquid-galaxy/source/browse/trunk/gnu_linux/etc/#etc/apache2).

Make sure you restart apache after changing the config files. sudo netstat -nlp |head can be a handy way to double check whether anything is listening on port 80.

## Set up squid to listen on port 80

Google Earth gets almost all its data from kh.google.com. So rather than doing transparent redirects using a firewall, we just pretend that localhost is kh.google.com, and tell squid to act as a web accelerator on port 80.

In git you can find our [/etc/squid3/squid.conf](http://code.google.com/p/liquid-galaxy/source/browse/trunk/gnu_linux/etc/squid/squid.conf) and [/etc/squid3/cachemgr.conf](http://code.google.com/p/liquid-galaxy/source/browse/trunk/gnu_linux/etc/squid/cachemgr.conf).

They set up squid to listen on port 80 and forward all requests to kh.google.com.

## Put kh.google.com in /etc/hosts

Next we set up [/etc/hosts](http://code.google.com/p/liquid-galaxy/source/browse/trunk/gnu_linux/etc/hosts) so that kh.google.com resolves to localhost.

But since squid needs to connect to the real kh.google.com, we also create [/etc/hosts.squid](http://code.google.com/p/liquid-galaxy/source/browse/trunk/gnu_linux/etc/hosts.squid), and set up the squid.conf to use that file instead of /etc/hosts.

Note that our /etc/hosts and hosts.squid also include entries for lg1 through lg8, the other machines in the galaxy. As you can read in the LiquidGalaxyHOWTO, we set up a second IP address for each machine with a static, private IP address so that the machines can always reach each other. Squid uses those aliases in its peer configuration, so make sure you can ping lg1 ... ping lg8.

## Try it out!

Restart squid with /etc/init.d/squid3 restart, then restart Earth on all the machines.

If you get an error about not being able to reach kh.google.com, you know you have a problem. Try using a web browser on one of the machines to reach [http://kh.google.com](http://kh.google.com/). If you get a Google logo and a "not found" error, you know you can reach kh.google.com successfully.

If you have the Squid CGI scripts installed (squid3-cgi), you can visit the cache manager interface on <http://localhost:81/cgi-bin/cachemgr3.cgi> (did you remember to move Apache to port 81?). Click "Continue...", then check out the "peer cache statistics". You should see each of the peers listed, as well as kh.google.com, and see the cache hit rate for each machines.

Don't expect to see huge hit rates for each peer -- if they're set up right as a distributed cache, then each one will only have one slice of the files you need. So for an 8-machine galaxy, you'd only expect about 12% cache hit rate for each host (and 0% for the machine you're currently looking at, since it doesn't need to ask itself for files it already has).

## Disable request batching

Once you have your cache working, you should probably disable request batching. Google Earth sends requests for multiple chunks of data at a time, improving performance by reducing the effects of network latency.

But squid has no way of separating those chunks out, so Earth only benefits when it happens to request the same batch of chunks.

However, Earth 5.2 includes an option in drivers.ini to disable request batching. This reduces Earth's performance if it's used without a cache, but can improve performance if you are.

Edit drivers.ini on each machine (or earth/config/drivers\_template.ini if you're using write-drivers-ini.sh as described in the LiquidGalaxyHOWTO), adding the following line to the SETTINGS { } stanza:

Connection/disableRequestBatching = true

Restart Earth, and fly around a bit, including somewhere you haven't been before, and verify that the change worked by examining the URLs Earth requests from kh.google.com. One way to do that is to look at /var/log/squid3/access.log.

If you see URLs like these:

http://kh.google.com/flatfile?q2-031211201120-q.324

http://kh.google.com/flatfile?f1c-02003332-d.3002.323

Then it's working. If you see multiple long numbers in URLs, then it's probably not working and you should double-check your drivers.ini.

## Keep an eye on performance

8 machines running Google Earth can generate a lot of network requests. We found that even when using squid on all 8 machines, sometimes Google Earth would really bog down. In our case, it turned out that squid was "spindle-bound" -- the hard drives we were using couldn't seek fast enough to keep up with the requests coming from Earth.

The "General runtime information" page in Squid's cachemgr3.cgi is useful for watching squid's performance; in particular, keep an eye on the "Median Service Times" for "Cache Hits"; if cache hits take longer than cache misses, your cache is actually slowing you down!

We found that solid state (SSD) hard drives work great due to their random access performance. You may also want to use the "relatime" or even "noatime,nodiratime" mount options to minimize disk activity.

## Common problems

### Out of memory

If you find squid crashing after a few days or weeks, it may be running out of memory. 32-bit OSes can only give about 3GB of RAM to any individual process, and squid needs several percent of the size of its disk cache in RAM to hold the cache index. This means you may only be able to use, say, 30GB of disk for squid if you're running it on a 32-bit machine.

### Out of inodes

Google Earth tends to request small chunks of data, especially if you've disabled request batching. If you're using ubuntu's default ext3/ext4 filesystem and squid is storing its cache as files on your hard drive (/var/spool/squid3 by default), you may run out of inodes if you use a very large cache.

The simple thing to do in that case is to clear out the existing cache and reduce the cache size. But you can also find other options in the squid docs, like using a disk partition directly.

If you have the option of formatting a new disk partition, you can allocate more inodes to support the large number of small files squid will create. If you do that while installing Ubuntu, for instance, you can set the "Typical usage" to "news".

# **SecurityConsiderations**

# Although we've been careful to implement the Liquid Galaxy features as correctly and safely as we can, they are still experimental, and not entirely bug free.

If you choose to turn them on (as described in the [QuickStart](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/QuickStart.md)), you'll need to control network access to your computers if you want to make sure that other people on the network don't send you unwanted view sync messages.

If you're just using Google Earth normally, without using the Liquid Galaxy features described on this site, don't worry. All the features we talk about here are turned off by default.

## View Synchronization and the monotonic counter bug

Liquid Galaxy works by sending view synchronization messages over the network. The messages are sent as UDP datagrams that describe where you are in Google Earth and where you're looking.

In Google Earth 5.2, if you enable ViewSync/receive as we describe in the [QuickStart](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/QuickStart.md), Google Earth will accept these messages from any source, so it's important to use a firewall or private network to limit who you can receive view sync messages from. (We're trying to improve that behavior in future versions of Google Earth).

In particular, there's a message ID field in the view sync messages that's currently used to reject old messages that might arrive out of order. So someone who can reach the UDP port you set in ViewSync/port could send a message with a very high message ID and cause Google Earth to reject future legitimate view sync messages with lower IDs until you re-launch Google Earth. (Sorry about that.)

Until then, you can work around this bug by limiting who can reach your UDP port with a firewall. And of course, if you don't need to worry if you're just using Google Earth normally (that is, with ViewSync/receive turned off or omitted from your drivers.ini).

## query.txt

Likewise, if you plan to control Google Earth using a [TouchScreen](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/TouchScreen.md) via the [query.txt](https://github.com/LiquidGalaxy/liquid-galaxy/wiki/QueryTxt.md) interface, you'll want to control who can reach the web pages that make up the user interface for the touch screen.

A properly configured firewall can help, as can techniques like setting the web server to only respond to requests from the same computer it's serving from. Or, you could set up a [.htpasswd](http://httpd.apache.org/docs/2.0/programs/htpasswd.html) file to require people to login before they can reach the touch screen controls.