# Logging Liquid Galaxy installation process in Ubuntu 16.04 and 18.04

It is known by the community that Liquid Galaxy has important issues running Google Earth Pro in the latest Ubuntu version: 18.04 or Bionic. It's because of this problems that it's recommended to install the project on Ubuntu 16.04 (Xenial) as said on Liquid Galaxy Lab's Github repo for Liquid Galaxy Core. In this task I've investigated about logging a bash script so I could optput the installation process in both versions to find the possible issues that affect Ubuntu Bionic.

## Logging Method

The code that I've used to log the whole installation process is this:

Listing 1: Logging Script at install.sh

```
#Create log file at Desktop
touch /home/lg/Desktop/installationXX.log
#Write stdout from terminal to .log
exec 1> >(tee -a /home/lg/Desktop/
  ↪ installationXX.log) 2>&1
#Installation script install.sh after this
```

The code above was set on the install.sh bash script provided by Liquid Galaxy Lab's Github repo for Liquid Galaxy Core and it creates a file on Desktop named ”installationXX.log”; being XX Ubuntu’s version: 16 for Xenial and 18 for Bionic. Touch command is in charge of the file’s creation while the second code line takes stdout and stderr and logs them in the file created in the touch command.

### Before checking for errors

Once I got the log file for each version I could compare them to see if something was wrong, and indeed I found some interesting things. Before you continue reading make sure you check the logs:

installation16.log – Ubuntu Xenial (16.04) on Google Drive.

installation18.log – Ubuntu Bionic (18.04) on Google Drive

I also reccomend to open them with any IDE like Atom as I’ll give the errors with its respective line number.

The two installations had the same configuration:

- MASTER: true
- LOCAL_USER: lg
- MACHINE_ID: 1
- MACHINE_NAME: lg1
- TOTAL_MACHINES: 2
- OCTET (UNIQUE NUMBER): 42
- INSTALL_DRIVERS: true
- GIT_URL: https://github.com/LiquidGalaxyLAB/liquid-galaxy
- GIT_FOLDER: liquid-galaxy
- EARTH_DEB: $http : //dl.google.com/dl/earth/client/current/google-earth - stable\_current\_amd64.deb$
- EARTH_FOLDER: /opt/google/earth/pro/
- NETWORK_INTERFACE: enp0s3

### Installation errors

The first error message on the logs is at line 838 and it seems to be a tee command error:

```
838. tee: /etc/lightdm/lightdm.conf: No such
   ↪ file or directory
```

Tee command as explained in **Logging Method** section takes an stdout from a program and writes it on a file. In this case the file it was trying to write on didn't exist at that moment. The file **/etc/lightdm/lightdm.conf** is pretty important for a correct Liquid Galaxy run, as it contains the LightDM package configuration with parameters that enable autologin. Note that this error is only found in the installation18.log and it is missing on installation16.log.

A possible solution to this issue could be installing the lightdm package before executing the install.sh bash script issuing the following command:

```
~$ sudo apt install lightdm
```

The second error is found in the very next line: 839. The error looks like this:

```
839. No such schema org.compiz.unityshell
```

The error refers to a missing Unity package dependencies, in this case unityshell is missing. The most probable cause of this is because tha package had a bad installation or maybe it wasn't installed at all.

The solution is simple, as Unity dependencies com along with lightdm package by installing it before Liquid Galaxy there wouldn't be any issues. The possible commands to do this would be:

```
~$ sudo apt install lightdm
```

or

```
~$ sudo apt install ubuntu-unity-desktop
```

Finally, the last important error is on line 895 and follows as:

```
895. chmod: cannot access '/etc/ssh/
  ↪ ssh_host_dsa_key': No such file or
  ↪ directory
```

This stderr is generated by the chmod (change mode) command, this command's main pourpouse is to change write and read permissions on a file so that for example, a bash script could write on it. As in error one on line 838, chmod is unable to locate the dsa key file for ssh package that should be on **/etc/ssh/ssh_host_dsa_key**. This error will probably cause a bad configuration of the ssh keys and its probably caused by a bad installation of package ssh that already comes with Ubuntu 18.04 Bionic. A possible solution would be to issue the following commands:

```
~$ sudo apt remove ssh
~$ sudo apt install ssh
~$ reboot
```

I've also found a last error present on both log files so that couldn't be the main cause of Liquid Galaxy not working properly on Ubuntu 18.04 Bionic. The error is caused by apache2 and is the following:

```
#Ubuntu 16.04 Xenial
1050. apache2_reload: apache2: Syntax error on
   ↪ line 208 of /etc/apache2/apache2.conf:
   ↪ Could not open configuration file /etc/
   ↪ apache2/httpd.conf: No such file or
   ↪ directory
```

```
#Ubuntu 18.04 Bionic
1053. apache2_reload: apache2: Syntax error on
   ↪ line 208 of /etc/apache2/apache2.conf:
   ↪ Could not open configuration file /etc/
   ↪ apache2/httpd.conf: No such file or
   ↪ directory
```

### Installation differences (Package uncompatibilities?)

The most probable issue that prevents Liquid Galaxy from running smoothly on Ubuntu 18.04 Bionic is package uncompatibility as the package versions installed on Ubuntu Xenial are different from the ones installed at Ubuntu 16.04 Xenial.

One of the main incompatibilities is in the PHP package:

PHP 7.0 – Ubuntu Xenial (16.04)

PHP 7.2 – Ubuntu Bionic (18.04)

It might be that this slight package version upgrade is causing some kind of incompatibility.

Finally another great incompartibility by nvidia packages may be creating some problems. The number of packages installed on Xenial is lower than the ones installed on Bionic. The Nvidia package list is the following:

| Package Name | Ubuntu 16.04 | Ubuntu 18.04 |
|---|---|---|
| nvidia-367 | ● | ● |
| nvidia-375 | ● | ● |
| nvidia-384 | ● | ● |
| nvidia-opencl-icd-3384 | ● | |
| nvidia-prime | ● | ● |
| nvidia-settings | ● | ● |
| libnvidia-cfg1-390 | | ● |
| libnvidia-common-390 | | ● |
| libnvidia-compute-390 | | ● |
| libnvidia-compute-390:i386 | | ● |
| libnvidia-decode-390 | | ● |
| libnvidia-decode-390:i386 | | ● |
| libnvidia-encode-390 | | ● |
| libnvidia-encode-390:i386 | | ● |
| libnvidia-fbc1-390 | | ● |
| libnvidia-fbc1-390:i386 | | ● |
| libnvidia-gl-390 | | ● |
| libnvidia-gl-390:i386 | | ● |
| libnvidia-ifr1-390 | | ● |
| libnvidia-ifr1-390:i386 | | ● |
| nvidia-compute-utils-390 | | ● |
| nvidia-dkms-390 | | ● |
| nvidia-driver-390 | | ● |
| nvidia-kernel-common-390 | | ● |
| nvidia-kernel-source-390 | | ● |
| nvidia-utils-390 | | ● |

Note that package nvidia-opencl-icd-3384 is being installed on Xenial but not on Bionic. That could be the source of the problem. The solution in this case would be to install the package manually:

```
~$ sudo apt install nvidia-opencl-icd-3384
```