



# Ubuntu Maverick Installation Guide

## Contents

- 1 The Options
- 2 Updated Open Source Driver PPA's
- 3 Installing Proprietary Drivers a.k.a. Catalyst/fglrx
  - 3.1 Restricted Drivers Manager
  - 3.2 Installing the drivers manually
    - 3.2.1 Before you start
    - 3.2.2 Download the latest Catalyst package.
    - 3.2.3 Create .deb packages.
    - 3.2.4 Install .debs.
    - 3.2.5 Generate a new /etc/X11/xorg.conf file
      - 3.2.5.1 Generic Config
      - 3.2.5.2 Minimal Config
      - 3.2.5.3 X2/Dual GPU Cards
      - 3.2.5.4 Dual/Multi Monitors
    - 3.2.6 Force use of the new xorg.conf (if necessary)
    - 3.2.7 Test your installation
    - 3.2.8 Just in case
- 4 Hardware Video Decode Acceleration (EXPERIMENTAL)
- 5 Updating Catalyst/fglrx
- 6 Removing Catalyst/fglrx
- 7 Issues
  - 7.1 Build Fails and Log Shows "mixed implicit and normal rules. Stop."
  - 7.2 "Errors were encountered while processing: fglrx-amdcccle" (on 64-bit systems)
  - 7.3 Problems Starting Xserver
  - 7.4 Slow Maximizing Windows/General 2D Slowness
  - 7.5 Unsupported Hardware Watermark
  - 7.6 Mesa drivers
  - 7.7 Hang at logout
  - 7.8 Suspend/Hibernation
  - 7.9 Can't remove fglrx with dpkg (diversion issue)
  - 7.10 This module/version combo is already installed
  - 7.11 New kernel installed?
  - 7.12 Aticonfig not found after installation & "module does not exist" after boot

## The Options

[\[edit\]](#)

Users with ATI cards have the following driver options:

- **vesa** - very basic, lacks 2D/3D acceleration, and focuses on compatibility with all VESA-compliant graphics cards. It is good for starting the GUI environment when no accelerated driver is available/working and little else.
- **ati** - actually a thin wrapper that will invoke the radeon driver (or another ati open-source driver for pre-Radeon cards).
- **radeon** - open source driver supporting all Radeon cards. This driver has excellent 2D acceleration and compatibility with the Linux graphics stack. 3D acceleration is sufficient for desktop effects and a nice set of native Linux games. Power management is now comparable to the Catalyst driver.
- **radeonhd** - an alternate driver supporting R520-R7x0 hardware. This driver is now officially deprecated in favor of radeon. If you still want to try it, see: <https://help.ubuntu.com/community/RadeonHD>
- **Catalyst (a.k.a fglrx)** a proprietary "blob" (closed source binary) driver designed by ATI, with 3D code based off of their Windows driver. Only RadeonHD chips are supported on recent Linux distros.

## Updated Open Source Driver PPA's

[\[edit\]](#)

- **"Ubuntu-X"** : This PPA offers the latest stable releases of video driver-related components. Follow the instructions at: <https://launchpad.net/~ubuntu-x-swat/+archive/x-updates>
- **Xorg-edgers**: This bleeding-edge PPA offers video driver-related components straight from their code (git) repositories. Follow the instructions at: <https://launchpad.net/~xorg-edgers/+archive/ppa>

## Installing Proprietary Drivers a.k.a. Catalyst/fglrx

[\[edit\]](#)

- PLEASE READ FIRST!

**Which cards are no longer supported by ATI?** The ATI Radeon 9500-9800, Xpress200-1250, 690G, 740G, X300-X2500 (including Mobility RadeonHD 2300, since it is really a DirectX 9 part). See the complete list [here](#). If your card is on that list, you are limited to open-source drivers on Ubuntu Lucid/10.04 (and later). If you really need the proprietary Catalyst/fglrx driver, you will have to use an older Linux distribution, such as Debian Lenny/5.0.x or Ubuntu Hardy/8.04.x.

### ATTENTION RADEON USERS

NOTE: If you enter your card information on AMD/ATI's driver page, it will offer you the Catalyst 9-3 driver to download. However, the Catalyst 9-3 driver doesn't support X servers past 1.5, and it will not work with Maverick! **!!!SO BE CAREFUL!!!** If you tried to install Catalyst on a system with one of these cards, see the 'Removing the Driver' section to restore the default/pre-installed drivers.

## Restricted Drivers Manager

[\[edit\]](#)

NOTE: You must have the restricted repository enabled in Applications -> Ubuntu Software Center -> Edit -> Software Sources... for this to work. You must also have the fglrx-modalises and jockey-gtk (or jockey-kde for Kubuntu) packages installed. You will be limited to the drivers for your version of Ubuntu that Canonical deems stable. This may not give you the latest drivers, but should be safest. On Ubuntu Maverick, this will install Catalyst 8.780, which is roughly equivalent to Catalyst 10-10. Go to the Additional Drivers Manager (System -> Administration -> Additional Drivers) and activate the "ATI/AMD proprietary FGLRX graphics driver" (or double-click the "available driver" notification icon). Ubuntu will then install and configure the driver for you.

## Installing the drivers manually

[\[edit\]](#)

I recommend copying and pasting the commands (do not include the leading '\$') to ensure there are no typing mistakes and speed up the install process. Remember to use **Ctrl + Shift + V** or **Shift + Insert** to paste into the terminal (or go to the terminals menu, select edit and click paste).

## Before you start

[edit]

If you have previously attempted installing Catalyst, remove any leftover files by following the [Removing the Driver](#) section. Make sure *universe* and *multiverse* are enabled in your repository sources (System -> Administration -> Software Sources), or Applications->Ubuntu Software Center->edit->Software sources->Other software: check canonical partners.

Install the prerequisite packages:

```
$ sudo apt-get install build-essential cdbsh fakeroot dh-make debhelper debconf libstdc++6 dkms libqtgui4 wget execstack libelfg0
```

If you are using the *x86\_64* architecture (64 bit), be sure to install "ia32-libs" before proceeding!

```
$ sudo apt-get install ia32-libs
```

## Download the latest Catalyst package.

[edit]

This package contains both the 32-bit and 64-bit driver.

```
$ cd ~/; mkdir catalyst11.3; cd catalyst11.3/  
$ wget http://www2.ati.com/drivers/linux/ati-driver-installer-11-3-x86.x86_64.run  
$ chmod +x ati-driver-installer-11-3-x86.x86_64.run
```

You MUST make sure that the standard radeon drivers are not installed. If you don't do this, the open-source radeon driver may be used instead of the newly-installed Catalyst driver, even if the Catalyst driver is correctly installed. This may cause the error message "aticonfig: No supported adapters detected":

```
$ sudo apt-get remove --purge xserver-xorg-video-radeon
```

## Create .deb packages.

[edit]

```
$ sh ati-driver-installer-11-3-x86.x86_64.run --buildpkg Ubuntu/maverick
```

## Install .debs.

[edit]

```
$ sudo dpkg -i fglrx*.deb
```

## Generate a new /etc/X11/xorg.conf file

[edit]

Unfortunately, there is no sure way to generate the ATI version of the Xorg.conf file. It is entirely dependent on your configuration. The following subsections will attempt to address possible (and tested) variations for their respective configurations.

### Generic Config

[edit]

This will work for most people:

```
$ sudo aticonfig --initial -f
```

### Minimal Config

[edit]

A very basic config might be what you need if you have a new card that's not fully supported by aticonfig. Here follows the entirety of a minimal xorg.conf file for the Radeon 6870:

```
Section "Device"  
    Identifier "ATI radeon 6870"  
    Driver "fglrx"  
EndSection
```

### X2/Dual GPU Cards

[edit]

If you have an X2 card (e.g. 4870X2 or 5970), use... !!Do not use for two separate cards in crossfire!!

```
$ sudo aticonfig --initial -f --adapter=all
```

### Dual/Multi Monitors

[edit]

If you have a dual monitor display (also known as "Big Desktop"), use:

```
$ sudo aticonfig --initial -f  
$ sudo aticonfig --set-pcs-str="DDX,EnableRandR12,FALSE"
```

This was confirmed in <http://phoronix.com/forums/showthread.php?t=18553>.

### Force use of the new xorg.conf (if necessary)

[edit]

Some people find that changes to xorg.conf don't get used by the driver. To force the ATI driver to adopt changes made to xorg.conf, use the following command:

```
$ sudo aticonfig --input=/etc/X11/xorg.conf --tls=1
```

## Test your installation

[\[edit\]](#)

NOTE: if you don't reboot first, fglrxinfo gives an error message. Reboot the computer and type

```
$ fglrxinfo
```

into the terminal. If the vendor string contains ATI, you have installed the driver successfully. Using fglrxinfo on a system with Catalyst 10-6 and a RadeonHD 4550 returns:

```
display: :0.0 screen: 0
OpenGL vendor string: ATI Technologies Inc.
OpenGL renderer string: ATI Radeon HD 4300/4500 Series (This line may be different depending on what graphics card you are using.)
OpenGL version string: 3.3.9901 Compatibility Profile Context (This line may be different depending on what graphics card and Catalyst version you are using.)
```

Now, try:

```
$ fgl_glxgears
```

If you experience issues or a hang, you may need to disable fast TLS.

```
$ sudo aticonfig --tls=0
```

## Just in case

[\[edit\]](#)

Write down or remember this series of Alt+PrtScr key combinations, just in case your screen should go black and Ctrl+Alt+F1 and Ctrl+Alt+Backspace doesn't work.

Alt+PrtScr+r, Alt+PrtScr+s, Alt+PrtScr+e, Alt+PrtScr+i, Alt+PrtScr+n, Alt+PrtScr+u, Alt+PrtScr+b

These key-presses will reboot the system safely. To remember the key-presses, remember this nonsensical phrase: "Raising Skinny Elephants Is Never Utterly Boring".

An alternative would be to hold down Ctrl+Alt+SysRq (SysRq is usually the same key as PrintScreen) and type very slowly R E I S U B. A way to remember this is by inverting the word: "BUSIER" or remembering a phrase: "Restart Even If System Utterly Broken". This would also safely shutdown the system.

## Hardware Video Decode Acceleration (EXPERIMENTAL)

[\[edit\]](#)

This is confirmed to work for newer RadeonHD GPU's (those with UVD2). If you have a RadeonHD 4000 series or newer, you have UVD2. To see the complete list:

[http://en.wikipedia.org/wiki/Unified\\_Video\\_Decoder#UVD\\_enabled\\_GPUs](http://en.wikipedia.org/wiki/Unified_Video_Decoder#UVD_enabled_GPUs) If you have an older RadeonHD with UVD, you can use Catalyst 10-7 to get working acceleration, but users have confirmed acceleration broken since then.

For acceleration to work, you will need libva from this PPA: <https://launchpad.net/~dt1131/+archive/catalystshacks> and the xvba va-api backend from: <http://www.splittee.com/~gbeauchesne/xvba-video/> With those installed, you can get acceleration from any video player that uses VA-API. A compatible version of VLC is available in the aforementioned PPA.

## Updating Catalyst/fglrx

[\[edit\]](#)

DO NOT try to install a new version over an old one. Follow the 'Removing the Driver' section below to remove your existing driver, and then you can start at 'Downloading the latest Catalyst' to install the new one.

## Removing Catalyst/fglrx

[\[edit\]](#)

The uninstall script in the first command will only exist if you downloaded the drivers and installed the directly (rather than building packages as this guide does). Skip the first command if it does not exist.

```
$ sudo sh /usr/share/ati/fglrx-uninstall.sh
$ sudo apt-get remove --purge fglrx fglrx_* fglrx-amdcccle* fglrx-dev* xorg-driver-fglrx
```

If you plan on using open-source drivers, you will need to reinstall some packages because Catalyst overwrites or diverts some key 3D libraries with proprietary versions. For more information on this issue, see [this Ubuntu wiki page](#) 📄

```
$ sudo apt-get remove --purge xserver-xorg-video-ati xserver-xorg-video-radeon
$ sudo apt-get install xserver-xorg-video-ati
$ sudo apt-get install --reinstall libgl1-mesa-glx libgl1-mesa-dri xserver-xorg-core
$ sudo mv /etc/X11/xorg.conf /etc/X11/xorg.conf.backup
```

## Issues

[\[edit\]](#)

### Build Fails and Log Shows "mixed implicit and normal rules. Stop."

[\[edit\]](#)

If the installation fails and you find the above message in /var/lib/dkms/fglrx/8.780/build/make.log, it may be because you're using a pentium-build wrapper around gcc. See what the following ls command returns:

```
$ ls -la /usr/bin/gcc
```

If it shows that gcc is a link to builder-cc, temporarily redirect the link to point to the real gcc (gcc-4.4 in Ubuntu Maverick). This should allow you to install fglrx:

```
$ sudo ln -sf /usr/bin/gcc-4.4 /usr/bin/gcc
```

When you're finished installing the driver, return the gcc link to its original value:

```
$ sudo ln -sf /usr/bin/builder-cc /usr/bin/gcc
```

## "Errors were encountered while processing: fglrx-amdcccle" (on 64-bit systems)

[edit]

Most likely, you probably did not have the "ia32-libs" package installed beforehand. If you have a 64 bit install, the above dpkg command may complain that "Errors were encountered while processing: fglrx-amdcccle". This is because of a dependency of the amdcccle package on 32 bit libraries. If you receive this error, use the following command, which will force the installation of all of the 32 bit dependencies, and then the amdcccle package:

```
$ sudo apt-get -f install
```

Catalyst 11.3 on 64-bit systems may require the `--force-overwrite` command in the above `dpkg` command:

```
$ sudo dpkg -i --force-overwrite fglrx*.deb
```

## Problems Starting Xserver

[edit]

If you get a black screen hang, the first thing to check is if `xorg.conf` is the problem.

You can disable the `xorg.conf` with:

```
$ sudo mv /etc/X11/xorg.conf /etc/X11/xorg.conf.disabled
```

Reboot and check to see if things work now.

You can reinstate the file with:

```
$ sudo mv /etc/X11/xorg.conf.disabled /etc/X11/xorg.conf
```

Before tweaking ACPI settings, try ensuring `/dev/null` is `chmodded` to 0666. This intermittently changes when using the `nano` (and possibly other) editors with `sudo` and the `group/world` permissions are unset. This leads to the ATI drivers hanging on boot or otherwise. A quick and dirty `init` script saved as `/etc/init/chmodnull` does the trick for me -

```
$ start on filesystem
$
$ script
$   chmod 0666 /dev/null
$   chmod 0666 /lib/udev/devices/null
$ end script
```

This has been tested using Ubuntu 10.04 64-bit on a ATI Radeon HD 4830 (HP Envy 15-1060ea). It's worth noting that I had to disable TLS (`aticonfig --tls=0`) to get things to stay stable!

If you've properly installed the driver, but experience problems when starting the X server, such as hanging, black/white/gray screen, distortion, etc., your system BIOS may have a buggy ACPI implementation. To work around, press `Ctrl+Alt+F1` to get to a terminal (or failing that, boot to recovery mode) and run:

```
$ sudo aticonfig --acpi-services=off
```

If this method works, you should consider checking your system vendor's BIOS changelogs for relevant ACPI fixes, updating your BIOS, and reenabling the driver's ACPI services.

## Slow Maximizing Windows/General 2D Slowness

[edit]

As of Catalyst 10-6, a new, faster 2D acceleration method is used as the default, replacing the old XAA method. If you're not running compositing/desktop effects, and are having problems with 2D operations, you may want to fall back to the old XAA. This command will do that:

```
$ sudo aticonfig --set-pcs-str=DDX,ForceXAA,TRUE
```

In the last case (at least) it is essential to run it without the Xorg server working. To do so, press `CTRL+ALT+F1`, log in and type the following (this will close all your programs so save your work before):

```
$ sudo service gdm stop
$ sudo aticonfig --set-pcs-str=DDX,ForceXAA,TRUE
$ sudo service gdm start
```

## Unsupported Hardware Watermark

[edit]

This can happen if your card's PCI ID wasn't officially certified to work with a particular version of Catalyst. It does not necessarily mean that your card is unsupported, but it does mean that you shouldn't file bugs with that particular card/driver combination. If you installed the driver by downloading it from AMD/ATI, installing a newer version of Catalyst will probably help.

If you installed the proprietary driver included with Ubuntu or you do not want to upgrade to a newer version, it may be possible to work around the issue by using a control file from a newer version of Catalyst than the one you're running.

```
$ cd ~/; mkdir catalyst11.3; cd catalyst11.3/
$ wget http://www2.ati.com/drivers/linux/ati-driver-installer-11-3-x86.x86_64.run
$ chmod +x ati-driver-installer-11-3-x86.x86_64.run
$ sh ati-driver-installer-11-3-x86.x86_64.run --extract driver
$ sudo mv /etc/ati/control ~/control.bak
$ sudo cp driver/common/etc/ati/control /etc/ati
```

## Mesa drivers

[edit]

If `fglrxinfo` reports that Indirect rendering by Mesa is in place, even though you have installed ATI driver, you might want to remove Mesa:

- Remove the package `xserver-xgl`.

```
$ sudo apt-get remove xserver-xgl
```

**Explanation:** If you installed this previously in order to make compiz work, it will not allow direct rendering on your display. You can check out if this is what it causing the problem by running

```
$ DISPLAY=:0 glxinfo | grep render
```

If it returns an ATI renderer, it means that xgl is being displayed indirectly on the display 1. (Taken from [\[1\]](#))

**Warning:** This might make your compiz stop working as it is configured to use XGL. A solution might be to run the Envy script in order to configure compiz. Or, if Compiz stopped working due to "Composite" problem, check that the following is set in the /etc/X11/xorg.conf

```
Section "Extensions"
    Option          "Composite"    "Enable"
EndSection
```

## Hang at logout

[\[edit\]](#)

If you experience hangs when logging out (of X) it is probably due to the /etc/ati/athatievevtsd.sh script looking for X authorisation files in the wrong place when it starts up. You can kill the hanging athatievevtsd.sh processes from a console tty to allow the shutdown of the X server. This can be fixed permanently with:

```
$ sudo mkdir -p /var/lib/xdm/authdir
$ sudo ln -s /var/run/xauth /var/lib/xdm/authdir/authfiles
```

If that doesn't work then you can disable atievevtsd with this command:

```
$ sudo /usr/sbin/update-rc.d -f atievevtsd remove
```

Before the above commands verify that /etc/ati/athatievevtsd.sh exists after build and install, if not just do: (assuming that the installer is in the directory we used to install)

```
$ cd ~/catalyst11.3
$ sh ati-driver-installer-11.3-x86.x86_64.run --extract driver
$ sudo cp driver/packages/Ubuntu/dists/lucid/replacements/athatievevtsd.sh /etc/ati/athatievevtsd.sh
$ sudo chmod +x /etc/ati/athatievevtsd.sh
```

You'll have to restart for this to take effect.

## Suspend/Hibernation

[\[edit\]](#)

Suspend hibernation **works** with the latest driver.

For Radeon 3200, to wake up from suspend, I had to add the following lines to /etc/X11/xorg.conf: (This settings is not good option, if you are using compiz-fusion or any other transparency-based thingie. Not working for HD 3850)

```
Section "Extensions"
    Option          "Composite"    "Disable"
EndSection

Section "ServerFlags"
    Option          "AIGLX"        "off"
EndSection
```

Another way to get it working is to do enable Composite, but when you want to suspend simply disable Compiz Fusion. How would you do that? Easy! Install the fusion-icon package (in repos by default):

```
$ sudo apt-get install fusion-icon
```

After you have done that, launch it (Applications > System > Compiz Fusion Icon) and it appears in your notification area. To switch, simply right-mouse click on the icon and select Metacity. Your desktop will flicker and windows will disappear, but after a while they appear again. Now try to suspend.

When you wake up again, you can (hopefully) unlock your screen and there you go! Now you want Compiz back again, so right-mouse-click on the Compiz Fusion Icon and select Compiz again. Desktop flickering again, but then voila! Your Compiz Fusion Desktop is back again! (At least, that is how it is supposed to work)

KNOWN PROBLEM: When you switch back, all your windows are on the same desktop. This happens because you switched to metacity.

Hopefully this helped some people, as it did for me! Solution posted by zwyber@gmail.com

## Can't remove fglrX with dpkg (diversion issue)

[\[edit\]](#)

If dpkg refuses to remove an fglrX package and complains about a diversion of a file, you might need to manually remove it. For example, if dpkg complains:

```
dpkg-divert: mismatch on divert-to
when removing `diversion of /usr/lib/libGL.so.1.2 to /usr/share/fglrX/diversions/libGL.so.1.2 by xorg-driver-fglrX'
found `diversion of /usr/lib/libGL.so.1.2 to /usr/lib/fglrX/libGL.so.1.2.xlibmesa by xorg-driver-fglrX'
```

then:

```
$ sudo dpkg-divert --remove /usr/lib/libGL.so.1.2
```

## This module/version combo is already installed

[\[edit\]](#)

If you get this error-message, simply uninstall the previous version before installing the new one with:

```
$ sudo dkms remove -m fglrX --all
```

## New kernel installed?

[\[edit\]](#)

In theory, DKMS should automatically install the fglrx kernel module for your new kernel the first time you boot it. Should you need to manually install it:

```
$ sudo dkms build -m fglrx -k `uname -r`
$ sudo dkms install -m fglrx -k `uname -r`
```

if amdccle doesn't work and says Identifier is not a valid word. Use lower case letter in xorg.conf

## Aticonfig not found after installation & "module does not exist" after boot

[\[edit\]](#)

This scenario is possible when the driver installation has seemingly succeeded and is possibly related to previous use of fglrx through the Jockey (i.e. you first used drivers provided by Ubuntu but then upgraded to ones available from AMD's website). When doing aticonfig --initial after driver installation, you might end up not having the aticonfig available at all:

```
aticonfig: command not found
```

After booting you might receive X error '(EE) Failed to load module "fglrx" (module does not exist, 0)'. These do not necessarily indicate that the installation has failed completely. On command line, do

```
$ ls /usr/lib/fglrx/bin
```

and see if the command lists some Ati related programs. If they are listed but not found from /usr/bin, it is possible that the "update-alternatives" fglrx .deb installation does has been ignored. See man update-alternatives for more information about the concept and workings of alternatives. In practice, update-alternatives is supposed to create several symbolic links to the files in the fglrx directory, but it will be ignored if the alternatives for the very related gl\_conf entry has been set to manual. Do

```
$ update-alternatives --get-selections | grep gl_conf
```

and see if the mode is manual instead of auto and if mesa is mentioned instead of fglrx in the path that is printed. In this case you need to

```
$ sudo update-alternatives --set gl_conf /usr/lib/fglrx/ld.so.conf
```

to set fglrx as the active alternative. You can alternatively (no pun intended) and additionally change the gl\_conf into automatic mode before the installation this way:

```
$ sudo update-alternatives --auto gl_conf
```

After that, the alternatives should automatically be configured correctly when the graphics driver .debs are installed.

Category: [Installation Documentation](#)



This page was last modified on 20 March 2011, at 17:24.

This page has been accessed 65,658 times. [Disclaimers](#)

Content is available under [Public Domain](#).

[Privacy policy](#)

[About chtml.com](#)

