

Contents

1	General HOWTO information	2
2	General Kernel configuration	3
3	AGPGart Kernel module install and setup	4
3.0.1	ATI's Internal AGP driver	5
3.0.2	Kernel's AGP driver	6
3.1	Getting the kernel drivers	6
3.2	Installing drivers	7
3.3	GART patch installation	7
3.4	Kernel configuration	7
4	Steps to take before installing the drivers on any distribution	8
5	Installing drivers on a ex-nVidia box	8
6	Installing ATI drivers on a RedHat Machine	9
6.1	Where to get the driver	9
6.2	Steps to take before installing the drivers	9
6.3	Installing the drivers	9
7	Installing ATI drivers on a Gentoo machine	10
7.1	Where to get the driver	10
7.2	Steps to take before installing the drivers	10
7.3	Installing the driver	10
7.4	Additional info/troubleshooting	10
8	Installing ATI drivers on a Slackware Machine	11
8.1	Where to get the driver	11
8.2	Steps to take before installing the drivers	11
8.3	Installing the driver	11
9	Installing the ATI drivers on a Mandrake machine	11
9.1	Where to get the driver	11
9.2	Steps to take before installing the drivers	11
9.3	Installing the drivers	12
10	Installing the ATI drivers on a SuSE machine	12
10.1	Where to get the driver	12
10.2	Steps to take before installing the drivers	12
10.3	Installing the drivers	12

11	Installing the drivers on a Debian machine	13
11.1	Where to get the driver	13
11.2	Steps to take before installing the drivers	13
11.3	Installing the drivers	14
12	Common steps for all distributions	14
12.1	Kernel 2.6 driver configuration	14
12.2	Building the module	15
12.3	Configuration of X Window	15
13	Two monitor configurations	16
13.1	Dual Head Configuration	16
13.1.1	Configuration trough fglrxconfig	16
13.1.2	XF86Config-4 configuration	18
13.1.3	Problems	18
13.2	Big Desktop (Xinerama like) configuration	19
13.2.1	Configuration trough fglrxconfig	19
13.2.2	XF86Config-4 configuration	20
13.2.3	Problems	21
13.3	Clone configuration	21
13.3.1	Configuration trough fglrxconfig	22
13.3.2	Editing XF86Config	23
13.3.3	Problems	23
13.4	Xinerama configuration	23
13.4.1	Problems	24
13.4.2	Credits	24
14	XF86Config-4 editing	24
14.1	Changing the keyboard layout and mouse mount point	24
14.2	Changing the ChipID	25

1 General HOWTO information

This HOWTO is divided into 2 main parts. In the first, there is information on how to correctly set up the agpgart. This is important, as a lot of motherboards are not supported in the driver agpgart. Since a lot of users are having problems getting it to work, there are step-by-step instructions written there.

The second part is about the driver installation in most of the widely used distributions.

The third part focuses on general settings, common setups and dualhead configuration.

IMPORTANT!!!

For any of this to work, you have to have the kernel sources installed. RedHat (fedora), Mandrake and SuSE allow the user to install the sources using their

package managers. Gentoo, Slackware and other similar distributions are built from the source, so they have it installed already.

It is very important, that you download the correct driver version. The 3 available choices are Xfree 4.1.0, 4.2.0 and 4.3.0. To find out which one to choose, run the Check.sh script, that comes with the ATI drivers.

- `sh check.sh` (perform this in the directory to which you unpacked the ATI Driver zip archive.

You can also use the following command:

- `cat /var/log/XFree86.0.log | grep -i version | head`

2 General Kernel configuration

This section applies to all distributions. It concerns the settings that must be enabled in the kernel for the driver to work. In this guide I will only look into the gfx specific settings, and completely ignore the other settings (IDE, eth...). Note, that without your distribution default settings, your kernel will not work. For information on these settings you should look at your distributions kernel guide:

Gentoo: www.gentoo.org

Slackware: The guide you followed during the installation and kernel configuration

Other: Usually there is a readme on kernel configuration found in the source directory (should be `/usr/src/`)

First of all you should make sure your kernel sources are all set up. After you have them unpacked in `/usr/src/your-kernel-sources` you must make sure to link them to `/usr/src/linux` with the following command:

- `ln -s /usr/src/your-kernel-sources /usr/src/linux`

If the linux link already exist you can remove it with the command:

- `rm /usr/src/linux`

Once your sources are set up you should configure them. For this follow the specific distribution kernel guides, as each distribution requires their own settings to be enabled. Once you are done with setting up your kernel sources enter menuconfig again:

- `cd /usr/src/linux`
- `make menuconfig`

And enable the following options:

- `/dev/agpgart` under “character devices” (look at “3 - AGPGart Kernel module install and setup” for more info on this)
- MTRR (Memory Type Range Register) under “processor type and features”

Then disable:

- Direct Rendering Manager (XFree86 DRI support) under “character devices”
- Kernel debugging under “Kernel hacking”

Once you have done this, exit the menuconfig (don’t forget to save). After that run (as root in your `/usr/src/linux` folder):

- `make dep && make clean bzImage modules modules_install`

With this command you will create a `bzImage` kernel file in your `/usr/src/linux/arch/i386/boot/`. The `i386` part may change accordingly to your arch. Now you must copy the kernel image to `/boot`. (in some distros you will have to mount `/boot` first)

- `mount /boot #` only do this if your `/boot` partition doesn’t get mounted automatically
- `cp /usr/src/linux/arch/i386/boot/bzImage /boot/bzImage.new #` change the `bzImage.new` to whatever you want your image to be named

After this you will have to edit your `lilo/grub` settings. For both it should be enough if you use the same data as for the already working kernel. The `lilo` conf file is in `/etc/lilo.conf`. After updating it you will have to run `/sbin/lilo`

- `/sbin/lilo`

For `grub` you will have to edit `/boot/grub/grub.conf`. After the changes nothing else is needed. Move on to AGPGart configuration, which is very IMPORTANT!!!

3 AGPGart Kernel module install and setup

Most users have two choices. They can either use the internal driver AGPGart module/support, or the external (kernel) module. Unfortunately some chipsets have problem with one or the other. Use the following table to decide which `agpgart` to use. If you decide on using the internal, skip this section. If you decide on running an external `agpgart`, read on. Also, you should finish this step (for external `agp` only) before trying to build the driver (as explained later in the howto).

The info was gathered by Rage3d member Mythos.

3.0.1 ATI's Internal AGP driver

Chipset	Status	Driver Info
Via KX133	Works!	At least with 2.9.6 and newer
Via KT133	Works!	At least with 3.2.8 (Probably with older and newer also)
Via KT266	Works!	At least with 3.7.0 (Probably with newer also)
Via KT266A	Works!	At least with 3.2.8 (Probably with newer also)
Via KT333	Works!	At least with 3.2.8 (Probably with newer also)
Via KT600	Works!	At least with 3.7.6 (Probably with older also)
Nvidia Nforce2	Unknown!	Should work with 3.7.0. Users with 3.7.6 have problems, although a user repo
Intel 845E	Works!	At least with 2.9.7 and newer
Intel 875P	Works!	At least with 3.7.0
SiS 645DX	Works!	At least with 3.2.8 and older (Probably with newer also)
SiS 735	Works!	At least with 3.2.8 and newer
AMD 762	Works!	At least with 3.2.8 PROBLEM: 3.7.0 do not work! (Probably the same fo

3.0.2 Kernel's AGP driver

Chipset	Status	Kernel Info
Via KX133	Works!	2.4 and newer (Via chipset support)
Via KT133	Works!	2.4 and newer (Via chipset support)
Via KT266	Works!	2.4 and newer (Via chipset support)
Via KT333	Works!	2.4.21 and newer (Via chipset support)
Via KT400	Works!	At least with 2.6.3 (Via chipset support)
Via KT600	Works!	2.4.21? and newer (Via chipset support)
Via K8T800	Works!	At least with 2.6.4 (Via chipset support) PROBLEM: Needs agp_try_u
Nvidia Nforce2	Works!	2.4.22 and newer (Nvidia nForce/nForce2 chipset support)
Intel 845E	Works!	At least with 2.4.19 and newer
Intel 875P	Works!	At least with 2.6.4 (Intel 440LX/BX/GX, 8xx and E7x05 chipset su
SiS 645DX	Should Work!	Needs 2.6.5 due to some agp changing mode problem!
SiS 735	Works!	At least with 2.4.21 and newer (SiS chipset support)
SiS 746	Does not Work!	Probably 2.6.6 will be Ok!
AMD 762	Works!	At least with 2.6 kernels

Part 3.1 and 3.2 are just general instructions for finding the right files. 3.3 is common for kernel patching. If you don't have to patch the kernel, you can safely skip to 3.4 (most users can do that, especially with kernels above 2.4.20).

3.1 Getting the kernel drivers

nForce users can download their agp drivers from <http://www.nvidia.com/object/linux.html>. If your distro is listed you should download the appropriate RPM. For others, simply download the tar.gz.

Most of the other motherboards are supported in the kernel by default, but if they are not, use google to find them or ask on various Linux message boards

(www.rage3d.com/board for example)

3.2 Installing drivers

Usually there are install instructions provided with the driver. Follow them and move on to the next step. nForce users can follow the next procedure:

Installing the drivers with the RPM is simple, just do this:

- `rpm -i nForce_driver.rpm`

For the tar.gz package there is a little more to do:

- `cd /where/the/package/is/located`
- `tar -xvzf nforce.tar.gz` (substitute nforce with the package name)
- `cd nforce` (substitute nforce with the package name)
- `make`
- `su` (become root)
- `make install`

This should set almost everything up.

3.3 GART patch installation

This step has to be done by everybody, in order to enable AGP Gart support in the kernel.

- `cd /usr/src/linux-2.4.20` (change this to your kernel path)
- `patch -p1 < linux-2.4.20-agpgart.diff` (again, change to appropriate kernel)

3.4 Kernel configuration

First of all we must configure the kernel to support our new nForce AGP Gart.

- `cd /usr/src/linux-2.4.20`
 - `su` (become root)
 - `make menuconfig`
1. In the 'Processor type and features', make sure that an x86 compatible CPU is selected.
 2. In the 'Character devices' section, select '/dev/agpgart (AGP Support)' as a modularized kernel feature (you can compile it into the kernel).

3. In the same section, enable your chipset agp driver as a modularized kernel feature (you can compile it into the kernel).

Now you can exit and save the config. Then you have to compile everything.

- `make dep && make clean bzImage modules modules_install`

When the compilation is complete just copy the bzImage to /boot. If you already have a bzImage in /boot it is wise to back it up

- `cd /boot`
- `mv bzImage bzImage.backup`
- `cd /usr/src/linux-2.4.20/arch/i386/boot/`
- `cp bzImage /boot/bzImage`

With this copied you may need to either change your bootloader configuration, or just refresh your lilo parameters. When you boot into your new kernel, just install the ATI drivers normally (follow the guide for your distribution), with one exception. When, during the driver configuration, you are asked about the AGPGart, select the EXTERNAL AGPGART.

For the driver to load properly, the agpgart and your specific chipset gart must be enabled at boot time (in case you are using a gfx login). This can be done by adding the following line to /etc/modprobe.conf :

- `pre-install fglrx /sbin/modprobe "-k" "agpgart"`

If you are using the 2.6 kernel, add the following line instead:

- `install fglrx /sbin/modprobe --ignore-install fglrx && { /sbin/modprobe via-agp; /bin/true; } #change via-agp to your chipset agp driver`

4 Steps to take before installing the drivers on any distribution

The most important step take is making sure you have loaded the agpgart modules (only when using external) prior to installing the driver. That way, when the driver builds, it will get all the correct settings.

5 Installing drivers on a ex-nVidia box

nVidia's driver changes some files upon installation. To make sure that everything will work you MUST uninstall nVidia's driver first.

- `/path/to/nvidia/driver/driver.run --uninstall`

This way everything will be returned to it's original value. From here on you can follow your distributions install instructions.

6 Installing ATI drivers on a RedHat Machine

6.1 Where to get the driver

The newest drivers can always be found at <http://www.ati.com/support/driver.html>.

6.2 Steps to take before installing the drivers

The first thing to do is to get the right driver version. In general RedHat 9.0 users have to get Xfree version 4.3.0 and RedHat 8.0 users must get the 4.2.0 Xfree version of the drivers. If you are unsure, or want to test it, you should use this file (<http://www2.ati.com/drivers/Check.sh>) to find out the right version for you (type `sh Check.sh` in the dir to which you downloaded the file). In some distributions it is possible to have a Xfree version, that doesn't match none of the driver versions (4.3.5 for example). In this case you should use the x.x.0 version of the driver (in the example this is version 4.3.0).

Once you know which Xfree version you need to download head here (<http://www.ati.com/support/driver.html>) select Linux as your operating system, select gfx driver (or FireGL if you own a FireGL card) and then select you gfx card. A page will open on which you can download the correct driver version for your PC. It is also important to have the kernel source installed on your PC. You can install it by using the Packet management tool found under System Settings.

6.3 Installing the drivers

It is important to understand that ATI drivers should be installed from the console/terminal. Thus, in this guide, everything will be done in the terminal.

Go into text mode (so no gfx modules interfere with the installation).

- `su` (become root)
- `init 3`

Here you will have to login again. Login as root. Go to the directory to which you downloaded the drivers.

- `cd /DIRECTORY_NAME/SUBDIRECTORY_NAME`

1st try to install the drivers.

- `rpm -ihv fglrx-glc22-4.3.0-3.2.5.i586.rpm` (change name accordingly to your file name)

If you are in luck everything should go smoothly and the driver should install and configure itself. Probably you will not be so lucky. The Xfree driver will be conflicting with your driver installation.

2nd try to install the drivers.

- `rpm -i -force fglrx-glc22-4.3.0-3.2.5.i586.rpm`

OR

- `rpm -e --nodeps Conflicting_File_and_Version`
- `rpm -ihv fglrx-glc22-4.3.0-3.2.5.i586.rpm`

In the first case you will be forcing the driver installation, thus ignoring any files that are conflicting with it. In the second case you will first be removing the file that is conflicting and then normally installing the driver. This can cause some dependencies checks to fail, thus breaking your X installation. For that reason it is wiser to use the first command.

Now you can move to part 11 - Common steps for all distributions

7 Installing ATI drivers on a Gentoo machine

7.1 Where to get the driver

You should use portage (emerge).

7.2 Steps to take before installing the drivers

The kernel has to be configured for the driver. Look at the Kernel configuration file for info.

7.3 Installing the driver

In gentoo installing the driver is pretty simple. All you have to do is to run emerge:

- `emerge ati-drivers`

Emerge will then do all the work (taking care of all the dependencies). Once the emerging is done you should run `opengl-update` to switch the `opengl` provider to ATI.

- `opengl-update ati`

7.4 Additional info/troubleshooting

If you are using the external `agpgart`, you should put the following line in `/etc/modules.autoload.d/kernel-2.6` (change to 2.4 accordingly):

- `agpgart`
- `nvidia-agp` (if you have an nForce2 motherboard, otherwise use the appropriate one)
- `fglrx`

A Gentoo only HowTo already exists at <http://odin.prohosting.com/wedge01/gentoo-radeon-faq.html>

8 Installing ATI drivers on a Slackware Machine

8.1 Where to get the driver

The newest drivers can always be found at <http://www.ati.com/support/driver.html>.

8.2 Steps to take before installing the drivers

You will have to make some changes to the kernel. Check the kernel configuration part for info on that.

8.3 Installing the driver

First of all, you have to convert the RPM from www.ati.com to a format Slackware understands. This can be done with the utility `rpm2targz` (in debian you can get it with the help of `apt-get`).

- `rpm2targz DRIVER_FILE_NAME.rpm`

(From here on you have to be root to continue)

- `su`

Use Slackwares program to install the driver:

- `installpkg DRIVER_FILE_NAME.tar.gz`
- `ldconfig # this is optional`

Once this is finished you will have to build the kernel module. Info on how to do this can be found in the Common steps part later in this HowTo. After that it is just a matter of running `fglrxconfig` to set up X Windows. Info on how to do it can also be found in the Common steps part.

9 Installing the ATI drivers on a Mandrake machine

9.1 Where to get the driver

You can either download them from www.ati.com (and in that case follow the RedHat guide), or use `RPMDrake/urpmi` to get them. Note that you can only get the drivers via `urpmi` if you are a paying Mandrake Club Member.

9.2 Steps to take before installing the drivers

The only step to take before the installation is to make sure you have the kernel sources installed. You can get them by using `RPMDrake`.

9.3 Installing the drivers

Use RPMDrake to download the drivers. They will get configured automatically. You only have to run `fglrxconfig` to configure X Windows.

10 Installing the ATI drivers on a SuSE machine

10.1 Where to get the driver

You can either download them from www.ati.com, but it is advisable to get them from SuSE (www.suse.com)

10.2 Steps to take before installing the drivers

Just make sure you have your kernel sources installed. You can use Yast2 to get and install them. Also, make sure they are linked to `/usr/src/linux`. If they are not, just link them:

- `ln -s /usr/src/linux-kernel-2.X.X /usr/src/linux`

If a symbolic link already exists, but you are sure it's not your kernels, just erase it before making the new one.

- `rm /usr/src/linux`

10.3 Installing the drivers

It is important to understand that ATI drivers should be installed from the console/terminal. Thus, in this guide, everything will be done in the terminal.

Go into text mode (so no `gfx` modules interfere with the installation).

- `su` (become root)
- `init 3`

Here you will have to login again. Login as root. Go to the directory to which you downloaded the drivers.

- `cd /DIRECTORY_NAME/SUBDIRECTORY_NAME`

1st try to install the drivers.

- `rpm -ihv fglrx-glc22-4.3.0-3.2.5.i586.rpm` (change name accordingly to your file name)

If you are in luck everything should go smoothly and the driver should install and configure itself. Probably you will not be so lucky. The Xfree driver will be conflicting with your driver installation.

2nd try to install the drivers.

- `rpm -i --force fglrx-glc22-4.3.0-3.2.5.i586.rpm`

OR

- `rpm -e --nodeps Conflicting_File_and_Version`
- `rpm -ihv fglrx-glc22-4.3.0-3.2.5.i586.rpm`

In the first case you will be forcing the driver installation, thus ignoring any files that are conflicting with it. In the second case you will first be removing the file that is conflicting and then normally installing the driver. This can cause some dependencies checks to fail, thus breaking your X installation. For that reason it is wiser to use the first command.

Now you can move to part 11 - Common steps for all distributions

11 Installing the drivers on a Debian machine

11.1 Where to get the driver

You should get the driver from www.ati.com. You will then convert them to .deb (debian package). Unless you are using the unstable branch of debian, you are using XFree 4.2.0. Download the according driver.

11.2 Steps to take before installing the drivers

First of all you should get your kernel sources ready. Unless you have them installed already do so now (use `apt-get` for that). Debian doesn't automatically unpack kernel sources, so you have to do it yourself:

- `cd /usr/src`
- `tar xjvf kernel-source-2.X.X.tar.bz2 #substitute with your kernel sources`
- `ln -s kernel-source-2.X.X linux #this way you will link your new sources with the expected sources directory. If /usr/src/linux exists you can delete the link: rm /usr/src/linux`

Now we will make sure our sources are configured to the working kernel:

- `cp /boot/config-2.X.X-Y-YYY .config #change this according to your kernel version and config name`
- Now you need to edit the make file. Use your favourite text editor. Change the `EXTRAVERSION` in the file to `"-Y-YYY"`
- `make oldconfig`
- `make dep` (in 2.4 kernel) OR `make prepare` (in 2.6 kernel)

- `ln -s /usr/src/linux /lib/modules/2.X.X-Y-YYY/build` #this will link the modules to the right directory. Again make sure you change the path to your kernel version

Next, we will convert the ATI rpm to deb. To do this you will need alien. Use apt-get to get it. Once you have it run the following command:

- `alien /directory/subdirectory/ati-driver.rpm` #this will create a .deb file in the directory you are currently in

11.3 Installing the drivers

Now we will use the .deb file we created earlier:

- `dpkg -i ati-driver.deb` #you may have to use some of the `-force` options to get it to install (conflicts with libglx). Use “`dpkg -help`” for more info.

Once the drivers are installed, you will have to build the module. Look at the “Building the module” in the Common steps part of the HowTo.

12 Common steps for all distributions

12.1 Kernel 2.6 driver configuration

Currently the ATI fglrx driver have some problems compiling against the 2.6 kernel. To get it to work you will need to make some changes to the driver itself.

First you will have to get the driver and install it.

- `rpm -ihv -force fglrx-driver.rpm`

Gentoo users have it easy here, as the emerge command automatically patches the driver, which then works correctly without any other changes.

However, if you are NOT using gentoo, than you will have to make two changes:

(This step is not needed if you are using 3.7.0 + drivers) First you will have to apply the VMALLOC patch. The patch can be found on top of the (online) page. Once you have it copy it into `/lib/modules/fglrx`:

- `cp vmalloc.patch /lib/modules/fglrx`

Then move to the `build_mod` directory and execute the patch:

- `cd /lib/modules/fglrx/build_mod`
- `patch -p1 < ../vmalloc.patch`

Once this is done, you will have to make some changes to `/lib/modules/fglrx/build_mod/firegl_public.c` Open the file and change all instances of:

- `#if ((PAGE_ATTR_FIX == 1) || (LINUX_VERSION_CODE == KERNEL_VERSION(2,4,19)))`

into this:

- `#if ((PAGE_ATTR_FIX == 1) || (LINUX_VERSION_CODE >= KERNEL_VERSION(2,4,19)))`

Save the changes and exit. Now you can build the module (look at the following section).

12.2 Building the module

If you changed anything about the kernel, since you last installed the driver, you should run the following commands. You should make sure your agpgart modules (if you are using the external agpgart) are loaded.

- `cd /lib/modules/fglrx/build_mod #(change directory to where our task lies)`
- `sh make.sh #(gathering kernel information)`
- `cd /lib/modules/fglrx/ #(change directory to where our second task lies)`
- `sh make_install.sh #(configuring the driver to our kernel)`

If you already have a working Xfree86 from previous driver installations/configurations you can keep it, though it is advisable to rerun `fglrxconfig`.

12.3 Configuration of X Window

It is advisable you enter text mode before running `fglrxconfig`, as some modules in the GFX mode may interfere with the configuration. In most distributions it is just a matter of running as root:

- `init 3`

In Gentoo and Debian however, you have to either run:

- `/etc/init.d/xdm stop # change xdm to your login manager (xdm, kdm...)`

or simply quit the window manager if you have a text login. Once you are in text mode simply run `fglrxconfig`.

- `fglrxconfig`

With the “fglrxconfig” command you will start the configuration program in which all you have to do is answer the questions that you are asked. Some info that might help:

- If you own a USB Wheel mouse select “PS/2 or USB with PS/2 protocol” and then “PS/2 Wheel Mouse”
 - When specifying the mouse mount point (default /dev/mouse) you may run into some issues with either X not starting or the mouse not responding. In that case you may want to try and specify the mount point as /dev/input/mice
 - If your keyboard layout is not listed in the list simply select US. You will be able to change this later.
 - Default settings (selected by simply pressing enter) are usually the best, but might not be correct in your case. Always be sure you select the right answer.
 - It is best to select no “restrictions - any user has access to the OpenGL ICD” so any user can use OpenGL acceleration.
 - For beginners it is best to select “Default” under Application Profiles.
 - Select “compatible operation (for WineX, VM emulators and some profilers)” to be sure that no compatibility problems will appear in WineX.
- Once this is done simply restart Linux, so that all the modules get loaded correctly.

13 Two monitor configurations

Samples XF86Config-4 has been provided for you. The link to it can be found on top of the online page. To view it first decompress it:

- `gzip -d /path/to/XF86Config-4*.gz`

After that you can read it with your favourite text editor. Note, that this file is not supposed to work on your config, but is instead only a guideline for your setup.

13.1 Dual Head Configuration

This is about having two different drivers running and consequently two different desktops, which is a bit different from the Clone option which displays the same content on both screens, or the Big Desktop (Xinerama like) option, where you have one desktop shared on both screens. In order to have a dual head setup you can either use the fglrxconfig utility (preferred one!) or just edit your XF86Config.

The Dual-Head XF86Config-4 sample can be found here:

<http://www.rage3d.com/content/articles/atilinuxhowto/XF86Config-4-Dualhead.gz>

13.1.1 Configuration trough fglrxconfig

1. Follow the standard steps until you reach the FireGL Screen Layout section.

2. Enable Dual Head (2 screens - two drivers), option in the FireGL "Screen Layout section -> Choose configuration from the list below" question, by pressing '5'.
3. Choose the location of your second screen in regard to your first screen. You can choose between right '1', left '2', above '3' and below '4'.
4. Select your primary (p) and secondary (s) display. The auto detect '1' option should work in most cases, but if you face any problem with it, you can select the appropriate options from the list. All you have to remember is that the primary head is the bottom DVI port on X1, LCD output on M9 or the only DVI port on other cards and that the secondary head is the top DVI port on X1 or the VGA port on all other cards.
5. Give the specs of your two monitors. Be careful with this to follow the manufacture's specs else you will burn your monitors! First you are asked about the horizontal sync range of your first monitor. You can enter the specs of your monitor if none of the predefined ones is appropriate. To do this press '8', enter the horizontal sync range (i.e. "31.5-61") and press Enter. Then you will have to insert the vertical sync range. Press '6' if you want to specify your own one. Then enter the vertical sync range (i.e. "52-120") and press Enter. Afterwards you have to do the same for your second monitor.
6. Set your preferred resolutions. If the default one is not the desired one press '1' to change it, else '2' to accept it. You can insert many resolutions by just inserting a series of numbers for the corresponding resolutions you want. The first one will be your default resolution (if your card and monitor can support it). The same resolutions will apply to both screens. You can also edit afterwards your XF86Config-4 to insert your own modelines, if you know what you are doing! 16bit is not supported in 3.7.0. Moreover if you do not want to have the same resolutions on your second monitor as in the first one you will also have to edit your XF86Config-4. Also for R200 family (Radeon8500-9000 Pro, Fire GL8700/8800/E1) the maximum OpenGL resolution is 2048x2048 and for R300 family (Radeon 9500-9700Pro, Fire GL X1/Z1) it is 2560x2560. (See Screen section in sample XF86Config)
7. Be sure you don't enable FSAA (look at the PROBLEMS section how you can enable it). Just answer "All (Visuals for 2,4 and 6 Samples per Pixel)" (1) in the "Please select multi sample visuals you want to share" question and no to the question "Do you want to force multi sample visuals for every OpenGL application", both in the Full Scene Anti Aliasing (FSAA) section. If you enable FSAA you will get a black screen when you try to get into X. (See Device section of the sample XF86Config)
8. Continue the configuration as always.

13.1.2 XF86Config-4 configuration

In order to use the driver dualheaded, you need to define in the XF86Config, two Devices, one for the first head and one for the second. Usually if your card supports dualhead configurations, if you run the lspci command, you will see something like:

- 01:00.0 VGA compatible controller: ATI Technologies Inc Radeon R300 NE [Radeon 9700]
- 01:00.1 Display controller: ATI Technologies Inc Radeon R300 [Radeon 9700](Secondary)

So our card has 2 driver displays, in 01:00.0 and 01:00.1. Normally you have to make 2 device section one for BusID "PCI:1:0:0" and one for BusID "PCI:1:0:1". The fglrx is an exception requiring both devices to have BusID of your first head, else it won't work. Also make sure you set option "DesktopSetup" to "0x00000000" (See the Device sections in the sample XF86Config)

Then you have to configure your monitors. You will need 2 Monitor sections, with the appropriate modelines and the appropriate settings for your 2 monitors. Be sure that you follow the manufacture's specs if you don't want to burn your monitors! (See the Monitor sections in the sample XF86Config)

The two display devices must also drive 2 screens, so you have to add in the XF86Config 2 Screen sections, with the resolutions and the depth of color you want. Be sure that each screen is connected to the appropriate device and monitor. (See the Screen sections in the sample XF86Config)

Finally you have to configure where your second monitor is located regarding your first one. (See the ServerLayout section in the sample XF86Config)

All the above instructions can also be used if you have more than one cards, allowing even triplehead or more configurations. You only have to be sure that you use the correct driver for the other card and that you have defined the appropriate Screen and Monitor sections and the physical layout of your monitors.

13.1.3 Problems

1. You should have hardware accelerated 3D graphics on your second head. Games like QuakeIII should work as expected on the second head, but some other 3D applications won't work. They will just get killed. Some users have also reported that the Xserver will just restart if you try to run a 3D application on the second head. So be careful when trying this!
2. Up to now it is probably better to not use 3.7.0 drivers because of some screen corruption. Normally you should be able to use NoAccel to avoid this corruption, but with current driver (3.7.0) if you set it to "yes" you will just have a cloned image of the first monitor in the second head, so you will loose Dualhead.

3. If you have problems with qt, and 3D acceleration doesn't work, try either to configure qt without opengl support, or use qt-3.3.0 with `-dlopen-opengl` parameter in the configure script.
4. If you want FSAA support, be sure that you have option `Option "EnablePrivateBackZ" "yes"` in both device sections of your `XF86Config` and that you have correct FSAA options in each Device. If you have no FSAA options on your second Device, then by default you won't have FSAA on the second head. If you don't enable `PrivateBackZ` in either of the Devices you will get a black screen and you will have to press `Ctrl-Alt-Del` to reboot your system. Then after rebooting, you will find the following message in `/var/log/XFree86.0.log`: `(EE) fglrx(1): Failed to allocate shared Z/stencil buffer!`
5. If you change the options of the second head, like `NoAccel` and `NoDRI`, the driver will override them with the options in the first head, so they won't work. The only options you can use are the FSAA options, as described above.

13.2 Big Desktop (Xinerama like) configuration

This is about Big Desktop (Xinerama like) option, where you have one desktop shared on both screens. Consequently you use one driver and the two screens share the same big framebuffer and one window manager. So you can have your desktop spreading on both displays and move applications from one monitor to the other! On the other hand, if you want two drivers and two different desktops you should look at the Dual Head configuration above. Also don't forget that you can use the driver dualheaded with the Xinerama extension to have the same effect as in Big Desktop (currently does not work). But in this case you won't be able to use DRI (accelerated 3D), since DRI and Xinerama are mutual exclusive! (See 12.4)

In order to have a Big Desktop setup you can either use the `fglrxconfig` utility (preferred one!) or just edit your `XF86Config`.

The `XF86Config-4` sample can be found here:

<http://www.rage3d.com/content/articles/atilinuxhowto/XF86Config-4-BigDesktop.gz>

13.2.1 Configuration through `fglrxconfig`

1. Follow the standard steps until you reach the FireGL Screen Layout section.
2. Enable Big Desktop (2 screens - one framebuffer), option in the FireGL "Screen Layout section -> Choose configuration from the list below" question, by pressing '4'.
3. Choose where you want your primary monitor to be located, regarding your second one. You can choose between left (1), right (2), top (3) and bottom (4).

4. Choose if you want overlapping screens, which is intended to allow image overlapping between the two monitors. Press 'y' or 'n'
5. If you selected overlapping screens, you have to insert the number of pixels of the overlapping area.
6. Select your primary (p) and secondary (s) display. The auto detect (1) option should work in most cases, but if you face any problem with it, you can select the appropriate options from the list. All you have to remember is that the primary head is the bottom DVI port on X1, LCD output on M9 or the only DVI port on other cards and that the secondary head is the top DVI port on X1 or the VGA port on all other cards.
7. Give the specs of your first monitor. Be careful with this to follow the manufacture's specs else you will burn your monitor! First you are asked about the horizontal sync range. You can enter the specs of your monitor if none of the predefined ones is appropriate. To do this press '8', enter the horizontal sync range (i.e. "31.5-61") and press Enter. Then you will have to insert the vertical sync range. Press '6' if you want to specify your own one. Then enter the vertical sync range (i.e. "52-120") and press Enter. Afterwards you have to do the same for your second monitor.
8. Set your preferred resolutions. If the default one is not the desired one press '1' to change it, else '2' to accept it. You can insert many resolutions by just inserting a series of numbers for the corresponding resolutions you want. The first one will be your default resolution (if your card and monitor can support it). The same resolutions will apply to both screens. You can also edit afterwards your XF86Config-4 to insert your own modelines, if you know what you are doing! 16bit is not supported in 3.7.0. Big Desktop mode will use the best resolution from the desired ones, your lower spec monitor is capable of, for both monitors! Also for R200 family (Radeon 8500-9000 Pro, Fire GL8700/8800/E1) the maximum OpenGL resolution is 2048x2048 and for R300 family (Radeon 9500-9700Pro, Fire GL X1/Z1) it is 2560x2560. (See Screen section in sample XF86Config)
9. Answer no by pressing 'n' in the question "Do you want to use a virtual desktop where the dimensions are bigger than your maximum x and y screen resolution" since Big Desktop doesn't support panning and scrolling.
10. Continue the configuration as always.

13.2.2 XF86Config-4 configuration

In order to use the driver in Big Desktop mode, you need to define in the XF86Config, one Device section as always. What will make the driver to switch mode is the value of option "DesktopSetup". Valid options are "0x00000200" which means that the monitor connected to the primary adapter is the left

display, "0x00000201" which means that it is the right display, "0x00000300" which means that it is the top display and "0x00000301" which means that it is the bottom display. Also you have to define options "HSync2" and "VRefresh2", which are referred to the second monitor (You can skip this since you will need to define 2 monitor sections). Also if you want to set an overlapping area for the two screens set option "ScreenOverlap", to the number of pixels you want this area to be. 0 to disable it.

(See the Device sections in the sample XF86Config)

Then you have to configure your monitors. You will need 2 Monitor sections, with the appropriate modelines and the appropriate settings for your 2 monitors. Be sure that you follow the manufacture's specs if you don't want to burn your monitors! The driver will use the same modeline for both screens and it will select it according to the lower spec monitor. So if you have a 640x480 capable monitor and a 1600x1200 one, big desktop will use a 2x(640)x480.

(See the Monitor sections in the sample XF86Config)

You will also need to define as always a Screen section, with the desired resolutions and depth of colors. The resolutions you will enter should be each screen's resolutions, not the resolution of both. For example a 1280x1024 resolution will give you a 2560x1024 desktop. Up to drivers 3.7.0, 16 bit color is not supported. Also don't use a bigger virtual resolution, since Big Desktop doesn't support panning and scrolling. Also for R200 family (Radeon 8500-9000 Pro, Fire GL 8700/8800/E1) the maximum OpenGL resolution is 2048x2048 and for R300 family (Radeon 9500-9700 Pro, Fire GL X1/Z1) 2560x2560, for both screens.

(See the Screen sections in the sample XF86Config)

13.2.3 Problems

1. You won't be able to maximize an application only on one monitor since fglrx driver with Big Desktop mode doesn't support Xinerama extensions, in order to make aware the window manager that you have one desktop across 2 monitors. Also firing up a 3D application, will always display it on your left monitor, or the top monitor, according to your configuration, which is the display with 0,0 coordinates. But a 3D application can make use of your whole desktop!
2. Vidix driver from xine and mplayer can only be used on the primary monitor. Xv should work for both monitors.
3. You may have problems with some games going to fullscreen.
4. If you enable any of the NoAccel and NoDRI options, the Xserver will crash.

13.3 Clone configuration

This is about Clone option, where the contents of your primary screen are displayed on the second one. This can be used for demonstration or presentation

purposes. Setup looks like the Big Desktop, with some differences.

In order to have a Clone setup you can either use the `fglrxconfig` utility (preferred one!) or just edit your `XF86Config`.

The `XF86Config-4` sample can be found here:

<http://www.rage3d.com/content/articles/atilinuxhowto/XF86Config-4-Dualhead.gz>

13.3.1 Configuration through `fglrxconfig`

1. Follow the standard steps until you reach the FireGL Screen Layout section.
2. Enable Clone Mode (2 screens - same content), option in the FireGL "Screen Layout section -> Choose configuration from the list below" question, by pressing '3'.
3. Select your primary (p) and secondary (s) display. The auto detect (1) option should work in most cases, but if you face any problem with it, you can select the appropriate options from the list. All you have to remember is that the primary head is the bottom DVI port on X1, LCD output on M9 or the only DVI port on other cards and that the secondary head is the top DVI port on X1 or the VGA port on all other cards.
4. Give the specs of your first monitor. Be careful with this to follow the manufacture's specs else you will burn your monitor! First you are asked about the horizontal sync range. You can enter the specs of your monitor if none of the predefined ones is appropriate. To do this press '8', enter the horizontal sync range (i.e. "31.5-61") and press Enter. Then you will have to insert the vertical sync range. Press '6' if you want to specify your own one. Then enter the vertical sync range (i.e. "52-120") and press Enter. Afterwards you have to do the same for your second monitor.
5. Set your preferred resolutions. If the default one is not the desired one press '1' to change it, else '2' to accept it. You can insert many resolutions by just inserting a series of numbers for the corresponding resolutions you want. The first one will be your default resolution (if your card and monitor can support it). The same resolutions will apply to both screens. You can also edit afterwards your `XF86Config-4` to insert your own modelines, if you know what you are doing! 16bit is not supported in 3.7.0. Clone mode will use the best resolution from the desired ones, your lower spec monitor is capable of, for both monitors! Also for R200 family (Radeon 8500-9000 Pro, FireGL 8700/8800/E1) the maximum OpenGL resolution is 2048x2048 and for R300 family (Radeon 9500-9700Pro, Fire GL X1/Z1) it is 2560x2560. (See Screen section in sample `XF86Config`)
6. Answer no by pressing 'n' in the question "Do you want to use a virtual desktop where the dimensions are bigger than your maximum x and y screen resolution" since Clone mode doesn't support panning and scrolling.
7. Continue the configuration as always.

13.3.2 Editing XF86Config

In order to use the driver in Clone mode, you need to define in the XF86Config, one Device section as always. What will make the driver to switch mode is the value of option "DesktopSetup". The valid option for Clone is "0x00000100". Also you have to define options "HSync2" and "VRefresh2", which are referred to the second monitor (You can skip this since you will need to define 2 monitor sections).

(See the Device sections in the sample XF86Config)

Then you have to configure your monitors. You will need 2 Monitor sections, with the appropriate modelines and the appropriate settings for your 2 monitors. Be sure that you follow the manufacture's specs if you don't want to burn your monitors! The driver will use the same modeline for both screens and it will select it according to the lower spec monitor.

(See the Monitor sections in the sample XF86Config)

You will also need to define as always a Screen section, with the desired resolutions and depth of colors. Up to drivers 3.7.0, 16 bit color is not supported. Also don't use a bigger virtual resolution, since Clone mode doesn't support panning and scrolling. Also for R200 family (Radeon 8500-9000 Pro, Fire GL 8700/8800/E1) the maximum OpenGL resolution is 2048x2048 and for R300 family (Radeon 9500-9700 Pro, Fire GL X1/Z1) 2560x2560.

(See the Screen sections in the sample XF86Config)

13.3.3 Problems

None known.

13.4 Xinerama configuration

You can find the Xinerama XF86Config-4 sample here:

<http://www.rage3d.com/content/articles/atilinuxhowto/XF86Config-4-Xinerama.gz>

Xinerama setup is like the Dual Head one. Currently, despite the fact that I tried hard, with 3.7.0 you can't get a Xinerama setup working!!!! If you follow the instructions described below you will have a desktop which is a combination of Dual Head and Clone. The first screen is cloned on the second but you can move the mouse cursor as if it was a Dual Head configuration. Basically you can't setup a Xinerama from fglrxconfig, so you will have to make your own XF86Config or edit the Xinerama XF86Config sample. The instructions are exactly the same as in Dual Head configuration, with the following differences:

You have to set the following option in the "ServerFlags" section: Option "Xinerama" "true"

You have to disable DRI, because DRI and Xinerama can't co-exist. In both Device sections set:

- Option "no_dri" "yes" # disable DRI

If you want to have overlapping screens you can use something like this, in the "ServerLayout" Section:

- Screen "Screen 2" #Resolution 1280 1024
- # Make screens overlap 80 pixels
- Screen "Screen 1" Relative "Screen 2" 1200 0 # Screen 1 is right of Screen 2

13.4.1 Problems

It doesn't work!

13.4.2 Credits

Submitted: Panagiotis Papadakos

Email: papadako@ics.forth.gr

14 XF86Config-4 editing

Logged in as root go to `/etc/X11` and open the file `XF86Config-4`. In here you can edit a lot of things. From keyboard layouts and mouse settings, to more advanced stuff, like your cards Chip ID and other settings.

14.1 Changing the keyboard layout and mouse mount point

Find these lines (pc105 and US can look different in your case):

- Option "XkbRules" "xfree86"
- Option "XkbModel" "pc105"
- Option "XkbLayout" "US"

Here you can edit the last line to make sure that your keyboard layout is used.

You can also change the mouse mount point to be used. Just scroll to the mouse part of the file and change the path (usually it is set to `/dev/mouse` - a good alternative is `/dev/input/mice`).

- Identifier "Mouse1"
- Driver "mouse"
- Option "Protocol" "ImPS/2"
- Option "ZAxisMapping" "4 5"
- Option "Device" "/dev/input/mice"

The `ZAxisMapping` can be used to change the wheel proprieties.

14.2 Changing the ChipID

In case your card is currently not supported by the drivers, there is a chance it will work with the ChipID from a similar card. For example, if you have an ATI Radeon 9800 XT and the card is not supported, you can change the chip ID into the Radeon 9800 Pro. Example:

- # Chipset "generic"

This is the generic setting. To override it, remove the # so it is uncommented, and change generic into the card:

- # Chipset "4168"

In this case I changed it into a Radeon 9800 ones. This will make the card act as a 9800 one. Changing the ChipID into a ChipID of a very different is not recommended. The list of ChipIDs can be found here:

<http://pciids.sourceforge.net/iii/?i=1002>