″%′"

DRI Compilation Guide

Precision Insight, Inc.

18 May 2000

1. Preamble

1.1 Copyright

1.2 Trademarks

OpenGL is a registered trademark and SGI is a trademark of Silicon Graphics, Inc. Unix is a registered trademark of The Open Group. The 'X' device and X Window System are trademarks of The Open Group. XFree86 is a trademark of The XFree86 Project. Linux is a registered trademark of Linus Torvalds. Intel is a registered trademark of Intel Corporation. 3Dlabs, GLINT, and Oxygen are either registered trademarks or trademarks of 3Dlabs Inc. Ltd. 3dfx, Voodoo3, Voodoo4, and Voodoo5 are registered trademarks of 3dfx Interactive, Incorporated. Matrox is a registered trademark of Matrox Electronic Systems Ltd. ATI Rage is a registered trademark of ATI Technologies, Inc. All other trademarks mentioned are the property of their respective owners.

2. Introduction

This document describes how to download, compile and install the DRI project. This information is intended for experienced Linux developers. Beginners are probably better off installing precompiled packages.

Edits, corrections and updates to this document may be mailed to brian@precisioninsight.com.

3. Prerequisites

You'll need the following:

- At least 400MB of free disk space. More is needed if you want to build with debugging information or keep several build trees.

- A fast system. Using a PIII-550 it takes about 1/2 hour to build everything.
- GCC compiler and related tools.
- ssh (secure shell) for registered developer downloading of the DRI source tree

For 3dfx Voodoo3 hardware, you'll also need:

- Glide3x headers and runtime library if you want to use the 3dfx driver. These can be obtained from linux.3dfx.com.
- Linux kernel 2.2.x. The DRI developers have been using stock RedHat 6.1 systems (kernel 2.2.12-20). Later kernel versions will be supported in the future.

For Matrox G200/G400 hardware, you'll also need:

- Linux kernel 2.3.51. Older kernel versions will not work.

For Intel i810 hardware, you'll also need:

- Linux kernel 2.3.99-pre6. Older kernel versions will not work.

For ATI Rage hardware, you'll also need:

- Linux kernel 2.3.51. Other kernel versions may work but this one is known to work.

4. Kernel preparation

You may have to upgrade your Linux kernel in order to use the DRI. This is because you need a kernel version which supports AGP. Building a new Linux kernel can be difficult for beginners but there are resources on the Internet to help. This document assumes experience with configuring, building and installing Linux kernels.

Linux kernels can be downloaded from www.kernel.org

Download the needed kernel and put it in /usr/src. Create a directory for the source and unpack it. For example:

```
cd /usr/src
rm -f linux
mkdir linux-2.3.51
ln -s linux-2.3.51 linux
bzcat linux-2.3.51.tar.bz2 | tar xf -
```

Now configure your kernel. You might, for example, use `make menuconfig` and do the following:

- Go to *Code maturity level options*
- Enable *Prompt for development and/or incomplete code/drivers*
- hit ESC
- Go to *Character devices*
- Disable *Direct Rendering Manager (XFree86 DRI support)* since we'll use the DRI module from the XFree86/DRI tree.
- Go to */dev/agpgart (AGP Support) (EXPERIMENTAL) (NEW)*
- Hit SPACE twice
- Enable all chipsets' support for AGP

It's recommended that you turn on MTRRs under *Processor type and Features*, but not required.

Configure the rest of the kernel as required for your system (i.e. Ethernet, SCSI, etc)

Exit, saving your kernel configuration.

Edit your /etc/lilo.conf file. Make sure you have an image entry as follows (or similar):

```
image=/boot/vmlinuz
      label=linux.2.3.51
      read-only
      root=/dev/hda1
```

The important part is that you have /boot/vmlinuz without a trailing version number. If this is the first entry in your /etc/lilo.conf AND you haven't set a default, then this will be your default kernel.

Now compile the new kernel:

```
cd /usr/src/linux-2.3.51
make dep ; make bzImage
make modules ; make modules_install
make install
```

Note that the final part, make install, will automatically run lilo for you.

Now reboot to use this new kernel.

5. Downloading the XFree86/DRI CVS Sources

The DRI project is hosted by VA Linux Systems' SourceForge. The DRI source code, which is a subset of the XFree86 source tree, is kept in a CVS repository there.

The DRI CVS sources may be accessed either anonymously or as a registered SourceForge user. It's recommended that you become a registered SourceForge user so that you may submit non-anonymous bug reports and can participate in the mailing lists.

5.1 Anonymous CVS download:

1. Create a directory to store the CVS files:

   ```
   cd ~
   mkdir DRI-CVS
   ```

   You could put your CVS directory in a different place but we'll use `~/DRI-CVS/` here.

2. Check out the CVS sources:

   ```
   cd ~/DRI-CVS
   cvs -d:pserver:anonymous@cvs.dri.sourceforge.net:/cvsroot/dri login
     (hit ENTER when prompted for a password)
   cvs -z3 -d:pserver:anonymous@cvs.dri.sourceforge.net:/cvsroot/dri co xc
   ```

   The -z3 flag causes compression to be used in order to reduce the download time.

5.2 Registered CVS download:

1. Create a directory to store the CVS files:

   ```
   cd ~
   mkdir DRI-CVS
   ```

   You could put your CVS directory in a different place but we'll use `~/DRI-CVS/` here.

2. Set the CVS_RSH environment variable:

```
setenv CVS_RSH ssh      // if using csh or tcsh
export CVS_RSH=ssh      // if using sh or bash
```

3. Check out the CVS sources:

```
cd ~/DRI-CVS
cvs -z3 -dYOURID@cvs.dri.sourceforge.net:/cvsroot/dri co xc
```

   Replace `YOURID` with your CVS login name. You'll be prompted to enter your sourceforge password.

   The -z3 flag causes compression to be used in order to reduce the download time.

6. Updating your CVS sources

In the future you'll want to occasionally update your local copy of the DRI source code to get the latest changes. This can be done with:

```
cd ~/DRI-CVS
cvs -z3 update -dA xc
```

The -d flag causes any new subdirectories to be created and -A causes most recent trunk sources to be fetched, not branch sources.

7. Compiling the XFree86/DRI tree

7.1 Make a build tree

Rather than placing object files and library files right in the source tree, they're instead put into a parallel *build* tree. The build tree is made with the `lndir` command:

```
cd ~/DRI-CVS
ln -s xc XFree40
mkdir build
cd build
lndir -silent -ignorelinks ../XFree40
```

The build tree will be populated with symbolic links which point back into the CVS source tree.

Advanced users may have several build trees for compiling and testing with different options.

7.2 Edit the host.def file

The `~/DRI-CVS/build/xc/config/cf/host.def` file is used to configure the XFree86 build process. You can change it to customize your build options or make adjustments for your particular system configuration

The default `host.def` file will look something like this:

```
#define DefaultCCOptions -Wall
#define DefaultGcc2i386Opt -O2
#define LibraryCDebugFlags -O2
#define BuildServersOnly YES
#define XF86CardDrivers vga tdfx mga r128 i810
#define LinuxDistribution LinuxRedHat
#define DefaultCCOptions -ansi GccWarningOptions -pipe
#define BuildXF86DRI YES
#define HasGlide3 YES
/* Optionally turn these on for debugging */
/* #define GlxBuiltInTdfx YES */
/* #define GlxBuiltInMga YES */
/* #define GlxBuiltInR128 YES */
/* #define DoLoadableServer NO */
#define SharedLibFont NO
```

The `ProjectRoot` variable specifies where the XFree86 files will be installed. You probably don't want to use `/usr/X11R6/` because that would overwrite your default X files. The following is recommended:

```
#define ProjectRoot /usr/XF86-main
```

Especially note the *XF86CardDrivers* line to be sure your driver is listed.

If you have 3dfx hardware be sure that the Glide 3x headers are installed in `/usr/include/glide3/` and that the Glide 3x library is installed at `/usr/lib/libglide3x.so`.

If you do not have 3dfx hardware comment out the `HasGlide3` line in `host.def`.

7.3 Compile

To compile the complete DRI tree:

```
cd ~/DRI-CVS/build/xc/
make World >& World.LOG
```

Or if you want to watch the compilation progress:

```
cd ~/DRI-CVS/build/xc/
make World >& World.LOG &
tail -f World.LOG
```

With the default compilation flags it's normal to get a lot of warnings during compilation.

Building will take some time so you may want to go check your email or visit slashdot.

7.4 Check for compilation errors

Using your text editor, examine `World.LOG` for errors by searching for the pattern `***`.

Verify that the DRI kernel module(s) for your system were built:

```
cd ~/DRI-CVS/build/xc/programs/Xserver/hw/xfree86/os-support/linux/drm/kernel
ls
```

For the 3dfx Voodoo, you should see *tdfx.o*. For the Matrox G200/G400, you should see *mga.o*. For the ATI Rage 128, you should see *r128.o*. For the Intel i810, you should see *i810.o*.

If the DRI kernel module(s) failed to build you should verify that you're using the right version of the Linux kernel. The most recent kernels are not always supported.

If your build machine is running a different version of the kernel than your target machine (i.e. 2.2.12-20 vs. 2.3.99-pre6), make will select the wrong kernel headers. This can be fixed by explicitly setting the value of `TREE`. If the path to your kernel source is `/bigdisk/linux-2.3.99-pre6`,

```
cd ~/DRI-CVS/build/xc/programs/Xserver/hw/xfree86/os-support/linux/drm/kernel
make TREE=/bigdisk/linux-2.3.99-pre6
```

or alternatively, edit Makefile to include this change.

After fixing the errors, do `make World` again. Later, you might just compile parts of the source tree but it's important that the whole tree will build first.

8. Installing

After the DRI tree has been compiled you can install the XFree86 headers, libraries, programs, etc for testing.

8.1 X Installation

As mentioned above, the install directory is specified by the `ProjectRoot` variable in the `host.def` file. Create that directory now if it doesn't already exist:

```
mkdir /usr/XF86-main
```

You'll have to change to root since the install process puts several files in `/etc/X11/` and sets the setuid flag on the X server executable.

```
cd ~/DRI-CVS/build/xc
su
make install
```

Edit your `/etc/ld.so.conf` file and put `/usr/XF86-main/lib` as the first line. Continue with:

```
ldconfig
exit
```

Look in `/usr/XF86-main` to be sure the files installed there.

Strictly speaking, installing the DRI tree isn't required. It's possible to run and test the X server directly from the build directory but it's a bit error prone.

8.2 Update Locale Information

To update your X locale information do the following:

```
cd ~/DRI-CVS/build/xc/nls
xmkmf -a
make
make install
```

This will prevent a locale error message from being printed when you run Xlib programs.

9. X Server Configuration

If your X server is currently running you'll have to stop it and return to a virtual terminal.

First, setup your XF86Config file. It should load the GLX and DRI modules and specify the driver to use for your hardware. See the DRI User Guide for detailed information.

You may want to make a backup copy of your existing `/etc/X11/XF86Config` file first.

It's very important that you set the `ModulePath` option to point to your installation directory:

```
ModulePath "/usr/XF86-main/lib/modules"
```

Double check with this:

```
grep ModulePath /etc/X11/XF86Config
```

Next, your `~/.xinitrc` file controls which clients will be launched when your X server starts. You might put the following in yours:

```
xset b off
xsetroot -solid "#004070"
xmodmap -e "clear mod4"
xrdb -merge ~/.Xdefaults
xterm -geometry +0+0 &
xterm -geometry +512+0 &
fvwm
```

10. X Server Start-up

The X server can be started with:

```
xinit -- /usr/XF86-main/bin/XFree86
```

Automatic loading of DRM kernel modules was added to the X server in XFree86 4.0.1. This feature, and manual loading of kernel modules, is documented in the DRI user guide.

At this point your X server should be up and running with hardware-accelerated direct rendering. Please read the DRI User Guide for trouble shooting information.

CONTENTS

`$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/DRIcomp.sgml,v 1.3 2000/06/17 02:16:49 dawes Exp $`