



INSTITUTE OF COMPUTER SCIENCE  
FACULTY OF NATURAL SCIENCES III  
MARTIN-LUTHER-UNIVERSITY HALLE-WITTENBERG, GERMANY

HowTo

# Installation of an older proprietary NVIDIA driver under Fedora Core 6

Hendrik Bugdoll

13th February 2007

(last changed: 16th February 2007)

## Abstract

This documentation gives relevant information for the use of the NVIDIA Accelerated Graphics Driver for Linux IA32 (x86/32 bit), version 87.76, under Fedora Core 6. No longer older drivers from NVIDIA can easily be installed on Linux systems with a newer kernel (2.6.18/2.6.19). Because the current NVIDIA drivers (starting from version 96 or 97) apparently do not longer support older graphics chips like the NV25 (e.g. GeForce4 Ti4600 or Quadro4 900 XGL),<sup>1</sup> you have to use earlier drivers.<sup>2</sup> Concretely, here is described the installation for the Kernel 2.6.19-1.2895.fc6 in a step-by-step guidance.

---

<sup>1</sup>Against the data by [http://www.nvidia.de/object/linux\\_supported\\_de.html](http://www.nvidia.de/object/linux_supported_de.html) it could be observed an appropriate error message on a system with ubuntu 6.10.

<sup>2</sup>There are no older drivers available for the kernel 2.6.19-1.2895.fc6 in the Livna repository and NVIDIA not (yet) adapted the Legacy driver to the newer kernel.

---

## Installation of the NVIDIA driver 87.76 under Fedora Core 6

1. Become root (or log in as root).

```
> su
```

2. Install the packages `kernel-devel`, `xorg-x11-server-sdk`, `pkg-config` and update the package `kernel-headers` (subsequently, start again).<sup>3</sup>

```
> yum install kernel-devel xorg-x11-server-sdk pkg-config
> yum update kernel-headers
> reboot
```

3. Examine whether the kernel development files were installed for the correct kernel version.

```
> rpm -qa | grep kernel-devel
```

4. Download the driver `NVIDIA-Linux-x86-1.0-8776-pkg1.run` from `www.nvidia.com` and save it in a directory (e.g. in `/root/install`).

```
> cd /root/install
> wget http://download.nvidia.com/
    XFree86/Linux-x86/1.0-8776/NVIDIA-Linux-x86-1.0-8776-pkg1.run
```

5. Extract the shellsript.

```
> sh NVIDIA-Linux-x86-1.0-8776-pkg1.run --extract-only
```

6. Edit `nv-linux.h`,<sup>4</sup> since there is no more header file `config.h` in the newer kernel versions [1].

```
> cd NVIDIA-Linux-x86-1.0-8776-pkg1/usr/src/nv
> nano nv-linux.h
```

```
- 17: #include <linux/config.h>
+ 17: #include <linux/autoconf.h>
```

7. Edit `nv-i2c.c`,<sup>5</sup> since something was changed in the implementation of the I2C bus system in the newer kernel versions [2].

```
> nano nv-i2c.c
```

```
- 26:  .slave_send  = NULL,
- 27:  .slave_recv  = NULL,
```

8. Stop the X server and install the driver (pay attention to the correct directory of the kernel sources).

```
> init 3
> cd /root/install/NVIDIA-Linux-x86-1.0-8776-pkg1
> ./nvidia-installer -n -s --kernel-source-path=/usr/src/kernels/2.6.19-1.2895.fc6-i686
```

---

<sup>3</sup>A functioning Internet connection is needed.

<sup>4</sup>Repairs the error `linux/config.h: File or directory not found`. Of course you can use every other editor in place of `nano`.

<sup>5</sup>Repairs the errors `unknown field 'slave_send' specified in initializer` and `unknown field 'slave_recv' specified in initializer`.

---

9. Now it would have been written to the end of the log file that the driver was installed.<sup>6</sup>

```
> nano /var/log/nvidia-installer.log
```

10. Configure the X server by editing `xorg.conf` (save the file before you modify it, if necessary).

```
> cd /etc/X11
> cp xorg.conf xorg.conf.old
> nano xorg.conf
```

in Section "Module" add the loading of the module `glx` and remove the module `dri`.<sup>7</sup>

```
-: Load "dri"
+: Load "glx"
```

in Section "Device" indicate the correct driver:<sup>8</sup>

```
-: driver "nv"
+: driver "nvidia"
```

in the SubSection "Display" of Section "Screen" the correct screen resolution has to be set, for example 1920x1200 pixel for the WUXGA resolution :-)

```
-: viewport 0 0
+: modes "1920x1200"
```

11. Reboot the machine.

```
> reboot
```

12. Repair problems relating to *Security Enhanced Linux* (SELinux) [3].<sup>9</sup>

```
> chcon -t texrel_shlib_t /usr/lib/xorg/modules/drivers/nvidia_drv.so
> chcon -t texrel_shlib_t /usr/lib/xorg/modules/extensions/libglx.so
> chcon -t texrel_shlib_t /usr/lib/tls/libnvidia-tls.so.1
> chcon -t texrel_shlib_t /usr/lib/libGLcore.so.1
> chcon -t texrel_shlib_t /usr/lib/libGL.so.1
```

13. The X server should start now again.

```
> startx
```

14. Test the 3D hardware acceleration. For a first check the program `glxgears` is suitable.<sup>10</sup>

```
> glxgears
```

**Note:** After a kernel update you have to repeat the steps 8 and 11 to 14.

---

<sup>6</sup>`echo "ERROR: Kernel configuration is invalid. include/linux/autoconf.h or ... are missing.";` is no error, but only the output of an instruction with a condition test!

<sup>7</sup>If no such section exists, it must be created with Section "Module" ... EndSection.

<sup>8</sup>`nv` is only a free generic driver, which does not offer any 3D hardware acceleration.

<sup>9</sup>Repairs errors cannot restore segment prot after reloc: Permission denied. The X server starts without the fifth `chcon` call, however no OpenGL applications like `glxgears` or `glxinfo`.

<sup>10</sup>You should get more than 1000 fps.

## **References**

[1] <http://www.nvnews.net/vbulletin/showthread.php?t=77704>

[2] <http://www.nvnews.net/vbulletin/showthread.php?t=76574>

[3] <http://www.nvnews.net/vbulletin/showthread.php?t=72490>

The largest thanks go to "zander" from NVIDIA, who gave the crucial information in the nvnews forum.