# Updating the Nvidia Driver

Timothy Vismor

November 2005

**Abstract**

Procedures for upgrading the proprietary Nvidia video driver on a Fedora machine are described.

**Note**

This document is retained for archival purposes. It does not describe current practice.

# Contents

## 1 Overview

This document describes procedures for building and installing updated Nvidia drivers on a Fedora Core 3 (FC3) or Fedora Core 4 (FC4) machine. It is based on the livna rpm packaging of the proprietary Nvidia video driver.

The following assumptions also apply:

- The new kernel is running when the drivers are updated.
- The new kernel's "devel" rpm is installed (FC4 only).
- The current user is a member of the "wheel" group (i.e. has "sudo" privileges).
- RPM sources, specs, etc. are stored in the default locations used by Redhat's rpmbuild utility. The utility's build tree is rooted at ~/src/rpm.
- Sources of drivers that are not packaged by rpm are stored in their own directory under ~/src.
- The system is running an "up to date" version of the Fedora Core 3 distribution of the Linux operating system

A shell script (update_nvidia.sh) implementing the procedures discussed in this document can be found in the public download area.

Note: This is a straightforward script. It contains no error checking or other amenities. Use it at you own risk.

## 2 Modifying the Livna Spec File

System configuration involves a minor modification to the Livna packaging of the Nvidia Linux driver (currently nvidia-glx-1.0.7174-0.lvn.1.3.src.rpm).

More specifically, the Livna spec file (nvidia-glx.spec) is modified . The following patch illustrates the required change.

```
-- nvlivna.spec 2005-04-11 03:37:41.000000000 -0400
+++ nvidia-glx.spec 2005-05-24 11:59:38.000000000 -0400
@@ -289,7 +289,7 @@ echo "%{nvidialibdir32bit}" &gt;&gt; $RPM_BUILD
# and thus remove, the device files needed.
mkdir -p $RPM_BUILD_ROOT%{_sysconfdir}/udev/devices
pushd $RPM_BUILD_ROOT%{_sysconfdir}/udev/devices
-    touch `seq -f "nvidia%g" 0 15`
+    touch `seq -f "nvidia%g" 0 0`
touch nvidiactl
```

```
popd
%endif
```

As you can see, there is just one change. A single device file ( nvidia0*) is created in the /etc/udev/devices directory. We don't need devices* nvidia1 - nvidia15* so we don't create them.

## 3 Updating the Driver

The procedure for building and installing the Nvidia device driver follows:

1. Build the full set of Nvidia rpm packages (one line).

   ```
   rpmbuild -bb --target "$(uname -m)" --define "kernel $(uname -r)"
   ~/src/rpm/SPECS/nvidia-glx.spec
   ```

2. Clean up after the build by deleting the nvidia debug rpm that is generated by this procedure. (Do you want to debug the Nvidia kernel module?)

   ```
   rm ~/src/rpm/RPMS/i686/nvidia-glx-debuginfo-*.rpm<
   ```

3. At this point you could just install the new modules using rpm (exercise left for the reader). However,we keep locally built rpm's in a yum repository so a few more steps are required to complete the installation. First, move the new kernel module to the local yum repository.

   ```
    mv ~/src/rpm/RPMS/i686/nvidia-glx*.rpm ~/local/repo
    mv ~/src/rpm/RPMS/i686/kernel-module-nvidia*.rpm ~/local/repo<
   ```

4. Update the local yum repository.

   ```
   createrepo ~/local/repo
   ```

5. Update the Nvidia driver and kernel module (this will also update the nvidia-glx-devel package if it is installed).

   ```
   sudo yum update "nvidia-glx*"
   ```

An alternate form of the update command (useful if you upgrading from a script and don't want to answer questions).

```
sudo yum -y update "nvidia-glx*"
```

If this is a new installation of the Nvidia driver, you must replace the yum "update" commands with "install" directives. Yum takes you literally (unlike the "rpm -Uvh" command).

## 4 Modifications for gdm-early-login

FC4 has an "early login" feature that is disabled by default. If you want to activate this feature, you must make a one line change to the "nvidia.init" file that ships with the with the current Livna rpm. The file contains a script that enables/disables the proprietary Nvidia driver during start up. The following patch illustrates the required change.

```
--- nvidia.init 2004-11-27 17:51:45.000000000 -0500
+++ /etc/init.d/nvidia 2005-05-24 12:13:45.000000000 -0400
@@ -2,7 +2,7 @@
#
# nvidia: Enable/Disable nvidia driver in XF86Config
#
-# chkconfig:    2345 98 02
+# chkconfig:    2345 07 02
# description:  This script will check if the nvidia kernel module is present
#               for the running kernel modify the XF86Config to the 'best' config
# processname:  nvidia
```

The Nvidia driver detection script must run before the "gdm-early-login" script which starts the X server. Otherwise, the system always starts with the non-proprietary "nv" driver. Changing the start priority to 7 (from 98) works. The early login script starts with priority 9.

## 5 Activating the New Driver

Reboot. There are ways to activate the new drivers without taking the system all the way down, but the simplest solution is to restart the machine.

## 6 Concluding Remarks

Over time we have migrated away from standalone Fedora boxes. We now typically run the Ubuntu Linux distribution as a virtual machine under Parallels on OS X boxes. Canonical is much less shy than Redhat about distributing "politically incorrect" drivers and provides an extensive pre-cooked inventory of drivers that work under Parallels. We rarely need to build kernel modules "from scratch" in this ecosystem. Therefore, this document has not been maintained recently. We continue to publish it because disk space is cheap and there is a remote possibility that someone might find it useful.