

Updating Locally Built Kernel Modules

Timothy Vismor
November 2005

Abstract

Procedures for updating third party drivers following a kernel upgrade on a Fedora Core 3 (FC3) or Fedora Core 4 (FC4) system. More specifically, upgrades of the Nvidia video driver and the SPCA50X USB camera driver are described.

Contents

1	Introduction	3
2	Updating the Nvidia Driver	3
3	Updating the SPCA50X Driver	4
4	Activating the New Modules	4
5	Concluding Remarks	4

1 Introduction

When a Fedora box uses device drivers that are not supplied by the kernel, updating the kernel is more complicated than installing a new binary rpm. Typically, the external device drivers must also be rebuilt under the new kernel. This document contains instructions for creating and installing the following drivers after a new kernel is installed:

- The Nvidia proprietary video driver (for an Nvidia Geforce2 Pro card).
- The SPCA50X USB camera driver (for an Intel PC Camera Pro webcam).

The following assumptions apply to all procedures in this document:

- 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_kernel_modules.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 your own risk.

2 Updating the Nvidia Driver

This procedure assumes the [livna rpm packaging](#) of the proprietary Nvidia kernel driver used. More specifically, the procedure is based on a minor variant of the [nvidia-glx-1.0.7174-0.lvn.1.3.src.rpm](#). See the document [Updating the Nvidia Driver](#) for more details.

1. Build the kernel module rpm package (at the command prompt enter all this on one line).

```
rpmbuild -bb --without driverp --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?)

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

3. At this point you could just install the new module 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 (one line).

```
mv ~/src/rpm/RPMS/i686/kernel-module-nvidia-$(uname -r)*.rpm
~/local/repo
```

4. Update the local yum repository

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

5. Install the new driver

```
sudo yum install "kernel-module-nvidia-$(uname -r)*"
```

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

```
sudo yum -y install "kernel-module-nvidia-$(uname -r)*"
```

3 Updating the SPCA50X Driver

The SPCA50X USB camera driver is used to support an Intel PC Camera Pro webcam. The driver is not packaged as an RPM so it is built and installed using the standard autotools procedure.

```
make clean
make
sudo make install
```

The [SPCA50X driver](#) is hosted on Sourceforge. However, the preferred download for the current version is from the maintainer's ([Michel Xhaard](#)) site.

4 Activating the New Modules

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.

5 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.