Moving to Ubuntu Linux®

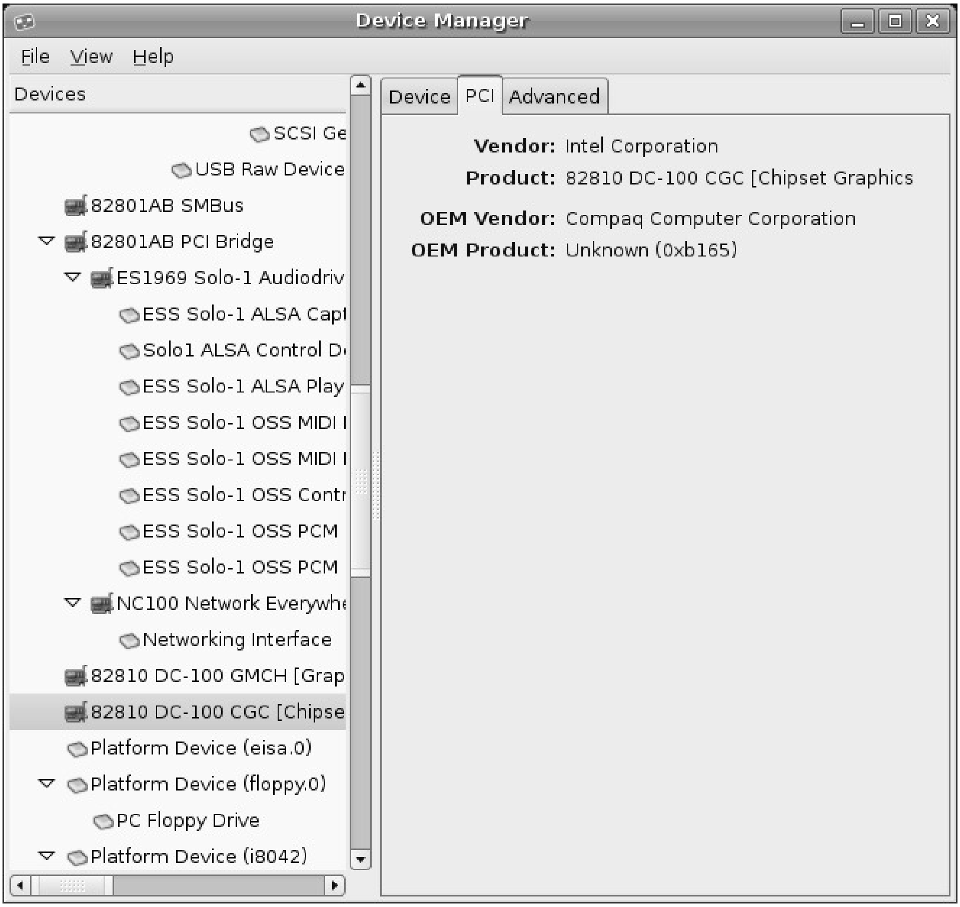
ubuntu
linux for human beings

DVD Included!

Marcel Gagné

Award-Winning Author of the *Linux Journal* "Cooking with Linux" Column

Moving to
Ubuntu Linux



Vendor: Intel Corporation
Product: 82810 DC-100 CGC [Chipset Graphics

OEM Vendor: Compaq Computer Corporation
OEM Product: Unknown (0xb165)

Figure 6–11 When listing PCI devices, the Device Manager adds a PCI information tab in the right pane.

Shell Out You can also run the command `/sbin/lspci` for a more succinct list of all PCI devices on the system. The following is a partial list from my notebook computer:

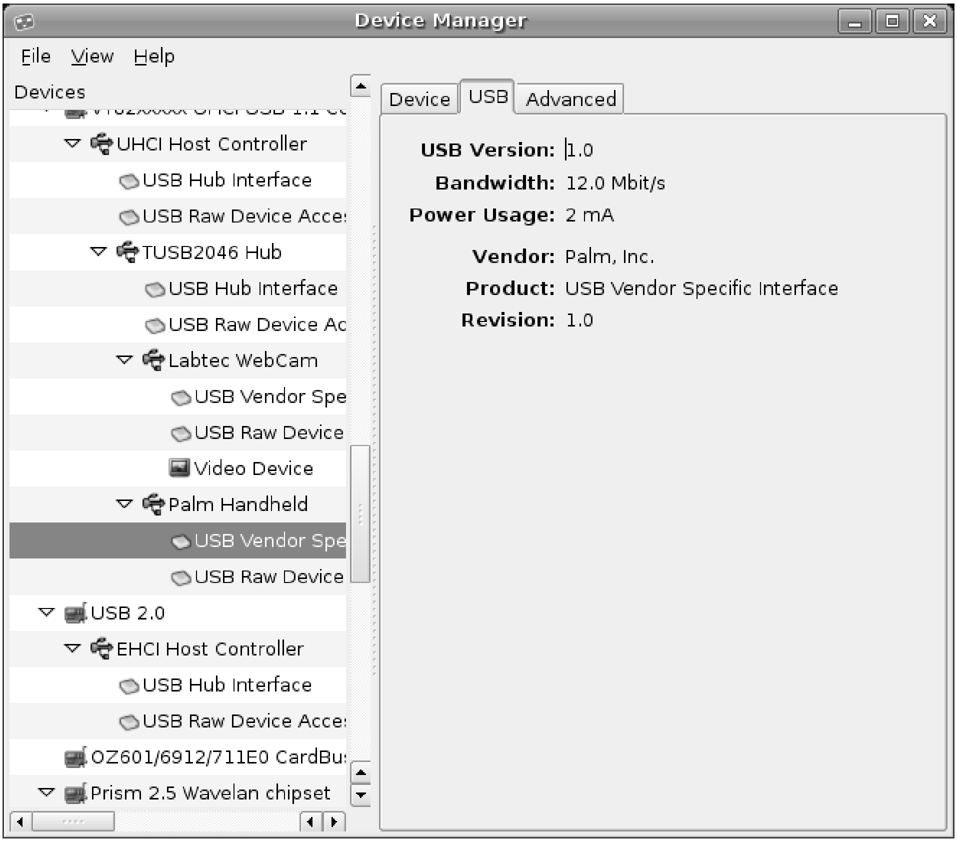
```
0000:00:00.0 Host bridge: ATI Technologies Inc
RS200/RS200M AGP Bridge [IGP 340M] (rev 02)

0000:00:01.0 PCI bridge: ATI Technologies Inc PCI
Bridge [IGP 340M]

0000:00:06.0 Multimedia audio controller: ALi Corpo-
ration M5451 PCI AC-Link Controller Audio Device
(rev 02)

0000:00:07.0 ISA bridge: ALi Corporation M1533 PCI
to ISA Bridge [Aladdin IV]

0000:00:08.0 Modem: ALi Corporation M5457 AC'97
Modem Controller

0000:00:09.0 Network controller: Intersil
Corporation Prism 2.5 Wavelan chipset (rev 01)

0000:00:0a.0 CardBus bridge: O2 Micro, Inc. OZ601/
6912/711E0 CardBus/SmartCardBus Controller
```

If the Linux kernel has the appropriate device drivers available as modules, they are automatically loaded, and nothing else needs to be done to make the device available. The reason that this information is useful has to do with those times when you do not have a driver handy or directly available. Being able to get the details on the troublesome device in this way is the first step toward getting it working.

A classic example of this is the Winmodem, so called because it was designed to work specifically with Windows. If you have one of these modems and it was not automatically configured by the system, never fear. I'll talk about Winmodems in more detail later in the chapter. For the moment, let's talk USB.

USB Devices

The whole idea behind USB was eventually to replace all those different connectors on the back of a computer. That includes serial ports, parallel ports, and mouse and keyboard connectors. The acronym stands for *Universal Serial Bus*. On any USB system, there is at least one USB hub and whatever devices are attached. If you look at Figure 6-12, you can see information displayed on two USB controllers (one is a USB 2.0 port, the other USB 1.1), a Labtec webcam, a mini USB wheel mouse, and my Palm Zire 72 (selected), all connected to my system.



USB Version: 1.0
Bandwidth: 12.0 Mbit/s
Power Usage: 2 mA

Vendor: Palm, Inc.
Product: USB Vendor Specific Interface
Revision: 1.0

Figure 6–12 The Hal Device Manager displaying USB information on a connected Palm device.

Notice that in the case of a USB device, like my Palm Zire 72, a third tab appears on the right pane. It is labeled USB and clicking it displays additional information specific to the USB device currently selected.

Trivia Time I casually mentioned that my notebook computer had both USB and USB 2.0 ports. It's worth noting because not all systems have both and there is a difference between the two. The new USB 2.0 uses an identical connector to the original USB 1 port but the hardware supports a much faster rate of information exchange. This is particularly useful for devices that must transmit a large amount of information, such as a digital video camera. USB 1 has a speed of 12Mb per second, whereas the new USB 2.0 can transmit data at 480Mb per second.

Shell Out To get a list of USB devices, try the command `lsusb`. The following output is from my notebook:

```
$ lsusb

Bus 003 Device 001: ID 0000:0000

Bus 001 Device 004: ID 0830:0061 Palm, Inc.

Bus 001 Device 003: ID 0c45:6029 Microdia
Triplex i-mini PC Camera

Bus 001 Device 002: ID 0451:2046 Texas Instru-
ments, Inc. TUSB2046 Hub

Bus 001 Device 001: ID 0000:0000

Bus 002 Device 002: ID 04fc:0013 Sunplus Tech-
nology Co., Ltd

Bus 002 Device 001: ID 0000:0000
```

The sheer number of USB devices available is phenomenal, to say the least, and the list is growing. Many of these devices use a standard set of drivers, which means that a number of things can literally be plugged in and used—no need to mess with loading drivers because it is all being done for you.

You noticed the word *many* in that last sentence, right? Keeping track of what works (*and what doesn't*) and providing access to drivers that aren't included in current distributions is the *raison d'être* of the *Linux USB Device Overview* Web site. If you find yourself looking at a new webcam, and you aren't sure whether it is supported under Linux, look there first:

```
http://www.qbik.ch/usb/devices/
```

The site is organized into sections, depending on the device type (audio, video, mass storage, and so on). Each device is assigned a status identifying just how well a device is supported, from *works perfectly* to *works somewhat* to *don't bother.*

Modems versus Winmodems

Well, here's something scary. . . .

Way back when, in the introductory chapter, I mentioned Winmodems as one of the few minuses of running Linux. You'll recall that a Winmodem is a modem designed to work only with Windows. They are sometimes referred to as *software* or *controllerless* modems and tend to be less expensive than controller-based modems.

If you are running a Winmodem, all is not lost. The Linux community is nothing if not resourceful. Even when manufacturers are slow to notice Linux users, the same isn't true the other way around. As more and more people run Linux, this becomes less and less of a problem. In time, hardware manufacturers may be building for Linux first and Windows second. In the meantime, check out the *Linmodems.Org* Web site at `http://www.linmodems.org` and you should be up and running shortly.

So just how do you transform a Winmodem into a Linmodem? Well, let me give you an example.

Among the more common Winmodems out there are those based on the Conexant chipset; these are starting to be very well supported. For the latest driver, just head on over to Linuxant's Web site (not related to Conexant) at http://www.linuxant.com. Not only can you get source drivers, but precompiled packages are available for a number of popular Linux distributions.

Identifying the Winmodem is your first step. You can use the GNOME device manager to browser through your PCI hardware, where you will get a lot of detail. You can also *shell out* and use the `lspci` command for a quick list of all the PCI devices found on your system. Here's what it looks like:

```
   $ lspci
   00:00.0 Host bridge: VIA Technologies, Inc. VT8367 [KT266]
   00:01.0 PCI bridge: VIA Technologies, Inc. VT8367 [KT266 AGP]
   00:06.0 Communication controller: Conexant HSF 56k Data/Fax/
Voice/Spkp    (w/Handset) Modem (WorldW SmartDAA) (rev 01)
   00:08.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
RTL-8139/8139C (rev 10)
   00:11.0 ISA bridge: VIA Technologies, Inc. VT8233 PCI to ISA
Bridge
   00:11.1 IDE interface: VIA Technologies, Inc. Bus Master IDE
(rev 06)
   00:11.2 USB Controller: VIA Technologies, Inc. USB (rev 18)
   00:11.5 Multimedia audio controller: VIA Technologies, Inc.
VT8233 AC97    Audio Controller (rev 10)
   01:00.0 VGA compatible controller: nVidia Corporation NV11
[GeForce2 MX DDR] (rev b2)
```

In some cases, you will find precompiled driver packages. Some are specific to your release, and others are generic. In the case of my Conexant-based Winmodem, the site provides a generic installer that should detect your system and install itself accordingly.

As per the instructions that followed the install, I typed the following command:

```
   /usr/sbin/hsfconfig
```

A short dialog followed, asking me for the country (Canada, in my case), after which the program compiled and installed my driver for me. It even linked the newly created device, `/dev/ttySHSF0`, to `/dev/modem`. I was ready to use my modem without a care.