

# PET: USB Digital Camera - How to Retrieve Those Snaps

Paul A. Marshall

## 1 Cameras Supported by this PET

This PET is for cameras that can act as USB storage devices *only*. If your camera is not of this type then this PET will not work for you. If that's the end of this PET for you, do not worry. There is still a pretty good chance that you can use your camera with the assistance of the photo libraries. Check the website <http://www.gphoto.org>.

## 2 Introduction

This Christmas was very kind to me, as it landed me with a digital camera (Santa, you da man). Unfortunately, it was supplied with a Windows driver disk which is about as much use to me as a chocolate teapot (before you suggest it, I already have a full set of AOL coasters, thank you very much).

If I had a default install of say, a recent version of Mandrake, I guess such a thing would work right out of the box. However, I tend not to have very much in my kernel or userland that I do not actually need. This meant that I had to discover just what it takes to make such a thing work. This is my effort to pull the pieces together.

## 3 What This PET Covers

This PET is for cameras that can be connected via USB and can act as *USB mass storage devices*. On my camera, this is achieved by switching to "DSC" mode. The idea here is that the camera can be mounted as though it were a regular disk. USB mass storage device may sound a bit of a misnomer (the smartmedia supplied with my camera is a whopping 16MB), but the important thing is that it uses the protocol. My camera has another mode which allows it to act somewhat like a webcam. However, this is of no interest to me. All I want to do is retrieve my snaps from the smartmedia.

The exact model of my camera is a Fujifilm FinePix 2600Zoom. My kernel is 2.4.20. My distribution is Debian Woody. All command line sequences supplied should be typed as root.

## 4 Kernel Configuration

Kernel compilation is, as they say, beyond the scope of this document. If you are unfamiliar with this procedure, refer to a tutorial (hint: LJR has a PET on this very subject).

Here, I list those kernel elements that should be enabled if they are not already part of your standard build. My own preference is to build the elements as loadable modules, where possible, and use `kmod` to load them on demand. This approach certainly makes sense for a device such as a camera which, from the point of view of your system, is only in use for short periods of time.

### 4.1 USB Support

No surprises here, you are going to need to enable USB support.

#### 4.1.1 Preliminary USB Device Filesystem

This provides an extension to the regular `/proc` file system and gives details of USB devices attached to your system and USB drivers in use.

#### 4.1.2 UHCI/UHCI Alternate (JE)/OHCI

You will need to use one of these drivers dependant on your USB chipset. Use the kernel configuration help text in order to choose the correct one. If you are not sure, select them all. Once you have got things working, you will be able to tell which driver is actually being used.

#### 4.1.3 USB Mass Storage Support

This module will enable the use of the USB mass storage protocol.

### 4.2 Support for Hot-Pluggable Devices

This will allow the detection of devices that are plugged in to or removed from the system while it is up and running. This is how I imagine most people would want to use a camera device (but hey, if you want to plug in and reboot, be my guest ;-))

### 4.3 SCSI Support

Hey, where did this come from? The USB mass storage device driver actually emulates a SCSI disk. Therefore SCSI drivers are required in order to make use of it. Enable the following SCSI elements:

- SCSI Disk Support
- SCSI Generic Support

## 4.4 DOS Fat FS Support

Your camera will present the contents of the smartmedia as being a "FAT" (ie Windows style) file system, or at least, I am fairly sure that that will be the case. Therefore FAT support is required to get at the contents. Enable the following FAT elements:

- MSDOS FS Support
- VFAT (Windows-95) FS Support

## 5 Non Kernel Set Up

### 5.1 Hotplug User Space Daemon

This is required to complement the hotplug kernel module in providing hotplug support. Install the correct package for your distribution. In the case of Debian, you can do it like this:

```
apt-get install hotplug
```

This installs the daemon and creates the necessary boot scripts.

### 5.2 /etc/fstab

Insert the following line after the proc line in your /etc/fstab file (you will need a text editor such as *pico* or *vi* to do this):

```
none /proc/bus/usb usbfs defaults 0 0
```

This will ensure that the USB extension to the proc file system is mounted at boot up.

### 5.3 Create a Mount Point

You will need to create a suitably named mount point directory for your camera device, for example "/mnt/camera". To do this, type

```
mkdir /mnt/camera
```

## 6 Ok, Let's Get Those Pics

Once you have all the elements in place, you have booted your system using your new kernel with your new /etc/fstab file, and your hotplug daemon is up and running; you are ready to try to retrieve some images from your camera.

Follow the instructions in your camera's manual for attaching the camera to your PC as a USB mass storage device. Check to see if this has been detected by typing:

```
cat /proc/bus/usb/devices
```

Hopefully, you will see an entry for the usb mass storage device looking something like this:

```
T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=01 Dev#= 3 Spd=12 MxCh= 0
D:  Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1
P:  Vendor=04cb ProdID=0100 Rev= 1.00
S:  Product=USB Mass Storage
S:  SerialNumber=Y-301~~~~~130523W0FC0004051429
C:* #Ifs= 1 Cfg#= 1 Atr=c0 MxPwr= 0mA
I:  If#= 0 Alt= 0 #EPs= 3 Cls=08(stor.) Sub=05 Prot=00 Driver=usb-storage
E:  Ad=81(I) Atr=02(Bulk) MxPS= 64 Iv1=0ms
E:  Ad=02(O) Atr=02(Bulk) MxPS= 64 Iv1=0ms
E:  Ad=83(I) Atr=03(Int.) MxPS= 8 Iv1=1ms
```

If all is well, attempt to mount the camera. This is done using a SCSI disk device file. In the example below, “sda1” is used. If you actually have some real SCSI disks, or something else posing as a SCSI disk, you will need to modify this accordingly (ie if you have two SCSI disks, it will be “sdc1”). Also, substitute your actual mount point directory where I have used “/mnt/camera”. Here is what your mount command might look like:

```
mount -t vfat /dev/sda1 /mnt/camera
```

If all is well, you will find your image files under your mount point directory. There may be several further directory layers before you get to the image files proper, but once you have located them, you can copy them to a destination of your choosing.

Once you have finished copying images from your camera, unmount the file system by typing something along the lines of (remember to adjust it for the actual SCSI disk device file being used):

```
umount /dev/sda1
```

Follow the instructions in your camera’s manual for disconnecting it from your PC. Again, a check of /proc/bus/usb/devices will confirm that the removal has been detected (the entry for the USB mass storage device will be gone).

## 7 Troubleshooting

If everything works first time, give yourself a pat on the back.

If there is a failure at any point during the image retrieval process with no obvious reason why, try looking at your log files for error messages (the exact file names depend on how your system is configured, but /var/log/messages and /var/log/syslog will probably be good bets). If you cannot interpret any error messages, try posting them on the LJR forum.

## **8 Comments, Corrections, Devastating Criticism etc.**

If there is anything here that could be done better or is plain inaccurate, please let me know.

My current email address can be found in my profile, or you can PM me if you are a LJR member.

Thanks

Paul