

Zip® Drives

Jason Bacon

acadix@execpc.com

FreeBSD is a registered trademark of the FreeBSD Foundation.

Adaptec is a registered trademark of Adaptec, Inc.

Iomega, Zip, and Jaz are either registered trademarks or trademarks of Iomega Corporation in the United States and/or other countries.

Microsoft, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Motif, OSF/1, and UNIX are registered trademarks and IT DialTone and The Open Group are trademarks of The Open Group in the United States and other countries.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and the FreeBSD Project was aware of the trademark claim, the designations have been followed by the “™” or the “®” symbol.

1 Zip Drive Basics

Zip® disks are high capacity, removable, magnetic disks, which can be read or written by ZIP drives from IOMEGA corporation. ZIP disks are similar to floppy disks, except that they are much faster, and have a much greater capacity. While floppy disks typically hold 1.44 megabytes, ZIP disks are available in two sizes, namely 100 megabytes and 250 megabytes. ZIP drives should not be confused with the super-floppy, a 120 megabyte floppy drive which also handles traditional 1.44 megabyte floppies.

IOMEGA also sells a higher capacity, higher performance drive called the Jaz®/JAZZ drive. Jaz drives come in 1 gigabyte and 2 gigabyte sizes.

ZIP drives are available as internal or external units, using one of three interfaces:

1. The SCSI (Small Computer Standard Interface) interface is the fastest, most sophisticated, most expandable, and most expensive interface. The SCSI interface is used by all types of computers from PC's to RISC workstations to minicomputers, to connect all types of peripherals such as disk drives, tape drives, scanners, and so on. SCSI ZIP drives may be internal or external, assuming your host adapter has an external connector.

Note: If you are using an external SCSI device, it is important never to connect or disconnect it from the SCSI bus while the computer is running. Doing so may cause file-system damage on the disks that remain connected.

If you want maximum performance and easy setup, the SCSI interface is the best choice. This will probably require adding a SCSI host adapter, since most PC's (except for high-performance servers) do not have built-in SCSI support. Each SCSI host adapter can support either 7 or 15 SCSI devices, depending on the model.

Each SCSI device has its own controller, and these controllers are fairly intelligent and well standardized, (the second 'S' in SCSI is for Standard) so from the operating system's point of view, all SCSI disk drives look about the same, as do all SCSI tape drives, etc. To support SCSI devices, the operating system need only have a driver for the particular host adapter, and a generic driver for each type of device, i.e. a SCSI disk driver, SCSI tape driver, and so on. There are some SCSI devices that can be better utilized with specialized drivers (e.g. DAT tape drives), but they tend to work OK with the generic driver, too. It is just that the generic drivers may not support some of the special features.

Using a SCSI zip drive is simply a matter of determining which device file in the `/dev` directory represents the ZIP drive. This can be determined by looking at the boot messages while FreeBSD is booting (or in `/var/log/messages` after booting), where you will see a line something like this:

```
da1: <IOMEGA ZIP 100 D.13> Removable Direct Access SCSI-2 Device
```

This means that the ZIP drive is represented by the file `/dev/da1`.

2. The IDE (Integrated Drive Electronics) interface is a low-cost disk drive interface used by many desktop PC's. Most IDE devices are strictly internal.

Performance of IDE ZIP drives is comparable to SCSI ZIP drives. (The IDE interface is not as fast as SCSI, but ZIP drives performance is limited mainly by the mechanics of the drive, not by the bus interface.)

The drawback of the IDE interface is the limitations it imposes. Most IDE adapters can only support 2 devices, and IDE interfaces are not typically designed for the long term. For example, the original IDE interface would not support hard disks with more than 1024 cylinders, which forced a lot of people to upgrade their hardware prematurely. If you have plans to expand your PC by adding another disk, a tape drive, or scanner, you may want to invest in a SCSI host adapter and a SCSI ZIP drive to avoid problems in the future.

IDE devices in FreeBSD are prefixed with an `a`. For example, an IDE hard disk might be `/dev/ad0`, an IDE (ATAPI) CDROM might be `/dev/acd1`, and so on.

3. The parallel port interface is popular for portable external devices such as external ZIP drives and scanners, because virtually every computer has a standard parallel port (usually used for printers). This makes things easy for people to transfer data between multiple computers by toting around their ZIP drive.

Performance will generally be slower than a SCSI or IDE ZIP drive, since it is limited by the speed of the parallel port. Parallel port speed varies considerably between various computers, and can often be configured in the system BIOS. Some machines will also require BIOS configuration to operate the parallel port in bidirectional mode. (Parallel ports were originally designed only for output to printers)

2 Parallel ZIP: The `vpo` Driver

To use a parallel-port ZIP drive under FreeBSD, the `vpo` driver must be configured into the kernel. Parallel port ZIP drives also have a built-in SCSI controller. The `vpo` driver allows the FreeBSD kernel to communicate with the ZIP drive's SCSI controller through the parallel port.

Since the `vpo` driver is not a standard part of the kernel (as of FreeBSD 3.2), you will need to rebuild the kernel to enable this device. The process of building a kernel is outlined in detail in the FreeBSD Handbook (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/kernelconfig.html). The following steps outline the process in brief for the purpose of enabling the `vpo` driver:

1. Run `/stand/sysinstall`, and install the kernel source code on your system.

2. Create a custom kernel configuration, that includes the driver for the vpo driver:

```
# cd /sys/i386/conf
# cp GENERIC MYKERNEL
```

Edit MYKERNEL, change the `ident` line to MYKERNEL, and uncomment the line describing the vpo driver.

If you have a second parallel port, you may need to copy the section for `ppc0` to create a `ppc1` device. The second parallel port usually uses IRQ 5 and address 378. Only the IRQ is required in the config file.

If your root hard disk is a SCSI disk, you might run into a problem with probing order, which will cause the system to attempt to use the ZIP drive as the root device. This will cause a boot failure, unless you happen to have a FreeBSD root file-system on your ZIP disk! In this case, you will need to “wire down” the root disk, i.e. force the kernel to bind a specific device to `/dev/da0`, the root SCSI disk. It will then assign the ZIP disk to the next available SCSI disk, e.g. `/dev/da1`. To wire down your SCSI hard drive as `da0`, change the line

```
device da0
to
disk da0 at scbus0 target 0 unit 0
```

You may need to change the target above to match the SCSI ID of your disk drive. You should also wire down the `scbus0` entry to your controller. For example, if you have an Adaptec® 15xx controller, you would change

```
controller scbus0
to
controller scbus0 at aha0
```

Finally, since you are creating a custom kernel configuration, you can take the opportunity to remove all the unnecessary drivers. This should be done with a great deal of caution, and only if you feel confident about making modifications to your kernel configuration. Removing unnecessary drivers will reduce the kernel size, leaving more memory available for your applications. To determine which drivers are not needed, go to the end of the file `/var/log/messages`, and look for lines reading “not found”. Then, comment out these devices in your config file. You can also change other options to reduce the size and increase the speed of your kernel. Read the section on rebuilding your kernel for more complete information.

3. Now it is time to compile the kernel:

```
# /usr/sbin/config MYKERNEL
# cd ../../compile/MYKERNEL
# make clean depend && make all install
```

After the kernel is rebuilt, you will need to reboot. Make sure the ZIP drive is connected to the parallel port before the boot begins. You should see the ZIP drive show up in the boot messages as device `vpo0` or `vpo1`, depending on which parallel port the drive is attached to. It should also show which device file the ZIP drive has been bound to. This will be `/dev/da0` if you have no other SCSI disks in the system, or `/dev/da1` if you have a SCSI hard disk wired down as the root device.

3 Mounting ZIP disks

To access the ZIP disk, you simply mount it like any other disk device. The file-system is represented as slice 4 on the device, so for SCSI or parallel ZIP disks, you would use:

```
# mount_msdos /dev/da1s4 /mnt
```

For IDE ZIP drives, use:

```
# mount_msdos /dev/ad1s4 /mnt
```

It will also be helpful to update `/etc/fstab` to make mounting easier. Add a line like the following, edited to suit your system:

```
/dev/da1s4 /zip msdos rw,noauto 0 0
```

and create the directory `/zip`.

Then, you can mount simply by typing

```
# mount /zip
```

and unmount by typing

```
# umount /zip
```

For more information on the format of `/etc/fstab`, see `fstab(5)`.

You can also create a FreeBSD file-system on the ZIP disk using `newfs(8)`. However, the disk will only be usable on a FreeBSD system, or perhaps a few other UNIX® clones that recognize FreeBSD file-systems. (Definitely not MS-DOS® or Windows®.)