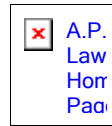


Search
FAQ
New to Sco
New to Unix
New Here
SCO News
Unix Articles
Unixware
Linux
Security
Code
Tests
Links
Books
Reviews
Opinion
Find a Consultant
Find a Job
Site Forum
Donations
Contact Info
Services
Sales
Consulting Rates
Advertising Rates
Site Map



A.P. Lawrence Home

Over 10,600 Unix/Linux users so far this month (48,400+ page views)

<http://www.microlite.com>

[More Articles](#)

SCSI

I know I'm ANSI, but am I SCSI? - anonymous customer requesting information about her hardware configuration

Thanks to Bill Vermillion for comments and corrections to this article.

The original SCSI was an 8 bit data bus running at 5 MHz which gave it a maximum transfer rate of 5 Megabytes per second. The 5MB/s is what I've seen in numerous places, but Bill Vermillion points out that other sources have it differently:

I grabbed a couple of books I have on SCSI - one is the Adaptec book Understand I/O Subsystems which has a lot of stuff in it.

I've just noted a minor discrepancy between the books so I'll note that.

You indicated that SCSI (the first one was a 5MB/sec rate). I recalled that it was 3.5MB. Adaptec's book puts it at 3MB/sec and the Book of SCSI puts it at 4MB second.

SCSI-2 is shown as 5MB/sec for both. ISTR - though I'm not going to look it up - as that was for synchronous transfer only and the asynch was still 3.5MB. That is where I may have remembered 3.5 as that is an option on Adaptec 154x series.

Fast SCSI (SCSI-2) doubled that to 10 MB/s by doubling the bus to 10 MHz. Then we got Wide Scsi added, which could be either 16 bit or 32 bit (but is usually 16- I'm not even sure anyone has done a 32 bit version yet), which gave us speeds of 20 or 40 MB/s. Then came SCSI-3, which can burst up to 80 MB/s on a 16 bit bus. And now we have Ultra160- 160 MB/s.

Unfortunately, there are other terms: "Fast-40", "Fast-80", "Ultra", "Ultra2" and more. A good chart that sorts some of this out (and shows you what the connectors look like) is <http://www.seagate.com/support/kb/disc/scsirev.html>.

Understanding SCSI

If all you have is devices with the same speed, then all you need to understand is termination and ID's. You don't need to grok low voltage differential vs. high voltage differential; you don't really even need to understand differential (it just means that signals are determined by the difference between a positive and a negative voltage source rather than by a fixed voltage level; it's more immune to noise). If all you understand is termination and ID's, you can successfully install SCSI devices. The rules for termination are simple: each end of a SCSI chain needs to be terminated. The rules for ID's are pretty simple, too, but there are gotcha's in all of this:

Termination

Normally this is simple: you have a cable running from the SCSI adapter to one or more devices, so you terminate the last one. The adapter often has "automatic" termination: it terminates itself if it is the other end of the chain and doesn't if it is in the middle (such as when you've attached external devices). A terminator is actually a resistor that absorbs signals so that they don't bounce back and confuse devices on the bus. As a resistor, it also affects current flow on the bus, which is why you can't terminate more than one device: if you do, you have resistors in parallel, which actually lowers resistance to current, and makes things worse. Adaptec likens it to a tin-can and string "telephone" where each end is taught (terminated) so that vibration (signal) can pass from one can to the other. An extraneous terminator in the middle "tightens" the string there, and cuts off the signal that should get to the other end. However you want to visualize this, remember that excess termination is as bad as no termination (with low speed devices, you often can "get away" with one unterminated device. You **shouldn't** do that, but it's surprising how many working single drive systems I find like that!).

It's a far better idea to use a terminator that attaches to the cable rather than terminating devices. It makes it easier to move stuff around or add or remove devices; your termination is always on the cable, and all your devices are always unterminated. You are also apt to get better termination that way, especially if your final device is a low-end tape drive. Terminators should be ACTIVE, which means that they contain a voltage regulator and don't have to depend on the +5 volts that they are supposed to have being dead on accurate. Even if they weren't any betterer than using one of the drives, it's still a better idea because you can replace drives, move them around or add new without worrying about where the terminated device is. Active terminators are of course more expensive. Your data, of course, is pretty much worthless, right? Pop for the good stuff.

Active terminators need termination power. That means that one or more of your devices needs to provide it (by setting a jumper or dip switch). Don't confuse **providing** termination power with termination- you won't be setting the termination jumper, just the one that provides termination power. Not having this set leads to strange problems- on one of my home systems, a backup would work and even finish, but part way through the hard drive would become (and remain) unavailable for anything else- effectively locking up the machine: see [Installing 5.06](#) for details.

What about these newer adapters with multiple internal connectors? With some cards, these are to give you a separate bus so that you can have more devices; with others it's the same

bus but designed for older, lower speed, devices. But with regard to termination, each of those connectors is a separate chain, so the other end of that chain must be terminated. Suppose you have 40 MB/s hard drives running off one of those connectors, and have cabled a tape drive off the other. You need termination at both places: the hard drive chain and the tape drive chain.

The issue of mixing devices gets covered below, but consider what happens if you have an older 8 bit device attached (by way of an adapter) to a 16 or 32 bit cable? That adapter needs to block the high bits of the cable, either by termination or just by physical bypass, and obviously it's very important for you to understand which it is doing.

If it's terminating the high bits, you can't stick that narrow device in the middle of the chain, unless you have a terminator at the end that only terminates the high bits. If it bypasses, so that only the low bits go to the narrow device, then you can (but you shouldn't, see below). If it's a bypass, then terminating the 8 bit device is NOT going to terminate the high bits, so you don't want that; it would be better to leave it unterminated and terminate the cable.

Adaptec has a nice page that illustrates this at <http://www.adaptec.com/support/configuration/connect.html>.

Be careful here in distinguishing between ribbon cables, which are essentially a parallel path, and external devices, which are usually cabled one to another. An adapter that blocks the high bits doesn't prevent access to wide devices farther along a ribbon cable, but for external devices it certainly could. Termination problems are essentially the same.

Adaptec (for example) sells both of these types of adapters: see [Adaptec Support](#). However, when we get to their Ultra2 adapters, you are advised not to do this at all:

The SCSI Card 2940U2W features two bus segments - one for Ultra2 SCSI and one for Ultra SCSI. If an Ultra SCSI device is attached to the Ultra2 SCSI segment, the entire Ultra2 SCSI segment, and any attached peripherals, will default to Ultra SCSI speeds. If an Ultra2 SCSI device is attached to the Ultra SCSI bus segment, it will function, but will default to Ultra SCSI speed.

(from <http://www.adaptec.com/technology/overview/u2devisolation.html>).

You have to think about the whole path that data takes when thinking about SCSI speed. Let's say you have a brand new Ultra 160 controller card. If you had 5 40 MB/s drives in a [RAID](#) configuration, you've got a theoretical possibility for 200 MB/s coming from the drives- more than the adaptor can handle. Not only that, but how fast is your bus? And what else is it handling at the same time? Here's a post from Robert Lipe that addressed some of that:

```
From: Robert Lipe <robertlipe@usa.net>  
Newsgroups: comp.unix.sco.misc  
Subject: Re: Is U160 REALLY Faster than Ultra-Wide on SCO?  
Date: Sat, 29 Apr 2000 19:31:21 GMT
```

```
Jeff Liebermann <jefffl@comix.santa-cruz.ca.us> writes:
```

```
>On Sat, 29 Apr 2000 15:33:18 GMT, Robert Lipe <robertlipe@usa.net> wrote:
```

>>I'll basically support Jeff's answer, but pick on a few minor nits.
>Nobody ever agrees with me. I'm not sure I know how to handle agreement.

I'll pick a fight if it makes you feel better, but I'd prefer to get back to the technical stuff pretty quickly.

Your truck is funny looking. :-)

>It was my impression (i.e. I wasn't paying attention) that 64bit PCI
>requires that every card on one side of the PCI bridge chip also be 64bit.
>Shove a junk video card on the same bus, and say goodbye to 64bit transfers.

I'm looking at MindShare book on PCI 2.1. The specification goes to great detail to allow independent autonegotiation of 64 bit address, 64 bit data. So (modulo chipset bugs and implementation lameness) I don't think that plugging in a 32 bit card onto your 64 bit bus will neuter the 64-bit capabilities of the rest of the system.

Of course, it is a shared medium. So if you have a PCI bus that's largely saturated by a 32 bit card, the reality is that your 64 bit card will be penalized by being forced to negotiate for the bus more often.

It's not entirely unlike having a 10Mb laser printer on your 100Mb ethernet network. The time your printer is on the wire, your 100Mb stuff can't be on the wire. It isn't like your 100Mb devices will all slow down to 10 just because there happens to be one thing on the wire that's slower. So you try to architect the system so that your biggest consumers of bandwidth use the fastest possible scheme and the slower consumers - be they PCI or Ethernet - stay out of the way as much as possible.

>I also vaguely recall that scatter gather (basically DMA block transfers
>from ram directly to the SCSI adapter) also requires 64 bit OS support.
>Even if 32/64bit PCI mode mixing were possible, running at 32bits part of
>the time would certainly be a big performance hit. So, how wrong am I?

64 bit addressing and 64 bit data are independent. Both of those are independent of whether the host processor is 64 bits. But both do need host driver (and some host OS) support.

On an OS like OpenServer where 4Gb of RAM is, errr, uncommon, 64 bit addressing is somewhat pointless. But 64-bit data transfers can still be effectively used if the host drivers carefully craft the DMA setups. (I don't know which drivers do or don't in OSR5; I'm speaking only that it might be done.)

>Also, it was my understanding that RAID striping is limited by the sustained
>transfer rate of the drives. If one drive of the stripe decides to

At some level of RAID, I'm sure this is true. You have to get that data onto a spinning round thing some time. If you build a RAID stripe out of crappy drives, you're still hosed. Drives that can sustain 25-50MB/sec of data aren't uncommon.

>re-calibrate or flush its cache, then the other drives automagically slow
>down until it catches up. That's why I used the sustained transfer rates
>instead of the burst rates in my guesstimates. Besides, the burst rate seems
>to be mostly determined by the size of the hard disk ram cache. Once the
>cache is full, it's back to real sustained transfer rates.

It's a matter of having more options to move around bottlenecks. Depending on the system and the load pattern, the bottleneck might be in any of:

- A) Host PCI saturation
- B) SCSI bus saturation
- C) burst drive saturation
- D) sustained drive I/O saturation

If your host PCI bus is soaked because you're running xico on some piece of crap video card with the vesa server all the time, don't expect swapping in one of these new U160 cards to help.

But if you're bound on item B and think that you can keep the HBA full of incoming data and drained of outgoing data on both ends, then U160 is a help.

SCSI ID

With the older devices, your ID could be 0 to 7. The host adapter was always ID 7, so that left you 0 to 6. Most PC operating systems wanted ID 0 as the boot drive (and most still do). HP machines would boot from the highest ID, Sun would boot from 3 but pretend it was 0, and so on. With SCO, there was no question: you booted from ID 0.

Wide devices now have ID's 0 to 15. The host adapter still usually gets the 7 slot, and it's still easier to set the boot disk to 0 for SCO and most other OS's.

The ID is usually set with jumpers or switches. These are often labeled (on the narrow devices) ID0, ID1, ID2. The values are binary; if ID0 is jumpered or turned on that's a "1", if ID1 is on that's "2", and if ID2 is on that's "4". By combining jumpers you can set any value from 0 to 7. For example, to get ID 5, you'd jumper ID0 and ID2. To get 3, you'd jumper ID0 and ID1, and for 6 you'd jumper ID1 and ID2.

I've only seen this once, but I had a SCSI drive where the jumpers needed to be REMOVED to set the bit! Therefore, to get ID 2, you'd leave jumpers on ID0 and ID2, but remove the one on ID1! That's extremely unusual!

Also watch out for drives that let you set the ID in more than one place. Sometimes this is done for convenience so that one set can be reached from the side and one from the bottom. Use one or the other, not both, because in most drives the effect is cumulative: if ID0 is jumpered at one location and ID2 is set at the other, your ID will be 5

With the newer drives, you have an extra jumper with a value of 8, which gives you the 0-15 range.

Some high end machines have a SCSI backplane where drives plug in and it's the position of the drive that determines its ID, either through jumpers on the backplane or software in the BIOS. These are usually combined with hardware raid (see [RAID](#)) and often are hot-pluggable, which might mean that you can pull out a bad drive from a RAID array and pop in a fresh one without taking the machine down.

Although that may be the case, you may still be required to bring the machine down to initiate an array rebuild; this is all dependent on hardware and software, so read the docs carefully- don't assume the array will rebuild without manual intervention.

Some Standards

Keep in mind that these are **SCO** standards, or perhaps more accurately, expectations:

- Boot Drive: ID 0
- Tape Drive: ID 2
- SCSI CDROM: ID 6

Can you use other ID's? With modern versions, you can generally use whatever you want for tape or cdrom. SCO still very much wants to install and boot from ID 0; however, if you really need to convince it otherwise, you can. This newsgroup post by Bill Vermillion explains it very well (included by kind permission of Bill):

```
Xref: world comp.unix.sco.misc:86166
Newsgroups: comp.unix.sco.misc
Path: world!newsswitch.lcs.mit.edu!newsfeed.cwix.com!207.172.3.37!feed1.news.rcn.
From: bill@amp;bilver.magicnet.netREMOVETHIS (Bill Vermillion)
Subject: Re: Would like to load OS 5.0.5 on non-primary disk
Organization: W.J.Vermillion - Orlando / Winter Park
Message-ID: <F4H4vI.30p@amp;bilver.magicnet.netREMOVETHIS>
References: <pt%f2.655$rb.2278442@amp;news1.midsouth.rr.com>
Date: Thu, 24 Dec 1998 14:42:54 GMT
```

In article <pt%f2.655\$rb.2278442@amp;news1.midsouth.rr.com>, Jared Tugwell <jtugwell1SPAMKILLER@amp;midsouth.rr.com> wrote:

```
>The disk that is my current system disk is 1GB and has a SCSI ID of
>0. I have a second 1GB disk that has some filesystems on it that is
>used by the current OS and has a SCSI ID of 1. The new disk I just
>added is 4.3GB and has a SCSI ID of 4.
```

```
>1. What makes a disk THE primary disk, it's SCSI ID?
```

The way the PC architecture works is that the first disk found is the C: for DOS, or the boot for all other systems.

To make a system install and boot from an ID other than 0 (or from 6 if you are in an IBM environment) is to have the first disk the scsi adapter finds be the boot disk.

Some SCSI adapters will let you pick the HD from which to boot. Others will require you to go into the adapter settings and turn off the 'scan for bios' for each drive you wish to ignore, and have the drive you want to use to boot be the first one that has the scan for bios.

```
>2. During the install of 5.0.5 it seemed to only want to load to
>the current system disk and wouldn't let me choose which hard disk
>I wanted to use. I really want to leave everything as intact as
>possible, how can I accomplish this?
```

Since the first disk the system sees on a SCSI adapter is the one

from which you perform the lowest level boot, that disk will always be the one where the lowest level boot is performed - no matter what OS you are using.

Once you can see your preferred drive displayed as the first drive the adapter sees upon power up - you need to be sure the SCSI device displays this - then you are ready to begin.

If you just install at that point, all will appear to go well up to a point. However you'll never boot that OS.

If, for example, you were going to boot from SCSI ID5, you'd have scan for bios turned off on units 0 thru 4. You'd put in the NO boot disk, and then do something similar to this:

```
boot: defbootstr Sdsk=blc(0,0,5,0)
```

The Sdsk defined the scsi disk
The blc in this examples defines a buslogic controller. You would substitute the proper identifying letter for blc, such as alad or arad for example. In the parentheses the first 0 defines which adapter you are using. The second zero is the channel on the scsi adapter, as you will find adapters that have up to three channels on them - giving up to 45 HDs. (I have heard there are 5 channel adapters but do not know what they might be). The third number, 5 in the above example, is the SCSI ID of the drive. The last is the LUN (logical Unit number), just leave it at zero. (No need to discuss this here as you need special devices for that - which could give you over 300 devices on one adapter).

Once you enter the appropriate string above, all should install easily and you will boot normally. To boot from the original drive just put scan for bios back on in that adapter.

>3. Even if I can somehow use this 4.3GB disk, is there some issues >with the 1024 cylinder limitation that I must be aware of?

I'd recommend having the / partition by about 750MB, and the rest in /u or whatever you wish to call it.

BTW - don't believe the often heard/written advice that you can only boot from device 0. The true definition is that you boot from the first device the adapter sees, which it defines as device zero (80h in the bios report), and it matters not what it's ID may be.

I think this could be considered for the FAQ, as it gets asked often enough.

--

Bill Vermillion bv @ wjv.com

Mixing SCSI and IDE

This has always been possible, but until very recently the BIOS of the machine itself would

insist on booting from the IDE drive. More modern machines have the capability to be told to boot from SCSI even if IDE is installed. If you don't need the IDE drive to be part of your SCO system, that type of BIOS could let you have SCO on a SCSI drive and something else on the IDE. However, mixing the two isn't going to work, at least not yet, because there are built-in assumptions within the OS that assume that if an IDE drive is present, it MUST be the boot drive.

This has apparently been worked out, at least on 5.0.5 and up: it has been reported that using a "defbootstr hd=Sdsk" will work (of course the machine itself has to be capable of selecting the boot drive). I have no ability to confirm this, but it came from a reputable source (John Gray)

You can trick it by booting from floppy and then switching to the hard drive. See [Bootting OSR5](#)

The following details came from a post by Bela Lubkin:

```
Subject: Re: Workaround for installing SCO on secondary drive needed
From: Bela Lubkin
Date: 1997/04/21
Newsgroups: comp.unix.sco.misc
```

[this is one of those messages-to-save...]

Rolin Nelson wrote:

```
> I currently have two hdd, a primary 2Gig IDE with Win95 and a 4Gig SCSI
> with SCO5.0.2 and NT4.0. I am able to boot Win95 and NT4.0 from the NT
> OS Loader that was added by NT4.0 to the Win95 installation. However,
> because SCO is not on my primary drive I am unable to activate its
> partition to make it boot. If I unplug the IDE cable or set primary hdd
> to NONE in CMOS then it boots.
>
> I've purchased System Commander and it did not help because that is one
> of its limitation. It listed SCO as one of the Oses that must reside on
> a primary hdd in order to boot. I don't know if this is a SCSI
> controller limitation or BIOS, I cannot make my SCSI a primary if there
> is an IDE hdd installed.
```

It is a BIOS limitation dating back to when hard disks were new to IBM PCs. IDE drives look the same to the BIOS as those early ST506 and ESDI drives, and are always assumed to be the boot device. I think there are finally starting to be some BIOSes which allow you to override this, but they aren't common -- and there's no guarantee that Oses will understand such a configuration anyway.

```
> Bela Lubkin suggested that the following might be possible:
> < This subject has been discussed dozens of times on this newsgroup. It
> < is possible to set your system up so that OpenServer boots mostly from
> < the second hard disk, but it will require at least a small Unix
> < partition on the first drive. If the first hard drive is already
> < entirely dedicated to Win95 then you've got a problem (although
> < apparently you can "safely" shrink a DOS partition with "Partition
> < Magic").
> <
> < If you had room, you could make a 16MB Unix partition on the IDE drive;
> < make a 15MB /stand division in that partition; copy the existing /stand
> < materials there; change some boot strings and mount options; and it
```

```
> < would work.
>
> Using the Partition-It disk utility software I was able to resize my hdd
> to make room for the 15MB /stand directory. What I need now are
> instructions to copy the /stand on SCSI drive to stand partition on IDE.
> And, what options in Default Boot String to change to boot from IDE hdd.
```

Thank you for the clearly stated question. This really helps me to frame a good answer.

It turns out that the answer is **very** complex. It is definitely possible, but it takes many steps which depend on each other. Following is a discussion of the steps. I will probably have made some mistakes; you need to be prepared to improvise along the way.

=====

"I installed OpenServer on a SCSI drive, then added an IDE drive to run DOS/Windows. How can I make a new /stand on the IDE drive and use it as a staging area to boot OpenServer from the SCSI drive?"

This can be done. The process is complex. The following applies, modulo any errors, to OpenServer Release 5.0.0 through 5.0.4. Similar techniques could be applied to older SCO operating systems (probably all the way back to Xenix), but details vary, you would have to figure out how to implement the steps yourself.

Read the whole thing first, then read it again. Then start.

Send corrections (with enough context that I know what the heck you're talking about) to belal@sco.com.

A rough outline is:

1. Prepare your kernel to be able to access IDE drives.
2. Prepare your device nodes to be sane when an IDE drive is present.
3. Boot the new arrangement with the IDE drive not present.
4. Boot the new arrangement with the IDE drive present, using boot floppy.
5. Partition, divvy and mkfs the new /stand filesystem on the IDE drive.
6. Copy old /stand contents to new /stand.
7. Modify new /stand /etc/default/boot appropriately.
8. Change the /stand mounting arrangements.
9. Try to boot.

So, in some detail:

1. Prepare your kernel to be able to access IDE drives.

At this point you are working in a pure-Unix environment, with the IDE drive physically disabled (cable disconnected).

The first step is to add the "wd" driver into your kernel, if it isn't already. (It will be present if you used an ATAPI CD-ROM to install.) Edit /etc/conf/sdevice.d/wd; change "N" to "Y". If it was already "Y" then the driver was already linked in. If not, first save a good copy of the kernel, then relink:

```
# btmnt -w          (mount /stand read-write)
# cd /stand
# cp unix unix.good
```

```
# cd /etc/conf/cf.d
# ./link_unix -y
```

Reboot and verify that the new kernel boots. Don't lose track of that unix.good; we may need it later.

2. Prepare your device nodes to be sane when an IDE drive is present.

This involves editing files in /etc/conf/node.d. The kernel link process does a directory search in node.d and uses all the files it finds, so be sure your editing procedure does not leave backup files in the directory. You need to copy the existing "hd" file as "Sdsk" and change all of the major device references (first column) from "hd" to "Sdsk". Then empty the "hd" file, leaving it empty but still present.

For example, a line:

```
hd hd00 b 0 sysinfo sysinfo 600
```

should currently exist in /etc/conf/node.d/hd. You will change it to:

```
Sdsk hd00 b 0 sysinfo sysinfo 600
```

in /etc/conf/node.d/Sdsk. One way to do this:

```
# cd /etc/conf/node.d
# sed 's/hd/Sdsk/' hd > Sdsk # no "/g" - only want to change 1st "hd"
# echo "" > hd
```

Also edit /etc/conf/cf.d/sassign, again changing "hd" to "Sdsk".

3. Boot the new arrangement with the IDE drive not present.

Now relink the kernel and reboot. It's important to verify that things are still sane. Once rebooted, run `fdisk` and `divvy`, just to verify their sanity. Also run `fsck -n -o full /dev/root` and let it run out. This may give you a few minor errors, since the root filesystem is active. That's OK. What you don't want to see is some sort of massive failure as if fsck doesn't believe /dev/root is a sane filesystem at all.

4. Boot the new arrangement with the IDE drive present, using boot floppy.

For this to work, there must be only one IDE hard disk in the system. There can be ATAPI CDs or tapes, but only one IDE hard disk. If you have others, temporarily remove them and make sure the primary's jumpers are set correctly. Some IDE drives have to be jumpered differently for master/slave-present and master/no-slave. The SCSI disk that contains Unix must be the first disk recognized by the host adapter BIOS. When these conditions are met, the SCSI host adapter's BIOS will put its drive into the BIOS's drive table as if it were the 2nd IDE drive.

Once that's true, you can boot your OpenServer boot floppy as follows:

```
Boot
: hd(104)unix root=Sdsk(42) swap=none dump=none
  ^^^^^^^^^^^ ~~~~~
```

Here you're using only /boot from the floppy. The kernel comes from your SCSI disk. The portion of the boot string labeled "^^^^" is interpreted by /boot, in terms of BIOS devices. hd(104) is the 0th

division of the active partition of the 2nd hard disk. The 2nd hard disk is your SCSI disk, as discussed above. The portion labeled "~~~~" is interpreted by /unix, in terms of Unix devices. Sdsk(42) is the 2nd division of the active partition of the 1st SCSI hard disk; which should be where your root filesystem is found.

Do not omit the "swap=none dump=none" portion. Some of these things could lead to a panic, and you don't want the system to write a panic dump onto a device node that was pointing to something unexpected (like your DOS filesystem)...

I might be a little off the mark. This may load the kernel but panic with a message that it's unable to mount the root filesystem (probably a "srmountfun" message). If so, the problem can probably be solved; report back with the exact message and exactly what you had been doing at the time (e.g. whether the IDE drive is enabled, what you had done to the kernel, etc.)

Once you are able to boot this way, go to single-user mode.

5. Partition, divvy and mkfs the new /stand filesystem on the IDE drive.

I think you can't use `mkdev hd` for this task, because it will state that since your root is SCSI, you can't add an IDE drive. So you must do it manually. I'm fuzziest about this step. I think you can do something like copy /etc/conf/node.d/Sdsk to /etc/conf/node.d/wd and make similar changes to before -- Sdsk becomes wd -- and also change all references from "hd0" to "hd1"; from "0s" to "1s". e.g.

```
Sdsk hd00 b 0 sysinfo sysinfo 600
Sdsk rdsk/0s0 c 0 sysinfo sysinfo 600
```

would become:

```
wd hd10 b 0 sysinfo sysinfo 600
wd rdsk/1s0 c 0 sysinfo sysinfo 600
```

You want to copy-and-modify all of the hd*, rhd*, dsk/0s*, rdsk/0s* entries; omit any others. When you're done, you should have an empty hd file, the same Sdsk file as created in step 2, and a new wd file based on Sdsk, but every line modified.

Once that's done, relink the kernel again. You don't need to reboot; this is just to build the device nodes and I'm too lazy to look up the right incantation to do that by itself.

Once that's all done,

```
# fdisk -f /dev/rhd10
```

and create a Unix partition, "use rest of disk";

```
# divvy /dev/hd1a
```

just to make sure it complains "no division table"; then

```
# divvy -m /dev/hd1a
```

to create one. Have it create 1 division, then go in to make changes, name it "boot2", tell it to create an EAfs filesystem (it will have

defaulted to HTFS, which won't work for a boot filesystem). Quit, install the new table, let it make the filesystem.

6. Copy old /stand contents to new /stand.

Now you can:

```
# mount /dev/boot2 /mnt
# cd /stand
# find . | cpio -pdmvu /mnt
```

7. Modify new /stand /etc/default/boot appropriately.

You need to change the DEFBOOTSTR in the new (/mnt)/etc/default/boot. It will say "hd(40)unix root=hd(42) ...". The first part is fine, since when you're done the boot files *will* be on first-drive, active-partition. The second part is wrong; it needs to be "root=Sdsk(42)", and all other references to "hd" also need to be changed.

Actually, there's probably a better choice. Remove all other "whatever=hd(##)" references; then the values linked into the kernel, from /etc/conf/cf.d/sassign, will take over.

8. Change the /stand mounting arrangements.

The system wants to mount something called /dev/boot as /stand. If you change that, I have no idea what you'll break, but probably something. So the right approach is:

```
# divvy /dev/hd0a    (the SCSI disk)
-- rename the old "boot" as something like "old-boot"
# divvy /dev/hd1a    (the IDE disk)
-- rename the new "boot2" as "boot"
```

Now when you reboot, /stand will point to the files on the IDE drive. >From that point on, when you relink the kernel, the resulting kernel will be stored on the IDE drive. That's where you want it, in order to be able to boot.

The old /stand won't be mounted by default, and its contents will get stale over time. That might be a concern if you were ever going to disable the IDE drive and still want to be able to boot from pure SCSI. Remedy this by occasionally mounting the old /stand (mount /dev/old-boot /mnt) and copying the current kernel onto it.

9. Try to boot.

If, by some miracle, all of the above works, then you should now be able to boot by having the small Unix partition active on the IDE drive, and the IDE drive enabled. To boot other OSes, use "dos" or "bootos ?" at the Boot: prompt. Or use System Commander; though I'm not sure what it really does for you in this situation.

>Bela<

PS: all of the kernel changes *could* be done as one huge change, with only one kernel relink. I wrote this so you might have a chance to back out if a step went wrong along the way, and to use relink-reboot as a sanity test along the way.

Mixing Narrow and Wide

Be careful.

Read the manufacturer's suggestions carefully. Understand the termination issues here, and if you are using adapters to go from wide to narrow, be sure you understand what's happening with the wide part of the signal beyond that adapter. Is this supported at all? Is it terminating? If so, then either you terminate the narrow device or you make sure you are **ONLY** terminating the narrow part of the bus later. If it's not terminating, is it blocking the high bits? If so, then you obviously can't put wide devices beyond that point.

Let's assume you get all that right, through luck, skill, or whatever: your performance is going to suffer. It's not because your wide devices won't work; they could.

But see the Adaptec warning above about mixing Ultra and Ultra2. This is a case where you'll lose the Ultra2 speed!

But even if this will work, any time the narrow device is in use, it's going to hog the bus, and since it's a slower device, that hogging is going to last longer than it would for a wide device. If at all possible, you want your narrow devices on a separate chain. Some adapters make that easy by providing a separate connector specifically for narrow devices. Remember that you need to terminate that chain also. *Some older models of these get very tricky to terminate correctly. The host adapter will have various settings for its termination depending on whether you are using the narrow chain or the external chain, etc. Some of these CANNOT handle devices hooked to all three connectors: check the docs carefully!*

Bad Blocks

Given the size of today's drives, it would be surprising if they didn't have a few bad blocks. Modern drives can handle this by transparently remapping bad areas; moving the data (this assumes that the drive circuitry detects a block going bad but is still able to read the existing data) to another block, and then rearranging things so that references to that bad block get turned into references to the new block instead. Therefore, the "repair" is transparent to your OS. You can read more than you probably want to know about this in "man badtrk". You can enable or disable this feature within "badtrk". Why on earth would you want to disable it? Well, I might rather know a drive is going bad than find out after it has used up all its spare sectors. On the other hand, that's pretty rare today, so I have no firm opinions here.

Adapter Settings

Most higher priced SCSI adapters (and some low priced models) give you considerable control over the behaviour of the adapter. For example, many Adaptec models can display a Ctrl-A prompt as the machine boots; if you respond to the invitation, you get to configure many things. An explanation of Adaptec's options can be found at:

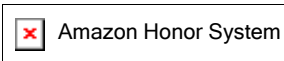
<http://www.adaptec.com/support/faqs/scsiselect.html>; other manufacturers will use similar

terminology.

Another excellent link for Scsi information is Granite Digital (<http://www.scsipro.com/>). These folks advertise that they will give you free tech support for SCSI issues whether you are using their products or not. I've never personally taken advantage of that offer, but I do use their products and can vouch for their quality.

Was this information valuable to you? Consider making a [contribution](#) to help pay for the costs of bringing it to you.

[Amazon Honor System](#) it's fast, safe and easy. It's also very private- Amazon doesn't share your information with anyone: I don't even get to find out who contributed!



Related Articles:

[SCO Unix/OSR5 Technical FAQ](#) (Basics)

[Shell Bashing](#) (Administration, Basics, Linux)

[VPN's and other remote access](#) (Basics, Linux, Security, Networking)

[Unix Permissions](#) (Basics, Linux)

[Transferring Data from an Old Hard Drive](#) (Disks/Filesystems)

[Basic Scripting](#) (Basics, Programming)

[Adding a Hard Drive to Linux](#) (Linux, Basics, Disks/Filesystems)

[Hard Disks](#) (Basics, Disks/Filesystems)

[Loose Cable](#) (Disks/Filesystems)

[SCO Unix/OSR5 Installation FAQ](#) (Basics)

[Floppies](#) (Basics, Disks/Filesystems)

[Basics](#) (Basics)

[General Trouble Shooting](#) (Basics, Administration)

[Cron, At and Batch](#) (Kernel/Internals, Basics)

[Booting OSR5- Filesystems](#) (Kernel/Internals, Disks/Filesystems)

[Booting OSR5](#) (Kernel/Internals, Disks/Filesystems)

[Booting OSR5- Definitions](#) (Kernel/Internals, Disks/Filesystems)

[Booting OSR5- Swap and Dump](#) (Kernel/Internals, Disks/Filesystems)

[Understanding Termcap and Termino](#) (Basics, Modems/Terminals)

[Understanding RAID](#) (Disks/Filesystems, Basics)

[DPT Raid Controller](#) (Reviews, Disks/Filesystems)

[Understanding Dumb Terminals](#) (Basics, Modems/Terminals)

[Veritas \(Online Data Manager\)](#) (Unixware, Disks/Filesystems)

[Using Tapes and tape drives](#) (Basics, Backup)

[Configuring SCO's GUI](#) (Misc., Administration, Basics)

[Networks 101](#) (Networking, Basics)

[Using Secondary drives](#) (Administration, Disks/Filesystems)

[OSR5 Boot Up Messages](#) (Kernel/Internals, Basics)

[Out of Disk space](#) (Disks/Filesystems)

[Adding another hard drive](#) (Administration, Disks/Filesystems)

[Using the Korn Shell](#) (Programming, Basics)

[Vi Primer](#) (Basics)

[Routing Basics](#) (Networking, Basics)

[Understanding Serial Wiring](#) (Basics, Modems/Terminals)

[bfind.c- Finds which file contains a block](#) (Programming, Disks/Filesystems, Kernel/Internals, Code)

© March 1999 Anthony Lawrence. All rights reserved.

This article is copyrighted material. You have permission to use it for any purpose, commercial or non-commercial, as long as it is kept intact and no modifications, additions, or deletions are made except as allowed herein.

You may publish it in paper or electronic form. That includes magazine, newsletters, and web pages, both internal and external, for profit or not. Banner ads and other graphics may be removed, but all other text, hyperlinks and copyright notices, including this, must remain (but see below also). You may not delete text, alter it, or add to it in any way that does not clearly delineate what is yours and what comes from this site. You may alter fonts, font sizes and the like and reformat text as is appropriate for your use.

You may select specific paragraphs or sections, but if you do so, you must include this entire notice also, noting that you have not published the entire article, or simply note that the paragraphs you have published are part of a larger article and give the http address of the actual article.

My main concern is that no one would be confused that you wrote something I wrote or vice-versa. If your use meets that concern, and allows people to find the original article here, I have no objection.

This general permission specifically does NOT apply to test questions and answers.

Some articles at <http://www.aplawrence.com/> and <http://www.pcunix.com/> are copyrighted by other individuals or corporations; these paragraphs do not apply to those articles even if accidentally included.

We do appreciate being advised of any such use: Email: tony@aplawrence.com.