

[spiralbound.net](http://spiralbound.net)

## my digital notebook



Search

Search

### Main menu

[Skip to primary content](#)

- [Home](#)
- [About](#)

### Post navigation

[← Previous](#) [Next →](#)

## Copy a Solaris Boot Drive to a New Disk

Posted on [May 10, 2005](#) by [Cliff](#)

If you've ever gone to mirror a system drive with Solstice Disk suite, you know how frustrating it can be when you either don't have any more slices to use for your meta database partitions, or all the space on the disk has already been allocated to existing partitions. While Disk Suite only requires one partition be reserved for its meta database information on boot drives, two are really suggested for redundancy purposes, and in the example below, I found myself needing to mirror a system disk that had only one remaining partition, and no space left that could be used for the meta database.

While I could have taken a small amount of space from the swap partition and re-allocated it to a new meta database partition on slice 7, this solution would not have been elegant, and I would have still only had one meta database partition. As it stood, the system had the following filesystems on the following disk slices:

c1t0d0

Part Tag

0    root

- 1 swap
- 2 backup
- 3 usr
- 4 usr/local
- 5 opt
- 6 var
- 7 unused

In order to bring the system into line with my standards and prepare it for proper mirroring, I would have to carve up another disk, and migrate the data to it.

Here is what the partition table on the new disk looked like:

c1t2d0

Current partition table (original):

Total disk cylinders available: 24620 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders		Size	Blocks	
0	root	wm	0	— 1088	1.50GB	(1089/0/0)	3146121
1	swap	wu	1089	— 6896	8.00GB	(5808/0/0)	16779312
2	backup	wu	0	— 24619	33.92GB	(24620/0/0)	71127180
3	—	wm	6897	— 6967	100.16MB	(71/0/0)	205119
4	—	wm	6968	— 7038	100.16MB	(71/0/0)	205119
5	opt	wm	7039	— 8853	2.50GB	(1815/0/0)	5243535
6	usr	wm	8854	— 12483	5.00GB	(3630/0/0)	10487070
7	var	wm	12484	— 24619	16.72GB	(12136/0/0)	35060904

Now that everything is all laid out, we can start moving all the data from c1t0d0 to c1t2d0, keeping in mind that we will be merging /usr/local onto /usr on the new system disk... Here we go

Make a new filesystem for /:

```
# newfs /dev/rdisk/c1t2d0s0
```

```
newfs: /dev/rdisk/c1t2d0s0 last mounted as /
```

```
newfs: construct a new file system /dev/rdisk/c1t2d0s0: (y/n)? Y
```

Mount the new / filesystem as /mnt:

```
# mount -F ufs -o rw /dev/dsk/c1t2d0s0 /mnt
```

```
Move the data from c1t0d0s0 to c1t2d0s0:
# ufsdump 0f - / | ( cd /mnt ;ufsrestore xvf - )
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] y
Directories already exist, set modes anyway? [yn] y
DUMP: 405886 blocks (198.19MB) on 1 volume at 406 KB/sec
DUMP: DUMP IS DONE
```

```
Unmount /mnt
# umount /mnt
```

That's the general idea... Now we just have to do the same thing for the other partitions, leaving out swap, backup, and our two meta database partitions of course. These partitions (1,2,3 and 4) should be left alone for the time being, as they are never mounted as filesystems.

```
# newfs /dev/rdisk/c1t2d0s5
# mount -F ufs -o rw /dev/dsk/c1t2d0s5 /mnt
# ufsdump 0f - /opt | ( cd /mnt ;ufsrestore xvf - )
# umount /mnt
# newfs /dev/rdisk/c1t2d0s6
# mount -F ufs -o rw /dev/dsk/c1t2d0s5 /mnt
# ufsdump 0f - /usr | ( cd /mnt ;ufsrestore xvf - )
# umount /mnt
# newfs /dev/rdisk/c1t2d0s7
# mount -F ufs -o rw /dev/dsk/c1t2d0s5 /mnt
# ufsdump 0f - /var | ( cd /mnt ;ufsrestore xvf - )
# umount /mnt
Finally, the /usr/local partition
# mount -F ufs -o rw /dev/dsk/c1t2d0s5 /mnt
# ufsdump 0f - /usr/local | ( cd /mnt/local ;ufsrestore xvf - )
# umount /mnt
```

Now that we have all the data moved, we still don't have a disk that is bootable. Since the whole idea here is for us to end up with a new bootable system disk, we have to install bootblocks onto the new system disk. This is done with the installboot command:

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdisk/c1t2d0s0
```

Now that we have the bootblocks needed to boot the solaris kernel, the last thing we have to do is make sure our new vfstab file points to all the right partitions.

Mount the new / partition:

```
# mount -F ufs -o rw /dev/dsk/c1t2d0s0 /mnt
```

Edit the new vfstab file:

```
# vi /mnt/etc/vfstab
```

For the information given in this example, this file will contain the following entries:

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
fd	—	/dev/fd	fd	—	no	—
/proc	—	/proc	proc	—	no	—
/dev/dsk/c1t2d0s1	—	—	swap	—	no	—
/dev/dsk/c1t2d0s0	/dev/rdsk/c1t2d0s0	/	ufs	1	no	—
/dev/dsk/c1t2d0s5	/dev/rdsk/c1t2d0s5	/usr	ufs	1	no	—
/dev/dsk/c1t2d0s6	/dev/rdsk/c1t2d0s6	/var	ufs	1	no	—
/dev/dsk/c1t2d0s7	/dev/rdsk/c1t2d0s7	/opt	ufs	2	yes	—
swap	—	/tmp	tmpfs	—	yes	—

Notice that the target number will remain 2, not move to 0 when we swap the disks and boot from the new one. To resolve this, it is strongly suggested that you [rebuild the solaris device tree](#) and change the vfstab file to reflect the new disk position.

That is everything! We now shutdown the system, swap the positions of c1t0d0 and c1t2d0 and reboot off our new system disk. We are now ready to move onto the mirroring process.

## People who looked at this item also looked at...

- [Rebuilding the Solaris Device Tree](#)
- [Moving Large Directories on Solaris](#)
- [How to Mirror a Boot Drive in a ZFS Root Configuration](#)
- [Solaris Disk Partition Layout & Mirroring Scripts](#)
- [Migrating Veritas Volume Manager disk groups between servers](#)

## Related items

- [Migrating Veritas Volume Manager disk groups between servers](#)
- [Solaris Disk Partition Layout & Mirroring Scripts](#)
- [Changing Linux Mount Points](#)
- [NetBackup, Solaris 9, and LTO2 drives](#)

- [Things to remember when creating Apache SSL certs](#)

This entry was posted in [Data and Technology](#) and tagged [computing](#), [disksuite](#), [linux](#), [solaris](#), [solstice](#), [storage](#), [technology](#) by [Cliff](#). Bookmark the [permalink](#).

## 18 thoughts on “Copy a Solaris Boot Drive to a New Disk”

1. Pingback: [spiralbound.net » Rebuilding the Solaris Device Tree](#)

2. Pingback: [Jokerbone » Solaris partitioning](#)

3. [cliff](#) on [February 2, 2006 at 9:12 am](#) said:

This sounds like it may be caused by mixed up device links. Did you rebilid them?

[Log in to Reply](#)

4. Pingback: [Life After Coffee » Determining credibility of web resources](#)

5. Pingback: [spiralbound.net » Taking Disk Cylinders From Swap on Solaris 8](#)



6. Maelstrom on [March 9, 2007 at 6:28 pm](#) said:

Should these be set to slice 6 and 7 respectively?

```
EG # mount -F ufs -o rw /dev/dsk/c1t2d0s6/mnt  
and
```

```
# mount -F ufs -o rw /dev/dsk/c1t2d0s7 /mnt  
instead of just using s5
```

```
# newfs /dev/rdisk/c1t2d0s6  
# mount -F ufs -o rw /dev/dsk/c1t2d0s5 /mnt  
# ufsdump 0f - /usr | ( cd /mnt ; ufsrestore xvf - )  
# umount /mnt  
# newfs /dev/rdisk/c1t2d0s7  
# mount -F ufs -o rw /dev/dsk/c1t2d0s5 /mnt  
# ufsdump 0f - /var | ( cd /mnt ; ufsrestore xvf - )  
# umount /mnt
```

Finally, the /usr/local partition

```
# mount -F ufs -o rw /dev/dsk/c1t2d0s5 /mnt
```

```
# ufsdump 0f - /usr/local | ( cd /mnt/local ;ufsrestore xvf - )
```

```
# umount /mnt
```

[Log in to Reply](#)



7. **kjs** on [June 27, 2007 at 10:46 am](#) said:

I have used the above procedure to copy the contents of an entire disk over to a different disk. It worked very nicely, and now I have a system with a mirrored disk set (using DiskSuite).

Thanks, great article

js

[Log in to Reply](#)

8. **cliff** on [June 27, 2007 at 9:40 pm](#) said:

Hey JS,

I'm glad the article was helpful to you 😊

[Log in to Reply](#)

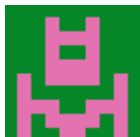


9. **BaffOfDaffy** on [July 23, 2007 at 6:01 pm](#) said:

You may also want to boot to an alternative source (cd/dvd/net) in order to ensure that you are not trying to deal with changing content

[Log in to Reply](#)

10. Pingback: [spiralbound.net](http://spiralbound.net) » [Moving Large Directories on Solaris](#)



11. **Prasen** on [August 20, 2007 at 11:51 am](#) said:

Good document.

With this document I did on SUNfire 440 with Solaris 10. But problem is that its took long time to take backup & restore with 3 GB data using #ufsdump 0f - /usr | ( cd /mnt ;ufsrestore xvf - )

Near about 3 hours it tooks take backup & restore with 3 gb data.

Is there any solution to minimise the time.

One more thing I am not able to rebuild /etc/path\_to\_inst file with following command in Solaris 10 release 10/6

```
# devfsadm -r /mnt -p /mnt/etc/path_to_inst
```

Thanks ...  
Prasen

[Log in to Reply](#)

12. Pingback: [How to copy a Solaris boot drive to a disk with a different partition layout | IDGLabs.COM - tips, tools and resource](#)



13. DaveK on [June 17, 2008 at 5:28 pm](#) said:

Thanks for this article, I've just used it to move to a bigger drive in my E4500.

[Log in to Reply](#)

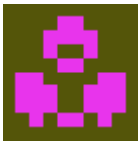


14. rob on [December 17, 2008 at 4:16 pm](#) said:

if you have nfs-shares remember to boot in single user mode, otherwise the procedure will fail

[Log in to Reply](#)

15. Pingback: [Linux Mount | Sekolah Linux Indonesia](#)



16. Cicuta on [April 24, 2011 at 9:05 am](#) said:

Just use dd to make an image of the boot disk. Both disks geometry must be exactly the same.  
Then use the new disk clone as a second boot device at the ok (Open Boot Prom) environment and boot from your new disk.

Easy as pie!

[Log in to Reply](#)



17. yossi soffer on [December 6, 2011 at 11:02 am](#) said:

in some cases (if this is FC disk) you might also need to perform:  
`devfsadm -r /mnt`  
`disks -r /mnt`

in order for the boot process not to stuck.  
also dont forget to change the /mnt/etc/vfstab if you insert the disk to other slot.

[Log in to Reply](#)



18. Alan W. Rateliff, II on [July 5, 2013 at 7:08 pm](#) said:

I came here to verify that I was doing my move correctly, then it hit me I missed one very important item: the 2GB limitation of OBP. Not only must the root filesystem be 2GB or smaller, it must reside in the first 2GB of the drive. Tonight I moved Solaris 8 SPARC from a failing 4GB hard drive, along with a still-operating 9GB secondary drive, to a 64GB SSD on a SCSI-to-SATA adapter. I originally tried using the entire disk as a single filesystem which I had done before on a Solaris 10 x86 system. This, of course, failed. Next I duplicated the slice layout of the original drive but using larger slices commensurate to their usage on the new drive. I neglected that my swap slice resides at the beginning of the drive and is 2GB in size, thus pushing my root filesystem out of the first 2GB of the drive. I swapped positions of the root and swap slices, ensuring root is 1.9GB, applied the boot block and tah dah! One revitalized Axil 320 running Solaris 8 on a 64GB SSD. Thanks for the help!

[Log in to Reply](#)

## Leave a Reply

You must be [logged in](#) to post a comment.

[Proudly powered by WordPress](#)