

Volume

2

New User To Technical Expert – Solaris Bookshelf

---

Stewart Watkiss

This document is currently  
under construction

This version is to be  
considered a preview only

# Solaris on SPARC System Administrator



# Copyright

All rights to this document are retained by the author Stewart Watkiss.

The document is made freely available over the Internet and may be printed for personal use or to pass on to a friend, colleague or family member provided that this copyright message is included in the document.

The document however cannot be sold for profit, whether as a computer file, printed document or any other form without written permission of the author. No part of the document may be copied or included into other works without the written permission of the author.

Whilst I do not forbid the electronic distribution of this document it is discouraged. Instead please direct them to the web page at <http://www.watkissonline.co.uk> where the latest version is referenced.

If the document is distributed in electronic format then the following rules must be observed:

This message must be included as part of the document.

The file must be in its original format without any modification (i.e. the document is provided in Portable Document Format and must be retained in this format).

The document may not be divided or sectioned other than how it is when downloaded, the individual parts may be distributed separately however each section must have this copyright message with it.

If in doubt about any of the above then you should E-mail the author for clarification. You should also E-mail the author if permission is required to go outside of the rules of these conditions. E-mail [solarisbook@watkissonline.co.uk](mailto:solarisbook@watkissonline.co.uk)

If you do not agree with the conditions above then you should immediately destroy any copies (electronic, printed or otherwise) that you may have.

I hope you find the document useful.

Stewart Watkiss



# Contents

<b>Preface to Bookshelf.....</b>	<b>v</b>
<b>Introduction to this book .....</b>	<b>vii</b>
<b>Chapter 1 : USER ADMINISTRATION.....</b>	<b>1</b>
<b>Users .....</b>	<b>3</b>
Special Userids.....	3
Swith User - su.....	3
<b>Groups.....</b>	<b>4</b>
<b>The User and Group Files.....</b>	<b>4</b>
<b>User Administration Commands.....</b>	<b>7</b>
Adding New Users (useradd).....	7
Creating a New Group (groupadd).....	7
Changing a User (usermod).....	8
Changing a Users Password (passwd).....	8
<b>Users Settings .....</b>	<b>8</b>
Home Directory.....	8
<b>Chapter 2 : SPARC HARDWARE .....</b>	<b>10</b>
<b>Hardware options.....</b>	<b>11</b>
SBUS vs. PCI.....	11
Sun Fire .....	11
<b>Chapter 3 : DEVICES .....</b>	<b>13</b>
<b>What is a device?.....</b>	<b>15</b>
<b>Device Representation.....</b>	<b>15</b>
Disk Names.....	17
<b>Chapter 4 : DISK MANAGEMENT .....</b>	<b>19</b>
<b>Managing Disk Slices .....</b>	<b>21</b>
<b>Filesystems.....</b>	<b>21</b>
<b>Using Solstice DiskSuite for Mirrored Disks.....</b>	<b>21</b>

<b>Chapter 5 : STARTUP SEQUENCE .....</b>	<b>29</b>
<b>OpenBoot.....</b>	<b>31</b>
Boot .....	31
Basic OpenBoot Commands .....	32
The eeprom Command .....	33
<b>Chapter 6 : PROBLEM DETERMINATION .....</b>	<b>35</b>
<b>Chapter 7 : SUBJECT .....</b>	<b>39</b>
<b>Chapter 8 : SUBJECT .....</b>	<b>43</b>
<b>Glossary .....</b>	<b>49</b>
<b>Command Summary .....</b>	<b>51</b>
<b>Index .....</b>	<b>53</b>

## Preface to Bookshelf

This is the second UNIX book that I've written. The first is a book on AIX and incorporates the full range of New User to Technical Expert in a single book. The book started out as an aide-mémoire as part of my revision towards becoming an AIX Certified Advanced Technical Expert. I passed the exams and continued writing until it took the form of a complete book on the subject.

After completing my AIX Certification and Book I set myself a second challenge of applying my acquired UNIX skills to learn Solaris. This bookshelf is intended to pass that on to others that may be interested. Rather than writing this as a single book I decided for the bookshelf approach, as it is easier to manage as several smaller books.

This was written whilst I had hands on role implementing some new services on Solaris. This has given me practical experience of how the operating system can be used and in some areas how it really works as opposed to what the manuals say.



# Introduction to this book

This is book 2 in the “New User to Technical Expert Series”.

This book covers information that needs to be known to the System Administrator of a Solaris based computer. It is primarily for when running the Solaris Operating System however some of the sections are equally applicable if the machine is to run Linux or another UNIX operating system.

The one area that won't be covered in this book is dealing with network configuration. This is all in the third book entitled Networking.



# User Administration

This chapter explains some of the common user admin functions, such as adding and deleting a user, and managing user groups.



## Users

All users on a system are identified by a username and a userid. The username is something that users would normally refer to, but as far as the operating system is concerned this is referred to using the userid (or uid). The username is typically a user friendly string, such as your name, whereas the userid is a number. The words username and userid are often (incorrectly) used interchangeably. The userid numbers should be unique (one number per user). If you had two usernames with the same userid, effectively their permissions would be the same and the files that they create would appear to have been created by the same user. This should not be allowed and the `useradd` command will not allow usernames to share the same userid.

Whilst there are no international naming conventions for usernames your company may have its own naming convention (examples usernames could consist of: first names; last names; last name + initial; first name + 1st letter of surname; personnel number etc.).

## Special Userids

There are some default users on all systems when first installed. Other than the root user however these can normally be disabled from logging in. You should however be more careful about deleting usernames as sometimes these are used by different tasks running on the system.

The special users normally have userid numbers that are less than 100, and have names such as `sys`, `bin`, `adm` etc.

The root user has an id of 0, which has a special meaning. The root user has full permissions to do anything on the system. It is not bound by any of the permissions on the system. There are some tasks that can only be performed by root, however it is recommended that you only run as root when necessary as mistakes could be devastating. To use root permissions you would normally login under a normal userid and “`su`” to root as and when required.

## Switch User - su

One of the features of UNIX is the ability to change userid when logged into a system. This command `su` is sometimes referred to as superuser, however this is not completely correct. In the early days of UNIX it was only possible to change to the root user, which made for the superuser command however it is now possible to change to any user using the `su` command. It is more correct to refer to the command as the switch user command.

The switch user command “`su`” is used to change between different users on a system, without having to logout. The most common use is to change to the root user, but it can be used to switch to any user depending upon the user's settings. To switch to a

different user other than root, then the username is used as the last option on the command.

It is also possible to change to another user by putting the username after the su command. There are two ways of switching users. By putting a '-' after the command will cause the users profile to be read and variables to be set. Without the '-' the previous users settings will still remain.

To use the new users profile and variables

```
su - username
```

To continue with the current profile and variables

```
su username
```

you can then return to the previous user by entering exit.

## Groups

Each of the users on a system is also a member of one or more groups. This is a further way of setting permissions for different users. Rather than having to setup individual access permissions for every single user, they can instead be placed in a group of users that have been given that permission. You may for example have a group called “design”, of which the designers are members allowing them access to some of the design documents stored on the system.

Each user will have a default group. If a new file is created this will be automatically be owned by the individual user, and also the users default group. On Solaris this is typically the “staff” group, although many people prefer to create group names, which are the same as the username as the default group, and with the user as the only member of that group. This is considered more secure as then the user will have to explicitly give permission for members of another group rather than it happening by default.

Like the users the groups use a number based system from the operating system point of view. This is referred to as the groupid (or gid).

## The User and Group Files

The user, and group information is stored within the following files:

Properties	Filename
User information	/etc/passwd
Passwords	/etc/shadow
Group Information	/etc/group

Most of the information about a user is in the `/etc/passwd` file, although considering it's name it doesn't normally include the users password. The reason for this is historical, in that it did used to be the file containing the password, however there were security concerns with this. The `/etc/passwd` file has to be available to be read from everyone on the system (so that they can see how the usernames translate into userid's etc.). However the passwords have instead been moved into the shadow file, which can only be read by root (or a program running as root – which is how the password is checked when you login). The password is encrypted anyway, but with a modern computer it is possible to find the password from the encrypted version in just a few hours by using a brute force attack (checking every possible permutation of password one after another).

It is not normally required to edit these files by hand, instead the user administration commands will update these. Entering invalid information may prevent people (including yourself) from logging onto the system, so it is better to use the commands. These are described later.

Below is a view of a password file:

```

root:x:0:1:Super-User:/root:/sbin/sh
daemon:x:1:1::/
bin:x:2:2::/usr/bin:
sys:x:3:3::/
adm:x:4:4:Admin:/var/adm:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
nobody:x:60001:60001:Nobody:/
noaccess:x:60002:60002:No Access User:/
nobody4:x:65534:65534:SunOS 4.x Nobody:/
user1:x:200:200:User 1 full name:/home/user1:/usr/bin/ksh
admin1:x:201:201:Administrator 1 full name:/home/admin1:/usr/bin/ksh
admin2:x:202:202:Administrator 2 full name:/home/admin2:/usr/bin/ksh

```

These are all the system usernames, except for the last three, which are the entries for normal users. The file consists of several entries separated by a colon.

These are:

1. username
2. password (in this case an x means it's now stored in the `/etc/shadow` file)
3. userid
4. primary group id
5. comment (normally the users full name)
6. users home directory
7. command to run (normally a shell)

Below is the matching shadow file

```

root:dklfja9830jkl:12236::::::
daemon:*LK*:12039::::::
bin:*LK*:12039::::::
sys:*LK*:12039::::::
adm:*LK*:12039::::::
lp:*LK*:12039::::::
uucp:*LK*:12039::::::
nuucp:*LK*:12039::::::
listen:*LK*:12039::::::
nobody:*LK*:12039::::::
noaccess:*LK*:12039::::::
nobody4:*LK*:12039::::::
user1:kdfkDFjkl3jklD:12246::::::
admin1: jkl898JKLJJJJ:12235::::::
admin2: j389jkhjJKHj:12241::::::

```

(note that the passwords in this are not real passwords, but instead are random characters).

The file lists the username followed by either a password hash (encrypted form of the users password), or \*LK\* which means that the user is locked and cannot login. This is the normal setting for these usernames. If the entry was blank (ie. nothing between the two colons), then the user would not need a password, and could login, by just entering the username (this should only be used if you have a requirement for an open machine, **it is not recommended**)

The group file is shown below:

```

root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,tty,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
wheel::13:admin1,admin2
sysadmin::14:
nobody::60001:
noaccess::60002:
nogroup::65534:
locked::101:root,daemon,bin,sys,adm,lp,uucp,nuucp,listen,nobody,noaccess,nobody4
user1::200:
admin1::201:
admin2::202:

```

Most of these are again the default groups, with the exception of the last 2. You can see the groupid is field 3 and the last field is a list of usernames that are members of that group. The wheel group is a special case and is normally used to only allow administrators to su to root. As you can see the users groups (the last 3 entries) don't

have any usernames listed against them. This is because that group is defined as their primary group, so the group does not need to be declared again.

## User Administration Commands

Whilst the users can be administered by editing the files, it is the equivalent of tight rope walking without a safety net. It will work if you know what your doing, but one slip and you can't even log back in to fix the problem. It is therefore recommended that you should use the built in commands for managing users whenever possible.

These commands are normally run by root.

### Adding New Users (useradd)

New users can be added using the useradd command.

A typical example of usage is listed below:

```
useradd -m -d /export/home/user1 -g groupname -c "User 1 Full Name"  
user1
```

The above command will add a new user. It will create the users home directory (-m), which will be /export/home/user1 (-d). If the directory is not specified then it will create a home directory in the default base\_dir/login. The primary group will be *groupname* (-g), which may be replaced with the username, if a corresponding group has been created. The comment is set to the users name and the username will be user1.

The users password will then need to be changed before they can login.

Rather than providing the full details of the directories etc. everytime you create a new user, these can be set by using the following command with the parameters you want to set. These will then apply to all newly created users.

```
useradd -D
```

The username should begin with a letter, and it is recommended that it is in lower case (to enable a matching group to be created). It may contain numbers, but with a few exceptions special characters are not allowed.

### Creating a New Group (groupadd)

To create a new group the groupadd command is used.

e.g.

```
groupadd user1
```

The group name should be in lower case and may contain numbers. If you want to use a group as the users primary group then it should be created before issuing the `useradd` command.

## Changing a User (`usermod`)

Users settings can be changed using the `usermod` command. Whilst many of the attributes can be changed the most common use of this command is to change the groups for which a user belongs. This can be done by using the following command:

```
usermod -G group1,203 user1
```

The above command will add set the groups for which `user1` belongs to `group1` and the group with the `gid` 203. As you can see either the `gid`, or the `groupname` can be used. The primary group will not be changed as a result, which can be changed using the `-g` option.

## Changing a Users Password (`passwd`)

The `passwd` command can be used by any user to change their own password. As the root user it is also possible to change other users passwords without having to know their old password. The command is run as:

```
passwd username
```

Another use of the `passwd` command is to force a user to change their password when they first login. This should be run after creating a new user, so that they do not leave the password the same as the administrator first set it to. This is done using the `force` (`-f` option).

```
passwd -f username
```

## Users Settings

There are a number of things that can be setup to define how a user's account behaves, or to make it more convenient for the user. These can sometimes be set in the global configuration files, so that they apply to everyone, or in the users home directory where they would only apply to that user, and can often be updated by the user themselves.

### Home Directory

The home directory is a personal directory setup for each user. It is by default the directory that the user will be in where they first login and is the place where profiles

and personal settings are stored. If you are familiar with the Microsoft® operating systems then you may compare this to the “My Documents” folder (although within UNIX user settings for the operating system and applications are also stored in this directory).

Solaris sets up the home directory differently to some other UNIX operating systems. Typically the home directories are all stored in the /home directory, in a subdirectory named after the user. E.g. the user called stewart would have a home directory /home/stewart . For Solaris however the /home directory is reserved for NFS (allowing the home directory to be stored on a central server and available from whatever machine that the user logs onto within that NFS domain). As a result it is not possible to create any subdirectories of the /home directory except through an NFS setup, or if the automount daemon is disabled. Instead the directory /export/home is used for the local storing of home directories. For the earlier example the user stewart will have a home directory called /export/home/stewart .

The above examples are the typical paths for the home directories, although if it is just a local home directory it could be any folder on the system. The users home directory can be specified when adding a new user (see useradd), and is stored within the /etc/passwd file. Care should be taken if you edit the /etc/passwd file directly as an error in the file could prevent users from being able to login.

The home directory can be referred to using either the variable \$HOME. This makes it possible for programs, and any shell scripts to store and retrieve preferences from the users home directory. The user can also change straight to their home directory by using the cd command without any arguments. In the screenshot below the user was originally in the root directory (viewed using the pwd command), and after issuing the cd command ended up in the home directory. The final command confirms that the \$HOME directory is the same as the directory the user has changed to.

```
$ pwd
/
$ cd
$ pwd
/export/home/idcadmin
$ echo $HOME
/export/home/idcadmin
```

## SPARC Hardware

This chapter explains some of the SPARC hardware options that are available.

## Hardware options

Machines purchased from SUN are based around the SPARC processor, which is a proprietary processor developed by Sun. Whilst SUN has provided a version of Solaris to run on Intel X86 (an similar) processors this appears to have been shelved in favour of SUNs flavour of the linux operating system. Therefore all references are to SUN hardware.

### SBUS vs. PCI

In the past SUN Systems came in two different types. One used the PCI bus format compatible with PC's and other UNIX systems, and their own proprietary architecture called sbus. Typically PCI was used on the low end systems such as ULTRA SPARC machines, whereas sbus was used on ULTRA Enterprise Servers. It is important to identify which you have as the drivers needed are different.

### Sun Fire

The modern Sun Fire machines are based upon the PCI, or cPCI (Compact PCI) formats. However the underlying architecture has been updated.



## Devices

This chapter explains how the devices are recognised on a SPARC and how you can go about viewing and changing them.

This chapter covers generic devices whereas the following chapters explain some specifics used by different devices.



## What is a device?

Mention the word computer device to most people and they think of something either attached externally to the computer or a card inserted into the slots in the back of the computer. Whilst these are all devices it does not full encompass what a device can be. A device could be incorporated into the motherboard or even inside the process. A device is anything physical on the machine performs a specific function. The word physical means that it is not a piece of software, rather than being a discrete component. After all some parts can have multiple devices within them.

Sometimes the software may create a virtual device driver to indicate something that does not actually exist. These are not devices in the true sense of the word however you should be aware of them.

An example of a virtual device is virtual memory. As it's name suggests virtual memory does not necessarily point to a physical device, but instead to a resource that is mapped to either physical memory or a swap file on the hard disk.

## Device Representation

It is possible to list the devices from either OpenBoot or from the Solaris Operating System. There are advantages to looking at or changing the device from the different views that are available.

From the OpenBoot prompt entering show-devs will list all the defined devices detected by the boot process.

```

OK show-devs
/SUNW,ffb@1e,0
/SUNW,UltraSPARC-II@0,0
/counter-timer@1f,1c00
/pci@1f,2000
/pci@1f,4000
/virtual-memory
/aliases
/openprom
/chosen
/packages
/pci@1f,4000/SUNW,hme@4,1
/pci@1f,4000/pci108e,1000@4
/pci@1f,4000/pci@2
/pci@1f,4000/scsi@3
/pci@1f,4000/network@1,1
/pci@1f,4000/ebus@1
/pci@1f,4000/scsi@3/tape
/pci@1f,4000/scsi@3/disk

```

The above display is a partial list only.

It is sometimes obvious what the device is used for, but here's some for a starter.

```
/SUNW,UltraSPARC-II@0,0
```

This represents the CPU which is a SPARC processor. It is from the SPARC II series.

```
/virtual-memory
```

This is the virtual-memory virtual device as mentioned earlier

```
/pci@1f,4000/SUNW,hme@4,1
```

This device refers to a network interface. It is on a pci interface card inserted into slot 4 (hence the address 4,1).

```
/pci@1f,4000/network@1,1
```

This refers to the built in network interface. You can see it's pathname is very different than if it were connected via a pci interface card (see above).

There is a separate but related list available from Solaris. Issuing the ls command against the directory /dev gives the following list.

```
$ ls /dev
```

arp	logindmux	ptye	ptyrb	ticots	ttyqa
audio	m640	ptypf	ptyrc	ticotsord	ttyqb
audioctl	md	ptyq0	ptyrd	tty	ttyqc
be	mem	ptyq1	ptyre	ttyp0	ttyqd
conslog	mouse	ptyq2	ptyrf	ttyp1	ttyqe
console	msglog	ptyq3	qe	ttyp2	ttyqf
cua	null	ptyq4	qfe	ttyp3	ttyr0
diskette	openprom	ptyq5	rawip	ttyp4	ttyr1
diskette0	pcmcia	ptyq6	rdiskette	ttyp5	ttyr2
dsk	poll	ptyq7	rdiskette0	ttyp6	ttyr3
dump	printers	ptyq8	rdsk	ttyp7	ttyr4
ecpp0	profile	ptyq9	rmt	ttyp8	ttyr5
fb	ptmajor	ptyqa	rts	ttyp9	ttyr6
fb0	ptmx	ptyqb	sad	ttypa	ttyr7
fbs	pts	ptyqc	se_hdlc	ttypb	ttyr8
fd	ptyp0	ptyqd	se_hdlc0	ttypc	ttyr9
hme	ptyp1	ptyqe	se_hdlc1	ttypd	ttyra
icmp	ptyp2	ptyqf	sound	ttype	ttyrb
ie	ptyp3	ptyr0	sp	ttypf	ttyrc
ip	ptyp4	ptyr1	stderr	ttyq0	ttyrd
isdn	ptyp5	ptyr2	stdin	ttyq1	ttyre
kbd	ptyp6	ptyr3	stdout	ttyq2	ttyrf
kmem	ptyp7	ptyr4	swap	ttyq3	udp
kstat	ptyp8	ptyr5	syscon	ttyq4	winlock
ksyms	ptyp9	ptyr6	sysmsg	ttyq5	wscons
le	ptypa	ptyr7	systty	ttyq6	zero
llc1	ptypb	ptyr8	tcp	ttyq7	
lockstat	ptypc	ptyr9	term	ttyq8	
log	ptypd	ptyra	ticlts	ttyq9	

There are considerably more entries in this list than the one provided from a show-devs. The reason for this is two fold:

Firstly there can be devices not detected by the initial hardware detection process. For example the show-devs may display a serial port, which has a terminal connected to it. The show-devs will show the serial port only whereas the dev directory may also have a tty entry for the terminal.

Secondly there will be a number of virtual devices. Some of these may be for physical devices that don't actually exist (for example there may be a number of tty terminal definitions that aren't used). The others are for virtual devices, which relate to logical resources. These don't physically exist as components on the system but are created as logical devices to make it easier to manage them.

Looking further at the devices in the /dev directory a lot of them are actually links to other files. For example:

```
ls -l /dev
drwxr-xr-x ... cua
lrwxrwxrwx ... diskette -> ../devices/pci@1f,0/pci@1,1/ebus@1/fdthree@14,3023f0:c
lrwxrwxrwx ... diskette0 -> ../devices/pci@1f,0/pci@1,1/ebus@1/fdthree@14,3023f0:c
drwxr-xr-x ... dsk
lrwxrwxrwx ... ecpp0 -> ../devices/pci@1f,0/pci@1,1/ebus@1/ecpp@14,3043bc:ecpp0
lrwxrwxrwx ... fb -> /devices/pci@1f,0/pci@1,1/SUNW,m64B@2:m640
lrwxrwxrwx ... fb0 -> fbs/m640
drwxr-xr-x ... fbs
dr-xr-xr-x ... fd
lrwxrwxrwx ... hme -> ../devices/pseudo/clone@0:hme
lrwxrwxrwx ... icmp -> ../devices/pseudo/clone@0:icmp
lrwxrwxrwx ... ie -> ../devices/pseudo/clone@0:ie
lrwxrwxrwx ... ip -> ../devices/pseudo/clone@0:ip
lrwxrwxrwx ... stderr -> ./fd/2
lrwxrwxrwx ... stdin -> ./fd/0
lrwxrwxrwx ... stdout -> ./fd/1
drwxr-xr-x ... swap
lrwxrwxrwx ... tcp -> ../devices/pseudo/clone@0:tcp
drwxr-xr-x ... term
lrwxrwxrwx ... tty -> ../devices/pseudo/sy@0:tty
lrwxrwxrwx ... ttyp0 -> ../devices/pseudo/ptsl@0:ttyp0
lrwxrwxrwx ... ttyp1 -> ../devices/pseudo/ptsl@0:ttyp1
```

Note the above has been cut down so that it fits on the screen.

There are 3 types of entry in the /dev directory.

- Real Devices (e.g. diskette) – These link to the physical device drivers in the /devices/pci\* directories
- Virtual Devices (e.g. ttyp0) – These link to virtual device drivers in the /devices/pseudo directories
- Directory Entries (e.g. cua) - These will have one or more items in them that link to physical entries

Device Drivers  
/etc/path\_to\_inst

## Disk Names

The disk devices are contained within the directory /dev/dsk (and also /dev/rdisk to access the raw devices). Each disk is split into slices and all of the individual slices are listed within the /dev/dsk directory.

There are 8 slices for each disk. These can be used for different filesystems, or as a swap device. Any that are not needed should be left with no space allocated to them and they will be left unused. The exception is slice 2 which cannot be used for user-defined filesystems. Slice 2 is instead referred to as the overlap slice and represents the entire disk. This can be used for making backups as a copy of slice 2 would copy the entire disk.

#	ls	/dev/dsk						
c0t0d0s0	c0t0d0s3	c0t0d0s6	c1t0d0s1	c1t0d0s4	c1t0d0s7	c1t1d0s2	c1t1d0s5	
c0t0d0s1	c0t0d0s4	c0t0d0s7	c1t0d0s2	c1t0d0s5	c1t1d0s0	c1t1d0s3	c1t1d0s6	
c0t0d0s2	c0t0d0s5	c1t0d0s0	c1t0d0s3	c1t0d0s6	c1t1d0s1	c1t1d0s4	c1t1d0s7	

The above screenshot shows 3 different disks, one is a CD-ROM drive and the other 2 are SCSI hard disk drives.

To take the name of a disk we can look at the different components.

c0t0d0s0

c = Controller number (0)

t = Target (0)

d = Device / Lun No. (0)

s = Slice / Partition (0)

In this example this is the CD-ROM drive, because that is the only device on the first controller, and the hard disks are on the 2<sup>nd</sup> (No. 1) controller. For a system with only a single hard disk this may instead be the hard disk drive and the CD-ROM would have the target as 1 instead.

# Disk Management

This chapter covers some of the important information on managing disks and filesystems.



## Managing Disk Slices

format

## Filesystems

newfs /dev/rdisk/c?t?d?s? (putting appropriate numbers instead of ?)

## Using Solstice DiskSuite for Mirrored Disks

For a fully comprehensive logical volume manager a third party tool such as “Veritas Volume Manager” should be used. If a basic mirroring is all that is required it is possible to use the Solstice DiskSuite. This is provided free on the installation CD’s but is not installed by default and is not supported by Sun.

Install Solstice DiskSuite 4.2.1 on Solaris 8, inserting CD Solaris 8 Software 2 of 2.

Mount the CD and change directory to the DiskSuite directory (assuming CD is mounted on /cdrom)

```
# cd /cdrom/Solaris_8/EA/products/DiskSuite_4.2.1/sparc/Packages  
(note the directory name will be different for different versions of the operating system).
```

Install SDS. You can install all the files or exclude the GUI part.

```

# pkgadd -d .

The following packages are available:
 1 SUNWlvma      Solaris Volume Management API's
                   (sparc) 1.0,REV=2001.11.02.03.17
 2 SUNWlvmg      Solaris Volume Management Application
                   (sparc) 1.0,REV=2001.11.14.03.19
 3 SUNWlvmr      Solaris Volume Management (root)
                   (sparc) 1.0,REV=2001.11.14.03.19
 4 SUNWmdg      Solstice DiskSuite Tool
                   (sparc) 4.2.1,REV=1999.11.04.18.29
 5 SUNWmdja      Solstice DiskSuite Japanese localization
                   (sparc) 4.2.1,REV=1999.12.09.15.37
 6 SUNWmdnr      Solstice DiskSuite Log Daemon Configuration Files
                   (sparc) 4.2.1,REV=1999.11.04.18.29
 7 SUNWmdnu      Solstice DiskSuite Log Daemon
                   (sparc) 4.2.1,REV=1999.11.04.18.29
 8 SUNWmdr      Solstice DiskSuite Drivers
                   (sparc) 4.2.1,REV=1999.12.03.10.00
 9 SUNWmdu      Solstice DiskSuite Commands
                   (sparc) 4.2.1,REV=1999.11.04.18.29
10 SUNWmdx      Solstice DiskSuite Drivers(64-bit)
                   (sparc) 4.2.1,REV=1999.11.04.18.29

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SUNWlvma> from </install>

Solaris Volume Management API's
(sparc) 1.0,REV=2001.11.02.03.17
Copyright 2001 Sun Microsystems, Inc. All rights reserved.
Using </> as the package base directory.
## Processing package information.
## Processing system information.
   14 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.

The following files are already installed on the system and are being
used by another package:
  /usr/sadm/lib/wbem/doc <attribute change only>

Do you want to install these conflicting files [y,n,?,q] y

```

You may get warnings that certain pre-requisites need to be met. In particular you may get warnings for:

```

WARNING:
  The <SUNWmdr> package "Solstice DiskSuite Drivers" is a
  prerequisite package and should be installed.

```

as this is due to be installed later in the process it is OK to select Y to continue. If you get any other missing pre-requisites then you should check they are installed, or install them.

Reboot the server after the successful installation.

```
shutdown -y -i6 -g0
```

The 2 disks will need to be partitioned the same and will need at least 2 meta DB's per disk. This partitions should be approx. 5MB in size. In this example it has been allocated as larger than that.

```

# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. clt0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@1f,0/pci@1/scsi@8/sd@0,0
  1. clt1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@1f,0/pci@1/scsi@8/sd@1,0
Specify disk (enter its number): 0
selecting clt0d0
[disk formatted]
Warning: Current Disk has mounted partitions.

FORMAT MENU:
  partition - select (define) a partition table
  quit
format> partition

PARTITION MENU:
  0 - change `0' partition
  1 - change `1' partition
  2 - change `2' partition
  3 - change `3' partition
  4 - change `4' partition
  5 - change `5' partition
  6 - change `6' partition
  7 - change `7' partition
  quit
partition> print
Current partition table (original):
Total disk cylinders available: 24620 + 2 (reserved cylinders)

Part      Tag      Flag      Cylinders      Size      Blocks
  0      root      wm         0 - 1451      2.00GB    (1452/0/0)  4194828
  1      swap      wu        1452 - 2903    2.00GB    (1452/0/0)  4194828
  2    backup      wm         0 - 24619     33.92GB   (24620/0/0) 71127180
  3      var       wm        2904 - 5081    3.00GB    (2178/0/0)  6292242
  4      usr       wm        5082 - 7985    4.00GB    (2904/0/0)  8389656
  5 unassigned  wm         0              0          (0/0/0)      0
  6 unassigned  wm         0              0          (0/0/0)      0
  7 unassigned  wm        7986 - 24619   22.91GB   (16634/0/0) 48055626

```

In this example all the disk space is used within the existing partitions. If there were some spare space then it could just be allocated to an unassigned partition, but in this case we need to take some of the swap space to create the new partitions. The partitions will only be about 10MB out of 2GB so it should have no impact on the use of swap memory.

First the machine needs to be in single user mode.

```
init 1
```

Disable swap as follows

```

# swap -l
swapfile      dev  swaplo blocks  free
/dev/dsk/clt0d0s1  32,1  16 4194800 4194800
# swap -d /dev/dsk/clt0d0s1
/dev/dsk/clt0d0s1 was dump device --
invoking dumpadm(1M) -d swap to select new dump device
dumpadm: no swap devices are available

```

Now repartition the disk. Note in this example the swap is first reduced then space is allocated to a unassigned disk slice (slice 5).

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. clt0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@1f,0/pci@1/scsi@8/sd@0,0
  1. clt1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@1f,0/pci@1/scsi@8/sd@1,0
Specify disk (enter its number): 0
selecting clt0d0
[disk formatted]
Warning: Current Disk has mounted partitions.

FORMAT MENU:
  partition - select (define) a partition table
  quit
format> partition

PARTITION MENU:
  0 - change `0' partition
  1 - change `1' partition
  2 - change `2' partition
  3 - change `3' partition
  4 - change `4' partition
  5 - change `5' partition
  6 - change `6' partition
  7 - change `7' partition
  quit

partition> 1
Part      Tag      Flag      Cylinders      Size      Blocks
  1      swap      wu      1452 - 2862      1.94GB      (1411/0/0)  4076379

Enter partition id tag[swap]:
Enter partition permission flags[wu]:
Enter new starting cyl[1452]:
Enter partition size[4076379b, 1411c, 1990.42mb, 1.94gb]: 2030mb

partition> 5
Part      Tag      Flag      Cylinders      Size      Blocks
  5 unassigned      wm      0 -              (0/0/0)      0

Enter partition id tag[unassigned]:
Enter partition permission flags[wm]:
Enter new starting cyl[2892]:
Enter partition size[37557b, 13c, 18.34mb, 0.02gb]: 11c

partition> print
Current partition table (unnamed):
Total disk cylinders available: 24620 + 2 (reserved cylinders)

Part      Tag      Flag      Cylinders      Size      Blocks
  0      root      wm      0 - 1451      2.00GB      (1452/0/0)  4194828
  1      swap      wu      1452 - 2891      1.98GB      (1440/0/0)  4160160
  2      backup      wm      0 - 24619      33.92GB      (24620/0/0)  71127180
  3      var      wm      2904 - 5081      3.00GB      (2178/0/0)  6292242
  4      usr      wm      5082 - 7985      4.00GB      (2904/0/0)  8389656
  5 unassigned      wm      2892 - 2902      15.52MB      (11/0/0)      31779
  6 unassigned      wm      0              0              (0/0/0)      0
  7 unassigned      wm      7986 - 24619      22.91GB      (16634/0/0)  48055626

partition> label
Ready to label disk, continue? y
```

Re-enable swap by issuing the command

```
swap -a /dev/dsk/c1t0d0s1
```

When using only two disks then the machine will not boot unless over half of the partitions exist (ie. more than 2 of the slices). Whilst this may work well for systems with multiple disks it means that for a 2 disk system it will not boot if one of the disks fail. Therefore the following command should be issued to change this setting.

```
echo "set md:mirrored_root_flag=1" >> /etc/system
```

To copy the partition table to the 2<sup>nd</sup> disk issue the following command.

```
prtvtoc /dev/rdisk/c1t0d0s2 | fmthard -s - /dev/rdisk/c1t1d0s2
```

verify this is as expected using the format command.

Create 2 state database replicas on each disk

```
metadb -a -f -c2 /dev/dsk/c1t0d0s5 /dev/dsk/c1t1d0s5
```

The force option is used as this is the first creation.

Now create the two submirrors for the / (root) filesystem, as well as a one way mirror between the meta device and its first submirror.

```
# metainit -f d10 1 1 c1t0d0s0
# metainit -f d20 1 1 c1t1d0s0
# metainit d0 -m d10
```

The first two commands create the two submirrors. The -f option forces the creation of the submirror even though the specified slice is a mounted filesystem. The second two options 1 1 specify the number of stripes on the metadevice and the number of slices that make up the stripe. In a mirroring situation, this should be 1 1. Finally, specify the logical device that we will be mirroring.

After mirroring the root partition, the metaroot command will update the root entry in /etc/vfstab with the new metadevice and add the appropriate configuration information into /etc/system.

Warning: If the metaroot command is not run before a reboot, you will not be able to boot the system!

```
# metaroot d0
```

Now create the mirrors for the rest of the slices

```

# metainit -f d11 1 1 clt0d0s1
# metainit -f d21 1 1 clt1d0s1
# metainit d1 -m d11

# metainit -f d13 1 1 clt0d0s3
# metainit -f d23 1 1 clt1d0s3
# metainit d3 -m d13

# metainit -f d14 1 1 clt0d0s4
# metainit -f d24 1 1 clt1d0s4
# metainit d4 -m d14

# metainit -f d17 1 1 clt0d0s7
# metainit -f d27 1 1 clt1d0s7
# metainit d7 -m d17

```

Edit `/etc/vfstab` with the new devices. Note that this has already been done for slice 0 by the `metaroot` command, so that can be used as a template. The actual device names are replaced with the `d` names created above, and the director has an extra `md` portion to it.

#### Before the change

fd	-	/dev/fd	fd	-	no	-		
/proc	-	/proc	proc	-	no	-		
/dev/dsk/clt0d0s1	-	-	-	swap	-	no	-	
/dev/md/dsk/d0	/dev/md/rdisk/d0	/	ufs	1	no	-		
/dev/dsk/clt0d0s7	/dev/rdisk/clt0d0s7	/export	ufs	2	yes	logging		
/dev/dsk/clt0d0s4	/dev/rdisk/clt0d0s4	/usr	ufs	1	no	logging		
/dev/dsk/clt0d0s3	/dev/rdisk/clt0d0s3	/var	ufs	1	no	nosuid,logging		
swap	-	/tmp	tmpfs	-	yes	nosuid,size=512m		

#### After the change

fd	-	/dev/fd	fd	-	no	-		
/proc	-	/proc	proc	-	no	-		
/dev/md/dsk/d1	-	-	swap	-	no	-		
/dev/md/dsk/d0	/dev/md/rdisk/d0	/	ufs	1	no	-		
/dev/md/dsk/d7	/dev/md/rdisk/d7	/export	ufs	2	yes	logging		
/dev/md/dsk/d4	/dev/md/rdisk/d4	/usr	ufs	1	no	logging		
/dev/md/dsk/d3	/dev/md/rdisk/d3	/var	ufs	1	no	nosuid,logging		
swap	-	/tmp	tmpfs	-	yes	nosuid,size=512m		

Check this is now as expected before proceeding.

Now attach the mirrors using the `metattach` command.

```

# metattach d0 d20
# metattach d1 d21
# metattach d3 d23

```

```
# metattach d4 d24
# metattach d7 d27
```

You can check the stat of the synchronisation by using the metastat command.

```
# metastat
d0: Mirror
  Submirror 0: d10
    State: Okay
  Submirror 1: d20
    State: Resyncing
  Resync in progress: 22 % done
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 4194828 blocks

d10: Submirror of d0
  State: Okay
  Size: 4194828 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Hot Spare
    clt0d0s0         0           No   Okay

d20: Submirror of d0
  State: Resyncing
  Size: 4194828 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Hot Spare
    clt1d0s0         0           No   Okay

d1: Mirror
  Submirror 0: d11
    State: Okay
  Submirror 1: d21
    State: Resyncing
  Resync in progress: 16 % done
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 4160160 blocks

d11: Submirror of d1
  State: Okay
  Size: 4160160 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Hot Spare
    clt0d0s1         0           No   Okay

d21: Submirror of d1
  State: Resyncing
  Size: 4160160 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Hot Spare
    clt1d0s1         0           No   Okay
```



## Startup Sequence

This chapter explains the boot process of the SPARC machine.

This uses a process known as OpenBoot.



## OpenBoot

One of the first processes that a SPARC machine will go through is known as OpenBoot. OpenBoot as it's name suggests is an open standard defined by the Institute of Electrical and Electronics Engineers (IEEE). It's role is to:

- Test and initialise system hardware
- Determine the configuration of hardware
- Boot into the Operating System
- Provide Interactive facilities for setup and debugging purposes

By being an open standard it encourages the development of device drivers that can be loaded during the OpenBoot startup. These are provided on the hardware.

If OpenBoot is not set to start the operating system automatically then the system will stop at an OK prompt during the startup. This is the OpenBoot prompt and can also be accessed by pressing the Stop and a keys simultaneously, or by issuing `init 0` from Solaris. If you have a serial terminal attached (or PC on the serial port), then either the break key, or `SHIFT # CTRL B` will put you into the OpenBoot prompt. The second of these is required if a software fix has been applied to remedy a bug in Solaris where it saw the disconnection of certain terminals as issuing a break.

To exit out of the prompt typing 'go' will restore the running program. Or if you want to restart the system due to a problem with the operating system the 'sync' command can be run to synchronize the filesystems and dump the kernel. It is then safe to reboot.

Another useful command that can help if there is a problem booting the system is the Stop-n key sequence. If you hold these keys down during startup (until the LEDs on the keyboard flash), then the NVRAM will be reset to it's factory defaults.

## Boot

One of the most common tasks form Openboot is to choose how to boot the system. You can choose to boot from cdrom, from the network, or from a specific disk.

Some examples of typical commands are listed:

### **Boot from a named Disk**

```
boot disk1
```

(disk 1 must exist as an alias – if it does not then the full device name can be used – see the Devices section for more details).

### **Boot into single user mode**

```
boot -s
```

### **Boot from cdrom**

```
boot cdrom
```

### **Boot from cdrom into single user mode**

```
boot cdrom -s
```

This command is useful if the root password is forgotten and you need to reset it.

### **Boot from Network**

```
boot net
```

### **Boot from network to install the operating system**

```
boot net - install
```

A Jumpstart Server is required to install the operating system from

### **Boot with reconfiguration**

```
boot -r
```

This is useful if you have made any hardware changes, as it will reconfigure the device aliases with the hardware.

## **Basic OpenBoot Commands**

The following are some of the most basic commands helping you to navigate around OpenBoot.

### **help**

Basic help on commands

### **reset**

Resets the system updating NVRAM so any changes are made permanent

### **set-defaults**

Set all the nvram parameters to the factory default settings

**banner**

Shows basic system information such as system type, ROM Version, Memory and Ethernet MAC address.

**printenv**

Displays the current boot settings

A typical use would be

```
printenv auto-boot?
```

which would show if the system is set to automatically boot if the power is reset.

Note the ? means that this is a boolean value so can hold either true or false.

**setenv**

This is complimentary to the printenv command and can be used to change the current settings.

e.g.

```
printenv auto-boot? true
```

**probe commands**

The probe-scsi and probe-ide commands will search for attached devices on channel 0. Added all to the end will probe all the channels

e.g.

```
probe-scsi all
```

## The eeprom Command

The eeprom command `/usr/sbin/eeprom` can be used to view and change the OpenBoot settings from Solaris. Using this command therefore means that you can make changes without affecting the running of the system, unlike when you switch to the OpenBoot prompt.

Running eeprom without any parameters will list all the settings currently defined on the system.

```
# eeprom
tpe-link-test?=true
scsi-initiator-id=7
keyboard-click?=false
keymap: data not available.
sbus-probe-list=e0123
ttyb-rts-dtr-off=false
ttyb-ignore-cd=true
ttya-rts-dtr-off=false
ttya-ignore-cd=true
ttyb-mode=9600,8,n,1,-
ttya-mode=9600,8,n,1,-
mfg-mode=off
diag-level=min
system-board-serial#=5013132254530
system-board-date=388fcedb
fcode-debug?=false
output-device=screen
input-device=keyboard
load-base=16384
boot-command=boot
auto-boot?=true

...
```

By specifying the name of a parameter it will only display the settings for that parameter.

```
eeprom "auto-boot?"
```

Note: the quotes are needed where a '?' is used in the parameter, otherwise it is not required.

To change the parameter

```
eeprom "auto-boot?"=true
```

System Boot Cycle

System Initialization

Run Levels

## Problem Determination

This chapter lists some of the diagnostic commands used to help determine the cause of problems.



```
fsck /dev/rdisk/c.....
```

(use -y to accept all suggested changes)

```
iostat -En
```

Displays error messages relating to the disk drives.



Chapter

6

Subject

Summary







Chapter

6

Subject

Summary











# Glossary



## Command Summary



# Index

Devices .....	<b>13</b>	OpenBoot .....	<b>29</b>
Displays .....	<b>19, 35</b>	Startup.....	<b>29</b>