

- information
> todo
= switches, software, books
=> syntax
[xx] subject
(xx) additional information
{xx} section description
=== section break
__xx sub-subject
--- divider
::xx command
>> response
>>_ response sub menu
[::xx] file
Ex: example
italic switch

====[::Commands]

---[special characters]

To enter ** type \<**

The only way to quote control character, such as Control-h and Control-m is to precede each with a Control-v, Quotation marks and backslashes don't work for control characters.

Special character Representations Used by od

\b	backspace	\r	carriage Return
\t	tab	\0	null
\n	new line	\f	form feed

---[redirect output]

==> command [arguments] > filename

==> cat banana >> fruit

(create a file without an editor)

> cat > newfile

or

> cat new.file

Ctrl-D

==> command [arguments] < filename

(redirect the standard error to a file)

==> 2>

> ls -l |sort > sorted.list

---[SUN/OS versus Solaris]

SunOS 4.x	SunOS 5.x
(Command)	(Equivalent)
add_services	pkgadd
arch	uname -m
	Note: This returns the kernel architecture.
bar	Not available. Use cpio -H bar to retrieve 4.x archives.
biff -y	chmod o+x /dev/tty
biff -n	chmod o-x /dev/tty
cc	Not available

dbxtool	debugger
devinfo	prtconf
df	df -k
dkctl	Not available
dkinfo	prtvtoc
du	du -k
dump	/usr/sbin/ufsdump
dumpfs	Not available
etherfind	snoop
exportfs	share
extract_files	Not available
extract_patch	Not available
extract_unbundled	pkgadd
fastboot	reboot or init 6
fasthalt	init 0
hostid	sysdef -h
hostname	uname -n
intr	Not available
leave	Use cron and at
lint	Not available
load	pkgadd
loadc	pkgadd
load_package	Not available
lpc	lpadmin
lpd	lpsched
lpq	lpstat
lpr	lp
lprm	cancel
lptest	Not available
mach	uname -p
modstat	modinfo
mount	mount -F [options]
mountall	mount -a
mount_tfs	mount -F
pax	cpio
paxcpio	cpio
portmap	rpcbind
printenv	env
ps -a	ps -e
ps -aux	ps -el
	Note: when ps is invoked without the "-f" flag, the SV "ps" prints only the first N characters of the name of the program being run. You have to pass the "-f" flag to get the full command line. For example, try ps -fe.
pstat	sar
pstat -s	swap -s
rdump	/usr/sbin/ufsdump
restore	/usr/sbin/ufsrestore
rm_client	admintool
rm_services	Not available
rpc.etherd	Not available
rpc.lockd	/usr/lib/nfs/lockd
rpc.mountd	/usr/lib/nfs/mountd
rpc.rexd	/usr/sbin/rpc.rexd
rpc.rquotad	/usr/sbin/rpc.rquotad
rpc.showfhd	Not available

```

rpc.statd          /usr/lib/nfs/statd
rpc.user_agentd   Not available
rpc.yppasswdd     Not available
rpc.ypupdated     ypupdated
rrestore         /usr/sbin/ufsrestore
rusage           Not available
showmount        dfmounts
swapon           swap -a
sys-config       admintool
umount -a        umountall
umount_tfs       umount -F
unload           pkgrm
update           fsflush
uptime          who -b
users           who -q
vipw            Not available
wall            /usr/sbin/wall
whereis         Not available
whoami          id
ypbatchupd      Not available
yppasswd        Use nispasswd for NIS+
ypserv          Not available

```

---[Get System Information]

```

> cd /usr/platform/sun4m/sbin
> prtdiag -v |more (Ultras)
> sysdef (Shows devices)
> showrev -p
> prtconf
> prtmem
> vmstat -5
> ipcs -mbo
> kmastat
> cat trussout
> psrinfo -v

```

---[System Admin Troubleshooting Commands]

```

adb      Analyze dumps on a running system
crash    Analyze crash dumps
diff     Compare file contents
dmesg    Analyze recent log messages
eeprom   Analyze and change boot PROM settings
file     Determine a file's type
find     Look for specific files in the file systems
format   Analyze or modify disk partition information
grep     Analyze file contents, look for specific patterns
ifconfig Analyze the status of network interfaces
kadb     Analyze trap and low-level faults
ls       Analyze file properties
modinfo  Lists modules loaded into a kernel
pkgchk   Check file integrity and accuracy of installation
prtconf  -v Get system device information from POST probe
ps       Analyze properties of running processes
strings  Analyze object and binary files for ASCII strings
sysdef   Analyze device and software configuration information
truss    Trace system calls used by a process

```

```

---[commands to try]
> netstat -in (col/pkgs)x100=collison rate %
> ls [b-f]* (-R) Directory and all subdirs
> more file /string
> touch -c *.*
> view passwd
> man ACL
> scp -r root@voodoo:/var/log/* /home/scot/3-servers/33-servers/voodoo/logs/
> /etc/lp/interface/syshp4k - > nobanner=yes
> lp -n2 todo.html
> atq
> crontab -l /var/spool/cron/crontabs - p.30 sa135-285.html
> crontab = m h dm m dw c > crontab -l crontab
> crontab scot.cronfile
> (?/var/spool/crontabs/root)
> look at mail "designate-host" mailhost in /etc/hosts
> catman -w&
> rusers -l stargate.uchicago.edu

```

```

---[::commands]

```

```

-----
---[::boot]
[corrupt /etc/system]
> boot -as
[corrupt /etc/path_to_inst]
> boot -a

```

```

---[::crash] - Useful in examing the system memory image of a running system or
a crashed system. Shows a snapshot of process information.
> crash
>> p
>> q

```

```

---[::crypt]
[Create crypted file]
> crypt < passwd > passwd.crypt
[crypt file]
> cd /4-servrs
> crypt < passwds.crypt
> root

```

```

---[::cut and paste]
> cat numbers
555-1212    peter
555-1313    sally
555-1414    john
555-1414    jane
> cut -f1 numbers > telnos
> cut -f2 numbers > names
> paste names telnos > new.list
> cat new.list
peter 555-1212
sally 555-1313
john 555-1414
jane 555-1414

```

```

---[::df]

```

```

> df -lk |grep -v mvfs
> df -k |grep -v mvfs

---[::eeprom]
> setenv autoboot? false
> setenv diag-switch? true
> setenv selftest-#megs 128

---[::find]
==> three parts; where to look, what to look for, what to do when you find
[(list all files whose names do not end in .c]
> find / \! -name "*.c" -print
[list all files modified within the last two days]
> find /etc -mtime -2 -print
[list and change the ownership of a file]
> find /dbopt -user 1026 -exec chown 210 \ {}; -print
[remove core files]
> find / -name core -exec rm {} \ 2 > /dev/null
[Damon]
> find . \( -name 'Result*.sql' -o -name 'Query*.sql' \) \
-atime +2 -exec rm -f {} \;
[Kumail]
> find . -name 'Results*.sql' -amin +120 | -exec ls -la {} \ 2 >> /tmp/scot.1
[David]
> porsche7:/usr2/porsche7# find . -user flexlm -exec chown xkfu {} \;

---[::file]
> file * |grep perl

[::grep]
> grep SERVER */license.dat | cut -d\ -f4 |sort -u > port-list
/^The/          match the word "The" at the beginning of a line
/help/          match the word "help"
/W...d/         match a "W", followed by three characters, followed by a "d".
                 (ex: World)
/^$/            match a blank line
/^[0-9]/        match a line that starts with a numeric
/\.$/          match a line that ends with a period. The period is escaped
                using the \ character. Needed, because the "." is a regular
                expression character.

---[::ifconfig]
___[Check the network card]
> ifconfig -a
> ifconfig ba0 down
> ifconfig ba0 up
> ping something
> netstat -nr
___[Temporarily change an IP]
> ifconfig pcelx0 down
> ifconfig pcelx0 129.100.200.199 up

---[::ldd] - Shows which shared libraries your program requires when it is
executed.
Ex: > ldd hello

---[::lsdf]

```

```

- used when dealing with finding a NIS problem
> /usr/local/bin/lsof -i udp -P -n | grep 926
ypserv      6262   root    4u  inet 0x61237b60

---[meminfo]
[see how much memory is being used]
> cat /proc/meminfo

---[::od, ::hexdump] - Dump the contents of a file
> od -c days
> hexdump -c days

---[::ps]
> kill -HUP `ps -ef |grep inetd |egrep -v grep |awk '{print $2}'`

---[::pkgadd]
___[::pkginfo]
> pkginfo -l | grep -i hpnp

---[::priocntl] - Modify or display the CPU scheduling settings for a process.
==> priocntl -l
==> pflags PID - Display signal and flag information for a specific process.
==> pcred PID - Prints the effective, real, and saved UIDs and GIDs for each
process.
==> pmap PID - Prints the address space map for each process.
==> pldd PID - Displays the dynamic libraries linked into each process.
==> psig PID - Displays the signal actions of each process.
==> pfiles PID - Reports status and control information for all open files
associated with a process.
==> pwdx PID - Prints the current working directory of a process.
==> ptree PID - Displays a tree of a process and its children.

---[::reset]
S]h, C]v, C]o

---[::rm]
[delete a "-file"]
> rm /home/scot/-todo

---[::showmount]
[Show who is logged in]
> showmount -e stargate.uchicago

---[::size] - Display the size of the text, data, and BSS(uninitialized data)
segments (or sections) and their totals.
Ex:> size -f hello

---[::sort]
> sort +1 +0 -1 numbers
(sort out any duplicates)
> sort -u origfile > newfile
(sort in alphabetical order)
> sort -n newfile > origfile

---[::ssh]
___[scp]
> scp scot@stargate:/home/scot/all/* /home/scot/all/

```

```

---[::sudo]
> mount -o ro -F hsfs /dev/dsk/c0t6d0s2 /cdrom (as root)
> vi /etc/sudoers (see dd230)
# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for the details on how to write a sudoers file.
#

# Host alias specification
Host_Alias PNMHOSTS=+all-hosts
Host_Alias THISHOST=dd230

# User alias specification
User_Alias ADMINS=+admins
User_Alias USERS=joeba

# Cmnd alias specification
Cmnd_Alias EXPORTMVFS=/etc/export_mvfs
Cmnd_Alias EXPORTMVFSA=/etc/export_mvfs -a
Cmnd_Alias MOUNTCD=/usr/sbin/mount /cdrom
Cmnd_Alias UMOUNTCD=/usr/sbin/umount /cdrom

# User privilege specification
root      ALL=(ALL) ALL
ADMINS    ALL=ALL
USERS     THISHOST=MOUNTCD,UMOUNTCD

> sudo mount /cdrom (as the user)

---[SUID and SGID]
> ls -la /usr/bin/passwd
-r-sr-xr-x 1 root bin 19031 Feb  7 13:47 /usr/bin/passwd*
shows the SUID bit is set
> ls -la /home/disks
-rwxr-sr-x 1 scot staff 19129 Feb 29 13:14 /home/disks
shows the SGID bit is set
This should be considered for any directory that can be written to by
nonprivileged users. (/tmp, /ftp/pub, etc.)

---[::tar]
___(tar across the network)
> cd /home
> tar cf - scot | rsh lilu dd of=/export/backups/all.tar bs=64k
(make sure you edit the /.rhosts if you are root or /etc/hosts.equiv and
~/.rhosts if you are a user)
> find . -depth -print | cpio -o | remsh ns721 '(cd /export/scot;cpio -idm)'

dd if=lilu:/home/all.tar | rsh air "( cd /home/scot/tmp/; tar -xf - )"

> tar cf - . | ssh idenlic2 '(cd /opt/vendor/Frame5.5; tar xfbp -)' = From Tom

> cd (sourcedir)
> find . -print |cpio -pvmud <dest>

```

(on Lilu)

```
Ex: > find . -print | cpio -pvmud air:/home/scot/tmp
    cd (sourcedir)
> tar cf - (source) | (cd <dest>;tar xpvf -)
```

```
___(copy a cd)
> cd /mnt/cdrom
> tar cf - . | (cd /home/ftp/pub; tar xf -)
```

```
---[::tee]
> ls -l | tee save.list
```

```
---[::telnet]
> telnet comm.mot.com 25
```

---[::truss] - trace system calls a process performs, the signals it receives, and the machine faults it incurs.

```
> truss hello
> truss -p PID (To view a running process)
(If you see alot of sleeping, usually the process is either waiting on some system resource, or there is very little activity.)
```

```
---[::xsetroot]
> xsetroot -bitmap nobackdrop
> xsetroot -solid red
```

```
---[::DISPLAY]
___[display a screen in a PC session in xterm]
> /usr/openwin/bin/xterm -display $DISPLAY -sb &
Note: ::pr can be used to format the output of any command. (> man -s 5 signal)
```

=====[::Administration]

---[Basic Tasks]

1. Check Mail

a. Check network logs. Look for unusual traffic or activity.

2. Login to Servers, check the following:

```
> who
```

```
> uptime
```

```
> ls -l /usr/bin/login
```

(Check the mtime)

Hackers will typically replace this file, but will not correct the mtime.

One thing that will typically be done is that they will try and put a trojan horse that allows space-space-space as the password for any user. This is very hard to track, because it will only show up as a blank line in the login file.

Note: > ::strings - finds every sequence of bytes

Note: > cat trojan /usr/bin/login will replace the contents of the login file but leave the original mtime intact.

Note: > ls -lc - shows creation time

Note: > ls -l - shows modification time

Note: > ::sum /usr/bin/login does the same thing as the above command as long as you remember the size it was. This command does a DES hash.

Note: You can copy /usr/bin/login from the SUN CD.

Check /var/log

---[Check logs]

```
> grep refused /var/log/tcplog
```

```
> grep refused /var/log/tcplog |grep rsh
> grep ais-pc10 /var/log/tcplog (In this example, to see if ais-pc10 connected)
(Note: The setup file for this is syslog.conf. All servers are setup to send
logs to melanoma)
(Note: There is a Perl script that rotates logs. It copies logs to an old file.)
```

```
---[login as root]
(non root access)
telnet into the account
su to root
ps -ef |grep xlock
kill -9 === (p.i.d. of xlock)
```

```
---[don't have a non-root account]
login to another machine as root
edit /etc/default/login
comment out the line that says: "console"
ftp /etc/default/login to locked machine
telnet in as root
[Replace a missing/damaged /etc/passwd, /etc/hosts file]
> stop-a
> boot cdrom -s
> fsck /dev/dsk/c0t3d0s0
> mount /dev/dsk/c0t3d0s0 /a
> (refer to files on CD through /etc)
> (refer to files on disk through /a/etc)
> umount /a
> reboot
```

```
---[add a drive to a live system]
From: Lukas Karlsson <karlsson@colltech.com>
#CI20446: What commands to add a disk to a "live" Solaris server
```

I want to add disks to a live Sun Enterprise 250(not a 450 as mentioned previously) server running Solaris 2.6.

There is a sequence of two or three commands that can be run at the command line which will do the same as "ok> boot -r"; that is, search for and configure new devices.

To get Solaris to see the disk, you'll need to do:

```
> drvconfig
> disks
If you are using Veritas, then you'll want it to see the disk as well,
which means you'll have to do a:
> vxdctl enable
```

```
====[:Backups]
---[Tape Restore]
> cd /temp
> script trust1.restore
> vi mapit
> _ mt rewind
> _ ufsrestore -tvf /dev/rmt/0 (0=id4;1=id5) file.1
> _ ufsrestore -tvf /dev/rmt/0 file.2
> _ ufsrestore -tvf /dev/rmt/0 file.3 (etc. as needed)
> _ wq!
```

```

> chmod 700 mapit
> mt rewind
> ufsrestore -xfs /dev/rmt/0 1 (filename)
> ufsrestore -xfs /dev/rmt/0 2 (filename), etc.

---[Restore files] - Note: This will change the root password.
> cd /temp/etc
> cp passwd /etc/.
> cp shadow /etc/.
> cp group /etc/.
> cd /temp
> cp -p .profile /

---[Restore a directory]
(Restore /usr/home)
> cd /
> mkdir /newusr
> mount /dev/dsk/c1t0d0s1 /newusr
> cd /newusr
> mt rewind
> ufsrestore -xfs /dev/rmt/0 2 (file 2)
> cd /
> umount /newusr
> newfs /dev/rdisk/c1t0d0s1
(Note: /home is often used for DNS, and /export/home is used for NIS)
(Restore /opt)
> cd /
> mkdir /newopt
> mount /dev/dsk/c1t0d0s1 /newopt
> cd /newopt
> ufsrestore -xfs /dev/rmt/0 4
(Restore /var/opt/oracle)
> cd /var/opt
> mkdir oracle
> cd oracle
> mt rewind
> ufsrestore -xfs /dev/rmt/0 3 (file 3)

---[ufsdump Table of Contents]
> ufsdump -tvf /dev/rmt/0

====[::Config Files]

---[Allow root logins]
> cd /etc/pam.d
> vi rlogin
#auth          required      /lib/security/pam_securetty.so

---[Comment out /etc/passwd or /etc/shadow]
/etc/passwd = * (entry)
/etc/shadow = *lk* (entry)

---[FILE Lists]
___[rlogin] - p634
/etc/hosts.equiv - system level logins allowed
&/home/.rhosts - user level logins allowed
___[inetd] - p635

```

/etc/inet/inetd.conf - manages network services such as ftp, telnet, etc.

___[ports] - p636

/etc/services -

___[cu]

/etc/uucp/systems

___[ftp]

\$/home/.netrc - automatic login

___[TCP/IP files]

~/.rhosts

/etc/ethers

/etc/hosts

/etc/hosts.equiv

/etc/inet/inetd.conf

/etc/inet/rc.inet

/etc/init.d/inetinit

/etc/netmasks

/etc/networks

/etc/protocols

/etc/services

/etc/confnet.d/inet/"interface file"

/etc/confnet.d/inet/config.boot.sh

___[initialization, init - (run level)]

/etc/inittab

___[NFS, sharing files] - p631

/etc/auto_master - list of directories to be mounted by automount

/etc/auto_home - list of home directories

___[shutdown security - HP]

/etc/shutdown.allow

___[Mail]

/etc/aliases

___[Restrict root login - HP]

/etc/securetty

___[syslogd config file]

/etc/syslogd.conf

___[shadow]

/tcb/files/auth/1stletter/username (man -f prpwd)

---[FTP]

___[Setting up anonymous FTP for Solaris]

I just set up anonymous FTP for Solaris (previously I had configured it for SunOS, which is of course totally different), and I have a few extra tips in addition to what is mentioned in the ftpd man page. You should add them to your security FAQ.

1) In order to show the correct user and group ID's in "ls -l", you need to copy the /usr/lib/libmp.so.1 library in addition to all of the other libraries in ~ftp/usr/lib.

2) If you're using NIS+, and want to search NIS+ for user and group ID's (again, for "ls -l"), you need to copy NIS_COLD_START and NIS_SHARED_DIRCACHE from /var/nis to ~ftp/var/nis. You also have to copy /etc/nsswitch.conf to ~ftp/etc, of course.

3) You need the correct timezone info in ~ftp/usr/share/lib/zoneinfo, but you don't need ALL the timezones. If you want to save almost 400k, you can copy just the timezone you need (for example "US/Pacific"). Do an "ls -l" from within FTP and from a shell prompt, and make sure the two times agree.

Here's another tip if you are having any problems with anonymous FTP (for example, if you want to set up wu-ftp). Follow these steps and you can run a system call trace on ls in the same chroot'ed environment that ftpd uses.

```
# mount -F lofs /proc ~ftp/proc # loopback mount /proc (truss needs it)
# cp /usr/bin/truss ~ftp/usr/bin # copy the truss command itself
# chroot ~ftp /bin/truss /bin/ls -l /pub # or any other command
# rm ~ftp/usr/bin/truss
# umount ~ftp/proc
You will need to substitute ~ftp for the absolute pathname of the FTP home
directory. Enjoy!
```

```
---[See ntpd date entry]
> cat /etc/crontab
```

```
---[machine name change]
___Solaris:
> vi /etc/hosts
> vi /etc/hostname.hme0
> vi /etc/nodename
> vi /etc/net/ticlts/hosts
> vi /etc/net/ticots/hosts
> vi /etc/net/ticosord/hosts
___SUN:
> vi /etc/hosts
> vi /etc/hostname.*
```

```
---[change an IP]
> sys-unconfig
```

```
---[Partition, Add a (Solaris)]
> format, 4, (pick a range Ex: 888-1066)
> _ Label
> _ exit
> newfs /dev/rdisk/c0t0d0s4
> mkdir /temp
> mount /dev/dsk/c1t0d0s0 /temp
```

```
---[edit /temp/etc/services, /temp/etc/system]
Cut and paste from /temp/etc/services to /etc/services
> textedit /etc/services &
> textedit /temp/etc/services &
> copy last six lines from /temp/etc/services file and paste into /etc/services
> Save and exit
Cut and paste from /temp/etc/system to /etc/system
> textedit /etc/system &
> textedit /temp/etc/system &
> copy the Oracle section (about ten lines) from /temp/etc/system file and
paste into /etc/system
> Save and exit
(> init 6 to take effect)
```

```
---[TERM]
> term=vt100
> stty rows=24 columns=80
```

```
====[::DNS]====
Root Domains
a.root-server.net - m.root.server.net
  Top Level Domains
  a. Organizational
```

- b. Geographics1 (www.iana.org/domain-names.html
."nasa".gov. (nasa is a label)

67 = max characters a domain or label can have (= < 12 recommended)
127 = max number of subdomains
255 = max characters in a FQDN

If a domain becomes too large, a subdomain is created. At some point it becomes necessary to delegate responsibility for the subdomain to another administration and its name server. The subdomain becomes a separate zone because it has a different name server. This is known as delegation.

DNS "resolvers" or "Name Servers", store information about the DNS tree.

There are three main types of resolvers:

- Primary Name Server
 - Secondary Name Server
 - Caching Only Server
 - Stub Resolver (C library function)
- See see p.598 & 599 for a breakdown of a typical DNS query.-

[Primary Name Server - PNS]

The PNS contains the authoritative zone database file. It contains resource records of hostname/IP, and other information about the zone. It queries other Name servers, caches replies, and its database is used to delegate responsibility for subdomains to other nameservers. To change the information for a domain, the zone database file on the PNS must be incremented (changed), in order for Secondary Name Servers to update their maps.

Note: To change a SOA serial number that has become too big, do the following:

1. Change the SOA to a smaller number on the PNS.
2. Shut down all Secondary Name Servers.
3. Remove SNS name database files.
4. Restart the in.named on the SNS's.

[Secondary Name Server - SNS]

Should have at least one for redundancy purposes. A SNS obtains a copy of the zone database from the PNS (usually), and also serves as an authoritative server for the zone. SNS's can also query and cache.

Periodically it does a zone transfer if needed. (The default is every 3 hours).

[Caching Only Name Server]

This server caches only in order to increase response time when resolving hosts.

[Stub Resolver]

This is actually just a C Library function that allows programs to directly query the Name Server to resolve hostnames.

[Resource Records]

Entries stored in the DNS database. They are stored in the zone database on the PNS, and is replicated out via zone transfers.

[Database Files]

These are the flatfiles that store the hierachical database information for your domain. There should only be one copy, stored and edited on the PNS, and transferred to all SNS's.

Database files have at least four files:

- named.hosts (ex: db.192.168.1) = IP mapping to hostname
(Usually PTR records, used for reverse address resolution)
- named.rev (ex: db.orion) = Reverse hostname to IP
(Usually A and CNAME records)
- named.local = local host map
(Copy seu p.602-603)
- named.boot = /etc/named.boot (inamed configuration file)

(Each file has three sections)

- 1 - SOA = Start of Authority
- 2 - NS = Name Server
- 3 - Database section

=> [TTL][class] type data

=> [class][TTL] type data

{Fields}

TTL (optional) = decimal number that indicates to the Name Server how often this record should be updated. (Usually a few minutes-few days, blank assumes default of 3hrs)

class (optional) = Indicates which class. The only main class that is used is IN (Internet Data).

(Types of Resource Records)

A = Address record. hostname to IP

AAAA = IPv6 Host Address record.

NS = Name Server Identification

PTR = Pointer, or PTR record. Associates an IP with a hostname. Typ

SOA = Start of Authority

WKS = Well Known Service

HINFO = Host information

MX = Mail Exchanger

TXT = General purpose text records

CNAME = Canonical Name (alias). To use a CNAME, you must have an A record for the host as well.

::SOA Section (sample)

```
orion.      IN      SOA      lilu.orion. corrie.orion. (  
    309          ; serial number  
    21600        ; refresh (3hrs)  
    3600         ; retry (1 hr)  
    432000      ; expire (5 days)  
    3600        ; minimum (1 hr)
```

(Description)

serial - revision

refresh - how often SNS to check for zone transfer

retry - how often SNS should reattempt a failed zone transfer

expire - how long to retry before SNS ceases to serve data for the zone

minimum - how long NS allowed to cache responcees that orginate from this zone

::NS Section

See seu p.602 - all

Contains a Name Server resource record for each of the primary and secondary NS for the zone that the database serves. The address records have to exist for each of the Name Servers if they are in the current zone.

::Database Section

Contains all the resource records that contain data for hosts in the zone.

::PTR

To lookup a hostname from an IP, the NIC creates a special domain (which is in-addr.arpa.) This domain stands for inverse address. The NIC has delegated authority of the 1.1.168.192.in-addr.arpa. domain to the lilu.orion. Primary Name Server.

=====

---[::DNS: Primary Name Server Setup - zorg.orion.com (Solaris)]

1. Apply for IP's at: <http://rs.internic.net>
2. Decide whether one or several domains are needed.
3. Get the latest version of BIND at: www.isc.org
(::in.named - This is the main BIND daemon)
(::named.boot - Provides in.named configuration information.)
4. Edit the /etc/named.boot configuration file.
(Description)
Directory - path to the zone database files
Cache - loads initial entries into the Name Server cache.
(Usually the root server addresses)
Loads a list of the root servers from the file db.cache in /var/named. Contains a list of resource records for the root name servers.
(Latest list: <ftp://rs.internic.net/netinfo/root-servers.txt>)
Primary - Declares the Primary Name Server, the zone, and the database must be specified with this directive.
Secondary - Declares the Secondary Name Servers, the zone, IP of the Primary Name Server, and the database file must be specified with this directive.

```
> vi /etc/named.boot
; directory /usr/lib/named (on Linux)
directory /usr/local/adm/named
```

```
primary      orion.com      named.hosts
primary      1.168.192.IN-ADDR.ARPA  1.168.192.rev
primary      0.0.127.IN-ADDR.ARPA    named.local
cache        .                named.ca
```

```
5. > vi /usr/local/adm/named/named.hosts (Linux = /usr/lib/named)
;
; Section 1: SOA Section
;
orion.com.   IN      SOA      zorg.orion.com. root.zorg.orion.com. (
                1          ; Serial
                14400     ; Refresh (4 hrs)
                3600      ; Retry (1 hr)
                604800    ; Expire (4 wks)
                86400 ) ; minimum TTL

;
```

```

; Section 2: Name Servers
;
orion.com. IN NS zorg.orion.com.
orion.com. IN NS lilu.orion.com.

;
; Section 3: Hosts
;
localhost.orion.com. IN A 127.0.0.1
zorg.orion.com. IN A 192.168.1.2
stargate.orion.com. IN A 192.168.1.10
starchild.orion.com. IN A 192.168.1.11
corbin.orion.com. IN A 192.168.1.20
oni.orion.com. IN A 192.168.1.21
fire.orion.com. IN A 192.168.1.22
water.orion.com. IN A 192.168.1.23
earth.orion.com. IN A 192.168.1.24
air.orion.com. IN A 192.168.1.25
misha.orion.com. IN A 192.168.1.26
outland.orion.com. IN A 192.168.1.29
starbase.orion.com. IN A 192.168.1.30
k7.orion.com. IN A 192.168.1.31
ftp.orion.com. IN CNAME lilu
www.orion.com. IN CNAME lilu

;
; Section 4: Multihomed Hosts
;
lilu.orion.com. IN A 192.168.1.1
lilu.orion.com. IN A 192.168.2.3

6. > vi /usr/local/adm/named/1.168.192.rev (Linux = /usr/lib/named)
;
; Section 1: SOA
;
1.168.192.IN-ADDR.ARPA. IN SOA zorg.orion.com. root.zorg.orion.com. (
1 ; Serial
14400 ; Refresh (4 hrs)
3600 ; Retry (1 hr)
604800 ; Expire (4 wks)
86400 ) ; minimum TTL

;
; Section 2: Name Servers
;
1.168.192.IN-ADDR.ARPA. IN NS zorg.orion.com.
1.168.192.IN-ADDR.ARPA. IN NS lilu.orion.com.

;
; Section 3: Reverse Address Mappings
;
1.1.168.192.IN-ADDR.ARPA. IN PTR lilu.orion.com.
2.1.168.192.IN-ADDR.ARPA. IN PTR zorg.orion.com.
10.1.168.192.IN-ADDR.ARPA. IN PTR stargate.orion.com.
11.1.168.192.IN-ADDR.ARPA. IN PTR starchild.orion.com.
20.1.168.192.IN-ADDR.ARPA. IN PTR corbin.orion.com.
21.1.168.192.IN-ADDR.ARPA. IN PTR oni.orion.com.

```

```

22.1.168.192.IN-ADDR.ARPA.    IN      PTR      fire.orion.com.
23.1.168.192.IN-ADDR.ARPA.    IN      PTR      water.orion.com.
24.1.168.192.IN-ADDR.ARPA.    IN      PTR      earth.orion.com.
25.1.168.192.IN-ADDR.ARPA.    IN      PTR      air.orion.com.
26.1.168.192.IN-ADDR.ARPA.    IN      PTR      misha.orion.com.
29.1.168.192.IN-ADDR.ARPA.    IN      PTR      outland.orion.com.
30.1.168.192.IN-ADDR.ARPA.    IN      PTR      starbase.orion.com.
31.1.168.192.IN-ADDR.ARPA.    IN      PTR      k7.orion.com.
; Not sure if the below is needed.
;1.1.168.192.IN-ADDR.ARPA.    IN      PTR      ftp.orion.com.
;1.1.168.192.IN-ADDR.ARPA.    IN      PTR      www.orion.com.

```

```

7. > vi /usr/local/adm/named/2.168.192.rev (Linux = /usr/lib/named)

```

```

;
; Section 1: SOA
;
2.168.192.IN-ADDR.ARPA. IN      SOA      zorg.orion.com. root.zorg.orion.com. (
                        1          ; Serial
                        14400     ; Refresh (4 hrs)
                        3600      ; Retry (1 hr)
                        604800    ; Expire (4 wks)
                        86400 ) ; minimum TTL

;
; Section 2: Name Servers
;
2.168.192.IN-ADDR.ARPA. IN      NS      zorg.orion.com.
2.168.192.IN-ADDR.ARPA. IN      NS      lilu.orion.com.

;
; Section 3: Reverse Address Mappings
;
1.2.168.192.IN-ADDR.ARPA.    IN      PTR      lilu.orion.com.
2.2.168.192.IN-ADDR.ARPA.    IN      PTR      zorg.orion.com.
10.2.168.192.IN-ADDR.ARPA.   IN      PTR      mangaloid.orion.com.

```

```

8. > vi /usr/local/adm/named/named.local (Linux = /usr/lib/named)

```

```

0.0.127.IN-ADDR.ARPA. IN      SOA      zorg.orion.com. root.zorg.orion.com. (
                        1          ; Serial
                        14400     ; Refresh (4 hrs)
                        3600      ; Retry (1 hr)
                        604800    ; Expire (4 wks)
                        86400 ) ; minimum TTL

;
; Section 2: Name Servers
;
0.0.127.IN-ADDR.ARPA.    IN      NS      zorg.orion.com.
0.0.127.IN-ADDR.ARPA.    IN      NS      lilu.orion.com.

;
; Section 3: Reverse Address PTR Mapping
;
1.0.0.127.IN-ADDR.ARPA. IN      PTR      localhost

```

```

9. > vi /var/local/adm/named.ca (Linux = /usr/lib/named)

```

```

;
; Note: This should not be used if you are not connected to the internet.
;
; Servers for the root domain are listed below
;
.      99999999      IN      NS      terp.umd.edu.
      99999999      IN      NS      aos.brl.mil.
      99999999      IN      NS      c.nyser.net.
      99999999      IN      NS      ns.nasa.gov.
      99999999      IN      NS      ns.nic.ddn.mil.
      99999999      IN      NS      nic.nordu.net.
      99999999      IN      NS      ns1.isc.org.
      99999999      IN      NS      ns.isc.org.
      99999999      IN      NS      ns.internic.net.

```

```

;
; Root Servers by address
;
terp.umd.edu.      99999999      IN      A      128.8.10.90
aos.brl.mil.      99999999      IN      A      192.5.25.82
c.nyser.net.      99999999      IN      A      192.33.4.12
ns.nasa.gov.      99999999      IN      A      192.52.195.10
ns.nic.ddn.mil.  99999999      IN      A      192.112.36.4
nic.nordu.net.   99999999      IN      A      192.36.148.17
ns1.isc.org.     99999999      IN      A      128.9.0.107
ns.isc.org.      99999999      IN      A      198.41.0.4
ns.internic.net. 99999999      IN      A      192.5.5.241

```

```

10. > /usr/sbin/in.named
     or > /usr/sbin/in.named -b /etc/inet/named.test.boot

```

(To force named to incorporate changes, send it the SIGHUP signal.)

-- Signals understood by named --

USR1 Increments debug level by 1 (data goes to named.run)

USR2 Turns off debugging

INT Dumps database to named_dump.db

IOT/ABRT Dumps statistics to named.stats

HUP Reloads root file and database files

KILL kills named, must restart by hand (flushes cache)

WINCH Toggles tracing of incoming queries (BIND 4.9)

```

11. > vi /etc/defaultdomain
     orion.com

```

```

12. > domainname orion.com

```

```

13. > vi /etc/nsswitch.conf
     hosts:          files dns

```

```

14. TEST.
     > nslookup (Ctrl-D or exit to Exit)
     a. _> <hostname>
        IP and FQHN returns
     b. _> set q=soa
        _> zorg.orion
        SOA record returns
     c. _> set q=ptr

```

```

    > 192.168.2.1
    FQHN returns
d. > server 207.217.126.81 (To change Name Server target)
e. > ls orion.com (Dumps the entire map)
f. > set type=any (means "all")
g. > set debug
h. > set d2 (turns on lots of debugging)
f. > help

```

[Troubleshooting - See seu p.611-612]

```

- Check the following
/usr/lib/libresolve
/etc/resolv.conf (on hosts)
/etc/nsswitch.conf

```

---[::DNS: Secondary Name Server Setup - zorg.orion.com]

```

1. Copy the following files from zorg
  > scp zorg:/etc/named.boot /etc
  > scp zorg:/usr/local/adm/named/named.ca /usr/lib/named/ (Linux target)
  > scp zorg:/usr/local/adm/named/named.local /usr/lib/named/ (Linux target)

```

2. The only file that needs changing is named.boot

```

> vi /etc/named.boot
directory      /usr/lib/named
;directory     /usr/local/adm/named - Solaris

secondary      orion.com          192.168.1.1 named.host.bak
secondary      1.168.192.IN-ADDR.ARPA  192.168.1.1 1.168.192.rev.bak
secondary      2.168.192.IN-ADDR.ARPA  192.168.1.1 2.168.192.rev.bak
secondary      0.0.127.IN-ADDR.ARPA   named.local
cache          .                    named.ca

```

3. > vi /etc/resolv.conf

```

search        orion.com
nameserver    192.168.2.1
nameserver    192.168.2.2

```

4. > /usr/sbin/in.named

- (Note: Copy example in seu p.610)

---[::DNS: Cache-Only Name Server Setup - mangoloid.orion.com]

(Note: If you are not part of the internet, but want to use DNS, you can declare your primary name server authoritative for the root domain. In which case, your cache file should point to local name servers, not to the root servers of the internet. But change back before you connect to the internet.)

1. Copy the following files from lilo. (Does not maintain any database files.)

```

  > scp zorg:/usr/local/adm/named/named.ca /usr/lib/named/ (Linux target)
  > scp zorg:/usr/local/adm/named/named.local /usr/lib/named/ (Linux target)

```

2. > vi /etc/named.boot

```

;
; Cache-Only server for the orion.com domain
;
primary        0.0.127.IN-ADDR.ARPA   /usr/lib/named/named.local
cache          .                    /usr/lib/named/named.ca

```

;

3. > /usr/sbin/in.named

---[DNS Client Setup]

::resolv.conf - Main resource for clients to find resolver

(Resolver Directives)

domain - appended to partially qualified domain names that are looked up.
search - allows multiple domain names to be attempted for partially qualified domain names.
nameserver - tells resolver the target for DNS queries. Up to three should be used. No entries here will default to 127.0.0.1

1. > vi /etc/resolv.conf

```
search orion.com
nameserver 192.168.2.1
nameserver 192.168.2.2
```

2. > vi /etc/nsswitch.conf

```
hosts: files dns
```

=====
---[::BIND Installation and Maintenance Chores]

Chore	For	How Often
1 Obtain domain name	Site	once
2 Choose name servers	Site	once
3 Obtain BIND distribution	Site	once (may already have)
4 Configure resolver	client	once and distribute (rdist)
5 Configure efficient resolver	client	each subnet and distribute (rdist)
6 Configure services switch	client	each arch and distribute (rdist)
7 Start named at boot time	Server	each name server
8 Configure boot file	Server	each type of server
9 Configure cache file	Server	once and distribute to servers
10 Configure zone files	Primary	once
11 Update sendmail	Mail Hub	For your smart mail hub
	all hosts	
12 Update zone files	Primary	As needed
13 Review log files	Log host	At least weekly
14 Educate users	all hosts	Ongoing

[BIND files in Solaris]

```
/etc/resolv.conf Resolver library configuration
/etc/in.named Name server daemon
/usr/sbin/named-xfer Zone transfer code
/etc/named.boot Boot file for name servers
/etc/named.pid Process ID
/var/tmp/named.run Output from debug mode
/var/tmp/named.status Statistics output
/var/tmp/named_dump.db Dump of entire database
```

--- Redhat Setup - First Configured on 01-07-01

1. /etc/init.d/named stop

2. vi /etc/named.boot

;

```

; a caching only nameserver config
; First Attempt at setting up a DNS server by Scot on 01-07-01
;
directory                /var/named
cache                    .                root-servers
primary                  orion.com       domain.hosts
primary                  1.168.192.IN-ADDR.ARPA domain.reverse
primary                  0.0.127.in-addr.arpa  named.local

```

3. vi /var/named/named.local

```

$TTL      86400
@         IN      SOA      localhost. root.localhost. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum
                                IN      NS      localhost.

1         IN      PTR     localhost.
lilu     IN      A       192.168.1.1
ns1      IN      CNAME   lilu
orion.com. IN    CNAME   lilu
corbin   IN      A       192.168.1.2
ns2      IN      CNAME   corbin
stargate IN      A       192.168.1.10
starchild IN     A       192.168.1.11
fire     IN      A       192.168.1.20
starbase IN      A       192.168.1.30
k7       IN      A       192.168.1.31

```

4.

```

ping      to test name resolution
nslookup to test DNS server information.
nismatch  to test NIS+ server information
ypmatch  to test YP  server information
grep     to test hosts file information  ;-)

```

---(Setup)

```

> vi /opt/named/hosts.db
> vi /opt/named/hosts.rev
[Kill and restart Named]
> /opt/named/update_dns

```

---[Help with Configuring for Different domains]

```

> /opt/SUNWnsdis/bin/newdom.sh

```

=====

=====[::Hardware]

```

---[Add a drive to the Ultra 5]
ok? setenv auto-boot? false
ok? reset-all
boot -r
probe-ide
ok? setenv auto-boot? true

```

```

---[Mount an external cdrom on a Ultra 2]
> stop -a
> setenv auto-boot? false
> reset-all
> boot -r
> setenv auto-boot? true

> setenv auto-boot? true
(Volume Manager has been disabled for /cdrom; see /etc/vold.conf )
> vi /etc/vfstab
/dev/dsk/c0t6d0s2      /dev/rdisk/c0t6d0s2      /cdrom  hsfs      0      no      ro
> mount -o ro -F hsfs /dev/dsk/c0t6d0s2 /cdrom (as root)
(See the section on sudo)

```

```

---[Display and upgrade Microcode (Firmware) for Volume Manager]
(Don't do this on a running system)
> ssaadm display c1
> ssaadm download -f /usr/lib/firmware/ssa/ssafirmware c1

```

```

=====[::Information]

```

```

---[Books]

```

```

Configuration and Capacity Planning for Solaris Servers - Brain Wong
Sun Performance and Tuning, Java and the Internet - Adrian Cockcroft,
Richard Pettit
KNRC The C Programming Language (Addison Wesley) - Second Edition
Advanced Programming in the UNIX Environment (Addison Wesley) - Stevens

```

```

---[SUN]

```

```

800-872-4786 (USA4SUN)
- Create new profile
- SUN Service ID
- Contract #
- Name
- "Open New Case"

```

```

=====[::Networking]

```

```

---[Tcplogger]

```

```

/opt/sbin/tcplogger -i le0 -a

```

```

---[::snoop ]

```

```

> snoop between river1 river2

```

```

---[compile ::scream]

```

```

> cd /src/scream
> gcc -o scream scr12.c

```

```

---[::syslog.conf]

```

```

> vi /etc/syslog.conf
> /etc/init.d/syslog stop
> /etc/init.d/syslog start

```

```

---[::logger]

```

```

There is a program called "logger" at /usr/bin/logger which allows you to send
messages to any priority listed in your syslog.conf, with messages. I can't
remember what you had in your syslog.conf, but if you had defined:

```

```

*.err;kern.debug;daemon.notice;mail.crit;user.none      /var/adm/messages

```

you could test it by running:

```
> logger -i -p mail.crit "testing mail.crit priority level"
```

And if you have it defined like that on one of the other machines, you can test it from stargate by using the ssh remote execution trick:

```
*.err;kern.debug;daemon.notice;mail.crit;user.none @melanoma.uchicago.edu  
> ssh voodoo logger -i -p mail.crit "testing mail.crit priority level"
```

After running these, just look at the output at the tail of the log file, and if your message appears, you are good to go. There is also a facility called "mark" which puts 20 minute timestamps in the log file, so you can have it timestamp between entries. So something like this in syslog.conf:

```
mark.crit /var/adm/messages
```

---[Multiple addresses]

[4.10) How can I have multiple addresses per interface?]

Solaris 2.x provides a feature in ifconfig that allows having more than one IP address per interface. Undocumented but existing prior to 2.5, documented in 2.5 and later.

```
==> ifconfig IF:N ip-address up
```

where "IF" is an interface (e.g., le0) and N is a number between 1 and <MAX>. Removing the pseudo interface and associated address is done with "ifconfig IF:N 0.0.0.0 down".

As with physical interfaces, all you need to do is make the appropriate /etc/hostname.IF:X file.

The maximum number of virtual interfaces, <MAX> above, is 255 in Solaris releases prior to 2.6. Solaris 2.6 and Solaris 2.5.1 with the Solaris Internet Server Supplement (SISS) allow you to set this value with ndd, upto a hard maximum of 8192.

```
/usr/sbin/ndd -set /dev/ip ip_addr_per_if 4000
```

There's no limit inspired by the code; so if you bring out adb you can increase the maximum even further.

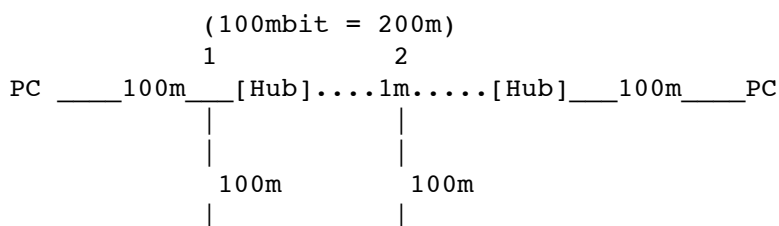
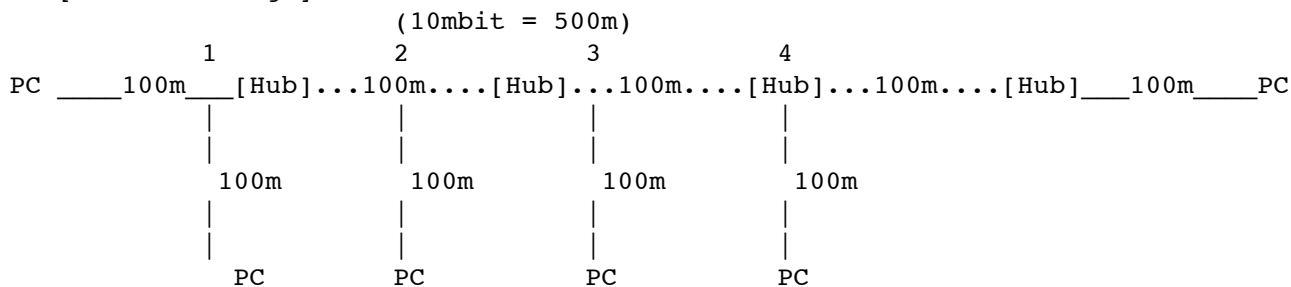
[4.12) Where can I get an SNMP agent for Solaris?]

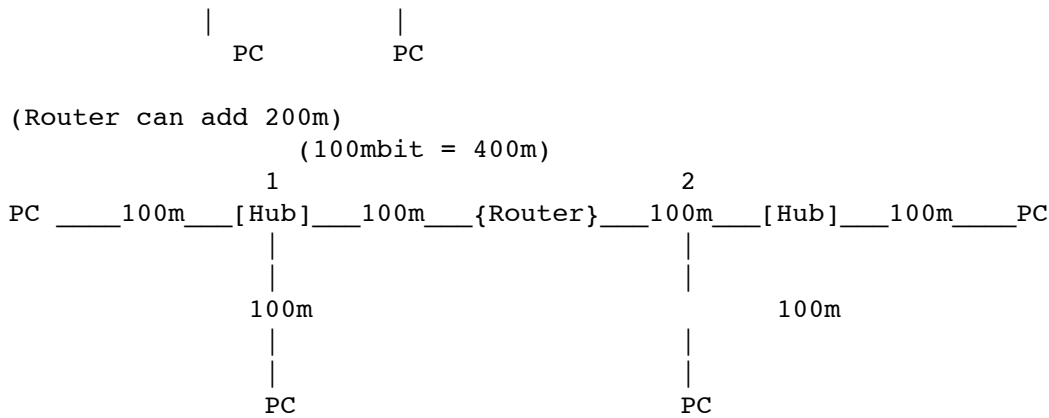
There are several agents available, including one from Sun.

```
ftp.ece.ucdavis.edu:/pub/snmp
```

Solstice Enterprise Agents, hit the "software download" icon Sun SNMP agents Solaris 2.6 ships with an SNMP agent.

---[Network Design]





====[::NFS]

---[Show who is nfs mounted] > ruser -a

---[Create an NFS Server] - Solaris

```

> cd /etc/dfs
> vi dfstab
/usr/sbin/share -F nfs -o rw=seven.uchicago.edu -d "Debian CD Rom" /cdrom/
debian2.0binary
> share -F nfs -o rw /cdrom/debian2.0binary
> share -F nfs -o rw /var/spool
> /etc/init.d/nfs.server [stop/start] or
> cd /etc/init.d
> share
Server          Client
mountd nfsd      statd lockd
/etc/dfs/dfstab /etc/vfstab
/etc/dfs/sharetab /etc/mnttab
share unshare    mount unmount
shareall unshareall mountall unmountall
-> /usr/lib/nfs/mountd

```

```

> share -F nfs -o rw,ro=zorg /export/home
> vi /etc/dfs/dfstab
> mount -F nfs -o rw lilu:/export/home /export/home

```

```

Server_____Client
nfsd          biod
mountd
lockd         lockd
statd         statd

```

```

> nfsstat -scrnz
> showmount -ade3

```

(See the following Tables)
net:309,311,315,319

---[Linux]

(on NFS server)

```

> vi /etc/exports
/home/scot *orion.com(rw,no_root_squash) *ns248.comm.mot.com(rw,no_root_squash)
> /etc/rc.d/init.d/nfs restart
> exportfs -ra

```

```
(on NFS client)
> vi /etc/fstab
ns528:/home/scot /home/air nfs user,exec,dev,suid,rw 1 1
lilu:/home /home/lilu nfs user,exec,dev,suid,rw 1 1
> mount -a
```

```
=====
[::Samba]
```

```
(Steps to setting up Samba 2.0.7 on Solaris 8 x86)
```

```
1. Install the following package.
```

```
> pkgadd -d samba-2.0.7-sol8-intel-local
```

```
2. Create the proper "smb.conf" file
```

```
> cd /usr/local/samba/lib
```

```
> vi smb.conf
```

```
# Samba config file created by Scot for Zorg
```

```
# Date: 2001/09/07 00:07:33
```

```
# Global parameters
```

```
[global]
```

```
workgroup = HOME
server string = Zorg - Samba 2.0.7
security = SHARE
map to guest = Bad User
keepalive = 30
os level = 2
kernel oplocks = No
```

```
[homes]
```

```
comment = home
path = /zorg
writeable = Yes
create mask = 0750
browseable = No
```

```
[printers]
```

```
comment = All Printers
path = /tmp
create mask = 0700
printable = Yes
browseable = No
```

```
[corrie]
```

```
path = /zorg/corrie
guest account = corrie
writeable = Yes
guest ok = Yes
```

```
[bianca]
```

```
path = /zorg/bianca
guest account = bianca
writeable = Yes
guest ok = Yes
```

```
3. Create the startup script.
```

```
> vi /etc/init.d/smb
```

```
#!/bin/sh
```

```

#
# Created by Scot for Zorg (Solaris 8.x86) on 9-6-01

PATH="/usr/local/samba/bin:$PATH"

case $1 in
    'start')
        echo "Starting smbd..."
        smbd -D
        echo "Starting nmbd..."
        nmbd -D
        ;;
    'stop')
        echo "Stopping smbd and nmbd..."
        /usr/bin/pkill -x -u 0 nmbd
        /usr/bin/pkill -x -u 0 smbd
        rm -f /usr/local/samba/var/locks/smbd.pid
        rm -f /usr/local/samba/var/locks/nmbd.pid
        ;;
    *)
        echo "usage: smb {start|stop}"
        ;;
esac

```

4. Modify the startup scripts

(Make sure the permissions are correct)

```

> cp smb /etc/init.d/smb
> cd /etc/rc3.d
> cp smb /etc/rc3.d/S81smb {or ln -s ../init.d/smb S81smb}

```

```

=====
[::NIS & NIS+]
---client/server database system.
-Used to keep files on a single source:
Ex: /etc/passwd and /etc/group
- files must be tabular with at least one unique entry per line item; and raw
  text only
- files are converted to DBM database format
- a separate DBM needs to be created for each key to be searched.
Ex: /etc/passwd.byname /etc/passwd.byuid
- clients and servers bind themselves to a single domain
- once bound, clients broadcast to find the NIS server
- filename = ypserv-1.1.7-1.i386.rpm

```

```

[::NIS-Setup]
(Redhat 7.0 Version)
1. Set the NIS domain name
> domainname man
2. /etc/rc.d/init.d/ypserv.init looks in:
> vi /etc/sysconfig/network
NIS_DOMAIN="man"
3. Decide where the database files will be located, and edit the Makefile:
> vi /var/yp/Makefile
PASSWD = $(YPPWDDIR)/passwd
SHADOW = $(YPPWDDIR)/shadow
ALIASES = $(YPPWDDIR)/aliases
GROUP = $(YPPWDDIR)/group

```

```
HOSTS    = $(YPPWDDIR)/hosts
```

(the following need to be distributed:

```
all: passwd hosts group netid protocols rpc services mail ypservers)
```

```
# all: # precedes comments.
```

```
all: passwd shadow aliases group hosts netid protocols rpc services mail \
ypservers
```

4. Initialize the NIS database.

```
> /usr/lib/yp/ypinit -m
```

```
- ***No rule to make target means a file is missing
```

```
- Once you solve the problem run:
```

```
> cd /var/yp;make
```

5. Configure the NIS client

```
> vi /etc/yp.conf
```

```
domainname man
```

```
ypserver lulu.orion.com
```

```
> vi /etc/domainname
```

```
man
```

```
> vi /etc/rc.d/rc.local
```

```
# Setup NIS domain name
```

```
if [ -f /etc/domainname ]; then
```

```
    domainname 'cat /etc/domainname'
```

```
    echo "NIS domain: 'domainname'"
```

```
fi
```

```
> vi /etc/nsswitch.conf
```

```
passwd:          files nis
```

```
shadow:         files nis
```

```
hosts:          files nis [NOTFOUND=return] nis+
```

```
netgroup:       nis
```

```
- RH - the standard C library comes with the NIS resolver "ypbind" built in.  
  ypwhich will not work though without ypbind, but you can test with yp  
  commands like ypcat
```

```
- RH - /etc/passwd does not need the following NIS entry:
```

```
+:0:0:::
```

```
(fingerd does though)
```

6. RH - Don't need to reboot

```
> domainname 'cat /etc/domainname'
```

```
- reboot at earliest convenience to verify setup
```

7. Test

```
> ypcat passwd
```

```
{Configure Secondary NIS Server}
```

8. First configure the machine as a NIS client.

```
> ypcat passwd to test this machine
```

9. Install ypserv-1.xxx.i386.rpm

```
> /etc/rc.d/init.d/ypserv.init start
```

10. Install ypbind-1.xxx.src.rpm

```
- After installed, it must be compiled
```

```
> cd /usr/src/redhat/SOURCES
```

```
> tar -xzvf ypbind-3.0.tar.gz
```

```
> cd ypbind
```

```
> patch < ../ypbind-3.0-glibc.diff
```

```
- The Makefile assumes you use the ns1 libraries. RedHat doesn't use  
  this, so remove it.
```

```
> mv Makefile Makefile.ns1;grep -v ns1 Makefile.ns1 > Makefile
```

```
> make
```

```
> cp ypbind /usr/lib/yp;chmod 700 /usr/lib/yp/ypbind
```

```

> /etc/rc.d/init.d/ypbind start
- test which maps are being served
> ypwhich -m
RH - ypbind is not needed by the client, but ypinit requires it
11. Tell the master server that a slave server exists
> vi /var/yp/ypservers
(This could have also been done while running ypinit -m on the master)
12. Set the NIS domainname in the startup sequence.
> vi /etc/sysconfig/network
NIS_DOMAIN=orion.com
13. Set the domainname without rebooting
> domainname orion.com
14. Initialize the slave server
> /usr/lib/yp/ypinit -s lilu.orion.com
- Ignore - "Trying ypxfrd...not running". This occurs because the master
server hasn't been setup to run the YP map transfer daemon "rpc.ypxfrd".
Don't set it up anyway. Use a server push method to update the domain.
15. Setup the Master NIS server to do a push.
{On the NIS server}
> vi Makefile
(Comment out the NOPUSH line)
#NOPUSH="True"
DOMAIN = 'basename\'pwd\'
DOMAIN = '/bin/orion.com'
16. Test
{on the NIS master server}
> cd /var/yp; make all
{on the NIS slave server}
> vi /etc/yp.conf
ypserver air.orion.com
> ypcat passwd
- Global user database characteristics is implicit in RedHat
(/etc/passwd by default is configured with a "+:~:~:~:~:")
17. Limit access to a particular machine
> vi /etc/passwd
+scotc::~::~:/bin/bash
+:~:~:~:~:/bin/false
18. Netgroups. Group people and machines for access control.
- Create a netgroup for system administrators
> vi /etc/netgroup
#groupname      memberlist or tuple
admins  scot it
# tuple examples
# (hostname, username, domainname)
# (air,,,) - blank is considered a wildcard
# (-,scot,) - only scot
it      (-,scot) (-,lamar,)
19. Reference the netgroups in /etc/passwd
> vi /etc/passwd
+@admins

---[daemons]
{NIS}

{NIS+}
rpc.nisd

```

```
---[commands]
{NIS}
-(Solaris 1.x)
ypcat
ypmake
ypserv
ypupdated
ypxfrd
-(Solaris 2.x)
ypbind
ypcat
ypinit
ypmatch
yppasswd
yppoll
ypset
ypwhich
ypxfr
```

```
{NIS+}
nesaddent
nis_cachemgr
nisaddcred
nisbladm
niscat
nischgrp
nischmod
nischown
nischtti
nisdefaults
nisgrep
nisgrpadm
nisinit
nisl
nisl
nismatch
nismkdir
nispasswd
nisping
nism
nismrmdir
nissetup
nisshowcache
nisupdkeys
```

```
---[files]
/var/yp
/var/yp/aliases
/var/yp/Makefile
```

```
---[Disable NIS]
> cd /etc
> mv defaultdomain _defaultdomain
> mv defaultdomain.old _defaultdomain.old
> mv defaultrouter _defaultrouter
> mv hostname.le0 _hostname.le0
> mv hostname.* _hostname.*
```

```

- replace /etc/nsswitch.conf with /etc/nsswitch.files
> boot -s
> ypcat -k hosts |grep ash
> ypcat -k aliases |grep ^scotc
> ypwhich -m |grep home
> ypmatch wieden auto_home.sa
> ypmatch nis_admin
> ypmatch scotc passwd
> arp -a (Troubleshoot)
> ls -la (Troubleshoot)

```

Notes: A NIS server is no longer available under Solaris2.x. Under Solaris 2.x look in the /usr/sbin and /usr/lib/nfs for useful programs and commands. Finally, some of missing commands are in the BSD Compatibility package.

Created - May.21.2001

[sah - ch.18, pg.385]

[NIS]

18.386 [System files that are commonly shared]

Filename	Function
/etc/passwd	User account information database
/etc/group	UNIX group definitions
/etc/hosts	Maps between hostnames and IP addresses
/etc/networks	Associates text names with IP network numbers
/etc/services	Lists port numbers for well-known network services
/etc/protocols	Maps text names to protocol numbers
/etc/ethers	Maps between hostnames and Ethernet addresses
/etc/aliases	Electronic mail aliases
/etc/rpc	Lists ID numbers for RPC services
/etc/netgroup	Defines collections of hosts, users, and networks

Most server-based systems are setup to work with only these and perhaps a few additional files.

- The passwd file is searched with the getpwuid, getpwnam, and getpwent routines. These routines take care of opening, reading, and parsing the passwd file so that user-level programs don't have to.

18.387 There are two models. "push" and "pull".

Push - The master server distributes the files periodically. Files, lists of clients, update scripts, and timetables are all stored in one place. Making the scheme easy to control.

Pull - In the pull system, each client is responsible for updating itself. This is less centralized, but more adaptable and more secure. Best if going across political boundaries. (i.e. different groups or subdomain).

::rdist - In most cases, this command is the best way to distribute files from a central server. Similar to make. Only copies files when they are out of date. Like make. rdist looks for a control file (distfile or Distfile).

```

> rdist -f <distfile> (specifies a specific file)
(distfile format)

```

18.390 ::expect - Pulls files

18.392 ::NIS - The Network Information Service

=====

```
===[::Routing]
```

```
- Steps to setup a router on Redhat 7.1
```

```
Router:      lilu.orion.com
            eth0  192.168.1.1
            eth1  19.9.9.2
```

```
1. > vi /etc/sysconfig/network
```

```
NETWORKING=yes
HOSTNAME=lilu.orion.com
GATEWAYDEV="eth1"
GATEWAY="19.9.9.2"
FORWARD_IPV4="yes"
```

```
2. vi /etc/sysconfig/routed
```

```
EXPORT_GATEWAY="yes"
SILENT="no"
```

```
3. > vi /etc/sysconfig/static-routes
```

```
eth0 net 19.9.9.2 netmask 255.0.0.0 gw 192.168.1.1
eth1 net 192.168.1.1 netmask 255.255.255.0 gw 19.9.9.2
```

```
4. "netcfg" looks like this:
```

```
X      Network Packet Forwarding
```

```
Default Gateway      19.9.9.2
```

```
Default Gateway Device eth1
```

```
Interface-----Network Address-----Netmask-----Gateway-----
eth0      19.9.9.2      255.0.0.0   192.168.1.1
eth1      192.168.1.1    255.255.255.0  19.9.9.2
```

```
5. Enable the routed daemon at startup (reboot?)
```

```
> ntsysv
```

```
X routed
```

```
---(Client Setup)
```

```
6. > vi /etc/defaultrouter
```

```
19.9.9.2 <or> (depending on network)
```

```
192.168.1.1
```

```
7. > route add default 192.168.1.1 1
```

```
====[::Printing]
```

```
---[Having trouble printing, - "disk is write protected" ]
```

```
> /var/spool/lp (delete all *.0 files)
```

```
> rm log
```

```
> cp /dev/null lastlog
```

```
> cp /dev/null messages
```

```
> cp /dev/null wtmpx
```

```
> /etc/init.d/lp start
```

```
---[printing/other problems?] - /var/spool/adm - delete files
```

```
---[Setup an HP Printer]
rlogin printserver
> vi /opt/named/hosts.db
> vi /opt/named/hosts.rev
> jetadmin
> (1) modify configuration
it.rpc.hmco.com
> (1) create printer configuration
```

```
[Solaris 2.5x Printing commands]
```

```
{directories}
- /usr/lib/lp/postscript = default postscript filter
- /etc/lp/fd = print filter descriptor files
- /etc/lp/filter.table = filter lookup table
- /usr/share/lib/terminfo /e = to see if the printer is listed
- /etc/lp/interfaces/[printer] = queue
- /var/spool/lp/requests = scheduler places a copy of the job here
```

```
{programs/daemons}
```

```
- lpsched
- lpadmin
- lpstat
```

```
> lp -d carqms1 -o nb -y font=courier7 -y landscape -y gaudy -y title="whatever"
filename
> /bin/lprequest nslj1-31(job number)
> lpfilter -f all -l
> /etc/lp/printers/nslj1/configuration
content type is postscript
> lpfilter -f nslj1 -x
> rm /etc/lp/fd/nslj1.fd
> banner test truss | lp -d
```

```
[Printer setup]
```

```
Instructions for adding/using printers via ethernet for System V.
(This document assumes that the printer has the hostname
syshp4k.uchicago.edu and you want to add it as syshp4k on your
machine)
```

```
{Solaris 2.5.1}
```

```
(first time only):
```

```
> cd /etc/lp/fd
> for filter in *.fd ; do
_ name=`basename $filter .fd`
_ lpfilter -f $name -F $filter
_ done
> lpsystem -t bsd -y "Printer in NSIT-AIS" syshp4k.uchicago.edu
> lpadmin -p syshp4k -s syshp4k.uchicago.edu -T PS -I postscript
> lpadmin -d syshp4k (optional -- sets syshp4k as default printer)
> accept syshp4k
> enable syshp4k
```

```
Now you can use lp to print to it, and lpstat -p to check its status.
```

```
{Solaris 2.6 & 7}
```

```
(first time only):
```

```
cd /etc/lp/fd
```

```

for filter in *.fd ; do
name=`basename $filter .fd`
lpfilter -f $name -F $filter
done
lpadmin -p nslj1 -v /dev/null
lpadmin -p nslj1 -i /usr/lib/lp/model/netstandard
lpadmin -p nslj1 -o dest=nslj1.comm.mot.com -o protocol=bsd
lpadmin -p nslj1 -I postscript -T PS
accept nslj1
enable nslj1
lpadmin -d nslj1
vi /etc/lp/interfaces/nslj1
nobanner=yes
/etc/init.d/lp stop
/etc/init.d/lp start

```

[Install HP Jetadmin]

```

> pkgadd -d jetadmin
> /opt/hpnp/jetadmin &
_> 1 Configuration
_ NIC Address: 0060B092C3CD

```

```

Card IP Address: syshp4k.uchicago.edu (128.135.168.66)
Network Mask: 255.255.255.0
Default Gateway: atm-1155-gw.uchicago.edu (128.135.168.1)
_> 3 Add printer
_> syshp4k
_> modify settings (5 for default)
_> 0 (To configure)
[Reboot to take effect]

```

====[::Programming]

```

---[C programming]
.h - header file
.c - c source file
.o - compiled object file

```

```

> ./configure --prefix=/opt/bin

```

---[C program basics] - an executing program is called a process

- An executable program can be broken down into the following parts:

- 1 text - Contains sharable program code (machine language)
- 2 data - Contains initialized variables within the program
- 3 BSS - Contains the uninitialized variables within the program
- 4 Stack - Handles the local variables and parameters passed via function calls

5 User Area - Maintains process information used by the UNIX kernel

* This is the only part of a process that may be shared by other processes. The other segments are entirely local to each individual process.

Ex:

```

#include <studio.h>
#include "function_x.h"

```

```

int main (void)
{
    int i = 0;

```

```

int x;
char *ch_ptr;

for (i=0;i<100;i++)
    printf("Hello World!!!\n");
*ch_ptr = "Hello World!!!";
for (x=0;x<100;x++)
    printf("%s\n",ch_ptr)
function_x(100);
return 0;
}

```

From the preceding program you see the following:

Text This is the compiled hello.c in machine language format

Data int i = 0

BSS int x and char *ch_ptr. (Later in the program, you assign this pointer the value of Hello World!!!)

Stack Temporary data storage used by the process. The stack contains an entry of 100. This is the value you are passing to the function/subprogram function_x.

User Information such as real and effective user IDs, real and effective group ID's, area pointers to additional process structures, and accounting information to name a few. This area is used by the UNIX kernel to manage the process during execution.

- Familiarize yourself with the header files on your system. (*.h). Some of the best documentation resides in /usr/include.

[processes] - UNIX automatically opens (and closes at termination) three files for each process 0,1,2. That is stdin, stdout, and stderr, respectively.

Process state list:

SIDL (idle) A new UNIX process is created by a call to fork, vfor, and exec. The process is available to be scheduled to run.

SRUN (run) The process exists on a run queue.

SSLEEP (sleep) A process is "put to sleep" or suspended. This usually occurs when the process is waiting on a system resource (for example, a disk subsystem)

SSTOP (stop) An executing process is stopped. Either by a signal or the parent.

SZOMB (zombie) Having exited, the process no longer exists, but leaves behind for the parent process some record of its termination status.

(See p139-140 of USI for explanation)

> ps -efc - shows process class

====[::Scripting]

---[while script]

> csh

> while 1

> ls -l all.tar

> sleep 60

> end

____SHELL programming____

The Korn Shell

- The official version (ksh - www.kornshell.com)

- The Public Domain version (pdksh - <ftp://ftp.cs.mun.ca:/pub/pdksh>)

- The Desktop version (dtksh)

[Pipes]

==> command1 | command2 | ...

Ex: > tar -cf - ./foo | {cd /tmp}; tar -xf - }

(The above is a way for root to copy files without changing ownership)

[Lists]

==> command1; command2; command3 ...

==> command1 | command2 | command3 | ...

==> command1 && command2

==> command1 || command2

Ex: > mkdir docs && cd docs

Ex: > grep root /etc/passwd || echo "Help! No one in charge!"

[Redirection]

==> command > file, or command >> file (to append)

==> list > file

Ex: > {date;uptime;who;} > mylog

==> command > file 2>&1

==> list > file 2>&1

Ex: > rm -rf /tmp/my_tmp_dir > /dev/null 2>&1; mkdir /tmp/my_tmp_dir

==> command >> file 2>&1

==> list >> file 2>&1

Ex: > mail someone@somewhere.com < /home/scot/someletter

[Subshells]

Any list can be executed in the current shell environment or in a subshell.

A list enclosed in braces { }.

==> { list;} - executed in the current shell (Best to run in current shell)

==> (list;) - executed in a subshell

There are instances where using a subshell is useful or required. One reason to use subshells is because they effectively make all variables local. This is illustrated by the following:

FRUIT="banana"; (FRUIT="watermelon"; echo \$FRUIT;) ; echo \$FRUIT;

>> watermelon

>> banana

Another reason would be to change the working directory.

Ex: > pwd; (cd /tmp pwd) ; pwd;

>> /home/scot

>> /tmp

>> /home/scot

[Variables] - Variables are "words" that hold a value.

==> name=value - defines the variable

Ex: > FRUIT=peach

==> command \$name - to access it

Ex: > echo \$FRUIT

[Array Variables]

The Bourne shell only supports "scaler" variables. Korn and Bash support array.

Array variables can be set in two ways.

The first form sets a single element:

==> name [index]=value

> FRUIT [0] =apple

> FRUIT [1] =banana

> FRUIT [2] =orange

> FRUIT [9] =alsogood

Strings can also be used.

```
FRUIT [apple] = 10
```

The second form of array initialization is used to set multiple elements at once:

```
(ksh) > set -A name value1 value2 ...
```

```
(bash) > name=(value1 value2 ...)
```

When setting multiple array elements in bash, an array index can be placed before the value:

```
> myarray=([0]=derri [3]=gene [2]=mike [1]=terry)
```

There are no requirements that the array indices be in order, as shown in the preceding example, or that the indices be integers. (This feature is not present in ksh)

After an array variable has been set, you access it as follows:

```
==> ${name[index]}
```

```
Ex: > echo ${FRUIT[2]} >> orange
```

All the items in an array can be accessed in one of the following ways:

```
==> ${name[*]}
```

```
==> ${name[@]}
```

Here, name is the name of the array you are interested in. If the FRUIT array was initialized as given previously, the command.

```
> echo ${FRUIT[*]}
```

```
>> apple banana orange
```

If any of the array items held values with spaces, then this form of array access would not work, and you would need to use the second form. The second form quotes all the array entries so that embedded spaces are preserved.

[Environment Variables]

Exporting can be done in two ways:

```
==> export name (only form supported by sh)
```

```
Ex:> PATH=/bin:/sbin ; export PATH
```

```
==> export name=value
```

```
Ex:> export PATH=/sbin:/bin
```

In bash and ksh, any combination of name or name=value pairs can be given to the export command:

```
Ex:> export FMHOME=/usr/frame CLEARHOME=/usr/atria PATH
```

____Table 7.1 Shell Variables____

PWD Current working directory.

UID Current user id.

SHLVL Increments by one each time bash is started. Used to determine if built-in exit will work.

REPLY Expands to the last input line read by the read built-in command when it is given no arguments. Not available in sh.

RANDOM Generates a random integer (0-32767). Not available in sh.

SECONDS Each time this parameter is referenced, the number of seconds since shell invocation is returned.

IFS The internal field separator that is used by the parser for word splitting after expansion. \$IFS is also used to split lines into words with the read built-in command.

PATH Well duh.

HOME Ditto duh.

(unsetting a variable)

```
==> unset name
```

```
==> unset -v name
```

[Command Substitution]

Mechanism by which the shell performs a given set of commands and then substitutes their output in the place of the commands. Command substitution is performed when a command is given as: `command`

Here command can be a simple command, a pipeline, or a list. Command substitution is generally used to assign the output of a command to a variable. Each of the following examples demonstrates command substitution:

```
DATE=`date`  
USERS=`who | wc -l`  
UP=`date ; uptime`
```

[Arithmetic Substitution]

In ksh and bash, the shell allows for integer arithmetic to be performed. This avoids having to run an extra program such as `expr` or `bc` to do math in a shell script.

```
==> $((expression))
```

```
Ex:> foo=$(( (5+3*4) -4) /2 )
```

Sets the value of `foo` to 3. Because this is integer arithmetic, the value is not 3.5, and because of operator precedence, the value is not 6.

[Quoting]

Strings enclosed in single quotes, `'string'`, have all the special characters in them disabled. Strings enclosed in double quotes, `"string"`, have all the special characters except `!`, `$`, ```, `\`, and `{` disabled. Characters preceded by a backslash (`\`) have their special meaning disabled.

In addition to quoting, several standard escape sequences familiar to C language programmers, such as `\t` for tab and `\n` for newline, are recognized.

[Flow Control]

- the `if-fi` block
- the `case-esac` block

The `if` statement is normally used for the conditional execution of commands, whereas case statements allow any of a number of command sequences to be executed depending on which one of several patterns matches a variable first. It is often easier to write `if` statements as case statements if they involve matching a variable to a pattern.

[if-fi block]

```
if list1 ; then  
    list2  
elif list3 ; then  
    list4  
else  
    list5  
fi
```

Most often, the list given to an `if` statement is one or more test commands, which can be invoked by calling the `test` command as follows:

```
==> test expression
```

```
==> [expression]
```

Table 7.2 TEST Options

<code>-d file</code>	True if file exists and has a directory
<code>-e file</code>	True if file exists
<code>-f file</code>	True if file exists and is a regular file
<code>-k file</code>	True if file exists has its sticky bit set
<code>-L file</code>	True if file exists and is a symbolic link
<code>-r file</code>	True if file exists and is readable
<code>-s file</code>	True if file exists and has a size greater than zero
<code>-t file</code>	True if fd is opened on a terminal
<code>-w file</code>	True if file exists and is writable
<code>-x file</code>	True if file exists and executable

```

-O file                True if file exists and is owned by the effective user ID
file1 -nt file2       True if file1 is newer (according to modification date) than
                    file2
file1 -ot file2       True if file1 is older than file2
-z string             True if the length of string is zero
-n string             True if the length of string is non-zero
string1 = string2
string1 == string2    True if the strings are equal
string1 != string2    True if the are not equal
! expr                True if expr is false. The expr can be any of the tests
                    given previously
expr1 -a expr2        True if both expr1 AND expr2 are true
expr1 -o expr2        True if either expr1 OR expr2 is true
arg1 OP arg2          OP is one of -eq, -ne, -lt, -le, -gt, or -ge. These
                    arithmetic binary

```

```

Ex:> if [-d $HOME/bin] ; then PATH="$PATH:$HOME/bin" ; fi
(Test is used to see if the directory exists, then take an action)
Ex:> if [-s $HOME/.bash_aliases] ; then . $HOME/.bash_aliases ; fi
(Test is used to determine whether a file exists and has non-zero size before
any action is taken)
Ex:> if [ -z "$DTHOME" ] && [-d /usr/dt] ; then DTHOME=/usr/dt ; fi
(This form is obvious what tests are being done and what the evaluation criteria
are.)
Ex:> if [ -z "$DTHOME" -a -s /usr/dt ] ; then DTHOME=/usr/dt ; fi
(This form only invokes the [ command once and may be marginally more
efficient.)

```

```

[case-esac Block]
==>
case word in
    pattern)
        list
    pattern2)
        list2
    ;;
esac

```

In this form, word is either a string or a variable, whose value is compared against each pattern until a match is found. The list following the matching pattern is executed. After a list is executed, the command ;; indicates that program flow should jump to the end of the entire case statement. This is similar to break in C.

If no matches are found, the case statement exits without performing any action. Some default actions can be performed by giving the * pattern, which matches anything.

There is no maximum number of patterns, but the minimum is one. The patterns can use the same special characters as patterns for pathname expansion, along with the OR operator, |. The ;; signifies to bash that the list has concluded.

```

[read] - Prompt the user for input and then read the user's response.
==> read name
(Reads the entire line of user input until the user presses Enter, and makes
that line the value of the variable specified by name.
Ex:>
YN=yes
printf "Do you want to play a game [$YN]?"

```

```

read YN
: ${YN:=yes}
case $YN in
  [yY]|[Yy][eE][sS]) exec xblast ;;
  *) echo "Maybe later." ;;
esac

```

[LOOPS] - Types of loops:

```

= for (sh) - Used when a set of commands needs to be executed repeatedly.
= while (sh) - Used when commands must run while a condition is true
= until (ksh, bash) - Used when commands must run until a condition ends
= select (ksh, bash) - Provides a selection interface

```

[for loop]

==>

```

for name in list1
do

```

```

    list2

```

```

done

```

(The variable specified by name is set to each element in list1, and list2 is executed for each element of list1.)

[while loop]

==>

```

while list1
do

```

```

    list2

```

```

done

```

(list1 is evaluated each time, and as long as it is true, list2 is executed. This allows for infinite loops to be written with /bin/tru or : as list1.)

Ex:>

```

x=1

```

```

while [$x -lt 10]
do

```

```

    echo $x

```

```

    x=$((x+1))

```

```

done

```

Ex:> This while loop copies its input to its output, like the cat program:

```

while read
do

```

```

    echo $REPLY;

```

```

done

```

```

done

```

(If input redirection is used, this loop writes the contents of the input file to the standard output, similar to cat.)

[until loop] - A variation on the while loop.

==>

```

until list1
do

```

```

    list2

```

```

done

```

(In general, the until loop is not favored because it can be written as the negation of a while loop.)

[select loop] - Easy way to create a numbered menu.

==>

```

select name in list1

```

```
do
    list2
done
```

Ex:> This loop displays a number list of the files in the directory /tmp and runs an ls -l on file that exist:

```
select file in /tmp/* QUIT
do
    if [-e $file] ; then
        ls -l $file
    else
        break
    fi
done
```

>>

```
1) /tmp/java                6) /tmp/job.control.ms
... etc...
```

#?

(Where #? is the prompt at which a number is typed by the user.)

[example script]

```
> vi new-script
#!/bin/sh
echo `who | wc -l` Users
> chmod a+x ./new-script
```

[Special Variables]

Special variables are set automatically by the shell and are available to the shell programmer on a read-only basis. Thus the programmer can read the value stored in these variables, but he cannot set their values.

Variable substitution enables the shell programmer to manipulate the value of a variable based on its state. Variable substitution falls into two categories. Namely, actions taken when a variable is set, and when it is unset.

The actions range from one-time value substitution to aborting the script.

____Table 8.1 - Special Shell Variables____

Var.	Description
\$0	The name of the command being executed. For shell scripts, this is the path with which it was invoked.
\$n	These variables correspond to the arguments with which a script was invoked. Here n is a positive decimal number corresponding to the position of an argument. (The first argument is \$1, the second is \$2, etc.)
\$#	The number of "positional" parameters supplied to a script.
\$*	All the arguments double quoted. If a script receives two arguments, "\$*" is equivalent to "\$1" "\$2".
\$@	All the arguments individually double quoted. If a script receives two arguments, "\$@" is equivalent to "\$1" "\$2".
\$?	The exit status of the last command executed.
\$\$	The process number of the current shell. For shell scripts, this is the process ID under which they are being executed.
\$_	The process number of the last background command.

Ex:>

```
#!/bin/sh
case $0 in
    *listtar) TARGS="-tvf $1" ;;
```

```

    *maketar) TARGS="-cvf $1.tar $1" ;;
esac
tar $TARGS
(The tar file to read or create is specified as the first argument, $1. The
script was called mytar, and had links listtar and maketar to it.)
> ln -s mytar listtar; ln -s mytar maketar;

```

```

[usage statement]
==> echo "Usage: $0 [options][files]"
#!/bin/sh
case $0 in
    *listtar) TARGS="-tvf $1" ;;
    *maketar) TARGS="-cvf $1.tar $1" ;;
    *) echo "Usages: $0 [file|directory]" exit 0 ;;
esac
tar $TARGS

```

(If run with only mytar, we get this:)

```
>> Usage: mytar [file|directory]
```

This is not what we want because it does not give the user any idea that the problem encountered was that the name of the script was wrong. We can change this by hard coding the valid names in the usage statement or we can change the script to use its arguments to decide in which mode it should run. In order to demonstrate the use of arguments, the next section will do the latter.

```

[arguments]
To illustate the use of arguments, let's change the script to use its first
argument, $1, as the mode argument and $2 as the tar file to read or create.
Our case statement becomes the following:
USAGE="Usage: $0 [-c|-t] [file|directory]"
case "$1" in
    -t) TARGS="-tvf $2" ;;
    -c) TARGS="-cvf $2.tar $2" ;;
    *) echo "$USAGE" ; exit 0 ;;
esac
tar $TARGS

```

```

[basename] - Change the variable $USAGE (which gives the full path), to:
==> USAGE="Usage: `basename $0` [-d|-t][file|directory]"
You could also do this:
was    ==> case $0 in
better    ==> case `basename $0` in

```

The simplest method for checking the necessary number of arguments is to see if the number of given arguments, \$# , matches the number of required arguments.

```

Ex:>
#!/bin/sh
USAGE="Usage: `basename $0` [-d|-t][file|directory]"
if [ $# -lt 2 ] ; then echo "$USAGE" ; exit 1; fi
case "$1" in
    -t) TARGS="-tvf $2" ;;
    -c) TARGS="-cvf $2.tar $2" ;;
    *) echo "$USAGE" ; exit 0 ;;
esac
tar $TARGS

```

The script deals only with the first file that is given as an argument and does

not check if the file argument is really a file.
Add the processing of all file arguments by using the special shell variable \$0. Let's start with the -t (list contents) option.

```
Ex:>
case "$1" in
  -t) TARGS="-tvf"
      for i in "$@" ; do
        if [-f "$i"] ; then tar "$STARGS" "$i" ; fi;
      done ;;
  -c) TARGS="-cvf $2.tar $2" ;
      tar "$TARGS" ;;
  *) echo "$USAGE" ; exit 0 ;;
esac
```

(Caution: \$* expands each argument with preserving quoting. \$@ expands each argument as it was quoted on the command line.)

All the arguments given to the script, including the first argument, \$1, are considered as files.

To remove the first argument from the list of arguments, we use the shift command. A similar change to the make mode of the script is also required.

Another issue is what the script should do when an operation fails.

Because the shell sets the variable \$? to the exit status of the most recent command, we can use that to determine if a tar operation failed.

Resolving the above issues, our script is as follows:

```
#!/bin/sh
USAGE="Usage: `basename $0` [-d|-t][file|directory]"
if [ $# -lt 2 ] ; then echo "$USAGE" ; exit 1 ; fi
case "$1" in
  -t) shift ; TARGS="-tvf" ;
      for i in "$@" ; do
        if [-f "$i"] ; then
          FILES=`tar "$TARGS" "$i" 2> /dev/null`
          if [ $? -eq 0 ] ; then
            echo ; echo "$i" ; echo "$FILES"
          else
            echo "ERROR: $i not a tar file."
          fi
        else
          echo "ERROR: $i not a tar file."
        fi
      done ;;
  -c) shift ; TARGS="-cvf" ;
      tar "$TARGS" archive.tar "$@" ;;
  *) echo "$USAGE" ; exit 0 ;;
esac
exit $?
```

[variable substitution]

==> \${parameter:-word} If parameter is null or unset, word is substituted for parameter.

 The value of parameter does not change.

```
Ex:> PS1=${HOST:-localhost}"$ " ; export PS1 ;
```

(Performs substitution only when the variable is unset)

==> `${parameter:=word}` If parameter is null or unset, parameter is set to the value of word.

Ex:> `PS1=${HOST:`uname -n`} "$ " ; export PS1 HOST ;`

(The default string to use does not have to be a fixed string but can be the output of a command. If this substitution did not exist in the shell, the same line would have to be written as follows:

Ex:> `if [-z "$HOST"] ; then HOST=`uname -n` ; fi ; PS1="$HOST$ " ; export PS1 HOST;`

(The variable substitution form is shorter and clearer than the explicit form.

==> `${parameter:?message}` If parameter is null or unset, message is printed to standard error.

 This is used to check that variables are set correctly.

Ex:> (from mytar.sh) : `${2:? "Insufficient Arguments, $USAGE"}`

In addition to using the variable substitution form previously described, we also are making use of a feature of the no operation command, `:`. This command evaluates the arguments passed to it. In this case, we pass the second argument of the script to it. If the second argument of the script is unset, and appropriate message prints to the screen and the script exits.

==> `${parameter:+word}` If parameter is set, then word is substituted for parameter.

 The value of parameter does not change.

Ex:> `echo ${DEBUG:+ "Debug is active."}`

(Used to print messages when a variable is set. This form does not alter the value of the variable, it alters only what is substituted. A frequent use is to indicate when a script is running in debug mode.)

[functions] - Mini shell scripts that enable a name to be associated with a set of commands.

==> `name () {list;}`

A function binds a name to the list of commands that compose the body of the function.

Ex:> `lsl () {ls -l ;} #valid`

The general format for the invocation of programs in UNIX is as follows:

==> `command [options] [file/parameters]`

(In shell scripts, it is desirable to adhere to this format)

[getopts]

==> `getopts option-string variable`

Here option-string is a string consisting of all the single character options getopts should consider, and variable is the name of the variable that the option should be set to. Usually the variable used is named OPTION.

Let's write a script that simplifies the task of uneneing a file.

The script should accept the following options:

-f to indicate the input filename

-o to indicate the output filename

-v to indicate the script should be verbose

Here's the command to implement these options:

> `getopts e:o:v OPTION`

This indicates that all the options except for -v require an additional parameter. The variables we need require are as follows:

- VERBOSE, which stores the value of the verbose flag. By default this will be false.

- INFILE, which stores the name of the input file
 - OUTFILE, which stores the name of the output file name. If this value is unset, decode uses the name supplied in the input file, and encode uses the name of the supplied input file and appends to it the .uu extension.
 The loop to implement these requirements is as follows:

```

VERBOSE=false
while getopts f:o:v OPTION ; do
    case "$OPTION" i
        f) INFILE="$OPTARG" ;;
        o) OUTFILE="$OPTARG" ;;
        v) VERBOSE=true ;;
        |?) echo "$USAGE" ; exit 1 ;;
    esac
done

```

We can make the script much more user friendly.

If we use the fact that getopts sets the variable OPTIND to the value of the last option that it scanned, we can have the script assume that the first argument after this is the input filename. If in this case the argument is not give, we should exit. Our error checking consosits of the followoin lines:

```

shift `echo "$OPTIND - 1" | bc`
: ${INFILE:=${1:? "Input file not specified."}}

```

Here is the complete script:

```

#!/bin/sh
USAGE="Usage: `basename $0` [-v] [-f] [filename] [-o] [filename]";
VERBOSE=false
while getopts f:o:v OPTION ; do
    case "$OPTION" i
        f) INFILE="$OPTARG" ;;
        o) OUTFILE="$OPTARG" ;;
        v) VERBOSE=true ;;
        |?) echo "$USAGE" ; exit 1 ;;
    esac
done
shift `echo "$OPTIND - 1" | bc`
: ${INFILE:=${1:? "Input file not specified."}} ${OUTFILE:=${INFILE}.uu}
if [ -f "$INFILE" ] ; then
    if [ "$VERBOSE" = "true" ] ; then
        printf "uuencoding $INFILE to $OUTFILE... \c"
    fi
    uuencode $INFILE $INFILE > $OUTFILE ; RET=$?
    if [ "$VERBOSE" = "true" ] ; then
        MSG="Failed" ; if [ $RET -eq 0 ] ; then MSG="Done." ; fi
        echo $MSG
    fi
fi

```

With this script, we can uuencode files in all the following ways (assuming the script is called uu):

```
> uu ch8.doc, or uu -f ch8.doc, or uu -f ch8.doc -o ch8.uu
```

[signals] - "Software interupts" that are sent to a program to indicate that an important "event" has occured. These events are a notification to the program that an unexpected problem has occured or that a user has asked the program to do something that is not usually in the flow of control. Due to their asynchronous nature, signals are one fo the more complicated tasks of shell

programming.

In UNIX, signals are represented by small positive integers, with each integer representing a specific type of event.

The complete list of all signals understood by the system are usually listed in the file /usr/include/signal.h.

___Important Signals for Shell Scripts___

SIGHUP	1	Hangup detected on controlling terminal or death of controlling process
SIGINT	2	Interrupt from keyboard
SIGQUIT	3	Quit from keyboard
SIGKILL	9	Kill signal
SIGTERM	15	Termination signal

[trap] - Used to set and unset the actions taken when a signal is received.

==> trap name signals

Here name is the set of commands or the shell function to execute when a signal in the list of specified signals is received. If name is not given, trap resets the action for the given signals to be the default.

There are three common uses for trap in shell scripts:

- Clean up temporary files
- Always ignore signals
- Ignore signals only during critical operations

Most shell scripts that create temporary files use sa trap commands similar to the following:

```
Ex:> trap "rm -f $TMPF; exit 2" 1 2 3 15
```

Sometimes when more complicated clean up is required, a shell function might be used. In order to make the uu script given in the previous section signal safe, we can add something similiae to the following into the beginnning of the script.

```
CleanUp() {  
    if [ -f "$OUTFILE" ] ; then  
        printf "Cleaning UP... " :  
        rm -f "$OUTFILE" 2> /dev/null ;  
        echo "Done." ;  
    fi  
}  
trap CleanUp 1 2 3 15
```

The main reason to use functions to handle signals is that it is more convenient to have a shell function invoked when a signal is received than to write in the appropriate code inline.

Also, the commands that should be executed when a signal is received might be different depending on which point in the script the signal was received. In many cases, it is difficult to capture that logic in a few commands, thus is it necessary to use a shell function as the signal handling routine.

See also:

==> trap `` signals

==> trap : singals

[debugging]

==> sh [option] [script]

==> set [option]

```
Ex:> sh -x ./uu test.tar
```

___Debugging Options___

```

-e Exits if a command fails
-n Reads all commands, but does not execute them
-u Treats all unset variables as an error when performing variable substitution
-v Prints all lines as they are read
-x Prints all commands and their arguments as they are executed
- Turns off all debugging flags

```

The "+" character preceding a line indicates the shell level in which each command was run.

[logging]

Two common techniques used to produce log files.

First, based on tee:

```

if ["$LOGGING" != "true"] ; then
    LOGGING="true" ; export LOGGING ;
    exec $0 | tee $LOGFILE
fi

```

(This if statement is usually one of the first commands in a script)

Second, based on script:

```
> script filename
```

To use script for logging, we trick it to use script as the shell:

```

if ["$LOGGING" != "true"] ; then
    LOGGING="true" ; export LOGGING ;
    exec script $0 $LOGFILE
fi

```

====[::Security]

---[Sample Commands to run on a Suspect System]

```

/root > last
/root > crontab -l
/root > cd /var/adm
/var/adm > ls
/var/adm > last -f ./wtmptx.1
/var/adm > cd /bin
/bin > find . -ctime -120 -ls
/bin > cd /dev
/dev > find . -type f -ls
/dev > ls MA*
/dev > ls -l | more
/dev > man MAKEDEV
/dev > cd /bin
/bin > strings ps | more
/bin > ifconfig -a
/bin > which ifconfig
/bin > cd /usr/sbin
/usr/sbin > ls

```

A quick check to ensure you do not have the Trojan horse installed is to execute the following command

```
> strings /bin/login | grep "HELO"
```

====[::Solaris Setup - 1.21.99]

[1] - INSTALL SOLARIS

[Solaris]

```
> init 6 as root or Stop-A and reset
```

```
[install Solaris]
ok > boot cdrom
> IP: 128.135.168.67
> Subnet: 255.255.255.0
> Domain: uchicago.edu
stargate                arthur
/                        /                        80
swap                    swap                    40
/usr                    /usr                    3900
/var                    /var                    40
/home                   /home                   10
```

```
> mkdir /usr/opt
> mv /opt /usr/opt
> rm -r /opt
> ln -s /usr/opt /opt
```

[x86 - Installation Notes for Solaris x86 on a PC (With Disk Update 11)]
 You will need a total of eight disks, and they will be used in the following order:

Boot Disk 1-3

Driver Disk 1-3

After the installation, before you start OpenWindows, install the Video Update

Video Disk 1-2

[DOS FAT and Solaris]

Start with a drive that is partitioned into 2 parts, make partition one active, don't do anything to partition two. Here is the boot sequence, which will depend on the install sequence:

(If you install Windows first, then install Solaris, the system will default to booting to Solaris, and the drive will look like the following before you install anything)

c:\ (formatted with DOS 6.2)

partition 2 - nothing on this

(If you install Solaris first, then install Windows, the system will default to booting to Windows after it finishes the install. It will look like the following before you install anything))

c:\ (formatted with DOS 6.2)

partition 2 - nothing on this,

- If you want to default to booting to Solaris when all is done, run FDISK and set partition two as active.

Partition the drive. If you start with a 2gig Drive, a good ratio is to split it evenly. Make partition 1 Active, leave partition alone.

Install Windows 95, and any additional software you want.

Install Solaris. Put Boot Disk 1 in the floppy drive, and the Solaris Cd in the CD-Rom. Follow the instructions as they appear. Choose the following:

Type the number that corresponds to your internal CD ROM as the device to boot from.

Interactive Install. (optional > bsh; > setprop boot-args "w install"; > run kernel/unix)

16 color display with scrolling capability. (This will be changed later).

9" monitor.

Integrated PS/2 compatible pointing device.

Update the Video Drivers. Do after the install is done, and the system is at the command prompt.

```

Stop Volume Manager. > /etc/init.d/volmgt stop
Use cpio to copy files off the diskette. > mkdir /tmp/Drivers
> cd /tmp/Drivers
> cpio -ifduBI /dev/rdiskette0 (Make sure Video Disk one is in the floppy)
>> Change to part 2 means to put Disk 2 in the drive
Merge the compressed image file > zcat vdullimage.Z | cpio -icudB
Run the install script. > ./installdu.sh
Start Volume Manager > /etc/init.d/volmgt start
Clean up. > cd / ; > rm -fr /tmp/Drivers
Configure a secondary display (such as a card in a docking station).
> /opt/SUNWnsdis/bin/kdmadd [kdmconfig -u = clears all settings;
kdmconfig -cf = changes primary display]
Start OpenWindows. > /usr/openwin/bin/openwin [ -display:1 or -display:2 ]
Install CDE (optional). (Note: a copy of the install manual is available at
/cdrom/solaris_2_5_1_desktop_1_1/CDE/x86/README/InstallGuide.ps)
> cd /cdrom/solaris_2_5_1_desktop_1_1/CDE/x86
> ./install-cde
To access DOS files within Solaris (i.e. the DOS/Windows 95 partition on the
drive you have Solaris installed):
> mount -F pcfs /dev/dsk/c0d0p0:c /mnt/dos
To make a permanent mount point type the following:
> vi /etc/vfstab
> /dev/dsk/c0d0p0:c /dev/rdisk/c0d0p1 /mnt/dos pcfs - no

```

[2] - EDIT CONFIGS

```

=====
__::/etc/passwd
scot:x:5545:11041:Scot Corrie:/home/scot:/usr/local/bin/bash

__::/etc/shadow

> mkdir /root
> mkdir -p /usr/local/home/scot
> ln -s /usr/local/home/scot /home/scot

__::$/ .profile
#
# Scot .profile - Jan. 13, 1999

stty istrip
#ENV=$HOME/.kshrc

PATH=/bin:/usr/sbin:/usr/bin:/usr/ccs/bin:/usr/ucb:/usr/local/bin:/usr/openwin/bin:/
usr/local/netscape:/usr/local/cdplayer:/etc:
MANPATH=/usr/local/man:/opt/man:/usr/share/man:/usr/openwin/share/man:/usr/dt/man
LD_LIBRARY_PATH=/usr/lib:/etc/lib:/opt/lib:/usr/local/lib:/usr/dt/lib:/usr/openwin/
lib:/opt/rvplayer5.0

export PATH MANPATH LD_LIBRARY_PATH ENV

#
# If possible, start the windows system
#
if [ "`tty`" = "/dev/console" ] ; then
    if [ "$TERM" = "sun" -o "$TERM" = "AT386" ] ; then

        if [ ${OPENWINHOME:-""} = "" ] ; then

```

```

        OPENWINHOME=/usr/openwin
        export OPENWINHOME
    fi

    echo ""
    echo "Starting OpenWindows in 5 seconds (type Control-C to interrupt)"
    sleep 5
    echo ""
    $OPENWINHOME/bin/openwin

    clear      # get rid of annoying cursor rectangle
    exit      # logout after leaving windows system

fi

EDITOR=vi
export EDITOR

#For bash
PS1="\u@\`hostname | cut -f 1 -d '.'`:\w> "
export PS1
#For ksh
#PS1=`whoami`'\@'\`hostname`'\: '$PWD> '

umask 022

LPDEST=syshp4k
export LPDEST

TERM=xterm
export TERM

DSM_DIR=/opt/IBMadsm-c
DSM_CONFIG=/opt/IBMadsm-c/dsm.opt
DSM_LOG=/var/log
export DSM_DIR DSM_CONFIG DSM_LOG

alias lss='ls -Fla'
alias lm='ls |more'
alias go='cd /home/scot/all'
alias jj='clear;ls -Fla'
alias disp='DISPLAY=stargate:0.0;export DISPLAY'
alias www='cd /opt/lib/apache/htdocs/'
alias sg='cd /home/scot/all/8-web/80-stargate/'

__::$/ .bashrc
if [ -f ~/.profile ] ; then
    . ~/.profile
fi

__::$/ .kshrc
set -o vi
set filec

__::/etc/defaultrouter
128.135.168.1

```

```
> route add default 128.135.168.1 1
```

```
__::/etc/netmasks  
128.135.0.0      255.255.255.0
```

```
__::/etc/resolv.conf  
domain uchicago.edu  
search uchicago.edu megsinet.net  
nameserver 128.135.20.100  
nameserver 128.135.12.73  
nameserver 128.135.4.2
```

```
__::/etc/nsswitch.conf  
passwd:      files  
group:       files  
hosts:       files dns
```

```
__::/etc/hosts  
127.0.0.1    localhost  
128.135.168.67 stargate.uchicago.edu stargate loghost  
128.135.168.61 seven.uchicago.edu seven
```

```
__::/etc/aliases  
# Following alias is required by the mail protocol, RFC 822  
# Set it to the address of a HUMAN who deals with this system's mail problems.  
Postmaster: root
```

```
# Alias for mailer daemon; returned messages from our MAILER-DAEMON  
# should be routed to our local Postmaster.  
MAILER-DAEMON: postmaster
```

```
# Aliases to handle mail to programs or files, eg news or vacation  
# decode: "|/usr/bin/uudecode"  
nobody: /dev/null
```

```
#####  
# Local aliases below #  
#####
```

```
netmon:      netmon@seamail.uchicago.edu  
unix:        unixadm@seamail.uchicago.edu
```

```
__/etc/hosts.allow  
in.telnetd   : .comm.mot.com localhost  
telnetd      : .comm.mot.com localhost  
klogind      : .comm.mot.com localhost  
kshd         : .comm.mot.com localhost  
ftpd         : 192.168. 145.1.105.  
ch.ftpd      : 192.168. 145.1.105. .megasinet.net .uu.net  
sshd         : ALL  
sendmail     : ALL  
fingerd     : ALL : severity local0.warn  
in.lpd       : 127.0.0. .comm.mot.com  
popper       : localhost .comm.mot.com .uu.net .megasinet.net
```

```
__::/etc/hosts.deny  
ALL         : ALL
```

```

__::/etc/inetd.conf
basically clear out everything
ssh      stream tcp      nowait  root    /usr/local/sbin/sshd  sshd -i

__::/etc/motd
Cray Systems   Version 1.139           Specific September 1998

__::/etc/syslog.conf
(2.6 Championeze)
# Important and security-related systemic messages
*.err;daemon.info;user,local0.none           /var/log/messages
*.err;kern,auth.notice;daemon.notice;user.none /dev/console

# Critical systemic stuff
#      already directed at console and /v/a/messages
#      also write to root and operator
#      write emergencies to everyone
*.alert;kern,daemon.err;user,local0.none     operator
*.alert;kern,daemon.err;user,local0.none     root
*.emerg;user,local0.none                     *

mail.info                                     /var/log/smtplog
lpr.info                                      /var/log/lprlog
user.debug                                    /var/log/userlog

# LOG_LOCAL0 is network monitoring/security logging which is not in auth
local0.debug                                  /var/log/tcplog
local0.crit                                   /dev/console
local1.debug                                  /var/log/local1log
kern.debug                                    /var/log/kernlog
daemon.debug                                  /var/log/daemonlog
auth.debug                                    /var/log/authlog
cron.debug                                    /var/log/cronlog

(Solaris 7 Normal)
#ident  "@(#)syslog.conf      1.4      96/10/11 SMI" /* SunOS 5.0 */
#
# Copyright (c) 1991-1993, by Sun Microsystems, Inc.
#
# syslog configuration file.
#
# This file is processed by m4 so be careful to quote (`') names
# that match m4 reserved words.  Also, within ifdef's, arguments
# containing commas must be quoted.
#
*.err;kern.notice;auth.notice                 /dev/console
*.err;kern.debug;daemon.notice;mail.crit     /var/adm/messages

*.alert;kern.err;daemon.err                  operator
*.alert                                       root

auth.info                                    /var/log/authlog
*.emerg                                       *

# if a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:

```

```

#auth.notice          ifdef(`LOGHOST', /var/log/authlog, @loghost)

mail.debug            ifdef(`LOGHOST', /var/log/syslog, @loghost)

#
# non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(`LOGHOST', ,
user.err              /dev/console
user.err              /var/adm/messages
user.alert            `root, operator'
user.emerg            *
)

___::/etc/services (see melanoma version)

```

```

___::/etc/sendmail.cf (see melanoma version)
(loopback resolution)
Your "cron" job

```

```

/opt/sbin/rotatelog -keep 7 -group staff -mode 640 /var/adm/pacct

```

produced the following output:

```

/var/adm/pacct.[0-9]*: No such file or directory

```

Ronald J. Thielen writes:

```

>
> What are these local configuration errors that seem to come in clusters?
>
> Ron
>

```

The problem was that aismail.uchicago.edu is an alias for alisoan.uchicago.edu. Now the machine thinks it's alisoan so when mail for aismail came to it, it saw that aismail looped back, throwing an error.

I fixed this by putting the following in /etc/mail/sendmail.cf
Cw aismail.uchicago.edu

[3] - SETUP SERVICES

```

=====
[Printer setup]
Instructions for adding/using printers via ethernet for System V.
(This document assumes that the printer has the hostname
syshp4k.uchicago.edu and you want to add it as syshp4k on your
machine)

```

```

{Solaris 2.5.1}
(first time only):
> cd /etc/lp/fd
> for filter in *.fd ; do
_ name=`basename $filter .fd`
_ lpfilter -f $name -F $filter

```

```

_ done
> lpsystem -t bsd -y "Printer in NSIT-AIS" syshp4k.uchicago.edu
> lpadmin -p syshp4k -s syshp4k.uchicago.edu -T PS -I postscript
> lpadmin -d syshp4k (optional -- sets syshp4k as default printer)
> accept syshp4k
> enable syshp4k

```

Now you can use lp to print to it, and lpstat -p to check its status.

```

{Solaris 2.6 & 7}
(first time only):
> cd /etc/lp/fd
> for filter in *.fd ; do
_ name=`basename $filter .fd`
_ lpfilter -f $name -F $filter
_ done
> lpadmin -p nslj1 -v /dev/null
> lpadmin -p nslj1 -i /usr/lib/lp/model/netstandard
> lpadmin -p nslj1 -o dest=syshp4k.uchicago.edu -o protocol=bsd
> lpadmin -p nslj1 -I postscript -T PS
> accept nslj1
> enable nslj1
> (if necessary) start admintool as root and set as default

```

```

[Install HP Jetadmin]
> pkgadd -d jetadmin
> /opt/hpnp/jetadmin &
_ > 1 Configuration
_ NIC Address: 0060B092C3CD
_ Card IP Address: syshp4k.uchicago.edu (128.135.168.66)
_ Network Mask: 255.255.255.0
_ Default Gateway: atm-1155-gw.uchicago.edu (128.135.168.1)
_ > 3 Add printer
_ > syshp4k
_ > modify settings (5 for default)
_ > 0 (To configure)
[Reboot to take effect]

```

```

===Mail
> vi /etc/mail/sendmail.cf
CW stargate.uchicago.edu
("Smart" relay host should be null)
DS

```

```

> vi /root/.forward
s-corrie@uchicago.edu

```

```

> vi ~/.signature
Scot Corrie
UNIX & NT Team Leader

```

```

===/var/logs
/var/log = processes (theory: allow everyone to log to loghosts)
/var/adm = kernel

```

```

> vi /var/spool/cron/crontabs/root
(see crontab.mel)

```

```

#ident "@(#)root      1.14    97/03/31 SMI"    /* SVr4.0 1.1.3.1    */
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * 0,4 /etc/cron.d/logchecker
#10 3 * * 0 /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1

## Custom log rotation
##
20      17      *      *      4      /opt/sbin/rotatelog -keep 5 -group staff -mode
640 /var/log/authlog /var/log/cronlog /var/log/daemonlog
/var/log/kernlog /var/log/local1log /var/log/lprlog /var/log/messages /var/log/
skipd.log /var/log/smtpllog /var/log/tcplog
30      17      *      *      4      /opt/sbin/rotatelog -keep 4 -group staff -mode
644 /var/log/userlog /var/log/ab2/logs/access-8888.log
/var/log/ab2/logs/errors-8888.log
40      17      *      *      4      /opt/sbin/rotatelog -keep 2 -group staff -mode
660 /var/log/dsm.stderr /var/log/dsmsched.log
50      17      *      *      4      /opt/sbin/rotatelog      -keep 4 -group daemon -
mode 664 /var/log/ab2/logs/access-8888.log
/var/log/ab2/logs/errors-8888.log
##
## NFS based file synchronization.
##
#15      0,2,4,6,8,10,12,14,16,18,20,22 * * * /root/bin/sync-cron

[4] - SETUP SOFTWARE
=====
===gzip-1.2.4-sol7-sparc-local
ftp sunfreeware.com
pkgadd -d gzip-1.2.4-sol7-sparc-local

===gcc-2.8.1-sol7-sparc-local.gz
ftp gnu.org
pkgadd -d gcc-2.8.1-sol7-sparc-local.gz
pkgadd -d zlib-1.1.3-sol7-sparc-local.gz
pkgadd -d libpng-1.0.2-sol7-sparc-local.gz

[x86 issues: /usr/bin:/usr/ucb
configure --with-gnu-as --with-gnu-ld --with-stabs
--prefix=/opt/bin (if you want to change from /usr/local/bin)]

===perl5
/usr/local

===ssh-1_2_26_tar.gz
./configure
make
make install

__::/etc/services
ssh      22/tcp      sshd      scp      # Secure Shell Service

```

```

__>::/etc/sshd_config
Port 22
ListenAddress 128.135.168.67
HostKey /etc/ssh_host_key
RandomSeed /etc/ssh_random_seed
ServerKeyBits 768
LoginGraceTime 600
KeyRegenerationInterval 3600
PermitRootLogin yes
IgnoreRhosts yes
StrictModes yes
QuietMode no
X11Forwarding yes
X11DisplayOffset 10
FascistLogging no
PrintMotd no
KeepAlive yes
SyslogFacility AUTH
RhostsAuthentication no
RhostsRSAAuthentication yes
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords no
UseLogin no
# CheckMail no
# PidFile /u/zappa/.ssh/pid
PidFile /etc/sshd.pid
# AllowHosts *.our.com friend.other.com
# DenyHosts lowsecurity.theirs.com *.evil.org evil.org
Umask 022
# SilentDeny yes

[setup to start at boot]
> cp /usr/local/src/ssh-1.2.6/ssh /etc/init.d/ssh
> ln -s /etc/init.d/ssh /etc/rc2.d/S75ssh
> kill -HUP PID of inetd

[setup to start from inetd]
__>::/etc/inetd.conf
ssh      stream  tcp      nowait  root    /usr/local/sbin/sshd  sshd

> kill -HUP PID of inetd

[to add log entries]
__>::/etc/syslog.conf
auth.info          /var/log/authlog

(for problems run > make clean
source.c = source
source.h = headerfiles)

-----REBOOT-----
===ncftp
> /usr/local/

===ftptool

```

```
> /usr/local/
```

```
===windowmaker 0.50.2
```

The following software is required to use Window Maker:

```
- X11R6.x
```

Window Maker can be compiled in older versions of X, like X11R5 (Solaris) or X11R4 (OpenWindows) but it will not work 100% correctly. In such servers there will not be application icons and you'll have trouble using the dock. Upgrading the client libraries (Xlib, Xt etc) will help if you can't upgrade the server.

```
=Optional
```

```
- libXPM 4.7 or newer
```

Older versions may not work!!!
Available at <ftp://sunsite.unc.edu/pub/Linux/libs/X/>
There is builtin support for XPM files, but it will not load images in some uncommon encodings.

```
- libpng 0.96 or newer and zlib
```

For PNG image support.

```
- libtiff 3.4 or newer
```

For TIFF image support.

You can get it at <ftp://ftp.sgi.com/graphics/tiff>

```
- libjpeg 6.0.1 or newer
```

For JPEG image support

```
- libgif 2.2 or libungif
```

For GIF image support

```
- GNU xgettext
```

If you want to use translated messages, you will need GNU gettext.

Other versions of gettext are not compatible and will not work.

Get the GNU version from <ftp://prep.ai.mit.edu>

(Most of these can be get from <ftp.uu.net/pub/graphics>)

```
=INSTALLATION:
```

Unpack WindowMaker-data.tar.gz and move the pixmaps directory in it to /usr/local/share/pixmaps

You may need to do the below:

```
> CPP_PATH="gcc -E -x c" ./configure
```

You also might need to use the --with-gfx-libs and --with-gfx-incs to supply the directory where libtiff is located.

Next:

Every user on your system that wishes to run Window Maker must do the following:

1. Install Window Maker configuration files in your home directory.

```
> wmaker.inst
```

wmaker.inst will install Window Maker configuration files and will setup X to automatically launch Window Maker at startup.

2. > vi /usr/dt/config/Xconfig

(uncomment the following line)

```
DTlogin * xdmMode
```

3. > vi .xsession

```
exec env PATH="/bin:/usr/sbin:/usr/bin:/usr/ccs/bin:/usr/ucb:
```

```
/usr/local/bin:/usr/openwin/bin:/opt/SUNWdoe/bin:/usr/local/netscape:
```

```
/usr/local/cdplayer:/etc:" /usr/local/bin/wmaker
```

```
#exec xterm
```

4. /usr/local/GNUstep/Apps/WPrefs.app/WPrefs

5. Read /home/scot/GNUstep/Library/WindowMaker/README to edit menus

```
===workman 1.3
(Sun Solutions CD)
> /cdrom/cdrom0/products/Hyperion/_install/run.sh
> vi .profile to add path

===Netscape 4.5
> /usr/local/netscape

===traceroute
> /usr/local/

===staroffice
> ./setup
Solaris 2.7 - SPARC
key is: GNJO9CYPB2K87X0RLYYW84 - good

Solaris 2.7x86 - Intel
key is: GNJ8LWZB51K8DN68LYYXY7

===adsm
> cp /home/scot/dsm.sys dsm.opt dsm.inc /opt/IBMadsm-c
ftp ftp.cse....
pkgadd -d sunadsm.pkg
__::$/ .profile

__::/opt/IBMadsm-c/dsm.opt
SErvername      adsm

SUBdir          yes

__::/opt/IBMadsm-c/dsm.sys
SErvername      adsm
  COMMmethod    TCPip
  TCPport       1500
  TCPserveraddress  adsm-server1
  NODENAME      stargate

errorlogr      5

schedlogr      5

# Include file if you want to exclude something
#INcllexcl     /opt/IBMadsm-a/dsm.inc

__::/opt/IBMadsm-c/dsm.inc
* Include/exclude list
* Scanned first-match from bottom up
*
exclude /proc/.../*
exclude /dev/.../*
exclude /devices/.../*
exclude /mnt/.../*
exclude /etc/ssh*
exclude /etc/v5srvtab
exclude /vol/.../*
```

```

exclude /cdrom/.../*
exclude /tmp/.../*
exclude /alt/.../*
exclude /cdr/.../*
exclude .netscape/.../*
*
exclude .nbk/.../*
exclude .ssh/.../*
exclude .pgp/.../*
exclude priv/.../*

___::/opt/IBMadsm-c/dsmc schedule -pass=funkymonkey &

===make (wants gnu style linker)

===xpm

===xmcd
> cp to /usr/local
> gzip and tar
> ./install.sh
> run the config.h file

===gimp & gtk
> install the gimp tool kit first
> cp to /usr/local
> gzip and tar
> ./configure
> make
> make install

===xemacs

===tcpwrappers
ftp ftp.win.tue.no/pub/security

===apache_1.3.3

===screen

===scream

===mutt (wants gnu style linker)

===tcsh

===Wabi 2.2
> pkgadd -d /cdrom/solaris_2_5_1_desktop_1_1_/Wabi/sparc/pkgcs
Login as the user, setup Wabi and install Windows
> /opt/SUNWwabi/bin/wabi & (as scot)
> install Windows 3.11 from disks, or from a CD
> install Word and Excel
> install TCP/IP

===[Rpms]
> mkdir /opt/bin

```

```
> cp rpm2cpio /opt/bin
> cp cpio /opt/bin
> cp make /opt/bin
> see 05rpms.doc for install instructions
```

[5] - OPTIONAL

```
=====
===Setup NIS if it doesn't already exist
```

```
Host name and IP address
Network and netmask
NIS domain
Printers and print servers
Mail service
Remotely mounted NFS file systems
[Setup home to follow]
> cd /etc/master
> vi auto_home
> cd /var/yp
[Setup password]
> ypmatch carries passwd
> pwconv
> make
```

```
====> vi /etc/default/login
# CONSOLE=/dev/console
```

```
=====[::Applications]
```

```
---[::ADSM]
```

```
[backup]
```

```
> dsmc selective /home - (Backs up Home)
> dsmc incremental (Backs up entire system)
```

```
[restore]
```

```
> dsmc
```

```
dsmc> q backup /home/scot -su=yes
```

```
dsmc> enter password
```

```
dsmc> restore /home/scot/* -su=yes
```

```
dsmc> restore -pick /holos/finststate_50/data -su=yes -ina=yes
```

```
dsmc> quit
```

```
[ftp site]
```

```
> ftp ftp.cac.psu.edu
```

```
> cd /adsm/fixes/v3r1/sol26
```

```
> mget *
```

```
---[::Apache]
```

```
___[For Solaris Install]
```

```
(This one was for ns296, a dual install)
```

```
{With Apaci}
```

```
> ./configure --prefix=/export/home/www
```

```
> make
```

```
> make install
```

```
> /export/home/www/sbin/apachectl start
```

```
___[Directory paths]
```

```
> netscape http://localhost
```

```
> home is: /home/http/html/index.html
```

```
> cgis in: /home/http
```

```
> configs: /etc/http/conf
```

```

---[::Clearcase]
> /usr/atria/bin/cleartool setview scot
[checkout with a comment]
> cleartool co -c "fixing scot" auto_home.ns netgroups
[abort checkout]
> cleartool unco
[check files in]
> cleartool ci file1 file2
[update the NIS maps]
> clearmake auto_home
[present directory]
> pwv

---[::Exceed]
___XSession
> has a custom X config
- xdmcp indirect
- host .11
- Selected First Display
___Screen
> Save geometry on exit
> Exit on server reset (No)

---[::Eudora]
[change passwd]
> popauth
[Setup]
(Mail is on Harper, filters through midway or plaisantes)
[PC Settings]
Getting Started
POP: scot@stargate.uchicago.edu
Realname: Scot Corrie
Return address: s-corrie@uchicago.edu
Personal Information
POP: scot@stargate.uchicago.edu
Realname: Scot Corrie
Return address: s-corrie@uchicago.edu
Hosts
POP: scot@stargate.uchicago.edu
SMTP: stargate.uchicago.edu
PH: ns.uchicago.edu
Finger: acs-email.uchicago.edu

---[Networker]
___[Show what is mounted on a tape, or status]
> mt -f /dev/rmt/1cbn
> mt -f /dev/rmt/0 stat
[Start Networker] > nwadmin &
[To check what tapes are loaded]
> nsrjb -v
> lp -d it /usr/sbin/nsrjb
> nsrjb -c -v -j jukel (basic mount)
[Reset the tape drive]
> nsrjb -H
[ ??]
> more ntestd (/usr/scripts)
> nsradmin

```

```

[recover a file/directory] - nwrecover
[umount a tape] > cancel
[To check the current status of a mounted drive]
> mt -f /dev/rmt/2cdn status
> mt -f /dev/rmt/0 status (lx)
[logs are kept in /var/adm; /nsr/adm; /nsr/log/daemon]
[To start or stop Networker]
> nsr_shutdown -a -q
> cd /etc/init.d
> ./networker stop
> ps -ef |grep nsr (nsrd - server; nsrexecd - client)
[::/usr/script/checkit] - Script that checks for runaway networker processes,
and if it finds one, it will kill networker, and restart it.

```

```

[Setup the enabler] > nsrcap -c -v e87c6a-a9142e-0aa1b2
Authorization code: 0efd01cd
[Put a tape in the drive, but not to be used by networker]
In nwadmin go to Devices....
select the device you are the putting the tape in
disable the device.
This will make this device unavailable to networker but available for anything
else.
And just enable it after you are finished with the tape.

```

```

Eject the tape from the other drive using Unmount
Go To Media Jukeboxes and select jukel (only have one juke anyway)
Select Details
Scroll down to Loaded Volumes
Highlight the volume name, change the name to a blank and click on Change (don't
click delete)
Do the same for Load Slots
Click on Apply
Load or inventory a tape in that device

```

```

[Put a tape in the drive by Scot, and access from River1] > rsh fusion dd if=/dev/rmt/0
bs=20b | tar -xvf

```

```

---[PCNFS]
___["PCNFS not responding" ]
> /etc/init.d/pcnfs start (stop)
[Restart PCNFS]
> /etc/init.d/rpc stop
> /etc/init.d/nfs.server stop
> /etc/init.d/nfs.client stop
> /etc/init.d/yp stop
> /etc/rpc start
> /etc/yp start
> /etc/nfs.server start
> /etc/nfs.client start
___[Test PCNFS and Named]
> nslookup bombur
> cat /etc/resolv.conf = (check /etc/resolv.conf)

```

```

---[::ph]
> ph s-corrie return all
> login
> me

```

```

> edit email

---[Sybase Shutdown - Irfserver]
> ssh irfserver
> su - sybase
> pwd
>> /home/sybase
> isql -Usa
>> Password:
>_ 1> sp_who - (See who's logged in)
>_ 2> go
>> spid      status      loginame      hostname      blk      dbname      cmd
-----
      1 recv sleep      sa            WinNT         0        fas        AWAITING COMMAND
      2 sleeping      NULL          NULL          0        master     NETWORK HANDLER
      3 sleeping      NULL          NULL          0        master     DEADLOCK TUNE
      4 sleeping      NULL          NULL          0        master     MIRROR HANDLER
      5 sleeping      NULL          NULL          0        master     HOUSEKEEPER
      6 sleeping      NULL          NULL          0        master     CHECKPOINT SLEEP
      8 recv sleep      sa            irfserver     0        pubs2     AWAITING COMMAND
      9 running        sa            irfserver     0        master     SELECT
     15 recv sleep      kfrey        MOONPIE       0        per       AWAITING COMMAND
     17 recv sleep      sa            voodoo        0        master     AWAITING COMMAND
(10 rows affected, return status = 0)
>_ 1> shutdown SYB_BACKUP - (Shutdown the Backup Server)
>_ 2> go
>> Backup Server: 3.48.1.1: The Backup Server will go down immediately.
>> Terminating sessions.
>_ 1> shutdown
>_ 2> go
>> Server SHUTDOWN by request.
>> The SQL Server is terminating this process.

---[Volume Manager]
___[Inventory the disks for the storage array]
> umount (all volumes that will be affected)
> vxinstall
>_ 2 (Custom)
>_ n (do not encapsulate. Encapsulate will cause the drive/s to be managed by Volume
Manager. One of the benefits of this is that if a daemon monitors the drive, it will
page root if there is a problem with the drive. If a .forward file is in the root
directory,
whoever is listed in this file will be paged.
>_ E]
>_ 4 (Leave them alone)
(Will be looking at c1 now)
>_ use the default name
___[Setup a Raid 5 File System using Volume Manager]
> cd /opt/vxva/bin
> vxva &
>_ click on the rootcfg icon
>_ middle-mouse click on the all drives you want to include
>_ click on Basic-Ops, File System, Create, Raid 5
>_ click on the Maxsize icon
>_ type in the Mount point (/u01 in this example)
>_ click on the Apply icon (Will turn gray as it applies, and blue when it is
completed)

```

>_ Exit

====[:vi]

The seven vi modes of operation are as follows:

- ex Command Mode
- ex Input Mode
- ex Shell Escape Mode
- vi Command Mode
- vi Input Mode
- vi Last line Mode
- vi Shell Escape Mode
- most commands in vi can be prefixed by a number

> (inside vi) :set term=xterm

> Ctrl-r - Redraws the screen

> 5dw - deletes from cursor, five words

> 5db - deletes from cursor, five words back

> D - deletes from cursor to end of line

> C]-d - deletes from cursor to beginning of the line

> esc:5,10d - deletes lines 5-10

> esc:set nu/nonu turns on and off numbers

> r - replaces a single character at current point

> C - replaces all text from the cursor position to the end of the line

> R - turns off insert mode

> cw - change word

> J - joins the line below with the current line

> . - can be used to repeat nearly any command

> esc:w filename - save as a different file

> esc:e filename - open a second file

> vi -r filename

> 10G - goto line 10

> w - forward one word

> b - backward one word

> e - end of current word

> cc - change the entire line

> 5yw - yanks five words

> yy - yanks the current line

> p - paste to the right of the cursor

> P - paste to the left of the cursor

> f - fa searches forward for the letter a, ";" repeats search

> F - Fa searches backward for the letter a, "," repeats search

> / - search for a pattern, "n" repeats search

==> :[address]s /pattern/newpattern/ [g]

==> 5,50 - means lines 5 to 50

==> .,100 - means the current line to line 100

==> 1,\$ or % - means line 1 to the end of the file

==> [g] - means global. Will change all instances.

Ex:

> :1,30s/users/people Search for the first occurrence of a word and replace it

> :1.\$s/^Unix\$/UNIX Search for "Unix" on a line by itself, replace with "UNIX"

> :%s/[0-9][0-9]*\./ Remove all numbers followed by a period from the beginning of any line in the file

> :!command - escape to perform one command

> :!sh - jump to a shell within vi, exit or C]-d

> C]-z - put vi to sleep

> :set all - shows all options available

> :set modename, set nomodename - turn an option on or off

```
> vi .exrc (put in $home)
```

```
---[search and replace returns]
```

```
> g /^V^M/s//((new)/g
```

```
set showmode
```

```
set autoindent
```

```
set wm=5
```

```
set shell=/bin/bash
```

```
or set showmode autoindent wm=5 shell=/bin/bash
```

```
=====[X and Window Managers]
```

```
---[CDE] - Chapter 9 USA
```

The login process on a machine with CDE is handled by the program dtlogin, which is based on xdm.

Prevents a user from having to use xinit or startx.

```
---[restore last session]
```

```
> cd $HOME/.dt/sessions ; cp -r current.old current
```

Solaris also offers a twm session.

The CDE desktop is controlled by the CDE window manager, dtwm.

Each of the buttons, icons, and pop-up menus are a distinct type of component.

The CDE has five different types of components:

- panel, box, control, sub-panel, switch

To solve an occasional xterm and dtterm bash or tcsh problem, you can do this:

```
> cd /usr/local/bin; ln -s bash \-bash;
```

```
---[dtterm] - Far more advanced than the traditional xterm and allows for menu-based customization.
```

Can be configured in .Xdefaults

```
---[CDE binaries] - located in /usr/dt/bin
```

```
---[Create and install an action]
```

```
> dtcreate &
```

Usually actions created by a user are stored in one of the following places:

\$HOME

\$HOME/.dt/appmanager

Once an action is created, install it in a submenu.

1. Use the arrow above the dtterm icon to display the Personal Application sub-panel.
2. Using the File Manager, dtfile, go to the directory containing your actions.
3. Click the icon for the action and drag it onto the Install Icon item in the Personal Applications subpanel.

```
---[CDE files]
```

= .dtprofile - controls the behavior of shells that are started while running under CDE.

DTSOURCEPROFILE=true (to ensure .profile is sourced)

dt_sessionlogfile=\$HOME/.dt/session.whatever.log

The local system version is stored in the file /etc/dt/config/sys.dtprofile

The distribution version is usually stored in /usr/dt/config/sys.dtprofile

= sessionetc - used to start CDE. Standard sh script similar to the .Xinitrc or

.Xclients file used in X Windows. Some common uses of this file are to start up programs such as xload or xlock and to set the background.

```
= sessionexit -  
= menus  
/users/scotc/.dt/C
```

```
---[web sites]  
www.plig.org/xwinman  
www.kde.org  
www.symark.com  
www.percolator.com
```

```
---[Modifying the CDE programs menu]  
> Use the Create Action Tool to create an action.  
> vi /home/scot/.dt/.dtwmrc  
(If this file does not already exist, copy a master from  
/usr/dt/config/C/sys.dtwmrc (This will end up with two files. a *.dt file and a  
file that ends up in $/.dt/appmanager Check the users guide and the Admin Guide  
for more information).
```

```
---[Add an icon]  
How do I add an Icon to the Solaris Front Panel?
```

SUMMARY:
The following are instructions on how to add an icon to a Solaris 7 front panel (style

manager).

#####

```
{  
DEFINITIONS:
```

- CONTROL - The name of the icons on the panel
- ACTION - what you want the control icon to do
- DATA_TYPE - file specific attributes needed; see the website for more details.
- TYPE - ICON for controls, and COMMAND for actions: Basically what the icon is for.
- ICON - name of icon (actual picture)
- LABEL - Comments that appear when icon is selected
- DISPLAY_CONTROL_LABELS - Show the labels on the front panel (yes=true no=false)
- PUSH_ACTION - What you want the icon to do once you press on it
- DROP_ACTION - What you want the icon to do once you drop an icon on it
- HELP_STRING - Comments if you select HELP over the icon, this is displayed
- CONTAINER_TYPE - Shape of the icon
- CONTAINER_NAME - Where you want the icon
- POSITION_HINTS - The position of the icon on the panel
- EXEC_STRING - Actual command "action" to be performed once the icon is selected
- WINDOW_TYPE - Describes the behavior of the window (PERM_TERMINAL to leave a window up after command executes: TERMINAL - closes the window after command executes: NO_STDIO - does not show a terminal window UNLESS it is needed for user input.)

".fp" - Front Panel extensions
".dt" - Display Terminal extensions

```
}  
#####  
#####  
}  
FILE LOCATIONS:
```

The following are the most useful default files for the front panel:

```
/usr/dt/appconfig/types/C [contains the CONTROL,ACTION, &  
                           DATATYPE files]  
  
/usr/dt/appconfig/app-defaults/C [contains dummy files (not very  
                                  important)]  
  
/usr/dt/appconfig/backdrops/ [contains background images]  
  
/usr/dt/appconfig/bin/ [binaries for default icons]  
  
/usr/dt/appconfig/config/ [contains configuration info for  
                           default desktop]
```

** NONE OF THE ABOVE FILES SHOULD BE EDITED!!!!!!

When you need to make changes, or add an icon, you should create the following directory, and any subsequent subdirectories that you may need:

```
/etc/dt/appconfig/types/C  
/etc/dt/appconfig/bin/  
...  
...
```

(I for example only have /etc/dt/appconfig/types/C)

Why make a new directory???

The order of the directory search is:

1. \$HOME/.dt/config (does not exist by default)
2. /etc/dt/config/ (does not exist by default)
3. /usr/dt/config

default environment variables are stored in /usr/dt/config
}

```
#####  
#####  
{  
EXAMPLES:
```

In this example, we will add an icon that when clicked will bring up window, and initiate telnet to server "ctsrv01":

Now let's add an icon:

1. go to /etc/dt/appconfig/types/C
2. vi telnet.fp (this will be the CONTROL file)
3. Add the following:

```
CONTROL telnet
{
  TYPE                icon
  CONTAINER_TYPE      BOX
  CONTAINER_NAME      Top
  POSITION_HINTS       last
  ICON                Dtbshll
  DISPLAY_CONTROL_LABELS True
  PUSH_ACTION         LaunchTelnet
  LABEL               telnet
  HELP_STRING         "This file launches telnet"
}
```

(* now save the file)

4. vi telnet.dt and add the following:

```
ACTION LaunchTelnet
{
  LABEL              Telnet
  TYPE               COMMAND
  WINDOW_TYPE        TERMINAL
  ARG_COUNT          0
  EXEC_STRING        sh -c 'exec /usr/dt/bin/dtterm -e /usr/bin/telnet \
                        ctsrv01'
  ICON               Dtactn
}
```

5. After you have saved these files, restart the window manager and Whalla!

```
}
#####
#####
```

REFERENCE:

For more detailed information, or if I left something out(?) take a look at the following website for help:

1. Solaris Common Desktop Environment: Advanced User's and System Administrator's Guide

[http://docs.sun.com/ab2/coll.8.40/CDEADMIN/@Ab2TocView?_AB2_SearchAllColl=CL&Ab2Lang=&Ab2En]

c=&scope=BOOK&DwebQuery=type]

I recommend looking at the ACTIONS MANUALLY section:

#####

Enjoy:

/.../ben

Remember to look out for:

 Yet another

 Bennovation...

PS: Thanks to James Mello (kingjamm@colltech.com) for helping me figure out the last leg of this adventure!

Scanner

<http://www.mir.com/mtek/>

Deneba Canvas 7 for Linux.

License: L181+VAEGQ6012527-1840

http://www.deneba.com/download/cv7_linux_beta/linux_dnld.html

=== [UNIX Temp Notes] =====
mount -t vfat -o uid=15457 /dev/hda7 /data2

[fstab for mounting winME for scot]

/dev/hda1 /winME vfat noauto,rw,uid=500,gid=101,mode=777 0 0

<http://www.linuxdoc.org/HOWTO/CD-Writing-HOWTO.html>