Select Page Select Page

Unix Notes

To see what's recently changed on this page - please look at the Changes page.

I've put together these notes over the years in order to help me with Unix systems support. And yes, a small bit of searching will get the same information. Probably more information in a better format, now that I think of it.

Anyway, I hope this helps.

Disclaimer:
Use at your own risk. If you see any problems with this stuff, please send an email explaining what I screwed up on: dburton3@tampabay.rr.com.

Links:
I have some links scattered throughout this page that will either take you to other Unix related sites or to other support pages I've put together (Server Build document, GSP Notes, Ignite Notes, /etc/inetd.conf and /var/adm/inetd.sec document). I've also duplicated them (and maybe added some) on my Links page.

Rosetta Stone for UNIX

Exceed Stuff

ReflectionX Stuff

Unix (HP) Stuff

Unix (Sun) Stuff

Scripts

Links to other Unix sites

Rosetta Stone for UNIX

- Can't say enough about this site. Great cross reference of Unix commands.

  http://bhami.com/rosetta.html

  Other great Unix pages can be found on my *Links* page.

Exceed Stuff - like ReflectionX

- Add this "/usr/bin/X11/xterm -display @d &", to get an xterm window.
- Do the following to get the entire CDE display:

```
open "xconfig -> communication"
  mode = XDMCP-query
  Press "configure" button, type in target hostname
  display class = HCLpcXserver
```

ReflectionX Stuff - like Exceed

- Start ReflectionX in Windows by selecting the "Start -> Programs -> Reflection -> ReflectionX" menu pick.
- Select the "Connection" pull down menu and select "New XDMCP Connection".
- In the "Connection type" section, add the following to the "Description:" area... "XDMCP Direct to IP/Name". The "IP/Name" is the IP address or name of the server you want to connect too.
- Select "Direct" in the "Method:" menu (at right).
- Add the target server name or its IP into the "Host name:" area.
- The "Backspace" key will not match the "stty" setting on HP. The ReflectionX "Backspace" key will be set to "^?", not the "^H" it needs to be. So... to fix this little problem, do the following:
    - Select the "Keyboard Settings" button at the top-middle area of the window.
    - Deselect the "Backspace key sends delete" button located in the "Options" area.
    - Select "OK".
- Click on "Connect" to get the CDE login screen.

Unix (HP) Stuff

```
HP Support:     1-800-633-3600
HP Contracts:   1-800-386-1115
HP Codewords:   1-800-538-1733 - 6am to 4:30pm PST
HP Passwords:   1-800-326-0411 - 6am to 6pm MST
HP Customer Business Center: 1-800-386-1117
```

You can find additional HP links on my Links page.

How to build a server

- **My version of an HP-UX 11.xx server**

  Server Build document

- **Building an HP-UX 11.00 bastion host**

  To build a pretty secure system...

  See Kevin Steves "Building a Bastion Host Using HP-UX 11.00" - HP document: USECKBAN00000800

CDE stuff

Change the login screen logo

- Obtain the application "XV" from one of these sites:

  http://hpux.cs.utah.edu/, or
  http://hpux.connect.org.uk/, or
  http://hpux.ee.ualberta.ca/

- Use XV to convert your existing bitmap file to XPM format.
  - Valid values for this resource can be any file, which can be viewed in the CDE icon editor (dticon). If the file is in either XMICONSEARCHPATH or XMBMICONSEARCHPATH, then only the filename is necessary. If the file is not in one of those paths, then the full path should be specified.
- Save the file to: /etc/dt/appconfig/icons/C/LoginLogo.xpm
- Change the perms:

  ```
  chmod  444  /etc/dt/appconfig/icons/C/LoginLogo.xpm
  ```

- Add the following to the /etc/dt/config/C/Xresources file:

  ```
  Dtlogin*logo*bitmapFile: /etc/dt/appconfig/icons/C/LoginLogo.xpm
  Dtlogin*logo*height: 139   (68  for Exceed)
  Dtlogin*logo*width:  552   (276 for Exceed)
  ```

Change the login screen "Welcome" message

- It can be set system-wide by adding these lines in the /etc/dt/config/C/Xresources file:

  ```
  Dtlogin*greeting.labelString: Welcome to %LocalHost%\n
  Insert additional message here\n
  Final additional text

  Note:
  The \n's indicate places where new lines should appear
  in the greeting.

  Inserting %DisplayName% will display the X server
  display name if different from the login host.
  ```

- Once the user name has been entered, the login server displays the message "Welcome username" by default. This greeting string can be customized by changing the line:

  ```
  Dtlogin*greeting.persLabelString:   Welcome %s

  Note:
  The "%s" will be replaced with the username by
  the system. So if you want to change the login
  to read "login_name is a goof", you would make
  the change look like "%s is a goof"; instead
  of "Welcome %s".
  ```

Change the login greeting (Copyright Blue Screen) to your company disclaimer

- Add this next line to the /etc/dt/config/Xconfig file:

  ```
  Dtlogin*environment: INFO_PATH=/etc/issue
  ```

- Run this command to reload the configuration information:

  ```
  /usr/dt/bin/dtconfig -reset

  Note:
  Sometimes (don't know why) adding this stinkin file
  will make the "new" login logo NOT show.
  ```

```
If this is your problem do the following...
```

- Change the default "dthello" execution:

  ```
  vi /etc/dt/config/Xsession.d/0091.dthello
  ```

- Add the following lines to this new script:

  ```
  #!/usr/bin/ksh
  dtstart_hello[0]="/usr/dt/bin/dthello -file /etc/issue &"
  ```

- Set the permissions:

  ```
  chmod 555 /etc/dt/config/Xsession.d/0091.dthello
  ```

Change the screen saver (locked screen) logo

- It can be set system-wide by adding this next line to the /etc/dt/config/C/sys.resources file:

  ```
  Dtsession*lockImageName: /etc/dt/appconfig/icons/C/LoginLogo.xpm
  ```

- Per-user changes can be made in $HOME/.Xdefaults. An example would be:

  ```
  Dtsession*lockImageName: ~/bitmaps/YourLoginLogo.xpm
  ```

- Also, you can change the width and height with the following additional lines:

  ```
  Dtsession*lockImageName*height: 129
  Dtsession*lockImageName*width:  282
  ```

Disable X-Client broadcast access

- Comment out the following lines in the /usr/dt/config/Xaccess file.

  ```
  *    # grant service to all remote displays
  *    CHOOSER BROADCAST    #any indirect host can get a chooser
  ```

Print Window

- Print out a selected window (xxx = You printer name).

  ```
  /usr/bin/X11/xwd | /usr/contrib/bin/X11/xpr | lp -dxxx
  ```

Check CDE configuration

- Run this to check CDE:

  ```
  /usr/contrib/bin/X11/dr_dt
  ```

Other CDE stuff

- Map the "del" key on the fly: xmodmap -e "keycode 109=hpDeleteChar"
- Change the mouse buttons on the fly: xmodmap -e "pointer = 123"
- Get motif version info: what /usr/lib/Motif1.2/libXm.sl
- Enable CDE: /usr/dt/bin/dtconfig -e
- Disable CDE: /usr/dt/bin/dtconfig -d
- Send sound to another server/workstation:
  - On target box as root...

    ```
    /usr/ausio/asecure -C
    Enter the root password
    /usr/audio/bin/asecure +h source.com
    ```

  - On source.com box...

    ```
    /usr/audio/bin/send_sound \
    /path/of/sound/file.au -server target.com
    ```

- Run this to see your fonts:

  ```
  The following command will provide you with a full list of fonts available on your system:

  /usr/bin/X11/xlsfonts
  ```

  Or view the appearance of each font: /usr/contrib/bin/X11/xfd -fn

---

Software Configurations

Ignite

You may want to take a peak at my Ignite document too.

- Boot to an ignite server after stopping the boot process:

  ```
  isl> hpux (;0)/boot/install
  ```

- To make a system image for ignite and place it into /tmp:

  ```
  make_sys_image -d /tmp -v -s local
  ```

- To make a recovery tape:

```
### Older version of Ignite
make_recovery -vAC -d /dev/rmt/0mn

### Newer version of Ignite
make_tape_recovery -v -I -x inc_entire=vg00 -a /dev/rmt/0mn

Note:
The tape recovery tool uses a no-rewind device file. It
also will not preserve the mirrored disk configurations.
You will need to recreate the mirror(s) and restore everything
else from a backup.
```

Sendmail stuff

- In order to send mail to uppercase users ("USER" for example). Add the "u" below:

```
From: Mlocal,P=/usr/bin/rmail, F=lsDFMAw5:/|@qm9,  S=10/30, R=20/40,
To:   Mlocal,P=/usr/bin/rmail, F=lsDFMAw5u:/|@qm9, S=10/30, R=20/40,
```

- How to make the root id change so when looked at from Outlook, pager, etc. instead of:

  root@batcave.gothamcity.com (userID@hostname.gothamcity.com), it now looks like root.batcave@gothamcity.com (userID.hostname@gothamcity.com)

  1. Uncomment the "O UserDatabaseSpec=/etc/mail/userdb.db" line in the /etc/mail/sendmail.cf file.
  2. vi /etc/mail/userdb and add: root:mailname root.batcave@gothamcity.com.
  3. Run this: makemap btree /etc/mail/userdb.db < /etc/mail/userdb.
  4. Restart sendmail: /sbin/init.d/sendmail stop;/sbin/init.d/sendmail start.
  5. Test by sending mail as root to somebody (like you).

- Check what version of sendmail your using:

```
echo \$Z | /usr/sbin/sendmail -bt -d0
```

  or this

```
sendmail -d0.1 < /dev/null | grep -i version
```

  or this

```
echo '$=w' | sendmail -bt -d0.4 | grep Version
```

- Send email attachments:

  To send a plain text doc:

```
mailx -m -s "Test attachments" root 1>/dev/null << EOF
$(cat some_text_file | ux2dos - | uuencode stuff.txt)
EOF
```

  To send a binary file:

```
mailx -m -s "Test attachments" root 1>/dev/null << EOF
$(cat some_binary_file | uuencode binary_file)
EOF
```

  or

```
uuencode [filename] [filename] | mailx -s "Subject" batman@batcave.com
```

- Spoof (sort of) your email address - this will show "name@bogus.com" sent the email.

```
mailx -r "name@bogus.com" -s "subject" you@somewhere.com
```

- Build a new sendmail.cf file? Run this: /usr/newconfig/etc/mail/cf/cf/gen_cf

- Using mailx and the "at" command, send an email at a given time so you can get out of meetings. Compose an email message at 2.00pm, but want to send it to your pager at 5pm. First, make sure your user ID is in the file /var/adm/cron/at.allow, so that you are allowed to use "at". Then do something like:

```
at 1700
cat mail.txt | mailx -s "test mail" Dougs@pager.com ^D
(The ^D is pressing Control-D to end the script).
```

- Additional sendmail info found in my Server Build document.

FTP Configuration

- 11.xx FTP configuration info found in my Server Build document.
- Find out what version your running: what `which ftp`

NTP stuff

- Learn about NTP (time): http://www.ntp.org
- To check ntp: ntpq -p
- To do other checking: ntpdate -d
- Additional NTP configuration info found in my Server Build document.

NIS stuff

- Use an NIS netgroup list to do stuff:

```
for i in `netgrouplist net_group_list ; do something ; done
```

- Use the /etc/hosts.equiv:

```
export -i -o rw=@net_group_list /users
```

- Add to /etc/passwd to let ONLY the admins in that are on this list:

```
+@admin_access::-24:-24:::
```

Networkish stuff

Check LAN cards

- Use "lanscan" to get "NUM ID" (for 10.20) or "Crd In#" (for 11.x) to use on stuff below.

```
lanadmin -x  0 = lan info (speed, half or full duplex, auto on or off)

lanadmin -X 100HD   0 = turn lan0 to 100 speed, half duplex, auto on
lanadmin -X 100FD   0 = turn lan0 to 100 speed, full duplex, auto off
lanadmin -X AUTO_ON 0 = turn Auto Neg. on
```

- Check for the LAN driver being used. Find it's /etc/rc.config.d config file. The "btlan" driver config looks like this:

```
HP_BASET_INTERFACE_NAME[0]=lan0 - the lan number
HP_BASET_SPEED[0]=100FD         - the speed you want
```

  - Here's some more:

```
driver   /etc/rc.config.d/file
------   ----------------------
btlan0   hpeisabtconf
btlan    hpbtlanconf
btlan1   hpbasetconf
btlan3   hpbase100conf
btlan4   hpgsc100conf
btlan5   hppci100conf
btlan6   hpsppci100conf
gelan    hpgelanconf
```

- Stop then start the cards: /sbin/init.d/net stop ; /sbin/init.d/net start
- Check loopback: linkloop -i 1 mac_address
- Verbose output: lanscan -v
- Status of network interfaces: ifconfig lan0 (or lan1, or lan2)
- IP interfaces: netstat -in
- Check the state of the FDDI card: /usr/sbin/fddistat /dev/lan0
- Check the FDDI network neighbors: /usr/sbin/fddinet /dev/lan0
- Check the ATM card: elstat -n ??? -v

```
NOTE:
"???" is the ATM card name gathered from the
"netstat -i" command.
```

Host lookup issues

- Trouble with host lookup using DNS and NIS? Try adding this to the /etc/nsswitch.conf file:

```
hosts: dns [NOTFOUND=continue UNAVAIL=continue TRYAGAIN=continue] nis [NOTFOUND=return UNAVAIL=continue] files[NOTFOUND=return UNAVAIL=con
```

IP to MAC address resolution (or who pings back on the subnet)

- Run the following commands:

```
ping XXX.XXX.XXX.255    example:(10.10.10.255)
arp -a
```

- You will get a return that looks something like the following:

```
"? (10.10.10.XX) at bi:g_:ma:c_:ad:dr:es:s_  ether"
```

- If you see an IP instead of a name, that means DNS (or the host file) can't resolve IP to name. You may have a spook on the line.

LVM/Disk Stuff

FSCK the beast.

```
fsck -F vxfs -o full -y /dev/vg00/lvol6
```

Make a bootable mirror disk (primary=c0t6d0, alternate=c0t5d0)

- The "-B" option is used to create a bootable Physical Volume:

```
pvcreate -B /dev/rdsk/c0t5d0
```

- Make sure to use the character device file when using mkboot:

```
mkboot /dev/rdsk/c0t5d0
```

- Add diagnostics to lif:

```
mkboot -b /usr/sbin/diag/lif/updatediaglif -p ISL -p AUTO -p HPUX  -p LABEL /dev/rdsk/c0t5d0
```

- Add boot info:

```
mkboot -a "hpux -lq (;0)/stand/vmunix" /dev/rdsk/c0t5d0

Note:
The "-lq" indicates no quorum when two disks
are used. When three or more are used you don't need "-lq."

You also need to do this with the primary disk. Otherwise,
when you boot normally to pri and the alt disk is not there
for whatever reason, you will **NOT** be able to boot. The
pri disk will "look" for the alt disk. No quorum????
To bad so sad, you loose. Ack! So do this as well:

mkboot -a "hpux -lq (;0)/stand/vmunix" /dev/rdsk/c0t6d0
```

- Adds the Physical Volume to the root volume group:

```
vgextend /dev/vg00 /dev/dsk/c0t5d0
```

- Do this next command for each "lvol"[1-7].....

```
lvextend -m 1 /dev/vg00/lvol1 /dev/dsk/c0t5d0

******* or *******

for x in lvol1 lvol2 lvol3 lvol4 lvol5 lvol6 lvol7 swap2 Crash Logs
  do
    echo "\n******** Doing ${x} ********\n"
    lvextend -m 1 /dev/vg00/${x} /dev/dsk/c0t5d0
done
```

- Don't forget any extra swap you added (like /dev/vg00/swap2). Do "swapinfo" and "bdf" to check for swap and any other lvols that may have been created.
- Specifies the root logical volume:

```
lvlnboot -r /dev/vg00/lvol3
```

- Specifies the swap logical volume:

```
lvlnboot -s /dev/vg00/lvol2
```

- Specifies the dump logical volume:

```
lvlnboot -d /dev/vg00/lvol2
```

- Specifies the boot logical volume:

```
lvlnboot -b /dev/vg00/lvol1
```

- Recovers any BDRA info:

```
lvlnboot -R
```

- Verify boot, root, swap and dump settings:

```
lvlnboot -v
```

- Display the Primary and Alternate boot paths that are currently set:

```
setboot
(You can also see what drive you booted off of: echo boot_string/S|adb /stand/vmunix /dev/mem)
```

- Change the Alternate Boot Path to the path of the Root Mirror:

```
setboot -a 10/0.5.0

Note:
Use "ioscan -funC disk" to find out what the
path should be. Run "setboot" again to confirm
the change.
```

- Do "shutdown -r 0" (or "reboot") to test. After halting the boot process, type in "bo alt" to boot from the alternate disk. Also, you may want to *remove* the primary disk to test as well. The server should boot from the alternate disk if the primary is not found and no quorum was set.

Add a mirror

- Adds alternate links then mirrors the logical volume:

```
vgextend /dev/vg00 /dev/dsk/c?d?t?
lvextend -m 1 /dev/vg00/lvol1 /dev/dsk/c?d?t?
```

Remove a mirror and/or the mirror drive

- Remove the mirror from a logical volume:

```
lvreduce -m 0 /dev/vg00/lvol1
```

- Remove the disk from the volume group:

```
vgreduce /dev/vg00 /dev/dsk/c?t?d?
```

To extend or reduce a logical volume
(you may need to be in single user mode for /var, /tmp, etc.)

- Increase the target logical volume:
  - With OnlineJFS:

    ```
    fsadm -F hfs (or vxfs) -b new_size /mounted_filesystem
    ```

  - Without OnlineJFS:

    ```
    Unmount filesystem
    lvextend -L 400 /dev/vg00/lvol3
    extendfs /dev/vg00/rlvol3
    Mount filesystem
    ```

- Reduce the target logical volume. Works great if you have no more disk space but have extra in /tmp. Reduce /tmp and put your new found disk space into (lvextend) /var.

  ```
  NOTE:
  You may loose data on the selected Logical Volume.
  ```

  - With OnlineJFS:

    ```
    fsadm -F hfs (or vxfs) -b new_size /mounted_filesystem
    ```

  - Without OnlineJFS:

    ```
    Unmount filesystem
    lvreduce -L 100 /dev/vg00/lvol5
    newfs -F hfs /dev/vg00/rlvol5
    Mount filesystem
    ```

Export/import a volume group

Let's say you need to reinstall the O/S on a server and retain the logical volume(s) and there volume group(s) that hold your data, or you want to move a set of disk that contain data and configured volume groups to a different server.

- Change directories to a safe place on your original server:

  ```
  cd /tmp
  ```

- Tar the volume group(s) "group" file.

  ```
  tar cvf group.tar /dev/vg01/group
  ```

- Make the volume group(s) inactive.

  ```
  vgchange -a n vg01
  ```

- Export the beast and make files needed to rebuild the volume group on the "new" server. When you run this command it will wipe out the /dev/vg01 directory. (Remember, your still in the /tmp directory.)

  ```
  vgexport -v -m vg01.map -f vg01.disk vg01
  ```

- You then need to move the *group.tar* and the *vg01.** files to your "new" server or some other server while you rebuild this one. If you don't create the files needed for the vgimport command (see below), then the command, when run, will simply (re)name every logical volume you have starting with lvol1 through lvol### (the number depends on how many lvols you have). Also, if for some reason you don't save the group file, you'll need to recreate one. Use this next command to make a group file.

  ```
  mknod /dev/vg01/group c 64 0x010000
  ```

- At this point your disks have been moved to your target server or the server was rebuilt. In either case you need the *group.tar* and the *vg01.**files on this "new" server. Untar the *group.tar* file then run the import command.

  ```
  vgimport -v -m vg01.map -f vg01.disk vg01
  ```

- Now make the volume group(s) active.

  ```
  vgchange -a y vg01
  ```

/etc/lvmtab file problems

- Down ALL applications first, then:
  - Move out the old (bad?) lvmtab:

    ```
    mv /etc/lvmtab /etc/lvmtab.old
    ```

  - Check the lvmtab file:

    ```
    vgscan -v -p
    ```

  - Generate the lvmtab file:

    ```
    vgscan -a
    ```

  - Activate all volume groups:

```
vgchange -a
```

- Correct boot information on disk:

```
lvlnboot -R
```

Volume Group problems

- Restore LVM volume group configuration from backup file:

```
vgcfgrestore -n /dev/vg00 -o /dev/rdsk/c?t?d?
```

- Activates volume group:

```
vgchange -a -y  /dev/vg00 /dev/rdsk/c?t?d?
```

- Synchronize stale logical volume mirrors in LVM volume groups:

```
vgsync /dev/vg00 /dev/rdsk/c?t?d?
```

- Synchronize stale mirrors in LVM logical volumes:

```
lvsync /dev/vg00/stuff
```

Convert a "nolargefiles" file system to a "largefiles" file system

- Display statistics

```
fsadm -F hfs (or vxfs) /dev/vg00/rlvol6
```

- To change it without having OnlineJFS (you may need to do this in "single user" mode):

```
Umount the filesystem

fsadm -F hfs (or vxfs) -o largefiles /dev/vg00/rlvol6
```

- To change it with OnlineJFS:

```
fsadm -F hfs (or vxfs) -o largefiles /mounted_filesystem
```

Other OnlineJFS stuff

- Make a snapshot in **seconds** using OnlineJFS:

```
lvcreate -L 100 -n lvsnap vg00    <-- 10-20% of original (snapped) filesystem

mkdir /test

mount -F vxfs -o snapof=/home /dev/vg00/lvsnap /test

# bdf /test
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvsnap   1024000   56968  959480    6% /test

# bdf /home
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvol4    1024000   56968  959584    6% /home

umount /test when done (backed up)
```

  And maybe do this if needed:

```
lvremove -f /dev/vg00/lvsnap
```

- Defrag with OnlineJFS:

```
Check a dir to see if we should defrag it:   fsadm -F vxfs -D -E /home

To defrag it:  fsadm -F vxfs -d -e /home
```

Recreate device files

- Use the "-e" option to re-install device files:

```
insf -e
```

Test the drives (read the drive and output to nothing)

```
dd if=/dev/rdsk/c?t?d? of=/dev/null bs=4096k
```

Wipe the drives

- Run this to wipe your drives (write "nothing" to the drive):

```
dd if=/dev/null of=/dev/rdsk/c?t?d? bs=1024k
```

- or better yet, write "0" to the drive:

```
dd if=/dev/zero of=/dev/dsk/c?t?d? bs=1024k
```

  - If you don't have /dev/zero, you can make it (11.11 (11i) already has /dev/zero):

```
mknod /dev/zero c 3 0x000003 #...on 10.20
mknod /dev/zero c 3 0x000004 #...on 11.x
chown bin:bin /dev/zero
chmod 666 /dev/zero
```

- or....

```
mediainit -v /dev/rdsk/c?t?d?
```

PFS mount

- Add "/dev/dsk/c?t?d? /SD_CDROM pfs-rrip xlat=unix 0 0" to the /etc/pfs_fstab file.

```
c?t?d? = The CD device
SD_CDROM = Mount point
```

- Next, run "nohup /usr/sbin/pfs_mountd"
- Then "nohup /usr/sbin/pfsd""

```
Note:
THE ORDER IS VERY IMPORTANT - pfs_mountd -
MUST BE STARTED FIRST!
```

- Check to be sure pfs started all the correct processes, use the following command. You should see the output below if everything is correct:

```
# ps -ef | grep pfs

root 17456 17455 0 15:15:24 ttyp1 0:00 pfsd.rpc
root 17453 17452 0 15:15:18 ttyp1 0:00 pfs_mountd.rpc
root 17455 16772 0 15:15:23 ttyp1 0:00 pfsd
root 17452 16772 0 15:15:18 ttyp1 0:00 pfs_mountd
root 17475 16772 2 15:15:32 ttyp1 0:00 grep pfs
```

- Now mount the cd using the following syntax, were cXtXd0 is the CD-ROM device file on your machine.
  - If the /etc/pfs_fstab was made:

    ```
    pfs_mount /SD-CDROM
    ```

  - To mount the CD in Rock Ridge format

    ```
    pfs_mount -o xlat=unix /dev/dsk/c?t?d0 /SD_CDROM
    ```

  - If /etc/pfs_fstab was not made:

    ```
    pfs_mount -o xlat=unix /dev/rdsk/c?t?d0 /SD_CDROM
    ```

- Only use pfs_umount to dismount the CD-ROM:

```
pfs_umount /SD_CDROM

Note:
A reboot will be required if at anytime the umount
command is issued on the pfs mounted CD-ROM instead
of using the pfs_umount command.
```

Export a PFS mounted CD

- On your local server with the CD drive:
  1. nohup /usr/sbin/pfs_mountd &
  2. nohup /usr/sbin/pfsd &
  3. Add the filesystem to /etc/pfs_fstab
  4. Add the filesystem to /etc/pfs_exports
  5. pfs_exportfs -a
  6. pfs_mount /cdrom
- On your target (remote) server:
  1. nohup /usr/sbin/pfs_mountd &
  2. nohup /usr/sbin/pfsd &
  3. pfs_mount -o xlat=unix local_system:/cdrom /local_mnt_pnt

Export a mounted CD

- On your local server with the CD drive:

```
mount /dev/dsk/c?t?d? /cdrom
exportfs -i -o ro /cdrom
```

- On your target (remote) server:

```
mount -r hostname_of_server_above:/cdrom /cdrom
```

Software Depot and Swinstall stuff

General Stuff

- To install a patch:
  - Un-shar the file:

    ```
    sh /tmp/Patch_Name
    ```

  - Now install it:

    ```
    swinstall -x autoreboot=false match_target=true -s /tmp/patch.depot
    ```

- To add depots to an 11.00 server using the GUI:

```
swcopy
```

- Without the GUI (from the command line):

```
swcopy -v -s /tmp/patch_dir/PH*.depot @ /tmp/patch_dir
```

- To swcopy a depot of 10.20 patches onto an 11.00 server using the GUI.

```
swcopy -x layout_version=0.8 -i
```

- To register the software for network installs:

```
swreg -l depot -v /PATH/of/depot
```

- Check filesets installed:

```
swlist -l fileset -a state
```

or to find anything NOT configured type:

```
swlist -l fileset -a state | grep -v -E "^#|configured"
```

- To complete the installation of the software:

```
swinstall \*
```

- To verify all products and dependency files:

```
swverify \*
```

- To cleanup (remove) old patches:

```
cleanup       (for 10.20)
cleanup -c 1  (for 11.x)
```

Create a new software depot for patches

- Creating a separate software depot for your patches (on another server?) is useful, when space is tight and patches have dependencies.
  - Assume you have two patches - PHNE_20000 and PHNE_20001.
  - Unshare the patches by typing the following commands.

```
sh PHNE_20000
sh PHNE_20001
```

  - Next, we create a new software depot in the directory /newdir/depot

```
mkdir /newdir/depot
swpackage -s PHNE_20000.depot -d /newdir/depot
```

  - Register your new depot.

```
swreg -l depot  /newdir/depot
```

  - Copy the next patch to the newly created software depot.

```
swcopy -s `pwd`/PHNE_20001.depot PHNE_20001 @  /newdir/depot
```

  - Install your patches.

```
swinstall -s /newdir/depot
```

Remove a software depot

- To find out what depots you have on your server:

```
cat /var/adm/sw/host_object | grep path | awk '{print $2}'
```

- To unregister the depot, type:

```
swreg -u -l depot /path/to/depot
```

Rebuild the software INDEX file

- The software INDEX file is bad or missing. To rebuild it:
  1. mv /var/adm/sw/products/INDEX /var/adm/sw/products/INDEX.bad
  2. vi /tmp/void.psf
  3. Add the next four lines to the /tmp/void.psf file:

```
product void
tag void
fileset void
tag void
```

  4. swpackage -s /tmp/void.psf
  5. swinstall void
  6. swremove void
  7. rm /tmp/void.psf
  8. swremove -d void

Kernel, Core and Swap stuff

- Make a new kernel in SAM but do not "put into place". Then do:

```
mv /stand/system /stand/system.prev
mv /stand/build/system.SAM /stand/system
/usr/sbin/mk_kernel
reboot

NOTE:
Ok... so I don't really know why you would want to
do this. I've always let SAM put the kernel into
place and rebooted.
```

Other Kernel stuff

- Make a kernel

  Create the files needed to build a new kernel using the current kernel parameters:

```
cd /stand/build
/usr/lbin/sysadm/system_prep -v -s system
```

  Run "kmtune" to see the current kernel parameters.

  Change a kernel parameter (let's say shmseg is currently set to 120 and we want it set to 128):

```
kmtune -s shmseg=128

   or add a driver to the kernel:

/usr/sbin/kmsystem -S /stand/build/system -c Y driver-name
```

  Build the new kernel:

```
/usr/sbin/mk_kernel -s /stand/build/system
```

  Perform a kernel update (moves /stand/vmunix to /stand/vmunix.prev and puts the new kernel into place as /stand/vmunix):

```
/usr/sbin/kmupdate
```

  Move the previous kernel just in case the new one is bad:

```
mv /stand/vmunix.prev /stand/vmunix.bkup
```

  Reboot the box: reboot or shutdown -r

  Uh-Oh - If you've made a bad kernel, boot to a good one from the ISL prompt (see "Other Kernel stuff" below for other ways to boot to a new kernel):

```
hpux (;0)/stand/vmunix.bkup
```

  Remove the bad kernel (this assumes you've booted to the "bkup" kernel):

```
kmupdate -d /stand/vmunix

NOTE:
This will remove /stand/vmunix as we know it.
Can you say Buh-Bye?
```

  Rebuild the kernel:
  You've booted to the backed-up version which was good. The box is up and running but now doesn't have a standard issue kernel to boot from. Remember the "kmupdate -d /stand/vmunix" you ran above? You now need to create a "new" kernel so go back to the top of this section and rebuild the kernel.

```
NOTE:
Remember NOT to overwrite the known good
kernel - /stand/vmunix.bkup
```

- Boot from desired kernel while in ISL mode:
  - Boot normally from the previous kernel:

```
hpux (;0)/stand/vmunix.prev
```

  - Boot in "single user" mode from the previous kernel:

```
hpux -is (;0)/stand/vmunix.prev
```

  - Boot in "lvm maintenance" mode from the previous kernel:

```
hpux -lm (;0)/stand/vmunix.prev


Note:
To boot from some other kernel, replace the kernel
file name "vmunix.prev" with the name you gave your
alternate kernel.
```

- The kernel files are:
  - A file containing the kernel itself:

```
/stand/vmunix
```

- A file containing the kernel configuration:

  ```
  /stand/system
  ```

- A directory containing a few files and directories for kernel configuration purposes (for HP-UX 11.0 and up only)

  ```
  /stand/dklm/*
  ```

Check Kernel parameters

- To check the current kernel parameters:

  ```
  kmtune -l   or   sysdef
  ```

SAM Kernel Settings

```
NOTE:
The old selections for DataBase server kernel changes are not
available in 11i. To have these again, download the following
files and place them in the /usr/sam/lib/kc/tuned directory.
Then start up SAM and select the one you need.

General OLTP/Database Client System      oltp_cli.tun
General OLTP/Database Monolithic System  oltp_sa.tun
General OLTP/Database Server System      oltp_ser.tun
```

Manually reconfigure the kernel

- Use the "system_prep" command to extract the system file from the current kernel.

  ```
  cd /stand/build
  system_prep -v -s system
  ```

- Edit the /stand/build/system file to perform your task or use "kmsystem" and "kmtune" to make changes. Build the new kernel:

  ```
  mk_kernel -s system
  ```

- Move the old system file and kernel so if anything goes wrong, you still have a bootable kernel:

  ```
  mv /stand/system /stand/system.prev
  ```

- Update the kernel with the newly generated kernel. Automatically saves the old versions of vmunix and dlkm in /stand:

  ```
  kmupdate /stand/build/vmunix_test
  ```

- Reboot the system.

Awesome Kernel Change

```
* Configurable IPC Message Parameters:
      mesg      Enable/disable IPC messages (Series 700 only)
         Minimum:    0 (Exclude System V IPC message parameters from
                       kernel)
         Maximum:    1 (Include System V IPC message parameters in kernel)
         Default:    1
      msgmap    message free-space map size
         Minimum:    3
         Maximum:    Memory limited
         Default:    msgtql+2
      msgmax    maximum message size
         Minimum:    0
         Maximum:    65535 bytes
         Default:    8192 bytes
      msgmnb    maximum bytes in message queue
         Minimum:    0
         Maximum:    65535 bytes
         Default:    16384 bytes
      msgmni    maximum message queues on system
         Minimum:    1
         Maximum:    Memory limited
         Default:    50
      msgseg     number of segments in message queue
         Minimum:    1
         Maximum:    32767
         Default:    2048
      msgssz     message segment size
         Minimum:    1
         Maximum:    Memory limited
         Default:    8 bytes
      msgtql     maximum total messages on system
         Minimum:    1
         Maximum:    Memory limited
         Default:    40

* Configurable IPC Semaphore Parameters:
      sema      Enable/disable semaphores (Series 700 only)
         Minimum:    0 (exclude System V IPC semaphore code from kernel)
         Maximum:    1 (include System V IPC semaphore code in kernel)
         Default:    1
      semaem    Semaphore value-change limit
         Minimum:    0
         Maximum:    semvmx or 32767, whichever is smaller
         Default:    16384
```

```
      semmap    Size of free-semaphore resource map
         Minimum:    4
         Maximum:    Memory limited
         Default:    semmni+2
      semmni    Maximum semaphores system-wide
         Minimum:    2
         Maximum:    Memory limited
         Default:    64
      semmns    Maximum user-accessible semaphores system-wide
         Minimum:    2
         Maximum:    Memory limited
         Default:    128
      semmnu    Maximum undos per semaphore
         Minimum:    1
         Maximum:    nproc-4
         Default:    30
      semume    Maximum semaphore undos per process
         Minimum:    1
         Maximum:    semmns
         Default:    10
      semvmx    Maximum allowed semaphore value
         Minimum:    1
         Maximum:    65535
         Default:    32767

* Configurable IPC Shared Memory Parameters:
      shmem     Enable/disable shared memory (Series 700 only)
         Minimum:    0 (exclude System V IPC shared memory code from
                        kernel)
         Maximum:    1 (include System V IPC shared memory code in kernel)
         Default:    1
      shmmax    Maximum shared memory segment size
         Minimum:    2 Kbytes
         Maximum:    1792 Mbytes
         Default:    0x04000000 (64 Mbytes)
      shmmni    Maximum segments on system
         Minimum:    3
         Maximum:    (memory limited)
         Default:    200 identifiers
      shmseg    Maximum segments per process
         Minimum:    1
         Maximum:    shmmni
         Default:    120
```

Check core dumps

- What made the stinkin core in the first place:

  ```
  file core
  ```

- To get the libs used:

  ```
  what core
  ```

- To debug the stinker:

  ```
  gdb -c core
  ```

- To debug the stinker:

  ```
  gdb ProgramName -c core
  ```

- Also see HP doc - KNC071499001

Swap

- Get info: swapinfo -a
- Opens entire system to swap: swapon /

Find and Move stuff

Find stuff

- Find 777 (rwxrwxrwx) files.

  ```
  find / \( -perm 000 \)  -exec ls -ld {} \;
  ```

- Find set-owner-id and set-group-id files:

  ```
  find / \( -perm -4000 -o -perm -2000 \) -type f -exec ls -ld {} \;
  ```

- Another way to find SUID and SGID stuff:

  ```
  for stuff in `awk '($3 ~ /^(hfs|vxfs)$/) { print $2 }' /etc/fstab`
    do
         find "$stuff" \( -perm -04000 -o -perm -02000 \) -type f -xdev -print
  done
  ```

- Find core files (not directories):

  ```
  find / -type f -name core -exec rm {} \;
  ```

- Find uppercase files, i.e.; Uppercase or UPPERcase

  ```
  find . -xdev -type f -name "[[:upper:]]*"
  ```

- Find files that a specific user owns.

```
find / \( -user oracle \) -type f -exec ls -ld {} \;
```

- Find then change files (and possibly the current directory) from one user name to a new user ID:

```
find . -user user_name | xargs chown new_user_name
```

- Grep "fsck" (or any other named item) in all files and list out the path to that file:

```
find / -name "*" -exec grep -l -i fsck {} \;
```

- Find files that are 1,000,000 (1 meg) or larger.

```
find /var -size +1000000c -exec ll {} \;
```

- Find files that are 100 meg or larger and sort them.

```
find . -size +100000000c -exec ls -l {} \; | sort -n -k5,5
```

- Try to delete a lot of files in /var/tmp (or wherever) and getting "The parameter list is too long". Do a find to get rid of the stuff.

```
find /var/tmp . -print -exec rm {} \;
```

or do this to "rm" them:

```
echo * | xargs rm
```

Move (copying) /usr

- Copy a directory from one location to another. Even if it has "special" files and links in it:

```
cd /usr
find . -print | cpio -dlmpuvx /usr2
  d = Do directories
  l = Do links
  m = Keep time
  p = Pass along
  u = Copy unconditionally
  v = verbose
  x = Save device special files
```

Backup stuff

General stuff

- Backup files and/or directories:

```
fbackup -0 -f /dev/rmt/c?t?d?BEST -I /tmp/index -v

  0 = Or "1" - (0 = full, 1 = incremental)
  f = Device file to use
  I = Directory to place index
  v = Verbose

frecover -f /dev/rmt/c?d?t?BEST -v -I /tmp

mt -t /dev/rmt/c?t?d?BEST rew
  rew   = Rewind tape.
  offl  = Rewind tape and go off line.
  fsf 5 = Go forward five setmarks on the tape.
  bss 2 = Go backwards two setmarks on the tape.

tar cvf /path/file_name (or device_file_name) /path/target_to_copy
  c = Copy
  x = Restore
  v = Verbose
  t = Index
  r = Add to the end of list
```

- To copy files to a tarball and gzip at the same time.

```
tar cvf - * | gzip -9> filename.tar.gz

NOTE:
Change the "*" to whatever filename or directory you chose or
you can use the "*" to grab everything in your current directory.
```

OmniBack stuff

- Robotarm stuff:

```
uma -ioctl /dev/rac/c?t?d?

**** or ****

/usr/sbin/mc -p /dev/rac/c?t?d? -s D1 -d S1 -r IDSM
    s = Source (the drive in the library).
    d = Tape source
    r = Print out status
```

- To get version info: omnidb -version

- Stop OmniBack (or do "/sbin/init.d/omni stop"): omnisv.sh stop
- Start OmniBack (or do "/sbin/init.d/omni start"): omnisv.sh start
- List all sessions in the database: omnidb -session
- List database space usage: omnidbutil -info
- Best info on specific database files: omnidbutil -extendinfo
- Add an "fvers.dat" file with it's maximum size of 2 gig. Total of 4 "fvers.dat" files can be made.

```
omnidbutil -extend /var/opt/omni/db/cdb -maxsize 2048
```

- Add an "fnames.dat" file with it's maximum size of 2 gig. Total of 2 "fnames.dat" files can be made.

```
omnidbutil -extendfilenames /var/opt/omni/db/cdb -maxsize 2048
```

- To clean up the database a bit:
  1. Delete items from the OmniBack "Monitor" GUI window
  2. Verify that no OmniBack sessions are running
  3. Strip the database using "omnidb -strip"
  4. More cleaning using "omnidbutil -clear"
  5. Do a purge using "omnidbutil -purge -detail -days ?"

```
purge  = Purge stuff
detail = Out of date file versions and names are removed
days   = Data older then "?" days are removed
```

- Copy the OmniBack database to another filesystem:
  1. mkdir /new_omni_database_location
  2. Create the new filesystem and mount it.
  3. cd /var/opt/omni/db
  4. find . -xdev -depth -print | cpio -pxdm /new_omni_database_location
  5. ln -s /var/opt/omni/db /new_omni_database_location

- The syntax for the exclude option in /etc/opt/omni/datalists/filename_related_to_a_server is:

```
/path         for Unix (/blah/de/blah)
Drive:\path   for NT (c:\blah\de\blah)
Volume:\path  for NetWare (sys:\blah\de\blah)
```

- For Host backup, the syntax used is:

```
/path         for Unix (/blah/de/blah)
Drive:\path   for NT (c:\blah\de\blah)
Volume:\path  for NetWare (sys:\blah\de\blah)
```

- Stuff placed in the /etc/opt/omni/NOTIFICATION file:

```
NOTE:
The item - "backup" - shown below is an email alias.
It can be changed to any email address or alias.

NOTIFICATION "Device Error"
{ -event "DeviceError"
-object "*"
-email "backup"

NOTIFICATION "Tape Mount Request"
{ -event "MountRequest"
-object "*"
-email "backup"
}

NOTIFICATION "DB Space Low"
{ -event "DbSpaceLow"
-object "*"
-email "backup"
}

NOTIFICATION "Backup Finished"
{ -event "EndOfSession"
-object "*"
-email "backup"
}
```

Veritas stuff

- Check Veritas Netbackup process:

```
/usr/openv/netbackup/bin/bpps -a
```

- Start Veritas GUI:

```
/usr/openv/netbackup/bin/xnb
```

- Check if drives are up:

```
/usr/openv/volmgr/bin/tpconfig -d
```

- Pending requests:

```
/usr/openv/volmgr/bin/vmoprcmd
```

- Bring a drive up:

```
/usr/openv/volmgr/bin/vmoprcmd -up 0 (0 = drive number)
```

- Run the robot arm (like UMA):

  ```
  /usr/openv/volmgr/bin/robtest
  ```

- Tape GUI:

  ```
  /usr/openv/volmgr/bin/vxadm
  ```

Errors and such

EMS is barking but it's ok

- If EMS has found something bad. Your getting hammered with email. However, it's ok *for the time being* that the part is bad. Configure EMS not to notify you about the bad part.

  To stop the EMS email run EMS (or look at your email) and add the EMS line to this file - /var/stm/data/tools/monitor/disabled_instances

  Lines may look like this:

  ```
  /adapters/events/FC_adapter/8_0.8
  /adapters/events/FC_adapter/8_4.0
  /storage/events/disks/default/10_0.6.0
  ```

  1. Run EMS: /etc/opt/resmon/lbin/moncheck
  2. K)ill monitoring (Stop here to simply turn EMS off)
  3. In another window add the line to: /var/stm/data/tools/monitor/disabled_instances
  4. E)nable Monitoring
  5. Do a C)heck monitors just for fun

Corrupted tracing and logging header

- Getting this at bootup:

  ```
  "Corrupted tracing and logging header
        Internal error (210)"

  The fix, run:

    /usr/sbin/nettl stop
    rm /usr/adm/nettl.log00
    /usr/sbin/nettl start
  ```

Enter key in an xterm window doesn't work correctly

- You open an xterm and hit "Enter" a couple of times and the line doesn't move up. If you make the stinkin window smaller it will start to scroll up the way it should. Well.... run this next time:

  ```
  eval `resize`
  ```

Get statistics and information

STM Information

- All hardware.

  ```
  echo 'selclass qualifier all;info;wait;infolog' | /usr/sbin/cstm
  ```

- Memory.

  ```
  echo "selclass qualifier memory;info;wait;infolog"|cstm
  ```

CPU Information

- To count CPUs (Processors), you can either use adb, ioscan, or cstm (if you have OnLineDiags installed).

  ```
  echo "processor_count/D" | adb /stand/vmunix /dev/kmem

  echo "ioscan -FkC processor | wc -l"

  echo "map\nquit\nok\n" | cstm | grep CPU | wc –l
  ```

- To report speed of CPU:

  ```
  echo itick_per_usec/D | adb -k /stand/vmunix /dev/mem
  ```

- Get type of CPU:

  ```
  pdcinfo -no_banner|grep PA|grep -v PATH|grep -v PAR|grep -v HPA
  ```

Drive Information

- Drive info every 30 sec, for 20 times:

  ```
  vmstat -dnS 30 20
  ```

- Same as vmstat:

[Top]

```
iostat 30 20
```

SAR Information

- Have "sar" collect data every 30 sec, 20 times:

  ```
  sar -A -o /tmp/sar_info 30 20
  ```

- Read the data from a given file (normally in /var/adm/sa/sa##):

  ```
  sar -A -f /tmp/sar_info
  ```

- Additional SAR info can be found on my Server Build doc.

Kernel Information

ipcs -mqsabcopt

32 or 64 bit supported server

- Check /etc/.supported_bits:

  ```
  grep `/usr/bin/model` /etc/.supported_bits
  ```

- See what an 11.x server is running:

  ```
  getconf KERNEL_BITS or chatr /stand/vmunix
  ```

- To see what version has been loaded:

  ```
  swlist -l bundle | grep Runtime
  ```

User stuff

- Change or add files in /etc/skel. When you add a new user they get all the new/configured files.
- Copy the /etc/skel files for a new user:

  ```
  cp /etc/skel/.[A-z]* /home/new_user
  ```

- List all added users and there comments found in the /etc/passwd file.

  ```
  listusers
  ```

- Add a user.

  ```
  useradd -u 101 -g 200 -c "Doug Burton" -d /home/dburton -s /usr/bin/ksh -m -e 08/01/2020 dburton
    -u = UID
    -g = GID
    -c = Comments
    -d = Users home directory
    -s = Users shell
    -m = Creates the home directory
    -e = Password expiration date
  ```

- I'm not recommending this due to possible security reasons, but this is pretty cool. Put this little script into ROOTs .profile and create the link. What this does is let root use YOUR user .profile to set $PATH and other stuff.

  ```
  # custom homedir for root users
  UID=`id | cut -d'=' -f2 | cut -d'(' -f1`
  [ "$UID" = "0" ] && {
      USER=`who am i | cut -d' ' -f1`
      echo "Setting up root environment for $USER"
      [ -L /home/root.$USER ] && {
         export HOME=/home/root.$USER
         . ~/.profile
      }
  }
  ```

  The link: ln -s /home/dburton /home/root.dburton

- Put an alias in roots profile (to speed up things).

  ```
  alias ra='/usr/lbin/modprpw -l -k'
  alias ga='/usr/lbin/getprpw -m lockout'
  ```

  then type "ra account_name" to reset the account then type "ga account_name" to ensure it is complete. You should see this: lockout=0000000

- Configure the .profile, .kshrc, .exrc and .logout files (See my Server Build page).

How to boot to single user mode

- For 9000/700 series:

  ```
  b p0 ipl
  isl>hpux -is boot disc(;0)/stand/vmunix
  ```

  - Boot to an alternate kernel in single user mode:

    ```
    isl>hpux -is boot disc(;0)/stand/vmunix.prev
    ```

- For 9000/800 series
  - Search for any bootable devices (LAN, tape, CD or drive).

    ```
    sea ipl
    ```

  - The "?" is the p number given for the bootable item you want to use (after doing your search).

    ```
    bo p? ipl    or    bo pri ipl
    ```

  - Select "YES" if asked to interact with IPL.

    ```
    hpux -is (;0)/stand/vmunix    or    "hpux -is"
    ```

Command line stuff

- Run some command forever until you ctrl-c the thing:

  ```
  while (true) ; do command ; sleep 2 ; done
  ```

- Check out a few drives, one right after another:

  ```
  for x in 1 2 3 ; do pvdisplay "/dev/dsk/c3t1d${i}" ; done | more
  ```

Set the stinkin date

- To set the system date (month=mm, day=dd, hour=hh, minute=mm, year=yy):

  ```
  date -u mmddhhmmyy
  ```

Setting/Resetting some system parameters

```
set_parms hostname
set_parms timezone
set_parms date_time
set_parms root_passwd
set_parms ip_address
set_parms addl_netwrk
```

Give root a REAL ksh shell at bootup

- Change the "/sbin/sh" line for root in the /etc/passwd file to "/sbin/ksh" then do this:

  ```
  cp -p /bin/ksh /sbin/ksh
  ```

  NOTE:
  If you do a "file /bin/ksh" (or /usr/bin/ksh), you'll notice that this shell is dynamically linked. In theory, you can't use this shell in single user mode due to the needed libs not being mounted. I don't know why, but I have used it with no adverse results. You need to test in your environment. In other words, *use at your own risk*.

  ADDITIONAL NOTE:
  Just to seal the deal, HP does not support the use of root using ksh at bootup.

Check /etc/passwd and /etc/group files

- Check the /etc/passwd and /etc/group files

  ```
  pwck  and  grpck
  ```

Rebuild /dev/null

```
mknod /dev/null c 3 0x000002
chown bin:bin /dev/null
chmod 666 /dev/null
```

Man Page Stuff

```
Run "catman" to build the "/usr/share/lib/whatis" file.

Then you can run the "man -k [some command] to see what
manual sections the command is in. For example, if you run
"man -k mkdir", it will show that "mkdir" is in
sections 1 and 2.

To see section 2 you would run "man 2 mkdir".
```

Filesystem filling up?

- Look for big files (see my "find" commands on this page). Also look here:

```
/var/adm/crash
/var/adm/syslog
/var/tmp
/tmp
/var/adm/lp
/var/mail
/var/adm/wtmp (to big?)
/dev/rmt/om (instead of /dev/rmt/0m)
/var/adm/lp/log
/var/spool/lp/request/
/var/spool/cron/tmp
/var/sam/log
/var/spool/mqueue
/usr/etc/yp
lost+found (found in any mounted file system)
Did you mount a filesystem *ON* another filesystem (thus hiding what's underneath)
Remove core files
Extra kernels saved in /stand
```

Copy a set of printers from one server to another

On server "A"

1. Open "sam ->Printers and Plotters->LP Spooler-> Save/Restore Spooler Configuration"
2. Go to "Actions"
3. Click on 'Save Spooler Configuration'
4. Exit out of SAM
5. tar cvf /tmp/sam.tar /var/sam/lp
6. Transfer the sam.tar file to system B:/tmp.

On server "B"

1. tar xvf /tmp/sam.tar
2. You may need to change the permissions and ownership of these files. To do that run:
   chmod -R 755 /var/sam/lp/* ; chown -R lp:bin /var/sam/lp/*
3. Open "sam ->Printers and Plotters->LP Spooler-> Save/Restore Spooler Configuration"
4. Go to "Actions"
5. Click on 'Restore Spooler Configuration'
6. Don't forget to restart the scheduler
7. Exit out of SAM

Console settings for N and L servers

- Log off the terminal before making these changes to the terminal configuration. Then perform the following steps on a 700/96 terminal:

    1. Press the user/system key to get to the configuration screen.
    2. Press f8 (config keys).
    3. Press f5 (terminal config).
    4. Tab to the Terminal Id field.
    5. Type VT100 in this field.
    6. Tab to the TermMode field.
    7. Press f2 (next choice), EM100 appears in the field.
    8. Press f1 (save config).
    9. Use the f8 (config keys) to go to the ansi config screen.
    10. Press f6 (ansi config).
    11. Tab to the backspace Def field.
    12. Press f2 (next choice).
    13. Verify the Backspace Def field now reads BackSpace/Del in the field.
    14. Tab to the EM100 ID field.
    15. Press f2 (next choice) to change the EM100 ID field to EM100.
    16. Press f1 (save config).

- The terminal should now display EM100 below the softkeys.
- This procedure enforced the following settings:

```
* terminal id to vt100,
* termmode to em100,
* correctly mapped the backspace and delete keys
* em100 id to EM100
```

- The terminal is now ready to correctly operate in vt100 mode.

Hitachi/Controller time out

- To get Hitachi controller info:

```
ioscan -fun
```

- Check the timeout, should be 60:

```
pvdisplay /dev/dsk/c?t?d? | grep Timeout
```

- Change the timeout value to 60:

```
pvchange -t 60 /dev/dsk/c?t?d?
```

- Check the queue depth, should be 2:

```
scsictl -a /dev/rdsk/c?t?d?
```

- Change it to the value of 2:

```
scsictl -m queue_depth=2 /dev/rdsk/c?t?d?
```

How many floppies in a ....

Well, I was talking to my brother some time back. Told him about some new SAN we got in. It had Terabytes of storage. "Ok" he said. "What's a Terabyte?" I started to tell him roughly how big it was. Tried to include his hard drive as a ref. His eyes glazed over. Drool started to come out. What a mess. I then told him a Terabyte is roughly a bunch of floppies laid end-to-end from St. Petersburg (Florida) to Orlando. It's actually a bit less but he got that. So I thought why not figure it out and put it down someplace so I can ref it again if I need it. So, here's what I figured out (I think it's pretty accurate), a floppy is 1.44 megabytes and is 3.5 inches long (or wide if you look at it that way):

How many floppies are in a Gigabyte, Terabyte, Petabyte, Exabyte and a Zettabyte					
	Gigabyte	Terabyte	Petabyte	Exabyte	Zettabyte
How many MB in a:	1,024	1,048,576	1,073,741,824	1,099,511,627,776	1,125,899,906,842,620
How many 1.44 MB floppies are in a:	711.11	728,177.78	745,654,044.44	763,549,741,511.11	781,874,935,307,378.00
How many feet of floppies are in a:	207.41	212,385.19	217,482,429.63	222,702,007,940.74	228,046,856,131,318.00
How many miles of floppies are in a:	0.04	40.22	41,189.85	42,178,410.59	43,190,692,449.11
	7/11 down the street	St. Petersburg, FL to Plant City, FL	About 5.19 times around the earth	Distance from the Earth to Mars	More than ten times the farthest distance of Pluto from the Sun

---

Stuff - I'm not sure where to put....

- Stuff #1
  - Open a window and run:

    ```
    script -a /tmp/STUFF.`date +%Y%m%d`
    ```

  - Open another window and "*tail -f /tmp/STUFF.`date +%Y%m%d`*" the file above.
  - Go to the first window and do "ll" or some such thing. EVERYTHING typed AND displayed is in the target file. Even SAM gets displayed correctly if you do a tail -f. If you checkout the file later (cat -v for example), SAM output looks weird. Not sure if you can "really" see SAM window(s) after the fact.
  - How about putting this into a users .profile?

- Stuff #2
  - Restoring Default File Permissions/Ownership:

    Somebody did a "chmod -R 777 /usr" for some stupid reason. Now you need to restore all the files to whatever perms and ownership they originally had.

    ```
    I assume your on the server in at least init level 2. Do this chmod so we
    can start fixing this mess.

    /sbin/chmod ugo+rx /usr/bin/cut /usr/bin/expand /usr/bin/grep /usr/sbin/swlist

    Then this: comp="/usr"
        (or this if there is more than one directory messed up)
               comp=' -e ^/usr  -e ^/var -e ^/sbin/init.d'

    And away we go...

       swlist -l file -a type -a path -a mode -a owner -a group | expand | grep \
       $comp | cut -f 2 -d : | /sbin/awk '{if(length($2) < 2 && $4 > 0)         \
       print "/sbin/chmod "$4,$3";/sbin/chown " $5":"$6,$3}' | /sbin/sh
    ```

Unix (Sun) Stuff

Support number: 1-800-USA-4SUN (800-872-4786)

- Get motif version info:

  ```
  /usr/ccs/bin/what /usr/dt/lib/lubXm.s0
  ```

- Free up /var... (Blow away "save" files):

  ```
  cd /var/sadm ; find . -name save
  ```

- To change the openwin logo to our (tif) logo, copy our file to:

  ```
  /usr/openwin/server/etc/Solaris.im8.Z
  ```

- Patch location:

  ```
  /var/sadm/patch
  ```

- Other worth while commands:

  ```
  scsiinfo -p
  boot -r
  sysdef -d
  devinfo
  prtconf
  proctool    - Like Glance on HP
  probe-scsi  - Like "ioscan" on HP
  admintool   - Like SAM on HP
  ```

  Check the **Rosetta Stone** for other commands similar to HP commands.

Scripts

```
Nothing yet........
```

Links to other interesting Unix sites

I moved all the links that had been here to my Links page.

I developed this site so please send comments to dburton3@tampabay.rr.com. Thanks!

Ok.. so I put this waaay down at the bottom of this web page. I don't expect anyone to actually toss a buck or two my way as a thank you for all the work I've done on this site (Hmmm... feel guilty yet?), but it sure would be nice.

Please donate!

U.S. Dollar Make a Donation

This donation link uses a secure PayPal connection.