VIOS

Network

Create Network Interface Backup (NIB) adapter

```
mkvdev -lnagg entX -attr backup_adapter=entY
```

Create Link Aggregated (802.3ad) adapter

```
mkvdev -lnagg entX entY -attr mode=8023ad hash_mode=src_dst_port
use_jumbo_frames=yes
```

Create Shared Ethernet Adapter (SEA)

```
mkvdev -sea entX -vadapter entX,entY -default entX -defaultid XXX -attr ha_mode=sharing
chdev -dev entX -perm -attr jumbo_frames=yes large_receive=yes largesend=1
chdev -dev entX -perm -attr  accounting=enabled
```

Check SEA mapping

```
lsmap -all -net
```

List active VLAN's being sent over the SEA

SEA en* device needs to be up in ifconfig, but doesn't need an IP address

```
tcpdump -i en19 -s 1500 -nn -e vlan
```

Manually change SEA failover mode

```
chdev -dev entX -attr ha_mode=<auto|sharing|standby>
```

Reset ethernet adapter statistics

```
entstat -reset entX
```

SEA statistics

Requires accounting to be enabled on adapter

```
seastat -d ent5 [-c] [-n search_criteria=value]
```

- `-c` resets statistics
- `-n` search on particular criteria
- `mac=<value>`
- `ip=<value>`
- `vlan=<value>`
- `host=<value>`

Preallocate virtual ethernet buffers

```
lsdev | awk '/Virtual I\/O Ethernet Adapter/{ print $1 }' | while read -r VETH; do chdev -l "${VETH}" -a max_buf_tiny=4096 -a min_buf_tiny=4096 -a max_buf_small=4096 -a min_buf_small=4096 -a max_buf_medium=2048 -a min_buf_medium=2048 -a max_buf_large=256 -a min_buf_large=256 -a max_buf_huge=128 -a min_buf_huge=128 -P; done
```

Storage

NPIV map client adapter to server FCS port

```
vfcmap -vadapter vfchost0 -fcp fcs0
```

To unassign the mapping, just leave the physical adapter off the command

```
vfcmap -vadapter vfchost0 -fcp
```

View NPIV mappings

Specific client adapter mapping

```
lsmap -vadapter vfchost0 -npiv
```

All client adapter mappings (CSV formatting)

```
lsmap -all -npiv -fmt ,
```

Show all client WWPN's from the VIOS

```
echo "svfcCI; svfcPrQs; svfcva" | kdb -i
```

Verbose adapter stats

```
echo "svfcCI; svfcPrQs; svfcva vfchost1" | kdb -i`
```

Replace FCS adapter being used with NPIV

1. Verify port(s) that are offline

The output below shows fcs5 isn't listed as NPIV capable. Even after a port down/up by the SAN team, the port still shows as offline.

```
$ lsnports
name            physloc                        fabric tports aports swwpns  awwpns
fcs0            U78CA.001.CSS134Y-P1-C2-C1-T1       1     64     30   3088    2984
fcs1            U78CA.001.CSS134Y-P1-C2-C1-T2       1     64     29   3088    2981
fcs4            U78CA.001.CSS134Y-P1-C4-C1-T1       1     64     30   3088    2976
fcs6            U78CA.001.CSS112R-P1-C2-C1-T1       1     64     27   3088    2965
fcs7            U78CA.001.CSS112R-P1-C2-C1-T2       1     64     26   3088    2962
fcs10           U78CA.001.CSS112R-P1-C4-C1-T1       1     64     37   3088    3005
fcs11           U78CA.001.CSS112R-P1-C4-C1-T2       1     64     37   3088    3005
```

2. Put vfchosts into a Defined state

As these are two port adapters, fcs4 which is still functional has active NPIV maps, so they need to be put into a Defined state. The below command shows all the current mappings down fcs4 and fcs5. The mappings down fcs5 wont show if the port isn't available.

```
lsmap -all -npiv -fmt : | awk -F: '$7 ~ /fcs[45]/'
```

Loop through above vfchost adapters and put them into a Defined state.

```
lsmap -all -npiv -fmt : | awk -F: '$7 ~ /fcs[45]/{ print $1 } | while read -r VFC; do rmdev -dev ${VFC} -ucfg; done
```

3. Put the physical ports of the card being replaced into a Defined state

```
rmdev -dev fcs4 -ucfg -recursive
rmdev -dev fcs5 -ucfg -recursive
```

4. Replace adapter and bring it back online

Once the card is verified as being back online, bring the devices back into an Available state. This should also put back all the above vfcmap's.

```
cfgdev
```

Virtual Media Library

Define a VML

```
$ mkrep -sp rootvg -size 10G
Virtual Media Repository Created
Repository created within "VMLibrary" logical volume
```

Add ISO image(s) to VML

```
# ls -l /var/vio/VMLibrary/AIX-6100-06-01-Disk1.iso
-r--------    1 root     system   4236148736 Apr 06 18:07 /var/vio/VMLibrary/AIX-6100-06-01-
Disk1.iso
$ lsrep
Size(mb) Free(mb) Parent Pool         Parent Size      Parent Free
   10198     6158 rootvg                 279552           236288

Name                            File Size Optical         Access
AIX-6100-06-01-Disk1.iso                4040 None            ro
```

Create File Backed Optical Device for LPAR

```
$ mkvdev -fbo -vadapter vhost0
vtopt0 Available
```

Load image into the vtopt0 device

```
$ loadopt -vtd vtopt0 -disk AIX-6100-06-01-Disk1.iso
$ lsmap -vadapter vhost1
SVSA            Physloc                                     Client Partition ID
--------------- -------------------------------------------- ------------------
vhost0          U9133.55A.066EB4H-V1-C34                    0x00000000

VTD                   vtopt0
Status                Available
LUN                   0x8200000000000000
Backing device        /var/vio/VMLibrary/AIX-6100-06-01-Disk1.iso
```

```
Physloc
Mirrored        N/A
```

Load a second disk

Can be used if you're prompted to insert disk 2 (for example)

```
loadopt -f -vtd vtopt0 -disk AIX-6100-06-01-Disk2.iso
```

Unload the image

```
unloadopt -vtd vtopt0
```

Shared Storage Pools

The following parameter are used when working with Shared Storage Pools (SSPs)

Parameter	Description
`clustername`	Name of the cluster
`repopvs`	Disk which will contain the repository (Around 1GB LUN)
`spname`	Name of the storage pool
`sppvs`	Disks in the storage pool
`hostname`	FQDN of the VIOS

Create cluster

```
cluster -create -clustername <name> -repopvs <hdiskX> -spname <sp_name> -
sppvs <hdiskY> <hdiskX> -hostname <vios_hostname1> <vios_hostname2>
```

Remove the cluster

```
cluster -delete -clustername <name>
```

Add node to cluster

```
cluster -addnode -clustername <name> -hostname <vios_hostname3>
```

List Shared Storage Pools in a cluster

```
lssp -clustername clusterA
```

Allocate disk to VM

```
mkbdsp -clustername <name> -sp <sp_name> xGB -bd <device_name> -
vadapter <vhost> [-thick]
```

- `<device_name>` is the virtual disk name, and can be whatever you want

Remove disk from VM

```
rmbdsp -clustername <name> -sp <sp_name> -bd <device_name>
```

- `<device_name>` The disk udid can be used here instead

Monitor disk pool

```
lssp -clustername <name> -sp <sp_name> -bd
lssp -clustername <name>
```

Add more LUN's to a pool

```
chsp -add -clustername <name> -sp <sp_name> <hdiskX> <hdiskY>
```

List nodes in a cluster

```
cluster -listnode -clustername <cluster_name>
```

Clean disk of all cluster information

Repository disk

```
cleandisk -r <hdisk>
```

Storage Pool Disk

```
cleandisk -f -s <hdisk>
```

List cluster storage interfaces

```
lscluster -d
```

Storage Pool threshold alerts

```
alert -list -clustername <cluster_name> -spname <sp_name>
alert -set -clustername <cluster_name> -spname <sp_name> -type threshold -
value <value>
```

Miscellaneous

Setup PS1 prompt

Append the follow in `/usr/ios/cli/.profile`

```
export ENV=/home/padmin/.kshrc
```

Append the following in `/home/padmin/.kshrc`

```
export EXTENDED_HISTORY=ON
export EDITOR=/usr/bin/vi
export HOST="$(/usr/bin/uname -n)"
export ENTSTAT_MODE=closed.error

alias aix="ioscli oem_setup_env"
alias history='fc -t'

if [ "$(whoami)" != "root" ]; then
  export PS1="[$(whoami)@${HOST}]$ "
else
  export PS1="[$(whoami)@${HOST}]# "
fi

set -o vi
```

Overall monitoring of all LPARs

```
topas -cecdisp
```

- Press `p` to see the Shared Processor Pools

- Press `~` to toggle between nmon and topas
- Press `E` to see SEA stats

See the underlying AIX commands being run from padmin shell

```
export CLI_DEBUG=33
```

Schedule backup of virtual and logical configuration

```
viosbr -backup -file `hostname` -frequency daily -numfiles 7
```

Scripts

sea_portinfo

Determine the physical switch name and port number by sniffing CDP packets.

```
#!/bin/ksh
#
# Name:
#       sea_portinfo.sh
#
# Description:
#       Tries to determine the physical switch name and port number by sniffing
#       CDP packets. Only works on Cisco switches and only if the port is configured to
#       publish CDP packets.
#
# Usage:
#       Optional:
#               -c : CSV output format.
#
# Requirements:
#       timeout - Comes as part of the coreutils RPM which can be installed with yum.
#
# Note:
#       This script isn't perfect. Sometimes the packet captured isn't always
#       the CDP one, so you may need to run it multiple times to get the switch details when
#       the real adapter is a link aggregation of multiple physical ports.

hostname=$(hostname -s | tr '[:upper:]' '[:lower:]')

# Arguments
while getopts "c" OPTION; do
   case ${OPTION} in
      c) CSVOUT="yes"
         ;;
      *) CSVOUT="no"
         ;;
   esac
```

```
done

if [[ ${CSVOUT} = "no" ]]; then
   printf '\n%s\n' "Sniffing Shared Ethernet Adapters (SEA) for switch details"
   printf '%s\n' "This may take up to 90 seconds per SEA..."
fi
# Iterate through all SEA's
for sea in $(lsdev -F 'name description' | awk '/Shared Ethernet Adapter/{ print $1 }'); do
   # Get SEA real adapter
   real_adapter=$(lsattr -E -a real_adapter -F value -l "${sea}")

   # If the real adapter is an EtherChannel, get a count of the number of aggregated
interfaces
   if [ "$(lsdev -l "${real_adapter}" -F 'description')" = "EtherChannel / IEEE 802.3ad Link
Aggregation" ]; then
      ent_count=$(lsattr -E -a adapter_names -F value -l "${real_adapter}" | awk -F ','
'{ print NF }')
   else
      ent_count=1
   fi

   # Bring SEA's en adapter up (if not already)
   sea_en_state=$(lsattr -E -a state -F value -l en"${sea#ent}")
   if [ "${sea_en_state}" != "up" ]; then
     chdev -a state=up -l en"${sea#ent}" > /dev/null 2>&1
   fi

   # Capture CDP packets from the SEA interface. If tcpdump is still running after 90
seconds,
   # we may have hit a switch that either isn't Cisco, or the port doesn't have CDP enabled
   unset switchinfo
   set -A switchinfo $(/home/padmin/scripts/bin/timeout 90s tcpdump -i en"${sea#ent}" -s
1500 -c "${ent_count}" -v 'ether[20:2] = 0x2000' 2>&1 | grep -E '(Device-ID|Port-ID)' | awk
-F ':' 'v && NR==n{ gsub(/ /, "", v); gsub(/ /, "", $3); print v","$3}/Device-ID/{ v=$3;
n=NR+1 }')
   switchinfo=${switchinfo:-"NONE"}

   # If the SEA interface was down when we first checked, lets put it back in that state
   if [ "${sea_en_state}" = "down" ]; then
      chdev -a state=down -l en"${sea#ent}" > /dev/null 2>&1
      ifconfig en"${sea#ent}" down detach
   fi

   # Print values
   if [[ ${CSVOUT} = "yes" ]]; then
      if [ "${switchinfo}" = "NONE" ]; then
         printf '%s\n' "${hostname},Switch not Cisco, CDP not enabled, or CDP packet not
seen in timeout duration."
      else
         printf '%s\n' "hostname,sea,real_adapter,switchname,switchport"
         count=0
         while [[ ${count} < ${#switchinfo[*]} ]]; do
            switchname=$(echo "${switchinfo[$count]}" | awk -F ',' '{gsub("\047", "", $1);
print $1}')    # \047 is octal for a single quote
            switchport=$(echo "${switchinfo[$count]}" | awk -F ','
'{gsub("\047", "", $2); print $2}')    # \047 is octal for a single quote
            printf "%s\n" "${hostname},${sea},${real_adapter},${switchname},$
```

```
{switchport}"
            (( count+=1 ))
          done
       fi
    else
       if [ "${switchinfo}" = "NONE" ]; then
          printf '\n%s\n' "+---------------------------------+"
          printf '%s\n' "SEA: ${sea}\n"
          printf '%s\n' "    -> Backing adapter: ${real_adapter}"
          printf '%s\n' "+---------------------------------+"
          printf '%s\n' "Unable to determine switch details."
          printf '%s\n' "  Switch not Cisco, CDP not enabled, or CDP packet not seen in
timeout duration."
       else
          count=0
          printf '\n%s\n' "+---------------------------------+"
          printf '%s\n' "SEA: ${sea}"
          printf '%s\n' "    -> Backing adapter: ${real_adapter}"
          printf '%s\n' "+---------------------------------+"
          printf '%-40s %s\n' "Switch" "Port"
          printf '%-40s %s\n' "------" "----"
          while [[ ${count} < ${#switchinfo[*]} ]]; do
             switchname=$(echo "${switchinfo[$count]}" | awk -F ',' '{gsub("\047", "", $1);
print $1}')    # \047 is octal for a single quote
             switchport=$(echo "${switchinfo[$count]}" | awk -F ','
'{gsub("\047", "", $2); print $2}')    # \047 is octal for a single quote
             printf "%-40s %s\n" "${switchname}" "${switchport}"
             (( count+=1 ))
          done
       fi
    fi
done
```