



118

AIX

August 2005

In this issue

- [3 AIX updates command cookbook](#)
 - [7 Migrating print queues between AIX systems](#)
 - [9 AIX Version 4.3.3 to 5.x migration \(with NIM\)](#)
 - [26 Tape management system – part 3](#)
 - [49 AIX news](#)
-

© Xephon Inc 2005

update

AIX Update

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690

Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Colin Smith
E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *AIX Update*, comprising twelve monthly issues, costs \$275.00 in the USA and Canada; £180.00 in the UK; £186.00 in Europe; £192.00 in Australasia and Japan; and £190.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the November 2000 issue, are available separately to subscribers for \$24.00 (£16.00) each including postage.

***AIX Update* on-line**

Code from *AIX Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at <http://www.xephon.com/aix>; you will need to supply a word from the printed issue.

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, scripts, and other contents of this journal before making any use of it.

Contributions

When Xephon is given copyright, articles published in *AIX Update* are paid for at the rate of \$160 (£100 outside North America) per 1000 words and \$80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of \$32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

© Xephon Inc 2005. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher.

Printed in England.

AIX updates command cookbook

My previous article (see *Proactive AIX update tools and techniques*, *AIX Update* issue 117, July 2005) described AIX update management facilities provided by IBM. This article will complement the theory by providing a list of common update scenarios and their solutions.

How do I find the maintenance level of my system?

```
# oslevel -r
5200-04
```

How do I find a list of all known maintenance levels ever installed on my system?

```
#oslevel -rq
Known Recommended Maintenance Levels
-----
5200-05
5200-04
5200-03
5200-02
5200-01
5200-00
```

How do I find the list of filesets that are above the maintenance level of my system?

```
#oslevel -rg 5200-05
Fileset                                Actual Level          Recommended ML
-----
bos.adt.include                        5.2.0.53              5.2.0.52
bos.mp64                               5.2.0.54              5.2.0.53
bos.mp                                  5.2.0.54              5.2.0.53
bos.up                                  5.2.0.54              5.2.0.53
```

How do I find whether all filesets of a particular ML have been installed?

```
# instfix -ilgrep ML
All filesets for 5.2.0.0_AIX_ML were found.
All filesets for 5200-01_AIX_ML were found.
All filesets for 5200-02_AIX_ML were found.
All filesets for 5200-03_AIX_ML were found.
All filesets for 5200-04_AIX_ML were found.
Not all filesets for 5200-05_AIX_ML were found.
```

How do I find which additional filesets have to be installed to bring my system to the required maintenance level?

```
# instfix -ivk 5200-05_AIX_ML |grep not | grep ':'
```

You can also determine which fileset updates are missing by issuing the command:

```
# oslevel -r1 5200-05
```

Please note that the `_AIX_ML` suffix has been omitted.

How do I find which filesets have to be upgraded to bring my system to the required maintenance level?

```
# instfix -ciqk 5200-05_AIX_ML | grep "::-"
```

How do I check to see whether a certain PTF was applied?

We have to find out the APAR number within the PTF first. Then use this command:

```
# instfix -ik IY39588
    Not all filesets for IY39588 were found.
```

where APAR IY39588 is fixed in the PTF we want to find.

How do I verify that the filesets have the required prerequisites and are completely installed?

Check the output of the following command:

```
# lppchk -v
```

How do I find which fileset(s) are required to apply certain PTFs?

```
# instfix -ivk IY39588
IY39588 Abstract: zh/Zh/ZH_CN:Cannot Change IM status from program

    Fileset bos.loc.com.CN is not applied on the system.
    Fileset bos.loc.iso.zh_CN is not applied on the system.
    Not all filesets for IY39588 were found.
```

How do I list all filesets with their V.R.M.F (Version.Release Maintenance.Fix) level?

```
# ls1pp -L
```

How do I make sure that two systems are installed with exactly the same versions of all filesets?

Use the `compare_report` command in the following fashion. Generate the list of installed filesets on your base system by executing the following command:

```
# ls1pp -Lc>base.1pp.rep
```

Generate a similar report on the system to be updated:

```
# ls1pp -Lc>tobeupdated.1pp.rep
```

Bring the files to a common system and execute the following command:

```
# compare_report -b /tmp/base.ls1pp.rpt -o /tmp/tobeupdated.1pp.rep -l  
-h -m -n
```

This command will create reports listing the following:

- Filesets on the base system that are at a lower level than on the other system.
- Filesets on the base system that are at a higher level than the other system.
- Filesets installed on the base system that are not installed on the other system.
- Filesets installed on the other system that are not installed on the base system.

If all reports (-l, -h, -m, and -n) are requested for this type of comparison, the following reports will be generated:

- `baselower.rpt` (generated with -l)
- `basehigher.rpt` (generated with -h)
- `baseonly.rpt` (generated with -m)
- `otheronly.rpt` (generated with -n).

How do I change the global configuration settings of SUMA, which do not appear on SMIT screens?

```
# suma -c HTTP_PROXY=myproxy:8080 DL_TIMEOUT_SEC=1000
```

How do I view all scheduled SUMA tasks?

```
# crontab -l | grep suma
```

How do I create and schedule a SUMA task that will download the latest critical fixes monthly (eg on the 15th of every month at 2:30am)?

```
# suma -s "30 2 15 * *" -a RqType=Critical -a DisplayName="Monthly Crit  
Fixes Download"
```

How do I create and schedule a task that will check for a specific APAR once a week (for example every Thursday at 2:00am), download it when it becomes available, send an e-mail notification to a user on a remote system, and then delete the task?

```
# suma -s "0 2 * * 4" -a RqType=APAR -a RqName=IY12676 -a  
NotifyEmail="user@host" -a Repeats=y
```

How do I create and schedule a permanent task that will check for the latest level of the x1C.rte fileset monthly (for example on the 10th of every month at 2:30am)?

```
# suma -s "30 2 10 * *" -a RqType=Fileset -a RqName=x1C.rte -a  
RqLevel=latest -a Repeats=y
```

How do I create and immediately execute a task that will download all fixes needed to bring my system to maintenance level 5200-05?

```
# suma -x -a Action=Download -a RqType=ML -a Rqname=5200-05
```

How do I create and immediately execute a task that will list all fixes released after the latest maintenance package, 5200-04?

```
# suma -x -a Action=Preview -a RqType=Latest -a Filter=5200-04 -a  
FilterSysFile=/dev/null
```

*Alex Polak
System Engineer
APS (Israel)*

© Xephon 2005

Migrating print queues between AIX systems

In this article, I will provide a procedure to migrate all your print queues from one system to another. This is a very useful procedure that I have used many times in the past, which helped reduce the time significantly when upgrading systems from one server to another.

The procedure is very simple and straightforward and requires no interruption to the print operations or to the users on the source system. For the target system, the print operations will be interrupted during the procedure. Note that this procedure works only on network and remote queues. Other local print queues that have local printer devices on a local serial or parallel port need the local devices to be created on the destination system before you can go ahead with this procedure.

- 1 Copy the */etc/qconfig* file from the source system to the destination system. (Make a copy of the original file on the destination system before you copy.)

- 2 Copy the following files:

```
/var/spool/lpd/pio/@local/custom/*  
/var/spool/lpd/pio/@local/dev/*  
/var/spool/lpd/pio/@local/ddi/*
```

- 3 When you copy the files, try to preserve the permissions and ownership. If permissions or ownership are not preserved, then change the permissions on the copied files to **664**, and the ownership to **root:printq**:

```
chmod 664 filenames  
chown root:printq filenames
```

At this point, if you have local print queues, you need to verify that the local devices on the local ports do exist on the destination system with the same device name as on the source system. You need to create these local devices on the destination system with the same device names as

on the source system before you continue with the procedure. Otherwise, you will receive errors in the next steps and those failed print queues will not be usable.

4 Re-digest the print queues:

```
enq -d
```

5 Parse the attribute value of the virtual printer by running the **chvirprt** command on each print queue:

```
chvirprt -q <queue name> -d <device name>
```

If you have a lot of print queues, the best thing is to create a short script to do this for you. Below is an example shell script that gets the queue name and device name by running **lsallq -c**:

```
#!/bin/ksh
#####
# Script :   chvirprt_all
#
# Description: This script will list all print queues and their device
#              names and will run chvirprt command to parse
#              the attribute values of all print queues.
#####
PROG= $Ø
print "\nStart of ${PROG} program"
for STR in $(lsallq -c | grep "\:")
do
    QU_NAME=$(echo ${STR} | cut -f 1 -d':')
    DV_NAME=$(echo ${STR} | cut -f 2 -d':')
    print "\nParsing virtual printer attributes for
${QU_NAME}:${DV_NAME}"
    chvirprt -q ${QU_NAME} -d ${DV_NAME}
done
print "\nEnd of ${PROG} program"
exit Ø
```

6 Re-cycle the print spooling subsystem for the changes to take effect:

```
stopsrc -cg spooler
startsrc -g spooler
```

Earlier in this article, we mentioned that the local devices had to be created for the local print queues before you use the print

queue migration procedure. In a future article in *AIX Update*, I will provide an undocumented mechanism or procedure to move local printers and TTY devices from one system to another. I did not want to include it with this procedure in order not to complicate it, and due to the fact that local print queues are not widely used these days because more and more work is performed by remote and network printers.

Basim Chafik

Senior Systems Analyst

IBM Certified Advanced Technical Expert (CATE)

Plexus (Division of BancTec) (Canada)

© Xephon 2005

AIX Version 4.3.3 to 5.x migration (with NIM)

I would like to share my experience about how to migrate AIX 4.3.3 servers to 5.x versions. You will find tips, information about preparation, and possible errors that you might face. The sequence is important for the upgrade.

The logic from the alternate disk install is easy to understand. After preparation for the new version, you break one of the mirrors of rootvg, and install the 5.x operating system on this disk using the NIM server. The good news is you can do this online (without any downtime). The only downtime needed is to boot the machine from the disk you have just installed. The machine comes up with the new version (5.x), and if things go wrong you still have your old version (4.3.3) there to boot from. In my opinion, after a one-week test, the old version can be deleted.

- 1 First of all copy a **mksysb** image to a tape from the server (just in case!):

```
# mksysb -i /dev/rmt0
```

- 2 Firmware update:

- Check whether you have the right version – 4.3.3 (ML 11):

```
# oslevel -r
```

- Boot the server to check whether it boots cleanly. If it doesn't come up cleanly solve that problem before you begin the upgrade. Otherwise, you might think the problem has occurred because of the upgrade.

```
# shutdown -Fr
```

- Stop all applications and databases that start automatically while booting.
- Check the firmware levels:

```
# lscfg -vp | grep alterable
ROM Level (alterable).....L02113          <== System FW Level
ROM Level (non-alterable)...ag010611
ROM Level (alterable).....ag010611
                                     <== Service Processor FW level
```

- Check the model from the server:

```
# lsattr -El sys0 | grep modelname | awk '{print $2}'
```

Let's say our server is an H70 server and the latest firmware file is 7026H70F.BIN.

Download the latest firmware file from the IBM Internet site.

- FTP to the target system.

Then:

```
# mkdir /tmp/fwupdate
# cd /tmp/fwupdate
# chmod +x 7026H70F.BIN
# ./7026H70F.BIN
```

These files will be added to */tmp/fwupdate*:

```
st03115.img
sst03115.img
ss020419.img
ReadMe.TXT
```

Check whether the file is OK:

```
# sum st03115.img
```

This command will produce the following output:

```
23789 1486 st03115.img
*** All programs and databases must be down
# cd /usr/lpp/diagnostics/bin
# ./update_flash -f /tmp/fwupdate/st03115.img
```

Here comes the auto reboot.

Check the current FW level:

```
# lscfg -vp |grep alternable
```

Do not power off the target system at any time before the update process completes.

3 Update the SSA device driver and adapter microcode:

- There is one package for all types of SSA adapters at V4.3.3 on the IBM site – ssacode433.tar.
- A single package for enclosures: for example, for enclosures 7133 D40/T40 see 7133_X40_enc.tar.
- There are microcode packages for various disk types:
 - S53B, S53C, S53D, T53B, T53C, T53D
 - ST37, ST33, ST31, 73LP, UCD2
 - UCDY/UCPR/VCDY
 - DFHC,DCHC,DGHC,DRHC,DRVC
 - DMVC.

(There is no need to upgrade drive code for DMVC drives.)

The files are:

- S53X_T53X_disk.tar
- STXX_73LP_UCD2_disk.tar

- UCDY_UCPR_VCDY_disk.tar
- DFHC_DCHC_DGHC_DRHC_DRVC_disk.tar.
- Stop all applications and databases.
- **Umount** all filesystems and break all connections to the disks.

To **umount** as group, do the following for all volume groups except rootvg.

Check which VGs we have:

```
# lspv
```

Do the following for all volume groups other than rootvg:

```
# lsvgfs <vg-name> | while read a
do
chfs -u <vg-name> $a
done
```

```
# umount -t <vg-name>
```

Check whether all LV STATEs are closed (raw devices become closed when the application is down):

```
# lsvg -l <vg-name>
```

```
# varyoffvg <vg-name>
```

- Copy all tar files under */tmp/ssacode* on the server.
- Check that your enclosures are connected to the server:

```
# lsdev -C | grep enclosure
```

If it is an old enclosure, there is no output. Then:

```
# rm /tmp/ssacode/7133_X40_enc.tar (remove it )
```

If you get an output as enclosure0,1, it is a 7133-T40/D40 type enclosure.

Check the level:

```
# lscfg -vpl enclosureX | grep ROS
ROS Level and ID .....0020 (check if it is the newest)
```

If it is the newest, remove the 7133_X40_enc.tar file from the */tmp/ssacode* directory.

- Check how many SSA adapters you have:

```
# lsdev -C | grep ssa (ex: ssa0, ssa1, etc ...)
# lscfg -v1 ssaX (see the current level)
```

- Extract all packages under */tmp/ssacode*:

```
tar -xvf ssacode433.tar
tar -xvf S53X_T53X_disk.tar
tar -xvf STXX_73LP_UCD2_disk.tar
tar -xvf UCDY_UCPR_VCDY_disk.tar
tar -xvf DFHC_DCHC_DGHC_DRHC_DRVC_disk.tar
tar -xvf 7133_X40_enc.tar
```

- Run Configuration Manager to see whether you have clean output before you start:

```
# cfgmgr
```

- Start the installation:

```
# cd /tmp/ssacode
# inutoc
```

Run **smitty install**.

Select **install & update software**.

Select **install & update from ALL available software**.

Use the directory that you saved and unpacked the ssacode433.tar file into as the install device:

```
Input Device [.]
Software to Install [ALL]
First "Review only [YES]" if OK "Review only [NO]"
```

Check the log from smitty to see whether the installation was successful.

Exit smitty

- Run **cfgmgr** to install the microcode to the correct types of adapter.

(Note that some care may be needed when doing this in an active HA environment, particularly if you are not going to reboot immediately.)

- To complete the device driver upgrade you must now reboot your system.
- Confirm that the upgrade was a success.

From the previous steps you know how many SSA adapters you have:

```
# lscfg -v1 ssaX (where X is 0,1... for all SSA adapters)
```

Check that the microcode levels shown in ROS Level and ID match the level shown in the table at the IBM site for your adapter card.

Check that the SSA fileset levels you have match the levels shown at the IBM site.

```
# ls1pp -llgrep SSA
```

If any of the SSA filesets is at a lower level than those shown at the IBM site, repeat the whole upgrade procedure. If, after repeating the procedure, the code levels do not match the latest ones, place a call to your local IBM Service Center.

If the adapters are in SSA loops containing other adapters in other systems, repeat the steps of this procedure on all systems as soon as possible.

- Stop all applications and databases that start automatically after a boot.
- **Umount** all filesystems and break all connections to the disks.

To **umount** as group do the following for all volume groups except rootvg.

Check which VGs we have:

```
# lspv
```

Do the following for all volume groups other than rootvg:

```
# lsvgfs <vg-name> | while read a
do
chfs -u <vg-name> $a
done
```

```
# umount -t <vg-name>
```

Check whether all LV STATES are closed (raw devices become closed when the application is down):

```
# lsvg -l <vg-name>
```

```
# varyoffvg <vg-name>
```

- Check the enclosures:

```
lsdev -C | grep enclosure
```

If it is an old enclosure there will be no output and nothing more to do.

If we get an output as enclosure0,1, then it is a 7133-T40/D40 type enclosure.

Check the level:

```
# lscfg -vpl enclosureX | grep ROS
ROS Level and ID .....0020
```

If it is the latest, there is nothing to do.

If not, type the following commands:

```
# cd /etc/microcode
# ssa_sesdld -u -f cora1XXX.hex
```

(where XXX is the highest number that you have in /etc/microcode).

- To install the latest level of microcode to all SSA drives, run **ssadload -u** from each system in turn. (Do it in any case.)

```
# ssadload -u
```

Allow **ssadload** to complete on one system before running it on another.

Boot the server.

4 Alternate disk installation:

- Check whether your kernel works in 32-bit or 64-bit mode and check whether your database version suits the new 5.x version:

```
# bootinfo -K          (32- or 64-bit)
```

For Oracle:

```
# su - oracle
# svrmgr1 (see the Oracle version)
```

- Check the name convention for your logical volumes in rootvg. If you have logical volume names longer than 11 characters, you will need to rename them:

```
# umount <filesystem>
# chlv -n <new LV-Name> <old LV-Name>
# mount <filesystem>
```

- Ensure that you have more than 500MB of free space on each internal disk owned by rootvg. Otherwise, you will receive an error saying that you don't have enough space for the new version.

To create some space you may commit installed software that is in the APPLIED state (if you have any):

```
# installp -s ALL (shows softwares in APPLIED state)
# installp -c ALL (commits all)
```

- Upgrade your x1C software to the newest version (> 5.0.2.0):

```
# ls1pp -l | grep x1C
```

```
x1C.aix43.rte      5.0.2.0  COMMITTED  C Set ++ Runtime for AIX 4.3
x1C.cpp           4.3.0.1  COMMITTED  C for AIX Preprocessor
x1C.msg.en_US.cpp 4.3.0.1  COMMITTED  C for AIX Preprocessor
x1C.msg.en_US.rte 4.0.2.0  COMMITTED  C Set ++ Runtime
x1C.rte           5.0.2.0  COMMITTED  C Set ++ Runtime
```

If it is not the same as the above list, install 20030822_iy17981_xlc_apar.

- Break the mirror from rootvg.

Let's assume we have two disks, hdisk0 and hdisk1, owned by rootvg, and rootvg is mirrored. First of all, the disks must be identical:

```
# lsvg -p rootvg (the number of total PPs must be the same)
```

Probably the dump device will be different; then:

```
(# mklvcopy <dump-device> 2 hdisk1)
```

```
# unmirrorvg rootvg hdisk1 (separate hdisk0 and hdisk1)
# lsvg -p rootvg (see that hdisk1 is not a mirror anymore)
```

Change the bootlist:

```
# bootlist -m normal hdisk0
# bootlist -m normal -o
# bosboot -a -d /dev/hdisk0
```

Take hdisk1 out of rootvg:

```
# reducevg rootvg hdisk1
```

Check whether it is OK with:

```
# lspv
```

- Install a NIM client on the system.

You can check whether it is already installed with:

```
# ls -al /etc/niminfo
```

(If the file exists, it has been installed before.)

If it is already installed:

```
# cat /etc/niminfo
#----- Network Install Manager -----
# warning - this file contains NIM configuration information
#          and should only be updated by NIM
# Filenet (nimclient1); IBM RS/6000 H70 - Rack oben

export NIM_NAME=nimclient1
export NIM_HOSTNAME=nimclient1
export NIM_CONFIGURATION=standalone
export NIM_MASTER_HOSTNAME=service
export NIM_MASTER_PORT=1058
```

```

export NIM_REGISTRATION_PORT=1059
export NIM_BOS_IMAGE=/SPOT/usr/sys/inst.images/bos
export NIM_BOS_FORMAT=rte
export NIM_HOSTS=" 10.13.2.12:nimclient1 10.64.222.55:service"
export NIM_MOUNTS=""
export ROUTES=" default:0:10.13.0.1 "

```

If it was not previously installed, before we can install it we need to know the network adapter name, IP address, subnetmask, and default gateway.

Let's say our NIM server's hostname is 'service':

```

# traceroute service
trying to get source for service
source should be 10.13.2.12
traceroute to service (10.64.250.50) from 10.13.2.12 (10.13.2.12), 30
hops max
outgoing MTU = 1492
 1 10.13.0.100 (10.13.0.100)  5 ms  3 ms  3 ms
 2 10.250.13.21 (10.250.13.21) 13 ms 12 ms 12 ms
 3 10.250.64.201 (10.250.64.201) 14 ms 14 ms 13 ms
 4 service (10.64.250.50) 13 ms 12 ms 13 ms

```

```
# netstat -in
```

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
en0	1500	link#2	0.4.ac.31.8b.48	580514355	0	545732153	0	0
en0	1500	10.13.16	10.13.18.12	580514355	0	545732153	0	0
tr0	1492	link#3	0.6.29.b9.30.b0	77165779	0	294151795	0	0
tr0	1492	10.13	10.13.2.12	77165779	0	294151795	0	0
lo0	16896	link#1		5703295	0	5704747	0	0
lo0	16896	127	127.0.0.1	5703295	0	5704747	0	0
lo0	16896	::1		5703295	0	5704747	0	0

So, our network adapter is tr0 (Token Ring) and IP=10.13.2.12:

```
# netstat -r
```

```
Routing tables
```

Destination	Gateway	Flags	Refs	Use	If	PMTU	Exp	Groups
-------------	---------	-------	------	-----	----	------	-----	--------

```
Route Tree for Protocol Family 2 (Internet):
```

default	10.13.0.1	UGc	0	0	tr0	-	-
10.13/20	imswest	U	32	6685041	tr0	-	-
10.13.16/20	imswest	U	3	545442069	en0	-	-
whurw011.huerth.	10.13.0.1	UGHW	1	4639	tr0	1492	-
whurw013.huerth.	10.13.0.1	UGHW	1	25814	tr0	1492	-
whurw042.huerth.	10.13.0.1	UGHW	1	30071	tr0	1492	-
whurw056.huerth.	10.13.0.1	UGHW	1	94	tr0	1492	-

```

whurw082.huerth. 10.13.0.1      UGHW      1      6920    tr0    1492    -
whuiv010.huerth. 10.13.0.1      UGHW      2      2437    tr0    1492    -
whuco006.huerth. 10.13.0.1      UGHW      1      2453    tr0    1492    -
rsw004e0.koeln.a 10.13.0.1      UGHW      1      2204    tr0    1492    -
imszakoeIn      10.13.0.1      UGHW      2      73551   tr0    1492    -
service         10.13.0.1      UGHW      2      2396653 tr0    -        -
s3koele0.eil.ris 10.13.0.1      UGHW      2      23428   tr0    1492    -
127/8          loopback       U          9      2605006 1o0    -        -

```

The default gateway from our Server is 10.13.0.1.

Run **smitty tcpip**.

Select **Further Configuration**.

Select **Network Interfaces**.

Select **Network Interface Selection**.

Select **tr0 Token Ring Network Interface**.

Change / Show a Token-Ring Network Interface

Type or select values in entry fields.

Press Enter AFTER making all desired changes.

```

[Entry Fields]
Network Interface Name      tr0
INTERNET ADDRESS (dotted decimal) [10.13.2.12]
Network MASK (hexadecimal or dotted decimal) [255.255.240.0]
Current STATE              up
Use Address Resolution Protocol (ARP)?      yes
Enable Hardware LOOPBACK Mode?            no
BROADCAST ADDRESS (dotted decimal)        []
Confine BROADCAST to LOCAL Token-Ring?     no
Subnetmask is 255.255.240.0

```

- Do the following on the service (NIM server):

```

# lsnim | grep 10_13
tok_10_13          networks      tok
ent-10_13_100_0   networks      ent
ent_10_13_16      networks      ent

```

```
# lsnim -l tok_10_13
```

```

tok_10_13:
  class      = networks
  type       = tok
  Nstate     = ready for use
  prev_state = ready for use

```

```

net_addr   = 10.13.0.0
snm        = 255.255.240.0
routing1   = default 10.13.0.1

```

Don't forget to write NIM server in */etc/hosts* of the NIM client, and NIM client in */etc/hosts* of the NIM server.

Run **smitty nim**.

Select **Perform NIM Administration Tasks**.

Select **Manage Networks**.

Select **Define a Network**.

Select **tok = token ring network**.

Change/Show Characteristics of a Network

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]
* Network Name	[tok_10_13]
Network Type	tok
Network Install State	ready for use
Network IP Address	[10.13.0.0]
Subnetmask	[255.255.240.0]
Other Network Type	
Comments	[]
Force	no

The network IP address would be 10.13.0.0 and this is the first IP address in this subnet. You can use an IP subnet calculator (search the Internet) to find this IP address.

- Find the ring speed value for the Token Ring on the NIM client:

```

# netstat -v
-----
ETHERNET STATISTICS (ent0) :
.....
.....

TOKEN-RING STATISTICS (tok0) :
Device Type: IBM PCI Tokenring Adapter (14103e00)

```

Hardware Address: 00:06:29:b9:30:b0
Elapsed Time: 171 days 17 hours 29 minutes 3 seconds

Transmit Statistics:

Packets: 294186295

Bytes: 105837728863
Interrupts: 246087251
Transmit Errors: 2
Packets Dropped: 0

Receive Statistics:

Packets: 96494038

Bytes: 19638687842
Interrupts: 96492991
Receive Errors: 0
Packets Dropped: 0
Bad Packets: 0

Max Packets on S/W Transmit Queue: 60

S/W Transmit Queue Overflow: 0

Current S/W+H/W Transmit Queue Length: 0

Elapsed Time: 171 days 17 hours 28 minutes 57 seconds

Broadcast Packets: 14390

Multicast Packets: 2

Timeout Errors: 0

Current SW Transmit Queue Length: 0

Current HW Transmit Queue Length: 0

Broadcast Packets: 64851620

Multicast Packets: 9

Receive Congestion Errors: 0

General Statistics:

No mbuf Errors: 0

Abort Errors: 47

Burst Errors: 2

Frequency Errors: 0

Internal Errors: 0

Lost Frame Errors: 0

Token Errors: 0

Ring Recovered: 14

Soft Errors: 48

Driver Flags: Up Broadcast Running

AlternateAddress 64BitSupport ReceiveFunctionalAddr

16 Mbps

Lobe Wire Faults: 0

AC Errors: 0

Frame Copy Errors: 2

Hard Errors: 0

Line Errors: 0

Only Station: 0

Remove Received: 0

Signal Loss Errors: 0

Transmit Beacon Errors: 0

IBM PCI Tokenring Adapter (14103e00) Specific Statistics:

Media Speed Running: 16 Mbps Half Duplex

Media Speed Selected: 16 Mbps Full Duplex

Receive Overruns : 0

Transmit Underruns : 0

ARI/FCI errors : 0

Microcode level on the adapter :001PX11B2

Num pkts in priority sw tx queue : 0

Num pkts in priority hw tx queue : 0

Open Firmware Level : 001PXR01

So our Speed is 16 Mbps Full Duplex

- Do the following on the service (NIM server).

Run **smitty nim**.

Select **Perform NIM Administration Tasks**.

Select **Manage Machines**.

Select **Define a Machine**.

Change/Show Characteristics of a Machine

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]
Machine Name	[nimclient1]
* Hardware Platform Type	[chrp]
* Kernel to use for Network Boot	[up]
Machine Type	standalone
Network Install Machine State	currently running
Network Install Control State	ready for a NIM
operation	
Primary Network Install Interface	
Network Name	tok_10_13
Host Name	[nimclient1]
Network Adapter Hardware Address	[0]
Network Adapter Logical Device Name	[tok]
Ring Speed	[16]
IPL ROM Emulation Device	[]
CPU Id	[004335BA4C00]
Comments	[]
Force	no

– Installing NIM client software:

```
# ls|pp -llgrep nim
```

```
X11.Dt.helpmin          4.3.3.0 COMMITTED  AIX CDE Minimum Help Files
X11.msg.en_US.Dt.helpmin 4.3.0.0 COMMITTED  AIX CDE Minimum Help Files -
bos.sysmgmt.nim.client   4.3.3.78 COMMITTED  Network Install Manager -
X11.Dt.helpmin          4.3.3.0 COMMITTED  AIX CDE Minimum Help Files
bos.sysmgmt.nim.client   4.3.3.75 COMMITTED  Network Install Manager -
```

If it does not look like the above list, install `lppsourcemi11 NetworkInstallManager-Client Tools – 4.3.3.75` and `4.3.3.78`.

- Do the following on the NIM client.

Run **smitty nim**.

Select Configure Network Installation Management Client Fileset.

Configure Network Installation Management Client Fileset

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]
* Machine Name	[nimclient1]
* Primary Network Install Interface	[tr0]
* Host Name of Network Install Master	[service]
Hardware Platform Type	[chrp]
Kernel to use for Network Boot	[up]
IPL ROM Emulation Device	[]
Comments	[]
Alternate Port Numbers for Network Communications (reserved values will be used if left blank)	
Client Registration	[]
Client Communications	[]

- You must disable TCB on the NIM client (nimclient1).

Check:

```
# odmget -q attribute=TCB_STATE PdAt
```

If tcb_enabled:

```
# odmget -q attribute=TCB_STATE PdAt > own_filename
```

Change tcb_enabled to tcb_disabled and save file:

```
# odmdelete -o PdAt -q attribute=TCB_STATE  
# odmadd own_filename
```

- Start the alternate disk installation on the NIM server.

For AIX 5.1:

```
# nimadm -c <CLIENT> -s <SPOT> -l <LPPSOURCE> -d hdiskX -Y
```

Example:

```
# nohup nimadm -c nimclient1 -s spot_aix510_LCD4-1061-03_m105 -l  
lppsrc_510_LCD4-1061-03_m105 -d hdisk1 -Y | tee /tmp/  
nimadm_nimclient1.log &
```

For AIX 5.2:

```
# nimadm -c <CLIENT> -s <SPOT> -l <LPPSOURCE> -d hdiskX -Y
```

Example:

```
# nohup nimadm -c nimclient1 -s spot_520_lppsrc_lcd4_1133_02_5200_02 -l  
lppsrc_520_lcd4_1133_02_5200_02 -d hdisk1 -Y |tee /tmp/  
nimadm_nimclient1.log &
```

After it finishes (it takes a long time), the next step is to check the entire log file, searching for errors and warnings. If it's clean at the end you see 'Phase 12' and then it's over.

```
# tail -f /tmp/nimadm_nimclient1.log
```

```
Changing logical volume names in volume group descriptor area.
```

```
Fixing LV control blocks...
```

```
Fixing file system superblocks...
```

```
Bootlist is set to the boot disk: hdisk1
```

```
+-----+  
Executing nimadm phase 12.
```

```
+-----+  
Cleaning up alt_disk_migration on the NIM master.  
Cleaning up alt_disk_migration on client imswest.
```

- Check whether the bootlist is correct:

```
# bootlist -m normal -o  
hdisk1
```

- Finally, you can boot the machine and it will start from the 5.x version this time:

```
# shutdown -Fr
```

and then check your OS version and maintenance level with:

```
# oslevel -r
```

5 'To dos' after the alternate disk install:

I advise you to test your applications for at least a week with the new version of the operating system, and if everything goes well you can destroy the 4.3.3 version on hdisk0.

- After your test week do the following:

```
# exportvg old_rootvg

# lspv

hdisk0          004335ba56d3bb26          None
hdisk1          004335badb11c3d5          rootvg
hdisk2          004335badb11c673          fnvg
hdisk3          004335badc26e662          fnvg
hdisk4          004335badc26ebe5          fnvg
hdisk5          004335badc26f037          fnvg

# extendvg rootvg hdisk0

0516-014 linstallpv: The physical volume appears to belong to
  another volume group.
004335ba85790ed7
0516-631 extendvg: Warning, all data belonging to physical
  volume hdisk0 will be destroyed.
extendvg: Do you wish to continue? y(es) n(o)? y
```

- Construct the mirror again:

```
# mirrorvg rootvg hdisk0

0516-1124 mirrorvg: Quorum requirement turned off, reboot
  system for this to take effect for rootvg.
0516-1126 mirrorvg: rootvg successfully mirrored, user should
  perform bosboot of system to initialize boot records.
  Then, user must modify bootlist to include: hdisk0 hdisk1.
```

- Create a logical volume for dump device – hdisk0 (eg hd70 with eight PPs):

```
# mklv -y hd70 -t sysdump rootvg 8 hdisk0
# sysdumpdev -P -s /dev/hd70

primary          /dev/hd7
secondary        /dev/hd70
copy directory   /var/adm/ras
forced copy flag TRUE
always allow dump FALSE
dump compression OFF
```

- Write a new boot device and determine the boot sequence:

```
# bosboot -a -d /dev/hdisk0
bosboot: Boot image is 16145 512 byte blocks.
```

```

# bootlist -m normal -o
hdisk1

# bootlist -m normal hdisk0 hdisk1
# bootlist -m normal -o
hdisk0
hdisk1

```

If you are taking your back-ups with Tivoli Storage Management, don't forget also to upgrade your TSM client.

Adnan Akbas
Senior System Administrator
TURKCELL (Germany)

© Xephon 2005

Tape management system – part 3

This month we conclude the code for a tape management system.

```

else
    loc="UNK"
fi
thisvolser=$(echo "$volser"|cut -c1-3)
if [ "$thisvolser" != "$lastvolser" ] ; then
echo "\nTotal records found for this tape type: $found\n" >> $tmp
((Gfound = Gfound + found))
found=0
fi
printf $format "$volser" "$creation_time" "$expiration_time"
"$host" "$loc" "$backup" "$passes" >> $tmp
((found = found + 1))
# ((Gfound = Gfound + 1))
lastvolser="$thisvolser"
echo ".\c"
fi
### progress meter
done
((Gfound = Gfound + found))
clear
echo "\nTotal records found for this tape type: $found" >> $tmp

```

```

echo "\n\nTotal records found: $Gfound\n" >>$tmp
cat $tmp
echo "\nWould you like a printed report "
echo "If Yes then enter the printer name"
echo "(or just press [ENTER] to NOT print the report)"
read q
if [ -n "$q" ] ; then
    $home/bin/printl.ksh $q $tmp
    sleep 3
fi
cp $tmp $home/reports/scratchreport.txt
rm $tmp
if [ "$menu" -eq "15" ] ; then
    rm $home/reports/scratchreport.txt 2>/dev/null
fi
fi
}
#####
all_tape_report() {
$label_log "All active tape report"
clear
now='$sec'
now_date='$s2d $now'
tmp="/tmp/report.$$"
>$tmp
echo "Include scratch tapes y/n ?"
read scryn
echo "All current tapes as of $now_date\n" >>$tmp
printf $format "Volser" "Creation time" "Expiration time" "Host" "LOC"
"Contents" " Usage" >> $tmp
printf $format "-----" "-----" "-----" "-----" "----" "----" "
-----" " ----" >> $tmp
for tape_rec in `cat $tapes|sort -n -k3,6` ; do
    volser='echo $tape_rec|cut -f1 -d:"'
    creation='echo $tape_rec|cut -f2 -d:"'
    lifetime='echo $tape_rec|cut -f3 -d:"'
    host='echo $tape_rec|cut -f4 -d:"'
    backup='echo $tape_rec|cut -f5 -d:"'
    location='echo $tape_rec|cut -f6 -d:"'
    creation_time='$s2d $creation'
    foo='echo "$creation+$lifetime"|bc'
    expiration_time='$s2d $foo'
    passes='grep $volser $tape_history|wc -l'
    if [ "$location" -eq "0" ] ; then
        loc="ON"
    elif [ "$location" -eq "1" ] ; then
        loc="OFF"
    else
        loc="UNK"
    fi
fi

```

```

    scrf=0
    if [ "$foo" -lt "$now" ] ; then
        volser1="{volser}${z1}"
        volser=$volser1
        scrf=1
    fi
    if [ "$scrf" -eq "1" ] ; then
        if [ "$scryn" = "n" ] ; then
            continue
        fi
    fi
    printf $format "$volser" "$creation_time" "$expiration_time" "$host"
"$loc" "$backup" "$passes" >> $tmp
done
    cat $tmp
    echo "\nWould you like a printed report "
    echo "If Yes then enter the printer name"
    echo "(or just press [ENTER] to NOT print the report)"
    read q
    if [ -n "$q" ] ; then
        $home/bin/printl.ksh $q $tmp
        sleep 3
    fi
    cp $tmp $home/reports/alltapereport.txt
    rm $tmp
}
backup_report() {
$label_log "Tapes by backup report"
clear
now='$sec'
now_date='$s2d $now'
tmp="/tmp/report.$$"
tmp99="/tmp/report2.$$"
>$tmp
> $tmp99
echo "Enter a backup name, or press [ENTER] for all"
read h
if [ -z "$h" ] ; then
    curhost="-vi none"
    curhost2="ALL"
else
    curhost=$h
    curhost2=$h
fi
echo "All backups by backup name: $curhost2 as of $now_date\n" >>$tmp
printf $format "Volser" "Creation time" "Expiration time" "Host" "LOC"
"Contents" " Usage" >> $tmp
printf $format "-----" "-----" "-----" "----" "----"
"-----" " ----" >> $tmp
for tape_rec in `cat $tapes|grep $curhost|sort -n -k3,6` ; do

```

```

volser='echo $tape_rec|cut -f1 -d":"'
creation='echo $tape_rec|cut -f2 -d":"'
lifetime='echo $tape_rec|cut -f3 -d":"'
host='echo $tape_rec|cut -f4 -d":"'
backup='echo $tape_rec|cut -f5 -d":"'
location='echo $tape_rec|cut -f6 -d":"'
creation_time='$s2d $creation'
foo='echo "$creation+$lifetime"|bc'
expiration_time='$s2d $foo'
passes='grep $volser $tape_history|wc -l'
if [ "$location" -eq "0" ] ; then
    loc="ON"
elif [ "$location" -eq "1" ] ; then
    loc="OFF"
else
    loc="UNK"
fi
if [ "$foo" -lt "$now" ] ; then
    volser1="{volser}${z1}"
    volser=$volser1
fi
printf $format "$volser" "$creation_time" "$expiration_time" "$host"
"$loc" "$backup" "$passes" >> $tmp99
done
sort -t " " -k 31.9 $tmp99 |sort -t " " -nk 31.16 >> $tmp
cat $tmp
echo "\nWould you like a printed report "
echo "If Yes then enter the printer name"
echo "(or just press [ENTER] to NOT print the report)"
read q
if [ -n "$q" ] ; then
    $home/bin/printl.ksh $q $tmp
    sleep 3
fi
cp $tmp $home/reports/backupreport.txt
rm $tmp
rm $tmp99
}
host_report() {
$label_log "Tapes by host report"
clear
now='$sec'
now_date='$s2d $now'
tmp="/tmp/report.$$"
>$tmp
echo "Enter a host name, or press [ENTER] for current host"
read h
if [ -z "$h" ] ; then
    curhost='hostname'
else

```

```

    curhost=$h
fi
echo "All backups for host: $curhost as of $now_date\n" >> $tmp
printf $format "Volser" "Creation time" "Expiration time" "Host" "LOC"
"Contents" " Usage" >> $tmp
printf $format "-----" "-----" "-----" "-----" "-----" "
-----" " ----" >> $tmp
for tape_rec in `cat $tapes|grep $curhost|sort -n -k3,6` ; do
    volser=`echo $tape_rec|cut -f1 -d":"`
    creation=`echo $tape_rec|cut -f2 -d":"`
    lifetime=`echo $tape_rec|cut -f3 -d":"`
    host=`echo $tape_rec|cut -f4 -d":"`
    backup=`echo $tape_rec|cut -f5 -d":"`
    location=`echo $tape_rec|cut -f6 -d":"`
    creation_time=`$s2d $creation`
    foo=`echo "$creation+$lifetime"|bc`
    expiration_time=`$s2d $foo`
    passes=`grep $volser $tape_history|wc -l`
    if [ "$location" -eq "0" ] ; then
        loc="ON"
    elif [ "$location" -eq "1" ] ; then
        loc="OFF"
    else
        loc="UNK"
    fi
    if [ "$foo" -lt "$now" ] ; then
        volser1="${volser}${z1}"
        volser=$volser1
    fi
    printf $format "$volser" "$creation_time" "$expiration_time" "$host"
"$loc" "$backup" "$passes" >> $tmp
done
cat $tmp
echo "\nWould you like a printed report "
echo "If Yes then enter the printer name"
echo "(or just press [ENTER] to NOT print the report)"
read q
if [ -n "$q" ] ; then
    $home/bin/printl.ksh $q $tmp
    sleep 3
fi
cp $tmp $home/reports/hostreport.txt
rm $tmp
}
history_report() {
$label_log "Detail report by tape"
clear
now=`$sec`
now_date=`$s2d $now`
tmp="/tmp/report.$$"

```

```

>$tmp
echo "All tape history to date as of $now_date\n" >>$tmp
printf $format "Volser" "Creation time" "Expiration time" "Host" "LOC"
"Contents" " Usage" >> $tmp
printf $format "-----" "-----" "-----" "----" "----" "
-----" " ----" >> $tmp
pvol=" "
tmphist="$home/tape_history.tmp"
tapelist="$home/tapelist.tmp"
cp $tape_history $tmphist
cat $tmphist|cut -f1 -d":"|sort|uniq|sort -n > $tapelist
jobspercpu="2"
CPUs='lsdev -Cc processor|wc -l|awk {'print $1'}'
jobsmax='echo "$jobspercpu*$CPUs" |bc'
Start='date'
echo "Thread value is: $jobsmax"
for volser in 'cat $tapelist' ; do
    echo "Starting job for volser $volser"
    $home/bin/proc_hist_report.ksh $volser &
    JOBS='jobs |wc -l|awk {'print $1'}'
    if [ "$JOBS" -ge "$jobsmax" ] ; then
        while : ; do
            JOBS='jobs |wc -l|awk {'print $1'}'
            if [ "$JOBS" -lt "$jobsmax" ] ; then
                break
            fi
            sleep 2
        done
    fi
done
# Just in case there are less than $jobsmax running, we'll
# see if anything is left running.
while : ; do
    JOBS='jobs |wc -l|awk {'print $1'}'
    if [ "$JOBS" -eq "0" ] ; then
        break
    fi
    sleep 5
    echo "Waiting for $JOBS jobs to finish"
done
End='date'
for volser in 'cat $tapelist' ; do
    cat $home/reports/tape_detail.$volser >> $tmp
    echo " " >> $tmp
    rm $home/reports/tape_detail.$volser
done
cat $tmp
echo "Job start: $Start"
echo "Job end: $End"
echo "\nWould you like a printed report "

```

```

echo "If Yes then enter the printer name"
echo "(or just press [ENTER] to NOT print the report)"
read q
if [ -n "$q" ] ; then
    $home/bin/printl.ksh $q $tmp
    sleep 3
fi
cp $tmp $home/reports/histreport.txt
rm $tmp
rm $home/tape_history.tmp
rm $home/tapelist.tmp
}
aging_report() {
$label_log "Tape aging report"
clear
now='$sec'
now_date='$s2d $now'
tmp="/tmp/report.$$"
>$tmp
echo "Tape aging report as of of $now_date\n" >>$tmp
format="%-13s%-29.26s%-29.26s%-5s\n"
printf $format "Volser" "First Use" "Last Use" "Usage" >> $tmp
printf $format "-----" "-----" "-----" "-----" >>
$tmp
pvol=" "
tmphist="$home/tape_history.tmp"
tapelist="$home/tapelist.tmp"
cp $tape_history $tmphist
cat $tmphist|cut -f1 -d":"|sort|uniq|sort -n > $tapelist
jobspercpu="3"
> $tmp.xx
for volser in `cat $tapelist` ; do
    first_use='grep $volser $tmphist|head -1'
    last_use='grep $volser $tmphist|tail -1'
    usecount='grep -c $volser $tmphist'
    first_epo='echo "$first_use"|cut -f2 -d":"'
    first_date='$s2d $first_epo'
    last_epo='echo "$last_use"|cut -f2 -d":"'
    last_date='$s2d $last_epo'
    printf $format "$volser" "$first_date" "$last_date" "$usecount" >>
$tmp.xx
done
cat $tmp.xx |sort -nr -k1.67 >> $tmp
cat $tmp
echo "\nWould you like a printed report "
echo "If Yes then enter the printer name"
echo "(or just press [ENTER] to NOT print the report)"
read q
if [ -n "$q" ] ; then
    $home/bin/printl.ksh $q $tmp

```



```

    sleep 3
fi
cp $tmp $home/reports/agereport.txt
rm $tmp
rm $tmp.xx
rm $home/tape_history.tmp
rm $home/tapelist.tmp
}
offsite_report() {
    $label_log "Offsite tape report"
    clear
    now='$sec'
    now_date='$s2d $now'
    tmp="/tmp/report.$$"
    >$tmp
    echo "All tape offsite as of $now_date\n" >>$tmp
    printf $format "Volser" "Creation time" "Expiration time" "Host"
"LOC" "Contents" " Usage" >> $tmp
    printf $format "-----" "-----" "-----" "----" "----"
"-----" " ----" >> $tmp
    pvol=" "
    echo "Processing..."
    for tape_rec in `cat $tapes|sort -n -k3,6` ; do
        volser='echo $tape_rec|cut -f1 -d":"'
        creation='echo $tape_rec|cut -f2 -d":"'
        lifetime='echo $tape_rec|cut -f3 -d":"'
        host='echo $tape_rec|cut -f4 -d":"'
        backup='echo $tape_rec|cut -f5 -d":"'
        location='echo $tape_rec|cut -f6 -d":"'
        creation_time='$s2d $creation'
        foo='echo "$creation+$lifetime"|bc'
        expiration_time='$s2d $foo'
        passes='grep $volser $tape_history|wc -l'
        if [ "$foo" -lt "$now" ] ; then
            volser1="{volser}${z1}"
            volser=$volser1
        fi
        if [ "$location" -eq "1" ] ; then
            loc="OFF"
            printf $format "$volser" "$creation_time" "$expiration_time"
"$host" "$loc" "$backup" "$passes" >> $tmp
        fi
    done
    cat $tmp
    echo "\nWould you like a printed report "
    echo "If Yes then enter the printer name"
    echo "(or just press [ENTER] to NOT print the report)"
    read q
    if [ -n "$q" ] ; then
        $home/bin/printl.ksh $q $tmp
    fi
}

```

```

        sleep 3
    fi
    cp $tmp $home/reports/offsitereport.txt
    rm $tmp
}
onsite_report() {
    $label_log "Onsite tape report"
    clear
    now='$sec'
    now_date='$s2d $now'
    tmp="/tmp/report.$$"
    >$tmp
    echo "All tape onsite as of $now_date\n" >>$tmp
    printf $format "Volser" "Creation time" "Expiration time" "Host"
"LOC" "Contents" " Usage" >> $tmp
    printf $format "-----" "-----" "-----" "----" "----"
"-----" " ----" >> $tmp
    pvol=" "
    echo "Processing..."
    for tape_rec in `cat $tapes|sort -n -k3,6` ; do
        volser='echo $tape_rec|cut -f1 -d":"'
        creation='echo $tape_rec|cut -f2 -d":"'
        lifetime='echo $tape_rec|cut -f3 -d":"'
        host='echo $tape_rec|cut -f4 -d":"'
        backup='echo $tape_rec|cut -f5 -d":"'
        location='echo $tape_rec|cut -f6 -d":"'
        creation_time='$s2d $creation'
        foo='echo "$creation+$lifetime"|bc'
        expiration_time='$s2d $foo'
        passes='grep $volser $tape_history|wc -l'
        if [ "$foo" -lt "$now" ] ; then
            volser1="{volser}{z1}"
            volser=$volser1
        fi
        if [ "$location" -eq "0" ] ; then
            loc="ON"
            printf $format "$volser" "$creation_time" "$expiration_time"
"$host" "$loc" "$backup" "$passes" >> $tmp
        fi
    done
    cat $tmp
    echo "\nWould you like a printed report "
    echo "If Yes then enter the printer name"
    echo "(or just press [ENTER] to NOT print the report)"
    read q
    if [ -n "$q" ] ; then
        $home/bin/printl.ksh $q $tmp
        sleep 3
    fi
    cp $tmp $home/reports/onsitereport.txt
}

```

```

    rm $tmp
}
make_tape_bad() {
# ### volser:creation:lifetime:host:backup
clear
while : ; do
    s9="9999999999"
    tmp="/tmp/report.$$"
    touch $tmp
    echo "\nMark a tape as bad. Enter the tape VOLSER to"
    echo "mark as bad/defective, or enter -1 to exit"
    echo "to main menu"
    read volser
    if [ "$volser" = "-1" ] ; then
        break
    fi
    echo "Tape $volser will be marked bad"
    tape_rec='grep $volser $tapes'
    wc1='grep $volser $tapes|wc -l'
    if [ "$wc1" -ne "1" ] ; then
        echo "\n\n ## Please specify only one volser ##"
        continue
    fi
    echo "Press Y or y to confirm marking $volser as bad"
    read Y
    y='echo $Y|tr [A-Z] [a-z]'
    if [ "$y" != "y" ] ; then
        echo "\n\n ## $volser not marked as bad ##"
        continue
    fi
    volser='echo $tape_rec|cut -f1 -d":"'
# ### creation='echo $tape_rec|cut -f2 -d":"'
# ### creation will be the date the tape was marked bad. BV 12/3/2003
    creation='$sec'
    lifetime='echo $tape_rec|cut -f3 -d":"'
    host='echo $tape_rec|cut -f4 -d":"'
    backup='echo $tape_rec|cut -f5 -d":"'
    location='echo $tape_rec|cut -f6 -d":"'
    lifetime="$s9"
    backup="Defective"
    host=$backup
    cat $tapes|grep -v $volser >>$tmp
    newrec="$volser:$creation:$lifetime:$host:$backup:$location"
    echo $newrec >> $tmp
    $home/bin/lockdb.ksh "mark tape bad"
    echo $newrec >> $tape_history
    cat $tmp|sort -n -k3,6 >$tapes
    rm $tmp
    $home/bin/unlockdb.ksh "mark tape bad"
    $label_log "Tape $volser marked bad"

```

```

done
    }
make_tape_good() {
# ### volser:creation:lifetime:host:backup
clear
while : ; do
    s9='$sec'
    tmp="/tmp/report.$$"
    touch $tmp
    echo "\nMark a tape as scratch. Enter the tape VOLSER to"
    echo "mark as scratch or enter -1 to exit"
    echo "to main menu"
    read volser
    if [ "$volser" = "-1" ] ; then
        break
    fi
    echo "Tape $volser will be marked scratch"
    tape_rec='grep $volser $tapes'
    wc1='grep $volser $tapes|wc -l'
    if [ "$wc1" -ne "1" ] ; then
        echo "\n\n ## Please specify only one volser ##"
        continue
    fi
    echo "Press Y or y to confirm marking $volser as scratch"
    read Y
    y='echo $Y|tr [A-Z] [a-z]'
    if [ "$y" != "y" ] ; then
        echo "\n\n ## $volser not marked as scratch ##"
        continue
    fi
    volser='echo $tape_rec|cut -f1 -d:"'
    creation='echo $tape_rec|cut -f2 -d:"'
    lifetime='echo $tape_rec|cut -f3 -d:"'
    host='echo $tape_rec|cut -f4 -d:"'
    backup='echo $tape_rec|cut -f5 -d:"'
    location='echo $tape_rec|cut -f6 -d:"'
    if [ "$backup" = "Defective" ] ; then
        if [ "$host" = "Defective" ] ; then
            lifetime="1"
            creation="$s9"
            backup="None"
            host=$backup
            cat $tapes|grep -v $volser >>$tmp
            newrec="$volser:$creation:$lifetime:$host:$backup:$location"
            $home/bin/lockdb.ksh "mark tape good"
            echo $newrec >> $tmp
            echo $newrec >> $tape_history
            cat $tmp|sort -n -k3,6 >$tapes
            $home/bin/unlockdb.ksh "mark tape good"
            $label_log "Tape $volser marked good"

```

```

        else
            echo "Tape not marked as bad in the database...\n"
        fi
    else
        echo "Tape not marked as bad in the database...\n"
    fi
    rm $tmp
done
    }
read_tape_label() {
# Let's find the right tape drive
ts='lsdev -Cc tape|grep -i available|awk {'print $1'}'
for t in $ts ; do
    lscfg -vl $t|grep -i ult 1>/dev/null 2>&1
    if [ "$?" -eq "0" ] ; then
        drive=/dev/$t
        break
    fi
    echo "LTO Tape Drive not found!  Exiting!"
    echo "Press [ENTER] to abend"
    read foo
    exit 1
done
chdev -l 'basename $drive' -a block_size=0 1>/dev/null 2>&1
clear
default="$drive"
echo "Read a tape label on a currently loaded tape\n"
echo "I'll use $default as my tape drive.\n"
# echo "If you want to change it, enter a new tape"
# echo "drive name here --> \c"
# read tape_drive
# if [ -z "$tape_drive" ] ; then
#     tape_drive=$default
# fi
# echo "Using $tape_drive\n"
tmp="/tmp/tmp.$$"
while :
do
    echo "Press ENTER to read the next tape label"
    echo "      or press Q to exit to previous menu"
    read foo
    if [ "$foo" = "Q" ] ; then
        break
    fi
    echo "Reading tape label...please wait."
    >$tmp
    $home/bin/tape_label.ksh get $tape_drive >$tmp
    tape_volser='cat $tmp|cut -f1 -d":"'
    create_date='cat $tmp|cut -f2 -d":"'
    scratch_date='cat $tmp|cut -f3 -d":"'

```

```

scr='echo "$create_date+$scratch_date"|bc'
contents='grep $tape_volser $tapes|cut -f5 -d ":"'
location='echo $tape_rec|cut -f6 -d ":"'
create_mdy='$s2d $create_date'
scratch_mdy='$s2d $scr'
$home/bin/scratch_test.ksh $tape_drive
if [ "$?" -eq "0" ] ; then
    scrF="Yes"
else
    scrF="No"
fi
if [ "$location" -eq "0" ] ; then
    loc="Onsite"
else
    loc="Offsite"
fi
echo "    Volser: $tape_volser"
echo "    Created: $create_mdy"
echo "Scratch on: $scratch_mdy"
echo "    Contents: $contents"
echo "    Scratch?: $scrF"
echo "    Location: $loc\n\n"
$label_log "Tape $volser read from drive $tape_drive"
done
}
backup_tape_db() {
    $label_log "Tape database backup"
    $home/bin/backup.ksh
    echo "Backup completed..."
    echo "Press ENTER to return to main menu"
    read F00
}
purge_backup_tape_db() {
    $label_log "Clean DB backups"
    $home/bin/clean_backups.ksh
    echo "Purge completed. Results mailed to admins"
    echo "Press ENTER to return to main menu"
    read F00
}
manual_edit_db() {
    $label_log "Manual tape DB edit"
    pid="$$"
    $home/bin/backup.ksh
    $home/bin/lockdb.ksh "Manual tape edit"
    vi $tapes
    vi $tape_history
    $home/bin/unlockdb.ksh "Manual tape edit"
    $home/bin/backup.ksh
}
init_tapes() {

```

```

while : ; do
    cd $home
    file="$host.foo.$$"
    >$file
    # Let's find the right tape drive
    ts='lsdev -Cc tape|grep -i available|awk {'print $1'}'
    for t in $ts ; do
        lscfg -v| $t|grep -i ult 1>/dev/null 2>&1
        if [ "$?" -eq "0" ] ; then
            drive=/dev/$t
            break
        fi
        echo "LTO Tape Drive not found!\n"
        echo "Enter tape drive you wish to use or..."
        echo "Press [ENTER] to abend"
        read foo
        QQ='echo "$foo"|wc -c|awk {'print $1'}'
        if [ "$QQ" -gt "2" ] ; then
            drive=$foo
            break
        else
            exit 1
        fi
    done
    chdev -l 'basename $drive' -a block_size=0 1>/dev/null 2>&1
    clear
    default="$drive"
    $home/bin/batch_tape_init.ksh $file
    echo "Initializing tape volsers:"
    cat $file
    $label_log "Initialize tape(s) `cat $file`"
    $home/bin/tape_label.ksh init $default $file
    echo "\n\n"
    rm $file
    echo "Press Y if there are more tapes, or N if not"
    echo "Then press ENTER"
    read sel input
    if [ "$sel" = "Y" -o "$sel" = "y" ] ; then
        :
    else
        break
    fi
done
}
mark_offsite() {
    tmp="/tmp/tapes.$$"
    tmp2="$tmp.1"
    > $tmp
    clear
    echo "Marking tapes offsite"
}

```

```

echo "Enter tape numbers to mark offsite.\n"
echo "Use format a-b,c,d,e-f ... ..\n"
echo "Enter a range of tapes and single tapes separated by commas"
echo "Or, enter just single tapes separated by commas.\n"
echo " "
echo "Press [ENTER] if no tapes to enter"
read volsers
if [ "$volsers" = "" ] ; then
    main
fi
echo "$volsers" | $home/bin/proc_range.ksh -q >>$tmp
A="A"
zero="00000000"
> $tmp2
for tape in `cat $tmp` ; do
    b="${zero}${tape}"
    c='echo "$b"|wc -c|awk {'print $1'}'
    d='expr $c - 1'
    e='expr $d - 6'
    f='echo "$b"|cut -c ${e}-${d}'
    g="${A}${f}"
    echo $g >> $tmp2
done
rm $tmp
for volser in `cat $tmp2` ; do
    record='grep $volser $tapes'
    flag='echo "$record" |cut -f6 -d:"'
    flag=1
    new_record1='echo "$record"|cut -f1-5 -d:"'
    new_record="${new_record1}:${flag}"
    $home/bin/db_update.ksh $new_record
    echo "Tape $volser marked offsite..."
    $label_log "Tape $volser marked offsite"
done
}
mark_onsite() {
    tmp="/tmp/tapes.$$"
    tmp2="$tmp.1"
    > $tmp
    clear
    echo "Marking tapes onsite"
    echo "Enter tape numbers to mark onsite.\n"
    echo "Use format a-b,c,d,e-f ... ..\n"
    echo "Enter a range of tapes and single tapes separated by commas"
    echo "Or, enter just single tapes separated by commas.\n"
    echo " "
    echo "Press [ENTER] if no tapes to enter"
    read volsers
    if [ "$volsers" = "" ] ; then
        main
    fi
}

```



```

fi
echo "$volsers" | $home/bin/proc_range.ksh -q >>$tmp
A="A"
zero="00000000"
> $tmp2
for tape in `cat $tmp` ; do
    b="${zero}${tape}"
    c='echo "$b"|wc -c|awk {'print $1'}'
    d='expr $c - 1'
    e='expr $d - 6'
    f='echo "$b"|cut -c ${e}-${d}'
    g="${A}${f}"
    echo $g >> $tmp2
done
rm $tmp
for volser in `cat $tmp2` ; do
    record='grep $volser $tapes'
    flag='echo "$record" |cut -f6 -d":"'
    flag=0
    new_record1='echo "$record"|cut -f1-5 -d":"'
    new_record="${new_record1}:${flag}"
    $home/bin/db_update.ksh $new_record
    echo "Tape $volser marked onsite..."
    $label_log "Tape $volser marked onsite"
done
}
change_retention() {
    base=86400
    tmp="/tmp/tapes.$$"
    tmp2="$tmp.1"
    > $tmp
    clear
    echo "Changing retention time for tapes"
    echo "Enter tape numbers to mark onsite.\n"
    echo "Use format a-b,c,d,e-f ... ..\n"
    echo "Tapes numbered from a thru b and e thru f will be changed"
    echo "as well as tapes numbered c and d.\n"
    read volsers
    retf=0
    while [ "$retf" -eq "0" ] ; do
        echo "\nEnter # of days to change retention"
        read retent
        if [ "$retent" -gt "0" ] ; then
            retf=1
        fi
    done
    newsave='echo "scale=0;$base*$retent"|bc'
    echo "$volsers" | $home/bin/proc_range.ksh -q >>$tmp
    A="A"
    zero="00000000"
}

```

```

> $tmp2
for tape in `cat $tmp` ; do
    b="${zero}${tape}"
    c='echo "$b"|wc -c|awk {'print $1'}'
    d='expr $c - 1'
    e='expr $d - 6'
    f='echo "$b"|cut -c ${e}-${d}'
    g="${A}${f}"
    echo $g >> $tmp2
done
rm $tmp
for volser in `cat $tmp2` ; do
    record='grep $volser $tapes'
    flag='echo "$record" |cut -f3 -d":"'
    new_record1='echo "$record"|cut -f1-2 -d":"'
    new_record2='echo "$record"|cut -f4-6 -d":"'
    new_record="${new_record1}:${newsave}:${new_record2}"
    written='echo "$record"|cut -f2 -d":"'
    $home/bin/db_update.ksh $new_record
    foo='expr $written + $newsave'
echo "Tape $volser retention changed..scratch date is now `s2d $foo`"
    $label_log "Tape $volser retention changed to $foo"
    sleep 5
done
}
other_tapes() {
    $label_log "Unknown tape report"
    clear
    now='$sec'
    now_date='s2d $now'
    tmp="/tmp/report.$$"
    >$tmp
    echo "All unknown tapes as of $now_date\n" >>$tmp
    printf $format "Volser" "Creation time" "Expiration time" "Host"
"LOC" "Contents" " Usage" >> $tmp
    printf $format "-----" "-----" "-----" "----" "----"
"-----" " ----" >> $tmp
    for tape_rec in `cat $tapes|sort -n -k3,6` ; do
        location='echo $tape_rec|cut -f6 -d":"'
        if [ "$location" -le "1" ] ; then
            continue
        fi
        volser='echo $tape_rec|cut -f1 -d":"'
        creation='echo $tape_rec|cut -f2 -d":"'
        lifetime='echo $tape_rec|cut -f3 -d":"'
        host='echo $tape_rec|cut -f4 -d":"'
        backup='echo $tape_rec|cut -f5 -d":"'
        creation_time='s2d $creation'
        foo='echo "$creation+$lifetime"|bc'
        expiration_time='s2d $foo'

```

```

        passes='grep $volser $tape_history|wc -l'
        if [ "$foo" -lt "$now" ] ; then
            volser1="${volser}${z1}"
            volser=$volser1
        fi
        loc="UNK"
        printf $format "$volser" "$creation_time" "$expiration_time"
"$host" "$loc" "$backup" "$passes" >> $tmp
    done
    cat $tmp
    echo "\nWould you like a printed report "
    echo "If Yes then enter the printer name"
    echo "(or just press [ENTER] to NOT print the report)"
    read q
    if [ -n "$q" ] ; then
        $home/bin/printl.ksh $q $tmp
        sleep 3
    fi
    cp $tmp $home/reports/othertapereport.txt
    rm $tmp
}
daily_report() {
    clear
    now='$sec'
    now_date='$s2d $now'
    tmp="/tmp/report.$$"
    echo "    To process tapes created within the past 24 hours, press
[ENTER]\n"
    echo "    Otherwise, enter the number of previous days to
process...\n"
    echo "    Enter days (or press [ENTER]) here --> \c"
    read chdate
    if [ ! -n "$chdate" ] ; then
        foo=1
    else
        foo="$chdate"
    fi
    foovar='echo "$foo*86400"|bc -l'
    target='echo "$now-$foovar"|bc -l'
    target_date='$s2d $target'
    >$tmp
    if [ "$TAPE_REPORT_TITLE" = "" ] ; then
        $label_log "Running daily tape movement report"
        title="Daily Tape Movement Report"
        reportfile="$home/reports/dailytapereport.txt"
    else
        title="$TAPE_REPORT_TITLE"
        $label_log "Running $title movement report"
        reportfile="$home/reports/$TAPE_REPORT_FILE"
    fi
}

```

```

echo "${title} For $now_date\n" >>$tmp
echo "Tapes created between $target_date and $now_date\n" >> $tmp
printf $format "Volser" "Creation time" "Expiration time" "Host"
"LOC" "Contents" " Usage" >> $tmp
printf $format "-----" "-----" "-----" "----" "----"
"-----" " ----" >> $tmp
counter=0
for tape_rec in `cat $tapes|sort -n -k3,6` ; do
    volser=`echo $tape_rec|cut -f1 -d":"`
    creation=`echo $tape_rec|cut -f2 -d":"`
    if [ "$creation" -lt "$target" ] ; then
        continue
    fi
# ###        sleep 2
counter=`expr $counter + 1`
lifetime=`echo $tape_rec|cut -f3 -d":"`
host=`echo $tape_rec|cut -f4 -d":"`
backup=`echo $tape_rec|cut -f5 -d":"`
location=`echo $tape_rec|cut -f6 -d":"`
creation_time=`$s2d $creation`
foo=`echo "$creation+$lifetime"|bc`
expiration_time=`$s2d $foo`
passes=`grep $volser $tape_history|wc -l`
if [ "$location" -eq "0" ] ; then
    loc="ON"
elif [ "$location" -eq "1" ] ; then
    loc="OFF"
else
    loc="UNK"
fi
scrf=0
if [ "$foo" -lt "$now" ] ; then
    volser1="${volser}${z1}"
    volser=$volser1
    scrf=1
fi
printf $format "$volser" "$creation_time" "$expiration_time"
"$host" "$loc" "$backup" "$passes" >> $tmp
done
echo "\nTotal tapes: $counter" >> $tmp
cat $tmp
echo "\nWould you like a printed report "
echo "If Yes then enter the printer name"
echo "(or just press [ENTER] to NOT print the report)"
read q
if [ -n "$q" ] ; then
    $home/bin/printl.ksh $q $tmp
    sleep 3
fi
cp $tmp $reportfile

```

```

        rm $tmp
    }
    # #####
    # Main logic
    # #####
    if [ "$batchf" -eq "0" ] ; then
        main
    fi
    if [ "$report" = "scratch" ] ; then
        scratch_report
    elif [ "$report" = "all" ] ; then
        all_tape_report
    elif [ "$report" = "daily" ] ; then
        daily_report
    elif [ "$report" = "offsite" ] ; then
        offsite_report
    elif [ "$report" = "onsite" ] ; then
        onsite_report
    elif [ "$report" = "UNK" ] ; then
        other_tapes
    elif [ "$report" = "backup" ] ; then
        backup_report
    elif [ "$report" = "host" ] ; then
        host_report
    elif [ "$report" = "history" ] ; then
        history_report
    elif [ "$report" = "kit" ] ; then
        scratch_report
        all_tape_report
        backup_report
        history_report
        offsite_report
        onsite_report
        other_tapes
    else
        echo "Invalid report entered\n"
        echo "Valid reports are:"
        echo "kit (all reports), scratch, all, backup, offsite, onsite, UNK,
host or history"
        exit 1
    fi

```

UNLOCKDB.KSH

```

#!/usr/bin/ksh
home=/var/tapesys
host='hostname'
pid="$$"
lock="$home/db/dbupdate.lck"

```

```

label_log="$home/bin/label_log.ksh"
Date='date'
Args="$@"
image="%-10.10s%-60.60s%-25.25s\n"
if [ -r $lock ] ; then
    lockhost='cat $lock|cut -f2 -d":"|awk {'print $1'}'
    if [ "$lockhost" != "$host" ] ; then
        printf $image "$host:" "database lock attempt failed $Date"
        "($Args)" >> $home/log/dblocklog.log
        $label_log "Database lock failed for unlock"
        while : ; do
            echo "The tape database is currently being updated"
            cat $lock
            echo "I will wait until the other process finishes"
            echo "To clear this condition, remove the file $lock."
            echo "But be sure you know what you are doing..."
            sleep 10
            if [ ! -r $lock ] ; then
                echo "\nLock has been released...continuing\n\n"
                break
            else
                echo "Lock is still present"
            fi
        done
        fi
        rm $lock 1>/dev/null 2>&1
        $label_log "Database unlocked"
        printf $image "$host:" "database unlocked $Date" "($Args)" >> $home/
log/dblocklog.log
    else
        printf $image "$host:" "No DB lock found $Date" "($Args)" >> $home/
log/dblocklog.log
        $label_log "Unable to unlock an unlocked database"
        echo "$host:" "No DB lock found $Date" "($Args)"
    fi
fi

```

WEEKEND_REPORT.KSH

daily_report.ksh

WEEKLY_REPORT.KSH

daily_report.ksh

Put the following in a directory called cron.

CRON.ENTRIES

```
30 18 * * 0 /var/tapesys/bin/clean_backups.ksh
0,30 * * * 1-5 /var/tapesys/bin/scratch_report.ksh 1>/dev/null 2>&1
0 13 * * 2-6 /var/tapesys/bin/daily_report.ksh 1>/dev/null 2>&1
0 8 * * 1 /var/tapesys/bin/weekend_report.ksh 1>/dev/null 2>&1
0 15 * * 5 /var/tapesys/bin/weekly_report.ksh 1>/dev/null 2>&1
```

The following files are placed in a directory called db:

- tape_drive.def
- tape_history.db
- tapes.db.

The files are in a directory called Log:

- db_backuplog.log
- db_cleanbackuplog.log
- dblocklog.log
- dblocklog.log.save
- tapesyslog, and
- tapesyslog.save.

The reports directory contains a number of reports including:

- agereport
- alltapereport
- backupreport
- dailytapereport
- discardreport
- hostreport
- offsitereport
- scratchreport
- weekendreport

- weeklyreport.

Bill Verzal
Project Leader
Komatsu America (USA)

© Bill Verzal 2005

In addition to *AIX Update*, the Xephon family of *Update* publications now includes *CICS Update*, *DB2 Update*, *MQ Update*, *MVS Update*, *RACF Update* and *TCP/SNA Update*. Details of all of these can be found on the Xephon Web site at www.xephon.com.

Please note that the correct contact address for Xephon Inc is PO Box 550547, Dallas, TX 75355, USA. The phone number is (214) 340 5690, the fax number is (214) 341 7081, and the e-mail address to use is info@xephon.com.

Why not share your expertise and earn money at the same time? *AIX Update* is looking for shell scripts, program code, JavaScript, etc that experienced users of AIX have written to make their life, or the lives of other users, easier. We are also looking for explanatory articles, and hints and tips, from experienced users.

We will publish your article (after vetting by our expert panel) and send you a cheque, as payment, and two copies of the issue containing the article. Articles can be of any length and should be e-mailed to the editor, Trevor Eddolls, at trevore@xephon.com.

AIX news

IBM has announced CICS TG V6.0.1 and CICS Universal Client (CICS UC) V6.0.1. Both products now support Linux on POWER and AIX (including V5.3).

In addition, they both now support Red Hat Enterprise Linux (RHEL)4.

IBM claims that this service release can also deliver significant run-time performance enhancements for request processing. The enhancement is provided in the interface to the Client daemon.

For further information contact:

URL: <http://www-306.ibm.com/common/ssi/fcgi-bin/ssialias?subtype=ca&infotype=an&appname=iSource&supplier=897&letternum=ENUS205-147>.

* * *

RSD has announced Mail2Folders, a new component that integrates e-mail with all enterprise business-critical content. It is designed to empower the enterprise to manage e-mail communications.

Working with an RSD Folders server, Mail2Folders works with all IMAP-compliant e-mail applications such as Outlook and Notes. Within the e-mail application, the user simply drags-and-drops an e-mail into an archive folder. Mail2Folders ensures that the e-mail is archived and available to all users authorized to view that folder. Mail2Folders stores e-mails in

a standard RFC 2822 archive format, independent from the e-mail application type.

RSD Folders is a document management system that runs on AIX, Sun Solaris, HP-UX, Linux, OS/390, Unix, and Windows (NT/2000/XP) server platforms.

For further information contact:

URL: www.rsd.com/newsEvents/pressReleases/press_Mail2Folders_160605.html.

* * *

Pronto has announced the latest version of Pronto-Xi. Its Digital Dashboard supports multiple levels of information consolidation, down to the underlying transactions. It also provides colour coding of information to allow at-a-glance review of detail.

PRONTO-Xi allows users to manage all phases of the supply chain. The company thinks of it as more than an Enterprise Resource Planning (ERP) System.

The new version can access PRONTO-Xi data on AIX, as well as Solaris V7, Red Hat Linux, and Windows 2000/XP.

For further information contact:

URL: www.pronto.com.au.

* * *

