



# 72

# AIX

*October 2001*

---

- 3 Find a file and execute a command
  - 5 AIX upgrade from V4.2.x to V4.3.3
  - 13 Using sendmail on AIX in a Microsoft Exchange environment
  - 15 Analysing AIX performance using the jazizo tool of PTX 3.0
  - 22 Number conversion utility
  - 30 Tape Manager
  - 52 AIX news
- 

© Xephon plc 2001

update

# AIX Update

---

## Published by

Xephon  
27-35 London Road  
Newbury  
Berkshire RG14 1JL  
England  
Telephone: 01635 38342  
From USA: 01144 1635 38342  
E-mail: [trevore@xephon.com](mailto:trevore@xephon.com)

## North American office

Xephon  
PO Box 350100  
Westminster, CO 80035-0100  
USA  
Telephone: 303 410 9344

## Subscriptions and back-issues

A year's subscription to *AIX Update*, comprising twelve monthly issues, costs £180.00 in the UK; \$275.00 in the USA and Canada; £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 1995 issue, are available separately to subscribers for £16.00 (\$24.00) 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/aixupdate>; you will need to supply a word from the printed issue.

## Editors

Trevor Eddolls and Richard Watson

## 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 £170 (\$260) per 1000 words and £100 (\$160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 (\$80) per 100 lines. In addition, there is a flat fee of £30 (\$50) per article. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from [www.xephon.com/contnote.html](http://www.xephon.com/contnote.html).

---

© Xephon plc 2001. 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. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

*Printed in England.*

## Find a file and execute a command

Quite often, while investigating a problem, you need to examine numerous files spread throughout a number of different directories. You thus have the choice of changing directories each time you examine another file or otherwise specifying the full path name of the file.

Script **fx** provides an alternative to this. By defining an environmental variable **FXPATH** as a colon-separated list of directories, it allows you to treat any file as though it were in the current directory by simply prefixing your command with **fx**.

For example, to check the permissions on the password command, enter:

```
fx ls -l passwd
```

In addition to searching for the specified file in your define **FXPATH** directory list, the **fx** script also searches all directories defined in your **PATH**.

### THE FX SCRIPT

```
#!/usr/bin/ksh
# Script: fx
# Aim:    Find a file then excute the specified command

awk="/usr/bin/awk"
basename="/usr/bin/basename"
head="/usr/bin/head"
ls="/usr/bin/ls"
pwd="/usr/bin/pwd"
sed="/usr/bin/sed"

# Function

abend()
{
    echo "$SCRIPT: $* " 1>&2
    exit 1
}

checkParms()
```

```

{
  count=`echo "$PARMS" | $awk '{print NF }' `
  if test $count -lt 2
  then
    abend "No file specified"
  fi
  COMMAND=`echo "$PARMS" | $awk '{print $1 }' `
  OPTIONS=`echo "$PARMS" | $awk '{for(i=2;i<NF;++i){printf "%s ", $i}
printf "\n" }' `
  FILENAM=`echo "$PARMS" | $awk '{print $NF }' `

  if test "$COMMAND" = ""
  then
    abend "No command specified"
  fi
  return
}

```

```
getFullNames()
```

```

{
  fullname=`whence $COMMAND`
  if test "$fullname" = ""
  then
    abend "Command $COMMAND not found in \ $PATH"
  fi
  PWD=`$pwd`
  possible=`echo "$PWD:$PATH:$FXPATH:" | $sed "s,,:,/$FILENAM ,g" `
  FULLNAME=`$ls $possible 2>/dev/null | $head -1 `
  if test "$FULLNAME" = ""
  then
    abend "File $FILENAM not found in \ $PATH or \ $FXPATH"
  fi

  return
}

```

```
# Start of main script
```

```

SCRIPT=`$basename $0`
PARMS=`echo "$*" | $sed "s,\,,g" `
checkParms
getFullNames
$COMMAND $OPTIONS $FULLNAME

exit $?

```

---

*Roger Wickings*  
*Systems Programmer*  
*FT Interactive Data (UK)*

© Xephon 2001

---

## AIX upgrade from V4.2.x to V4.3.3

The following steps outline the basic procedure for upgrading to AIX V4.3.3. The entire upgrade may take several hours, depending on the speed of the hardware. Doing a good full back-up of the system to tape before starting is, of course, strongly recommended. The back-up tape should be write-locked and labelled with the date and a label such as 'AIX pre-V433 upgrade', and the tape should be stored in a safe place.

### PRE-UPGRADE STEPS

Create a directory called **/upgrade\_save** in order to save some important files prior to beginning the upgrade:

```
mkdir /upgrade_save
cp -p /etc/motd /upgrade_save/
cp -p /etc/inittab /upgrade_save/
cp -p /etc/inetd.conf /upgrade_save/
cp -p /etc/rc.net /upgrade_save/
```

For CDE customization preservation:

```
cp -p /.dtprofile /upgrade_save/
cp -p /usr/dt/app-defaults/en_US/Dtterm /upgrade_save/
```

Verify the correctness of the user accounts – the command is:

```
usrck -y ALL
```

Rebuild the user login account cache. For pre-V4.3 systems use:

```
mkpasswd -v /etc/passwd
```

Note: after the upgrade to V4.3, the equivalent command to use from then on would be:

```
mkpasswd -f
```

which rebuilds indexes.

Verify the correctness of the group definitions using the command:

```
grpck -y ALL
```

Check for sufficient disk space in **rootvg**, and specifically the / (root)

and **/tmp** directories. (It's a good idea for **/** and **/tmp** to have at least 40MB free to ensure that the migration will have enough room to succeed.) Use **smitty fs** to increase the **/** or **/tmp** space if necessary.

Prevent any remote access (including user logins, file transfers and job submissions) by either creating an **/etc/nologin** file with an appropriate message in it, or by temporarily disconnecting the network LAN cable from the back of the RS/6000. This means that only the console will be able to sign in.

You can create an **/etc/nologin** file containing an informative message such as:

```
*****
The system is currently being upgraded and logins are disabled at this
time. Please try again later.
*****
```

Note: the presence of the **/etc/nologin** disables FTP access and user logins.

Stop any application jobs and kill any user processes still lingering around.

Use the appropriate application shutdown scripts for your running applications.

If users are logged in, broadcast a message to them to sign off:

```
wall    "*** SYSTEM GOING DOWN FOR MAINTENANCE ***"
wall    "*** PLEASE LOG OFF ***"
```

After giving them a minute or two to voluntarily logoff, stop any remaining user processes:

```
kill -9 <pid>
```

etc.

If you are using the **at** queue for scheduling jobs, send any pending jobs to the save directory:

```
mkdir /upgrade_save/atjobs
cd /var/spool/cron/atjobs
mv * /upgrade_save/atjobs/
```

To stop the cron jobs from executing during the upgrade:

```
cd /var/spool/cron/crontabs
mv root root_save
```

Do any other non-root crons as necessary.

Do a full system back-up using **Sysback** or **mksysb** to tape.

If desired, verify the integrity of the **Sysback** back-up using the command:

```
sysrestore -lf /dev/rmt0
```

If connected to either RAID or SSA for the non-rootvg volume groups, unmount the filesystems and vary off the volume group(s):

```
lsvg -o | lsvg -il > /upgrade_save/vg_info.out
                                                    (save the VG information)
umount all
varyoffvg <non-rootvg vname>
                                                    (do for each non-root VG)
mount all
```

## AIX UPGRADE STEPS

Note: ensure that there is no tape in the tape drive! This is so that the system does not boot from tape instead of from CD.

Insert the AIX V4.3.3 (1 of 4) CD into the CD drive (use a CD caddy if necessary).

Turn the key to the *service* position (if the system uses a physical key).

If the system has no physical key, then set it to boot from CD using either the **diag** command or the SMS utility. The easiest way may be to simply set the bootlist from the command prompt prior to shutting down the system, as follows:

```
bootlist -m normal cd0
```

Then verify the bootlist with the command:

```
bootlist -m normal -o
```

Restart the system using the command:

```
shutdown -Fr
```

If the system does not autoreboot, power-on the system or double-press the reset button.

The RS/6000 should now boot the migration installation from the CD drive.

At the *select console* screen, select by pressing the digit 1 key and *Enter* (or F1 if using a graphics console).

At the *select language* screen, select by pressing digit 1 key and *Enter* (or F1 if using a graphics console).

At the next screen, select **#2** (*Change/Show Installation Settings and Install*).

Ensure that the following options are set:

```
migration
hdisk0
#2  English (United States) (ensure that all 3 settings say ENGLISH)
Install T.C.B. ?   No
#0  Install with current settings
```

At the *migration confirmation* menu, choose **A0**.

Migration will say:

```
>n of (several hundred) processed...=
```

as it goes and, depending on the RS/6000 model, should take more than one hour.

Note that there may be times when it appears to be hung and doing nothing for a while. This may occur just after it says:

```
<<end of copyright notice for bos.terminfo>>
```

This is expected and patience is required.

Once migration is underway, turn the key to the *normal* position, if the system has a physical key.

Once the installation is complete, the system will do a normal reboot (as long as the key is in the *normal* position, for those systems that have a physical key).

## POST-UPGRADE STEPS

The system should come up with the *installation assistant* menu. You can select any desired items from that menu, or choose **Exit Installation Assistance**.



Perform some checks using the following commands:

- **date** (check the date, time and timezone, and daylight savings time setting).
- Make a new **/etc/nologin** file (the reboot deletes it).
- **oslevel** (at this point it should only read '4.3.0.0').
- **view /var/adm/ras/devinst.log** (if desired, for information).
- **view /var/adm/sw/installp.summary** (if desired, for information).

If failures were logged for netbios.rte and perfagent.\*, you can ignore these for now.

- **smitty alog** (if desired, to check any logs).
- **lspv -a** (just to take a look at the paging space).

Check the **/var/adm/ras/BosMenus.log** file. It shows what options you chose when you started.

Check dumpspace size – the system must have enough space for the dump – use the following command to calculate the MBs required:

```
sysdumpdev -e
```

AIX will use the paging space for writing system dumps to by default. It is recommended that **hd7** be used as a dedicated dump logical volume instead.

Make sure that the dedicated dump logical volume **hd7** is set as the dump device and is large enough using the following commands:

- **sysdumpdev -l** – make sure hd7 is set as primary device.
- **lslv hd7** – show the PP size and number of PPs.
- **mklv -y hd7 -t sysdump rootvg ?** – if hd7 doesn't exist.
- **extendlv hd7 ?** – if hd7 isn't large enough.
- **sysdumpdev -P -p=/dev/hd7'** – if hd7 isn't the primary dump device.

'?' is the number of physical partitions needed to get the required amount of dumpspace.

Check for autoreboot to be true using the command:

```
lsattr -El sys0
```

Look in right-hand column.

Update the **/etc/motd** with the one previously saved (assuming you have a customized message-of-the-day file) using the command:

```
cp -p /upgrade_save/motd /etc/
```

Check the differences between the saved **inittab** and the new one using the command:

```
diff /etc/inittab /upgrade_save/inittab
```

Check the differences between **inetd.conf** and the new one. You might have to disable some things that became enabled as a result of the migration that you prefer to be inactive. Use the command:

```
diff /etc/inetd.conf /upgrade_save/inetd.conf
```

Put back the saved DT (Common Desktop Environment) files in **/upgrade\_save** to the appropriate directories if required.

Edit **/etc/rc.net** if necessary to ensure that any *no* options you may have set in that script are still there (see version saved in **/upgrade\_save**).

Create a file called **/etc/netsvc.conf** containing the following text:

```
hosts=bind4,local
```

The **/etc/netsvc.conf** file is used to specify the ordering of certain services in AIX: specifically, name resolution for sendmail, the **gethostbyname**, **gethostaddr**, and **gethostent** subroutines, in addition to alias resolution for sendmail. Assuming that DNS is in use on the system (**/etc/resolv.conf** file exists), the above configuration should work fine. The **bind4** variable uses BIND/DNS services for resolving only IPV4 addresses, and **local** uses the **/etc/hosts** file.

Insert the AIX 4.3.3 (2 of 4) CD and use **smitty update\_all** to apply updates to all the filesets installed on your system.

Repeat the above step using the AIX 4.3.3 (3 of 4) CD and **smitty update\_all**.

Insert the latest (white cover) AIX UPDATE CD for V4.3.3 now. On

the back of the CD case it should say something like LCD4-0995-*xx* (the higher the *xx* number, the more recent the level of the updates contained within). Use the command:

```
smitty update_all
```

This may take an hour or more to update the system to the latest filesets. It will probably update over 100 filesets.

Note: if you see certain failures for netbios.rte or perfagent.\*, these are normally not a problem and can be dealt with later on.

Shutdown the system and reboot it using:

```
shutdown -Fr
```

The system should be rebooted for the software installations to take full effect.

Ensure that once the system reboots, the non-rootvg volume groups are varied on again using the command:

```
lsvg  
varyonvg <each non-root VG>  
mount all
```

Verify that **mount = true** is set in **/etc/filesystems** for all the desired filesystems.)

If you are using the **syslog** utility and directing the output to **/tmp/syslog**, the upgrade removes the **syslog** file so it must now be recreated. The **syslog** configuration is done by editing the file **/etc/syslog.conf**. For all levels of system information to be logged to it (which is very useful when troubleshooting various AIX problems), the following line should exist at the end of the file:

```
*.info /tmp/syslog
```

*Touch* a new **/tmp/syslog** and *refresh* **syslogd** using the following commands:

```
touch /tmp/syslog  
refresh -s syslogd
```

Restore the cron and **at** jobs that were saved in an earlier step:

```
cd /var/spool/cron/atjobs
```

```
mv /upgrade_save/atjobs/* ./
```

```
cd /var/spool/cron/crontabs
```

Note: the root crontab will be a new version. You may have to examine the saved version of the root crontab in **/upgrade\_save** and put back any desired root cron items from your saved version into the new V4.3.3 root crontab. It is a good idea to uncomment the **/etc/skulker** line in the root crontab as well.

For any non-root crons that were renamed before, you should rename them back to their normal names again now, and then refresh the cron job as shown below:

```
ps -ef |grep cron
```

to get the PID of the cron process.

```
kill -TERM <pid of cron process>
```

to refresh cron.

Check the operating system level using the command:

```
oslevel
```

It should now show 4.3.3.0.

Note: if the AIX level does not show 4.3.3.0, then enquire to find out what fileset(s) are required to get it up to the 4.3.3 level using the command:

```
oslevel -l 4.3.3.0
```

Ensure that the **/etc/nologin** file has been removed in order to permit users to access the system again.

Once the upgrade is done, there may be other steps to take to ensure that existing non-IBM products installed on the system still function properly. For example, the Merant (MicroFocus) COBOL product will need to have its rts32 (run time library) rebuilt for its File Handler Utilities to work properly with the V433 AIX libraries. Check with your vendors ahead of time to find out if there are any steps to take for their products to be compatible with AIX V4.3.3.

---

*Mike Stanton*  
*Supervisor Mid-range Systems*  
*Mercedes-Benz (USA)*

© Xephon 2001

---

## Using sendmail on AIX in a Microsoft Exchange environment

We have 14 SP/2 nodes and seven RS/6000s, none of which use Internet services or TCP/IP name resolution; just basic hostname (no domains) and **/etc/hosts** address resolution – we've kept it simple because we could. We did reach a point, however, where we wanted to send e-mail from back-up scripts or alerts to our Microsoft Exchange e-mail accounts. The desktop side of things here, like many other places, is exclusively Microsoft. Mail is via Outlook and Exchange.

After a couple of trial and error experiments we found that it was quite easy to send e-mail from one of the nodes or stand-alone boxes to our Outlook accounts.

The first step is to add the IP address of your Exchange server to **/etc/hosts**, for example: 199.92.146.55 NTMAILBOX.

Then take a vanilla *sendmail.cf* file, and make the following changes. Change *Dyourhostname* to *Dwlocalhost* (localhost is presumably in your **/etc/hosts**). Make sure the *Cwlocalhost \$w \$?m\$w.\$m\$.* line is not commented and reads as I've shown.

Make sure you have an **/etc/netsvc.conf** file that contains the line *hosts=local*.

At this point you should be able to send an e-mail. Try a simple experiment:

```
mail -sthis_is_a_test @NTMAILBOX:bob.smith@bhs.com < /etc/motd.
```

Replace *bob.smith@bhs.com* with your own Internet e-mail address. **-s** is followed by the subject, **/etc/motd** is the text you are re-directing to your e-mail.

Note the format of the e-mail address, which is a bit different from standard. This is because the mail command expects the address to begin with an @, and of course, your own e-mail address is going to have an @. The colon in the middle of the address is used to separate them.

## DEBUGGING TIP

If you are having timeout or handshaking problems, try using **-v** on the mail command line. For example:

```
mail -v -sthis_is_a_test @NTMAILBOX:bob.smith@bhs.com < /etc/motd
```

You'll get back a session that looks something like this, and contains lots of helpful information:

```
@ntmail:bob.smith@bhs.com... Connecting to springfiled.bhs.com. via
esmtplib...
220 bhsexcl.bhs.com ESMTP Server (Microsoft Exchange Internet Mail
Service 5.5.2
650.21) ready
>>> EHL0 bhsspctrl
250-bhsexcl.bhs.org Hello [199.92.146.231]
250-XEXCH50
250-HELP
250-ETRN
250-DSN
250-SIZE 0
250-AUTH LOGIN
250 AUTH=LOGIN
>>> MAIL From:<root@bhsspctrl> SIZE=687
250 OK - mail from <root@bhsspctrl>; can accomodate 687 bytes
>>> RCPT To:<@springfiled.bhs.com:bob.smith@bhs.com>
250 OK - Recipient <bob.smith@bhs.com>
>>> DATA
354 Send data. End with CRLF.CRLF
>>> .
250 OK
@ntmail:bob.smith@bhs.com... Sent (OK)
Closing connection to springfiled.bhs.com.
>>> QUIT
221 closing connection
```

## CONCLUSION

As you can see, it is a simple task to get this working, but quite beneficial, especially if you prefer to see all of your e-mails in one place each morning. This also saves us the trouble of cross-training staff on Unix, who would be using it only for mail.

---

*David Miller*  
*Database Architect*  
*Baystate Health Systems (USA)*

© Xephon 2001

---

# Analysing AIX performance using the jazizo tool of PTX 3.0

Performance Toolbox Version 3.0 is an LPP (Licensed Packaged Product) designed to enable both short- and long-term performance monitoring of the AIX operating system. Although an attempt has been made to provide compatible versions for Solaris and HP-UX operating systems, in practice these versions had limited functionality both in features and, most importantly, in support for recent versions of these operating systems. This article will describe one of the new features of the recently announced Version 3.0 of this important product.

## INSTALLATION

You'll first notice new features in the product during the initial installation. In addition to the latest operating system patches (on top of the base AIX 4.3.3), you must ensure that Java Version 1.3 has already been installed. This is required because some of the new features have been implemented using the Java programming language. The same version of the product supports both AIX 4.3.3 and AIX 5.1.

## INTRODUCING JAZIZO

The jazizo program was designed to enable the evaluation of system data over long periods of time, allowing system administration staff to identify system usage patterns and to plan for future upgrades. The tool can be configured to show only data of interest in graphical or tabular formats. Custom data display configurations can be created, edited, and saved by the operator. In addition, reports can be generated covering specific time periods, and data-reduction options are provided to assist the analyst. The data, analysed by jazizo, is collected and stored in record files by the **xmtrend** daemon.

## XMTREND

The syntax for the `xmtrend` daemon is as follows:

```
xmtrend [ -f config_file ] [ -d recording_dir ] [ -n recording_file ] [ -t trace_level ]
```

where:

- *-f config\_file* is the name of a configuration file containing a specification of the system statistics to be monitored by `xmtrend`. The default configuration file name is */etc/perf/xmtrend.cf*.
- *-d recording\_dir* is the name of the directory to contain the recording files. The default directory is */etc/perf*.
- *-n recording\_file* is the base name of the recording file. The extension of the filename is always the file creation date in the format `YYMMDD`. The default base name is **xmtrend**.
- *-t trace\_level* is an integer between 1 and 9 specifying the desired level of trace information to be reported by `xmtrend` into the log files. The higher the value, the greater the amount of reported trace information. The name of the default log file is */etc/perf/xmtrend.log1*. The directory and base name of the log file is influenced by the values of the *-d* and *-n* flags.

In order to create a suitable set-up file you can use sample configuration file */usr/lpp/perfagent/xmtrend.cf*. You use output from the **xmpeek -l** command in order to get a list of system statistics available on your system. You can ensure that **xmtrend** starts automatically with each system reboot by placing its invocation line in a standard system start-up file such as */etc/rc.local*. You can verify the daemon execution by performing the following command:

```
ps alxww | grep xmtrend | grep -v grep
```

If the daemon is not running you should examine the log files. One possible cause is that the recording directory does not have enough free space to accommodate the recorded data.



## JAZIZO MENUS

The jazizo program has five different menus:

- File menu
- Edit menu
- View menu
- Configurations menu
- Report menu
- Help menu.

The *file* menu contains the following options:

- The *open recording file* option is used to select a recording file to be analysed by jazizo. By default only files starting with **xmtrend** will be displayed, but you can change the file type to be “All files (\*.\*)” if your recording files have different name. You can use various navigation icons to select different directories to look for recording files. Select one recording file and press *OK* to proceed. When a recording file is selected, the *metric selection* menu is displayed. At this stage it is possible to choose the metrics as well as timeframe to be displayed by the tool. I will describe the metric and timeframe selection later.
- The *open configuration file* option enables the selection of a configuration file containing pre-defined characteristics of data that the user is interested in analysing. Navigation and selection processes are similar to those of *recording files*, except that the only supported file extension is *.cf*.
- The *save configurations* and *save configurations as* options enable users to save the characteristics of the currently displayed metrics as a named configuration file, to be used for similar analysis situations in the future.
- The *print* option initiates a print dialog window that enables output from the graph displayed by jazizo to be sent to a printer or postscript file.

- The *exit* option closes all windows and terminates the jazizo program.

The *edit* menu has the following options:

- The *metric selection* option is used to select the metrics and the timeframe to be displayed by the tool. The top part of the dialog box shows the recording period to be graphed; the user can adjust this period by the selection of the *Edit Start/Stop* button. The user can adjust the dates as well as exact hours for the start and the end of the analysed period. The rest of the dialog box displays on the left the metrics contained in the recordings file and on the right the metrics that have been selected by the user for analysis. In both parts the metrics can be displayed in either *Tree View* or *List View*. Selecting the *Tree View* enables the display of the metrics in hierarchical fashion according to their logical classification. For instance the CPU statistics appear in the general class of CPU metrics, further classified according to different CPU IDs, which are further broken down by type of CPU metrics such as kernel, user, or idle time. Selecting the *List View* displays each metric on its own line, sorted in alphabetical order by name. The selection of statistics for inclusion or removal from the jazizo display is done in the following fashion:
  - One selection – single mouse click.
  - More than one selection – *Shift* and mouse click.
  - A range of selections – *Control* and mouse click.
- The *graph properties* option is used to define how the metric data is graphed, as well as to define the exact statistics that have to be displayed. This is a very complex dialog box that has the following components:
  - A text field that enables selection of a title for the displayed graph.
  - A Y-axis section that enables the user to select whether the graphs are to be plotted based on the *common scale* or displayed with a *separate scale* for each plotted graph.

- A *mean* options section enables the user to select the display of a *trend line* for the data, which is calculated according to the *least squares best-fit* method. The possible options are to display the *trend line* based on all data points or only visible data points, or to turn *off* the display of the trend line.
- The *ranges* section enables the display of the maximum or minimum value for each sample instead of the default mean.
- Then you may select whether the specified options will be activated for all metrics or the metrics that are going to be added to the graph from this time onwards.
- Finally the *select a metric* option enables the manipulation of the display options for a particular displayed metric, which can be selected from the displayed metrics list. The metric's label, colour, and additional properties can be adjusted.

The *view* menu enables the user to filter out the displayed data according to the desired time period. The default displayed time period is *day by hour*. *Week by hour* is another predefined selection. Additionally it is possible to select *All*, *Year*, *Quarter*, or *Month* data and plot it, selecting ticks for *month*, *week*, *day*, or *hour*. The *reduce data by* tick box enables users to reduce the number of displayed data points, such that only a single data point is selected for each tick mark.

The *configurations* menu enables the user to manipulate configuration files that contain collections of definitions of particular jazizo options. The user can add the current configuration to the menu of available configurations, remove a configuration from a menu, update the current configuration with newly-selected options, and select a new configuration from the list of available ones.

The *reports* menu enables display and printing of a table that contains the numerical values of the data being graphed. The *summary reports* can contain only the points displayed in the visible Window or all points contained in the graph. The report table is organized by the individual metrics. The following are the columns of the report:

- Name – the metrics label.
- Observations – the total number of points for the metric.
- Average – the average of all metric values.
- Maximum – the maximum of all metric values.
- Minimum – the minimum of all metric values.

The *selected metric report* enables the generation of a report for only a single highlighted metric. Values for each measurement are displayed along with a timestamp. The *print* button enables printing of the report, the *close* button closes the report window.

The *help* menu has a single selection, which enables display of a jazizo chapter from the manual.

## JAZIZO DISPLAY ELEMENTS

The main window of the jazizo tool contains three additional elements besides the menu bar – graph window, legend panel, and status panel.

A scrollable graph window displays the activity of the selected metrics in the format specified by the user. The top horizontal axis represents time and the left vertical axis displays a measurement range. If the displayed metrics have different ranges, the vertical axis is displayed only if one of the metrics has been selected.

The legend panel of jazizo is displayed below the graph. The legend displays a list of the graphed metrics. A coloured rectangle appearing to the left of the metric's name indicates the colour of the appropriate graph. The Y-range of the metric is displayed to the right of its name.

When the metric name is left-clicked the metric's graph is highlighted, allowing better identification of the metric. The highlighted metric is plotted using a bright white colour; the Y-axis is changed to conform to the selected metric. A second left-click deselects the metric.

When the metric is right-clicked, a menu containing the following selections is displayed:

- *Edit* metric – opens a metric property menu. This menu is similar

to the one that is displayed when *Edit* from the main menu bar is selected and a metric label is selected.

- *Edit* graph – opens a graph property menu. This is the same menu that is displayed when *Edit/Graph properties* from the main menu bar is selected and a metric label is selected.
- *Hide* or *show* the metric in the graph – allows users to temporarily hide or display selected metrics.
- *Hide-all* – hides all metric graphs defined in the legend from the graph.
- *Show-all* – shows all metric graphs defined in the legend from the graph.

Below the legend panel appears the display of the status panel. This panel displays general information such as the name of the recording file and information messages. Only the last message is displayed, but the user can scroll back to review previously issued messages.

## SUMMARY

Previous versions of jazizo had a limited ability to display historical data. The new edition enables the system administrator to perform valuable historical analysis of system performance data in order to be able to perform real-time monitoring as well as capacity planning.

## REFERENCES

- 1 *Performance Toolbox Version 2 and 3 Guide and Reference*, SC23-2625-06
- 2 *AIX 5L Performance Tool Handbook*, SG24-6039-00, to be published by IBM
- 3 [http://www-1.ibm.com/servers/aix/products/ibmsw/system\\_man/perftoolbox.html](http://www-1.ibm.com/servers/aix/products/ibmsw/system_man/perftoolbox.html)

---

*Alex Polak*  
*System Engineer*  
*APS (Israel)*

© Xephon 2001

---

# Number conversion utility

## INTRODUCTION

Usage of hexadecimal number is very common at the system level. Lots of system software requires that you input data in hex. The utilities dtoh.sh (decimal to hex) and htod.sh (hex to decimal) can be used stand-alone or with other programs to perform conversion involving decimal and hexadecimal numbers.

## LISTING OF HTOD.SH

```
#####
# Name      : htod.sh(hexadecimal to decimal )
# Overview  : The script converts a hexadecimal number to a decimal
#            number.
# Input     : signed/unsigned Hexa Decimal Number
# Notes    : 1. The algorithm for conversion is as follows:
#            A = 10 B = 11 C = 12 D = 13 E = 14 F = 15
#
#            place value -> 6      5      4      3      2      1      0
#            base        -> 16     16     16     16     16     16     16
#
# number to be converted
# -> 1      2      3      4      5      E      B
#
#      6      5      4      3      2      1      0
# 16 * 1  + 16 * 2  + 16 * 3 + 16 * 4 + 16 * 5 + 16 * 14 + 16 * 11
# 1677216 + 2097152 + 196608 + 16384 + 1280 + 224 + 11
# = 19088875
#
# 2. The script has the following functions:
#      o main
#      o DefineVariables
#      o PerformConversion
#
# 3. When using the command in interactive mode,
#      usage: htod.sh <hex number>
#
# 4. When using the command in batch mode ,
#      usage: dec_num='htod.sh <hex number> 2>/dev/null'
#      check the return code $?
#####
#####
```

```

# Name      : DefineVariables
# Overview  : The function defined all variables.
# Notes    :
#####
DefineVariables ()
{
#
# message
#
USAGE="Usage:htod.sh <hex number>"
INVALID_HEX_NUMBER="htod.sh:ERROR:Invalid number"
#
DIGIT_POSITION=0      # position of digit being evaluated
integer HEX_LENGTH=0  # length of input hex number
HEX_NUMBER=""         # unsigned hex number
DEC_NUM=0             # converted decimal number
DEC_EQUIVALENT=0     # decimal equivalent of a specific hex digit
PLACE_VALUE=0        # place value of digit being evaluated
THIS_DIGIT=""        # digit being evaluated
SIGN=""              # plus or minus sign
SUCCESS=0
FAILURE=1
TRUE=0
FALSE=1

}
#####
# Name      : PerformConversion
# Overview  : The function performs the number conversion.
# Notes    :
#####
PerformConversion ( )
{
#
# establish length of hex number
#
HEX_LENGTH='echo "${P_HEX_NUM}\c" | wc -c'
#
# establish signed or unsigned number
#
SIGN='echo "${P_HEX_NUM}" | cut -c1-1'
if [ "${SIGN}" = "+" -o "${SIGN}" = "-" ]
then
    if [ $HEX_LENGTH -eq 1 ]
    then
        echo "${INVALID_HEX_NUMBER}" 1>&2
        return $FALSE
    fi
#
HEX_NUM='echo "${P_HEX_NUM}" | cut -c2-${HEX_LENGTH}'

```

```

        HEX_LENGTH='expr ${HEX_LENGTH} - 1'
else
    SIGN=""
    HEX_NUM="${P_HEX_NUM}"
fi
#
# set the starting digit position as the rightmost digit
#
DIGIT_POSITION=${HEX_LENGTH}
#
while true
do
    THIS_DIGIT='echo "${HEX_NUM}" | cut -c ${DIGIT_POSITION}'
    if [ "${THIS_DIGIT}" = "A" -o "${THIS_DIGIT}" = "a" ]
    then
        DEC_EQUIVALENT='bc <<!'
        ( ( 16 ^ ${PLACE_VALUE} ) * 10 )
    !'
    elif [ "${THIS_DIGIT}" = "B" -o "${THIS_DIGIT}" = "b" ]
    then
        DEC_EQUIVALENT='bc <<!'
        ( ( 16 ^ ${PLACE_VALUE} ) * 11 )
    !'
    elif [ "${THIS_DIGIT}" = "C" -o "${THIS_DIGIT}" = "c" ]
    then
        DEC_EQUIVALENT='bc <<!'
        ( ( 16 ^ ${PLACE_VALUE} ) * 12 )
    !'
    elif [ "${THIS_DIGIT}" = "D" -o "${THIS_DIGIT}" = "d" ]
    then
        DEC_EQUIVALENT='bc <<!'
        ( ( 16 ^ ${PLACE_VALUE} ) * 13 )
    !'
    elif [ "${THIS_DIGIT}" = "E" -o "${THIS_DIGIT}" = "e" ]
    then
        DEC_EQUIVALENT='bc <<!'
        ( ( 16 ^ ${PLACE_VALUE} ) * 14 )
    !'
    elif [ "${THIS_DIGIT}" = "F" -o "${THIS_DIGIT}" = "f" ]
    then
        DEC_EQUIVALENT='bc <<!'
        ( ( 16 ^ ${PLACE_VALUE} ) * 15 )
    !'

    elif [ "${THIS_DIGIT}" != "0" -a "${THIS_DIGIT}" != "1" -a \
          "${THIS_DIGIT}" != "2" -a "${THIS_DIGIT}" != "3" -a \
          "${THIS_DIGIT}" != "4" -a "${THIS_DIGIT}" != "5" -a \
          "${THIS_DIGIT}" != "6" -a "${THIS_DIGIT}" != "7" -a \
          "${THIS_DIGIT}" != "8" -a "${THIS_DIGIT}" != "9" ]
    then

```



```

        echo "${INVALID_HEX_NUMBER}" 1>&2
        return $FALSE
    else
        DEC_EQUIVALENT='bc <<!(
            ( ( 16 ^ ${PLACE_VALUE} ) * ${THIS_DIGIT} )
        !'
        fi
        DEC_NUM='bc <<!(
            ${DEC_NUM} + ${DEC_EQUIVALENT}
        !'
        #
        # increment place_value by 1
        #
        PLACE_VALUE='expr ${PLACE_VALUE} + 1'
        #
        # decrement $DIGIT_POSITION by 1
        #
        DIGIT_POSITION='expr ${DIGIT_POSITION} - 1'
        if [ ${DIGIT_POSITION} -eq 0 ]
        then
            # all the digits in the number have been evaluated
            break
        fi
    done
    #
    return $TRUE
}
#####
# Name      : main
# Overview  : The function implements processing structure.
# Notes     :
#####
main ()
{
    #
    DefineVariables
    #
    if [ "${P_HEX_NUM}" = "" ]
    then
        echo "${USAGE}" 1>&2
        exit $FAILURE
    fi
    #
    if ! PerformConversion
    then
        exit $FAILURE
    else
        echo "${SIGN}${DEC_NUM}"
        exit $SUCCESS
    fi
}

```

```

}
#
# assign the parameter
#
P_HEX_NUM="$1"
main
#

```

## LISTING OF DTOD.SH

```

#####
# Name      : dtod.sh(decimal to hexadecimal )
#
# Overview  : The script converts a decimal number to a hexadecimal
#             number.
#
# Input     : signed/unsigned  Decimal Number
#
# Notes     : 1. The algorithm for conversion is as follows:
#
#             A = 10  B = 11  C = 12  D = 13  E = 14  F = 15
#
# number to convert -> 109
#                 109 /16    = 6 Remainder 13 (D)
#                 6  /16    = 0 Remainder 6
#                 converted hex number -> 6D
#
#           2. The script has the following functions:
#                o main
#                o DefineVariables
#                o PerformConversion
#
#           3. When using the command in interactive mode ,
#                usage: dtod.sh <dec number>
#
#           4. When using the command in batch mode ,
#                usage: hex_num='dtod.sh <dec number> 2>/dev/null'
#                check the return code $?
#####
#####
# Name      : DefineVariables
# Overview  : The function defined all variables.
# Notes     :
#####
DefineVariables ()
{
#
# message

```

```

#
USAGE="Usage:dectohex.sh <decimal number>"
INVALID_DEC_NUMBER="dtoh.sh:ERROR:Invalid number"
#
DIGIT_POSITION=0           # position of digit being evaluated
integer DEC_LENGTH=0      # length of input decimal number
DEC_NUM=0                 # unsigned hex number
HEX_NUM=""               # converted hex number
DEC_EQUIVALENT=0         # decimal equivalent of a specific hex digit
PLACE_VALUE=0            # place value of digit being evaluated
THIS_DIGIT=""           # digit being evaluated
SIGN=""                  # plus or minus sign
SUCCESS=0
FAILURE=1
TRUE=0
FALSE=1
}
#####
# Name      : PerformConversion
# Overview  : The function performs the number conversion.
# Notes    :
#####
PerformConversion ( )
{
#
# establish length of hex number
#
DEC_LENGTH='echo "${P_DEC_NUM}\c" | wc -c'
#
# establish signed or unsigned number
#
SIGN='echo "${P_DEC_NUM}" | cut -c1-1'
if [ "${SIGN}" = "+" -o "${SIGN}" = "-" ]
then
    if [ $DEC_LENGTH -eq 1 ]
    then
        echo "${INVALID_DEC_NUMBER}" 1>&2
        return $FALSE
    fi
#
    DEC_NUM='echo "${P_DEC_NUM}" | cut -c2-${DEC_LENGTH}'
    DEC_LENGTH='expr ${DEC_LENGTH} - 1'
else
    SIGN=""
    DEC_NUM="${P_DEC_NUM}"
fi
#
# check all the digits in the number
#
INDEX=${DEC_LENGTH}

```

```

while true
do
    THIS_DIGIT='echo ${DEC_NUM} | cut -c${INDEX}'
    if [ "${THIS_DIGIT}" != "0" -a "${THIS_DIGIT}" != "1" -a \
        "${THIS_DIGIT}" != "2" -a "${THIS_DIGIT}" != "3" -a \
        "${THIS_DIGIT}" != "4" -a "${THIS_DIGIT}" != "5" -a \
        "${THIS_DIGIT}" != "6" -a "${THIS_DIGIT}" != "7" -a \
        "${THIS_DIGIT}" != "8" -a "${THIS_DIGIT}" != "9" ]
    then
        echo "${INVALID_DEC_NUMBER}" 1>&2
        return $FALSE
    fi
    #
    INDEX='expr $INDEX - 1'
    if [ $INDEX -eq 0 ]
    then
        break
    fi
done
#
# convert to hex
#
while true
do
    REMAINDER='bc <<!
    ( ${DEC_NUM} % 16 )
    !'
    if [ $REMAINDER -eq 10 ]
    then
        HEX_NUM="A${HEX_NUM}"
    elif [ $REMAINDER -eq 11 ]
    then
        HEX_NUM="B${HEX_NUM}"
    elif [ $REMAINDER -eq 12 ]
    then
        HEX_NUM="C${HEX_NUM}"
    elif [ $REMAINDER -eq 13 ]
    then
        HEX_NUM="D${HEX_NUM}"
    elif [ $REMAINDER -eq 14 ]
    then
        HEX_NUM="E${HEX_NUM}"
    elif [ $REMAINDER -eq 15 ]
    then
        HEX_NUM="F${HEX_NUM}"

```

```

else
    HEX_NUM="${REMAINDER}${HEX_NUM}"
fi
DEC_NUM='bc <<!
( ( ${DEC_NUM} - ${REMAINDER}) / 16 )
!'
if [ $DEC_NUM -eq 0 ]
then
    break
fi
done
#
return $TRUE
}
#####
# Name      : main
# Overview  : The function implements processing structure.
# Notes     :
#####
main ()
{
#
DefineVariables
#
if [ "${P_DEC_NUM}" = "" ]
then
    echo "${USAGE}" 1>&2
    exit $FAILURE
fi
#
if ! PerformConversion
then
    exit $FAILURE
else
    echo "${SIGN}${HEX_NUM}"
    exit $SUCCESS
fi
}
#
# assign the parameter
#
P_DEC_NUM="$1"
main
#

```

---

*Arif Zaman*  
Analyst/Programmer  
High-Tech Software (UK)

© Xephon 2001

---

# Tape Manager

## INTRODUCTION

The most difficult part of back-up operations is to manage tape cartridges. Operators usually need to copy files to cartridges and save them for a while. So they should stick a label on each cartridge to identify when the back-up was taken, how long it is to be retained, and which files reside on it. When a file restore is needed, the operator should find the right cartridge immediately. If the back-up process involves a lot of files and cartridges, management of cartridges becomes very difficult and error-prone. Besides, more cartridges are used than actually needed.

With Tape Manager, it is enough to give a retention period and logical name to a file or file group to be backed up. Tape Manager will request a cartridge number, copy the file(s), and keep the cartridge unless the retention period is exceeded. The same cartridge can be used for more than one file if they will expire on the same day and free space is available on the cartridge. If all files on a cartridge have expired, it is re-used by **tmwrite** as a new one. Tape Manager keeps a record of how many times a cartridge is used. By looking at this number, we can decide not to use old and frequently used cartridges.

By default, a logical name cannot be used for more than one file. If we want to use the same logical name for periodic back-ups, we must define a version and give a **-v1** option for **tmwrite**. Then, we can restore the files by giving the **-v** option. For example, if we want to restore the last back-up, we should use **-v0**. For a previous back-up, we should use **-v-1**, and so on.

Each cartridge is initialized for Tape Manager's use and given a number by using **tminit**. A label specifying the number should be stuck on each cartridge. **Tminit** copies a file containing the label number as the first file on the cartridge. Tape manager keeps records about files on all the cartridges in DB2 tables. It knows which files are on which cartridge and how long they have been retained.

Tape Manager can be used by many Unix servers concurrently. They

all access the same database by cataloguing the main database. Each Unix server uses its own tape units by specifying the **-g** option of the **tmwrite**, **tmread**, and **tminit** commands. The information for all tape units is kept in a table, `TAPE_DEVICES`. Each Unix server may have many tape units. Tape Manager will request an available tape unit according to the **-g** option.

Different Unix servers may use the same cartridge pool. We can also use different cartridge pools by specifying the **-p** option of the command. This allows us to use different types of cartridge in the system. For example one cartridge pool consists of 4mm cartridges while another has 8mm cartridges. This feature can also be used for cartridges in different locations. We can define two cartridge pools for two different locations. These definitions are kept in the table, `TAPE_POOLS`.

Tape Manager can be used concurrently by many back-up processes. Each process will check the cartridges and tape units and use them if they are not used by other processes. The maximum number of processes running at a time is the number of tape units.

## USAGE OF PROGRAMS

**Tminit** – initializes cartridges for Tape Manager use:

```
tminit -n number_of_new_carts -p pool_name -g group_name
```

where:

- **-n** – number of cartridges to be initialized.
- **-p** – pool name.
- **-g** – group name.

**Tmread** – reads files from cartridges by using Tape Manager.

```
tmread -f filename -v version -s -c -k
```

where:

- **-f** – file name to be restored.
- **-v** – version ( may be 0,-1, -2,... ).
- **-s** – do not rewind tape (stay at the current location).

- -c – do not look at the tape label. Continue even if labels do not match.
- -g – group name.
- -k – key name.

Tmwrite – writes files to cartridges by using Tape Manager:

```
tmwrite -f filename -v version -r retention period -n -x -p
pool_name -c -s -k key_name files_to_be_copied
```

where:

- -f – file name to be copied.
- -n – use a new cartridge. If -n is not coded an old cartridge that best matches with this retention period will be used.
- -x – do not use this cartridge for other files (exclusive).
- -v – which version of the file after which we copy.
- -r – retention period.
- -p – pool name. If not entered, it is set to the default.
- -s – do not rewind tape (stay at the current location).
- -c – do not look at the tape label. Continue even if labels do not match.
- -g – group name.
- -k – key name.

Tmllist – lists the files archived with Tape Manager:

```
tmllist -f filename -s serno -r
```

where:

- -r – lists all files archived in Tape Manager in the order of expiration date.
- -f – lists the location and version information for a given file name in Tape Manager.



- -s – lists all files in a given cartridge.

Tmver – defines a filename that is controlled by its version:

```
tmver -c version -d -f filename
```

where:

- -c – create version definition. Version is the maximum version to be kept.
- -d – drop version definition.

Tmscratch – runs periodically and scratches the cartridges on which all files are expired.

## TABLES USED BY TAPE MANAGER

### CURRENT\_CART

KEY CHAR(20)	PK
SERIAL_NO dec(10)	
DEV_NAME CHAR(20)	
DEV_GROUP dec(3)	

### TAPE\_DEVICES

DEV_NAME CHAR(20)	PK
BUSY dec(1)	
CAPACITY_IN_KB dec(10)	
DEV_GROUP dec(3)	PK
DEVICE_LABEL CHAR(10)	
DEVICE_TEXT VARCHAR(200)	

### TAPE\_FILES

FILE_NAME VARCHAR(100)	PK
FILES_TO_BE_COPIED VARCHAR(1000)	
VERSION DEC(4)	PK
SERIAL_NO DEC(10)	
SEQ_NO DEC(5)	
CRE_DATE TIMESTAMP	
RET_PD DEC(5)	
IS_SHARE CHAR(1)	
SIZE_IN_KB DEC(15)	

### TAPE\_POOLS

```
POOL_NO DEC(3) PK
POOL_NAME CHAR(10)
FIRST_SER DEC(10)
LAST_SER DEC(10)
```

#### TAPE\_SERIALS

```
SERIAL_NO DEC(10) PK
INIT_DATE TIMESTAMP
IN_USE CHAR(1)
USE_COUNT DEC(8)
LAST_USED_DATE TIMESTAMP
DEV_NAME CHAR(20)
DEV_GROUP DEC(3)
```

#### VERSIONS

```
FILE_NAME VARCHAR(100) PK
VERSION_MAX DEC(5)
CURRENT_VER DEC(5)
```

### TMINIT.SQC

```
#include <stdio.h>
#include <sqlca.h>
#include <unistd.h>
#include <string.h>
#include <time.h>

int random_number;

EXEC SQL INCLUDE SQLCA;
EXEC SQL BEGIN DECLARE SECTION ;
    char v_pool_name[11];
    char v_dev_name[21];
    short new_serial_no=0;
    char v_dev_label[11];
    short dummy_var;
    short nullind;
    short dev_group_num=0;
EXEC SQL END DECLARE SECTION;

void release_device();

void sql_error(int err_pl)
{
    printf("SQLCODE=%ld\n",sqlca.sqlcode);
    printf("Place of the error=%d\n",err_pl);
}
```

```

    exec sql rollback;
    release_device();
    exit(1);
}

void release_device()
{
    char sys_cmd[1000];
    int sys_rc;

    exec sql update TAPE_DEVICES set BUSY=0
        where DEV_NAME=:v_dev_name and
            DEV_GROUP=:dev_group_num;
    exec sql commit;

    sprintf(sys_cmd,"rm -r %ld",random_number);
    sys_rc=system(sys_cmd);
    if (sys_rc!=0) {
        printf("sys_rc=%d\n",sys_rc);
        exit(2);
    }
}

void get_random_number()
{
    struct timeval tv;
    long msec;
    gettimeofday(&tv, NULL);
    msec=tv.tv_usec;
    srand(msec);
    random_number=rand();
}

int main(int argc,char **argv)
{
    char dummy;
    int ind1;
    int dev_group_num=0;
    char sys_cmd[200];
    int sys_rc;
    FILE *label_file;
    int rc,lp1;

    int copt,i=0;
    int new_cart_num=0;

    while(1) {
        copt=getopt(argc,argv,"n:p:g:");
        if (copt==-1) break;
        switch(copt) {

```

```

    case 'n' :
        new_cart_num=atoi(optarg);
        break;
    case 'p' :
        strcpy(v_pool_name,optarg);
        break;
    case 'g' :
        dev_group_num=atoi(optarg);
        break;
    otherwise :
        exit(3);
        break;
}
}

if (strcmp(v_pool_name,"")==0) strcpy(v_pool_name,"default");
if (new_cart_num==0) exit(4);

exec sql connect to tmdb;
if ( sqlca.sqlcode < 0 ) sql_error(1);

exec sql declare crs1 cursor for
    select dev_name
    from TAPE_DEVICES
    where BUSY=0 and DEV_GROUP=:dev_group_num and
           DEV_NAME||char(DEV_GROUP)
           not in ( select DEV_NAME||char(DEV_GROUP)
                   from CURRENT_CART )
    for update of BUSY;
if ( sqlca.sqlcode < 0 ) sql_error(2);

exec sql open crs1;
if ( sqlca.sqlcode < 0 ) sql_error(3);

exec sql fetch crs1 into :v_dev_name;
if ( sqlca.sqlcode < 0 ) sql_error(4);

if ( sqlca.sqlcode==100 ) {
    printf("Could not find any available device...\n");
    exit(5);
}

exec sql update TAPE_DEVICES set BUSY=1
    where current of crs1;
if ( sqlca.sqlcode < 0 ) sql_error(5);

exec sql close crs1;
if ( sqlca.sqlcode < 0 ) sql_error(6);

exec sql commit;

```

```

if ( sqlca.sqlcode < 0 ) sql_error(7);
printf("Please use device %s ...\n",v_dev_label);

get_random_number();

sprintf(sys_cmd,"mkdir %ld",random_number );
sys_rc=system(sys_cmd);
if (sys_rc!=0) {
    printf("sys_rc=%d\n",sys_rc);
    exit(7);
}

sprintf(sys_cmd,"%ld",random_number );
chdir(sys_cmd);

exec sql select POOL_NO into :dummy_var
        from TAPE_POOLS
        where POOL_NAME =:v_pool_name;
if ( sqlca.sqlcode < 0 ) sql_error(8);
if ( sqlca.sqlcode == 100 ) {
    printf("Could not find pool %s .\n",v_pool_name);
    release_device();
    exit(6);
}

for ( ind1=0;ind1<new_cart_num;ind1++) {

    exec sql select max(SERIAL_NO)+1
        into :new_serial_no:nullind from TAPE_SERIALS
        where SERIAL_NO >= ( select FIRST_SER from TAPE_POOLS
                            where POOL_NAME =:v_pool_name) and
        SERIAL_NO <= ( select LAST_SER from TAPE_POOLS
                            where POOL_NAME =:v_pool_name );

    if ( sqlca.sqlcode < 0 ) sql_error(9);
    if (nullind==-1) {
        exec sql select FIRST_SER into :new_serial_no:nullind
            from TAPE_POOLS
            where POOL_NAME =:v_pool_name;
        if ( sqlca.sqlcode <0 ) sql_error(10);
    }

    printf("Please insert cartridge no. %ld (Press x for
exit)\n",new_serial_no);
    dummy=getchar();

    if (dummy=='x') {
        chdir("../");
        release_device();
        exit(8);
    }
}

```

```

}

label_file=fopen("./label_file","w");
fprintf(label_file,"%ld\n",new_serial_no);
fclose(label_file);

while(1) {
    sprintf(sys_cmd,"mt -f %s rewind",v_dev_name);
    rc=system(sys_cmd);
    if (rc!=0) {
        printf("Tape is not ready. rc=%d\n",rc);
        printf("Press any key to continue. (Press x for exit)\n");
        dummy=getchar();
        if (dummy=='x') {
            chdir("..");
            release_device();
            exit(9);
        }
    } else {
        sprintf(sys_cmd,"tar -cf %s ./label_file",v_dev_name);
        rc=system(sys_cmd);
        if (rc!=0) {
            printf("Tape is not ready. rc=%d\n",rc);
            printf("Press any key to continue. (Press x for exit)\n");
            dummy=getchar();
            if (dummy=='x') {
                chdir("..");
                release_device();
                exit(10);
            }
        }
        else break;
    }
}

exec sql insert into TAPE_SERIALS
    values(:new_serial_no,current timestamp,'Y',0,current
timestamp,'',0);
    if ( sqlca.sqlcode < 0 ) sql_error(11);

    printf("Cartridge numbered %d is successfully
initialized...\n",new_serial_no);
}

chdir("..");
release_device();

exit(0);
}

```

## TMVER.SQC

```
#include <stdio.h>
#include <sqlca.h>
#include <unistd.h>
#include <string.h>

EXEC SQL INCLUDE SQLCA;
EXEC SQL BEGIN DECLARE SECTION ;
    char file_name[100];
    short version=0;
EXEC SQL END DECLARE SECTION;

void sql_error(int err_pl)
{
    printf("SQLCODE=%ld\n",sqlca.sqlcode);
    printf("Place of the error=%d\n",err_pl);
    exec sql rollback;
    exit(1);
}

int main(int argc,char **argv)
{
    int copt,i=0;
    char operation_flag=' ';

    while(1) {
        copt=getopt(argc,argv,"c:df:");
        if (copt==-1) break;
        switch(copt) {
            case 'c' :
                operation_flag='c';
                version=atoi(optarg);
                break;
            case 'd' :
                operation_flag='d';
                break;
            case 'f' :
                strcpy(file_name,optarg);
                break;
            otherwise :
                exit(2);
                break;
        }
    }

    if ( strcmp(file_name,"")==0) {
```

```

    printf("File name must be entered...\n\n");
    exit(3);
}

if ( version == 0 & operation_flag == 'c' ) {
    printf("Maximum version must be entered...\n\n");
    exit(4);
}

exec sql connect to tmdb;
if ( sqlca.sqlcode < 0 ) sql_error(1);

switch ( operation_flag ) {
    case 'd' :
        exec sql delete from VERSIONS
            where FILE_NAME=:file_name;
        if ( sqlca.sqlcode < 0 ) sql_error(2);
        if ( sqlca.sqlcode==100) {
            printf("Could not find such a version...\n");
            break;
        }
        printf("Version dropped... Filename=%s\n",file_name);
        break;
    case 'c' :
        exec sql insert into VERSIONS
            values(:file_name,:version,0);
        if ( sqlca.sqlcode==-803) {
            printf("Version already
created...Filename=%s\n",file_name);
            exit(5);
        }
        else if ( sqlca.sqlcode<0 ) sql_error(3);
        printf("Version created... Filename=%s\n",file_name);
        break;
}

exec sql commit;
if ( sqlca.sqlcode < 0 ) sql_error(4);

exit(0);
}

```

## TMWRITE.SQC

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sqlenv.h>

```



```

int new_flag=0;
int continue_flag=0;
char v_dev_name_n[22];
int random_number;

EXEC SQL INCLUDE SQLCA;
EXEC SQL BEGIN DECLARE SECTION ;
    short nullind;
    short new_serial_no;
    short seq=0;
    short retpd;
    char file_name[100];
    char files_to_be_copied[1000];
    short version=0;
    short dev_group_num=0;
    char share_flag[2]="Y";
    short stay_flag=0;
    char v_dev_name[21];
    char v_dev_label[11];
    char v_last_used_date[20];
    long tot_size;
    long v_capacity_in_kb;
    long v_size_in_kb;
    char continue_key[20];
    char pool_name[50]="";
    short dummy_var;
    short current_version=0;
    char dummy[21];
EXEC SQL END DECLARE SECTION;

void release_device(int opt);

void sql_error(int err_pl)
{
    printf("SQLCODE=%ld\n",sqlca.sqlcode);
    printf("Place of the error=%d\n",err_pl);
    exec sql rollback;
    release_device(0);
    exit(1);
}

void release_device(int opt)
{
    char sys_cmd[1000];
    int sys_rc;

    exec sql update TAPE_DEVICES set BUSY=0
        where DEV_NAME=:v_dev_name and
            DEV_GROUP=:dev_group_num;

```

```

exec sql update TAPE_SERIALS set IN_USE='Y'
      where SERIAL_NO=:new_serial_no;

if ( opt == 0 ) {
    exec sql delete from CURRENT_CART
      where KEY=:continue_key;
}

exec sql commit;

sprintf(sys_cmd,"rm -r %ld",random_number);
sys_rc=system(sys_cmd);
if (sys_rc!=0) {
    printf("sys_rc=%d\n",sys_rc);
    exit(2);
}
}

void get_random_number()
{
    struct timeval tv;
    long msec;
    gettimeofday(&tv, NULL);
    msec=tv.tv_usec;
    srand(msec);
    random_number=rand();
}

int get_file_size()
{
    FILE *fp1;
    char sys_cmd[1000];
    char files_copied_temp[1000] = "";
    char file_out[50];
    char *file_dummy = " > /dev/null 2>&1 ";
    int rc;
    long size_of_files;
    char temp_file_name[100];
    char dummy_str[1000];

    strcpy(files_copied_temp,files_to_be_copied);
    memcpy(files_copied_temp+strlen(files_copied_temp)-
1,file_dummy,strlen(file_dummy));

    sprintf(sys_cmd,"ls %s ",files_copied_temp);
    if ( rc=system(sys_cmd) != 0 ) {
        printf("File %s could not be found...\n",files_copied_temp);
        exit(3);
    }
}

```

```

strcpy(files_copied_temp,files_to_be_copied);
sprintf(file_out,"> ./%ld/size_of_files",random_number);
memcpy(files_copied_temp+strlen(files_copied_temp)-
1,file_out,strlen(file_out));

sprintf(sys_cmd,"du -k %s ",files_copied_temp);

if ( rc=system(sys_cmd) == -1 ) {
    printf("error in du...\n");
    exit(4);
}

sprintf(temp_file_name,"./%ld/size_of_files",random_number);
if ( ( fp1=fopen(temp_file_name,"r") ) == NULL ) {
    printf("There is a problem while reading file size_of_files...\n");
    exit(5);
}

while ( fscanf(fp1,"%d%s",&size_of_files,&dummy_str)!=-1)
    tot_size+=size_of_files;

fclose(fp1);
}

int give_scratch()
{
    /* give a scratch cart no */
    if (new_flag==1) {
        exec sql declare scrtc_crs1 cursor for
            select SERIAL_NO, LAST_USED_DATE
            from TAPE_SERIALS
            where SERIAL_NO not in ( select SERIAL_NO from TAPE_FILES ) and
            SERIAL_NO not in ( select SERIAL_NO from CURRENT_CART ) and
            SERIAL_NO >= ( select FIRST_SER from TAPE_POOLS
                where POOL_NAME=:pool_name ) and
            SERIAL_NO <= ( select LAST_SER from TAPE_POOLS
                where POOL_NAME=:pool_name ) and
            IN_USE='Y'
            for update of IN_USE ;
        exec sql open scrtc_crs1;

        if ( sqlca.sqlcode < 0 ) sql_error(1);
        exec sql fetch scrtc_crs1
            into :new_serial_no:nullind,:v_last_used_date:nullind;
        if ( sqlca.sqlcode < 0 ) sql_error(2);
        if ( nullind==-1 ) {
            printf("There is no new cartridge...\n");
            release_device(0);
            exit(6);
        }
    }
}

```

```

}
exec sql update TAPE_SERIALS set IN_USE='C'
      where current of scrtc_crs1;
if ( sqlca.sqlcode < 0 ) sql_error(3);
exec sql close scrtc_crs1;
}
else {

exec sql select min(SERIAL_NO) into :new_serial_no:nullind
      from (
select a.SERIAL_NO,max(date(CRE_DATE)+RET_PD days) as EXP_DATE,
      sum(SIZE_IN_KB)
      from TAPE_FILES a ,TAPE_SERIALS b
      where a.SERIAL_NO not in ( select SERIAL_NO
                                from tape_files
                                where IS_SHARE='N' ) and
      a.SERIAL_NO not in ( select SERIAL_NO from CURRENT_CART ) and
      a.SERIAL_NO >= ( select FIRST_SER from TAPE_POOLS
                       where POOL_NAME=:pool_name ) and
      a.SERIAL_NO <= ( select LAST_SER from TAPE_POOLS
                       where POOL_NAME=:pool_name ) and
      RET_PD<>0 and IN_USE='Y' and
      a.SERIAL_NO=b.SERIAL_NO
      group by a.SERIAL_NO
      having sum(size_in_kb) < :v_capacity_in_kb - :tot_size ) table1
      where EXP_DATE=current date+:retpd days;

if (nullind==-1)
{
exec sql declare scrtc_crs2 cursor for
      select SERIAL_NO,LAST_USED_DATE
      from TAPE_SERIALS
      where SERIAL_NO not in ( select SERIAL_NO from TAPE_FILES ) and
      SERIAL_NO not in ( select SERIAL_NO from CURRENT_CART ) and
      SERIAL_NO >= ( select FIRST_SER from TAPE_POOLS
                    where POOL_NAME=:pool_name ) and
      SERIAL_NO <= ( select LAST_SER from TAPE_POOLS
                    where POOL_NAME=:pool_name ) and
      IN_USE='Y'
      for update of IN_USE ;

exec sql open scrtc_crs2;

exec sql fetch scrtc_crs2
      into :new_serial_no:nullind,:v_last_used_date:nullind;
if ( sqlca.sqlcode < 0 ) sql_error(4);
if (nullind==-1) {
printf("Could not find any suitable cartridge...\n");
release_device(0);
exit(7);
}
}
}

```

```

    }
    exec sql update TAPE_SERIALS set IN_USE='C'
        where current of scrtc_crs2;
    if ( sqlca.sqlcode < 0 ) sql_error(5);
    exec sql close scrtc_crs2;
} else
{
    exec sql update TAPE_SERIALS set IN_USE='C'
        where SERIAL_NO=:new_serial_no and
            IN_USE='Y';
    if ( sqlca.sqlcode == 100 ) {
        return(give_scratch());
    }
    if ( sqlca.sqlcode < 0 ) sql_error(6);
}
}

exec sql commit;
if ( sqlca.sqlcode < 0 ) sql_error(7);

return(new_serial_no);
}

void check_parameters()
{

    if ( strcmp(file_name,"")==0) {
        printf("Filename must be entered...\n\n");
        exit(8);
    }

    exec sql select distinct(1) into :dummy_var
        from TAPE_FILES
        where FILE_NAME=:file_name;
    if ( sqlca.sqlcode < 0 ) sql_error(8);
    if ( sqlca.sqlcode != 100 && version==0 ) {
        printf("File name %30s is already used...\n",file_name);
        printf("Please use another name...\n");
        exit(9);
    }

    if ( retpd==0 && version==0) {
        printf("One of the Retention period and version must be given
.\n\n");
        exit(10);
    }

    if ( version<1 && version!=0 ) {
        printf("Version must be equal to 0 or greater than 0...\n");
        exit(11);
    }
}

```

```

}

if ( strcmp(continue_key,"")==0 && ( continue_flag == 1 || stay_flag
== 1 )) {
    printf("-k option must be used with -c or -s option...\n");
    exit(12);
}

if ( strcmp(continue_key,"")==0) strcpy(continue_key,"DUMMY");

if (strcmp(pool_name,"")==0) strcpy(pool_name,"default");

}

void get_cart()
{
    FILE *label_file;
    int sys_rc;
    int carts_serial_no;
    char sys_cmd[1000];
    char dummy_char;

    seq=0;

    new_serial_no=give_scratch();

    sprintf(sys_cmd,"%ld",random_number );
    chdir(sys_cmd);

    while ( 1 ) {
        printf("Please insert cartridge number %d and press
ENTER...\n",new_serial_no);
        printf("Press x to quit...\n");
        fflush(stdout);
        dummy_char=getchar();
        if (dummy_char=='x') {
            chdir("..");
            release_device(0);
            exit(13);
        }

        printf("The cartridge is being rewinded...\n");
        fflush(stdout);
        sprintf(sys_cmd,"%s%s%s","mt -f ",v_dev_name," rewind >/dev/null
2>&1");
        sys_rc=system(sys_cmd);
        if (sys_rc!=0) {
            printf("sys_rc=%d\n",sys_rc);
            chdir("..");
            release_device(0);

```

```

    exit(14);
}

sprintf(sys_cmd,"%s%s%s","tar -xf ",v_dev_name," >/dev/null 2>&1");

printf("Label file is being read...\n");
fflush(stdout);
sys_rc=system(sys_cmd);
if (sys_rc!=0) {
    printf("sys_rc=%d\n",sys_rc);
    chdir("..");
    release_device(0);
    exit(15);
}

if ( ( label_file=fopen("./label_file","r") ) == NULL ) {
    printf("There is a problem while reading ./label_file...\n");
    chdir("..");
    release_device(0);
    exit(16);
}
fscanf(label_file,"%d",&carts_serial_no);
fclose(label_file);

if ( carts_serial_no == new_serial_no ) {
    break;
}
printf("Wrong cartridge is inserted...\n");
}

exec sql update TAPE_SERIALS set LAST_USED_DATE=current timestamp,
                                USE_COUNT=USE_COUNT+1,
                                DEV_NAME=:v_dev_name,
                                DEV_GROUP=:dev_group_num
                                where SERIAL_NO=:new_serial_no;

if ( sqlca.sqlcode < 0 ) {
    chdir("..");
    sql_error(9);
}

exec sql select max(SEQ_NO) into :seq:nullind
            from TAPE_FILES
            where serial_no=:new_serial_no;
if ( sqlca.sqlcode < 0 ) {
    chdir("..");
    sql_error(10);
}

if ( nullind== -1) seq=0;

```

```

    if ( seq > 0 ) {
        sprintf(sys_cmd,"mt -f %s fsf %d >/dev/null
2>&1",v_dev_name_n,seq+1);
        printf("Winding the first empty point of the cartridge...\n");
        fflush(stdout);
        system(sys_cmd);
        if (sys_rc!=0) {
            printf("sys_rc=%d\n",sys_rc);
            chdir("..");
            release_device(0);
            exit(17);
        }
    } else
    {
        sprintf(sys_cmd,"%s%s%s","tar -cf ",v_dev_name_n," ./label_file >/
dev/null 2>&1");
        sys_rc=system(sys_cmd);
        if (sys_rc!=0) {
            printf("sys_rc=%d\n",sys_rc);
            chdir("..");
            release_device(0);
            exit(18);
        }
    }
    chdir("..");
}

void norew_dev_name()
{
    int lp1;

    strcpy(v_dev_name_n,v_dev_name);
    for ( lp1=0;lp1<strlen(v_dev_name_n);lp1++) {
        if ( v_dev_name_n[lp1]==' ' ) {
            v_dev_name_n[lp1]='.';
            v_dev_name_n[lp1+1]='1';
            v_dev_name_n[lp1+2]='\0';
            break;
        }
    }
}

void tm_write()
{
    FILE *label_file;
    int sys_rc;

    char sys_cmd[1000];

```



```

get_random_number();

sprintf(sys_cmd,"mkdir %ld",random_number );
sys_rc=system(sys_cmd);
if (sys_rc!=0) {
    printf("sys_rc=%d\n",sys_rc);
    exit(19);
}

get_file_size();

if ( continue_flag == 0 ) {
    exec sql declare crs1 cursor for
        select dev_name,capacity_in_kb
        from TAPE_DEVICES
        where BUSY=0 and DEV_GROUP=:dev_group_num and
            DEV_NAME||char(DEV_GROUP)
            not in ( select DEV_NAME||char(DEV_GROUP)
                    from CURRENT_CART )
        for update of BUSY ;
    if ( sqlca.sqlcode < 0 ) sql_error(11);

    exec sql open crs1;

    if ( sqlca.sqlcode < 0 ) sql_error(12);

    exec sql fetch crs1 into :v_dev_name,:v_capacity_in_kb;
    if ( sqlca.sqlcode < 0 ) sql_error(13);

    if ( sqlca.sqlcode==100 ) {

        printf("Could not find any available device...\n");
        release_device(0);
        exit(20);
    }

    while ( sqlca.sqlcode!=100 )
    {
        if ( v_capacity_in_kb > tot_size ) break;
        printf("%s device do not have enough capacity for this
file...\n",v_dev_name);
        printf("Either use another device or split the file...\n");
        exec sql fetch crs1 into :v_dev_name,:v_capacity_in_kb;
        if ( sqlca.sqlcode < 0 ) sql_error(14);
    }

    if ( sqlca.sqlcode==100 ) {
        printf("Could not find any available device...\n");

```

```

        release_device(0);
        exit(21);
    }

    exec sql update TAPE_DEVICES set BUSY=1
           where current of crs1;
    if ( sqlca.sqlcode < 0 ) sql_error(15);
    exec sql close crs1;

    exec sql select DEVICE_LABEL into :v_dev_label
           from TAPE_DEVICES
           where DEV_NAME=:v_dev_name and
                 DEV_GROUP=:dev_group_num;
    if ( sqlca.sqlcode < 0 ) sql_error(16);

    printf("Please use device %s ...\n",v_dev_label);

    norew_dev_name();
    get_cart();

} else {
    exec sql select SERIAL_NO,DEV_NAME,DEV_GROUP
           into :new_serial_no,:v_dev_name,:dev_group_num
           from CURRENT_CART
           where KEY=:continue_key;
    if ( sqlca.sqlcode < 0 ) sql_error(17);
    if (sqlca.sqlcode==1000) {
        new_serial_no=0;
        printf("You did not use -s flag in the previous operation, you can
not use -c flag...\n");
        release_device(0);
        exit(22);
    }

    exec sql select max(SEQ_NO),sum(SIZE_IN_KB)
           into :seq:nullind,:v_size_in_kb
           from TAPE_FILES
           where serial_no=:new_serial_no;
    if ( sqlca.sqlcode < 0 ) sql_error(18);

    exec sql select capacity_in_kb into :v_capacity_in_kb
           from TAPE_DEVICES
           where DEV_NAME=:v_dev_name and
                 DEV_GROUP=:dev_group_num;
    if ( tot_size > v_capacity_in_kb ) {
        printf("%s device do not have enough capacity for this
file...\n",v_dev_name);
        printf("Either use another device or split the file...\n");
        release_device(0);
    }
}

```

```

        exit(23);
    }

    printf("Device %s will be used...\n",v_dev_name);
    norew_dev_name();
    if ( v_size_in_kb + tot_size > v_capacity_in_kb ) {
        get_cart();
        exec sql update CURRENT_CART
                set SERIAL_NO=:new_serial_no
                where KEY=:continue_key;
        if ( sqlca.sqlcode < 0 ) sql_error(19);

    } else {
        exec sql update TAPE_SERIALS set IN_USE='C'
                where SERIAL_NO=:new_serial_no and
                IN_USE='Y';
        if ( sqlca.sqlcode < 0 ) sql_error(20);
    }
    exec sql commit;
    if ( sqlca.sqlcode < 0 ) sql_error(21);
}

/* find the version number */
if ( version!=0 ) {
    exec sql select CURRENT_VER+:version into :current_version
            from VERSIONS
            where FILE_NAME=:file_name;
    if ( sqlca.sqlcode < 0 ) sql_error(22);

    if (sqlca.sqlcode==100 ) {
        printf("Could not find version...\n\n");
        release_device(1);
        exit(24);
    }
}

if ( version!=0 ) {
    exec sql update VERSIONS
            set CURRENT_VER=CURRENT_VER+:version
            where FILE_NAME=:file_name;
    if ( sqlca.sqlcode < 0 ) sql_error(23);
}

```

*Editor's note: this article will be concluded in the next issue.*

---

*Abdullah Ongul*  
*DBA*  
*Disbank (Turkey)*

© Xephon 2001

---

# AIX news

---

Entegrity has announced Version 1.5 of its AssureAccess user access management software that is said to help reduce application security development time for Web and application server-based Java applications.

It provides cross-domain single sign-on functionality, a new COM interface for easier integration with Windows-based applications, real-time security policy updates, instant blocking of malicious users, and increased support for platforms, application servers, Web servers, and servlet engines.

Once users are authenticated, they can access applications within the organization's network or on any partner or affiliate network without needing to login again.

Platform support is now extended to include AIX 4.3 in addition to its current support for NT 4.0, Windows 2000, Solaris 2.7 and 2.8, and HP-UX 11.0.

For further information contact:  
Entegrity Solutions, 2077 Gateway Place,  
Suite 200, San Jose, CA 95110, USA.  
Phone: (408) 487 8600.  
URL: <http://www2.entegrity.com/products/assureaccess.shtml>.

\* \* \*

Sybase has announced availability of its EAServer 4.0 application server with support for J2EE 1.3, including EJB 2.0, updated connectors, an exposed Java Messaging Service API, Web server plug-ins, unified install, and a new Java Dynamic Management Kit API.

It will continue to support CORBA 2.3, COM, and PowerBuilder for component deployment, along with the J2EE Connector Architecture API. The new version JSP and Servlet action requests from the user's Web server without requiring a third-party JSP/Servlet engine or a change of Web server.

Application deployment is addressed by a new unified installation feature, allowing deployment without end-user intervention. System administrators can monitor application servers in real-time with the added JDMK API.

Version 4.0 now supports Linux, along with AIX, HP-UX, Solaris, and Windows.

For further information contact:  
Sybase, 6475 Christie Ave, Emeryville, CA  
94608, USA.  
Tel: (510) 922 3500.  
URL: <http://www.sybase.com/easerver>.

\* \* \*

IBM has announced new and deeper integration between Lotus Domino, WebSphere Application Server, and DB2 Universal Database in the next version of Domino.

The Domino server beta is available for AIX, Solaris, Linux, and Windows NT

For further information contact your local IBM representative.  
URL: <http://www.lotus.com>.

\* \* \*



**xephon**