August 2003

In this issue

3 Recovery of an inadvertently removed filesystem
6 AIX system time synchronization using NTP protocol
16 Using tape libraries with AIX – part 2
46 Shell script library
55 AIX news

© Xephon plc 2003
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

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 1999 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/aix; you will need to supply a word from the printed issue.

Editors
Trevor Eddolls

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/nfc.

© Xephon plc 2003. 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.
Recovery of an inadvertently removed filesystem

If you have ever removed a filesystem by accident, you probably wanted to recover from the situation without needing to restore from a back-up. There is indeed a way to do this, but you will need some information taken prior to the accident.

As you know, the logical partitions of a logical volume are mapped to physical partitions on one or more hdisks; and the information for this mapping is essential to recreate the logical volume and its associated filesystem in the case of an accident.

Please be aware that doing an `rmfs -r <filesystemname>` will remove the filesystem itself, the underlying logical volume, and the associated mountpoint.

I believe that you, being a good system administrator, will have already set up some procedures to back up critical information on your managed systems; so you only have to enhance these procedures to store the logical volume mapping in a safe place – if you have not already done so.

The following will describe, in a step-by-step way, a kind of training scenario so that you can become familiar with the tasks involved.

SETTING UP THE TEST SCENARIO

First, we will create a logical volume and an associated filesystem for test purposes with only one logical partition. Let us assume the existence of a volume group named lvtestvg already created on hdisk2:

```
mklv -y lvtest lvtestvg 1 hdisk2
crfs -v jfs -d lvtest -m /to_be_removed -A yes
mount /to_be_removed
```

Now we will create a subdirectory, and put into it a file containing some data:

```
mkdir -p /to_be_removed/subdir1
ps -ef > /to_be_removed/subdir1/data1
```
The next step shows how to collect the mapping information of all logical volumes within all active volume groups of a system; please be aware that we are only interested in getting the information for the first logical volume copy even if the removed logical volume has been mirrored. In our scenario we use an unmirrored logical volume but the procedures described will also work well with a mirrored one:

```bash
for VG in `lsvg -o`
do
for LV in `lsvg -l $VG | grep -Ev "$VG|LV" | awk '{print $1}'`
do
lslv -m $LV | grep ^[0-9] | awk '{print $3 ":" $2}' > map_$LV
done
done
```

You now have a mapping file named `map_<logical volume name>` for each logical volume within all your active volume groups.

Now we will remove the filesystem to complete our test scenario:

```bash
umount /to_be_removed
rmfs -r /to_be_removed
```

**RECREATING THE LOST DATA**

The next command will work only if the physical partition of our removed logical volume is not associated with another logical volume which, for example, has been created just after deleting our test logical volume. So if you inadvertently removed a filesystem, make sure that nobody else is working on your system and creating filesystems. Be aware that you must specify the number of logical partitions that the logical volume consists of:

```bash
mklv -y lvtest -m map_lvtest lvtestvg 1
```

Now you have to manually edit the `/etc/filesystems` and create the lost entry of your filesystem. Do not create the filesystem by using the `crfs` command or you will be sorry:

```bash
/to_be_removed:
    dev = /dev/lvtest
    vfs = jfs
```
log = /dev/loglv00
mount = true
check = false
options = rw
account = false

The /dev entry represents your logical volume name and the log-device can be found by looking at your volume group.

Do a filesystem check just to verify that almost everything is OK:

```
fsck -y /to_be_removed
```

When you have a look at the Logical Volume Control Block (LVCB), you will see that there is just a little work to do:

```
getlvcb -AT lvtest
```

The output will look similar to this (I truncated the unimportant lines):

```
AIX LVCB
  intrapolicy = m
  copies = 1
  interpolicy = x
  lvid = 0000715716b08348.10
  lvname = lvtest
  label = None
  ....
```

and as you can see here, the label – which is the mountpoint of our filesystem – is still missing. Just do the following:

```
chfs -a log=/dev/loglv00 /to_be_removed
mkdir /to_be_removed
mount /to_be_removed.
```

For the paranoid among us, do a `getlvcb -AT lvtest` again to see that everything is now OK.

To finalize the whole procedure, do not forget to mirror your logical volume if it was a mirrored one or if you would like to have it mirrored now.

That's it.

Marcus Zitranski
IT Specialist
IBM (Germany)

© IBM 2003
AIX system time synchronization using NTP protocol

Time value synchronization is essential for the operation of many computer systems. Services such as DNS, Kerberos, cron, and advanced scheduler systems such as IBM’s Load Leveler depend on the synchronization of the clocks on the servers involved.

The Network Time Protocol (NTP) was designed to automate time synchronization across a network. It has replaced the previously available and now obsolete TIMED protocol.

NTP OPERATION OVERVIEW
NTP works as a hierarchical client/server network consisting of the following types of node:

- A client node queries the reference time from one or more servers.
- A server node makes its time available as the reference time for other clients.
- A peer node compares its system time with other peers until all the peers finally agree about the ‘true’ time to synchronize on.
- A broadcast/multicast server – an NTP server can also operate in a broadcast or multicast mode. Both work similarly – broadcast servers send periodic time updates to a broadcast address, while multicast servers send periodic updates to a multicast address. Using broadcast packets can greatly reduce the NTP traffic on a network, especially for a network with many NTP clients.
- A broadcast/multicast client – an NTP broadcast or multicast client listens for NTP packets on a broadcast or multicast address. When the first packet is received, it attempts to
quantify the delay to the server in order to better quantify the
correct time from later broadcasts. This is accomplished by
a series of brief interchanges where the client and server act
as a regular (non-broadcast) NTP client and server. Once
these interchanges occur, the client has an idea of the
network delay and thereafter can estimate the time based
only on broadcast packets. If this interchange is not desirable,
it can be disabled using NTP’s access control features. The
\texttt{-r} option can be used when \texttt{xntpd} is started to hardwire a
delay if the interchange fails because of access control
issues or other problems.

The hierarchical levels of the time synchronization structure are
called stratum levels. A smaller stratum number means a higher
level in the hierarchy structure. On top of the hierarchy there is
the daemon, which has the most accurate time and therefore the
smallest stratum number.

By default, a daemon’s stratum level is always one level below
the level of its reference time source. The top-level daemon often
uses an external device as a reference time source. Such
devices have a stratum number of 0, so a daemon that uses that
precision time device as a reference time source will be a stratum
1 time server, which has the highest priority level in the NTP
hierarchy. In large networks, it is a good practice to install one or
more stratum 1 time servers, which make a reference time
available to several server computers in each department. Thus
the servers in the departments become stratum 2 time servers,
which can be used as reference time sources for workstations
and other network devices of the department.

In fact there are several ways to obtain authoritative time:

- To connect the system to an external autonomous atomic
clock.
- To dial your national institute of standards using a modem.
- To connect your system to a GPS-based (Global Positioning
  System) device, which receives time values as well as
  positioning information from satellites.
• It is possible to obtain authoritative time values from external stratum 1 and 2 NTP Internet servers. This is the most common source for Internet-connected organizations that don’t require more extreme precision. Note that some servers require advance permission before time synchronization can be implemented.

Each NTP daemon can be configured to use several independent reference time sources. It synchronizes to the reference time source with the highest stratum and lowest jitter and dispersion. If that reference time source becomes unavailable then the daemon automatically switches to the best of the remaining time sources, which may also result in a change to the daemon’s stratum value.

When an NTP daemon is working on a client, periodic adjustments to a system clock are made based on the authoritative time data that is received from the NTP servers. If the current time on the client system differs by more than 128 milliseconds, the NTP daemon resets the local system clock. If the difference is smaller, the NTP adjusts the local clock gradually in small steps. Over time the NTP daemon records and analyses successive time errors and continues to correct the time automatically based on this data, even when the time server systems are unreachable.

It is customary to set up the NTP using multiple sources of authoritative time. This precaution is taken in order to protect the NTP server from single points of failure and unreliability of a single time server (resulting from hardware failure, malicious tampering, etc). Similarly NTP clients are set with at least three sources of time information.

BASIC NTP CONFIGURATION ON AIX

The first step in setting up any NTP-based network is thorough planning. Among the things to consider are:

• Number and origin of authoritative time servers.
• Number and location of NTP servers within the organization.
• Distribution of time servers between the NTP clients in a balanced and protective way.

In AIX, the NTP support is installed by default as part of the bos.tcpip.client fileset.

The NTP system configuration resides in file /etc/ntp.conf.

To define an NTP server that uses well-known Internet-accessible servers, use the following:

```
# Direct the NTP to NOT use broadcast based synchronization
broadcastclient no
# Define three Internet-accessible servers by IP address
server 128.46.136.95 # US IN PERDUE
server 140.239.10.5  # US MA CONCORD
server 128.118.25.3  # US PA PENN ST
# Next two lines define local computer system clock as very low priority
# (last chance) time source
server 127.127.1.0
fudge 127.127.1.0 stratum 12
# Define the file to save the local clock drift from the reference time
# to be used after the system restart
driftfile /etc/ntp.drift
```

A typical local client can have the following /etc/ntp.conf contents:

```
# Direct the NTP to NOT use broadcast-based synchronization
broadcastclient no
# Define three local NTP servers by host name
server nphost1
server nphost2
server nphost3
# Define the file to save the local clock drift from the reference time
# to be used after the system restart
driftfile /etc/ntp.drift
```

There are many more options that can be set up using the configuration file. Please refer to the NTP manual pages for details.

The next step is to uncomment the following line in the file /etc/rc.tcpip:

```
start /usr/sbin/xntpd "$src_running"
```

This will enable the automatic restart of the NTP daemon following a reboot of the server.
The next step is to execute the `ntpd` utility with the name of the existing NTP server in order to perform initial synchronization of the local system clock with the reference server:

```
ntpd -d ntphost1 ntphost2 ntphost3
```

Wait for an acknowledgement that a good connection has been established.

Perform the actual time synchronization:

```
ntpd -d ntphost1 ntphost2 ntphost3
```

Finally you can start the NTP daemon from the command line using:

```
startsrc -s xntpd
```

**MONITORING NTP STATUS**

The `/usr/sbin/ntpq` utility enables the state of the NTP daemon on a local or remote computer to be queried. Using `ntpq`, an administrator can check the configuration of a remote host. If such queries are allowed on a host, this can be a useful way of choosing hosts to synchronize with, because information such as their peers and reference clock types can be determined. Since `ntpq` uses UDP packets, hosts may be falsely unreachable on congested networks.

`ntpq` can be run in an interactive mode or in batch mode. In batch mode, `ntpq` executes a command and returns to the command prompt. The parameter `-p` (‘peers’) lets `ntpq` print the status of an NTP daemon. Enter:

```
ntpq -p
```

to display the status of the daemon on the local machine, or:

```
ntpq -p ntphost
```

to display the status of the daemon on the remote host `ntp_server`. The command should print a table with one status line for each reference time source that has been configured for the NTP daemon on the specified host:
The table above shows the output for an NTP daemon, which has three reference time sources: hosts running NTP protocol and named ntphost1, ntphost2, and ntphost3.

If the first character of a line is not blank then it contains a qualifier for the corresponding reference time source. Immediately after the daemon has been started, all qualifiers are blank. The NTP daemon needs several polling cycles to check the available time sources and declare one of them as the reference it synchronizes with.

An asterisk * in the first column marks the reference time source which is currently preferred by the NTP daemon, the + character marks high-quality candidates for the reference time that could be used if the currently selected reference time source should become unavailable.

The column remote displays the IP address or the host name of the reference time source, where LOCAL refers to the local clock. The refid shows the type of the reference clock, where LOCAL or LCL refers to the local clock, .DCFa refers to a standard DCF77 time source, and .PPS indicates that the reference clock is disciplined by a hardware pulse-per-second signal. Other identifiers are possible, depending on the type of the reference clock.

The column st reflects the stratum number of the reference time source. In the example above, the remote time server ntphost1 has stratum 4, which is the best you can find on this network, and the other servers have stratum 5.

Every time a when count reaches the poll number in the same line, the NTP daemon queries the time from the corresponding time source and resets the when count to 0. The query results of
each polling cycle are filtered and used as a measure for the clock’s quality and reachability.

The column *reach* shows whether a reference time source could be reached at the last polling intervals, i.e., data could be read from the reference time source, and the reference time source was synchronized. The value must be interpreted as an 8-bit shift register whose contents are displayed as octal values. If the NTP daemon has just started, the value is 0. Each time a query was successful a ‘1’ is shifted in from the right, so after the daemon has been started the sequence of *reach* numbers 0, 1, 3, 7, 17, 37, 77, 177, 377. The maximum value 377 means that the eight last queries were completed successfully. The NTP daemon must have reached a reference time source several times (reach not 0) before it selects a preferred time source and puts an asterisk in the first column.

The columns *delay*, *offset*, and *disp* show some timing values that are derived from the query results. All values are in milliseconds. The *delay* value is derived from the roundtrip time of the queries. The *offset* value shows the difference between the reference time and the system clock. The *disp* value indicates the magnitude of dispersion between several time queries.

The /usr/sbin/xntpdc program also allows the state of a local or remote NTP daemon to be queried; however, xntpd can also make runtime configuration requests to a remote machine. This allows the configuration to be changed on-the-fly. In order to make runtime configuration changes, an authentication key is needed. This requires the creation of an NTP keys file, which is described in the xntpd man page. Like ntpq, xntpd uses UDP packets.

The /usr/sbin/ntptrace is an informational command that traces the source of a given client’s time. The information found by ntptrace can be determined by successive runs of ntpq on each client’s preferred server. However, ntptrace provides an easy and convenient way of learning this information. This can be a useful tool for debugging.
This program sends an NTP packet to the specified server. If possible, the server will respond with an NTP packet. The host ntptrace was run on will use this information to determine the stratum, offset, and synchronization distance, as well as the next server in the hierarchy. Since ntptrace now knows the next server in the hierarchy, it sends that server an NTP packet. This continues up the hierarchy. If the next server is not reachable, a time-out results.

However, because of the above implementation, there are some problems with ntptrace. If the machine ntptrace is run on is not authorized to get time from one of the servers, ntptrace will show a time-out, even though the servers in the chain may have the necessary access permission. Because ntptrace uses normal NTP packets (mode 3), rather than NTP queries (mode 6), or control requests (mode 7), it is less likely that this will be a problem (since few sites restrict time requests).

The following is an example of ntptrace output:

```
localhost: stratum 6, offset -0.000063, synch distance 1.19437
ntphost1: stratum 5, offset 1.903098, synch distance 0.13986
ntphost2: stratum 4, offset 1.900733, synch distance 0.12984
ntphost3: stratum 3, offset -1.880003, synch distance 0.10143
ntphost4: stratum 2, offset 1.904045, synch distance 0.06825
ntphost5: stratum 1, offset 1.907003, synch distance 0.00797, refid 'TRUE'
```

The ntptrace output lists the client name, its stratum, its time offset from the local host, the synchronization distance, and the ID of the reference clock attached to a server, if one exists. The synchronization distance is a measure of clock accuracy, assuming that it has a correct time source. For the exact derivation, refer to the NTP specification.

INTERNET RESOURCES

All included documents discuss the public domain NTP implementation but are nevertheless useful for an AIX environment as well.

More technical information can be found on the following sites:

- [http://www.eecis.udel.edu/~mills/ntp.htm](http://www.eecis.udel.edu/~mills/ntp.htm)
- [http://www.eecis.udel.edu/~mills/papers.htm](http://www.eecis.udel.edu/~mills/papers.htm)
- [http://www.eecis.udel.edu/~mills/reports.htm](http://www.eecis.udel.edu/~mills/reports.htm)
- [http://www.eecis.udel.edu/~mills/memos.htm](http://www.eecis.udel.edu/~mills/memos.htm)

The following RFCs relate to NTP. These can be found in many places throughout the Internet, but are also available from the NTP home page:

- RFC-1165? – *Network Time Protocol (NTP) over the OSI*


Sun Microsystems’ staff have published a series of blueprints describing NTP theory, set up, and monitoring at http://www.sun.com/blueprints:

- Using NTP to Control and Synchronize System Clocks – Part I: Introduction to NTP.

- Using NTP to Control and Synchronize System Clocks – Part II: Basic NTP Administration and Architecture.

- Using NTP to Control and Synchronize System Clocks – Part III: NTP Monitoring and Troubleshooting.

REFERENCES


Alex Polyak
System Engineer
APS (Israel) © Xephon 2003
Using tape libraries with AIX – part 2

This month we conclude the shell script that adds a tapeutil management option to SMIT’s top level menu.

```plaintext
sm_menu_opt:
    id_seq_num = ""
    id = "smc"
    next_id = "smc_query"
    text = ""
    text_msg_file = ""
    text_msg_set = Ø
    text_msg_id = Ø
    next_type = "n"
    alias = "y"
    help_msg_id = ""
    help_msg_loc = ""
    help_msg_base = ""
    help_msg_book = ""

sm_name_hdr:
    id = "smc_query"
    next_id = "smc_query_hdr"
    option_id = "smc_opt"
    has_name_select = "n"
    name = "Medium Changer"
    name_msg_file = ""
    name_msg_set = Ø
    name_msg_id = Ø
    type = ""
    ghost = "y"
    cmd_to_classify = ""
    cmd_to_classify_postfix = ""
    raw_field_name = "logicname"
    cooked_field_name = ""
    next_type = "d"
    help_msg_id = ""
    help_msg_loc = ""
    help_msg_base = ""
    help_msg_book = ""

sm_cmd_hdr:
    id = "smc_query_hdr"
    option_id = "smc_query_opt"
    has_name_select = "y"
    name = "Medium Changer List Command"
    name_msg_file = ""
    name_msg_set = Ø
    name_msg_id = Ø
```

© 2003. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.
cmd_to_exec = "tapeutil \\nask = "n" 
exec_mode = ""  
ghost = "n" 
cmd_to_discover = "" 
cmd_to_discover_postfix = "" 
name_size = Ø 
value_size = Ø 
help_msg_id = "" 
help_msg_loc = "" 
help_msg_base = "" 
help_msg_book = "" 

sm_cmd_opt: 
  id_seq_num = "Ø" 
id = "rmt_opt" 
disc_field_name = "logicname" 
name = "Tape Drives and Medium Changers" 
name_msg_file = "" 
name_msg_set = Ø 
name_msg_id = Ø 
op_type = "" 
entry_type = "" 
entry_size = Ø 
required = "y" 
prefix = "-f /dev/" 
cmd_to_list_mode = "1"  
cmd_to_list = "lsdev -Cc tape -S Available | sort -n -k 1.4" 
cmd_to_list_postfix = "" 
multi_select = "" 
value_index = Ø 
disp_values = "" 
values_msg_file = "" 
values_msg_set = Ø 
values_msg_id = Ø 
aix_values = "" 
help_msg_id = "" 
help_msg_loc = "" 
help_msg_base = "" 
help_msg_book = "" 

sm_cmd_opt: 
  id_seq_num = "Ø" 
id = "smc_opt" 
disc_field_name = "logicname" 
name = "Medium Changer" 
name_msg_file = "" 
name_msg_set = Ø 
name_msg_id = Ø 
op_type = "" 
entry_type = "" 
entry_size = Ø
required = "y"
prefix = ".f /dev/

sm_cmd_opt:
  id_seq_num = "010"
id = "smc_query_opt"
disc_field_name = "logicname"
name = "Medium Changer"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = ""
entry_size = Ø
required = "y"
prefix = ".f /dev/

sm_cmd_opt:
  id_seq_num = "020"
id = "smc_query_opt"
disc_field_name = ""
name = "Selected Command"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = "r"
entry_type = "n"
entry_size = Ø
required = "y"
prefix = 

multi_select = 
value_index = Ø
disp_values = "elementinfo,inventory,Device IDs,Device Info"
values_msg_file = 
values_msg_set = Ø
values_msg_id = Ø
aix_values = "elementinfo,inventory,devids,devinfo"
help_msg_id = 
help_msg_loc = 
help_msg_base = 
help_msg_book = 

sm_menu_opt:
id_seq_num = "2002"
id = "tapeutil"
next_id = "smcpath"
text = "Medium Changer Path Commands"
text_msg_file = 
text_msg_set = Ø
text_msg_id = Ø
next_type = "m"
alias = 
help_msg_id = 
help_msg_loc = 
help_msg_base = 
help_msg_book = 

sm_menu_opt:
id_seq_num = 
id = "smcpath"
next_id = "smc_path"
text = 
text_msg_file = 
text_msg_set = Ø
text_msg_id = Ø
next_type = "n"
alias = "y"
help_msg_id = 
help_msg_loc = 
help_msg_base = 
help_msg_book = 

sm_name_hdr:
id = "smc_path"
next_id = "smc_path_hdr"
option_id = "smc_opt"
has_name_select = "n"
name = "Medium Changer"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
type = ""
ghost = "y"
cmd_to_classify = ""
cmd_to_classify_postfix = ""
raw_field_name = "logicname"
cooked_field_name = ""
next_type = "d"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_hdr:
  id = "smc_path_hdr"
  option_id = "smc_path_opt"
  has_name_select = "y"
  name = "Medium Changer Path Command"
  name_msg_file = ""
  name_msg_set = Ø
  name_msg_id = Ø
  cmd_to_exec = "tapeutil"
  ask = "y"
  exec_mode = ""
  ghost = "n"
  cmd_to_discover = ""
  cmd_to_discover_postfix = ""
  name_size = Ø
  value_size = Ø
  help_msg_id = ""
  help_msg_loc = ""
  help_msg_base = ""
  help_msg_book = ""

sm_cmd_opt:
  id_seq_num = "010"
  id = "smc_path_opt"
disc_field_name = "logicname"
name = "Medium Changer"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = ""
entry_size = Ø
sm_opt:
  required = "y"
  prefix = ".f /dev/"
  cmd_to_list_mode = ""
  cmd_to_list = ""
  cmd_to_list_postfix = ""
  multi_select = ""
  value_index = Ø
  disp_values = ""
  values_msg_file = ""
  values_msg_set = Ø
  values_msg_id = Ø
  aix_values = ""
  help_msg_id = ""
  help_msg_loc = ""
  help_msg_base = ""
  help_msg_book = ""
sm_cmd_opt:
  id_seq_num = "020"
  id = "smc_path_opt"
  disc_field_name = ""
  name = "Selected Path Command"
  name_msg_file = ""
  name_msg_set = Ø
  name_msg_id = Ø
  op_type = "r"
  entry_type = "n"
  entry_size = Ø
  required = "y"
  prefix = ""
  cmd_to_list_mode = ""
  cmd_to_list = ""
  cmd_to_list_postfix = ""
  multi_select = ""
  value_index = Ø
  disp_values = "query path,check path,enable primary,enable alternate,disable primary,disable alternate,reset path"
  values_msg_file = ""
  values_msg_set = Ø
  values_msg_id = Ø
  aix_values = "qrypath,checkpath,enablepath 'primary,enablepath' 'alternate,disable path' 'primary,disable path' 'alternate,reset path"
  help_msg_id = ""
  help_msg_loc = ""
  help_msg_base = ""
  help_msg_book = ""
sm_menu_opt:
  id_seq_num = "2007"
  id = "tapeutil"
  next_id = "rmtrr"
  text = "Reservation For Exclusive Usage"
text_msg_file = ""
text_msg_set = Ø
text_msg_id = Ø
next_type = "m"
alias = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_menu_opt:
  id_seq_num = ""
id = "rmt_rr"
next_id = "rmt_rr"
text = ""
text_msg_file = ""
text_msg_set = Ø
text_msg_id = Ø
next_type = "n"
alias = "y"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_name_hdr:
  id = "rmt_rr"
next_id = "rmt_rr_hdr"
option_id = "rmt_opt"
has_name_select = "n"
name = "Tape Drive"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
type = ""
ghost = "y"
cmd_to_classify = ""
cmd_to_classify_postfix = ""
raw_field_name = "logicname"
cooked_field_name = ""
next_type = "d"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_hdr:
  id = "rmt_rr_hdr"
option_id = "rmt_rr_opt"
has_name_select = "y"
name = "Reservation For Exclusive Usage"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
cmd_to_exec = "tapeutil"
ask = "y"
exec_mode = ""
ghost = "n"
cmd_to_discover = ""
cmd_to_discover_postfix = ""
name_size = Ø
value_size = Ø
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "010"
id = "rmt_rr_opt"
disc_field_name = "logicname"
name = "Tape Drive"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = ""
entry_size = Ø
required = "y"
prefix = "-f /dev/

sm_cmd_opt:
id_seq_num = "020"
id = "rmt_rr_opt"
disc_field_name = ""
name = "Select Exclusive Usage To"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = "r"
entry_type = "n"
entry_size = 0
required = "y"
prefix = ""
cmd_to_list_mode = ""
cmd_to_list = ""
cmd_to_list_postfix = ""
multi_select = ""
value_index = 0
disp_values = "status, reserve, release"
values_msg_file = ""
values_msg_set = 0
values_msg_id = 0
aix_values = "status, reserve, release"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_menu_opt:
id_seq_num = "2009"
id = "tapeutil"
next_id = "rmtsense"
text = "Show Sense Codes or Binary Information"
text_msg_file = ""
text_msg_set = 0
text_msg_id = 0
next_type = "m"
alias = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_menu_opt:
id_seq_num = ""
id = "rmtsense"
next_id = "rmt_sense"
text = ""
text_msg_file = ""
text_msg_set = 0
text_msg_id = 0
next_type = "n"
alias = "y"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_name_hdr:
id = "rmt_sense"
next_id = "rmt_sense_hdr"
option_id = "rmt_opt"
has_name_select = "n"
name = "Tape Drive"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
type = ""
ghost = "y"
cmd_to_classify = ""
cmd_to_classify_postfix = ""
raw_field_name = "logicname"
cooked_field_name = ""
next_type = "d"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_hdr:
    id = "rmt_sense_hdr"
    option_id = "rmt_sense_opt"
    has_name_select = "y"
    name = "Show Sense Codes or Binary Information"
    name_msg_file = ""
    name_msg_set = Ø
    name_msg_id = Ø
    cmd_to_exec = "x() 
        { 
            case $3 in
                reqsense|vpd)              tapeutil $1 $2 $3 
            ;;
                inquiry|logpage|modepage) tapeutil $1 $2 $3 $4 
            ;;
            *)          echo ERROR in tapeutil . $1 . $2 . $3 . $4 
            ;;
        }
    x"
}

ask = "n"
exec_mode = ""
ghost = "n"
cmd_to_discover = ""
cmd_to_discover_postfix = ""
name_size = Ø
value_size = Ø
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "010"
id = "rmt_sense_opt"
disc_field_name = "logicname"
name = 'Tape Drive'
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = ""
entry_size = Ø
required = "y"
prefix = "f /dev/
cmd_to_list_mode = ""
cmd_to_list = ""
cmd_to_list_postfix = ""
multi_select = ""
value_index = Ø
disp_values = ""
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "020"
id = "rmt_sense_opt"
disc_field_name = ""
name = 'Issue SCSI Command To Show'
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = "r"
entry_type = "n"
entry_size = Ø
required = "y"
prefix = ""
cmd_to_list_mode = ""
cmd_to_list = ""
cmd_to_list_postfix = ""
multi_select = ""
value_index = Ø
disp_values = "inquiry data, log sense data, mode sense data, request sense data, vital product data"
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = "inquiry, logpage, modepage, reqsense, vpd"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
  id_seq_num = "030"
  id = "rmt_sense_opt"
  disc_field_name = ""
  name = "Selected Page (hex characters)"
  name_msg_file = ""
  name_msg_set = Ø
  name_msg_id = Ø
  op_type = ""
  entry_type = "t"
  entry_size = 2
  required = "n"
  prefix = ""
  cmd_to_list_mode = ""
  cmd_to_list = ""
  cmd_to_list_postfix = ""
  multi_select = ""
  value_index = Ø
  disp_values = ""
  values_msg_file = ""
  values_msg_set = Ø
  values_msg_id = Ø
  aix_values = ""
  help_msg_id = ""
  help_msg_loc = ""
  help_msg_base = ""
  help_msg_book = ""

sm_menu_opt:
  id_seq_num = "2001"
  id = "tapeutil"
  next_id = "smcaudit"
  text = "Medium Changer Audit Commands"
  text_msg_file = ""
  text_msg_set = Ø
  text_msg_id = Ø
  next_type = "m"
  alias = ""
  help_msg_id = ""
  help_msg_loc = ""
  help_msg_base = ""
  help_msg_book = ""

sm_menu_opt:
  id_seq_num = ""
  id = "smcaudit"
  next_id = "smc_aud"t"
  text = ""
  text_msg_file = ""
  text_msg_set = Ø
  text_msg_id = Ø
next_type = "n"
alias = "y"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""
sm_name_hdr:
id = "smc_audit"
next_id = "smc_audit_hdr"
option_id = "smc_opt"
has_name_select = "n"
name = "Medium Changer"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
type = ""
ghost = "y"
cmd_to_classify = ""
cmd_to_classify_postfix = ""
raw_field_name = "logicname"
cooked_field_name = ""
next_type = "d"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""
sm_cmd_hdr:
id = "smc_audit_hdr"
option_id = "smc_audit_opt"
has_name_select = "y"
name = "Medium Changer Audit Command"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
cmd_to_exec = "tapeutil "
ask = "y"
exec_mode = ""
ghost = "n"
cmd_to_discover = ""
cmd_to_discover_postfix = ""
name_size = Ø
value_size = Ø
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""
sm_cmd_opt:
id_seq_num = "000"
id = "smc_audit_opt"
disc_field_name = "logicname"
name = "Medium Changer"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = ""
entry_size = Ø
required = "y"
prefix = ".f /dev/
 cmd_to_list_mode = ""
 cmd_to_list = ""
 cmd_to_list_postfix = ""
multi_select = ""
value_index = Ø
disp_values = ""
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = ""
help_msg_id = Ø
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
  id_seq_num = "020"
id = "smc_audit_opt"
disc_field_name = ""
name = "Starting Address (all if missing)"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = "!
entry_type = "#"
entry_size = Ø
required = "y"
prefix = "audit"
 cmd_to_list_mode = "1"
 cmd_to_list = "callme() \n\n{      
   tapeutil -f /dev/$1 inventory 2>/dev/null | grep '^[A-Z]' | awk 
   '{if(NR>1)print "$NF,\"is a\",$1,$2}'\n}\ncallme "
 cmd_to_list_postfix = "logicle name"
multi_select = ""
value_index = Ø
disp_values = ""
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "030"
id = "smc_audit_opt"
disc_field_name = ""
name = "Number Of Elements (1 if missing)"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = "#"
entry_size = Ø
required = "n"
prefix = ""
cmd_to_list_mode = ""
cmd_to_list = ""
cmd_to_list_postfix = ""
multi_select = ""
value_index = Ø
disp_values = ""
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_menu_opt:
id_seq_num = "2003"
id = "tapeutil"
next_id = "smcmount"
text = "Tape Mount to Drive"
text_msg_file = ""
text_msg_set = Ø
text_msg_id = Ø
next_type = "m"
alias = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_menu_opt:
id_seq_num = ""
id = "smcmount"
next_id = "smc_mount"
text = ""
text_msg_file = ""
text_msg_set = Ø
text_msg_id = Ø
next_type = "n"
alias = "y"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""
sm_name_hdr:
id = "smc_mount"
next_id = "smc_mount_hdr"
option_id = "rmt_clear_opt"
has_name_select = "n"
name = "Tape Drive to Work with"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
type = ""
ghost = "y"
cmd_to_classify = ""
cmd_to_classify_postfix = ""
raw_field_name = "logicname"
cooked_field_name = ""
next_type = "d"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""
sm_cmd_hdr:
id = "smc_mount_hdr"
option_id = "smc_mount_opt"
has_name_select = "y"
name = "Mount Tape From Slot"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
cmd_to_exec = "tapeutil"
ask = "y"
exec_mode = ""
ghost = "n"
cmd_to_discover = ""
cmd_to_discover_postfix = ""
name_size = Ø
value_size = Ø
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""
sm_cmd_opt:
  id_seq_num = "Ø1Ø"
id = "smc_mount_opt"
disc_field_name = "logicname"
name = "Tape Drive"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = ""
entry_size = Ø
required = "y"
prefix = "-f /dev/"
cmd_to_list_mode = ""
cmd_to_list = ""
cmd_to_list_postfix = ""
multi_select = ""
value_index = Ø
disp_values = ""
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

ds m cmd opt:
id_seq_num = "Ø3Ø"
id = "smc_mount_opt"
disc_field_name = ""
name = "Slot Address"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = "l"
entry_type = "#"
entry_size = Ø
required = "y"
prefix = "mount "]
cmd_to_list_mode = "1"
cmd_to_list = "callme() "
  
  (tapeutil -f /dev/\$1 inventory 2>/dev/null | grep -p 'Media Present.*Yes' | grep '^[A-Z]' | grep -v Drive | awk '{if(NR>1)print \$NF, "is a", \$1, \$2}"
  ) "
callme "
cmd_to_list_postfix = "logicname"
multi_select = ""
value_index = Ø
disp_values = *
values_msg_file = *
values_msg_set = Ø
values_msg_id = Ø
aix_values = *
help_msg_id = *
help_msg_loc = *
help_msg_base = *
help_msg_book = *

sm_cmd_opt:
id_seq_num = "0"
id = "rmt_clear_opt"
disc_field_name = "logicname"
name = "Empty Tape Drives"
name_msg_file = *
name_msg_set = Ø
name_msg_id = Ø
op_type = *
entry_type = *
entry_size = Ø
required = "y"
prefix = "-f /dev/

cmd_to_list_mode = "1"
cmd_to_list = "for tape in \$(lsdev -Cc tape -S Available | grep rmt | awk '{print \$1}'); do if [ \$(tapeutil -f /dev/\$tape status | grep 'Medium Type' | awk '{print \$NF}') -eq Ø ]; then lsdev -Cc tape -l \$tape ; fi ; done | sort -n -k 1.4"
cmd_to_list_postfix = *
multi_select = *
value_index = Ø
disp_values = *
values_msg_file = *
values_msg_set = Ø
values_msg_id = Ø
aix_values = *
help_msg_id = *
help_msg_loc = *
help_msg_base = *
help_msg_book = *

sm_menu_opt:
id_seq_num = "2004"
id = "tapeutil"
next_id = "smcunmount"
text = "Tape Unmount from Drive"
text_msg_file = *
text_msg_set = Ø
text_msg_id = Ø
next_type = "m"
alias = ""

help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_menu_opt:
id_seq_num = ""
id = "smc_unmount"
next_id = "smc_unmount"
text = ""
text_msg_file = ""
text_msg_set = Ø
text_msg_id = Ø
next_type = "n"
alias = "y"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_name_hdr:
id = "smc_unmount"
next_id = "smc_unmount_hdr"
option_id = "rmt_loaded_opt"
has_name_select = "n"
name = "Tape Drive to Work with"
name_msg_file = ""
nname_msg_set = Ø
name_msg_id = Ø
type = ""
ghost = "y"
cmd_to_classify = ""
cmd_to_classify_postfix = ""
raw_field_name = "logicname"
cooked_field_name = ""
next_type = "d"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_hdr:
id = "smc_unmount_hdr"
option_id = "smc_unmount_opt"
has_name_select = "y"
name = "Unmount Tape To Slot"
name_msg_file = ""
nname_msg_set = Ø
name_msg_id = Ø
cmd_to_exec = "tapeutil"
ask = "n"
exec_mode = ""
ghost = "n"
cmd_to_discover = ""
cmd_to_discover_postfix = ""
name_size = 0
value_size = 0
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
  id_seq_num = "010"
id = "smc_unmount_opt"
disc_field_name = "logicname"
name = "Tape Drive"
name_msg_file = ""
name_msg_set = 0
name_msg_id = 0
op_type = ""
entry_type = ""
entry_size = 0
required = "y"
prefix = "-f /dev/

sm_cmd_opt:
  id_seq_num = "030"
id = "smc_unmount_opt"
disc_field_name = ""
name = "Slot Address"
name_msg_file = ""
name_msg_set = 0
name_msg_id = 0
op_type = "l"
entry_type = "#"
entry_size = 0
required = "y"
prefix = "unmount"
cmd_to_list_mode = "1"
cmd_to_list = "callme() \n\n"


```

{ 
    tapeutil -f /dev/$1 inventory 2>/dev/null | grep -p 'Media Present.*No' | grep -p '^[A-Z]' | grep -v Drive | awk '{if(NR>1)print 
    $NF," is a", $1, $2}'
} 

```
callme *

```

while cmd_to_list_postfix = "logicname"
    multi_select = ""
    value_index = 0
    disp_values = ""
    values_msg_file = ""
    values_msg_set = 0
    values_msg_id = 0
    aix_values = ""
    help_msg_id = ""
    help_msg_loc = ""
    help_msg_base = ""
    help_msg_book = ""

sm_cmd_opt:
    id_seq_num = "0"
    id = "rmt_loaded_opt"
    disc_field_name = "logicname"
    name = "Tape Drives with Cartridges"
    name_msg_file = ""
    name_msg_set = 0
    name_msg_id = 0
    op_type = ""
    entry_type = ""
    entry_size = 0
    required = "y"
    prefix = ".f /dev/"
    cmd_to_list_mode = "1"
    cmd_to_list = "for tape in $(lsdev -Cc tape -S Available | grep rmt | awk '{print $1}'); do if [ $(tapeutil -f /dev/$tape status | grep 'Medium Type' | awk '{print $NF}')] -gt 0 ]; then lsdev -Cc tape -l $tape ; fi ; done | sort -n -k 1.4"
    cmd_to_list_postfix = ""
    multi_select = ""
    value_index = 0
    disp_values = ""
    values_msg_file = ""
    values_msg_set = 0
    values_msg_id = 0
    aix_values = ""
    help_msg_id = ""
    help_msg_loc = ""
    help_msg_base = ""
    help_msg_book = ""

sm_menu_opt:
    id_seq_num = "2005"
```

36 © 2003. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.
id = "tapeutil"
next_id = "smcmove"
text = "Medium Changer Moving Tapes"
text_msg_file = **
text_msg_set = Ø
text_msg_id = Ø
next_type = "m"
alias = ""
help_msg_id = **
help_msg_loc = **
help_msg_base = **
help_msg_book = **

sm_menu_opt:
id_seq_num = ""
id = "smcmove"
next_id = "smc_move"
text = ""
text_msg_file = ""
text_msg_set = Ø
text_msg_id = Ø
next_type = "n"
alias = "y"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_name_hdr:
id = "smc_move"
next_id = "smc_move_hdr"
option_id = "smc_opt"
has_name_select = ""
name = "Medium Changer"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
type = "j"
ghost = "y"
cmd_to_classify = ""
cmd_to_classify_postfix = ""
raw_field_name = "logicname"
cooked_field_name = ""
next_type = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_hdr:
id = "smc_move_hdr"
option_id = "smc_move_opt"
has_name_select = "y"
name = "Medium Changer Moving Tapes"
nname_msg_file = ""
nname_msg_set = Ø
name_msg_id = Ø
cmd_to_exec = "tapeutil"
ask = ""
exec_mode = ""
ghost = ""
cmd_to_discover = ""
cmd_to_discover_postfix = ""
name_size = Ø
value_size = Ø
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "000"
id = "smc_move_opt"
disc_field_name = "logicname"
name = "Medium Changer"
nname_msg_file = ""
nname_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = ""
entry_size = Ø
required = "y"
prefix = "-f /dev/
"cmd_to_list_mode = ""
cmd_to_list = ""
cmd_to_list_postfix = ""
multi_select = ""
value_index = Ø
disp_values = ""
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "020"
id = "smc_move_opt"
disc_field_name = ""
name = "From Slot Number"
nname_msg_file = ""
nname_msg_set = Ø
name_msg_id = Ø
op_type = "l"
entry_type = "#"
entry_size = Ø
required = "+
prefix = "move"
cmd_to_list_mode = "$1"
cmd_to_list = "callme() \n\
{n\ntapeutil -f /dev/$1 inventory 2>/dev/null | grep -p 'Media Present.*Yes' | grep '^A-Z' | awk '{if(NR>1)print $NF,"is a",$1,$2}'\n} \n\ncall me"
cmd_to_list_postfix = "logicname"
multi_select = "*
value_index = Ø
disp_values = ""
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "30"
id = "smc_move_opt"
disc_field_name = ""
name = "To Slot Number"
name_msg_id = Ø
name_msg_file = Ø
name_msg_set = Ø
name_msg_id = Ø
op_type = "$1"
entry_type = "#"
entry_size = Ø
required = "+
prefix = "*
cmd_to_list_mode = "$1"
cmd_to_list = "callme() \n\n{n\ntapeutil -f /dev/$1 inventory 2>/dev/null | grep -p 'Media Present.*No' | grep '^A-Z' | awk '{if(NR>1)print $NF,"is a",$1,$2}'\n} \n\ncall me"
cmd_to_list_postfix = "logicname"
multi_select = "*
value_index = Ø
disp_values = "" 
values_msg_file = "" 
values_msg_set = Ø 
values_msg_id = Ø 
aix_values = "" 
help_msg_id = "" 
help_msg_loc = "" 
help_msg_base = "" 
help_msg_book = ""

sm_cmd_opt:
  id_seq_num = "Ø"
  id = "libaddr_opt"
  disc_field_name = "slotnumber"
  name = "Library Addresses"
  name_msg_file = ""
  name_msg_set = Ø 
  name_msg_id = Ø 
  op_type = "I"
  entry_type = ""
  entry_size = Ø 
  required = "+"
  prefix = ""
  cmd_to_list_mode = "1"
  cmd_to_list = "callme() \n\n{\n   tapeutil -f /dev/$1 inventory 2>/dev/null | grep '^[A-Z]' | awk
   '{if(NR>1)print \$NF,"is a",\$1,\$2}'
\n} \n\ncallme "
  cmd_to_list_postfix = "logicle name"
  multi_select = ""
  value_index = Ø 
  disp_values = ""
  values_msg_file = ""
  values_msg_set = Ø 
  values_msg_id = Ø 
aix_values = "" 
  help_msg_id = "" 
  help_msg_loc = "" 
  help_msg_base = "" 
  help_msg_book = ""

sm_menu_opt:
  id_seq_num = "2006"
  id = "tapeutil"
  next_id = "smcexch"
  text = "Medium Changer Exchanging Tapes"
  text_msg_file = ""
  text_msg_set = Ø 
  text_msg_id = Ø 
  next_type = "m"
alias = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_menu_opt:
id_seq_num = ""
id = "smcexch"
next_id = "smc_exch"
text = ""
text_msg_file = ""
text_msg_set = Ø
text_msg_id = Ø
next_type = "n"
alias = "y"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_name_hdr:
id = "smc_exch"
next_id = "smc_exch_hdr"
option_id = "smc_opt"
has_name_select = "n"
name = "Medium Changer"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
type = ""
ghost = "y"
cmd_to_classify = ""
cmd_to_classify_postfix = ""
raw_field_name = "logicname"
cooked_field_name = ""
next_type = "d"
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_hdr:
id = "smc_exch_hdr"
option_id = "smc_exch_opt"
has_name_select = "y"
name = "Medium Changer Exchanging Tapes"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
cmd_to_exec = "tapeutil"
ask = "n"
exec_mode = ""
ghost = "n"
cmd_to_discover = ""
cmd_to_discover_postfix = ""
name_size = Ø
value_size = Ø
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "010"
id = "smc_exch_opt"
disc_field_name = "logicname"
name = "Medium Changer"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = ""
entry_type = ""
entry_size = Ø
required = "y"
prefix = "" /dev/
cmd_to_list_mode = ""
cmd_to_list = ""
cmd_to_list_postfix = ""
multi_select = ""
value_index = Ø
disp_values = ""
values_msg_file = ""
values_msg_set = Ø
values_msg_id = Ø
aix_values = ""
help_msg_id = ""
help_msg_loc = ""
help_msg_base = ""
help_msg_book = ""

sm_cmd_opt:
id_seq_num = "020"
id = "smc_exch_opt"
disc_field_name = "slotnumber"
name = "Insert Tape From Slot Number"
name_msg_file = ""
name_msg_set = Ø
name_msg_id = Ø
op_type = "l"
entry_type = "#"
entry_size = Ø
required = "+"
prefix = "exchange"
cmd_to_list_mode = "1"
cmd_to_list = "callme() \n{
    tapeutil -f /dev/\$1 inventory 2>/dev/null | grep -p 'Media Present.*Yes' | grep '^[A-Z]' | awk '{if(NR>1)print \$NF,\\"is a\",\$1,\$2}'\n} \n" 
callme "

 cmd_to_list_postfix = "logicname"
 multi_select = "*"
 value_index = 0
 disp_values = ""
 values_msg_file = ""
 values_msg_set = 0
 values_msg_id = 0
 aix_values = ""
 help_msg_id = ""
 help_msg_loc = ""
 help_msg_base = ""
 help_msg_book = ""

sm_cmd_opt:
 id_seq_num = "030"
 id = "smc_exch_opt"
 disc_field_name = "slotnumber"
 name = "Into The Slot Number"
 name_msg_file = ""
 name_msg_set = 0
 name_msg_id = 0
 op_type = "I"
 entry_type = "#"
 entry_size = 0
 required = "+"
 prefix = " 
 cmd_to_list_mode = "1"
 cmd_to_list = "callme() \n{
    tapeutil -f /dev/\$1 inventory 2>/dev/null | grep '^[A-Z]' | awk '{if(NR>1)print \$NF,\\"is a\",\$1,\$2}'\n} \n" 
callme "

 cmd_to_list_postfix = "logicname"
 multi_select = "*"
 value_index = 0
 disp_values = ""
 values_msg_file = ""
 values_msg_set = 0
 values_msg_id = 0
 aix_values = ""
 help_msg_id = ""
 help_msg_loc = ""
 help_msg_base = ""
help_msg_book = ""
sm_cmd_opt:
  id_seq_num = "040"
  id = "smc_exch_opt"
  disc_field_name = "slotnumber"
  name = "After Moving Any Present Tape To Slot"
  name_msg_file = ""
  name_msg_set = Ø
  name_msg_id = Ø
  op_type = "1"
  entry_type = "#"
  entry_size = Ø
  required = "+"
  prefix = ""
  cmd_to_list_mode = "1"
  cmd_to_list = "callme() \n |
    tapeutil -f /dev/\$1 inventory 2>/dev/null | grep -p 'Media Present.*No' | grep '^[A-Z]' | awk '{if(NR>1)print \$NF,\"is a",\$1,\$2}'
    callme"
  \n |
  cmd_to_list_postfix = "loginame"
  multi_select = ""
  value_index = Ø
  disp_values = ""
  values_msg_file = ""
  values_msg_set = Ø
  values_msg_id = Ø
  aix_values = ""
  help_msg_id = ""
  help_msg_loc = ""
  help_msg_base = ""
  help_msg_book = ""

***
/usr/bin/odmadd /tmp/smitty.add
return=?
/usr/bin/rm -f /tmp/smitty.add

# Allow easy removal of the new menu entries

# Allow easy removal of the new menu entries

cat <<<"$HOME/not$NAME

ODMDIR=/usr/lib/objrepos
odmdelete -o sm_name_hdr -q id=smc_query
odmdelete -o sm_name_hdr -q id=smc_path
odmdelete -o sm_name_hdr -q id=rmt_rr
odmdelete -o sm_name_hdr -q id=rmt_sense
odmdelete -o sm_name_hdr -q id=smc_audit
odmdelete -o sm_name_hdr -q id=smc_mount
odmdelete -o sm_name_hdr -q id=smc_unmount


44
odmdelete -o sm_name_hdr -q id=smc_move
odmdelete -o sm_name_hdr -q id=smc_exch
odmdelete -o sm_cmd_hdr -q id=smc_query_hdr
odmdelete -o sm_cmd_hdr -q id=smc_path_hdr
odmdelete -o sm_cmd_hdr -q id=rmt_rr_hdr
odmdelete -o sm_cmd_hdr -q id=rmt_sense_hdr
odmdelete -o sm_cmd_hdr -q id=smc_audit_hdr
odmdelete -o sm_cmd_hdr -q id=smc_mount_hdr
odmdelete -o sm_cmd_hdr -q id=smc_unmount_hdr
odmdelete -o sm_cmd_hdr -q id=rmt_opt
odmdelete -o sm_cmd_opt -q id=rmt_clear_opt
odmdelete -o sm_cmd_opt -q id=rmt_loaded_opt
odmdelete -o sm_cmd_opt -q id=smc_opt
odmdelete -o sm_cmd_opt -q id=smc_query_opt
odmdelete -o sm_cmd_opt -q id=smc_path_opt
odmdelete -o sm_cmd_opt -q id=rmt_rr_opt
odmdelete -o sm_cmd_opt -q id=rmt_sense_opt
odmdelete -o sm_cmd_opt -q id=smc_mount_opt
odmdelete -o sm_cmd_opt -q id=smc_unmount_opt
odmdelete -o sm_cmd_opt -q id=smc_audit_opt
odmdelete -o sm_cmd_opt -q id=smc_move_opt
odmdelete -o sm_cmd_opt -q id=smc_exch_opt
odmdelete -o sm_cmd_opt -q id=smc_mount
odmdelete -o sm_cmd_opt -q id=smc_unmount

***

chmod a+x $HOME/not$NAME
echo "\n\tThe menus will be removed by "$HOME/not$NAME\".\n"
##########################################################################
echo "\tThe fastpath to your application is "$NAME"."
INTRODUCTION

As in C programming, a shell script can make use of a predefined library that would contain variable and function definitions. The attached library listing includes the following functions:

- IsNumeric
- IsInteger
- IsReal
- CheckDateFormat
- FormatDate
- AddDayToDate.

INCLUSION OF A LIBRARY IN A SCRIPT

The library must be included in the current shell as follows:

```sh
# at the beginning of script, use the . command to include the
# library in the current shell
.
```

Andreas Neuper
PROFI Engineering Systems (Germany) © Dr Andreas Neuper 2003
BUILD ON THE CURRENT LIBRARY

There are six functions defined in this library. The definition of these functions can be expanded to implement further requirements.

For example:

```
AddDayToDate < date> < days to added or subtracted >
```

where `date` must be in the format DDMMYYYY and `days` must be a positive or negative integer.

Now the constraint on input date format can be expanded to include the following formats:

- YYYYMMDD
- DD-MON-YYYY

Using a similar algorithm, you can develop a function called AddMonToToDate and so on.

LIBRARY.KSH

```
#!/bin/ksh
#########################################################################
# Name     : library.ksh
# Overview : The script is a library of functions.
# Notes    : 1. The following functions are included in the script:
#             o IsNumeric
#             o IsInteger
#             o IsReal
#             o CheckDateFormat
#             o FormatDate
#             o AddDayToDate
#             2. Functions return either $TRUE, $FALSE, NULL or a value.
#             3. These functions will display error message(s) if the
#                calling script defines and exports following variable
#                as follows:
#                export DEBUG=Ø
#########################################################################
# Name     : InitialiseVariables
# Overview : The function initializes variables.
# Input    :
# Returns  :
# Notes    : 1. This is a function internal to the library.
```

InitialiseVariables ()
{
  # return codes
  TRUE=0
  FALSE=1
  META_CHARS="\$*;\%/\!&()=._\"\@$/£><'\`
  FUNCTION=""
  MODULE="library.ksh"
  INFO="\${MODULE}:\${INFO}:
  ERROR="\${MODULE}:\${FUNCTION}:ERROR:"
  NULL_STRING="Empty string"
  NULL_DATE_STRING="Empty date string"
  NULL_NO_DAYS="No of days is null"
  NULL_FORMAT_STRING="Empty date format string"
  NULL_FROM_DATE_FORMAT="Empty from date format string"
  INVALID_FROM_DATE_FORMAT="Invalid from date format"
  INVALID_TO_DATE_FORMAT="Invalid to date format"
  MINVALID_CHARS_IN_STRING="String contains meta character(s)"
  INVALID_FIRST_CHAR_IN_STRING="First character in string is invalid"
  INVALID_CHAR_IN_STRING="String contains non-numeric character(s)"
  STRING_NOT_NUMERIC="String contains non-numeric characters"
  NO_DECIMAL_PLACES="String contains no decimal places"
  TOO_MANY_DECIMAL_POINTS="String contains too many decimal points"
  DECIMAL_POINT_MISSING="String contains no decimal point"
  INVALID_MONTH="Invalid month"
  INVALID_DAY="Invalid day"
  TOO_MANY_DIGITS="Date contains too many digits"
  NOT_ENOUGH_DIGITS="Date contains not enough digits"
  INVALID_DATE_FORMAT="Invalid required date format"
  INVALID_NO_ARGS="Invalid number of arguments supplied"
}

# Name: DisplayMessage
# Overview: The function displays message
# Input: 1. Message type (E = Error, I = Informative)
# Notes: 2. Error Code as defined in DefineMessages().
DisplayMessage ( )
{
  MESSAGE_TYPE=$1
  MESSAGE_TEXT='eval echo $2'
  # re-evaluate $ERROR prefix
  ERROR='eval echo $ERROR | sed s:///
  # display message
  echo "$ERROR$MESSAGE_TEXT"
  # re-initialize $FUNCTION
  FUNCTION=""
# re-define $ERROR prefix
ERROR="${MODULE}: \${FUNCTION}: ERROR:"

###########################################################################
# Name     : IsNumeric
# Overview : The function validates a string for being numeric.
# Input    : string
# Returns  : TRUE   if string contains numeric characters only
#           FALSE  otherwise
# Usage    : if ! IsNumeric     99x
#            then
#                echo "String is not numeric"
#            fi
# Notes    : 1. A numeric string will only contain digits 0 to 9
#           (eg 12345 88900 098763 00000)
###########################################################################

IsNumeric ()
{
    # define function name
    FUNCTION="IsNumeric"
    # assign parameter
    P_STRING="\$1"
    # validate against null
    if [ "\${P_STRING}" = "\"
    then
        if [ "${DEBUG}" = "${TRUE}" ]
        then
            DisplayMessage E "${NULL_STRING}"
        fi
        return $FALSE
    fi
    # look for unwanted characters
    LEN_BEFORE_P_STRING='echo "${P_STRING}\c" | wc -c'
    LEN_AFTER_P_STRING='echo "${P_STRING}\c" | tr -d "${META_CHARS}" | wc -c'
    if [ $LEN_BEFORE_P_STRING -ne $LEN_AFTER_P_STRING ]
    then
        if [ "${DEBUG}" = "${TRUE}" ]
        then
            DisplayMessage E "${META_CHARS_IN_STRING}"
        fi
        return $FALSE
    fi
    # examine each character
    LEN_P_STRING='echo "${P_STRING}\c" | wc -c'
    COLUMN_POS=1
    while [ $COLUMN_POS -le $LEN_P_STRING ]
    do
        NEXT_CHAR='echo "${P_STRING}" | cut -c$COLUMN_POS-$COLUMN_POS'
        if [ "$\{NEXT_CHAR\}" != "$Ø" ]
        then
            return $FALSE
        fi
        ((COLUMN_POS++))
    done
}

"${NEXT_CHAR}" != "1" -a "${NEXT_CHAR}" != "2" -a "${NEXT_CHAR}" != "3" -a "${NEXT_CHAR}" != "4" -a "${NEXT_CHAR}" != "5" -a "${NEXT_CHAR}" != "6" -a "${NEXT_CHAR}" != "7" -a "${NEXT_CHAR}" != "8" -a "${NEXT_CHAR}" != "9"
then
  if [ "${DEBUG}" = "${TRUE}" ]
  then
    DisplayMessage E "${STRING_NOT_NUMERIC}"
  fi
  return ${FALSE}
fi
COLUMN_POS='expr $COLUMN_POS + 1'
done
FUNCTION=""
return $TRUE
}
###########################################################################
#  Name     : IsReal
#  Overview : The function validates a string for being a real number.
#  Input    : string
#  Returns  : TRUE   if string is a real number
#             FALSE  otherwise
#  Usage    : if !  IsReal   1ØØ
#             then
#                  echo "Invalid real number"
#             fi
#  Notes    :1. The range of real number is
#                 -0.1  to 99999999.99999999
#            2. Examples
#                Positive  real numbers = 0.0, 1.2 , 2.333  4.5566
#                Negative  real numbers = -0.0, -1.2 , -2.333 -4.5566
###########################################################################
IsReal ()
{
  # define function name
  FUNCTION="${FUNCTION}:IsReal"
  # assign parameter
  P_STRING="$1"
  # validate against null
  if [ "${P_STRING}" = "" ]
  then
    if [ "${DEBUG}" = "${TRUE}" ]
    then
      DisplayMessage E "${NULL_STRING}"
    fi
    return $FALSE
  fi

  NEGATIVE_REAL_NUMBER="${FALSE}"
  # check for meta characters
LEN_BEFORE_P_STRING='echo "${P_STRING}\c" | wc -c'
LEN_AFTER_P_STRING='echo "${P_STRING}\c" | tr -d "${META_CHARS}" | \wc -c'
if [ $LEN_BEFORE_P_STRING -ne $LEN_AFTER_P_STRING ]
then
  if [ "${DEBUG}" = "${TRUE}" ]
  then
    DisplayMessage E "${META_CHARS_IN_STRING}"
  fi
  return $FALSE
fi
# examine each character
LEN_P_STRING='echo "${P_STRING}\c" | wc -c'
COLUMN_POS=1
while [ $COLUMN_POS -le $LEN_P_STRING ]
do
  NEXT_CHAR='echo "${P_STRING}" | cut -c$COLUMN_POS-$COLUMN_POS'
  if [ $COLUMN_POS -eq 1 ]
  then
    # consider - sign and allowed characters
    if [ "${NEXT_CHAR}" != "-" -a "${NEXT_CHAR}" != "Ø" -a "${NEXT_CHAR}" != "1" -a "${NEXT_CHAR}" != "2" -a "${NEXT_CHAR}" != "3" -a "${NEXT_CHAR}" != "4" -a "${NEXT_CHAR}" != "5" -a "${NEXT_CHAR}" != "6" -a "${NEXT_CHAR}" != "7" -a "${NEXT_CHAR}" != "8" -a "${NEXT_CHAR}" != "9" -a "${NEXT_CHAR}" != "." ]
    then
      if [ "${DEBUG}" = "${TRUE}" ]
      then
        DisplayMessage E "${INVALID_FIRST_CHAR_IN_STRING}"
      fi
      return $FALSE
    elif [ "${NEXT_CHAR}" = "-" ]
    then
      NEGATIVE_REAL_NUMBER=${TRUE}
    fi
  else
    # consider allowed characters
    if [ "${NEXT_CHAR}" != "Ø" -a "${NEXT_CHAR}" != "1" -a "${NEXT_CHAR}" != "2" -a "${NEXT_CHAR}" != "3" -a "${NEXT_CHAR}" != "4" -a "${NEXT_CHAR}" != "5" -a "${NEXT_CHAR}" != "6" -a "${NEXT_CHAR}" != "7" -a "${NEXT_CHAR}" != "8" -a "${NEXT_CHAR}" != "9" -a "${NEXT_CHAR}" != "." ]
    then
      if [ "${DEBUG}" = "${TRUE}" ]
      then
        DisplayMessage E "${INVALID_CHAR_IN_STRING}"
      fi
      return $FALSE
    fi
  fi
else
  # consider allowed characters
  if [ "${NEXT_CHAR}" != "Ø" -a "${NEXT_CHAR}" != "1" -a "${NEXT_CHAR}" != "2" -a "${NEXT_CHAR}" != "3" -a "${NEXT_CHAR}" != "4" -a "${NEXT_CHAR}" != "5" -a "${NEXT_CHAR}" != "6" -a "${NEXT_CHAR}" != "7" -a "${NEXT_CHAR}" != "8" -a "${NEXT_CHAR}" != "9" -a "${NEXT_CHAR}" != "." ]
  then
    if [ "${DEBUG}" = "${TRUE}" ]
    then
      DisplayMessage E "${INVALID_CHAR_IN_STRING}"
    fi
    return $FALSE
  fi
fi
COLUMN_POS='expr $COLUMN_POS + 1'
done
# string must contain one decimal point
LEN_BEFORE_P_STRING='echo "${P_STRING}\c" | wc -c'
LEN_AFTER_P_STRING='echo "${P_STRING}\c" | tr -d "." | wc -c'
if [ 'expr $LEN_BEFORE_P_STRING - $LEN_AFTER_P_STRING' -eq 0 ]
then #
    # number does not contain decimal points
    if [ "${DEBUG}" = "${TRUE}" ]
    then
        DisplayMessage E "${DECIMAL_POINT_MISSING}"
    fi
    return $FALSE
elif [ 'expr $LEN_BEFORE_P_STRING - $LEN_AFTER_P_STRING' -gt 1 ]
then #
    # number contain too many decimal points
    if [ "${DEBUG}" = "${TRUE}" ]
    then
        DisplayMessage E "${TOO_MANY_DECIMAL_POINTS}"
    fi
    return $FALSE
fi
# number must have at least one decimal place ( eg. 1.2 )
DECIMAL_PARTS='echo "${P_STRING}" | cut -d'.' -f2'
if [ 'echo "${DECIMAL_PARTS}\c" | wc -c' -lt 1 ]
then #
    # number does not contain minimum number of decimal places
    if [ "${DEBUG}" = "${TRUE}" ]
    then
        DisplayMessage E "${NO_DECIMAL_PLACES}"
    fi
    return $FALSE
fi
FUNCTION=""
return $TRUE
#
# Name : IsInteger
# Overview : The function validates a string for being an integer.
# Input : string
# Returns : TRUE if string is an integer
#        : FALSE otherwise
# Usage : if ! IsInteger 12.22
#        : then
#        :    echo "Invalid integer"
#        : fi
# Notes : 1. The valid integer range is

© 2003. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.
# -1 to 99999999 (whatever is allowed by ksh)
# 2. Examples
# Positive integers = 0, 1, 2, 8888
# Negative integers = 0, -1, -22, -8888

IsInteger ()
{
FUNCTION=${FUNCTION}:IsInteger
# assign parameter
P_STRING="$1"
# validate against null
if [ "${P_STRING}" = "" ]
then
  if [ "${DEBUG}" = "${TRUE}" ]
  then
    DisplayMessage E "${NULL_STRING}"
  fi
  return $FALSE
fi
LEN_BEFORE_P_STRING=`echo "${P_STRING}\c" \ wc -c`
LEN_AFTER_P_STRING=`echo "${P_STRING}\c" \ tr -d "${META_CHARS}" \ wc -c`
if [ $LEN_BEFORE_P_STRING -ne $LEN_AFTER_P_STRING ]
then
  if [ "${DEBUG}" = "${TRUE}" ]
  then
    DisplayMessage E "${META_CHARS_IN_STRING}"
  fi
  return $FALSE
fi
LEN_P_STRING=`echo "${P_STRING}\c" \ wc -c`
# examine each character
NEGATIVE_INTEGER=${FALSE}
COLUMN_POS=1
while [ $COLUMN_POS -le $LEN_P_STRING ]
do
  NEXT_CHAR=`echo "${P_STRING}" \ cut -c$COLUMN_POS-$COLUMN_POS`
  if [ $COLUMN.Pos -eq 1 ]
  then
    # consider - sign and allowed characters
    if [ "${NEXT_CHAR}" != "-" -a "${NEXT_CHAR}" != "Ø" -a "${NEXT_CHAR}" != "1" -a "${NEXT_CHAR}" != "2" -a "${NEXT_CHAR}" != "3" -a "${NEXT_CHAR}" != "4" -a "${NEXT_CHAR}" != "5" -a "${NEXT_CHAR}" != "6" -a "${NEXT_CHAR}" != "7" -a "${NEXT_CHAR}" != "8" -a "${NEXT_CHAR}" != "9" ]
    then
      if [ "${DEBUG}" = "${TRUE}" ]
      then
        DisplayMessage E "${INVALID_FIRST_CHAR_IN_STRING}"
      fi
    fi
  fi
  COLUMN_POS=$COLUMN_POS+1
  NEGATIVE_INTEGER=${NEGATIVE_INTEGER}+1
done
if [ $NEGATIVE_INTEGER -eq 0 ]
then
  return $TRUE
fi
return $FALSE
}
fi
return ${FALSE}
elseif ['${NEXT_CHAR}' = '.' ]
then
    NEGATIVE_INTEGER=${TRUE}
fi
else

Editor’s note: this article will be concluded next month.

Arif Zaman
ETL Developer (UK)  © Xephon 2003
Brio has announced its Performance Suite 8.1 with support for AIX, HP-UX, and Solaris and localized Windows versions in nine languages.

It promises enhanced Web-based application services by supporting Apache 1.3 or higher, Sun ONE Application Server, WebLogic 6.x SP3 or higher, and WebSphere 4.0.3 or higher.

The nine localized languages supported include Chinese - Simplified and Traditional, French, German, Italian, Japanese, Korean, Portuguese Brazilian, and Spanish.

For further information contact:
Brio, 4980 Great America Parkway, Santa Clara, CA 95054, USA.
Tel: (408) 496 7400.

** * **

Gresham Computing has announced that ADIC customers can now implement Tivoli Storage Manager (TSM) LAN-free software using Gresham’s Enterprise DistribuTape (EDT) tape storage management integration application.

EDT works with the Tivoli Storage Manager and ADIC’s storage networking library architecture to provide high performance SAN back-up that shares resources, reduces network congestion, and simplifies management, giving better data protection.

The software provides a range of LAN-free features and library management functionality for ADIC libraries operating in TSM back-up environments.

It manages communications between TSM and the library or media manager and augments TSM’s native support for ADIC libraries with an array of feature enhancements, which allow multiple TSM servers and LAN-Free clients to share multiple tape drives dynamically and streamline media management processes.

The software is certified and supported by Tivoli Storage Software and can be purchased through Gresham or IBM and its Business Partners. Supported platforms include AIX, Solaris, HP-UX, and Windows NT and 2000. It’s compatible with all ADIC AML, Scalar 10K, and Scalar 1000 storage networking libraries.

For further information contact:
Gresham, 28 Queen Street, London, EC4R 1BB, UK.
Tel: (020) 7653 0200.

** * **

Micromuse has announced Netcool/Precision for Transmission Networks Version 5.3, providing auto-discovery and dynamic topology/connectivity modelling of both IP and transmission networks. The expanded range now offers dynamic network asset and inventory management, topology visualization and mapping, and topology-based event correlation (RCA).

Netcool/Precision for Transmission Networks v5.3 runs on AIX.

For further information contact:
Micromuse, 139 Townsend St, San Francisco, CA 94107, USA
Tel: 415 538 9090.