HP DDS drives

technical reference manual

volume 5 : unix configuration

DDS Evolution II drives:
HP C1537A DDS-3 drive (24 GB)—USB
HP C5683A DDS-4 drive (40 GB)—USB
HP C7438A DAT-72 drive (72 GB)—USB and SCSI

Part number: DW007–90901 volume 5
Edition 1.1: November 2005

Legal and notice information

HP Evolution II DDS drives technical reference manual, volume 5 : unix configuration

Contents

About this guide

NOTE: DDS Evolution II drives, available mid-2005 with U160 SCSI or USB 2.0 interfaces, are identifiable by the round LEDs on the front panel, as opposed to rectangular or oval.

USB drives are only supported on Linux (and Windows).

This manual contains information on connecting to various operating systems. The information is given in good faith, but since the operating systems and any upgrades that are made to them are outside Hewlett-Packard's control, HP cannot guarantee that the details are correct. Please consult the operating system documentation in conjunction with this manual.

This volume provides basic information on configuring the following drives with various operating systems:

- HP C1537A DDS-3 drive, USB, capacity 24 GB
- HP C5683A DDS-4 drive, USB, capacity 40 GB
- HP C7438A DAT 72 drive, SCSI and USB, capacity 72 GB

The capacities use hardware data compression with a compression ratio of 2:1.

HP DDS technical manual

The 6-volume HP DDS Technical Manual also includes the following:

- *Hardware Integration Guide*, volume 1
- *Software Integration Guide*, volume 2
- *The SCSI Interface*, volume 3
- *Specifications*, volume 4
- *Background to DDS Products*, volume 6

Please contact your HP supplier for copies.

Documentation map

To find where information you need is in the Manual, the following may help:

Drives —general	Electronics	**6** Background: *ch.* 2
	Front panel—drive	**1** HW Integration: *ch.* 4
	Mechanism	**6** Background: *ch.* 1
	Overview of all drives	**1** HW Integration: *ch.* 1
	Specifications	**4** Specs
	Supplies	**1** HW Integration: *ch.* 1
	USB drives	**1** HW Integration: *ch.* 2 **6** Background: *ch.* 4

Installation and Configuration	Airflow & Cooling	**1** HW Integration: *ch.* 3
	Connectors	**1** HW Integration: *ch.* 3
	Determining the configuration	**2** SW Integration: *ch.* 2
	Installation	**1** HW Integration: *ch.* 3
	Power-on initialization	**1** HW Integration: *ch.* 6
	Reset initialization	**1** HW Integration: *ch.* 6
	UNIX configuration	**5** UNIX Config
	USB networks	**1** HW Integration: *ch.* 2

Operation	Drives	**1** HW Integration: *ch.* 4

Cartridges	Dealing with cartridges through software	**2** SW Integration: *ch.* 3
	Managing the use of cartridges	**2** SW Integration: *ch.* 1
	Using cartridges	**1** HW Integration: *ch.* 5

Interface	SCSI Guide	**3** SCSI
	Commands	**3** SCSI: *ch.* 5
	Drive error codes	**1** HW Integration: *ch.* 7
	Implementation	**3** SCSI: *ch.* 1
	Interpreting sense data	**2** SW Integration: *ch.* 3
	Logs—see the LOG SENSE command	**3** SCSI: *ch.* 5
	Messages	**3** SCSI: *ch.* 3
	Mode pages—see the MODE SENSE command	**3** SCSI: *ch.* 5
	Pre-execution checks	**3** SCSI: *ch.* 4
	Responding to Sense Keys and ASC/Q	**2** SW Integration: *ch.* 8
	SCSI over USB	**3** SCSI: *ch.* 2
	Sense Keys and ASC/Q—see REQUEST SENSE command	**3** SCSI: *ch.* 5
	USB interface	**6** Background: *ch.* 4

Maintenance and Troubleshooting	Cleaning	**1** HW Integration: *ch.* 4
	Diagnostics	**1** HW Integration: *ch.* 6
	Firmware upgrade	**1** HW Integration: *ch.* 4
	Forcing ejection	**2** SW Integration: *ch.* 9
	Software troubleshooting techniques	**2** SW Integration: *ch.* 1
	Troubleshooting the drive	**1** HW Integration: *ch.* 8

Dealing with Errors	Error Codes	**1** HW Integration: *ch.* 7
	Handling errors	**2** SW Integration: *ch.* 5
	How error correction works	**6** Background: *ch.* 5
	Logs—see the LOG SENSE command	**3** SCSI: *ch.* 5
	Responding to errors	**2** SW Integration: *ch.* 9
	Software response to error correction	**2** SW Integration: *ch.* 3
	Software response to logs	**2** SW Integration: *ch.* 3
	TapeAlert log	**2** SW Integration: *ch.* 9

DDS Features	Data compression, controlling	**1** HW Integration: *ch.* 3
	Data compression, how it works	**6** Background: *ch.* 6
	Data compression, managing	**2** SW Integration: *ch.* 7
	DDS format	**6** Background: *ch.* 3
	Fast-searching, how it works	**6** Background: *ch.* 5
	Fast-searching, supporting	**2** SW Integration: *ch.* 5
	Indexing, how it works	**6** Background: *ch.* 5
	Load and unload	**2** SW Integration: *ch.* 9
	Load and unload, timings and time-out values	**2** SW Integration: *ch.* 6
	Partitioning	**2** SW Integration: *ch.* 5
	Performance optimization	**2** SW Integration: *ch.* 1
	Performance, factors affecting	**2** SW Integration: *ch.* 4
	Software design	**2** SW Integration: *ch.* 1
	Supporting DDS features	**2** SW Integration: *ch.* 5

Related documents

The following documents provide additional information:

General documents and standardization

- Small Computer System Interface (SCSI-1), ANSI X3.131-1986. This is the INCITS authorized standard for SCSI implementation, available through INCITS
- Enhanced Small Computer System Interface (SCSI-2), ANSI X3T9.2-1993 Rev. 10L, available through INCITS
- DDS-3
  - ECMA-236
- DDS-4
  - ECMA-288
- DAT 72

- "3.81 mm Wide Magnetic Tape Cartridge for Information Exchange - Helical Scan Recording DAT 72 Format using 170m Length Tapes" — controlled HP document

Copies of General Documents can be obtained from:

INCITS	11 West 42nd Street, New York, NY 10036-8002, USA
ISO	CP 56, CH-1211 Geneva 20, Switzerland
ECMA	114 Rue du Rhône, CH-1204 Geneva, Switzerland *Tel:* +41 22 849 6000 *Web URL:* http://.www.ecma.ch
Global Engineering Documents	2805 McGaw, Irvine, CA 92714, USA *Tel:* 800 854 7179 or 714 261 1455

USB Specifications

- *Universal Serial Bus Specification* Revision 2.0 April 27, 2000
- *Universal Serial Bus Mass Storage Class Specification Overview* Revision 1.2 June 23, 2003
- *Universal Serial Bus Mass Storage Class Specification—Mass Storage Class—Bulk Only Transport* Revision 1.0 September 31, 1999

These can be obtained from:

USB Implementers Forum, Inc.
5440 S.W. Westgate Drive, Suite 217
Portland, OR 97221 U.S.A.
Tel: 503-296-9892
Fax: 503-297-1090
Web: www.usb.org
Email: admin@usb.org

1 Introduction

Drivers and backup software

Drivers

All supported UNIX operating systems provide native driver support for HP DDS tape drives. In some cases the drivers require configuration to perform optimally.

UNIX applications

You can write scripts to control DDS drives in UNIX using standard backup utilities such as `cpio` and `tar`. To achieve more sophisticated control of the drives, and to exploit the full range of DDS features, it is worth considering software applications specifically designed for the task.

Application software availability

Most backup software companies provide applications for HP DDS products. Contact your software supplier for details. Alternatively, contact your HP supplier, who can provide you with details of a wide range of compatible software.

2 HP Alpha UNIX 5.1x *(DAT 72 drives only)*

NOTE: Only DAT 72 SCSI drives are currently supported on HP Alpha Tru64 systems.

Updating the tape driver

1. Modify the SCSI Tape Density Table to include:

```
scsi_tape_density[0x47] =   "163000_bpi"       163000      0 (DAT72)
```

2. Add the following entry to your `/dev/ddr.dbase` file:

```
SCSIDEVICE
   Type = tape
   Name = "HP" "C7438A"
   #
PARAMETERS:
   TypeSubClass        = rdat
   TagQueueDepth       = 0
   MaxTransferSize     = 0x0ffffff      # (16MB - 1)
   ReadyTimeSeconds    = 120            # seconds

MODESELECT:
   ModeSelectNumber = 0
   SavePage = No
   PageFormat = scsi2
   BlockDescriptor = yes
   TransferLength = 16
   Hdr.Tape.BufferMode = 0x1
   Data.UBYTE[0] = 0x3D  #Vendor Unique Page Code 0x3D
   Data.UBYTE[1] = 0x02
   Data.UBYTE[2] = 0x01

DENSITY:
   DensityNumber = 0,3,4,5,6,7
   DensityCode = default
   CompressionCode = 0x0
   Buffered = 0x1

DENSITY:
   DensityNumber = 1,2
   DensityCode = default
   CompressionCode = 0x1
   Buffered = 0x1
```

3. Rebuild the kernel by running:

```
/sbin/ddr_config -c /etc/ddr.dbase
```

then reboot the system with the tape drive attached. The device files for the DAT 72 drive will be generated in `/dev/tape` and `/dev/ntape` when you reboot.

4. The names of the device files can be interpreted as follows:

   Devices in the `/dev/ntape` directory are "no-rewind" devices, those in `/dev/tape` will do a rewind on close.

   The device files then have the syntax, `tapeX_dn`

   where:

   `X` is the instance of the drive
   `n` is the density number

   For example, `/dev/ntape/tape66_d1` is a device file for device 66, no-rewind using density number 1. Since all density numbers have the same parameters it does not matter which density number file is used.

What next?

Once the device files have been created, you should confirm that your new tape drive is working properly. Chapter 7, "Verifying the installation" provides instructions on backing up and restoring a sample file to test your installation.

3 HP servers and workstations — HP-UX 11.*x*

NOTE: Only DAT72 SCSI drives are supported on HP-UX.

Before you install your tape drive log on to the HP web site, www.hp.com, and download the latest hardware enablement patch bundle for your operating system. This ensures that you will have the correct device driver for your tape drive.

Determining the SCSI ID

Before you configure your system to support your new HP drive, you need to determine what SCSI ID to use. The SCSI ID must be unique for each device attached to the SCSI bus. To list the existing devices, use the following command:

```
% /sbin/ioscan -f
```

The output of this should look similar to the following example:

```
Class     I H/W Path    Driver     S/W State H/W Type  Description
=============================================================================
bc        0             root         CLAIMED  BUS_NEXUS
bc        1 8           bc           CLAIMED  BUS_NEXUS Psudo Bus Converter
ba        0 8/0         GSCtoPCI     CLAIMED  BUS_NEXUS GSCtoPCI Bridge
ext_bus   1 8/0/2/0     c720         CLAIMED  INTERFACE SCSI C895 Ultra2 Wide LVD
target    0 8/0/2/0.7   tgt          CLAIMED  DEVICE
ctl       1 8/0/2/0.7.0 sctl         CLAIMED  DEVICE    Initiator
lan       0 8/0/20/0    btlan3       CLAIMED  INTERFACE PCI(10110019) -- Built-in #1
ba        1 8/16        bus_adapter  CLAIMED  BUS_NEXUS Core I/O Adapter
tty       0 8/16/4      asio0        CLAIMED  INTERFACE Built-in RS-232C
ext_bus   2 8/16/5      c720         CLAIMED  INTERFACE Built-in SCSI
target    1 8/16/5.5    tgt          CLAIMED  DEVICE
disk      0 8/16/5.5.0  sdisk        CLAIMED  DEVICE    SEAGATE ST34573N
target    2 8/16/5.7    tgt          CLAIMED  DEVICE
ctl       2 8/16/5.7.0  sctl         CLAIMED  DEVICE    Initiator
processor 0 62          processor    CLAIMED  PROCESSOR Processor
memory    0 63          memory       CLAIMED  MEMORY    Memory
```

After you have installed the new tape drive, you can check that it has been attached successfully. From a shell window (`hpterm/xterm`), execute `ioscan` to display the list of attached devices:

```
% /sbin/ioscan -C tape -fn
```

The new lines should look similar to the following, where the `4` in the `I` field represents the instance of the SCSI tape driver, not the SCSI ID:

```
tape       4  2/0/1.5.0  stape        CLAIMED    DEVICE      HP       C7438A
```

If you cannot find the drive, this may be because the kernel does not contain the correct driver. Use the System Administration Manager (`sam`) to add `stape` to the kernel:

To add stape to the kernel using sam:

1. `% sam`
2. Select the following:

   ```
   Kernel Configuration
   Drivers
   ```
3. Highlight the `stape` driver. If the driver has not been added to the kernel, both Current State and Pending State will read “Out”.
4. Select the following:

   ```
   Actions
   Add Driver to Kernel
   ```
   The Pending State will now read “In”.
5. To add the new driver to the kernel, select:

   ```
   Actions
   Create a New Kernel
   ```
6. The `stape` driver will now be added to the kernel and then the system will reboot.

Creating the device files

Once you have verified the tape drive connection, you will need to create the appropriate device files for the drive. Normally, you would have rebooted your system after attaching the tape drive, and this process runs `insf`. However, if you have not rebooted your system since attaching the drive, you can create device files by one of two ways, either through the System Administration Manager (`sam`), or by executing the `mksf` command.

To add device files using sam:

This is the recommended and simplest way to create device files.

1. `% sam`

   This will bring up the graphical user interface for the utility.
2. Select the following:

   ```
   Peripheral Devices
   Tape Drives
   ```
   `sam` will then scan the system for any tape drives connected.

   When a drive is found, it will be displayed as:

   ```
   Hardware Path       Driver          Description
   ============================================================
   8/0/2/0.3.0         stape           HP      C7438A
   ```
3. Highlight the drive and select the following from the tool bar:

   ```
   Actions
   Create Device Files
   Create Default Device Files
   ```

This will create default device files for the drive. To view the device files that have been created, select:

```
Actions
Create Device Files
Show Device Files
```

where:

Device File	Description
where <I>	is the instance number of the drive
<I>m	AT&T encoding, rewind driver
<I>mn	AT&T encoding, non-rewind driver
<I>mb	Berkeley encoding, rewind driver
<I>mnb	Berkeley encoding, rewind driver
where <X>	is the card number
<Y>	is the target number
<Z>	is the LUN number
cXtYbZBEST	Best compression driver, AT&T encoding, with rewind
cXtYbZBESTb	Best compression driver, Berkeley encoding, with rewind
cXtYbZBESTn	Best compression driver, AT&T encoding, non-rewind
cXtYbZBESTnb	Best compression driver, Berkeley encoding, non-rewind

4. When you have exited `sam`, run `ioscan` to see the tape drive:

```
%/sbin/ioscan -C tape -fn
```

To create device files using mksf:

NOTE: This method is *not* recommended.

1. Run `insf` as follows:

```
% /sbin/insf -C tape
```

2. Create the device files for the devices using the `mksf` command as follows:

```
% /sbin/mksf -d stape -I <instance> [-n] [-u] /dev/rmt/X<name>
```

where:

Argument	Description
-d stape	Specifies the SCSI tape driver
-I <instance>	Specifies the tape drive's hardware address via the instance of the SCSI tape driver. The first instance is 0, the second 1, and so on.
[-n]	Specifies no rewind; absence of this parameter indicates rewind mode

Argument	Description		
[-u]	Specifies Berkeley mode; absence of this parameter indicates AT&T mode. Berkeley and AT&T modes differ in their read-only close behavior: • In Berkeley mode, the tape position will remain unchanged by a device close operation. • In AT&T mode, a device close operation will cause the tape to be repositioned just after the next tape filemark (the start of the next file). In most cases, Berkeley mode should be used.		
/dev/rmt/X<name>	Specifies the path of the device file, where:		
	X	Specifies the tape device identifier. Use the next available identifier. You can examine the contents of /dev/rmt using the ls command to determine which identifiers have already been used.	
	<name>	Specifies the short name (in HP-UX 9.x-style) of the device file:	
		mnb	No rewind, compression disabled, Berkeley-mode device
		hnb	No rewind, compression disabled, Berkeley-mode device
		mnb	No rewind, compression disabled, Berkeley-mode device
		hnb	No rewind, compression enabled, Berkeley-mode device

See the man page (man 1m mksf) for other options of the mksf command. The stape section covers the SCSI tape driver options. The man page man 7 mt describes the long filenames used in HP-UX 10.*x* and later.

Example

To create a device file with the following characteristics:

- A hardware address specified by instance 5 (-I 5)
- No rewind (-n)
- Berkeley mode tape positioning on close (-u)
- A filename of 4mnb, where 4 is the tape device identifier (/dev/rmt/4mnb)

You would execute the following:

```
% /sbin/mksf -d stape -I 4 -n -u /dev/rmt/4mnb
```

You can check that the appropriate device file was created using the lssf command as follows:

```
% /sbin/lssf /dev/rmt/4mnb
```

This should produce the following output to show that the device file now exists:

```
stape card instance 0 SCSI target 6 SCSI LUN 0 berkeley no rewind
BEST density at address 2/0/1.6.0 /dev/rmt/4mnb
```

To create a device file for a drive in uncompressed mode, you should use a command such as:

```
mksf -H -a -b U_18
```

and for compressed mode (default):

```
mksf -H -a -b U_18C
```

The hardware path can be found from previous `ioscan` output.

What next?

Once the device files have been created, you should confirm that your new tape drive is working properly. Chapter 7, "Verifying the installation" provides instructions on backing up and restoring a sample file to test your installation.

4 IBM (AIX) servers and workstations

NOTE: Only DAT72 SCSI drives are supported on AIX.

Determining the SCSI ID

Before you configure your system to support your drive, you need to determine which SCSI ID to use. IDs must be unique for each device attached to the SCSI bus. To list the existing devices, use the following command:

```
% lsdev -C |grep SCSI
```

This will produce output that looks similar to:

```
scsi0 Available 00-00-0S Standard SCSI I/O Controller
hdisk0 Available 00-00-0S-0 1.0 GB SCSI Disk Drive
rmt1 Defined 00-00-0S-2,0 Other SCSI Tape Drive
```

The SCSI ID is in the series `00-00-0S-X,0`, where *X* is the SCSI ID. Review the list of existing SCSI IDs and choose an available ID to assign to the new tape drive.

Configuring the device files

To install a DDS-format drive on an IBM server you need to create the appropriate device files for the drive.

NOTE: Do *not* choose the `smit` option of "`4mm2gb`" as the Tape Device Type. This is reserved for Connor drives. If you use it with HP drives, you will get the error "`Device to be configured does not match the physical device at the specified connection location`".

To change to variable block mode, use the following procedure:

1. *If you are using a graphics terminal* running X-Windows, then at a Windows terminal, type:
   `smit tape`
   If you are using a non-graphics terminal, at the command line type:
   `% smit -C tape`
2. If no device has been configured at this address before, select "`add a tape drive`" to set up the address. From the pop-up window, select "`ost`" or "`Other SCSI tape drive`" as the tape drive you wish to change and choose connection addresses as appropriate.
3. Set `maximum delay for the READ/WRITE command = 1200`.
4. Change the block size field to 0, and click on the "`DO`" button or press [Enter] to apply the change.

HP DDS-format drives will work with `tar`, `cpio`, `backup`, `restore` and `dd`. For some systems, the drive is also boot-capable, provided a boot tape is generated using `mkszfile` and `mksysb`.

What next?

Once the device files have been created, you should confirm that your new tape drive is working properly. Chapter 7, "Verifying the installation" provides instructions on backing up and restoring a sample file to test your installation.

Device filenames under AIX

Use device filenames as listed below for the combination of Rewind on Close, Retension on Open, and Compression that you want:

Filename	Rewind on Close	Retension on Open	Compression
/dev/rmtn	Yes	No	enabled
/dev/rmtn.1	No	No	enabled
/dev/rmtn.2	Yes	Yes	enabled
/dev/rmtn.3	No	Yes	enabled
/dev/rmtn.4	Yes	No	disabled
/dev/rmtn.5	No	No	disabled
/dev/rmtn.6	Yes	Yes	disabled
/dev/rmtn.7	No	Yes	disabled

The ***n*** in the filename is the instance number assigned to the drive by the operating system, where 0 is the first device, 1 is the second and so on.

Rewind on Close Normally, the drive repositions the tape to BOT (Beginning of Tape) when the device file is closed. Using the no rewind option is useful when creating and reading tapes that contain multiple files.

Retension on Open Retensioning consists of winding to EOT (End of Tape) and then rewinding to BOT, in order to reduce errors. If this option is selected, the tape is positioned at BOT as part of the open process.

Compression Compression can be disabled or enabled.

5 Linux servers and workstations

NOTE: Both SCSI and USB drives are supported on Linux.

Determining the SCSI ID

Look at the output of `dmesg` to find out what SCSI channel number is used for each connection.

To find out the SCSI IDs in use on each channel, type:

```
cat /proc/scsi/scsi
```

This will produce output similar to the following for each device:

```
Attached Devices
Host: SCSI0 Channel: 00 Id:00 Lun:00
Vendor: HP Model ------------
Type: Direct-Access ANSI SCSI Revision 02
```

Look at the ID information to establish which IDs are in use.

Configuring on Linux systems

No changes are needed to support DDS-format drives on Linux platforms, however you should ensure that you have the relevant drivers loaded.

To see the device drivers loaded currently, execute an `lsmod` command, this will give output like:

```
Module      Size     Used by
sgm         4376     1
ide-scsi    7200     0
lockd       30792    1
sunrpc      53316    1
st          24656    0
sym53c8xx    39696    1
aic79xx     186044   3
scsi_mod    100408   5 [ide-scsi st aic79xx mptscsih]
```

The lines of interest here are:

`st`	Tape driver. Its presence shows the driver is loaded.
`sym53c8xx`	SCSI chipset driver for the LSI Logic family of HBAs (among others).
`aix79xx`	SCSI chipset driver for the Adaptec 79*xx* chipset family (such as Adaptec 29320).

Latest SCSI controller drivers for Linux will be available from the manufacturer's web site.

In order to communicate with a tape device, the operating system needs to have drivers for the tape and the underlying transport mechanism (the host bus adaptor) loaded. Ensure that both are

available as either loadable modules (for example, usable with `insmod` and visible with `lsmod`) or are statically built into your kernel.

NOTE: In order to add drivers to the statically built kernel you need the Linux source code available on disk and knowledge of how to use the kernel building tools that ship with various Linux distributions. This should not be attempted by novice users.

In order to determine if the drive has been detected by the tape driver at module load time, execute:

```
dmesg | grep "st"
```

This should find a number of lines. One should look like:

```
Detected SCSI tape st0 at scsi1, channel 0, id 5, lun 0
```

To load the tape driver module if it is not loaded as above, execute:

```
insmod st
```

to load it. This should happen naturally if your system is rebooted after attaching the drive.

When the `ST` driver module has been added, a list of tape device files will be created automatically. They reside in the `/dev/` directory and have the syntax:

```
/dev/stp or dev/nstp
```

where:

p	Instance number of the device file (0 if only one drive is connected to the system)
n	`No-rewind driver`

In order to enable large transfers under Linux (>64 KB per write), edit the file `/usr/src/linux/drivers/scsi/st_options.h` and change the definition of `ST_BUFFER_BLOCKS`.

If you want requests to space to end of data to be faster, you should also enable `ST_FAST_MTEOM` in the same file. After changing this file, rebuild the modules and install the new binary. At the very least, this requires:

```
make modules
make modules_install
```

from the `/usr/src/linux` directory. See your kernel documentation.

Installing USB drivers on Linux

Two drivers are required in order to use HP DAT USB tape drives. These are included with the operating system and should be loaded automatically.

Use the following procedure to check that both drivers are present:

usb_storage driver

1. At the command prompt type:
   ```
   lsmod | grep usb_storage
   ```

2. The output of this command should contain a line similar to:
   `usb_storage 61193 0`
   If the line is not present type
   `modprobe usb_storage`
   at the command line to load the usb-storage driver.

st tape driver

1. At the command prompt type:
   `lsmod | grep st`
2. The output of this command should contain a line similar to:
   `st 35933 0`
   If the line is not present type:
   `modprobe st`
   at the command line to load the st driver.

What next?

Once the device files have been created, you should confirm that your new tape drive is working properly. Chapter 7, "Verifying the installation" provides instructions on backing up and restoring a sample file to test your installation.

6 Sun SPARC servers and workstations—Solaris

NOTE: Only DAT72 SCSI drives are supported on Solaris.

Determining the SCSI ID

Before you configure your system to support a DDS-format drives, you need to determine which SCSI ID to use. IDs must be unique for each device on attached to the SCSI bus.

1. Use the `modinfo` command to identify SCSI controller drivers installed on the system.

   For FAS or ESP devices:

   ```
   % modinfo | grep "HBA Driver"
   ```

   This will produce output similar to the following:

   ```
   104 78032000 12660 33 1 glm (glm SCSI HBA Driver)
   ```

   This indicates that there a GLM-based SCSI controller on the system. For the adapter to which the new tape drive is attached, you will need to determine what SCSI IDs are already used.
2. Determine the SCSI IDs of the existing devices attached to the SCSI controller:

   For all adapters:

   ```
   % dmesg | egrep ".*xxx.*target" | sort | uniq
   ```

   where xxx = the type of adapter (`esp`, `glm`, `fas` or `isp`), as appropriate.

   For example, for an GLM-based adapter:

   ```
   % dmesg | egrep ".*glm.*target" | sort | uniq
   ```

   This produces a list similar to:

   ```
   sd6 at glm0: target 6 lun 0
   ```

   This indicates that SCSI ID 6 is used for an existing device. SCSI ID 7 is generally used for the adapter itself. In this situation, you would use a SCSI ID from 1 to 5 for the new tape drive.

Driver configuration

NOTE: Drives should then work well with Solaris without modifications to the kernel, and you are recommended to try this.

Only if necessary, make the following file modifications to enhance performance:

1. In the file `/kernel/drv/st.conf`, after these lines:

   ```
       ########
   # Copyright (c) 1992, by Sun Microsystems, Inc.
   #ident  "@(#)st.conf    1.6     93/05/03 SMI"
   ```

   add the following (the spaces are significant in the strings):

for Solaris 2.6, 5.7 and 8 without the latest st patch:

```
tape-config-list =
 "HP      C7438A","HP DAT72 4mm DAT, "HP-DAT72";
 HP-DAT72 = 1,0x34,0,0xd639,1,0x47,0;
name="st" class="scsi"
        target=X lun=0;
```

where X is the SCSI target address of the device you have attached.

for Solaris 9 and 10 (and 8 with the latest st patch):

```
tape-config-list =
 "HP      C7438A","HP DAT72 4mm DAT, "HP-DAT72";
 HP-DAT72 = 1,0x34,0,0x18679,1,0x00,0,60,300,600,1200,600,600,18000;
name="st" class="scsi"
        target=X lun=0;
```

where X is the SCSI target address of the device you have attached.

See "HP-data values" on page 26 below for the values of the parameters in these lines.

2. If you are replacing an existing tape device on the same SCSI ID, remove the contents of the `/dev/rmt` directory as follows:

```
% cd /dev/rmt
% rm *
```

3. Instead of rebooting the device, follow these steps.
   a. Find the kernel module ID:

```
# modinfo | grep "st ("
96 60dcc000 cdb0 33 1 st (SCSI Sequential Access Driver)
```

   In this example the ID is 96.

   b. Unload the kernel module:

```
# modunload -i 96
```

   c. Load the kernel module back in:

```
# modload -p drv/st
```

   d. Rebuild the device paths:

```
devfsadm -C
devfsadm -i st
```

   For further details, see "How do you load st.conf changes without rebooting," SunSolve document 18010, on http://sunsolve.sun.com/search/document.do?assetkey=1-9-18010-1&searchclause=18010

4. You should now be able to use the drive.
   - Use `/dev/rmt/Xcb` if you require a compression rewind device file, where X is the relevant device address.
   - Use `/dev/rmt/Xcbn` when you require a compression non-rewind device.

HP-data values

The values for `HP-DAT72` and name, which provide normal DDS mode, have the following meanings.

The syntax for HP_DAT72 on Solaris 9, 10 (and 8 with the latest st patch) is:

```
<drive type> = <version>, <type>, <bsize>, <options>,
  <no. of densities>, <density 0>, <density 1>, ..., <density n>,
  <default density>, <non-motion timeout>, <read/write timeout>,
  <rewind timeout>, <space timeout>, <load timeout>,
  <unload timeout>, <erase timeout>
```

where:

Parameter	Value	Meaning			
<version>	1	Indicates the format of the following parameters.			
<type>	0x34	The value for a DAT drive in /usr/include/sys/mtio.h.			
<bsize>	0	Indicates variable block size.			
<options>	0xd639 or 0x18679	Derived from constants in /usr/include/sys/scsi/targets/stdef.h. The value determines which operations the driver can perform with the attached device by using a unique value for each feature and then adding them together to form the options value:			
			Options value:	0xd639	0x18679
		0x001	Device supports variable length records.	Yes	Yes
		0x008	Device can backspace over files (as in the 'mt bsf' option).	Yes	Yes
		0x010	Device supports backspace record (as in 'mt bsr').	Yes	Yes
		0x020	Device requires a long time-out period for erase functions.	Yes	Yes
		0x040	Device will automatically determine the tape density.	No	Yes
		0x0200	Device knows when end of data has been reached.	Yes	Yes
		0x0400	Device driver is unloadable.	Yes	Yes
		0x1000	Time-outs five times longer than normal.	Yes	No
		0x4000	Driver buffers write requests and pre-acknowledges success to application.	Yes	No
		0x8000	Variable record size not limited to 64 KB.	Yes	Yes
		0x10000	Device determines which of the two mode pages the device supports for selecting or deselecting compression.	No	Yes
<no. of densities>	1	There is one density code following in the parameter list.			
<density *n*>	0x47 or 0x00	Supported density code. The value of 0x00 used in the Solaris 8 (with the latest st patch), 9 and 10 method means use the default density chosen by the drive—which is 0x47.			

Parameter	Value	Meaning
<default density>	0	Density 0 (0x47) is the default.
<X timeout>		All timeouts are in seconds

Values for the parameters for name are as follows:

Parameter	Value	Meaning
target	X	X specifies the SCSI ID (target) of the device.
lun	0	Specifies the LUN for the device.

What next?

Once the device files have been created, you should confirm that your new tape drive is working properly. Chapter 7, "Verifying the installation" provides instructions on backing up and restoring a sample file to test your installation.

7 Verifying the installation

As part of the installation process, you will have installed the appropriate device driver for your UNIX system, and created device files to communicate with the tape drive.

This section describes how you can verify that the installation has been performed correctly.

In outline, the procedure is as follows:

1. Write test data to a tape.
2. Read the test data from the tape.
3. Compare the data read from the tape with the original data on disk.

To verify the installation

1. Test the SCSI connection to the tape drive by doing a rewind operation:
   a. If there is a tape cartridge already in the drive, remove it.
   b. Insert a tape cartridge.
   c. Rewind the tape using the command line:
      ```
      % mt -f <archive name> rewind
      ```
   If you do not see the Tape light flash as the tape rewinds, the hardware installation may be faulty. Check the troubleshooting section of the User's Guide for help in identifying the problem.
2. Write a sample file to tape, using 'tar':
   ```
   % cd /
   % tar cvf <archive name> <file>
   ```
   The options to `tar` have the following meanings:

`c`	Create a new archive (backup file) on the device.
`v`	Operate in verbose mode.
`f`	Specify the archive name explicitly.

   The arguments follow the `cvf` options in the command line. Their values depend on the operating system; suggested values are given in "System-specific arguments" on page 31. The arguments are as follows:

`<archive name>`	The name of the archive name to be created. *Example:* `/dev/rmt/0m`
`<file>`	The name of the file to archive, prefixed with '`./`'. *Example:* `./stand/vmunix`

NOTE: Make sure you prefix the file name with '.' when you back it up to tape. If you do not, the restore operation in step 3 will overwrite the original copy on disk.

3. Read the file back from tape:

```
% cd /tmp
% tar xvf <archive name>
```

The 'x' option to `tar` here means "extract from the archive".

Use the same value for the `<archive name>` argument as in step 2.

4. Compare the original with this retrieved file:

```
% cmp <original file> /tmp/<retrieved file>
```

This step compares the retrieved file and the original file byte by byte. If they are the same, there should be no output, and this verifies that the installation is correct. The arguments are as follows:

`<original file>`	The name of the original file, prefixed with '`/`'. *Example:* `/stand/vmunix`
`<retrieved file>`	The name of the file retrieved from the archive. *Example:* `stand/vmunix`

Example

Suppose you are verifying the installation of an HP DDS-format tape drive on an HP-UX 10.X system. The procedure would be as follows. See "System-Specific Arguments" below for the choice of `<archive name>` and `<file>` arguments:

1. Change directory to root:

```
% cd /
```

2. Back up `/stand/vmunix` to tape:

```
% tar cvf /dev/rmt/0m ./stand/vmunix
```

Note the prefix of '.' to the filename.

3. Change to the temporary directory:

```
% cd /tmp
```

4. Extract the file from the tape:

```
% tar xvf /dev/rmt/0m
```

5. Compare the original with the restored version:

```
% cmp /stand/vmunix /tmp/stand/vmunix
```

Note that the original filename is *not* prefixed with '.'.

System-specific arguments

The following table lists suggested values for the arguments <archive name> and <file> in the verification procedure described above. If any of the suggested files are symbolic links on your system, choose another file appropriate for your system.

System	File Name	Description	Archive Name	Notes
HP Alpha	vmunix	OSF kernel	/dev/tape/tape*X*.d*n*	*X* is the instance of the drive *n* in the density code
HP-UX 11.x	stand/vmunix	HP-UX kernel	/dev/rmt/*Y*m	*Y* is the instance of the drive
IBM AIX	unix	AIX kernel	/dev/rmt*Y*.1	*Y* is the device ID reported back as available when you ran 'smit -C tape' to create the device files.
Linux	/boot/vmlinux	Kernel 2.4.x	/dev/[n]st*X*	n means no rewind *X* is the instance of the drive
SUN Solaris 2 (SunOS 5.x)	bin/csh	C shell	Determine the archive name as described below.	

Determining the archive name for SUN Solaris 2

Determine the archive name by typing:

```
% ls -l /dev/rmt/*m | grep "st@X"
```

where X is the SCSI ID. Identify the line for the tape drive. For example, if the drive was at SCSI ID 2, look for the line containing "st@2,0". This might be as follows (but on a single line):

```
lrwxrwxrwx 1 root root 63 Mar 1 00:00 /dev/rmt/0m
../../devices/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@2,0:m
```

Here you could use /dev/rmt/0m (shown underlined above) as the archive name.

Glossary

AT&T mode Berkeley and AT&T functional modes differ in “read-only” close functionality. In AT&T mode, a device close operation will cause the tape to be repositioned just after next filemark on the tape (the start of the next file).

Berkeley mode Berkeley and AT&T functional modes differ in “read-only” close functionality. In Berkeley mode the tape position will remain unchanged by a device close operation.

block A logical unit of information. Called “record” in the DDS-format specification.

BOP Beginning Of Partition. The position at the beginning of the permissible recording region of a partition.

buffered mode A mode of data transfer in write operations that facilitates tape streaming.

compression A procedure in which data is transformed by the removal of redundant information in order to reduce the number of bits required to represent the data. This is done by representing strings of bytes with codewords.

DAT Digital Audio Tape

data transfer phase On a SCSI bus, devices put in requests to be able to transfer information. Once a device is granted its request, it and the target to which it wants to send information can transfer the data using one of three protocols (assuming both devices support them): asynchronous, synchronous, and wide.

In *asynchronous* transfers, the target controls the flow of data. The initiator can only send data when the target has acknowledged receipt of the previous packet. All SCSI devices must support asynchronous transfer.

In *synchronous* data transfer, the initiator and target work in synchronization, allowing transmission of a packet of data to start before acknowledgment of the previous transmission.

In *wide* (16-bit) data transfer, two bytes are transferred at the same time instead of a single byte.

HP DDS drives support asynchronous, synchronous and narrow (8-bit) wide transfers.

DDS Digital Data Storage is a recording format that builds on the DAT format to support the storage of computer data. It was developed originally by Hewlett-Packard and Sony as an industry standard. The first generation standard was DDS-1 (or simply DDS), to which was added data compression to produce the DDS-DC standard.

Further enhancements, notably narrower tracks and thinner tape, led to DDS-2, which can typically provide double the capacity of DDS-1.

DDS-3 uses a new magnetic coating on the tape that allows twice the recording density. Together with the use of time-tracking, this gives a DDS-3 tape approximately three times the capacity of a DDS-2 tape.

DDS-4 uses longer tapes (150m). HP's DDS-4 drives, which are ultra-wide SCSI devices, allow transfer rates from 3 to 4 times greater than DDS-3 and capacities that are two-thirds as much again.

DAT 72 tapes are 170m long and data is written in narrower tracks, again increasing data density, enabling tapes to hold 80% more data than DDS-4 tapes.

filemark A mark written by the host to the tape that can be searched for, often using the drive's fast-search capability. It does not necessarily separate files. It is up to the host to assign a meaning to the mark.

group A fixed capacity set of tracks written to or read from tape, defined in the DDS format.

immediate mode A mode of responding to SCSI commands where the drive or other peripheral does not wait until the command has finished before returning status information back to the host. For writing filemarks, Immediate mode can significantly improve the performance of systems that do not set the Immediate bit when sending a SCSI WRITE FILEMARKS command. On the other hand, data is not flushed to tape in response to a filemark command.

infinite flush By default, the buffer in the drive is flushed every 5 seconds. Infinite flush avoids frequent starting and stopping of the mechanism when using a very slow application. It also avoids losing capacity through the flushing of partly written groups. On the other hand, infinite flush means that data can remain in the buffer for very long periods of time, and could be lost in the event of a power failure.

LUN Logical Unit Number, by which different devices at a particular SCSI ID can be addressed individually. The drive has a fixed LUN of 0.

Media Recognition System (MRS) A method by which a drive can recognize data-grade tape. The tape has a series of stripes on its transparent leader tape that the drive can detect. By default, the drive treats a non-Media Recognition System tape as read-only and will not write data to it.

partition A part of a tape that can be treated as a complete and independent whole. A tape can have one or two partitions.

SCSI Small Computer System Interface

sequential access Sequential access devices store data sequentially in the order in which it is received. Tape devices are the most common sequential access devices. Devices such as disk drives are direct access devices, where data is stored in blocks, not necessarily sequentially. Direct access allows for speed of retrieval, but is significantly more costly.

Index

S

U

V

W