NAME

st – driver for SCSI tape devices

SYNOPSIS

st@*target,lun:[l,m,h,c,u][b][n]*

DESCRIPTION

The **st** device driver is an interface to various SCSI tape devices. Supported tape devices include 1/4″ Tandberg 2.5 Gigabyte QIC tape drive, 1/4″ Archive Viper QIC-150 streaming tape drive, 1/4″ Emulex MT-02 tape controller, HP-88780 1/2″ tape drive, Exabyte EXB-8200/8500/8505/8505XL 8mm cartridge tape, and the Archive Python 4 mm DAT tape subsystem. **st** provides a standard interface to these various devices; see **mtio**(7I) for details.

The driver can be opened with either rewind on close or no rewind on close options. A maximum of four tape formats per device are supported (see **FILES** below). The tape format is specified using the device name. Often tape format is also referred to as tape density.

Read Operation

If the driver is opened for reading in a different format than the tape is written in, the driver overrides the user selected format. For example, if a 1/4″ cartridge tape is written in QIC-24 format and opened for reading in QIC-150, the driver will detect a read failure on the first read and automatically switch to QIC-24 to read the data.

Note that if the low density format is used, no indication is given that the driver has overridden the user selected format. Other formats issue a warning message to inform the user of an overridden format selection. Some devices automatically perform this function and do not require driver support (1/2″ reel tape drive, for example).

Write Operation

Writing from the beginning of tape is performed in the user-specified format. The original tape format is used for appending onto previously written tapes.

Tape Configuration

The **st** tape driver has a built-in configuration table for all Sun supported tape drives. In order to support the addition of third party tape devices or to override a built-in configuration, drive information can be supplied in **st.conf** as global properties that apply to each node, or as properties that are applicable to one node only. The **st** driver looks for the property called “tape-config-list”. The value of this property is a list of triplets, where each triplet consists of three strings.

The formal syntax is:

```
tape-config-list = <triplet> [, <triplet> *];
```

where

```
<triplet> := <vid+pid>, <pretty print>, <data-property-name>
```

and

```
<data-property-name> = <version>, <type>, <bsize>,
        <options>, <number of densities>,
        <density> [, <density>*], <default-density>;
```

Note that a semicolon (;) is used to terminate a prototype devinfo node specification. Individual elements listed within the specification should not be separated by a semicolon. Refer to **driver.conf**(4) for more information.

<vid+pid> is the string that is returned by the tape device on a SCSI inquiry command. This string may contain any character in the range 0x20-0x7e. Characters such as “ " ” (double quote) or “ ’ ” (single quote), which are not permitted in property value strings, are represented by their octal equivalent (for example, \042 and \047). Trailing spaces may be truncated.

<pretty print> is used to report the device on the console. This string may have zero length, in which case the **<vid+pid>** will be used to report the device.

<data-property-name> is the name of the property which contains all the tape configuration values (such as **<type>**, **<bsize>**, etc.) corresponding for the tape drive for the specified **<vid+pid>**.

<version> is a version number and should be 1. In the future, higher version numbers may be used to allow for changes in the syntax of the **<data-property-name>** value list.

<type> is a type field. Valid types are defined in **/usr/include/sys/mtio.h**. For third party tape configuration, the following generic types are recommended:

MT_ISQIC	0x32
MT_ISREEL	0x33
MT_ISDAT	0x34
MT_IS8MM	0x35
MT_ISOTHER	0x36

<bsize> is the preferred block size of the tape device. The value should be 0 for variable block size drives.

<options> is a bit pattern representing the drive options, as defined in **/usr/include/sys/scsi/targets/stdef.h**. Valid flags for tape configuration are:

ST_VARIABLE	0x0001
ST_QIC	0x0002
ST_REEL	0x0004
ST_BSF	0x0008
ST_BSR	0x0010
ST_LONG_ERASE	0x0020
ST_AUTODEN_OVERRIDE	0x0040
ST_NOBUF	0x0080
ST_KNOWS_EOD	0x0200
ST_UNLOADABLE	0x0400
ST_SOFT_ERROR_REPORTING	0x0800
ST_LONG_TIMEOUTS	0x1000
ST_BUFFERED_WRITES	0x4000
ST_NO_RECSIZE_LIMIT	0x8000
ST_MODE_SEL_COMP	0x10000

ST_VARIABLE
The flag indicates the tape device supports variable length record sizes.

ST_QIC
The flag indicates a Quarter Inch Cartridge (QIC) tape device.

ST_REEL
The flag indicates a 1/2–inch reel tape device.

ST_BSF If flag is set, the device supports backspace over EOF marks (bsf - see **mt**(1)).

ST_BSR
If flag is set, the tape device supports the backspace record operation (bsr - see **mt**(1)). If the device does not support bsr, the **st** driver emulates the action by rewinding the tape and using the forward space record (fsf) operation to forward the tape to the correct file. The driver then uses forward space record (fsr - see **mt**(1)) to forward the tape to the correct record.

ST_LONG_ERASE
The flag indicates the tape device needs a longer time than normal to erase.

ST_AUTODEN_OVERRIDE
The auto-density override flag. The device is capable of determining the tape density automatically without issuing a “mode-select”/“mode-sense command”.

ST_NOBUF
The flag disables the device’s ability to perform buffered writes. A buffered write occurs when the device acknowledges the completion of a write request after the data has been written to the device’s buffer, but before all of the data has been written to the tape.

ST_KNOWS_EOD
If flag is set, the device can determine when EOD (End of Data) has been reached. When this flag is set, the **st** driver uses fast file skipping. Otherwise, file skipping happens one file at a time.

ST_UNLOADABLE
The flag indicates the device will not complain if the **st** driver is unloaded and loaded again (see **modload**(1M) and **modunload**(1M)). That is, the driver will return the correct inquiry string.

ST_SOFT_ERROR_REPORTING
The flag indicates the tape device will perform a “request sense” or “log sense” command when the device is closed. Currently, only Exabyte and DAT drives support this feature.

ST_LONG_TIMEOUTS
The flag indicates the tape device requires timeouts that are 5 times longer than usual for normal operation.

ST_BUFFERED_WRITES
If the flag is set, when data is written to the tape device, the data is buffered by the driver. The application may receive acknowledgement of completion of the write request before the data has been written to tape.

ST_NO_RECSIZE_LIMIT (SPARC Only)
The flag applies to variable-length tape devices. If this flag is set, the record size is not limited to a 64 Kbyte record size. The record size is only limited by the smaller of either the record size supported by the device or the maximum DMA transfer size of the system. Refer to **Large Record Sizes** and **WARNINGS**.

ST_MODE_SEL_COMP
If the **ST_MODE_SEL_COMP** flag is set, the driver uses the device configuration mode pages to enable or disable compression. This is only needed for those tape devices which do not support enabling compression via density codes. If the "c" or "u" device is used, compression will be enabled. If any other device density is used, it will be disabled.

<number of densities> is the number of densities specified. Each tape drive can support up to four densities. The value entered should therefore be between 1 and 4; if less than 4, the remaining densities will be assigned a value of 0x0.

<density> is a single byte hexadecimal number. It can either be found in the drive specification manual or be obtained from the drive vendor.

<default-density> has a value between 0 and (<number of densities> - 1).

IOCTLS

The behavior of SCSI tape positioning ioctls is the same across all devices which support them. Refer to **mtio**(7I). However, not all devices support all ioctls. The driver returns an **ENOTTY** error on unsupported ioctls.

The retension ioctl only applies to 1/4″ cartridge tape devices. It is used to restore tape tension, thus improving the tape's soft error rate after extensive start-stop operations or long-term storage.

In order to increase performance of variable-length tape devices (particularly when they are used to read/write small record sizes), two operations in the **MTIOCTOP** ioctl, **MTSRSZ** and **MTGRSZ**, can be used to set and get fixed record lengths. The ioctl also works with fixed-length tape drives which allow multiple record sizes. The min/max limits of record size allowed on a driver are found by using a SCSI-2 **READ BLOCK LIMITS** command to the drive. If this command fails, the default min/max record sizes allowed are 1 byte and 63k bytes. An application that needs to use a different record size opens the device, sets the size with the **MTSRSZ** ioctl and then continues with I/O. The scope of the change in record size remains until the device is closed. The next open to the device resets the record size to the default record size (retrieved from **st.conf**).

Note that the error status is reset by the **MTIOCGET** get status ioctl call or by the next read, write, or other ioctl operation. If no error has occurred (sense key is zero), the current file and record position is returned.

ERRORS

EACCES The driver is opened for write access and the tape is write protected.

EBUSY The tape drive is in use by another process. Only one process can use the tape drive at a time. The driver will allow a grace period for the other process to finish before reporting this error.

EINVAL The number of bytes read or written is not a multiple of the physical record size (fixed-length tape devices only).

EIO During opening, the tape device is not ready because either no tape is in the drive, or the drive is not on-line. Once open, this error is returned if the requested I/O transfer could not be completed.

ENOTTY This indicates that the tape device does not support the requested ioctl function.

ENXIO During opening, the tape device does not exist.

EXAMPLES

Example of a global tape-config-list property:

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

tape-config-list =
        "Magic DAT", "Magic 4mm Helical Scan", "magic-data";

magic-data    = 1,0x34,1024,0x1639,4,0,0x8c,0x8c,0x8c,3;

name="st" class="scsi"
         target=0 lun=0;

name="st" class="scsi"
         target=1 lun=0;

name="st" class="scsi"
         target=2 lun=0;
         .
         .
         .
name="st" class="scsi"
         target=6 lun=0;
```

Example of a tape-config-list property applicable to target 2 only:

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

name="st" class="scsi"
         target=0 lun=0;

name="st" class="scsi"
```

```
        target=1 lun=0;
name="st" class="scsi"
        target=2 lun=0
        tape-config-list =
          "Magic   DAT", "Magic 4mm Helical Scan", "magic-data"
        magic-data = 1,0x34,1024,0x1639,4,0,0x8c,0x8c,0x8c,3;
name="st" class="scsi"
        target=3 lun=0;
        .
        .
        .
name="st" class="scsi"
        target=6 lun=0;
```

Large Record Sizes SPARC ONLY

To support applications such as seismic programs that require large record sizes, the flag **ST_NO_RECSIZE_LIMIT** must be set in drive option in the configuration entry. A SCSI tape drive that needs to transfer large records should OR this flag with other flags in the 'options' field in **st.conf**. Refer to **Tape Configuration**. By default, this flag is set for the built-in config entries of Archive DAT and Exabyte drives.

If this flag is set, the **st** driver issues a SCSI-2 **READ BLOCK LIMITS** command to the device to determine the maximum record size allowed by it. If the command fails, **st** continues to use the maximum record sizes mentioned in the **mtio**(7I) man page.

If the command succeeds, **st** restricts the maximum transfer size of a variable-length device to the minimum of that record size and the maximum DMA size that the host adapter can handle. Fixed-length devices are bound by the maximum DMA size allocated by the machine. Note that tapes created with a large record size may not be readable by earlier releases or on other platforms.

Refer to the **WARNINGS** section for more information.

EOT Handling

The Emulex drives have only a physical end of tape (PEOT); thus it is not possible to write past EOT. All other drives have a logical end of tape (LEOT) before PEOT to guarantee flushing the data onto the tape. The amount of storage between LEOT and PEOT varies from less than 1 Mbyte to about 20 Mbyte, depending on the tape drive.

If EOT is encountered while writing an Emulex, no error is reported but the number of bytes transferred is zero and no further writing is allowed. On all other drives, the first write that encounters EOT will return a short count or zero. If a short count is returned, then the next write will return zero. After a zero count is returned, the next write returns a full count or short count. A following write returns zero again. It is important that the number and size of trailer records be kept as small as possible to prevent data loss. Therefore, writing after EOT is not recommended.

Reading past EOT is transparent to the user. Reading is stopped only by reading EOF's. For 1/2" reel devices, it is possible to read off the end of the reel if one reads past the two file marks which mark the end of recorded media.

Write Data Buffering

Tape drives with data compression require a much higher data rate in order to stream the tape. Write data buffering in the driver improves streaming to the drive without changing the application and augments the buffering in the tape drive itself. If write data buffering is enabled, data is buffered in the driver and the request is immediately acknowledged by the driver before it has been written to the tape drive. This enables the driver to submit the next request as soon as the previous request completes and the application to prepare the next request while the current request is in progress. A SCSI tape drive that allows buffering requires ORing the flag **ST_BUFFERED_WRITES** with other flags in the 'options' field in **st.conf**. Refer to **Tape Configuration**. By default, this option is set for the built-in config entries of the Archive DAT and Exabyte drives.

In order for write buffering to work properly, sufficient space after LEOT must be available to empty the write buffers. Older tape devices usually do not have sufficient space after LEOT.

To turn on tape buffering, a property in **st.conf** called "tape-driver-buffering" should be added. The value assigned to this property is the maximum number of buffered write requests allowed. For example, **0** indicates no write request buffering allowed, while **2** indicates buffer up to 2 write requests. If this property is not specified in **st.conf**, the driver defaults to a value of **0**. The maximum size of write request that can be buffered is specified through a property in **st.conf** called "tape-driver-buf-max-size". If this property is not specified in **st.conf,** the driver defaults the buffer size to a value of 1 Mbyte.

An example of **st.conf**, where the maximum number of write requests buffered is 4 and maximum size of write request buffered is 2 Mbyte, is given below. This applies to all nodes in this **conf** file.

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

tape-driver-buffering = 4; tape-driver-buf-max-size = 0x200000;

name="st" class="scsi"
    target=0 lun=0;

name="st" class="scsi"
    target=1 lun=0;

name="st" class="scsi"
    target=2 lun=0;

...
```

In the case of a SCSI bus reset, a medium error, or any other fatal transport error on a buffered request, the driver returns an error on subsequent write requests and allows no more writes. If no further write requests occur, an error is returned on close.

Since some applications may perceive write buffering as a potential data integrity problem, this feature is disabled by default and needs to be explicitly enabled in the config entry and turned on by means of the property in **st.conf**. Furthermore, some fault tolerant backup servers make assumptions about the data buffering in the tape drive itself. These assumptions may not be valid if write buffering has been enabled.

Write buffering may be superseded by other performance enhancements in a future release.

FILES

/kernel/drv/st.conf
driver configuration file

/usr/include/sys/mtio.h
structures and definitions for mag tape io control commands

/usr/include/sys/scsi/targets/stdef.h
definitions for SCSI tape drives

/dev/rmt/[0– 127][l,m,h,u,c][b][n]
where **l,m,h,u,c** specifies the density (low, medium, high, ultra/compressed), **b** the optional BSD behavior (see **mtio**(7I)), and **n** the optional no rewind behavior. For example, **/dev/rmt/0lbn** specifies unit 0, low density, BSD behavior, and no rewind.

For 1/2” reel tape devices (HP-88780), the densities are:

l	800 BPI density
m	1600 BPI density
h	6250 BPI density
c	data compression (not supported on all modules)

For helical-scan tape devices (Exabyte 8200/8500/8505):

l	Standard 2 Gbyte format
m	5 Gbyte format (8500, 8505 only)
h,c	5 Gbyte compressed format (8505 only)

For 4mm DAT tape devices (Archive Python):

l	Standard format
m,h,c	data compression

For QIC-150 tape devices (Archive Viper):

l	QIC-150 Format
m	QIC-150 Format
h	QIC-150 Format
c	QIC-150 Format

For QIC-24 tape devices (Emulex MT–02):

l	QIC-11 Format
m	QIC-24 Format
h	QIC-24 Format
c	QIC-24 Format

SEE ALSO

mt(1), **modload**(1M), **modunload**(1M), **read**(2), **write**(2), **driver.conf**(4), **esp**(7D), **isp**(7D), **mtio**(7I), **ioctl**(9E)

DIAGNOSTICS

Error for command '<command name>'Error Level: Fatal
Requested Block <n>, Error Block: <m>
Sense Key: <sense key name>
Vendor '<name>': ASC = 0x<a> (<extended sense code name>),
ASCQ = 0x<b>, FRU = 0x<c>

The command indicated by <command name> failed. The Requested Block is the block where the transfer started and the Error Block is the block that caused the error. Sense Key, ASC, ASCQ and FRU information is returned by the target in response to a request sense command.

write/read: not modulo <n> block size

The request size for fixed record size devices must be a multiple of the specified block size.

recovery by resets failed

After a transport error, the driver attempted to recover with device and bus reset. This recovery failed.

Periodic head cleaning required

The driver reported that periodic head cleaning is now required.

Soft error rate (<n>%) during writing/reading was too high

The soft error rate has exceeded the threshold specified by the vendor.

SCSI transport failed: reason 'xxxx': {retrying | giving up}

The host adapter has failed to transport a command to the target for the reason stated. The driver will either retry the command or, ultimately, give up.

WARNINGS

Beginning with Solaris 2.4, the **ST_NO_RECSIZE_LIMIT** flag is set for the built-in config entries of the Archive DAT and Exabyte drivers by default. Refer to **Large Record Sizes**. Tapes written with large block sizes prior to Solaris 2.4 may cause some applications to fail if the number of bytes returned by a read request is less than the requested block size (e.g. asking for 128 Kbytes and receiving less than 64 Kbytes).

The **ST_NO_RECSIZE_LIMIT** flag can be disabled in the config entry for the drive as a work around. Refer to **Tape Configuration**. This action disables the ability to read and write with large block sizes and allows the reading of tapes written prior to Solaris 2.4 with large block sizes.

Refer to **mtio**(7I) for a description of maximum record sizes.

BUGS

Tape devices that do not return a BUSY status during tape loading prevent user commands from being held until the device is ready. The user must delay issuing any tape operations until the tape device is ready. This is not a problem for Sun Microsystem Computer Corporation supplied tape devices.

Tape devices that do not report a blank check error at the end of recorded media may cause file positioning operations to fail. Some tape drives for example, mistakenly report media error instead of blank check error.