NAME

scsi_ifgetcap, scsi_ifsetcap – get/set SCSI transport capability

SYNOPSIS

#include <sys/scsi/scsi.h>

int scsi_ifgetcap(struct scsi_address **ap*, **char** **cap*, **int** *whom***);**

int scsi_ifsetcap(struct scsi_address **ap*, **char** **cap*, **int** *value*, **int** *whom***);**

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

ap	Pointer to the **scsi_address** structure.
cap	Pointer to the string capability identifier.
value	Defines the new state of the capability.
whom	Determines if all targets or only the specified target is affected.

DESCRIPTION

The target drivers use **scsi_ifsetcap( )** to set the capabilities of the host adapter driver. A *cap* is a name-value pair whose name is a null terminated character string and whose value is an integer. The current value of a capability can be retrieved using **scsi_ifgetcap( )**. If *whom* is **0** all targets are affected, else the target specified by the **scsi_address** structure pointed to by *ap* is affected.

A device may support only a subset of the capabilities listed below. It is the responsibility of the driver to make sure that these functions are called with a *cap* supported by the device.

The following capabilities have been defined:

dma-max	Maximum dma transfer size supported by host adapter.
msg-out	Message out capability supported by host adapter: **0** disables, **1** enables.
disconnect	Disconnect capability supported by host adapter: **0** disables, **1** enables.
synchronous	Synchronous data transfer capability supported by host adapter: **0** disables, **1** enables.
wide-xfer	Wide transfer capability supported by host adapter: **0** disables, **1** enables.
parity	Parity checking by host adapter: **0** disables, **1** enables.
initiator-id	The host's bus address is returned.
untagged-qing	The host adapter's capability to support internal queueing of commands without tagged queueing: **0** disables, **1** enables.
tagged-qing	The host adapter's capability to support tagged queuing: **0** disables, **1** enables.
auto-rqsense	The host adapter's capability to support auto request sense on check conditions: **0** disables, **1** enables.
sector-size	The target driver sets this capability to inform the HBA of the

granularity, in bytes, of DMA breakup; the HBA's DMA limit structure will be set to reflect this limit (See **ddi_dma_lim_sparc**(9S) or **ddi_dma_lim_x86**(9S)). It should be set to the physical disk sector size. This capability defaults to **512**.

total-sectors The target driver sets this capability to inform the HBA of the total number of sectors on the device, as returned from the SCSI **get capacity** command. This capability must be set before the target driver ''gets'' the **geometry** capability.

geometry This capability returns the HBA geometry of a target disk. The target driver must set the **total-sectors** capability before ''getting'' the **geometry** capability. The geometry is returned as a 32-bit value: the upper 16 bits represent the number of heads per cylinder; the lower 16 bits represent the number of sectors per track. The **geometry** capability cannot be ''set.''

reset-notification The host adapter's capability to support bus reset notification: **0** disables, **1** enables. Refer to **scsi_reset_notify**(9F).

linked -cmds The host adapter's capability to support linked commands: **0** disables, **1** enables.

qfull-retries This capability enables/disables **QUEUE FULL** handling. If **0**, the HBA will not retry a command when a **QUEUE FULL** status is returned. If greater than **0**, then the HBA driver will retry the command at specified number of times at an interval determined by the "**qfull-retry-interval**". The range for **qfull-retries** is **0-255**.

qfull-retry-interval This capability sets the retry interval (in ms) for commands that were completed with a **QUEUE FULL** status. The range for **qfull-retry-intervals** is **0-1000** ms.

RETURN VALUES

scsi_ifsetcap( ) returns:

1 if the capability was successfully set to the new value,

0 if the capability is not variable, and

–1 if the capability was not defined, or setting the capability to a new value failed.

scsi_ifgetcap( ) returns:

the current value of a capability

–1 if the capability was not defined.

CONTEXT

These functions can be called from user or interrupt context.

EXAMPLES

```
un->un_arq_enabled =
        ((scsi_ifsetcap(&devp->sd_address, "auto-rqsense", 1, 1) == 1)? 1: 0);

if (scsi_ifsetcap(&devp->sd_address, "tagged-qing", 1, 1) == 1) {
        un->un_dp->options |= SD_QUEUEING;
        un->un_throttle = MAX_THROTTLE;
```

```
} else if (scsi_ifgetcap(&devp->sd_address, "untagged-qing", 0) == 1) {
        un->un_dp->options |= SD_QUEUEING;
        un->un_throttle = 3;
} else {
        un->un_dp->options &= ~SD_QUEUEING;
        un->un_throttle = 1;
}
```

SEE ALSO

scsi_reset_notify(9F), **scsi_address**(9S), **scsi_arq_status**(9S)

Writing Device Drivers