NAME

scsi_cname, scsi_dname, scsi_mname, scsi_rname, scsi_sname – decode a SCSI name

SYNOPSIS

#include <sys/scsi/scsi.h>

char ∗**scsi_cname(u_char** *cmd*, **char** ∗∗*cmdvec***);**

char ∗**scsi_dname(int** *dtype***);**

char ∗**scsi_mname(u_char** *msg***);**

char ∗**scsi_rname(u_char** *reason***);**

char ∗**scsi_sname(u_char** *sense_key***);**

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

cmd A SCSI command value.

cmdvec Pointer to an array of command strings.

dtype Device type.

msg A message value.

reason A packet reason value.

sense_key A SCSI sense key value.

DESCRIPTION

scsi_cname( ) decodes SCSI commands. *cmdvec* is a pointer to an array of strings. The first byte of the string is the command value, and the remainder is the name of the command.

scsi_dname( ) decodes the peripheral device type (for example, direct access or sequential access) in the inquiry data.

scsi_mname( ) decodes SCSI messages.

scsi_rname( ) decodes packet completion reasons.

scsi_sname( ) decodes SCSI sense keys.

RETURN VALUES

These functions return a pointer to a string. If an argument is invalid, they return a string to that effect.

CONTEXT

These functions can be called from user or interrupt context.

EXAMPLE

scsi_cname( ) decodes SCSI commands as follows:

```
static char *st_cmds[] = {
        "\000test unit ready",
        "\001rewind",
        "\003request sense",
        "\010read",
        "\012write",
        "\020write file mark",
        "\021space",
        "\022inquiry",
        "\025mode select",
        "\031erase tape",
        "\032mode sense",
        "\033load tape",
        NULL
};
..
cmn_err(CE_CONT, "st: cmd=%s", scsi_cname(cmd, st_cmds));
..
```

SEE ALSO

Writing Device Drivers