NAME

t_optmgmt – manage options for a transport endpoint

SYNOPSIS

cc [*flag* ...] *file* ... **–lnsl** [*library* ...]

#include <tiuser.h>

int t_optmgmt(int *fildes*, **const struct t_optmgmt** **req*, **struct t_optmgmt** **ret***);**

MT-LEVEL

MT-Safe

DESCRIPTION

The **t_optmgmt( )** function enables a transport user to retrieve, verify, or negotiate protocol options with the transport provider. *fildes* identifies a bound transport endpoint.

The *req* and *ret* arguments point to a **t_optmgmt** structure containing the following members:

```
struct netbuf opt;
long flags;
```

The **opt** field identifies protocol options and the **flags** field is used to specify the action to take with those options.

The options are represented by a **netbuf** (see **t_connect**(3N); also for **len**, **buf**, and **maxlen**) structure in a manner similar to the address in **t_bind**(3N). *req* is used to request a specific action of the provider and to send options to the provider. **len** specifies the number of bytes in the options, **buf** points to the options buffer, and **maxlen** has no meaning for the *req* argument. The transport provider may return options and flag values to the user through *ret*. For *ret*, **maxlen** specifies the maximum size of the options buffer and **buf** points to the buffer where the options are to be placed. On return, **len** specifies the number of bytes of options returned. **maxlen** has no meaning for the *req* argument, but must be set in the *ret* argument to specify the maximum number of bytes the options buffer can hold. The actual structure and content of the options is imposed by the transport provider.

The **flags** field of *req* can specify one of the following actions:

T_NEGOTIATE This action enables the user to negotiate the values of the options specified in *req* with the transport provider. The provider will evaluate the requested options and negotiate the values, returning the negotiated values through *ret*.

T_CHECK This action enables the user to verify whether the options specified in *req* are supported by the transport provider. On return, the **flags** field of *ret* will have either **T_SUCCESS** or **T_FAILURE** set to indicate to the user whether the options are supported. These flags are only meaningful for the **T_CHECK** request.

T_DEFAULT This action enables a user to retrieve the default options supported by the transport provider into the **opt** field of *ret*. In *req*, the **len** field of **opt** must be zero and the **buf** field may be NULL.

If issued as part of the connectionless-mode service, **t_optmgmt( )** may block due to flow control constraints. The function will not complete until the transport provider has processed all previously sent data units.

RETURN VALUES

t_optmgmt( ) returns 0 on success. On failure **t_optmgmt( )** returns −1, **t_errno** is set to indicate the error, and possibly **errno** is set.

ERRORS

On failure, **t_errno** will be set to one of the following:

TBADF The specified file descriptor does not refer to a transport endpoint.

TACCES The user does not have permission to negotiate the specified options.

TBADFLAG An invalid flag was specified.

TBADOPT The specified protocol options were in an incorrect format or contained illegal information.

TBUFOVFLW The number of bytes allowed for an incoming argument is not sufficient to store the value of that argument. The information to be returned in *ret* will be discarded.

TOUTSTATE The function was issued in the wrong sequence.

TSYSERR A system error has occurred during execution of this function, **errno** will be set to the specific error.

SEE ALSO

t_bind(3N), **t_connect**(3N), **t_getinfo**(3N), **t_open**(3N)

Transport Interfaces Programming Guide

NOTES

This interface is safe in multithreaded applications.