NAME

t_accept – accept a connect request

SYNOPSIS

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

#include <tiuser.h>

int t_accept(int *fildes***, int** *resfd***, struct t_call** ∗*call***);**

MT-LEVEL

MT-Safe

DESCRIPTION

This function is issued by a transport user to accept a connect request. *fildes* identifies the local transport endpoint where the connect indication arrived, *resfd* specifies the local transport endpoint where the connection is to be established, and *call* contains information required by the transport provider to complete the connection. *call* points to a **t_call** structure that contains the following members:

```
struct netbuf   addr;
struct netbuf   opt;
struct netbuf   udata;
int             sequence;
```

struct netbuf contains the following members:

```
unsigned int    maxlen;
unsigned int    len;
char *      buf;
```

In *call*, **addr** is the address of the caller, **opt** indicates any protocol-specific parameters associated with the connection, **udata** points to any user data to be returned to the caller, and **sequence** is the value returned by **t_listen** that uniquely associates the response with a previously received connect indication.

A transport user may accept a connection on either the same, or on a different, local transport endpoint from the one on which the connect indication arrived. If the same endpoint is specified (that is, *resfd=fildes*), the connection can be accepted unless the following condition is true: The user has received other indications on that endpoint but has not responded to them (with **t_accept( )** or **t_snddis**(3N)**).** For this condition, **t_accept**(3N) will fail and set **t_errno** to **TBADF**.

If a different transport endpoint is specified (*resfd*!=*fildes*), the endpoint must be bound to a protocol address and must be in the **T_IDLE** state (see **t_getstate**(3N)) before the **t_accept**(3N) is issued.

For both types of endpoints, **t_accept( )** will fail and set **t_errno** to **TLOOK** if there are indications (for example, a connect or disconnect) waiting to be received on that endpoint.

The values of parameters specified by **opt** and the syntax of those values are protocol specific. The **udata** argument enables the called transport user to send user data to the caller and the amount of user data must not exceed the limits supported by the transport provider, as returned in the **connect** field of the *info* argument of **t_open**(3N) or **t_getinfo**(3N). If the **len** field of **udata** is zero, no data will be sent to the caller.

RETURN VALUES

Upon successful completion, a value of 0 is returned. Otherwise, a value of –1 is returned, **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:

TACCES	The user does not have permission to accept a connection on the responding transport endpoint or to use the specified options.
TBADF	The specified file descriptor does not refer to a transport endpoint, or the user is illegally accepting a connection on the same transport endpoint on which the connect indication arrived.
TBADDATA	The amount of user data specified was not within the bounds allowed by the transport provider.
TBADOPT	The specified options were in an incorrect format or contained illegal information.
TBADSEQ	An invalid sequence number was specified.
TLOOK	An asynchronous event has occurred on the transport endpoint referenced by *fildes* and requires immediate attention.
TNOTSUPPORT	This function is not supported by the underlying transport provider.
TOUTSTATE	The function was issued in the wrong sequence on the transport endpoint referenced by *fildes*, or the transport endpoint referred to by *resfd* is not in the **T_IDLE** state.
TSYSERR	A system error has occurred during execution of this function, **errno** will be set to the specific error.

SEE ALSO

t_accept(3N), **t_connect**(3N), **t_getinfo**(3N), **t_getstate**(3N), **t_listen**(3N), **t_open**(3N), **t_snddis**(3N), **t_rcvconnect**(3N)

Transport Interfaces Programming Guide

NOTES

This interface is safe in multithreaded applications.