NAME

t_rcvconnect – receive the confirmation from a connect request

SYNOPSIS

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

#include <tiuser.h>

int t_rcvconnect(int *fildes*, **struct t_call** **call***);**

MT-LEVEL

MT-Safe

DESCRIPTION

This function enables a calling transport user to determine the status of a previously sent connect request and is used in conjunction with **t_connect**(3N) to establish a connection in asynchronous mode. The connection will be established on successful completion of this function.

fildes identifies the local transport endpoint where communication will be established, and *call* contains information associated with the newly established connection. *call* points to a **t_call** structure which contains the following members:

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

netbuf is described in **t_connect**(3N). In *call*, **addr** returns the protocol address associated with the responding transport endpoint, **opt** presents any protocol-specific information associated with the connection, **udata** points to optional user data that may be returned by the destination transport user during connection establishment, and **sequence** has no meaning for this function.

The **maxlen** (see **netbuf** in **t_connect**(3N)) field of each argument must be set before issuing this function to indicate the maximum size of the buffer for each. However, *call* may be NULL, in which case no information is given to the user on return from **t_rcvconnect( )**. By default, **t_rcvconnect( )** executes in synchronous mode and waits for the connection to be established before returning. On return, the **addr**, **opt**, and **udata** fields reflect values associated with the connection.

If **O_NDELAY** or **O_NONBLOCK** is set (using **t_open**(3N) or **fcntl**(2)), **t_rcvconnect( )** executes in asynchronous mode, and reduces to a poll for existing connect confirmations. If none are available, **t_rcvconnect( )** fails and returns immediately without waiting for the connection to be established. (See **TNODATA** below.) **t_rcvconnect( )** must be re-issued at a later time to complete the connection establishment phase and retrieve the information returned in *call*.

RETURN VALUES

t_rcvconnect( ) returns 0 on success. On failure **t_rcvconnect( )** 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.

TBUFOVFLW The number of bytes allocated for an incoming argument is not sufficient to store the value of that argument and the connect information to be returned in *call* will be discarded. The provider's state, as seen by the user, will be changed to **DATAXFER**.

TLOOK An asynchronous event has occurred on this transport connection and requires immediate attention.

TNODATA **O_NDELAY** or **O_NONBLOCK** was set, but a connect confirmation has not yet arrived.

TNOTSUPPORT This function is not supported by the underlying transport provider.

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

SEE ALSO

fcntl(2), **t_accept**(3N), **t_bind**(3N), **t_connect**(3N), **t_listen**(3N), **t_open**(3N)

Transport Interfaces Programming Guide

NOTES

This interface is safe in multithreaded applications.