NAME

t_rcvudata – receive a data unit

SYNOPSIS

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

#include <tiuser.h>

int t_rcvudata(int *fildes***, struct t_unitdata** ∗*unitdata***, int** ∗*flags***);**

MT-LEVEL

MT-Safe

DESCRIPTION

This function is used in connectionless mode to receive a data unit from another transport user. *fildes* identifies the local transport endpoint through which data will be received, *unitdata* holds information associated with the received data unit, and *flags* is set on return to indicate that the complete data unit was not received. *unitdata* points to a **t_unitdata** structure containing the following members:

```
struct netbuf    addr;
struct netbuf    opt;
struct netbuf    udata;
```

The **maxlen** (see **netbuf** in **t_connect**(3N)) field of **addr**, **opt**, and **udata** must be set before issuing this function to indicate the maximum size of the buffer for each.

On return from this call, **addr** specifies the protocol address of the sending user, **opt** identifies protocol-specific options that were associated with this data unit, and **udata** specifies the user data that was received.

By default, **t_rcvudata( )** operates in synchronous mode and will wait for a data unit to arrive if none is currently available. However, if **O_NDELAY** or **O_NONBLOCK** is set (using **t_open**(3N) or **fcntl**(2)), **t_rcvudata( )** will execute in asynchronous mode and will fail if no data units are available.

If the buffer defined in the **udata** field of *unitdata* is not large enough to hold the current data unit, the buffer will be filled and **T_MORE** will be set in *flags* on return to indicate that another **t_rcvudata( )** should be issued to retrieve the rest of the data unit. Subsequent **t_rcvudata( )** call(s) will return zero for the length of the address and options until the full data unit has been received.

RETURN VALUES

t_rcvudata( ) returns 0 on successful completion. On failure **t_rcvudata( )** 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 the incoming protocol address or options is greater than zero but not sufficient to store the information. The unit data information to be returned in *unitdata* will be discarded.
TLOOK	An asynchronous event has occurred on this transport endpoint and requires immediate attention.

TNODATA	**O_NDELAY** or **O_NONBLOCK** was set, but no data units are currently available from the transport provider.
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_connect**(3N), **t_rcvuderr**(3N), **t_sndudata**(3N)

Transport Interfaces Programming Guide

NOTES

This interface is safe in multithreaded applications.