NAME t_sync – synchronize transport library

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

#include <tiuser.h>

int t_sync(int *fildes***);**

MT-LEVEL MT-Safe

DESCRIPTION For the transport endpoint specified by *fildes*, **t_sync( )** synchronizes the data structures managed by the transport library with information from the underlying transport provider. In doing so, it can convert a raw file descriptor (obtained using **open**(2), **dup**(2), or as a result of a **fork**(2) and **exec**(2)) to an initialized transport endpoint, assuming that file descriptor referenced a transport provider. This function also allows two cooperating processes to synchronize their interaction with a transport provider.

For example, if a process **fork**(2)S a new process and issues an **exec**(2), the new process must issue a **t_sync( )** to build the private library data structure associated with a transport endpoint and to synchronize the data structure with the relevant provider information.

It is important to remember that the transport provider treats all users of a transport endpoint as a single user. If multiple processes are using the same endpoint, they should coordinate their activities so as not to violate the state of the provider. **t_sync( )** returns the current state of the provider to the user, thereby enabling the user to verify the state before taking further action. This coordination is only valid among cooperating processes; it is possible that a process or an incoming event could change the provider's state *after* a **t_sync( )** is issued.

If the provider is undergoing a state transition when **t_sync( )** is called, the function will fail.

RETURN VALUES **t_sync( )** returns the state of the transport provider on successful completion. **t_sync( )** returns -1 on failure, **t_errno** is set to indicate the error, and possibly **errno** is set. The state returned may be one of the following:

T_UNBND	unbound
T_IDLE	idle
T_OUTCON	outgoing connection pending
T_INCON	incoming connection pending
T_DATAXFER	data transfer
T_OUTREL	outgoing orderly release (waiting for an orderly release indication)
T_INREL	incoming orderly release (waiting for an orderly release request)

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.

TSTATECHNG The transport provider is undergoing a state change.

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

SEE ALSO

dup(2), **exec**(2), **fork**(2), **open**(2)

Transport Interfaces Programming Guide

NOTES

This interface is safe in multithreaded applications.