TPSETCTXT(3)

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 SYNOPSIS 1

2 DESCRIPTION 2

3 RETURN VALUE 3

4 ERRORS 4

5 EXAMPLE 5

6 BUGS 6

7 SEE ALSO 7

8 COPYING 8

Chapter 1

SYNOPSIS

#include <atmi.h>

int tpsetctxt(TPCONTEXT_T *context*, long *flags*);

Link with *-latmisrv|-latmisrvnomain|-latmisrvinteg -latmi -lubf -lnstd -lpthread -lrt -lm*

Chapter 2

DESCRIPTION

This function is used to set context data which previously was returned by **tpgetctxt(3)** funcion. If previous thread was associated with XA global transaction, then it will be resumed at this thread. *context* parameter accepts either thread context returned by **tpgetctxt()** or **TPNULLCONTEXT**. **TPNULLCONTEXT** will enter the current thread in NULL context and will free up the currently associated automatically allocateed thread local storage (TLS). Automatically allocated TLS is one which is not made by **tpnewctxt(3)** but created implicitly by ATMI any other ATMI operation which requires context.

If current thread was running in some context, it will be terminated (with **tpterm(3)**), TLS data will be freed. And new context will be installed.

This function uses underlaying thread local storage infrastructure which is provided separately for each of the major Enduro/X libraries - libnstd (Standard library), libufb (UBF buffer library) and libatmi (ATMI library). If operations at library levels are required, then following functions can be used:

1. ndrx_nstd_tls_new(), ndrx_ubf_tls_new(), ndrx_atmi_tls_new() - allocate TLS data for library.
2. ndrx_nstd_tls_get(), ndrx_ubf_tls_get(), ndrx_atmi_tls_get() - get the TLS data for library (currently associated with thread).
3. ndrx_nstd_tls_set(), ndrx_ubf_tls_set(), ndrx_atmi_tls_set() - set the thread local data from saved pointer.
4. ndrx_nstd_tls_free(), ndrx_ubf_tls_free(), ndrx_atmi_tls_free() - free the thread local data.

Chapter 3

RETURN VALUE

On success, **tpsetctxt()** return 0; on error, -1 is returned, with **tperrno** set to indicate the error.

Chapter 4

ERRORS

Note that **tpstrerror()** returns generic error message plus custom message with debug info from last function call. Error data (tperrno) is valid only if return was -1. In case if process was running in **TPNULLCONTEXT**, new client context will be made to store the error code.

TPENOENT *context* is pointing to invalid data.

TPESYSTEM Failed to context data. The state of TLS storage is uncertain.

Chapter 5

EXAMPLE

See **atmitest/test016_contextsw/atmiclt16.c** for sample code.

Chapter 6

BUGS

Report bugs to madars.vitolins@gmail.com

Chapter 7

SEE ALSO

tpgetctxt(3) tpfreectxt(3) tpsrvsetctxdata(3) tpsrvgetctxdata(3) tpcontinue(3) tpinit(3)

Chapter 8

COPYING