TPFORWARD(3)

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

SYNOPSIS

#include <atmi.h>

void tpforward (char *_svc_, char *_data, long 'len_, long _flags_);

Link with _-latmisrv|-latmisrvnomain|-latmisrvinteg -latmi -lubf -lnstd -lpthread -lrt -lm_

Chapter 2

DESCRIPTION

Send the current request control to another process/service and free up current processing instance for next request. This basically makes original **tpcall()** chaining in **relay race** mode. The original caller is unaware that it's call now will be serviced by another service. The destination service is provided in parameter *svc*, data buffer is passed in *data*/*len* pair. *data* buffer must be allocated by **tpalloc()** or can be auto-buffer. Auto-buffer is made free by default when call state is return or forwarded. If the buffer in *data* is not auto-buffer, then it will be deallocated too.

Normally the auto-buffer shall not be made free by user, as that will cause the corruption at the stage of the **tpreturn**. But auto-buffer free by user can be suitable in case if doing call state transfer over the non XATMI sub-system, then call context restore in other OS process with **tpsrvsetctxdata(3)** must be done with **TPNOAUTBUF** flag specified, and the original service which received request prior doing **tpcontinue(3)** shall make the auto buffer free **tpfree(3)**. If the *TPNOAUTBUF* was specified at state restore, then tpforward() command will not try to free the auto buffer.

If current process was in global transaction, then it will make the target process to join the global transaction too. If invalid service name is specified in *svc* (service is not advertised in system), then call will be returned the caller with service error (**TPESVCERR**). *flags* is reserved for future use.

tpreturn() will make the current service routine go back/pop-up the stack to Enduro/X polling mechanisms for next incoming message.

This function is available only for XATMI servers.

Chapter 3

RETURN VALUE

Function is marked as void and it have no return values.

Chapter 4

ERRORS

No errors available (but some details might be logged in trace files).

Chapter 5

EXAMPLE

See **atmitest/test002_basicforward/atmisv2FIRST.c** for sample code.

Chapter 6

BUGS

Report bugs to madars.vitolins@gmail.com

Chapter 7

SEE ALSO

tpreturn(3) tpcall(3)

Chapter 8

COPYING