

TPNOTIFY(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 tpnotify(CLIENTID *clientid, char *data, long len, long flags);
```

For XATMI client link with *-latmiclt -latmi -lubf -lnstd -lpthread -lrt -lm*

For XATMI server link with *-latmisrvl -latmisrvnomainl -latmisrvinteg -latmi -lubf -lnstd -lpthread -lrt -lm*

Chapter 2

DESCRIPTION

Function is used for sending unsolicited message to the client. The client is identified by *clientid*. The *clientid* can be acquired by server process, with incoming service call. The services parameter *svcinfo* by type **TPSVCINFO** contains *cltid* field. This can be used as client identifier for sending the notifications to. Also notifications can be delivered to server processes while it is doing **tpcall(3)**. The data sent to client is set in variable *data* which is XATMI buffer allocated by **tpalloc(3)** (or auto buffer). The complementary *len* is used only for buffer types which do not contain length descriptor internally (i.e. **CARRAY** buffer type).

The *data* field can be NULL. In this situation client callback will be called, but no data will be sent to client. In case when client is present on other cluster node, the message is delivered to bridge process. And on remote machine unsolicited message dispatching is performed by **tpbrdcstsv(8)**. when dispatching on remote machine timeout condition is not reported to the caller.

Valid flags

TPNOBLOCK Do not block on full client queue, instead return error.

TPNOTIME Do not timeout when waiting on full queue (**TPNOBLOCK** is not set).

TPSIGRSTRT Restart the system call in progress if interrupted by signal handler. This affects only underlaying mq_* function calls.

TPACK Reserved for future use, Enduro/X silently ignores this flag. Thus **tpnotify()** call does not get any acknowledgement signal that client is processed the message. This is limitation of Enduro/X.

Chapter 3

RETURN VALUE

On success, **tpcall()** return zero; 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.

TPEINVAL Environment variables not configured, see **ex_env(5)** page, or invalid parameters have been passed to the function, for example *clientid* is NULL or corrupted.

TPENOENT The local delivery was about to be performed (no remote client call) and the client process did not exist on local machine. This error will be reported regardless of the **TPACK** flag.

TPETIME Blocking message delivery did timeout. Meaning that client queue was full and **TPNOBLOCK** nor **TPNOTIME** was set. Error is returned from local clients only regardless of the **TPACK** flag. If client resists on remote node, then this error can be returned only when time-out occurred while sending message to then local bridge server.

TPEBLOCK Client queue was full and **TPNOBLOCK** flag was not specified.

TPESYSTEM System failure occurred during serving. See logs i.e. user log, or debugs for more info.

TPEOS System failure occurred during serving. See logs i.e. user log, or debugs for more info.

Chapter 5

EXAMPLE

See `atmitest/test038_tpnotify/atmisv38.c` for sample code.

Chapter 6

BUGS

Report bugs to support@mavimax.com

Chapter 7

SEE ALSO

tpsetunsol(3) tpbroadcast(3) tpchkunsol(3) tpinit(3) tpbrdcestdsv(8)

Chapter 8

COPYING

© Mavimax, Ltd