

**TAPCALL(3)**

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>SYNOPSIS</b>	<b>1</b>
<b>2</b>	<b>DESCRIPTION</b>	<b>2</b>
<b>3</b>	<b>RETURN VALUE</b>	<b>3</b>
<b>4</b>	<b>ERRORS</b>	<b>4</b>
<b>5</b>	<b>EXAMPLE</b>	<b>5</b>
<b>6</b>	<b>BUGS</b>	<b>6</b>
<b>7</b>	<b>SEE ALSO</b>	<b>7</b>
<b>8</b>	<b>COPYING</b>	<b>8</b>

## Chapter 1

# SYNOPSIS

```
#include <atmi.h>
```

```
int tpacall(char *svc, 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

Call the XATMI service by given *svc* name asynchronously. The *data* is optional input XATMI buffer. If it is not a NULL, then it must be allocated with **tpcalloc()** call, *len* is used for buffer types such **CARRAY**, where buffer length is not described by type. If call succeeds, the function return the call descriptor. With call descriptor it is possible later to receive the response by using **tpgetrply()** If **TPNOTRAN** is not specified and current process is in global transaction, then system will make destination process run in same destination process.

### Valid flags

**TPNOTRAN** Do not call service in transaction mode. This is effective in case if caller process is running in transaction mode, but destination process shall not run in the same global transaction

**TPSIGRSTR** Restart the system call in progress if interrupted by signal handler. This affects only underlying mq\_\* function calls.

**TPNOBLOCK** In case of target service queue is full, do not wait on queue, but return error. The error code for this situation is **TPEBLOCK**.

**TPNOREPLY** Do not expect reply from the call. This works with principle "send and forget". In this case return value for successful call is **0**.

## Chapter 3

# RETURN VALUE

On success, **tpacall()** return call descriptor (>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.

**TPEINVAL** Invalid parameter is given to function. Either service name is NULL or flags does not allow to change the value.

**TPENOENT** No service (*svc* parameter) advertised in system.

**TPETIME** Service did not reply in given time (*NDRX\_TOUT*).

**TPESVCFAIL** Service returned *TPFAIL*. This is application level failure.

**TPESVCERR** System level service failure. Server died during the message presence in service queue.

**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.

**TPEBLOCK** Service queue was full and **TPNOBLOCK** flag was specified.

**TPNOTIME** Do not expire call by server process, if message age is older than *NDRX\_TOUT* timeout (or timeout overridden by **tptoutset(3)**).

## Chapter 5

# EXAMPLE

See `atmitest/test001_basiccall/atmiclt1.c` for sample code.



## Chapter 6

# BUGS

Report bugs to [support@mavimax.com](mailto:support@mavimax.com)

## Chapter 7

## SEE ALSO

**tpcall(3) tpgetrply(3) tpcancel(3) tptoutset(3)**

## **Chapter 8**

# **COPYING**

© Mavimax, Ltd