

# TPCONNECT(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 tpconnect(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

Connect to conversational service. This is first step to enter in two way streamed communications between caller client and server. The service is given in *svc* parameter. Destination service must be programmed that way, that it supports conversations. *data* is optional data buffer that must be allocated by **tpcalloc()**, and the *len* parameter is used only in case if *data* is not NULL and buffer is not self describing the length, i.e. **CARRAY**.

### 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.

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

**TPNOTIME** Ignore timeout setting (**NDRX\_TOUT** env variable). Wait for reply for infinitely. Global transaction time-out still applies.

**TPTRANSUSPEND** Suspend the current transaction in progress and continue it with destination process. This is suitable in cases when **XA** adapter does not allow multiple processes/sessions to have active same transaction in the same transaction branch.

**TPSENDONLY** At the call point, caller want's to enter in send only mode.

**TPRECVONLY** At the call point, caller want's to enter in receive only mode. **TPSENDONLY** and **TPRECVONLY** are each other exclusive and cannot be combined.

## Chapter 3

# RETURN VALUE

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

## Chapter 4

# ERRORS

Note that **tpsterror()** 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 *data* is not NULL, but not allocated by **tpalloc()**

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

**TPELIMIT** Max number of connections are reached. Currently max number of connections is limited to 5 (**MAX\_CONNECTIONS** macro).

**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.

---

## Chapter 5

# EXAMPLE

See `atmitest/test003_basicconvers/atmict3.c` for sample code.



## Chapter 6

# BUGS

Report bugs to [madars.vitolins@gmail.com](mailto:madars.vitolins@gmail.com)

## Chapter 7

## SEE ALSO

`tpsend(3)` `tprecv(3)` `tpdiscon(3)`

## Chapter 8

# COPYING

© ATR Baltic, SIA