

**TPINIT(3)**

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<a href="#">1</a>	<a href="#">SYNOPSIS</a>	<a href="#">1</a>
<a href="#">2</a>	<a href="#">DESCRIPTION</a>	<a href="#">2</a>
<a href="#">3</a>	<a href="#">RETURN VALUE</a>	<a href="#">3</a>
<a href="#">4</a>	<a href="#">ERRORS</a>	<a href="#">4</a>
<a href="#">5</a>	<a href="#">EXAMPLE</a>	<a href="#">5</a>
<a href="#">6</a>	<a href="#">BUGS</a>	<a href="#">6</a>
<a href="#">7</a>	<a href="#">SEE ALSO</a>	<a href="#">7</a>
<a href="#">8</a>	<a href="#">COPYING</a>	<a href="#">8</a>

## Chapter 1

# SYNOPSIS

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

```
int tpinit(TPINIT *tpinfo);
```

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 intended to be called by XATMI clients, to initialize XATMI session. *tpinfo* is defined as:

```
struct  tpinfo_t
{
    char  username[MAXTIDENT+2];
    char  cltname[MAXTIDENT+2];
    char  passwd[MAXTIDENT+2];
    char  grpname[MAXTIDENT+2];
    long  flags;
    long  datalen;
    long  data;
};
typedef struct  tpinfo_t TPINIT;
```

If *tpinfo*→*cltname* is set, then it is using for client process identification. If it is not set, then client process is identified as executable name. Other fields of *tpinfo* are reserved for future use. *tpinfo* can be set to **NULL**. Function can be used also by XATMI server's worker threads, to initialize them. However call of **tpinit()** is optional, as it is automatically issued, then any of *tp*\* commands are invoked. At current Enduro/X version it is recommended to use **tpinit()** with **NULL** parameter. Also when client process does threading, then each thread needs to call **tpinit()** or it will be automatically invoked as described before. Each new thread has its own thread local storage (TLS), which describes Enduro/X state, thus new thread will get un-initialised TLS which results in need of calling **tpinit()**.

Internally function opens any Posix queues needed by client, reads the environment variables and initializes XA sub-system, if configured and not set to lazy.

### Valid flags for TPINIT.flags

**TPU\_IGN** Ignore (drop) incoming unsolicited messages.

## Chapter 3

# RETURN VALUE

On success, **tpinit()** returns 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** Environment variables not configured, see **ex\_env(5)** page.

**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/test017_srvthread/atmisv17.c` for sample code.



## Chapter 6

# BUGS

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

## Chapter 7

## SEE ALSO

`ex_env(5)`

## Chapter 8

# COPYING

© Mavimax, Ltd