

**TX\_INFO(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 <tx.h>
```

```
int tx_info(TXINFO *info);
```

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 returns current transaction status information. This includes current control settings, timeout active XID. Information is returned in following structure

```
struct tx_info_t {
    XID xid;
    COMMIT_RETURN when_return;
    TRANSACTION_CONTROL transaction_control;
    TRANSACTION_TIMEOUT transaction_timeout;
    TRANSACTION_STATE transaction_state;
};
typedef struct tx_info_t TXINFO;
```

Field *xid* is returned only in case if there is active transaction. If there is no transaction, then empty **xid** is returned (xid.formatID set to -1).

*transaction\_state* may contain following values:

- **TX\_ACTIVE** transaction is active.
- **TX\_TIMEOUT\_ROLLBACK\_ONLY** transaction for current control thread is active, but transaction manager is not aware of transaction, thus it is assumed that there was timeout and transaction was rolled back.
- **TX\_ROLLBACK\_ONLY** transaction in current control thread is marked as rollback only.

Field *when\_return* returns either **TX\_COMMIT\_COMPLETED** or **TX\_COMMIT\_DECISION\_LOGGED**, set by **tx\_set\_commit\_r**.

Field *transaction\_control* returns either **TX\_UNCHAINED** or **TX\_CHAINED** set by **tx\_set\_transaction\_control(3)**.

Field *transaction\_timeout* returns current timeout setting in seconds applied by **tx\_set\_transaction\_timeout(3)**.

---

## Chapter 3

# RETURN VALUE

On success, and if there is transaction started, then **tx\_info()** return **1**, if there is no local transaction, **0** is returned. On error, error code is returned.

## Chapter 4

# ERRORS

**TX\_PROTOCOL\_ERROR** The function was called in an improper context (for example, the caller has not yet called `tx_open(3)`).

**TX\_FAIL** Transaction manager (**tmsrv(8)**) failed to return transaction information. For exact information, see either process and/or transaction manager logs

---

## Chapter 5

# EXAMPLE

See `atmitest/test021_xafull/atmict21tx.c` for sample code.



## Chapter 6

# BUGS

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

## Chapter 7

## SEE ALSO

`tx_begin(3)` `tpcommit(3)` `tx_rollback(3)` `tx_info(3)` `tx_close(3)` `tx_set_transaction_timeout(3)` `tx_set_transaction_control(3)`  
`tx_set_commit_return(3)`

## **Chapter 8**

# **COPYING**

© Mavimax, Ltd