

## TX\_OPEN(3)

<b>REVISION HISTORY</b>
-------------------------

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_open(void);
```

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

For XATMI server link with *-latmisrv|-latmisrvnomain|-latmisrvinteg -latmi -lubf -lnstd -lpthread -lrt -lm*

---

## Chapter 2

# DESCRIPTION

Connect to the XA resource manager. Note that environment for process must be configured, so that Enduro/X can load correct resource manager drivers. See the **ex\_env(5)** man page for environment variables, that must be set. Basically Enduro/X will dynamically load library set in **NDRX\_XA\_DRIVERLIB** env variable. This is the driver which provides in unified way of getting **xa\_switch**. As the **xa\_switches** can be located in different Resource Managers, the second library might be need to load and that is marked in **NDRX\_XA\_RMLIB** env variable. The resource manager ID is set in **NDRX\_XA\_RES\_ID** env variable. If your application needs different resource managers or different transaction branches, then you can use environment variable override functionality, see **ex\_envover(5)** and **ndrxconfig.xml(5)**. XA connection strings for Enduro/X drivers are stored into **NDRX\_XA\_OPEN\_STR** and **NDRX\_XA\_CLOSE\_STR** variables. Enduro/X will try to init XA sub-system as soon as possible (i.e. load drivers, etc.). If to delay initial XA drivers loading till actual **tx\_open(3)** call, then environment variable **NDRX\_XA\_LAZY\_INIT** should be set to 1.

This function at lowest level, basically calls XA switches **xa\_open\_entry()** function with **NDRX\_XA\_OPEN\_STR** string.

If **tx\_open(3)** was already called and XA is initialized, then function will succeed, and no error will be generated.

TX API is base on TP API. This function is based on **tpclose(3)** and it is possible to mix these two API kinds.

## Chapter 3

# RETURN VALUE

On success, `tx_close()` return `TX_OK`; on error, error code is returned

---

## Chapter 4

# ERRORS

**TX\_FAIL** Failed to open XA resource (cannot load Enduro/X XA driver, cannot load resource manager driver or resource manager returned)

---

## 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_close(3)` `topen(3)` `tx_open(3)` `tx_commit(3)` `tx_rollback(3)` `tx_info(3)`

---

## **Chapter 8**

# **COPYING**

© Mavimax, Ltd

---