

TX_COMMIT(3)

| REVISION HISTORY | | | |
|------------------|------|-------------|------|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

Contents

| | | |
|-------------------|------------------------------|-------------------|
| 1 | SYNOPSIS | 1 |
| 2 | DESCRIPTION | 2 |
| 3 | RETURN VALUE | 3 |
| 4 | ERRORS | 4 |
| 5 | EXAMPLE | 5 |
| 6 | BUGS | 6 |
| 7 | SEE ALSO | 7 |
| 8 | COPYING | 8 |

Chapter 1

SYNOPSIS

```
#include <tx.h>
```

```
int tx_rollback(void);
```

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 rolls back global transaction associated with current thread previously started by **tx_begin(3)**. If transaction control was changed by **tx_set_transaction_control(3)** flag **TX_CHAINED**, the new transaction is started immediately after the rollback.

Function use following attributes set by:

- **tx_set_transaction_control(3)** - if applied with **TX_CHAINED**, new transaction is started. The default is **TX_UNCHAINED** - meaning after rollback, control thread is no more in global transaction.

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

Chapter 3

RETURN VALUE

On success, **tx_rollback()** return **TX_OK**; on error, error code is returned

Chapter 4

ERRORS

TX_NO_BEGIN The transaction committed successfully; however, a new transaction could not be started and the caller is no longer in transaction mode. This return value occurs only when the `transaction_control` characteristic is **TX_CHAINED**.

TX_MIXED The transaction was partially committed and partially rolled back. In addition, if the `transaction_control(3)` characteristic is **TX_CHAINED**, a new transaction is started.

TX_MIXED_NO_BEGIN The transaction was partially committed and partially rolled back. In addition, a new transaction could not be started and the caller is no longer in transaction mode. This return value can occur only when the `transaction_control(3)` characteristic is **TX_CHAINED**.

TX_HAZARD Due to a failure, the transaction may have been partially committed and partially rolled back. In addition, if the `transaction_control(3)` characteristic is **TX_CHAINED**, a new transaction is started.

TX_HAZARD_NO_BEGIN Due to a failure, the transaction may have been partially committed and partially rolled back. In addition, a new transaction could not be started and the caller is no longer in transaction mode. This return value can occur only when the `transaction_control(3)` characteristic is **TX_CHAINED**.

TX_PROTOCOL_ERROR The function was called in an improper context (for example, the caller is not in transaction mode). The caller's state with respect to the transaction is not changed.

TX_FAIL Either the transaction manager or one or more of the resource managers encountered a fatal error. The nature of the error is such that the transaction manager and/or one or more of the resource managers can no longer perform work on behalf of the application. The exact nature of the error is determined in a product-specific manner. The caller's state with respect to the transaction is unknown.

Chapter 5

EXAMPLE

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

Chapter 6

BUGS

Report bugs to support@mavimax.com

Chapter 7

SEE ALSO

`tx_begin(3)` `tpabort(3)` `tx_commit(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