

Transaction Inventory Page - type 0x03

Every database has at least one [Transaction Inventory Page \(TIP\)](#).

The highest possible [transaction](#) number is 2,147,483,647 or 0x7fffffff in a 32-bit system. Once you hit this transaction, no more can be created and the database needs to be [shutdown](#), [backed up](#) and then [restored](#) to reset the transaction numbers back to zero. The reason it has this maximum value is simply because the code for allocating transaction numbers uses a signed value.

The C code representation of the TIP page is:

```
struct tx_inv_page
{
    pag tip_header;
    SLONG tip_next;
    UCHAR tip_transactions[1];
};
```

Tip_header: The TIP starts off with a [standard page header](#).

Tip_next: Four bytes, signed. Bytes 0x10 - 0x13 on the page. This is the page number of the next TIP page, if one exists, within the database. Zero here indicates that the current TIP page is the last TIP page.

Tip_transactions: Bytes 0x14 onwards. The remainder of the page, is an array of two bit values where each pair of bits represents a transaction and its status. Each transaction can have one of 4 status values:

- 0x00 - this transaction is [active](#), or has not yet started.
- 0x01 - this transaction is [in limbo](#). A [two phase transaction](#) has committed the first phase but the second phase has not committed.
- 0x02 - this transaction is dead (was [rolled back](#)).
- 0x03 - this transaction was [committed](#).

Looking at a hex dump of the first few bytes of a new database, which has had a few transactions run against it, we see the following:

Offset	Data	Description
-----	-----	-----
000a0014	fc ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff
	tip_transactions[]	
000a0024	ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff
000a0034	ff ff ff ff ff ff ff ff	ff ff ff ff ff 00 00 00

Now, if a new transaction starts we won't see any changes because a live transaction and one that has not started yet, shows up as two zero bits in the [tip_transactions array](#). However, if it commits, limbo's or rolls back, we should see a change. The following is the above database after a session connected using [isql](#) and immediately exited without doing anything:

Offset	Data	Description

000a0014	fc ff ff ff ff ff ff ff ff ff ff ff ff ff ff	
	tip_transactions[]	
000a0024	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff	
000a0034	ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00	

You can see that it looks remarkably like loading up a connection to `isql` and then exiting actually executes 4 separate transactions. We can see at the end of the last line that one byte has changed from 0x00 to 0xff and with 2 bits per transaction, that equates to 4 separate transactions, all of which committed.

Other tools may run fewer, or indeed more, transactions just to connect to a database and do whatever it is that they have to do to initialise themselves.

From:
<http://ibexpert.com/docu/> - IBExpert

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-08-firebird-documentation:firebird-internals:transaction-inventory-page-type0x03>

Last update: 2023/07/11 10:45

