

Database Header Page - type 0x01

The first page of the first file of a Firebird database is a very important page. It holds data that describes the database, where its other files are to be found, [shadow file names](#), database [page size](#), [ODS version](#) and so on. On startup, the Firebird engine reads the first part (1,024 bytes) of the first page in the first file of the database and runs a number of checks to ensure that the file is actually a database and so on. If the database is multi-file, then each file will have a header page of its own.

The C code representation of the database header page is:

```
struct header_page
{
    pag hdr_header;
    USHORT hdr_page_size;
    USHORT hdr_ods_version;
    SLONG hdr_Pages;
    ULONG hdr_next_page;
    SLONG hdr_oldest_transaction;
    SLONG hdr_oldest_active;
    SLONG hdr_next_transaction;
    USHORT hdr_sequence;
    USHORT hdr_flags;
    SLONG hdr_creation_date[2];
    SLONG hdr_attachment_id;
    SLONG hdr_shadow_count;
    SSHORT hdr_implementation;
    USHORT hdr_ods_minor;
    USHORT hdr_ods_minor_original;
    USHORT hdr_end;
    ULONG hdr_page_buffers;
    SLONG hdr_bumped_transaction;
    SLONG hdr_oldest_snapshot;
    SLONG hdr_backup_pages;
    SLONG hdr_misc[3];
    UCHAR hdr_data[1];
};
```

[back to top of page](#)

Hdr_header: The database header page has a [standard page header](#), as do all pages.

Hdr_page_size: Two bytes, unsigned. Bytes 0x10 - 0x11 on the page. This is the page size, in bytes, for each and every page in the database.

Hds_ods_version: Two bytes, unsigned. Bytes 0x12 and 0x13 on the page. The [ODS major version](#) for the database. The format of this word is the ODS major version ANDed with the Firebird flag of 0x8000. In the example below, the value is 0x800b for ODS version 11. The minor ODS version is held elsewhere in the header page - see [hdr_ods_minor](#) below.

Hdr_pages: Four bytes, signed. Bytes 0x14 - 0x17 on the page. This is the page number of the first

pointer page for the **table** named **RDB\$PAGES**. When this location is known, the database engine uses it to determine the locations of all other **metadata** pages in the database. This field is only valid in the header page of the *first file* in a multi-file database. The remaining files in the database have this field set to zero.

Hdr_next_page: Four bytes, unsigned. Bytes **0x18 - 0x1b** on the page. The page number of the header page in the next file of the database - if this is a multi-file database. Zero otherwise.

Hdr_oldest_transaction: Four bytes, signed. Bytes **0x1c - 0x1f** on the page. The transaction ID of the oldest active (ie, uncommitted - but may be **in limbo** or **rolled back**) transaction against this database. This field is only valid in the header page of the *first file* in a multi-file database. The remaining files in the database have this field set to zero.

Hdr_oldest_active: Four bytes, signed. Bytes **0x20 - 0x23** on the page. The transaction ID of the **oldest active transaction** against this database, when any active transaction started. This field is only valid in the header page of the *first file* in a multi-file database. The remaining files in the database have this field set to zero.

Hdr_next_transaction: Four bytes, signed. Bytes **0x24 - 0x27** on the page. The transaction ID that will be assigned to the next transaction against this database. This field is only valid in the header page of the *first file* in a multi-file database. The remaining files in the database have this field set to zero.

Hdr_sequence: Two bytes, unsigned. Bytes **0x28** and **0x29** on the page. The sequence number of this file within the database.

[back to top of page](#)

Hdr_flags: Two bytes, unsigned. Bytes **0x2a** and **0x2b** on the page. The database flags. The bits in the flag bytes are used as follows:

Flag name	Flag value	Description
hdr_active_shadow	0x01 (bit 0)	This file is an active shadow file .
hdr_force_write	0x02 (bit 1)	The database is in forced writes mode.
Unused	0x04 (bit 2)	Was previously for short term journalling, no longer used.
Unused	0x08 (bit 3)	Was previously for long term journalling, no longer used.
hdr_no_checksums	0x10 (bit 4)	Don't calculate checksums.
hdr_no_reserve	0x20 (bit 5)	Don'r reserve space for record versions in pages.
Unused	0x40 (bit 6)	Was used to indicate that the shared cache file was disabled.
hdr_shutdown_mask (bit one of two)	0x1080 (bits 7 and 12)	Used with bit 12 (see below) to indicate the database shutdown mode.
hdr_sql_dialect_3	0x100 (bit 8)	If set, the database is using SQL dialect 3 .
hdr_read_only	0x200 (bit 9)	Database is in read only mode.
hdr_backup_mask	0xC00 (bits 10 and 11)	Indicates the current backup mode.
hdr_shutdown_mask (bit two of two)	0x1080 (bits 7 and 12)	Used with bit 7 (see above) to indicate the database shutdown mode.

The final two database flags use a pair of bits to indicate various states of [backup](#) and [shutdown](#).

Hdr_backup_mask: These two bits determine the current database backup mode, as follows:

Flag value	Description
0x00 (Both bits zero)	Database is not in backup mode. User changes are written directly to the database files.
0x400	The database is running in backup mode so all changed made by the users are written to the diff file.
0x800	The database is still in backup mode, but changes are being merged from the diff file into the main pages.
0xC00	The current database state is unknown and changes need to be read from disc.

Hdr_shutdown_mask: The shutdown mask uses two bits to indicate the current database shutdown status, as follows:

Flag value	Description
0x00 (Both bits 7 and 12 are zero)	Database is not shutdown. Any valid user can connect.
0x80	The database has been shutdown to, or started up in multi-user maintenance mode. The database can only be connceted to by SYSDBA or the database owner.
0x1000	The database has been fully shutdown. No connections are permitted.
0x1080	The database has been shutdown to, or started up in single-user maintenance mode. Only one SYSDBA or database owner connection is permitted.

[back to top of page](#)

hdr_creation_date: Eight bytes, signed. Bytes 0x2c - 0x33 on the page. The data and time (in Firebird's own internal format) that the database was either originally created/rewritten or created from a backup.

hdr_attachment_id: Four bytes, signed. Bytes 0x34 - 0x37 on the page. The ID number that will be assigned to the next connection to this database. As this is signed, the maximum value here is 232-1 and any database which reaches this maximum value must be backed up and restored in order to allow new connections. This field is only valid in the header page of the first file in a multi-file database. The remaining files in the databasewhich have this field set to zero.

hdr_shadow_count: Four bytes, signed. Bytes 0x38 - 0x3c on the page. Holds the event count for shadow file synchronisation for this database. The remaining files in the database have this field set to zero.

hdr_implementation: Two bytes, signed. Bytes 0x3c and 0x3d on the page. This is a number which indicates the environment on which the database was originally created. It is used to determine if the database file can be used sucessfully on the current hardware. This avoids problems caused by [little-endian](#) numerical values as compared with [big-endian](#), for example.

hdr_ods_minor: Two bytes, unsigned. Bytes 0x3e and 0x3f on the page. The current [ODS minor version](#).

hdr_ods_minor_original: Two bytes, unsigned. Bytes 0x40 and 0x41 on the page. The ODS minor version when the database was originally created.

hdr_end: Two bytes, unsigned. Bytes 0x42 and 0x43 on the page. The offset on the page where the `hdr_data` finishes. In other words, where a new clumplet will be stored if required. This is effectively a pointer to the current location of `hdr_end` (see clumplet details below) on this page.

hdr_page_buffers: Four bytes, unsigned. Bytes 0x44 - 0x47 on the page. Holds the number of `buffers` to be used for the `database cache`, or zero to indicate that the default value should be used. This field is only valid in the header page of the first file in a multi-file database. The remaining files in the database have this field set to zero.

hdr_bumped_transaction: Four bytes, signed. Bytes 0x48 - 0x4b on the page. Used to be used for the bumped transaction ID for log optimisation, but is currently always set to 0x01. This field is only valid in the header page of the first file in a multi-file database. The remaining files in the database have this field set to zero.

hdr_oldest_snapshot: Four bytes, signed. Bytes 0x4c - 0x4f on the page. Holds the transaction number for the oldest snapshot of active transactions. This is also documented as the confusing and redundant variant of `Oldest Active Transaction`.

hdr_backup_pages: Four bytes, signed. Bytes 0x50 - 0x53 on the page. Holds the number of pages in the database currently locked for a backup using `nbackup`. This field is only valid in the header page of the first file in a multi-file database. The remaining files in the database have this field set to zero.

Hdr_misc: Twelve bytes. Bytes 0x54 - 0x5f on the page. Set to zero. These 12 bytes are currently unused.

[back to top of page](#)

The following is an example of a header page from a multi-file database on a little-endian system:

```
00000000 01 00 39 30 08 00 00 00 00 00 00 00 00 00 00 Standard header
00000010 00 10                                     hdr_page_size
00000012 0b 80                                     hdr_ods_version
00000014 03 00 00 00                               hdr_PAGES
00000018 00 00 00 00                               hdr_next_page
0000001c 01 00 00 00
hdr_oldest_transaction
00000020 02 00 00 00
hdr_oldest_active
00000024 05 00 00 00
hdr_next_transaction
00000028 00 00                                     hdr_sequence
0000002a 00 01                                     hdr_flags
0000002c 5e d7 00 00 f4 79 00 23
hdr_creation_date
00000034 01 00 00 00
hdr_attachment_id
00000038 00 00 00 00
hdr_shadow_count
0000003c 13 00
hdr_implementation
```


The format of each clumplet is as follows:

Type_byte: The first byte - unsigned - in each clumplet determines the type of data stored within the clumplet. There are a number of different clumplet types:

Type name	Value	Description
HDR_end	0x00	End of clumplets.
HDR_root_file_name	0x01	Original name of the root file for this database.
HDR_journal_server	0x02	Name of the journal server.
HDR_file	0x03	Secondary file name.
HDR_last_page	0x04	Last logical page of the current file.
HDR_unlicemsed	0x05	Count of unlicensed activity. No longer used.
HDR_sewwp_interval	0x06	Number of transactions between sweep .
HDR_log_name	0x07	Replay log name.
HDR_journal_file	0x08	Intermediate journal file name.
HDR_password_file_key	0x09	Key to compare with the password database.
HDR_backup_info	0x0a	Write Ahead Log (WAL) backup information. No longer used.
HDR_cache_file	0x0b	Shared cache file. No longer used.
HDR_difference_file	0x0c	Diff file used during the times when the database is in backup mode.
HDR_backup_guid	0x0d	UID generated when database is in backup mode. Overwritten on subsequent backups.

Length_byte: The second byte - again unsigned - in each clumplet specifies the size of the data that follows.

Data: The next n bytes are the actual clumplet data. The miscellaneous data stored in the header from the above database, at `hdr_data`, is shown below.

```

00000060 03                                     Type = HDR_file
00000061 2b                                     Length = 43 bytes
00000062 2f 75 30 30 2f 66 69 72 65 62 69 72 64 2f   Data
'/u00/firebird/'
00000070 64 61 74 61 62 61 73 65 73 2f 6d 75 6c 74 69 5f
'databases/multi_'
00000080 65 6d 70 6c 6f 79 65 65 2e 66 64 62 31
'employee.fdb1'

0000008d 04                                     Type =
HDR_last_page
0000008e 04                                     Length = 4 bytes
0000008f a2 00 00 00   Data 0xa2 = 162

00000093 00                                     Type = HDR_end.

```

From the above we can see that in our multi-file database:

- The next file (after this one) is named `'/u00/firebird/databases/multi_employee.fdb1'`.
- The current file has 162 pages only - and with a 4Kb page size this means that the current file should be 663,552 bytes in size, which a quick run of `ls -l` will confirm.

- HDR_end is located at offset 0x93 in the page, exactly as the header field `hdr_end` told us. (See above).

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

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-08-firebird-documentation:firebird-internals:database-header-page-type0x01>

Last update: **2023/07/11 10:25**

