# [DBF File format](https://www.dbf2002.com/dbf-file-format.html" \l ":~:text=A%20DBF%20file%20consists%20of,actual%20text%20of%20the%20fields.)

A **DBF file** consists of a header record and data records. The header record defines the structure of dbf file and contains any other information related to the table. The header record starts at file position zero. Data records follow the header, in consecutive bytes, and contain the actual text of the fields.

**Note:** The data in dbf file starts at the position indicated in bytes 8 to 9 of the header record. Data records begin with a delete flag byte. If this byte is an ASCII space (0x20), the record is not deleted. If the first byte is an asterisk (0x2A), the record is deleted. The data from the fields named in the field subrecords follows the delete flag.

（dbf 文件中的数据从头记录的第 8 到 9 字节指示的位置开始。 数据记录以删除标志字节开始。 如果这个字节是一个 ASCII 空格（0x20），则记录不会被删除。 如果第一个字节是星号 (0x2A)，则记录被删除。 来自字段子记录中命名的字段的数据遵循删除标志。）

The length of a record, in bytes, is determined by summing the defined lengths of all fields. Integers in dbf files are stored with the least significant byte first（little-endian）.（记录的长度（以字节为单位）是通过将所有字段的定义长度相加来确定的。 dbf 文件中的整数首先存储最低有效字节。）

## DBF File Header

|  |  |
| --- | --- |
| **Byte offset** | **Description** |
| 0 | DBF File type: 0x02 FoxBASE 0x03 FoxBASE+/Dbase III plus, no memo 0x30 Visual FoxPro 0x31 Visual FoxPro, autoincrement enabled 0x32 Visual FoxPro with field type Varchar or Varbinary 0x43 dBASE IV SQL table files, no memo 0x63 dBASE IV SQL system files, no memo 0x83 FoxBASE+/dBASE III PLUS, with memo 0x8B dBASE IV with memo 0xCB dBASE IV SQL table files, with memo 0xF5 FoxPro 2.x (or earlier) with memo 0xE5 HiPer-Six format with SMT memo file 0xFB FoxBASE |
| 1 - 3 | Last update (YYMMDD) |
| 4 - 7 | Number of records in file |
| 8 - 9 | Position of first data record |
| 10 - 11 | Length of one data record, including delete flag |
| 12 - 27 | Reserved |
| 28 | Table flags: 0x01 file has a structural .cdx 0x02 file has a Memo field 0x04 file is a database (.dbc) This byte can contain the sum of any of the above values. For example, the value 0x03 indicates the table has a structural .cdx and a Memo field. |
| 29 | Code page mark |
| 30 - 31 | Reserved, contains 0x00 |
| 32 - n | Field subrecords The number of fields determines the number of field subrecords. One field subrecord exists for each field in the table. |
| n+1 | Header record terminator (0x0D) |
| n+2 to n+264 | Visual Foxpro only: A 263-byte range that contains the backlink, which is the relative path of an associated database (.dbc) file, information. If the first byte is 0x00, the file is not associated with a database. Therefore, database files always contain 0x00. |

## Field Subrecords Structure

|  |  |
| --- | --- |
| **Byte offset** | **Description** |
| 0 - 10 | Field name with a maximum of 10 characters. If less than 10, it is padded with null characters (0x00). |
| 11 | Field type: C - Character Y - Currency (Visual Foxpro) N - Numeric F - Float D - Date T - DateTime (Visual Foxpro) B - Double (Visual Foxpro) I - Integer (Visual Foxpro) L - Logical M - Memo G - General C - Character (binary) M - Memo (binary) P - Picture + - Autoincrement (dBase Level 7) O - Double (dBase Level 7) @ - Timestamp (dBase Level 7) V - Varchar type (Visual Foxpro, character field with variable size, real size in the last byte of field) |
| 12 - 15 | Displacement of field in record |
| 16 | Length of field (in bytes) |
| 17 | Number of decimal places |
| 18 | Field flags: 0x01 System Column (not visible to user) 0x02 Column can store null values 0x04 Binary column (for CHAR and MEMO only) 0x06 (0x02+0x04) When a field is NULL and binary (Integer, Currency, and Character/Memo fields) 0x0C Column is autoincrementing |
| 19 - 22 | Value of autoincrement Next value |
| 23 | Value of autoincrement Step value |
| 24 - 31 | Reserved |

## Limitations for dBase III format

* Maximum number of fields: 128
* Maximum length of Character field: 254

## Limitations for Visual Foxpro format

* Maximum number of fields: 255
* Maximum length of Character field: 254
* Maximum # of records per table file: 1 billion