Data File Header Structure for the dBASE Version 7 Table File

Note: Unless prefaced by "0x", all numbers specified in the Description column of the following tables are decimal.

1.1 Table File Header

Byte	Contents	Description
0	1 byte	Valid dBASE for Windows table file, bits 0-2 indicate version number: 3 for dBASE Level 5, 4 for dBASE Level 7. Bit 3 and bit 7 indicate presence of a dBASE IV or dBASE for Windows memo file; bits 4-6 indicate the presence of a dBASE IV SQL table; bit 7 indicates the presence of any .DBT memo file (either a dBASE III PLUS type or a dBASE IV or dBASE for Windows memo file).
1-3	3 bytes	Date of last update; in YYMMDD format. Each byte contains the number as a binary. YY is added to a base of 1900 decimal to determine the actual year. Therefore, YY has possible values from 0x00-0xFF, which allows for a range from 1900-2155.
4-7	32-bit number	Number of records in the table. (Least significant byte first.)
8-9	16-bit number	Number of bytes in the header. (Least significant byte first.)
10-11	16-bit number	Number of bytes in the record. (Least significant byte first.)
12-13	2 bytes	Reserved; filled with zeros.

14	1 byte	Flag indicating incomplete dBASE IV transaction.
15	1 byte	dBASE IV encryption flag.
16-27	12 bytes	Reserved for multi-user processing.
28	1 byte	Production MDX flag; 0x01 if a production .MDX file exists for this table; 0x00 if no .MDX file exists.
29	1 byte	Language driver ID.
30-31	2 bytes	Reserved; filled with zeros.
32-63	32 bytes	Language driver name.
64-67	4 bytes	Reserved.
68-n	48 bytes each	Field Descriptor Array (see 1.2).
n+1	1 byte	0x0D stored as the Field Descriptor terminator.
n+2	See below for calculations of size	Field Properties Structure

n above is the last byte in the field descriptor array. The size of the array depends on the number of fields in the table file.

1. 2 Field Descriptor Array

(One for each field in the table)

Byte	Contents	Description
0-31	32 bytes	Field name in ASCII (zero-filled).
32	1 byte	Field type in ASCII (B, C, D, N, L, M, @, I, +, F, 0 or G).
33	1 byte	Field length in binary.

34	1 byte	Field decimal count in binary.
35-36	2 bytes	Reserved.
37	1 byte	Production .MDX field flag; 0x01 if field has an index tag in the production .MDX file; 0x00 if the field is not indexed.
38-39	2 bytes	Reserved.
40-43	4 bytes	Next Autoincrement value, if the Field type is Autoincrement, 0x00 otherwise.
44-47	4 bytes	Reserved.

1.3 Field Properties Structure

This contains a header describing the Field Properties array, followed by the actual array, followed by property data. It is contained in the .DBF header and comes immediately after the Field Descriptor terminator (See Table 1.1).

Byte	Contents	Description
0-1	16-bit number	Number of Standard Properties.
2-3	16-bit number	Start of Standard Property Descriptor Array. (see 1.3.1)
4-5	16-bit number	Number of Custom Properties.
6-7	16-bit number	Start of Custom Property Descriptor Array. (see 1.3.2)
8-9	16-bit number	Number of Referential Integrity (RI) properties.
10-11	16-bit number	Start of RI Property Descriptor Array. (see 1.3.3)

12-13	16-bit number	Start of data - this points past the Descriptor arrays to data used by the arrays - for example Custom property names are stored here.
14-15	16-bit number	Actual size of structure, including data (Note: in the .DBF this will be padded with zeroes to the nearest 0x200, and may have 0x1A at the end). If the structure contains RI data, it will not be padded.
16-n	15 bytes each	Standard Property Descriptor Array (n = (15*number of standard properties) + 16). (see 1.3.1)
(n+1)-m	14 bytes each	Custom Property Descriptor Array (m = n+ 14*number of custom properties). (see 1.3.2)
(m+1)-o	22 bytes each	RI Property Descriptor Array (o = m+ 22*number of RI properties). (see 1.3.3)

1.3.1 Standard Property and Constraint Descriptor Array

Byte	Contents	Description
0-1	16-bit number	Generational number. More than one value may exist for a property. The current value is the value with the highest generational number.
2-3	16-bit number	Table field offset - base one. 01 for the first field in the table, 02 for the second field, etc. Note: this will be 0 in the case of a constraint.
4	8-bit number	Which property is described in this record: 01 Required 02 Min 03 Max 04 Default 06 Database constraint
		Field Type: 00 No type - constraint 01 Char 02 Numeric

5	1 byte	03 Memo 04 Logical 05 Date 06 Float 08 OLE 09 Binary 11 Long 12 Timestamp 13 Double 14 AutoIncrement (not settable from the Inspector)
6	1 byte	0x00 if the array element is a constraint, 0x02 otherwise.
7-10	4 bytes	Reserved
11-12	16-bit number	Offset from the start of this structure to the data for the property. The Required property has no data associated with it, so it is always 0.
13-14	16-bit number	Width of database field associated with the property, and hence size of the data (includes 0 terminator in the case of a constraint).

1.3.2 Custom Property Descriptor Array

Byte	Contents	Description
0-1	16-bit number	Generational number. More than one value may exist for a property. The current value is the value with the highest generational number.
2-3	16-bit number	Table field offset - base one. 01 for the first field in the table, 02 for the second field, etc.
		Field Type 01 Char 02 Numeric 03 Memo

4	1 byte	04 Logical 05 Date 06 Float 08 OLE 09 Binary 11 Long 12 Timestamp 13 Double 14 AutoIncrement (not settable from the Inspector)
5	1 byte	Reserved
6-7	16-bit number	Offset from the start of this structure to the Custom property name.
8-9	16-bit number	Length of the Custom property name.
10-11	16-bit number	Offset from the start of this structure to the Custom property data.
12-13	16-bit number	Length of the Custom property data (does not include null terminator).

1.3.3 Referential Integrity Property Descriptor Array

Byte	Contents	Description
0	8-bit number	0x07 if Master (parent), 0x08 if Dependent (child).
1-2	16-bit number	Sequential number, 1 based counting. If this number is 0, this RI rule has been dropped.
3-4	16-bit number	Offset of the RI rule name - 0 terminated.
5-6	16-bit number	Size of previous value.
7-8	16-bit number	Offset of the name of the Foreign Table - 0 terminated.

9-10	16-bit number	Size of previous value.
11	1 byte	Update & delete behaviour: Update Cascade 0x10 Delete Cascade 0x01
12-13	16-bit number	Number of fields in the linking key.
14-15	16-bit number	Offset of the Local Table tag name - 0 terminated.
16-17	16-bit number	Size of previous value.
18-19	16-bit number	Offset of the Foreign Table tag name - 0 terminated.
20-21	16-bit number	Size of previous value.

(Foreign = in the other table, Local = in this table)

Property Data

For standard properties, everything is stored exactly as it is in the Table records. Custom property data is stored as the Name string, followed immediately by the Value string, and a null terminator. The Constraint text is stored as a null-terminated string.

Table Records

The records follow the header in the table file. Data records are preceded by one byte, that is, a space (0x20) if the record is not deleted, an asterisk (0x2A) if the record is deleted. Fields are packed into records without field separators or record terminators. The end of the file is marked by a single byte, with the end-of-file marker, an OEM code page character value of 26 (0x1A).

Storage of dBASE Data Types

Except for autoincrement fields, all types are initialized to binary zeroes. In addition, any fields which have been assigned a default property will contain the default value.

Symbol	Data Type	Description
B	Binary, a	10 digits representing a .DBT block number. The number is stored as a string, right justified and padded

	string	with blanks.
C	Character	All OEM code page characters - padded with blanks to the width of the field.
D	Date	8 bytes - date stored as a string in the format YYYYMMDD.
N	Numeric	Number stored as a string, right justified, and padded with blanks to the width of the field.
L	Logical	1 byte - initialized to 0x20 (space) otherwise T or F.
M	Memo, a string	10 digits (bytes) representing a .DBT block number. The number is stored as a string, right justified and padded with blanks.
@	Timestamp	8 bytes - two longs, first for date, second for time. The date is the number of days since 01/01/4713 BC. Time is hours * 3600000L + minutes * 60000L + Seconds * 1000L
I	Long	4 bytes. Leftmost bit used to indicate sign, 0 negative.
+	Autoincrement	Same as a Long
F	Float	Number stored as a string, right justified, and padded with blanks to the width of the field.
O	Double	8 bytes - no conversions, stored as a double.
G	OLE	10 digits (bytes) representing a .DBT block number. The number is stored as a string, right justified and padded with blanks.

Binary, Memo, OLE Fields and .DBT Files

Binary, memo, and OLE fields store data in .DBT files consisting of blocks numbered sequentially (0, 1, 2, etc.). SET BLOCKSIZE determines the size of each block. The first block in the .DBT file, block 0, is the .DBT file header.

Each binary, memo, or OLE field of each record in the .DBF file contains the number of the block (in OEM code page values) where the field's data actually begins. If a field contains no data, the .DBF file contains blanks (0x20) rather than a number.

When data is changed in a field, the block numbers may also change and the number in the .DBF may be changed to reflect the new location.