Accessing *Starling* Databases

Rob Verhoeven

April 3, 2014

Abstract

This document contains information over the way information is stored in and can be extracted from *Starling* databases (`.dbf`). The information has been compiled using existing documentation on *dBase III* databases and reverse engineering efforts by mainly prof.dr. Andries Brouwer.

Document status

revision	date	reason for change
1.0	28–11–2013	initial document release
1.1	18–02–2014	update of special character table and addition of note on packing

Contents

1 *Starling* **3**

2 Database header **3**

3 Records **3**
3.1 Record header 3
3.2 Record contents 4

4 References to associated `.var` file **4**

5 Character encodings **5**
5.1 Single byte mode 5
5.2 Double byte mode 7

6 Layout codes **10**

7 Auxiliary files **10**
7.1 `.inf` file 10
7.2 `.prt` file 10

1 *Starling*

Starling is a tool for comparative linguists, which allows them to catalogue linguistic data in a flat database and apply a diverse range of procedures upon that data. The tool was created by Sergei Starostin, who passed away in 2005, and is currently maintained by his son George.[1]

Much credit goes to prof.dr. Andries Brouwer,[2] who, through personal correspondence, shared with me some of his code for extracting data from *Starling* databases, allowing me to profit from his reverse engineering efforts on the database structure and encoding schemes.

2 Database header

Starling databases (`.dbf`) are in an adapted *dBase III* database format. Going by the *dBase III* header as documented on the *DBF Viewer 2000* website[3], the header format seems to be as follows:

bytes	description
0	`.dbf` file type
	`0x03` *dBase III*
1	Year of last update (offset from 1900)
2	Month of last update
3	Day of last update
4–7	Number of data records
8–9	Offset of first data record (i.e. header size (HDR))
10–11	Record size (including delete flag)
12–27	Reserved bytes (unknown content)
28	Table bit flags (zero by default?)
	`0x01` has a database index (`.cdx`)
	`0x02` has a memo field
	`0x04` is a database container (`.dbc`)
29	Code page mark (zero by default)
30–31	Reserved bytes (0 and 1 by default, respectively)
32–HDR	Record structure information (see Sec. 3.1)

3 Records

To avoid complications, it is assumed that we are only dealing with packed databases, i.e. databases on which the operation `Pack` from the menu `Assist` has been invoked. This avoids having to deal with fragmentation in *Starling*'s variable length fields (which are discussed in Sec. 4).

3.1 Record header

Record structure information is included in the *Starling* database header, where each field is specified in a block of 32 bytes.

[1] `http://starling.rinet.ru/`

[2] `http://www.win.tue.nl/~aeb/`

[3] `http://www.dbf2002.com/dbf-file-format.html`

bytes	description
0–10	Name (padded with null characters)
11	Field type (ASCII)
	`C` character
	`Y` currency
	`N` numeric (can be a real number)
	`F` float
	`D` date
	`T` datetime
	`B` double
	`I` integer
	`L` logical
	`M` memo
	`G` general
	`P` picture
12–15	Offset of the field, within the record. (Do not trust this value.)
16	Field length
17	Number of decimal places
18	Field bit flags
	`0x01` system field (invisible to user)
	`0x02` nullable
	`0x04` binary (for field types *C* and *M* only)
19–31	Reserved bytes

3.2 Record contents

Each records start with a 'delete byte' with the following possible values.

`0x20`	normal record
`0x2A`	deleted record

Numeric fields have their value stored in text (each digit is represented by its ASCII encoded value). Furthermore, fields containing character data are padded with null characters ($0x00$).

4 References to associated `.var` file

A `.dbf` file is typically accompanied by a homonymous `.var` file. Due to the way records are specified in *dBase III*, a field (containing character data) has a maximum length of 255 characters. To accommodate longer texts, *Starling* databases treat a character field with a length of 6 bytes as a reference to a chunk of data, stored within the associated `.var` file.

bytes	description
0–3	data offset
4–5	data length

When a field that normally contains such a reference should instead be empty, its 6 bytes are all ASCII encoded spaces (`0x20`).

5 Character encodings

Stemming from a time before Unicode, *Starling* works with its own character encoding. Unfortunately, this encoding is not straightforward. Much work has been done on reverse engineering this encoding by prof.dr. Andries Brouwer and the following information relies heavily on the information compiled on his website.[4]

Characters are encoded in either one or two bytes. By default, upcoming data is to be read in 'single byte mode'. Byte flags are used to indicate when upcoming text is to be interpreted in 'double byte mode' or when to return to 'single byte mode'.

`0x01`	Enter 'double byte mode'
`0x7F`	Return to 'single byte mode'

There are, however, complications. When in double byte mode, encountering a byte in the ASCII range (below `0x7F`) also indicates a return to single byte mode (with the exception of the `0x01` flag for double byte mode). However, combining characters that are encoded in a single byte can seemingly occur in both single and double byte mode without affecting the mode.

5.1 Single byte mode

Among the byte values up to `0x1F` (the range for ASCII control characters) are those for switching byte mode (see above). In addition to those, the following bytes have their own semantics:

`0x09`	Tab character.
`0x0A`	Newline.
`0x0D`	Occasionally follows a `0x0A`, together forming one newline.
`0x15`	New paragraph.
`0x1D`	Marks the next byte as 'special' (see below).

The characters encoded by the remainder of the code points (`0x20`–`0xFF`) is shown in Table 1. The encoding is a customization of the Cyrillic code page CP866.

In case a byte is marked as special, by a preceding byte `0x1D`, the byte encodes the characters displayed in Table 2.

[4] `http://www.win.tue.nl/~aeb/natlang/charsets/starling-charset.html`

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
2_	0020	! 0021	” 0022	# 0023	$ 0024	% 0025	& 0026	’ 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3_	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4_	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5_	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	◌̂ 0302	_ 005F
6_	‘ 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7_	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	◌̃ 0303	
8_	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
9_	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
A_	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
B_	ā 0101	◌́ 0301	ä 00E4	◌̨ 0328	ā̈ 01DF	ç 0063 0323	č 010D	ç̌ 010D 0323	ō 03B4	ē 0113		◌̇ 0307	◌̈ 0308	ɛ 025B	ʢ 02A1	◌̰ 032F
C_	ç 00E7	ɣ 0263	ʁ 0281	ħ 0127	◌̄ 0304	ī 012B	ɨ 0268	ɨ̄ 0268 0304	◌̊ 030A	◌̥ 0325	ḳ 1E33	ʎ 028E	ƛ 019B	- 002D	ƛ̣ 019B 0323	ɬ 026B
D_	Ł 2C62	ŋ 014B	ō 014D	ö 00F6	◌̋ 022B	ɔ 0254	ɔ̄ 0254 0304	ṗ 1E57	q̇ 0071 0307	ß 00DF	~ 007E	◌̀ 0300	◌̣ 0323	š 0161	ṭ 1E6D	◌̌ 0306
E_	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F
F_	ϑ 03D1	ū 016B	ü 00FC	ṻ 01D6	ə 0259	ə̄ 0259 0304	◌̌ 030C	ʷ 02B7	ħ 0266	χ 03C7	ʒ 0292	ǯ 01EF	ž 017E	ʔ 0294	ʕ 0295	ʌ 028C

Table 1: The range of characters encoded by a single byte, each along with their Unicode code point(s). A red background denotes the ASCII code space, a green background denotes a character from the CP866 code page, and a blue background denotes characters from the custom *Starling* encoding. Red characters indicate combining characters.

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
4_		Æ 04D4			đ 0111								ɬ 026C			
5_					ŧ 0167										◌̋ 0311	
6_		æ 00E6	ƀ 0180	ɕ 0255	ð 00F0	œ 0153		ǧ 01E5	ḫ 1E2B	ɪ 0131	ɟ 0237					ø 00F8
7_			ɾ 027E	ʃ 0283	þ 00FE				ʒ 1D9A							
8_																
9_																
A_				ɠ 0260		ꞑ 0467		ꞩ 0455				ҙ 0459		ҥ 045A	ӂ 046B	
B_		◌̏ 030B														◌̯ 032E
C_																
D_												◌̓ 030F				
E_	ʁ 0281													є 0454		ž 017E
F_									ɦ 0195		ŋ 0452					

Table 2: Characters encoded by a ‘special’ byte in single byte mode.

5.2 Double byte mode

A sequence of two bytes encodes a character using the Chinese Big5 encoding,[5] of which the first byte always has a value greater than `0x7F`. *Starling* makes use of code points in the space reserved for user-defined characters (`0x81`–`0xA0`) for its own character encoding. The following provides an overview of *Starling*’s user defined characters.

1st byte	2nd byte	character
`0x83`	*(refer to Table 3)*	
`0x85`	`0xAF`	ʃ (U+03DD)
`0x87`	*(refer to Table 4)*	
`0x88`	`0x81`	ʏ (U+0475)
	`0x83`	ꞑ (U+0467)

[5] Cf. Microsoft’s code page 950.

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
9_	ά́ 0344	ἄ́ 0314 0301	ᾄ́ 0313 0301	ά́ 0301							ἂ 0308 0300	ἄ̀ 0314 0300	ᾄ̀ 0313 0300	ἆ́ 0314 0342	· 0387	
A_	ά̔ 0314	ἀ̔ 0313		ά̈ 0308	Α 0391	Β 0392	Χ 03A7	Δ 0394	Ε 0395	Φ 03A6	Γ 0393	Η 0397	Ι 0399	ῳ 1FF3	Κ 039A	Λ 039B
B_	Μ 039C	Ν 039D	Ο 039F	Π 03A0	Θ 0398	Ρ 03A1	Σ 03A3	Τ 03A4	Υ 03A5	ῃ 1FC3	Ω 03A9	Ξ 039E	Ψ 03A8	Ζ 0396		
C_	ἆ́ 0313 0342	ἂ 0300	α 03B1	β 03B2	χ 03C7	δ 03B4	ε 03B5	φ 03C6	γ 03B3	η 03B7	ι 03B9	ς 03C2	κ 03BA	λ 03BB	μ 03BC	ν 03BD
D_	ο 03BF	π 03C0	θ 03B8	ρ 03C1	σ 03C3	τ 03C4	υ 03C5	ᾳ 1FB3	ω 03C9	ξ 03BE	ψ 03C8	ζ 03B6	ᾶ 0342			

Table 3: The Greek characters encoded by the second byte in double byte mode, when the first byte is `0x83`. The 'α' that is present with the combining characters (shown in red) is merely there for reference.

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
8_				ґ̂ 0433 0311	б 0431		ю 044E									
9_				ж 0436			Ю 042E				И 0418	С 0421			А 0410	П 041F
A_	Р 0420		О 041E	Л 041B	Д 0414				З 0417		К 041A		Е 0415	Г 0413	М 041C	
B_		Н 041D		х 0445					ъ 044A	ф 0444	и 0438	с 0441	в 0432	у 0443	а 0430	п 043F
C_	р 0440	ш 0448	о 043E	л 043B	д 0434	ь 044C	т 0442		з 0437		к 043A	ы 044B	е 0435	г 0433	м 043C	ц 0446
D_	ч 0447	н 043D	я 044F	Х 0425		Ъ 0463			ꙗ A657		ѧ 0467					
E_								ӂ 046B	Е 0415	е 0065						
F_			ѡ 0461	Ҍ 046C	ҍ 046D	Ҿ 0464	ҿ 0465									

Table 4: The Cyrillic characters encoded by the second byte in double byte mode, when the first byte is `0x87`.

6 Layout codes

Starling allows character data to be formatted. For this, it uses layout tags that are inserted into the character data. These tags are the following:

tag	formatting style
`\B`...`\b`	bold text
`\I`...`\i`	italic text
`\U`...`\u`	underline text
`\H`...`\h`	superscript text
`\L`...`\l`	subscript text
`\C`...`\c`	condensed text

In addition to these layout tags, there are also tags that produce links. The exact format of these links is somewhat unclear, but seems to be as follows:

tag	formatting style
`\X0`...`\x`	Link to an entry containing the tagged text. The text itself also makes up the clickable link. (Perhaps the number 0 is variable and refers to the column number.)
`\X<0.`n`>`...`\x`	The clickable link is again the text between the tags, but the target is now entry n. (Perhaps the number 0 is variable and is an index to a list of different databases.)

Starling does not enforce the contents of a cell to have correctly nested tags. Even closing tags without a preceding opening tag can occur.

7 Auxiliary files

A *Starling* database file (`.dbf`) can be accompanied by multiple homonymous files with a different extension. One of these is the `.var` file, already mentioned in Sec. 4, but beside that one, there can also be an `.inf` file and/or a `.prt` file.

7.1 `.inf` file

This is a file containing metadata for the database, such as multilingual aliases for the field names (because they can only be at most 10 characters long) or additional information describing the contents of the database. The file is encoded in the same *Starling* encoding as set forth in Sec. 5.

7.2 `.prt` file

This file contains settings for the *Starling* print functionality, such as column delineators etc. It is itself a *Starling* database.