File Structure of Alphadict Dictionary
Version 1

Introduce

This file specifies the structure of dict of AlphaDict -- a cross-platform open source software.
Dict File is a binary file with a UTF-8 coding including three areas --Header, Index, and Data. You can make your dictionary easily by using front-end format based on xml (more detail see “front-end_format.xml”) , edit the xml using any editor and convert it to a dictionary of Alphadict by a command tool “AlConvert”.

Index is a tree-like structure, called Index Tree.
The Running Time for lookup is O(lgn1+lgn2+..+lgn(L)).

L: Length of the word/phrase.

n: The number of nodes for every level. Usually (n1 > n2 > .. > n(L)).

In November 2013, I wrote the first draft version 1.

History

version	revise	author	date
1		LiQiong Lee <LiQiong.karotrz.lee@gmail.com>	Nov-22-2013

1 Definitions and Convention

For the purposes of this standard, the following definitions and conventions apply.

1.1 Numerical values

A numerical value shall be represented in binary notation by 8/16/32/64-bit fields.
The byte order of multi-byte numerical shall be ***little-endian(***Least significant byte first).

For example, the decimal number 30600 has (77 88) as its hexadecimal represention and is recorded as (88 77).

1.2 NULL

A null number value shall be represented in binary notation by all-bit being one in a field . It is const. The NULL of shall be 0xff for 8-bit field, 0xffff for 16-bit, oxffffffff for 32-bit.

1.3 W-Char values

A variation of numerical value as a 32-bit number representing a character by its unicode value.

1.4 Date values

A variation of numerical value as a 32-bit number, representing a date.
The format shall be satisfied as follows.

BP1	BP2	BP3 to BP4
The Number of Day	The Number of Month	The Number of Year

For example, the hexadecimal numbers (19 0B 62 13) represents “Nov 25, 1998”.

1.5 Character sets and coding

The characters and u8string shall be coded according to UTF-8.
In each fixed-length filed, the characters shall be left-justified and remaining byte positions on the right shall be set to zero(0).
u8string shall include a the terminating null byte ('\0') at the end,

1.6 Logical Block

A group of 256 bytes treated as a logical unit. Each Logical Block shall be identified by a unique Logical Block Number. Logical Block Numbers shall be integers assigned in ascending order starting with 1.
The Size of Logical Block should align the sector's size of filesystem, Make driver record a “significant-data” whithin a or muti sector(s) as possible. 256 is a conservative number.

2 Notation

BP Byte position within a descriptor, starting with 1.

3 Header Descriptor

The Header Descriptors shall occupy the first Logical block, and shall specify the layout of dictionary file structural, shall identify the dictionary, the version of the dictionary, the author , the source and destination language and some other meta informations.

3.1 Format of a Header Descriptors

BP	Field name	Content
1 to 2	Magic Word	0x77 0x88
3	Header Version	numerical value
4 to 7	Publish Date	date value
8 to 67	Publisher Identifier	u8string
68 to 69	Dict Version	numerical value
70 to 129	Dict Identifier	u8string
130 to 133	Entries	numerical value
134	Location of Index Character Area	numerical value
135 to 138	Location of Index String Area	numerical value
139 to 142	Location of Data Area	numerical value
143 to 157	Source Language	u8string
158 to 172	Destination Language	u8string
173	Flags	numerical value

3.1.1 Publish Date (BP 4 to 7)

This field shall specify the published date of the dictionary.

3.1.2 Publisher Identifier(BP 8 to 67)

This field shall specify an identification of the dictionary owner or maker.

3.1.3 Dict Version (BP 68 to 69)

This field shall specify as an 16-bit number an identification of the version of the dictionary.

BP 68	BP 69
Major Version Number	Minor Version Number

3.1.4 Dict Identifier (BP 70 to 129)

This field shall specify an identification for the dictionary.

3.1.5 Entries (BP 130 to 133)

This field shall specify as a 32-bit number how many items in the Data Area.

3.1.6 Location of Index Character Area (BP 134)

This field shall specify as a 8-bit number the Logical Block Number of the First Logical Block allocated to Index Area. if non-existent, shall be NULL.

3.1.7 Location of Index String Area (BP 135 to 138)

This field shall specify as a 32-bit number the Logical Block Number of the First Logical Block allocated to Index String Area. if non-existent, shall be NULL.

3.1.8 Location of Data Area (BP 139 to 142)

This field shall specify as a 32-bit number the Logical Block Number of the First Logical Block allocated to Data Area, which contains data items.

3.1.9 Source Language and Destination Language (BP 143 to 157)

This two fields shall specify which language(Source Language) this dictionary should translate to which language(Destination Language).

3.1.10 Number of Index Character (BP 158 to 172)

This field shall specify as a 32-bit number how many index characters in Index Character Area.

3.1.11 Number of Index Character (BP 173)

This field shall specify as a 8-bit number flag of special feature.

7 .. 1	0
reserve	duplicate index

duplicate index : If there are duplicate indexes – – it is a unpleasant feature.
for example, two items : boy: explantion1; boy: explantion2
if 'boy' was looked up, shall show “explantion1 + explantion2”.

3.2 The Layout of Dictionary File Structure:

Header Descriptor	Index Character Area	Index String Area	Data Area
BLOCK 1	BLOCK 2 to n	BLOCK n to m	BLOCK m to k

4 Index Area

The first Logical Block of Index Area shall follow the Logical Block of Header Descriptor and it shall be specified in Header Descriptor too. There are two kinds of Index, one is Index Character, the other one is Index String.

4.1 Index Character Area

It is a tree structure as follows.

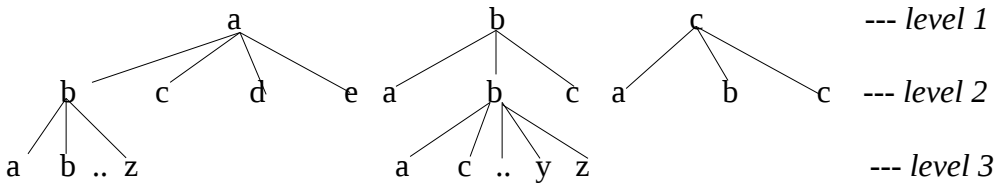

Figure 4-1

A Index, which also is a prefix, consists of the nest Index Characters among different level along the tree. The Content of a Index is all the words or phrases beginning with that .If a Index is only a word or phrase, not a prefix for any word, the content is meaningless.

If a Index is a phrase, should use SPACE character as a concat character connecting words.

The data structure of a Index Character calls Index Item which shall be a fix-length record. The Index Items shall be recorded at together.

4.1.1 Format of Index Item

BP	Field name	Content
1 to 4	Index Character	w-char
5 to 8	Location	numerical
9 to 10	Length of Content	numerical

4.1.2 Index Character (BP 1 to 4)

The interpretation of this field depends as follows on whether Index is a word/phrase or not.

If Index Character set to 0, it shall mean this Index(except this 0 Index Character) is a word/phrase :

- The Location Field shall specify the address allocated to this word in Data Area.
- The Length of Content shall set to 0.

otherwise, the Index Character field shall set as a 32-bit number to the utf-32(ucs) value of a descending index character.

If a Index are both 'Index' and 'Word', shall put a 0 Index Character in its content area. If there are duplicate Indexes, shall use 0 Index Characters too.

4.1.3 Location (BP 5 to 8)

This field shall specify as a 32-bit number address in bytes allocated to the Index. Address should be the relative address (offset to base address of every area).

The interpretation of this field depends as follows on whether Index is a word/phrase or not.
If this Index Item is only a word/phrase(means it doesn't contain any Index Item), it shall mean:

- The Location Field shall specify the address allocated to this word in Data Area.
- The Length of Content shall set to 0.

otherwise, it shall mean:

- The Location Field shall specify the address allocated to the content of this index within either Index Character Area or Index String Area.
- The Length of Content shall specify the number of Index Item within the Index's content.
- The Location Field should conform the following format.

31	30 0
Flag	Address

Flag has two values “0” and “1”.

0 : specifying the address in Index Character Area.

1 : specifying the address in Index String Area.

4.1.4 Length of Content (BP 9 to 10)

This field shall specify as a 16-bit number the number of Index Items within the Index's Content. If non-existent Content(for hash item), shall be NULL.

4.2 Index String Area

A Index String is the rest of target string in dictionary. The data structure of a Index String calls Index Item which shall be a varied-length record. Index Item are recorded within a BLOCK.

4.2.1 Format of Index Item

BP	Field name	Content
1 to 4	Location in Data Area	numerical
5	The Length of String (LEN_S)	numerical
5 to (4+LEN_S)	Index String	characters

4.2.4 Location in Data Area (BP 1 to 4)

This field shall specify as a 32-bit number the address in bytes allocated to the Index String in Data Area.

4.2.2 The Length of String (BP 5 to 6)

This field shall specify as a 8-bit number the length of Index String.
If this field sets to 0, it shall mean the Index in Index Character Area is a word/phrase.

4.2.3 Index String [BP 5 to (5+LEN_S)]

This field shall specify as utf-8 string the rest of word or phrase in dictionary.
A item should be within a block.

4.3 Example for Index Area Layout

Here is a example: There are four words a English dictionary : “abb” at 1024, “abba” at 1124, “alpha” at 1224, “daa” at 1324, “dict” at 1424, a two duplicate “daa”s at 1524 and 1624.

Index Character Area (Block 2) :

address	Index Item (12 bytes)
0	0061(a) 14 2 0064(d) 46 2
14	0062(b) 28 1 006C(l) 3C 1
28	0062(b) 32 2
32	0 1024 0 0061(a) 1124 0
3C	0070(p) 0x10000000 1
46	0061(a) 5A 1 0079(i) 0x10000007 1
5A	0061(a) 6E 3
6E	0 1324 0 0 1524 0 0 1624 0

Index String Area (Block 3) :

Address	Index Item
0	1224 2 ha
7	1424 2 ct

Data Area (Block 4):

5 Data Area

Data Area shall follow the Index Area, Its location is specified in Header Descriptor. It contains all contents of words. Data Item is a unit to record a word's content, which is a varied-length record. To make things simple, record both length of a string and string which don't have a '\0' at the end.

5.1 Format of Data Item

Field name	Content
The Length of Word/Phrase (LEN_S)	8-bit numerical
Word/Phrase String	characters
The Length of Phonetic Characters (LEN_P)	8-bit numerical
Phonetic Characters	characters
The Length of Explanation(LEN_P)	16-bit numerical
Explanation	characters