

**NAME**

trietool - trie manipulation tool

**SYNOPSIS**

**trietool** [ *options* ] *trie command arg ...*

**DESCRIPTION**

**trietool** is the command-line tool for manipulating double-array trie data. It can be used to query, add and remove words in a trie.

**The Trie**

The *trie* argument specifies the name of the trie to manipulate. A trie is stored in a file with ‘.tri’ extension. However, to create a new trie, one needs to prepare a file with ‘.abm’ extension, describing the Unicode ranges of alphabet set of the trie. The ABM defines a set of vectors that map Unicode characters into a continuous range of integers. The mapped integers will be used as internal alphabet for the trie. Such mapping can improve the space allocation within the trie data, regardless of non-continuity of the character set being used, as the mapped range is always continuous.

The ABM file is a plain text file, with each line listing a range of 32-bit Unicodes to be added to the alphabet set, in the format:

```
[0xSSSS,0TTTT]
```

where ‘0xSSSS’ and ‘0TTTT’ are hexadecimal values of starting and ending character code for the range, respectively.

For example, for a dictionary that contains only English words without any punctuations, one may prepare ‘*trie.abm*’ as:

```
[0x0041,0x005a]  
[0x0061,0x007a]
```

The first line lists the ASCII codes for A-Z, and the second for a-z.

No more than 255 alphabets are allowed in a trie.

The created ‘.tri’ file will incorporate the ABM data. So, the ‘.abm’ file is not required after the first creation, and will be ignored.

**COMMANDS**

Available commands are:

**add** *word data* ...

Add *word* to trie, associated with integer *data*. Arbitrary number of words-data pairs can be given. Two arguments will be read at a time, the first will be treated as *word*, and the second as *data*.

**add-list** [ *options* ] *list-file*

Add words with associated data listed in *list-file* to trie. The *list-file* must be a text file listing one word per line. The associated data can be put after the word in the same line, separated with tab ('t') character. If the data field is omitted, a default value (-1) will be used instead.

*Options* are available for this command:

**-e, --encoding** *enc*

Specify character encoding of the *list-file* contents, such as 'UTF-8'. If omitted, current locale codeset is assumed.

**delete** *word* ...

Delete *word* from trie. Arbitrary number of words to delete can be given.

**delete-list** [ *options* ] *list-file*

Delete words listed in *list-file* from trie. The *list-file* must be a text file listing one word per line.

*Options* are available for this command:

**-e, --encoding** *enc*

Specify character encoding of the *list-file* contents, such as 'UTF-8'. If omitted, current locale codeset is assumed.

**query** *word*

Search for *word* in trie. If *word* exists, its associated data is printed to standard output. Otherwise, error message is printed to standard error, with nothing printed to standard output.

**list** List all words in trie to standard output. The output lists one word-data pair per line, separated with tab ('t') character, the format appropriate for being *list-file* for the **add-list** command.

## OPTIONS

This program follows the usual GNU command line syntax, with long options starting with two dashes ('--'). A summary of options is included below.

**-p, --path *dir***

Set trie directory to *dir* [default='.']

**-h, --help**

Show summary of options.

**-V, --version**

Show version of program.

**AUTHOR**

libdatrie was written by Theppitak Karoonboonyanan.

This manual page was written by Theppitak Karoonboonyanan <theppitak@gmail.com>.