

Namelur

Written by admin

Sunday, 28 May 2006

Namelur version 0.2

[Download version 0.2](#)

[SourceForge.net Namelur Project Home Page](#)

Random name generator library with rule files. Fast, lightweight and easy. Can generate rule files from text, you can easily use it into your application and use one of the attached rule files for rpg/fantasy/wars/etc. name generation.

Licensed under [GPL](#)

1. API documentation

How to convert a text resource into a NAR rule set

```
void nr_txt2nar(RWops *sourcefile, RWops *destination_nar);
```

If 'destination_nar' is NULL only the internal arrays are updated and the library will be in any case ready for name generation. You should NOT generate names using nr_txt2nar on the same source file. If the source file is the same, save a .NAR file and use that with the below function.

Howt to load a NAR rule file

```
// like in fopen("rulname.nar", "rb"), which is not allowed because the file needs to be cached through  
rulefile = RWFromFile("rulname.nar", "rb");  
nr_load(rulefile);
```

After this the needed rules for the name generation will have been recognized.

How to generate a single name

```
void nr_name(char *genname, int minsyl, int maxsyl);
```

minsyl is the minimum number of syllables and maxsyl the maximum number of syllables of the generated name.

Note: genname must be a char * array at least big maxsyl*MAX_SYLLABE_LEN+1 characters!

2. Features

3. Rulname command line tool

The rulname command line tool can be used to strip text from any file and to generate the proper NAR file. It can also be used to generate random names based on a NAR rule file.

rulname [-h] [-o] srcfile [destination.nar]
 rulname [-h] [-o] [-rnarfile] [-nnames] [-imins] [-amaxs] [-sseed] [-t]
 -h Show this syntax help page
 -o Show only the output without any message
 srcfile Input file. All the valid syllables within it will be read
 and used to generate a NAR rule file
 destination.nar Destination NAR file. If not specified NAR file output
 will be sent to stdout (unless maxnames is specified)
 -rnarfile Source NAR file. If not specified and neither 'srcfile'
 this help text will be used as 'srcfile' content
 -nnames Output 'names' number of random names to stdout
 -imins -amaxs Minimum/maximum number of syllables
 -sseed Specify a seed number to feed the internal random generator
 If not specified the current time will be used as seed
 -t Separate each name through a tab instead of a newline

Note: rulname has 2 behaviours:

* when you specify a source file, it will produce the NAR file into the
'destination.nar', the standard output or none (if maxnames is positive)

* when you specify a NAR file (or a source file + maxnames) random names will be generated

* when no parameter is specified 10 random names will be generated using the internal text found in the
program itself

4. Rule files (*.NAR)

Each rule file contains plain ASCII text, lines that cannot be interpreted are skipped. Comments should
be entered using #

Example of comment

v12 a 6 e 7 i 7 o7 u 4

c3 b 7 c 7 d 7 f 7 g 7 h 7

The first part of each valid line must contain 'v' or 'c' (vowel or consonant) followed by the number of
times (e.g. the density).

Must follow pairs of a string and a rule code. Each string cannot be longer than 3 characters, following
rule codes for the consonant/vowel:

1 can be at end of word (EOW)

2 can be at beginning of word (BOW)

4 can be in middle of word (MOW)

Since the rule code works as a binary encoding (middle) (beginning) (end), we also have (as combination of
previous codes):

5 can be in middle or at ending of word (MEW)

6 can be at beginning or ending (BEW)

7 can be anywhere in word (AIW)

No particular error checking is performed, so you should void to:

In both lists, duplicates increase the likelihood of the occurrence.
To have "no vowel/consonant at the end", simply use two spaces.

5. Working scheme of rulname (from text to NAR files)

When you use rulname or the namelur lib itself to parse a file, it will:

- isolate a word
- split it into syllables
- save each syllable into the array without making duplicates but using a reference count and the relative position in the word
- iterate till there are more words

This will let you produce random names of the same style of the source text. I suggest you to use a v text file as basis, convert it to a NAR file and use that.

6. Using Namelur in your application

7. Disclaimer

I guess that there may be bugs. The worst thing that can happen is a bad crash or an infinite loop, any the license for the complete disclaimer notice.

8. History

The idea was given by [David A. Wheeler's Totro](#) a javascript name generator. First I mocked up a C conversion, then I added more features coming up with the current Namelur.

I let it run to choose a good name for itself and after 3-4 runs I finally got its nice name, Namelur!

9. Contact

I would be happy to answer your feedback on [project Forum](#)

Please help me debugging!

Also, I have not put a great effort on porting and some minor changes are needed since [RNewslib](#) is already portable.

10. Subversion control (SVN)

The online Subversion source code repository is available at:

svn.sourceforge.net/viewcvs.cgi/namelur/

If you have a subversion client you can better explore the repository through:

<svn://svn.sourceforge.net/viewcvs.cgi/telbless/>

Last updated (Monday, 01 October 2007)