Onegin Sorter

Generated by Doxygen 1.8.14

Contents

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

document_t	**2**

2 File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

include/document.h	**2**
include/unicode.h	**4**
include/unicode_tables.h	**??**

3 Data Structure Documentation

3.1 document_t Struct Reference

```
#include <document.h>
```

Data Fields

- char ∗ data

  Raw data from file.
- int data_size

  Raw data size; used for internal purposes.
- line_t ∗ lines

  Array of lines of text.
- int lines_cnt

  Length of `lines` array.

3.1.1 Detailed Description

Text file separated into lines by '\n'` symbol

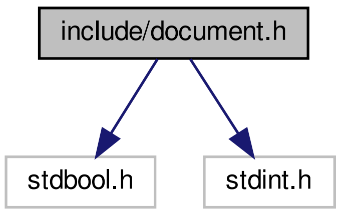
The documentation for this struct was generated from the following file:

- include/document.h

4 File Documentation

4.1 include/document.h File Reference

```
#include <stdbool.h>
#include <stdint.h>
```
Include dependency graph for document.h:



Data Structures

- struct document_t

Typedefs

- typedef char ∗ **line_t**

Functions

- document_t ∗ read_document (const char ∗filename)
- bool check_document (const document_t ∗document, int ∗err_pos)
- bool print_document (const document_t ∗document, const char ∗filename)
- void close_document (document_t ∗document)
- int32_t symbol_at (const document_t ∗document, int pos)

4.1.1 Function Documentation

4.1.1.1 check_document()

```
bool check_document (
             const document_t ∗ document,
             int ∗ err_pos )
```

Check if the document is correct utf-8 file

Parameters

out	*err_pos*	If file is malformed, position of the first illegal byte is written here

4.1.1.2 close_document()

```
void close_document (
             document_t ∗ document )
```

Free al structures related to the document

4.1.1.3 print_document()

```
bool print_document (
             const document_t ∗ document,
             const char ∗ filename )
```

Write lines of the document to the file `filename`

Returns

`true` if succeded and `false` otherwise

4.1.1.4 read_document()

```
document_t* read_document (
             const char * filename )
```

Open file `filename` and read it

Returns

pointer to the document if no errors occured or NULL otherwise

4.1.1.5 symbol_at()

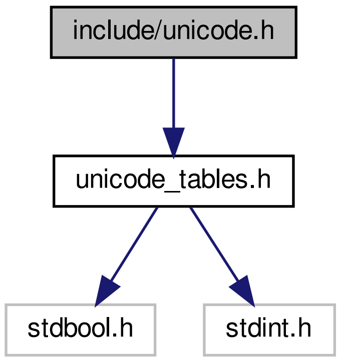
```
int32_t symbol_at (
             const document_t * document,
             int pos )
```

Get symbol by its position in original file

4.2 include/unicode.h File Reference

```
#include "unicode_tables.h"
```

Include dependency graph for unicode.h:



Functions

- int32_t next_symbol (const char **pos)
- int unicode_lex_cmp (const void *str_a, const void *str_b)
- int unicode_rev_lex_cmp (const void *str_a, const void *str_b)

4.2.1 Function Documentation

4.2.1.1 next_symbol()

```
int32_t next_symbol (
            const char ** pos )
```

Read next utf8-encoded symbol from char buffer and move the pointer

Parameters

`in,out`	*pos*	Is changed if there is a valid encoded symbol

Returns

code of symbol or -1 if it is invalid

4.2.1.2 unicode_lex_cmp()

```
int unicode_lex_cmp (
            const void * str_a,
            const void * str_b )
```

Compare two utf8-encoded strings ignoring all symbols except letters and case-insensitively and return negative, zero or positive value if first string is less, equal or greater than second string in terms of lexicographical order.

4.2.1.3 unicode_rev_lex_cmp()

```
int unicode_rev_lex_cmp (
            const void * str_a,
            const void * str_b )
```

Compare two utf8-encoded strings ignoring all symbols except letters and case-insensitively and return negative, zero or positive value if first string is less, equal or greater than second string in terms of reversed lexicographical order.

Index