

libunifloat
1.0.0

Generated by Doxygen 1.8.9.1

Mon May 22 2017 10:32:37

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	1
2.1	File List	1
3	Data Structure Documentation	2
3.1	CString Struct Reference	2
3.1.1	Detailed Description	2
3.2	Unifloat Struct Reference	2
3.2.1	Detailed Description	3
3.2.2	Field Documentation	3
3.3	UnifloatComplex Struct Reference	3
3.3.1	Detailed Description	3
4	File Documentation	3
4.1	bessel.h File Reference	3
4.1.1	Detailed Description	4
4.2	cexp.h File Reference	4
4.2.1	Detailed Description	4
4.3	constants.h File Reference	4
4.3.1	Detailed Description	5
4.4	cstring.h File Reference	5
4.4.1	Detailed Description	5
4.5	ctrig.h File Reference	5
4.5.1	Detailed Description	6
4.5.2	Function Documentation	6
4.6	debug.h File Reference	7
4.6.1	Detailed Description	7
4.7	exp.h File Reference	7
4.7.1	Detailed Description	7
4.8	gamma.h File Reference	7
4.8.1	Detailed Description	8
4.9	libunifloat.h File Reference	8
4.9.1	Detailed Description	8
4.10	trig.h File Reference	8
4.10.1	Detailed Description	9
4.10.2	Function Documentation	9
4.11	unifloat.h File Reference	10

4.11.1 Detailed Description	14
4.11.2 Typedef Documentation	14
4.11.3 Enumeration Type Documentation	14
4.11.4 Function Documentation	14
4.12 unifloat_complex.h File Reference	20
4.12.1 Detailed Description	20
4.12.2 Typedef Documentation	21
4.12.3 Function Documentation	21
Index	23

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

CString	
The structure that represents a string in libunifloat	2
Unifloat	
The main structure that represents a number in libunifloat	2
UnifloatComplex	
Represents a complex number in libunifloat	3

2 File Index

2.1 File List

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

bessel.h	
Bessel functions	3
cexp.h	
Complex exponential functions	4
constants.h	
Collection of constants used for internal calculations	4
cstring.h	
Auxiliary structure and functions to work with text	5
ctrig.h	
Complex trigonometric functions	5
debug.h	
Auxiliary module to help programmer to debug the library	7
exp.h	
Real exponential functions	7

gamma.h	
Gamma functions	7
libunifloat.h	
Includes all the other header files in the project	8
trig.h	
Real trigonometric functions	8
unifloat.h	
Basic Unifloat operations	10
unifloat_complex.h	
Basic UnifloatComplex operations	20

3 Data Structure Documentation

3.1 CString Struct Reference

The structure that represents a string in libunifloat.

```
#include <cstring.h>
```

Data Fields

- int **len**
- char * **buf**

3.1.1 Detailed Description

The structure that represents a string in libunifloat.

Definition at line 31 of file `cstring.h`.

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

- [cstring.h](#)

3.2 Unifloat Struct Reference

The main structure that represents a number in libunifloat.

```
#include <unifloat.h>
```

Data Fields

- int [sign](#)
The sign of [Unifloat](#) is the same as of this field.
- int [exp](#)
The power of two.
- uint [mant](#) [[MAX_SIZE_UNIFLOAT](#)]
Mantissa of the number.
- [UnifloatKind](#) [kind](#)
Kind of floating point number (Normal, Infinity, NaN).

3.2.1 Detailed Description

The main structure that represents a number in libunifloat.

Definition at line 67 of file unifloat.h.

3.2.2 Field Documentation

3.2.2.1 uint Unifloat::mant[**MAX_SIZE_UNIFLOAT**]

Mantissa of the number.

Not all bits are used. See **PRECISION**

Definition at line 76 of file unifloat.h.

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

- [unifloat.h](#)

3.3 UnifloatComplex Struct Reference

Represents a complex number in libunifloat.

```
#include <unifloat_complex.h>
```

Data Fields

- [Unifloat * Re](#)
The real part of complex number.
- [Unifloat * Im](#)
The imaginary part of complex number.

3.3.1 Detailed Description

Represents a complex number in libunifloat.

The main structure for complex calculations.

Definition at line 34 of file unifloat_complex.h.

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

- [unifloat_complex.h](#)

4 File Documentation

4.1 `bessel.h` File Reference

Bessel functions.

```
#include "unifloat/unifloat.h"
```

Functions

- `Unifloat * j0_UF (Unifloat *x)`
- `Unifloat * j1_UF (Unifloat *x)`
- `Unifloat * jn_UF (int n, Unifloat *x)`
- `Unifloat * jnPowerSeries_UF (int n, Unifloat *x)`
- `Unifloat * jnHankel_UF (int n, Unifloat *x)`
- `Unifloat * jnSteed_UF (int n, Unifloat *x)`
- `Unifloat * jnMeisselFirst_UF (int n, Unifloat *x)`
- `Unifloat * jnMeisselSecond_UF (int n, Unifloat *x)`
- `Unifloat * jnRecurrent_UF (int n, Unifloat *x)`
- `Unifloat * y0_UF (Unifloat *x)`
- `Unifloat * y1_UF (Unifloat *x)`
- `Unifloat * yn_UF (int n, Unifloat *x)`
- `Unifloat * ynPowerSeries_UF (int n, Unifloat *x)`
- `Unifloat * ynHankel_UF (int n, Unifloat *x)`
- `Unifloat * ynSteed_UF (int n, Unifloat *x)`
- `Unifloat * ynMeisselFirst_UF (int n, Unifloat *x)`
- `Unifloat * ynMeisselSecond_UF (int n, Unifloat *x)`
- `Unifloat * ynRecurrent_UF (int n, Unifloat *x)`
- `int bessellMethod_UF (int n, Unifloat *x)`

4.1.1 Detailed Description

Bessel functions.

4.2 cexp.h File Reference

Complex exponential functions.

```
#include "unifloat/unifloat_complex.h"
#include "unifloat/exp.h"
```

Functions

- `UnifloatComplex * cexp_UF (UnifloatComplex *x)`
- `UnifloatComplex * clog_UF (UnifloatComplex *x)`
- `UnifloatComplex * clog10_UF (UnifloatComplex *x)`
- `UnifloatComplex * cpow_UF (UnifloatComplex *x, UnifloatComplex *y)`
- `UnifloatComplex * csqrt_UF (UnifloatComplex *x)`

4.2.1 Detailed Description

Complex exponential functions.

4.3 constants.h File Reference

Collection of constants used for internal calculations.

Macros

- ```

• #define PI "11.00100100001111110110101010001000100001011010001100001000110\1001100010011000110011000101001
• #define GAMMA "0.5772156649015328606065120900824024310421593359399235988057672\34884867726777664670936
• #define _E "2.71828182845904523536028747135266249775724709369995957496696762\772407663035354759457138217
• #define EPSILON "0.00000000000000000000000000000001"
• #define LN_2 "0.101100010111001000010111111011110100011100111101111001101010\1111001001111000111011001
• #define LOG2_E "1.0111000101010100011101100101001010111000001011111100001011101\1101111101000011111111
• #define LOG10_E "0.01101111001011011110110001010100100110111001010000111000110010\1010011010101010110110
• #define LOG2_10 "11.01010010011010011110000100101110011010001101110001010111111\1001001001001010111111

```

#### 4.3.1 Detailed Description

### Collection of constants used for internal calculations.

## 4.4 cstring.h File Reference

### Auxiliary structure and functions to work with text.

```
#include "unifloat/config.h"
```

## Data Structures

- struct CString

*The structure that represents a string in libunifloat.*

## Typedefs

- typedef struct CString CString

*The structure that represents a string in libunifloat.*

## Functions

- void **delete\_CString** (CString \*cstr)
- CString \* **create\_CString** (const char \*buf)
- CString \* **clone\_CString** (CString \*cstr)
- int **length\_CString** (CString \*cstr)
- char **charAt\_CString** (CString \*cstr, int i)
- CString \* **substring\_CString** (CString \*src, int start, int len)
- int **indexOfChar\_CString** (CString \*cstr, char c)

#### 4.4.1 Detailed Description

### Auxiliary structure and functions to work with text.

## 4.5 ctrig.h File Reference

Complex trigonometric functions.

```
#include "unifloat/unifloat_complex.h"
#include "unifloat/trig.h"
```

## Functions

- `UnifloatComplex * cacosh_UF (UnifloatComplex *x)`  
The complex arc hyperbolic cosine of the given `Unifloat` number.
- `UnifloatComplex * casinh_UF (UnifloatComplex *x)`  
The complex arc hyperbolic sine of the given `Unifloat` number.
- `UnifloatComplex * catanh_UF (UnifloatComplex *x)`  
The complex arc hyperbolic sine of the given `Unifloat` number.
- `UnifloatComplex * ccosh_UF (UnifloatComplex *x)`  
The complex hyperbolic cosine of the given `Unifloat` number.
- `UnifloatComplex * csinh_UF (UnifloatComplex *x)`  
The complex hyperbolic sine of the given `Unifloat` number.
- `UnifloatComplex * ctanh_UF (UnifloatComplex *x)`  
The complex hyperbolic tangent of the given `Unifloat` number.

### 4.5.1 Detailed Description

Complex trigonometric functions.

### 4.5.2 Function Documentation

#### 4.5.2.1 `UnifloatComplex* cacosh_UF ( UnifloatComplex * x )`

The complex arc hyperbolic cosine of the given `Unifloat` number.

#### Returns

The complex arc hyperbolic cosine of `x`.

The function creates and returns the object that have to be removed later using `delete_UF`.

#### 4.5.2.2 `UnifloatComplex* casinh_UF ( UnifloatComplex * x )`

The complex arc hyperbolic sine of the given `Unifloat` number.

#### Returns

The complex arc hyperbolic sine of `x`.

The function creates and returns the object that have to be removed later using `delete_UF`.

#### 4.5.2.3 `UnifloatComplex* catanh_UF ( UnifloatComplex * x )`

The complex arc hyperbolic sine of the given `Unifloat` number.

#### Returns

The complex arc hyperbolic sine of `x`.

The function creates and returns the object that have to be removed later using `delete_UF`.

#### 4.5.2.4 `UnifloatComplex* ccosh_UF ( UnifloatComplex * x )`

The complex hyperbolic cosine of the given `Unifloat` number.

#### Returns

The complex hyperbolic cosine of `x`.

The function creates and returns the object that have to be removed later using `delete_UF`.



#### 4.5.2.5 UnifloatComplex\* csinh\_UF ( UnifloatComplex \* x )

The complex hyperbolic sine of the given Unifloat number.

##### Returns

The complex hyperbolic sine of **x**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.5.2.6 UnifloatComplex\* ctanh\_UF ( UnifloatComplex \* x )

The complex hyperbolic tangent of the given Unifloat number.

##### Returns

The complex hyperbolic tangent of **x**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

## 4.6 debug.h File Reference

Auxiliary module to help programmer to debug the library.

```
#include "unifloat/config.h"
```

### 4.6.1 Detailed Description

Auxiliary module to help programmer to debug the library.

## 4.7 exp.h File Reference

Real exponential functions.

```
#include "unifloat/unifloat.h"
```

### Functions

- Unifloat \* **exp\_UF** (Unifloat \*x)
- Unifloat \* **exp2\_UF** (Unifloat \*x)
- Unifloat \* **expm1\_UF** (Unifloat \*x)
- Unifloat \* **log\_UF** (Unifloat \*x)
- Unifloat \* **log1p\_UF** (Unifloat \*x)
- Unifloat \* **sqrt\_UF** (Unifloat \*x)

### 4.7.1 Detailed Description

Real exponential functions.

## 4.8 gamma.h File Reference

Gamma functions.

```
#include "unifloat/unifloat.h"
```

## Functions

- `Unifloat * gamma_UF (Unifloat *x)`
- `Unifloat * lgamma_UF (Unifloat *x)`
- `Unifloat * tgamma_UF (Unifloat *x)`
- `Unifloat * gammaSeries_UF (Unifloat *x)`

## Variables

- `int signgam`

### 4.8.1 Detailed Description

Gamma functions.

## 4.9 libunifloat.h File Reference

Includes all the other header files in the project.

```
#include "unifloat/unifloat.h"
#include "unifloat/unifloat_complex.h"
#include "unifloat/cstring.h"
#include "unifloat/trig.h"
#include "unifloat/ctrig.h"
#include "unifloat/exp.h"
#include "unifloat/cexp.h"
#include "unifloat/gamma.h"
#include "unifloat/bessel.h"
```

### 4.9.1 Detailed Description

Includes all the other header files in the project.

## 4.10 trig.h File Reference

Real trigonometric functions.

```
#include "unifloat/unifloat.h"
```

## Functions

- `Unifloat * sin_UF (Unifloat *x)`  
*The sine of the given `Unifloat` number.*
- `Unifloat * cos_UF (Unifloat *x)`  
*The cosine of the given `Unifloat` number.*
- `Unifloat * tan_UF (Unifloat *x)`  
*The tangent of the given `Unifloat` number.*
- `Unifloat * asin_UF (Unifloat *x)`  
*The arc sine of the given `Unifloat` number.*
- `Unifloat * acos_UF (Unifloat *x)`  
*The arc cosine of the given `Unifloat` number.*

- `Unifloat * atan_UF (Unifloat *x)`  
*The arc tangent of the given `Unifloat` number.*
- `Unifloat * atan2_UF (Unifloat *x, Unifloat *y)`  
*The arc tangent of `Unifloat` number divided by another one.*
- `Unifloat * arrangeArgument_UF (Unifloat *x)`  
*Make a number to be between 0 and  $2\pi$ .*
- `Unifloat * calcAtan_UF (Unifloat *x, Unifloat *y, int i)`  
*Auxiliary function needed to calculate `atan` iteratively.*
- `Unifloat * calcTan_UF (Unifloat *x, int i)`  
*Auxiliary function needed to calculate `tan` iteratively.*

#### 4.10.1 Detailed Description

Real trigonometric functions.

#### 4.10.2 Function Documentation

##### 4.10.2.1 `Unifloat* acos_UF ( Unifloat * x )`

The arc cosine of the given `Unifloat` number.

##### Returns

The arc cosine of `x`.

The function creates and returns the object that have to be removed later using `delete_UF`.

##### 4.10.2.2 `Unifloat* arrangeArgument_UF ( Unifloat * x )`

Make a number to be between 0 and  $2\pi$ .

##### Returns

The function creates and returns the object that have to be removed later using `delete_UF`.

##### 4.10.2.3 `Unifloat* asin_UF ( Unifloat * x )`

The arc sine of the given `Unifloat` number.

##### Returns

The arc sine of `x`.

The function creates and returns the object that have to be removed later using `delete_UF`.

##### 4.10.2.4 `Unifloat* atan2_UF ( Unifloat * x, Unifloat * y )`

The arc tangent of `Unifloat` number divided by another one.

##### Returns

The arc tangent of `x/y`.

The function creates and returns the object that have to be removed later using `delete_UF`.

#### 4.10.2.5 Unifloat\* atan\_UF ( Unifloat \* x )

The arc tangent of the given Unifloat number.

##### Returns

The arc tangent of **x**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.10.2.6 Unifloat\* calcAtan\_UF ( Unifloat \* x, Unifloat \* y, int i )

Auxiliary function needed to calculate atan iteratively.

##### Returns

The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.10.2.7 Unifloat\* calcTan\_UF ( Unifloat \* x, int i )

Auxiliary function needed to calculate tan iteratively.

##### Returns

The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.10.2.8 Unifloat\* cos\_UF ( Unifloat \* x )

The cosine of the given Unifloat number.

##### Returns

The cosine of **x**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.10.2.9 Unifloat\* sin\_UF ( Unifloat \* x )

The sine of the given Unifloat number.

##### Returns

The sine of **x**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.10.2.10 Unifloat\* tan\_UF ( Unifloat \* x )

The tangent of the given Unifloat number.

##### Returns

The tan of **x**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

### 4.11 unifloat.h File Reference

Basic Unifloat operations.

```
#include "unifloat/config.h"
#include "unifloat/cstring.h"
```

## Data Structures

- struct [Unifloat](#)

*The main structure that represents a number in libunifloat.*

## Macros

- #define [MAX\\_SIZE\\_UNIFLOAT](#) 7  
*Maximal size of [Unifloat](#) mantissa.*
- #define **true** 1
- #define **false** 0
- #define [PRECISION](#) 90  
*Maximal count of mantissa bits, that can be used for calculations.*
- #define [COMPARE\\_PRECISION](#) 15  
*Specifies the accuracy of Unifloats comparing.*
- #define **maxExp\_DoubleT** 1024
- #define **minExp\_DoubleT** -1021
- #define **digMant\_DoubleT** 53
- #define **digExp\_DoubleT** 11
- #define **size\_DoubleT** 8
- #define **sizeInLongs\_DoubleT** (sizeof(double) - 4) / sizeof(long) + 1

## Typedefs

- typedef unsigned int **uint**
- typedef unsigned long **ulong**
- typedef int **Bool**
- typedef enum [UnifloatKind](#) [UnifloatKind](#)
- typedef struct [Unifloat](#) [Unifloat](#)  
*The main structure that represents a number in libunifloat.*
- typedef [Unifloat](#) \*(\* [caller\\_UF](#)) ([Unifloat](#) \*)  
*Represents a template of any function that takes one [Unifloat](#) argument.*
- typedef [Unifloat](#) \*(\* [caller\\_UF\\_nx](#)) (int, [Unifloat](#) \*)  
*Represents a template of any function that takes integer and [Unifloat](#) arguments.*
- typedef [Unifloat](#) \*(\* [caller\\_UF\\_xy](#)) ([Unifloat](#) \*, [Unifloat](#) \*)  
*Represents a template of any function that takes two [Unifloat](#) arguments.*

## Enumerations

- enum [UnifloatKind](#) { **Infinity**, **NaN**, **Normal** }

## Functions

- void [initialize\\_UF](#) (void)  
*Initialize global variables.*
- void [finalize\\_UF](#) (void)  
*Frees the memory used by global variables.*
- [Unifloat](#) \* [call1\\_arg1](#) ([caller\\_UF](#) func, [Unifloat](#) \*x)  
*brief Function needed to simplify memory management.*
- [Unifloat](#) \* [call2\\_arg1](#) ([caller\\_UF\\_xy](#) func, [Unifloat](#) \*x, [Unifloat](#) \*y)  
*Function needed to simplify memory management.*

- `Unifloat * call2_arg2 (caller_UF_xy func, Unifloat *x, Unifloat *y)`  
*Function needed to simplify memory management.*
- `Unifloat * create_UF (int sign, int exponent, UnifloatKind kind)`  
*Create a new Unifloat number.*
- `void delete_UF (Unifloat *u)`  
*Delete a Unifloat number and frees the used memory.*
- `void delete_UFs (Unifloat *u,...)`  
*Delete a list of Unifloat numbers and frees the used memory.*
- `Unifloat * createZero_UF (void)`  
*Create a new Unifloat object that represent Zero.*
- `Unifloat * createOne_UF (void)`  
*Create a new Unifloat object that represent One.*
- `Unifloat * clone (Unifloat *src)`  
*Creates a copy of given Unifloat number.*
- `void copy (Unifloat *src, Unifloat *dst)`  
*Copies a given Unifloat number from /b src to /b dst.*
- `Bool isNormal_UF (Unifloat *x)`  
*Check if the Kind of number is Normal.*
- `Bool isInfinity_UF (Unifloat *x)`  
*Check if the Kind of number is Infinity.*
- `Bool isNan_UF (Unifloat *x)`  
*Check if the Kind of number is NaN.*
- `Bool isZero_UF (Unifloat *x)`  
*Check if the Unifloat number is Zero.*
- `Bool isOverflow_UF (Unifloat *x)`  
*Check if the Unifloat number is overflowed.*
- `Bool isUnderflow_UF (Unifloat *x)`  
*Check if the Unifloat number is underflowed.*
- `Unifloat * abs_UF (Unifloat *x)`  
*Returns a absolute value of given number.*
- `Unifloat * normalize_UF (Unifloat *x)`  
*Returns a normalized value of a given number.*
- `void setMant_UF (Unifloat *x, uint index, uint bit)`  
*Sets one bit of Unifloat mantissa to a given value.*
- `uint getMant_UF (Unifloat *x, uint index)`  
*Gets one bit of Unifloat mantissa.*
- `Unifloat * round_UF (Unifloat *x, uint precision)`  
*Rounds a Unifloat number.*
- `Unifloat * changeSign_UF (Unifloat *x, int s)`  
*Rounds a Unifloat number.*
- `int compare_UF (Unifloat *x, Unifloat *y)`  
*Compares two Unifloat numbers.*
- `int compareWithError_UF (Unifloat *x)`  
*Checks that the given Unifloat number is Error.*
- `int compareWithPrecision_UF (Unifloat *x, Unifloat *y, int amount)`  
*Compares two Unifloat numbers with the specified precision.*
- `Unifloat * add_UF (Unifloat *x, Unifloat *y)`  
*The sum of two Unifloat numbers.*
- `Unifloat * sub_UF (Unifloat *x, Unifloat *y)`  
*The difference between two Unifloat numbers.*
- `Unifloat * mul_UF (Unifloat *x, Unifloat *y)`

- The product of two *Unifloat* numbers.

  - *Unifloat* \* *div\_UF* (*Unifloat* \*x, *Unifloat* \*y)
- The ratio of two *Unifloat* numbers.

  - *Unifloat* \* *convertFloat\_UF* (float x)
- Convert a given **float** number into *Unifloat* number.

  - *Unifloat* \* *convertDouble\_UF* (double x)
- Convert a given **double** number into *Unifloat* number.

  - *Unifloat* \* *convertLongDouble\_UF* (long double x)
- Convert a given long double number into *Unifloat* number.

  - float *convertUnifloat\_Float* (*Unifloat* \*x)
- Convert a given *Unifloat* number into **float** number.

  - double *convertUnifloat\_Double* (*Unifloat* \*x)
- Convert a given *Unifloat* number into **double** number.

  - long double *convertUnifloat\_LongDouble* (*Unifloat* \*x)
- Convert a given *Unifloat* number into long double number.

  - *Unifloat* \* *convertInteger\_UF* (int number)
- Convert a given **integer** number into *Unifloat* number.

  - int *convertUnifloat\_Integer* (*Unifloat* \*x, int \*error)
- Convert a given *Unifloat* number into **integer** number.

  - *Unifloat* \* *convertBinaryString\_UF* (CString \*number)
- Parse the **CString** that contains a binary number and create the *Unifloat* number.

  - *Unifloat* \* *convertString\_UF* (CString \*number)
- Parse the **CString** that contains a decimal number and create the *Unifloat* number.

  - *Unifloat* \* *power\_UF* (*Unifloat* \*x, int n)
- Raise the value of *Unifloat* number **x** to the power of **n**.

  - *Unifloat* \* *factorial\_UF* (int n)
- Compute factorial of an integer number **n**.

  - void *print\_UF* (*Unifloat* \*x)
- Print value of the *Unifloat* number **x** in double format to stdout.

  - double *call\_UF* (*caller\_UF* func, double x)
- Call any *Unifloat* function **func** with the **x** argument in double.

  - double *call\_UF\_nx* (*caller\_UF\_nx* func, int n, double x)
- Call any *Unifloat* function **func** with the **n** and **x** arguments.

#### Variables

- *Unifloat* \* *Pi*

The number PI.
- *Unifloat* \* *Gamma*

The number GAMMA.
- *Unifloat* \* *E*

The number E.
- *Unifloat* \* *Ln\_2*
- *Unifloat* \* *Log2\_e*
- *Unifloat* \* *Log10\_e*
- *Unifloat* \* *Log2\_10*
- *Unifloat* \* *Epsilon*
- *Unifloat* \* *max\_UF*

Used to test the *Unifloat* for overflow.
- *Unifloat* \* *min\_UF*

Used to test the *Unifloat* for underflow.

- `Unifloat * infinity_UF`  
*The positive infinity `Unifloat`.*
- `Unifloat * nan_UF`  
*The Not-A-Number `Unifloat`.*

#### 4.11.1 Detailed Description

Basic `Unifloat` operations.

#### 4.11.2 Typedef Documentation

##### 4.11.2.1 `typedef enum UnifloatKind UnifloatKind`

Kind of `Unifloat`.

#### 4.11.3 Enumeration Type Documentation

##### 4.11.3.1 `enum UnifloatKind`

Kind of `Unifloat`.

Definition at line 57 of file `unifloat.h`.

#### 4.11.4 Function Documentation

##### 4.11.4.1 `Unifloat* abs_UF ( Unifloat * x )`

Returns a absolute value of given number.

##### Returns

The `Unifloat` object that represents absolute value of a number `x`.  
Function creates and returns the object that have to be removed later using `delete_UF`.

##### 4.11.4.2 `Unifloat* add_UF ( Unifloat * x, Unifloat * y )`

The sum of two `Unifloat` numbers.

##### Returns

The `Unifloat` object that represents a sum of `x` and `y`.  
The function creates and returns the object that have to be removed later using `delete_UF`.

##### 4.11.4.3 `Unifloat* call1_arg1 ( caller_UF func, Unifloat * x )`

brief Function needed to simplify memory management.

In code "`x = Foo(x)`" variable `x` should be deleted first. Call "`x = call1_arg1(&Foo, x)`" and the old `Unifloat` `x` will be removed.

##### 4.11.4.4 `Unifloat* call2_arg1 ( caller_UF_xy func, Unifloat * x, Unifloat * y )`

Function needed to simplify memory management.

In code "`x = Foo(x, y)`" variable `x` should be deleted first. Call "`x = call2_arg1(&Foo, x, y)`" and the old `Unifloat` `x` will be removed.



## 4.11.4.5 Unifloat\* call2\_arg2 ( caller\_UF\_xy func, Unifloat \* x, Unifloat \* y )

Function needed to simplify memory management.

In code "x = Foo(y, x)" variable x should be deleted first. Call "x = call2\_arg2(&Foo, y, x)" and the old Unifloat x will be removed.

## 4.11.4.6 double call\_UF ( caller\_UF func, double x )

Call any Unifloat function **func** with the **x** argument in double.

## Returns

The return value of the function **func** in double.

## 4.11.4.7 double call\_UF\_nx ( caller\_UF\_nx func, int n, double x )

Call any Unifloat function **func** with the **n** and **x** arguments.

## Returns

The return value of the function **func** in double.

## 4.11.4.8 Unifloat\* changeSign\_UF ( Unifloat \* x, int s )

Rounds a Unifloat number.

## Returns

The function creates and returns the object that have to be removed later using **delete\_UF**.

## Parameters

|   |                                                                                    |
|---|------------------------------------------------------------------------------------|
| s | if it is set to 0 - change the sign to opposite. Otherwise multiply the sign by s. |
|---|------------------------------------------------------------------------------------|

## 4.11.4.9 Unifloat\* clone ( Unifloat \* src )

Creates a copy of given Unifloat number.

## Returns

The copy of given Unifloat number.

The function creates and returns the object that have to be removed later using **delete\_UF**.

## 4.11.4.10 int compare\_UF ( Unifloat \* x, Unifloat \* y )

Compares two Unifloat numbers.

## Returns

- 0, if x==y (or they are both Infinity or NaN);
- 1, if x>y;
- 1, if x<y;
- 2, if one (and only one) of x or y is NaN.

## 4.11.4.11 int compareWithError\_UF ( Unifloat \* x )

Checks that the given Unifloat number is Error.

Error is a Unifloat with **exp** set to -150

### Returns

The result of comparing the internal Error object with the given one using **compare\_UF**.

#### 4.11.4.12 **int** `compareWithPrecision_UF ( Unifloat * x, Unifloat * y, int amount )`

Compares two **Unifloat** numbers with the specified precision.

### Returns

- 0, if x==y (or they are both Infinity or NaN);
- 1, if x>y;
- 1, if x<y;
- 2, if one (and only one) of x or y is NaN.

#### 4.11.4.13 **Unifloat\*** `convertBinaryString_UF ( CString * number )`

Parse the **CString** that contains a binary number and create the **Unifloat** number.

### Returns

The **Unifloat** object that represents **x** in **Unifloat**.  
The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.11.4.14 **Unifloat\*** `convertDouble_UF ( double x )`

Convert a given **double** number into **Unifloat** number.

### Returns

The **Unifloat** object that represents a given number **x** in **Unifloat**.  
The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.11.4.15 **Unifloat\*** `convertFloat_UF ( float x )`

Convert a given **float** number into **Unifloat** number.

### Returns

The **Unifloat** object that represents a given number **x** in **Unifloat**.  
The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.11.4.16 **Unifloat\*** `convertInteger_UF ( int number )`

Convert a given **integer** number into **Unifloat** number.

### Returns

The **Unifloat** object that represents a given number **number** in **Unifloat**.  
The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.11.4.17 **Unifloat\*** `convertLongDouble_UF ( long double x )`

Convert a given long double number into **Unifloat** number.

### Returns

The **Unifloat** object that represents a given number **x** in **Unifloat**.  
The function creates and returns the object that have to be removed later using **delete\_UF**.

## 4.11.4.18 Unifloat\* convertString\_UF ( CString \* number )

Parse the **CString** that contains a decimal number and create the **Unifloat** number.

## Returns

The **Unifloat** object that represents **x** in **Unifloat**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

## 4.11.4.19 double convertUnifloat\_Double ( Unifloat \* x )

Convert a given **Unifloat** number into **double** number.

## Returns

**double** number that represents **x**.

## 4.11.4.20 float convertUnifloat\_Float ( Unifloat \* x )

Convert a given **Unifloat** number into **float** number.

## Returns

**float** number that represents **x**.

## 4.11.4.21 int convertUnifloat\_Integer ( Unifloat \* x, int \* error )

Convert a given **Unifloat** number into **integer** number.

## Parameters

|              |                                                                                                                                                                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>error</i> | The function returns an error code using this parameter.<br><b>0</b> means that it is possible to convert x to integer number,<br><b>1</b> means that conversion is impossible,<br><b>2</b> means that conversion isn't precisely and<br><b>3</b> means that result isn't representable in <b>int</b> type. |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Returns

The **integer** number that represent a given **Unifloat** number **x**.

## 4.11.4.22 long double convertUnifloat\_LongDouble ( Unifloat \* x )

Convert a given **Unifloat** number into long double number.

## Returns

long double number that represents **x**.

## 4.11.4.23 Unifloat\* create\_UF ( int sign, int exponent, UnifloatKind kind )

Create a new **Unifloat** number.

Mantissa is filled with zeros.

## Returns

The new **Unifloat** object.

The created object have to be removed later using **delete\_UF**.

#### 4.11.4.24 Unifloat\* createOne\_UF ( void )

Create a new [Unifloat](#) object that represent One.

##### Returns

The new [Unifloat](#) One object.

The created object have to be removed later using **delete\_UF**.

#### 4.11.4.25 Unifloat\* createZero\_UF ( void )

Create a new [Unifloat](#) object that represent Zero.

##### Returns

The new [Unifloat](#) Zero object.

The created object have to be removed later using **delete\_UF**.

#### 4.11.4.26 void delete\_UFs ( Unifloat \* u, ... )

Delete a list of [Unifloat](#) numbers and frees the used memory.

The last argument should be **NULL**

#### 4.11.4.27 Unifloat\* div\_UF ( Unifloat \* x, Unifloat \* y )

The ratio of two [Unifloat](#) numbers.

##### Returns

The [Unifloat](#) object that represents a ratio of **x** and **y**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.11.4.28 Unifloat\* factorial\_UF ( int n )

Compute factorial of an integer number **n**.

##### Returns

The [Unifloat](#) object that represents factorial of **n** in [Unifloat](#).

The function creates and returns the object that have to be removed later using **delete\_UF**.

#### 4.11.4.29 void finalize\_UF ( void )

Frees the memory used by global variables.

Call it when you end working with libunifloat.

#### 4.11.4.30 uint getMant\_UF ( Unifloat \* x, uint index )

Gets one bit of [Unifloat](#) mantissa.

##### Parameters

|              |                        |
|--------------|------------------------|
| <i>index</i> | the bit number to get. |
|--------------|------------------------|

#### 4.11.4.31 void initialize\_UF ( void )

Initialize global variables.

Call it when you start working with libunifloat.

**4.11.4.32 Unifloat\* mul\_UF ( Unifloat \* x, Unifloat \* y )**

The product of two [Unifloat](#) numbers.

**Returns**

The [Unifloat](#) object that represents a product of **x** and **y**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

**4.11.4.33 Unifloat\* normalize\_UF ( Unifloat \* x )**

Returns a normalized value of a given number.

Normalized [Unifloat](#) have the first bit of mantissa set to 1.

**Returns**

The [Unifloat](#) object that represents normalized value of a number **x**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

**4.11.4.34 Unifloat\* power\_UF ( Unifloat \* x, int n )**

Raise the value of [Unifloat](#) number **x** to the power of **n**.

**Returns**

The [Unifloat](#) object that represents **x** in the power of **n** in [Unifloat](#).

The function creates and returns the object that have to be removed later using **delete\_UF**.

**4.11.4.35 Unifloat\* round\_UF ( Unifloat \* x, uint precision )**

Rounds a [Unifloat](#) number.

**Returns**

The function creates and returns the object that have to be removed later using **delete\_UF**.

**Parameters**

|                  |                                    |
|------------------|------------------------------------|
| <i>precision</i> | count of significant bits to save. |
|------------------|------------------------------------|

**4.11.4.36 void setMant\_UF ( Unifloat \* x, uint index, uint bit )**

Sets one bit of [Unifloat](#) mantissa to a given value.

**Parameters**

|              |                           |
|--------------|---------------------------|
| <i>index</i> | the bit number to be set. |
| <i>bit</i>   | the value of bit          |

**4.11.4.37 Unifloat\* sub\_UF ( Unifloat \* x, Unifloat \* y )**

The difference between two [Unifloat](#) numbers.

**Returns**

The [Unifloat](#) object that represents a difference between **x** and **y**.

The function creates and returns the object that have to be removed later using **delete\_UF**.

## 4.12 unifloat\_complex.h File Reference

Basic [UnifloatComplex](#) operations.

```
#include "unifloat/unifloat.h"
```

### Data Structures

- struct [UnifloatComplex](#)  
*Represents a complex number in libunifloat.*

### Typedefs

- typedef struct [UnifloatComplex](#) [UnifloatComplex](#)  
*Represents a complex number in libunifloat.*

### Functions

- [UnifloatComplex](#) \* [clone\\_Complex](#) ([UnifloatComplex](#) \*src)  
*Creates a copy of given [UnifloatComplex](#) number.*
- void [copy](#) ([Unifloat](#) \*src, [Unifloat](#) \*dst)  
*Copies a given [Unifloat](#) object data to another one.*
- void [copy\\_Complex](#) ([UnifloatComplex](#) \*src, [UnifloatComplex](#) \*dst)  
*Copies a given [UnifloatComplex](#) object data to another one.*
- [UnifloatComplex](#) \* [create\\_UFComplex](#) ([Unifloat](#) \*Re, [Unifloat](#) \*Im)  
*Create a new [UnifloatComplex](#) number.*
- void [delete\\_UFComplex](#) ([UnifloatComplex](#) \*x)  
*Deletes the [UnifloatComplex](#) number.*
- void [delete\\_UFsComplex](#) ([UnifloatComplex](#) \*p1,...)  
*Deletes the list of [UnifloatComplex](#) numbers.*
- [Unifloat](#) \* [abs\\_UFComplex](#) ([UnifloatComplex](#) \*x)  
*Get the absolute value of the given [UnifloatComplex](#) number.*
- [Unifloat](#) \* [carg\\_UF](#) ([UnifloatComplex](#) \*x)  
*Get the argument of the given [UnifloatComplex](#) number.*
- [UnifloatComplex](#) \* [add\\_UFComplex](#) ([UnifloatComplex](#) \*x, [UnifloatComplex](#) \*y)  
*The sum of two [Unifloat](#) numbers.*
- [UnifloatComplex](#) \* [sub\\_UFComplex](#) ([UnifloatComplex](#) \*x, [UnifloatComplex](#) \*y)  
*The difference between two [Unifloat](#) numbers.*
- [UnifloatComplex](#) \* [mul\\_UFComplex](#) ([UnifloatComplex](#) \*x, [UnifloatComplex](#) \*y)  
*The product of two [Unifloat](#) numbers.*
- [UnifloatComplex](#) \* [div\\_UFComplex](#) ([UnifloatComplex](#) \*x, [UnifloatComplex](#) \*y)  
*The ratio of two [Unifloat](#) numbers.*

#### 4.12.1 Detailed Description

Basic [UnifloatComplex](#) operations.

### 4.12.2 Typedef Documentation

#### 4.12.2.1 typedef struct UnifloatComplex UnifloatComplex

Represents a complex number in libunifloat.

The main structure for complex calculations.

### 4.12.3 Function Documentation

#### 4.12.3.1 Unifloat\* abs\_UFComplex ( UnifloatComplex \* x )

Get the absolute value of the given [UnifloatComplex](#) number.

##### Returns

The absolute value of **x**.

The function creates and returns the object that have to be removed later using **delete\_UFComplex**.

#### 4.12.3.2 UnifloatComplex\* add\_UFComplex ( UnifloatComplex \* x, UnifloatComplex \* y )

The sum of two [Unifloat](#) numbers.

##### Returns

The [UnifloatComplex](#) object that represents a sum of **x** and **y**.

The function creates and returns the object that have to be removed later using **delete\_UFComplex**.

#### 4.12.3.3 Unifloat\* carg\_UF ( UnifloatComplex \* x )

Get the argument of the given [UnifloatComplex](#) number.

##### Returns

The argument of **x**.

The function creates and returns the object that have to be removed later using **delete\_UFComplex**.

#### 4.12.3.4 UnifloatComplex\* clone\_Complex ( UnifloatComplex \* src )

Creates a copy of given [UnifloatComplex](#) number.

##### Returns

The copy of given [UnifloatComplex](#) number.

The function creates and returns the object that have to be removed later using **delete\_UFComplex**.

#### 4.12.3.5 UnifloatComplex\* create\_UFComplex ( Unifloat \* Re, Unifloat \* Im )

Create a new [UnifloatComplex](#) number.

Mantissa is filled with zeros.

##### Returns

The new [UnifloatComplex](#) object.

The created object have to be removed later using **delete\_UFComplex**.

#### 4.12.3.6 void delete\_UFComplex ( UnifloatComplex \* x )

Deletes the [UnifloatComplex](#) number.

This function have to be called for every created [UnifloatComplex](#) to free the memory it uses.

#### 4.12.3.7 `UnifloatComplex* div_UFComplex ( UnifloatComplex * x, UnifloatComplex * y )`

The ratio of two `Unifloat` numbers.

##### Returns

The `UnifloatComplex` object that represents a ratio of `x` and `y`.

The function creates and returns the object that have to be removed later using `delete_UFComplex`.

#### 4.12.3.8 `UnifloatComplex* mul_UFComplex ( UnifloatComplex * x, UnifloatComplex * y )`

The product of two `Unifloat` numbers.

##### Returns

The `UnifloatComplex` object that represents a product of `x` and `y`.

The function creates and returns the object that have to be removed later using `delete_UFComplex`.

#### 4.12.3.9 `UnifloatComplex* sub_UFComplex ( UnifloatComplex * x, UnifloatComplex * y )`

The difference between two `Unifloat` numbers.

##### Returns

The `UnifloatComplex` object that represents a difference between `x` and `y`.

The function creates and returns the object that have to be removed later using `delete_UFComplex`.



## Index

abs\_UF  
    unifloat.h, 14  
abs\_UFComplex  
    unifloat\_complex.h, 21  
acos\_UF  
    trig.h, 9  
add\_UF  
    unifloat.h, 14  
add\_UFComplex  
    unifloat\_complex.h, 21  
arrangeArgument\_UF  
    trig.h, 9  
asin\_UF  
    trig.h, 9  
atan2\_UF  
    trig.h, 9  
atan\_UF  
    trig.h, 9  
  
bessel.h, 3  
  
CString, 2  
cacosh\_UF  
    ctrig.h, 6  
calcAtan\_UF  
    trig.h, 10  
calcTan\_UF  
    trig.h, 10  
call1\_arg1  
    unifloat.h, 14  
call2\_arg1  
    unifloat.h, 14  
call2\_arg2  
    unifloat.h, 14  
call\_UF  
    unifloat.h, 15  
call\_UF\_nx  
    unifloat.h, 15  
carg\_UF  
    unifloat\_complex.h, 21  
casinh\_UF  
    ctrig.h, 6  
catanh\_UF  
    ctrig.h, 6  
ccosh\_UF  
    ctrig.h, 6  
cexp.h, 4  
changeSign\_UF  
    unifloat.h, 15  
clone  
    unifloat.h, 15  
clone\_Complex  
    unifloat\_complex.h, 21  
compare\_UF  
    unifloat.h, 15  
compareWithError\_UF  
    unifloat.h, 15  
compareWithPrecision\_UF  
    unifloat.h, 16  
constants.h, 4  
convertBinaryString\_UF  
    unifloat.h, 16  
convertDouble\_UF  
    unifloat.h, 16  
convertFloat\_UF  
    unifloat.h, 16  
convertInteger\_UF  
    unifloat.h, 16  
convertLongDouble\_UF  
    unifloat.h, 16  
convertString\_UF  
    unifloat.h, 16  
convertUnifloat\_Double  
    unifloat.h, 17  
convertUnifloat\_Float  
    unifloat.h, 17  
convertUnifloat\_Integer  
    unifloat.h, 17  
convertUnifloat\_LongDouble  
    unifloat.h, 17  
cos\_UF  
    trig.h, 10  
create\_UF  
    unifloat.h, 17  
create\_UFComplex  
    unifloat\_complex.h, 21  
createOne\_UF  
    unifloat.h, 17  
createZero\_UF  
    unifloat.h, 18  
csinh\_UF  
    ctrig.h, 6  
cstring.h, 5  
ctanh\_UF  
    ctrig.h, 7  
ctrig.h, 5  
    cacosh\_UF, 6  
    casinh\_UF, 6  
    catanh\_UF, 6  
    ccosh\_UF, 6  
    csinh\_UF, 6  
    ctanh\_UF, 7  
  
debug.h, 7  
delete\_UFComplex  
    unifloat\_complex.h, 21  
delete\_UFs  
    unifloat.h, 18  
div\_UF  
    unifloat.h, 18  
div\_UFComplex

- unifloat\_complex.h, 21
- exp.h, 7
- factorial\_UF
  - unifloat.h, 18
- finalize\_UF
  - unifloat.h, 18
- gamma.h, 7
- getMant\_UF
  - unifloat.h, 18
- initialize\_UF
  - unifloat.h, 18
- libunifloat.h, 8
- mant
  - Unifloat, 3
- mul\_UF
  - unifloat.h, 18
- mul\_UFComplex
  - unifloat\_complex.h, 22
- normalize\_UF
  - unifloat.h, 19
- power\_UF
  - unifloat.h, 19
- round\_UF
  - unifloat.h, 19
- setMant\_UF
  - unifloat.h, 19
- sin\_UF
  - trig.h, 10
- sub\_UF
  - unifloat.h, 19
- sub\_UFComplex
  - unifloat\_complex.h, 22
- tan\_UF
  - trig.h, 10
- trig.h, 8
  - acos\_UF, 9
  - arrangeArgument\_UF, 9
  - asin\_UF, 9
  - atan2\_UF, 9
  - atan\_UF, 9
  - calcAtan\_UF, 10
  - calcTan\_UF, 10
  - cos\_UF, 10
  - sin\_UF, 10
  - tan\_UF, 10
- Unifloat, 2
  - mant, 3
- unifloat.h, 10
  - abs\_UF, 14
  - add\_UF, 14
  - call1\_arg1, 14
  - call2\_arg1, 14
  - call2\_arg2, 14
  - call\_UF, 15
  - call\_UF\_nx, 15
  - changeSign\_UF, 15
  - clone, 15
  - compare\_UF, 15
  - compareWithError\_UF, 15
  - compareWithPrecision\_UF, 16
  - convertBinaryString\_UF, 16
  - convertDouble\_UF, 16
  - convertFloat\_UF, 16
  - convertInteger\_UF, 16
  - convertLongDouble\_UF, 16
  - convertString\_UF, 16
  - convertUnifloat\_Double, 17
  - convertUnifloat\_Float, 17
  - convertUnifloat\_Integer, 17
  - convertUnifloat\_LongDouble, 17
  - create\_UF, 17
  - createOne\_UF, 17
  - createZero\_UF, 18
  - delete\_UFs, 18
  - div\_UF, 18
  - factorial\_UF, 18
  - finalize\_UF, 18
  - getMant\_UF, 18
  - initialize\_UF, 18
  - mul\_UF, 18
  - normalize\_UF, 19
  - power\_UF, 19
  - round\_UF, 19
  - setMant\_UF, 19
  - sub\_UF, 19
  - UnifloatKind, 14
- unifloat\_complex.h, 20
  - abs\_UFComplex, 21
  - add\_UFComplex, 21
  - carg\_UF, 21
  - clone\_Complex, 21
  - create\_UFComplex, 21
  - delete\_UFComplex, 21
  - div\_UFComplex, 21
  - mul\_UFComplex, 22
  - sub\_UFComplex, 22
  - UnifloatComplex, 21
- UnifloatComplex, 3
  - unifloat\_complex.h, 21
- UnifloatKind
  - unifloat.h, 14