Ghoti.io CUtil

0.1

Generated by Doxygen 1.9.1

Chapter 1

Ghoti.io CUtil Library

1.0.1 Overview

The Ghoti.io CUtil Library is a collection of C libraries to aid in the development of C applications by providing helpful and commonly used tools and features.

1.0.2 Installation

1.0.2.1 Build From Source

```
make build
make install
```

1.0.3 Compiling With The Library

```
cc `pkg-config --libs --cflags ghoti.io-cutil_dev` <YOUR SOURCE FILE>
```

Chapter 2

File Index

2.1 File List

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

Chapter 3

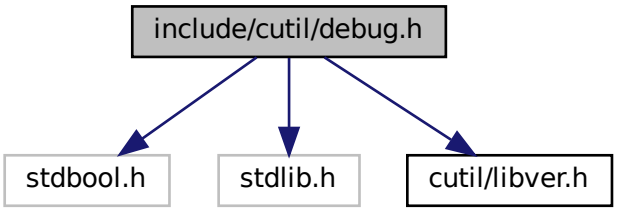
File Documentation

3.1 include/cutil/debug.h File Reference

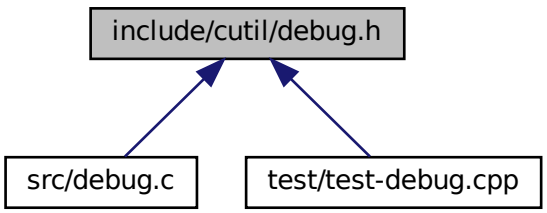
Header file for debugging-related functions.

```
#include <stdbool.h>
#include <stdlib.h>
#include "cutil/libver.h"
```

Include dependency graph for debug.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **malloc**(size) gcu_malloc(size, __FILE__, __LINE__)
- #define **calloc**(nitems, size) gcu_calloc(nitems, size, __FILE__, __LINE__)
- #define **realloc**(pointer, size) gcu_realloc(pointer, size, __FILE__, __LINE__)
- #define **free**(pointer) gcu_free(pointer, __FILE__, __LINE__)

Functions

- void * gcu_malloc (size_t size, const char *file, size_t line)

  Wrapper for the standard malloc() function.
- void * gcu_calloc (size_t nitems, size_t size, const char *file, size_t line)

  Wrapper for the standard calloc() function.
- void * gcu_realloc (void *pointer, size_t size, const char *file, size_t line)

  Wrapper for the standard realloc() function.
- void gcu_free (void *pointer, const char *file, size_t line)

  Wrapper for the standard free() function.
- void gcu_mem_start (void)

  Signal that intercepted memory management calls should be logged to stderr.
- void gcu_mem_stop (void)

  Signal that intercepted memory management calls should no longer be logged to stderr.

3.1.1 Detailed Description

Header file for debugging-related functions.

3.1.2 Function Documentation

3.1.2.1 gcu_calloc()

```
void* gcu_calloc (
            size_t nitems,
            size_t size,
            const char * file,
            size_t line )
```

Wrapper for the standard calloc() function.

Parameters

nitems	The number of items to allocate.
size	The number of bytes in each item.
file	The name of the file from which the function was called.
line	The line number on which the function was called.

Returns

The beginning byte of the allocated memory.

3.1.2.2 gcu_free()

```
void gcu_free (
            void * pointer,
            const char * file,
            size_t line )
```

Wrapper for the standard free() function.

Parameters

pointer	The beginning byte of the currently allocated memory.
file	The name of the file from which the function was called.
line	The line number on which the function was called.

3.1.2.3 gcu_malloc()

```
void* gcu_malloc (
            size_t size,
            const char * file,
            size_t line )
```

Wrapper for the standard malloc() function.

Parameters

size	The number of bytes requested.
file	The name of the file from which the function was called.
line	The line number on which the function was called.

Returns

The beginning byte of the allocated memory.

3.1.2.4 gcu_realloc()

```
void* gcu_realloc (
            void * pointer,
```

```
size_t size,
const char * file,
size_t line )
```

Wrapper for the standard realloc() function.

Parameters

pointer	The beginning byte of the currently allocated memory.
size	The newly requested size.
file	The name of the file from which the function was called.
line	The line number on which the function was called.

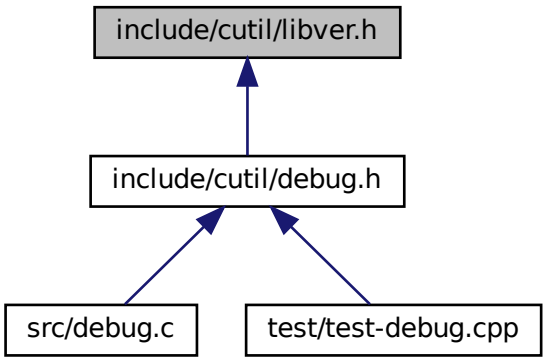
Returns

The beginning byte of the reallocated memory.

3.2 include/cutil/libver.h File Reference

Header file used to control the version numbering and function namespace for all of the library.

This graph shows which files directly or indirectly include this file:



Macros

- #define GHOTIIO_CUTIL_NAME ghotiio_cutil_dev

  Used in conjunction with the GHOTIIO_CUTIL...
- #define GHOTIIO_CUTIL_VERSION "dev"

  String representation of the version, provided as a convenience to the programmer.
- #define GHOTIIO_CUTIL(NAME) GHOTIIO_CUTIL_CONCAT(GHOTIIO_CUTIL_NAME, _ ## NAME)

  Macro to generate a "namespaced" version of an identifier.
- #define GHOTIIO_CUTIL_CONCAT_INNER(a, b) a ## b

  Helper macro to concatenate the `#define`s properly.
- #define GHOTIIO_CUTIL_CONCAT(a, b) GHOTIIO_CUTIL_CONCAT_INNER(a,b)

  Helper macro to concatenate the `#define`s properly.

3.2.1 Detailed Description

Header file used to control the version numbering and function namespace for all of the library.

3.2.2 Macro Definition Documentation

3.2.2.1 GHOTIIO_CUTIL

```
#define GHOTIIO_CUTIL(
              NAME ) GHOTIIO_CUTIL_CONCAT(GHOTIIO_CUTIL_NAME, _ ## NAME)
```

Macro to generate a "namespaced" version of an identifier.

Parameters

NAME	The name which will be prepended with the GHOTIIO_CUTIL_NAME.

3.2.2.2 GHOTIIO_CUTIL_CONCAT

```
#define GHOTIIO_CUTIL_CONCAT(
              a,
              b ) GHOTIIO_CUTIL_CONCAT_INNER(a,b)
```

Helper macro to concatenate the #defines properly.

It requires two levels of processing.

Parameters

a	The first part of the identifier.
b	The second part of the identifier.

Returns

A call to the GHOTIIO_CUTIL_CONCAT_INNER() macro.

3.2.2.3 GHOTIIO_CUTIL_CONCAT_INNER

```
#define GHOTIIO_CUTIL_CONCAT_INNER(
              a,
              b ) a ## b
```

Helper macro to concatenate the `#define`s properly.

It requires two levels of processing.

This macro should only be called by the `GHOTIIO_CUTIL_CONCAT()` macro.

Parameters

a	The first part of the identifier.
b	The second part of the identifier.

Returns

The concatenation of `a` to `b`.

3.2.2.4 GHOTIIO_CUTIL_NAME

```
#define GHOTIIO_CUTIL_NAME ghotiio_cutil_dev
```

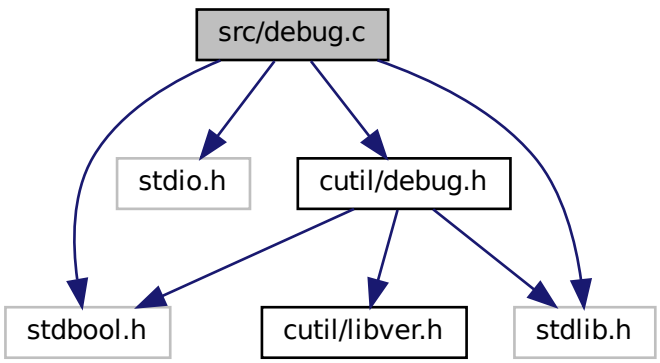
Used in conjunction with the GHOTIIO_CUTIL...

macros to produce a namespaced function name for use by all exported functions in this library.

3.3 src/debug.c File Reference

```
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include "cutil/debug.h"
```

Include dependency graph for debug.c:



Functions

- void * gcu_malloc (size_t size, const char *file, size_t line)

  Wrapper for the standard malloc() function.
- void * gcu_calloc (size_t nitems, size_t size, const char *file, size_t line)

  Wrapper for the standard calloc() function.
- void * gcu_realloc (void *pointer, size_t size, const char *file, size_t line)

  Wrapper for the standard realloc() function.
- void gcu_free (void *pointer, const char *file, size_t line)

  Wrapper for the standard free() function.
- void gcu_mem_start (void)

  Signal that intercepted memory management calls should be logged to stderr.
- void gcu_mem_stop (void)

  Signal that intercepted memory management calls should no longer be logged to stderr.

Variables

- bool **capture** = true

3.3.1 Function Documentation

3.3.1.1 gcu_calloc()

```
void* gcu_calloc (
            size_t nitems,
            size_t size,
            const char * file,
            size_t line )
```

Wrapper for the standard calloc() function.

Parameters

nitems	The number of items to allocate.
size	The number of bytes in each item.
file	The name of the file from which the function was called.
line	The line number on which the function was called.

Returns

The beginning byte of the allocated memory.

3.3.1.2 gcu_free()

```
void gcu_free (
            void * pointer,
            const char * file,
            size_t line )
```

Wrapper for the standard free() function.

Parameters

pointer	The beginning byte of the currently allocated memory.
file	The name of the file from which the function was called.
line	The line number on which the function was called.

3.3.1.3 gcu_malloc()

```
void* gcu_malloc (
            size_t size,
            const char * file,
            size_t line )
```

Wrapper for the standard malloc() function.

Parameters

size	The number of bytes requested.
file	The name of the file from which the function was called.
line	The line number on which the function was called.

Returns

The beginning byte of the allocated memory.

3.3.1.4 gcu_realloc()

```
void* gcu_realloc (
            void * pointer,
            size_t size,
            const char * file,
            size_t line )
```

Wrapper for the standard realloc() function.

Parameters

pointer	The beginning byte of the currently allocated memory.
size	The newly requested size.
file	The name of the file from which the function was called.
line	The line number on which the function was called.

Returns

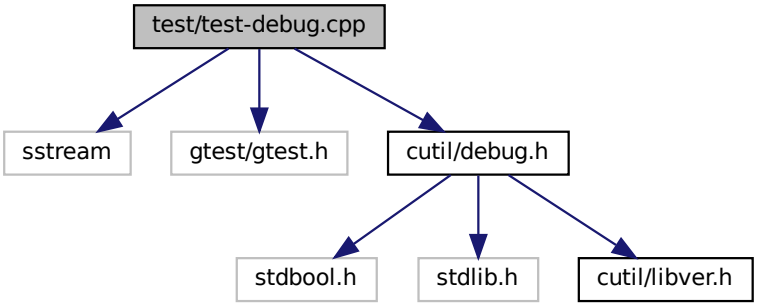
The beginning byte of the reallocated memory.

3.4 test/test-debug.cpp File Reference

Test the behavior of Ghoti.io Util HasParameters class.

```
#include <sstream>
#include <gtest/gtest.h>
#include "cutil/debug.h"
```

Include dependency graph for test-debug.cpp:



Functions

- **TEST** (Memory, MallocReallocFree)
- **TEST** (Memory, CallocFree)
- **TEST** (Memory, StopStartCapture)
- int **main** (int argc, char ∗∗argv)

3.4.1 Detailed Description

Test the behavior of Ghoti.io Util HasParameters class.

Index