

yabool

Generated by Doxygen 1.9.8

1 yabool	1
2 License	3
3 File Index	5
3.1 File List	5
4 File Documentation	7
4.1 LICENSE.md File Reference	7
4.2 README.md File Reference	7
4.3 yabool.h File Reference	7
4.3.1 Detailed Description	8
4.3.2 Macro Definition Documentation	8
4.3.2.1 YAFALSE	8
4.3.2.2 YATRUE	8
4.3.3 Typedef Documentation	9
4.3.3.1 yabool	9
4.3.4 Function Documentation	9
4.3.4.1 yabool_is_false()	9
4.3.4.2 yabool_is_true()	9
4.3.4.3 yabool_set_false()	10
4.3.4.4 yabool_set_true()	10
4.4 yabool.h	10
Index	13

Chapter 1

yabool

A simple header only bool library for the ya ecosystem.

Chapter 2

License

Copyright (c) 2026 asaeppe14231

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

[yabool.h](#)

Drop-in replacement of stdbool.h to make my programs and other ya- libraries ANSI C compatible [7](#)

Chapter 4

File Documentation

4.1 LICENSE.md File Reference

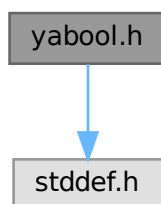
4.2 README.md File Reference

4.3 yabool.h File Reference

Drop-in replacement of stdbool.h to make my programs and other ya- libraries ANSI C compatible.

```
#include <stddef.h>
```

Include dependency graph for yabool.h:



Macros

- #define **YATRUE** 1
Boolean true value for yabool.
- #define **YAFALSE** 0
Boolean false value for yabool.

Typedefs

- typedef unsigned char `yabool`
< Just included for the NULL pointer

Functions

- static int `yabool_is_false` (`yabool` boolean)
Checks whether a yabool value is false.
- static int `yabool_is_true` (`yabool` boolean)
Checks whether a yabool value is true.
- static int `yabool_set_false` (`yabool` *boolean)
Sets a yabool variable to false.
- static int `yabool_set_true` (`yabool` *boolean)
Sets a yabool variable to true.

4.3.1 Detailed Description

Drop-in replacement of `stdbool.h` to make my programs and other ya- libraries ANSI C compatible.

Author

asaeppe14231

Version

0.1

This header defines the `yabool` type and related helper functions. It is designed to be minimal, portable, and usable in ANSI C environments where `stdbool.h` is not available.

4.3.2 Macro Definition Documentation

4.3.2.1 YAFALSE

```
#define YAFALSE 0
```

Boolean false value for yabool.

4.3.2.2 YATRUE

```
#define YATRUE 1
```

Boolean true value for yabool.

4.3.3 Typedef Documentation

4.3.3.1 yabool

yabool

< Just included for the NULL pointer

Boolean type (optionally) used by the ya-framework.

This is an unsigned char that represents boolean values in ANSI C. Use YATRUE and YAFALSE instead of 1 and 0 for clarity.

4.3.4 Function Documentation

4.3.4.1 yabool_is_false()

```
static int yabool_is_false (
    yabool boolean ) [inline], [static]
```

Checks whether a yabool value is false.

Parameters

<i>boolean</i>	The value to check.
----------------	---------------------

Returns

1 if false, 0 otherwise.

```
00041                                     {
00042     return boolean == YAFALSE;
00043 }
```

4.3.4.2 yabool_is_true()

```
static int yabool_is_true (
    yabool boolean ) [inline], [static]
```

Checks whether a yabool value is true.

Parameters

<i>boolean</i>	The value to check.
----------------	---------------------

Returns

1 if true, 0 otherwise.

```
00050                                     {
00051     return boolean == YATRUE;
00052 }
```

4.3.4.3 yabool_set_false()

```
static int yabool_set_false (
    yabool * boolean ) [inline], [static]
```

Sets a yabool variable to false.

Parameters

<i>boolean</i>	Pointer to the variable to modify.
----------------	------------------------------------

Returns

YATRUE on success, YAFALSE if the pointer is NULL.

```
00059                                     {
00060     if(boolean == NULL){
00061         return YAFALSE;
00062     }
00063     *boolean = YAFALSE;
00064     return YATRUE;
00065 }
```

4.3.4.4 yabool_set_true()

```
static int yabool_set_true (
    yabool * boolean ) [inline], [static]
```

Sets a yabool variable to true.

Parameters

<i>boolean</i>	Pointer to the variable to modify.
----------------	------------------------------------

Returns

YATRUE on success, YAFALSE if the pointer is NULL.

```
00073                                     {
00074     if(boolean == NULL){
00075         return YAFALSE;
00076     }
00077     *boolean = YATRUE;
00078     return YATRUE;
00079 }
```

4.4 yabool.h

[Go to the documentation of this file.](#)

```
00001
00010 #ifndef YABOOL_H
00011 #define YABOOL_H
00012
00013 #include <stddef.h>
00014
00022 typedef unsigned char yabool;
00023
00028 #define YATRUE 1
00029
```

```
00034 #define YAFALSE 0
00035
00041 static inline int yabool_is_false(yabool boolean){
00042     return boolean == YAFALSE;
00043 }
00044
00050 static inline int yabool_is_true(yabool boolean){
00051     return boolean == YATRUE;
00052 }
00053
00059 static inline int yabool_set_false(yabool* boolean){
00060     if(boolean == NULL){
00061         return YAFALSE;
00062     }
00063     *boolean = YAFALSE;
00064     return YATRUE;
00065 }
00066
00067
00073 static inline int yabool_set_true(yabool* boolean){
00074     if(boolean == NULL){
00075         return YAFALSE;
00076     }
00077     *boolean = YATRUE;
00078     return YATRUE;
00079 }
00080
00081 #endif
```


Index

License, [3](#)
LICENSE.md, [7](#)

README.md, [7](#)

yabool, [1](#)
 yabool.h, [9](#)
yabool.h, [7](#)
 yabool, [9](#)
 yabool_is_false, [9](#)
 yabool_is_true, [9](#)
 yabool_set_false, [9](#)
 yabool_set_true, [10](#)
 YAFALSE, [8](#)
 YATRUE, [8](#)
yabool_is_false
 yabool.h, [9](#)
yabool_is_true
 yabool.h, [9](#)
yabool_set_false
 yabool.h, [9](#)
yabool_set_true
 yabool.h, [10](#)
YAFALSE
 yabool.h, [8](#)
YATRUE
 yabool.h, [8](#)