/\* ==========================================

Unity Project - A Test Framework for C

Copyright (c) 2007-21 Mike Karlesky, Mark VanderVoord, Greg Williams

[Released under MIT License. Please refer to license.txt for details]

========================================== \*/

#ifndef UNITY\_INTERNALS\_H

#define UNITY\_INTERNALS\_H

#ifdef UNITY\_INCLUDE\_CONFIG\_H

#include "unity\_config.h"

#endif

#ifndef UNITY\_EXCLUDE\_SETJMP\_H

#include <setjmp.h>

#endif

#ifndef UNITY\_EXCLUDE\_MATH\_H

#include <math.h>

#endif

#ifndef UNITY\_EXCLUDE\_STDDEF\_H

#include <stddef.h>

#endif

#ifdef UNITY\_INCLUDE\_PRINT\_FORMATTED

#include <stdarg.h>

#endif

/\* Unity Attempts to Auto-Detect Integer Types

\* Attempt 1: UINT\_MAX, ULONG\_MAX in <limits.h>, or default to 32 bits

\* Attempt 2: UINTPTR\_MAX in <stdint.h>, or default to same size as long

\* The user may override any of these derived constants:

\* UNITY\_INT\_WIDTH, UNITY\_LONG\_WIDTH, UNITY\_POINTER\_WIDTH \*/

#ifndef UNITY\_EXCLUDE\_STDINT\_H

#include <stdint.h>

#endif

#ifndef UNITY\_EXCLUDE\_LIMITS\_H

#include <limits.h>

#endif

#if defined(\_\_GNUC\_\_) || defined(\_\_clang\_\_)

#define UNITY\_FUNCTION\_ATTR(a) \_\_attribute\_\_((a))

#else

#define UNITY\_FUNCTION\_ATTR(a) /\* ignore \*/

#endif

#ifndef UNITY\_NORETURN

#if defined(\_\_cplusplus)

#if \_\_cplusplus >= 201103L

#define UNITY\_NORETURN [[ noreturn ]]

#endif

#elif defined(\_\_STDC\_VERSION\_\_) && \_\_STDC\_VERSION\_\_ >= 201112L

#include <stdnoreturn.h>

#define UNITY\_NORETURN noreturn

#endif

#endif

#ifndef UNITY\_NORETURN

#define UNITY\_NORETURN UNITY\_FUNCTION\_ATTR(noreturn)

#endif

/\*-------------------------------------------------------

\* Guess Widths If Not Specified

\*-------------------------------------------------------\*/

/\* Determine the size of an int, if not already specified.

\* We cannot use sizeof(int), because it is not yet defined

\* at this stage in the translation of the C program.

\* Also sizeof(int) does return the size in addressable units on all platforms,

\* which may not necessarily be the size in bytes.

\* Therefore, infer it from UINT\_MAX if possible. \*/

#ifndef UNITY\_INT\_WIDTH

#ifdef UINT\_MAX

#if (UINT\_MAX == 0xFFFF)

#define UNITY\_INT\_WIDTH (16)

#elif (UINT\_MAX == 0xFFFFFFFF)

#define UNITY\_INT\_WIDTH (32)

#elif (UINT\_MAX == 0xFFFFFFFFFFFFFFFF)

#define UNITY\_INT\_WIDTH (64)

#endif

#else /\* Set to default \*/

#define UNITY\_INT\_WIDTH (32)

#endif /\* UINT\_MAX \*/

#endif

/\* Determine the size of a long, if not already specified. \*/

#ifndef UNITY\_LONG\_WIDTH

#ifdef ULONG\_MAX

#if (ULONG\_MAX == 0xFFFF)

#define UNITY\_LONG\_WIDTH (16)

#elif (ULONG\_MAX == 0xFFFFFFFF)

#define UNITY\_LONG\_WIDTH (32)

#elif (ULONG\_MAX == 0xFFFFFFFFFFFFFFFF)

#define UNITY\_LONG\_WIDTH (64)

#endif

#else /\* Set to default \*/

#define UNITY\_LONG\_WIDTH (32)

#endif /\* ULONG\_MAX \*/

#endif

/\* Determine the size of a pointer, if not already specified. \*/

#ifndef UNITY\_POINTER\_WIDTH

#ifdef UINTPTR\_MAX

#if (UINTPTR\_MAX <= 0xFFFF)

#define UNITY\_POINTER\_WIDTH (16)

#elif (UINTPTR\_MAX <= 0xFFFFFFFF)

#define UNITY\_POINTER\_WIDTH (32)

#elif (UINTPTR\_MAX <= 0xFFFFFFFFFFFFFFFF)

#define UNITY\_POINTER\_WIDTH (64)

#endif

#else /\* Set to default \*/

#define UNITY\_POINTER\_WIDTH UNITY\_LONG\_WIDTH

#endif /\* UINTPTR\_MAX \*/

#endif

/\*-------------------------------------------------------

\* Int Support (Define types based on detected sizes)

\*-------------------------------------------------------\*/

#if (UNITY\_INT\_WIDTH == 32)

typedef unsigned char UNITY\_UINT8;

typedef unsigned short UNITY\_UINT16;

typedef unsigned int UNITY\_UINT32;

typedef signed char UNITY\_INT8;

typedef signed short UNITY\_INT16;

typedef signed int UNITY\_INT32;

#elif (UNITY\_INT\_WIDTH == 16)

typedef unsigned char UNITY\_UINT8;

typedef unsigned int UNITY\_UINT16;

typedef unsigned long UNITY\_UINT32;

typedef signed char UNITY\_INT8;

typedef signed int UNITY\_INT16;

typedef signed long UNITY\_INT32;

#else

#error Invalid UNITY\_INT\_WIDTH specified! (16 or 32 are supported)

#endif

/\*-------------------------------------------------------

\* 64-bit Support

\*-------------------------------------------------------\*/

/\* Auto-detect 64 Bit Support \*/

#ifndef UNITY\_SUPPORT\_64

#if UNITY\_LONG\_WIDTH == 64 || UNITY\_POINTER\_WIDTH == 64

#define UNITY\_SUPPORT\_64

#endif

#endif

/\* 64-Bit Support Dependent Configuration \*/

#ifndef UNITY\_SUPPORT\_64

/\* No 64-bit Support \*/

typedef UNITY\_UINT32 UNITY\_UINT;

typedef UNITY\_INT32 UNITY\_INT;

#define UNITY\_MAX\_NIBBLES (8) /\* Maximum number of nibbles in a UNITY\_(U)INT \*/

#else

/\* 64-bit Support \*/

#if (UNITY\_LONG\_WIDTH == 32)

typedef unsigned long long UNITY\_UINT64;

typedef signed long long UNITY\_INT64;

#elif (UNITY\_LONG\_WIDTH == 64)

typedef unsigned long UNITY\_UINT64;

typedef signed long UNITY\_INT64;

#else

#error Invalid UNITY\_LONG\_WIDTH specified! (32 or 64 are supported)

#endif

typedef UNITY\_UINT64 UNITY\_UINT;

typedef UNITY\_INT64 UNITY\_INT;

#define UNITY\_MAX\_NIBBLES (16) /\* Maximum number of nibbles in a UNITY\_(U)INT \*/

#endif

/\*-------------------------------------------------------

\* Pointer Support

\*-------------------------------------------------------\*/

#if (UNITY\_POINTER\_WIDTH == 32)

#define UNITY\_PTR\_TO\_INT UNITY\_INT32

#define UNITY\_DISPLAY\_STYLE\_POINTER UNITY\_DISPLAY\_STYLE\_HEX32

#elif (UNITY\_POINTER\_WIDTH == 64)

#define UNITY\_PTR\_TO\_INT UNITY\_INT64

#define UNITY\_DISPLAY\_STYLE\_POINTER UNITY\_DISPLAY\_STYLE\_HEX64

#elif (UNITY\_POINTER\_WIDTH == 16)

#define UNITY\_PTR\_TO\_INT UNITY\_INT16

#define UNITY\_DISPLAY\_STYLE\_POINTER UNITY\_DISPLAY\_STYLE\_HEX16

#else

#error Invalid UNITY\_POINTER\_WIDTH specified! (16, 32 or 64 are supported)

#endif

#ifndef UNITY\_PTR\_ATTRIBUTE

#define UNITY\_PTR\_ATTRIBUTE

#endif

#ifndef UNITY\_INTERNAL\_PTR

#define UNITY\_INTERNAL\_PTR UNITY\_PTR\_ATTRIBUTE const void\*

#endif

/\*-------------------------------------------------------

\* Float Support

\*-------------------------------------------------------\*/

#ifdef UNITY\_EXCLUDE\_FLOAT

/\* No Floating Point Support \*/

#ifndef UNITY\_EXCLUDE\_DOUBLE

#define UNITY\_EXCLUDE\_DOUBLE /\* Remove double when excluding float support \*/

#endif

#ifndef UNITY\_EXCLUDE\_FLOAT\_PRINT

#define UNITY\_EXCLUDE\_FLOAT\_PRINT

#endif

#else

/\* Floating Point Support \*/

#ifndef UNITY\_FLOAT\_PRECISION

#define UNITY\_FLOAT\_PRECISION (0.00001f)

#endif

#ifndef UNITY\_FLOAT\_TYPE

#define UNITY\_FLOAT\_TYPE float

#endif

typedef UNITY\_FLOAT\_TYPE UNITY\_FLOAT;

/\* isinf & isnan macros should be provided by math.h \*/

#ifndef isinf

/\* The value of Inf - Inf is NaN \*/

#define isinf(n) (isnan((n) - (n)) && !isnan(n))

#endif

#ifndef isnan

/\* NaN is the only floating point value that does NOT equal itself.

\* Therefore if n != n, then it is NaN. \*/

#define isnan(n) ((n != n) ? 1 : 0)

#endif

#endif

/\*-------------------------------------------------------

\* Double Float Support

\*-------------------------------------------------------\*/

/\* unlike float, we DON'T include by default \*/

#if defined(UNITY\_EXCLUDE\_DOUBLE) || !defined(UNITY\_INCLUDE\_DOUBLE)

/\* No Floating Point Support \*/

#ifndef UNITY\_EXCLUDE\_DOUBLE

#define UNITY\_EXCLUDE\_DOUBLE

#else

#undef UNITY\_INCLUDE\_DOUBLE

#endif

#ifndef UNITY\_EXCLUDE\_FLOAT

#ifndef UNITY\_DOUBLE\_TYPE

#define UNITY\_DOUBLE\_TYPE double

#endif

typedef UNITY\_FLOAT UNITY\_DOUBLE;

/\* For parameter in UnityPrintFloat(UNITY\_DOUBLE), which aliases to double or float \*/

#endif

#else

/\* Double Floating Point Support \*/

#ifndef UNITY\_DOUBLE\_PRECISION

#define UNITY\_DOUBLE\_PRECISION (1e-12)

#endif

#ifndef UNITY\_DOUBLE\_TYPE

#define UNITY\_DOUBLE\_TYPE double

#endif

typedef UNITY\_DOUBLE\_TYPE UNITY\_DOUBLE;

#endif

/\*-------------------------------------------------------

\* Output Method: stdout (DEFAULT)

\*-------------------------------------------------------\*/

#ifndef UNITY\_OUTPUT\_CHAR

/\* Default to using putchar, which is defined in stdio.h \*/

#include <stdio.h>

#define UNITY\_OUTPUT\_CHAR(a) (void)putchar(a)

#else

/\* If defined as something else, make sure we declare it here so it's ready for use \*/

#ifdef UNITY\_OUTPUT\_CHAR\_HEADER\_DECLARATION

extern void UNITY\_OUTPUT\_CHAR\_HEADER\_DECLARATION;

#endif

#endif

#ifndef UNITY\_OUTPUT\_FLUSH

#ifdef UNITY\_USE\_FLUSH\_STDOUT

/\* We want to use the stdout flush utility \*/

#include <stdio.h>

#define UNITY\_OUTPUT\_FLUSH() (void)fflush(stdout)

#else

/\* We've specified nothing, therefore flush should just be ignored \*/

#define UNITY\_OUTPUT\_FLUSH()

#endif

#else

/\* If defined as something else, make sure we declare it here so it's ready for use \*/

#ifdef UNITY\_OUTPUT\_FLUSH\_HEADER\_DECLARATION

extern void UNITY\_OUTPUT\_FLUSH\_HEADER\_DECLARATION;

#endif

#endif

#ifndef UNITY\_OUTPUT\_FLUSH

#define UNITY\_FLUSH\_CALL()

#else

#define UNITY\_FLUSH\_CALL() UNITY\_OUTPUT\_FLUSH()

#endif

#ifndef UNITY\_PRINT\_EOL

#define UNITY\_PRINT\_EOL() UNITY\_OUTPUT\_CHAR('\n')

#endif

#ifndef UNITY\_OUTPUT\_START

#define UNITY\_OUTPUT\_START()

#endif

#ifndef UNITY\_OUTPUT\_COMPLETE

#define UNITY\_OUTPUT\_COMPLETE()

#endif

#ifdef UNITY\_INCLUDE\_EXEC\_TIME

#if !defined(UNITY\_EXEC\_TIME\_START) && \

!defined(UNITY\_EXEC\_TIME\_STOP) && \

!defined(UNITY\_PRINT\_EXEC\_TIME) && \

!defined(UNITY\_TIME\_TYPE)

/\* If none any of these macros are defined then try to provide a default implementation \*/

#if defined(UNITY\_CLOCK\_MS)

/\* This is a simple way to get a default implementation on platforms that support getting a millisecond counter \*/

#define UNITY\_TIME\_TYPE UNITY\_UINT

#define UNITY\_EXEC\_TIME\_START() Unity.CurrentTestStartTime = UNITY\_CLOCK\_MS()

#define UNITY\_EXEC\_TIME\_STOP() Unity.CurrentTestStopTime = UNITY\_CLOCK\_MS()

#define UNITY\_PRINT\_EXEC\_TIME() { \

UNITY\_UINT execTimeMs = (Unity.CurrentTestStopTime - Unity.CurrentTestStartTime); \

UnityPrint(" ("); \

UnityPrintNumberUnsigned(execTimeMs); \

UnityPrint(" ms)"); \

}

#elif defined(\_WIN32)

#include <time.h>

#define UNITY\_TIME\_TYPE clock\_t

#define UNITY\_GET\_TIME(t) t = (clock\_t)((clock() \* 1000) / CLOCKS\_PER\_SEC)

#define UNITY\_EXEC\_TIME\_START() UNITY\_GET\_TIME(Unity.CurrentTestStartTime)

#define UNITY\_EXEC\_TIME\_STOP() UNITY\_GET\_TIME(Unity.CurrentTestStopTime)

#define UNITY\_PRINT\_EXEC\_TIME() { \

UNITY\_UINT execTimeMs = (Unity.CurrentTestStopTime - Unity.CurrentTestStartTime); \

UnityPrint(" ("); \

UnityPrintNumberUnsigned(execTimeMs); \

UnityPrint(" ms)"); \

}

#elif defined(\_\_unix\_\_) || defined(\_\_APPLE\_\_)

#include <time.h>

#define UNITY\_TIME\_TYPE struct timespec

#define UNITY\_GET\_TIME(t) clock\_gettime(CLOCK\_MONOTONIC, &t)

#define UNITY\_EXEC\_TIME\_START() UNITY\_GET\_TIME(Unity.CurrentTestStartTime)

#define UNITY\_EXEC\_TIME\_STOP() UNITY\_GET\_TIME(Unity.CurrentTestStopTime)

#define UNITY\_PRINT\_EXEC\_TIME() { \

UNITY\_UINT execTimeMs = ((Unity.CurrentTestStopTime.tv\_sec - Unity.CurrentTestStartTime.tv\_sec) \* 1000L); \

execTimeMs += ((Unity.CurrentTestStopTime.tv\_nsec - Unity.CurrentTestStartTime.tv\_nsec) / 1000000L); \

UnityPrint(" ("); \

UnityPrintNumberUnsigned(execTimeMs); \

UnityPrint(" ms)"); \

}

#endif

#endif

#endif

#ifndef UNITY\_EXEC\_TIME\_START

#define UNITY\_EXEC\_TIME\_START() do{}while(0)

#endif

#ifndef UNITY\_EXEC\_TIME\_STOP

#define UNITY\_EXEC\_TIME\_STOP() do{}while(0)

#endif

#ifndef UNITY\_TIME\_TYPE

#define UNITY\_TIME\_TYPE UNITY\_UINT

#endif

#ifndef UNITY\_PRINT\_EXEC\_TIME

#define UNITY\_PRINT\_EXEC\_TIME() do{}while(0)

#endif

/\*-------------------------------------------------------

\* Footprint

\*-------------------------------------------------------\*/

#ifndef UNITY\_LINE\_TYPE

#define UNITY\_LINE\_TYPE UNITY\_UINT

#endif

#ifndef UNITY\_COUNTER\_TYPE

#define UNITY\_COUNTER\_TYPE UNITY\_UINT

#endif

/\*-------------------------------------------------------

\* Internal Structs Needed

\*-------------------------------------------------------\*/

typedef void (\*UnityTestFunction)(void);

#define UNITY\_DISPLAY\_RANGE\_INT (0x10)

#define UNITY\_DISPLAY\_RANGE\_UINT (0x20)

#define UNITY\_DISPLAY\_RANGE\_HEX (0x40)

#define UNITY\_DISPLAY\_RANGE\_CHAR (0x80)

typedef enum

{

UNITY\_DISPLAY\_STYLE\_INT = (UNITY\_INT\_WIDTH / 8) + UNITY\_DISPLAY\_RANGE\_INT,

UNITY\_DISPLAY\_STYLE\_INT8 = 1 + UNITY\_DISPLAY\_RANGE\_INT,

UNITY\_DISPLAY\_STYLE\_INT16 = 2 + UNITY\_DISPLAY\_RANGE\_INT,

UNITY\_DISPLAY\_STYLE\_INT32 = 4 + UNITY\_DISPLAY\_RANGE\_INT,

#ifdef UNITY\_SUPPORT\_64

UNITY\_DISPLAY\_STYLE\_INT64 = 8 + UNITY\_DISPLAY\_RANGE\_INT,

#endif

UNITY\_DISPLAY\_STYLE\_UINT = (UNITY\_INT\_WIDTH / 8) + UNITY\_DISPLAY\_RANGE\_UINT,

UNITY\_DISPLAY\_STYLE\_UINT8 = 1 + UNITY\_DISPLAY\_RANGE\_UINT,

UNITY\_DISPLAY\_STYLE\_UINT16 = 2 + UNITY\_DISPLAY\_RANGE\_UINT,

UNITY\_DISPLAY\_STYLE\_UINT32 = 4 + UNITY\_DISPLAY\_RANGE\_UINT,

#ifdef UNITY\_SUPPORT\_64

UNITY\_DISPLAY\_STYLE\_UINT64 = 8 + UNITY\_DISPLAY\_RANGE\_UINT,

#endif

UNITY\_DISPLAY\_STYLE\_HEX8 = 1 + UNITY\_DISPLAY\_RANGE\_HEX,

UNITY\_DISPLAY\_STYLE\_HEX16 = 2 + UNITY\_DISPLAY\_RANGE\_HEX,

UNITY\_DISPLAY\_STYLE\_HEX32 = 4 + UNITY\_DISPLAY\_RANGE\_HEX,

#ifdef UNITY\_SUPPORT\_64

UNITY\_DISPLAY\_STYLE\_HEX64 = 8 + UNITY\_DISPLAY\_RANGE\_HEX,

#endif

UNITY\_DISPLAY\_STYLE\_CHAR = 1 + UNITY\_DISPLAY\_RANGE\_CHAR + UNITY\_DISPLAY\_RANGE\_INT,

UNITY\_DISPLAY\_STYLE\_UNKNOWN

} UNITY\_DISPLAY\_STYLE\_T;

typedef enum

{

UNITY\_WITHIN = 0x0,

UNITY\_EQUAL\_TO = 0x1,

UNITY\_GREATER\_THAN = 0x2,

UNITY\_GREATER\_OR\_EQUAL = 0x2 + UNITY\_EQUAL\_TO,

UNITY\_SMALLER\_THAN = 0x4,

UNITY\_SMALLER\_OR\_EQUAL = 0x4 + UNITY\_EQUAL\_TO,

UNITY\_NOT\_EQUAL = 0x0,

UNITY\_UNKNOWN

} UNITY\_COMPARISON\_T;

#ifndef UNITY\_EXCLUDE\_FLOAT

typedef enum UNITY\_FLOAT\_TRAIT

{

UNITY\_FLOAT\_IS\_NOT\_INF = 0,

UNITY\_FLOAT\_IS\_INF,

UNITY\_FLOAT\_IS\_NOT\_NEG\_INF,

UNITY\_FLOAT\_IS\_NEG\_INF,

UNITY\_FLOAT\_IS\_NOT\_NAN,

UNITY\_FLOAT\_IS\_NAN,

UNITY\_FLOAT\_IS\_NOT\_DET,

UNITY\_FLOAT\_IS\_DET,

UNITY\_FLOAT\_INVALID\_TRAIT

} UNITY\_FLOAT\_TRAIT\_T;

#endif

typedef enum

{

UNITY\_ARRAY\_TO\_VAL = 0,

UNITY\_ARRAY\_TO\_ARRAY,

UNITY\_ARRAY\_UNKNOWN

} UNITY\_FLAGS\_T;

struct UNITY\_STORAGE\_T

{

const char\* TestFile;

const char\* CurrentTestName;

#ifndef UNITY\_EXCLUDE\_DETAILS

const char\* CurrentDetail1;

const char\* CurrentDetail2;

#endif

UNITY\_LINE\_TYPE CurrentTestLineNumber;

UNITY\_COUNTER\_TYPE NumberOfTests;

UNITY\_COUNTER\_TYPE TestFailures;

UNITY\_COUNTER\_TYPE TestIgnores;

UNITY\_COUNTER\_TYPE CurrentTestFailed;

UNITY\_COUNTER\_TYPE CurrentTestIgnored;

#ifdef UNITY\_INCLUDE\_EXEC\_TIME

UNITY\_TIME\_TYPE CurrentTestStartTime;

UNITY\_TIME\_TYPE CurrentTestStopTime;

#endif

#ifndef UNITY\_EXCLUDE\_SETJMP\_H

jmp\_buf AbortFrame;

#endif

};

extern struct UNITY\_STORAGE\_T Unity;

/\*-------------------------------------------------------

\* Test Suite Management

\*-------------------------------------------------------\*/

void UnityBegin(const char\* filename);

int UnityEnd(void);

void UnitySetTestFile(const char\* filename);

void UnityConcludeTest(void);

#ifndef RUN\_TEST

void UnityDefaultTestRun(UnityTestFunction Func, const char\* FuncName, const int FuncLineNum);

#else

#define UNITY\_SKIP\_DEFAULT\_RUNNER

#endif

/\*-------------------------------------------------------

\* Details Support

\*-------------------------------------------------------\*/

#ifdef UNITY\_EXCLUDE\_DETAILS

#define UNITY\_CLR\_DETAILS()

#define UNITY\_SET\_DETAIL(d1)

#define UNITY\_SET\_DETAILS(d1,d2)

#else

#define UNITY\_CLR\_DETAILS() { Unity.CurrentDetail1 = 0; Unity.CurrentDetail2 = 0; }

#define UNITY\_SET\_DETAIL(d1) { Unity.CurrentDetail1 = (d1); Unity.CurrentDetail2 = 0; }

#define UNITY\_SET\_DETAILS(d1,d2) { Unity.CurrentDetail1 = (d1); Unity.CurrentDetail2 = (d2); }

#ifndef UNITY\_DETAIL1\_NAME

#define UNITY\_DETAIL1\_NAME "Function"

#endif

#ifndef UNITY\_DETAIL2\_NAME

#define UNITY\_DETAIL2\_NAME "Argument"

#endif

#endif

#ifdef UNITY\_PRINT\_TEST\_CONTEXT

void UNITY\_PRINT\_TEST\_CONTEXT(void);

#endif

/\*-------------------------------------------------------

\* Test Output

\*-------------------------------------------------------\*/

void UnityPrint(const char\* string);

#ifdef UNITY\_INCLUDE\_PRINT\_FORMATTED

void UnityPrintF(const UNITY\_LINE\_TYPE line, const char\* format, ...);

#endif

void UnityPrintLen(const char\* string, const UNITY\_UINT32 length);

void UnityPrintMask(const UNITY\_UINT mask, const UNITY\_UINT number);

void UnityPrintNumberByStyle(const UNITY\_INT number, const UNITY\_DISPLAY\_STYLE\_T style);

void UnityPrintNumber(const UNITY\_INT number\_to\_print);

void UnityPrintNumberUnsigned(const UNITY\_UINT number);

void UnityPrintNumberHex(const UNITY\_UINT number, const char nibbles\_to\_print);

#ifndef UNITY\_EXCLUDE\_FLOAT\_PRINT

void UnityPrintFloat(const UNITY\_DOUBLE input\_number);

#endif

/\*-------------------------------------------------------

\* Test Assertion Functions

\*-------------------------------------------------------

\* Use the macros below this section instead of calling

\* these directly. The macros have a consistent naming

\* convention and will pull in file and line information

\* for you. \*/

void UnityAssertEqualNumber(const UNITY\_INT expected,

const UNITY\_INT actual,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_DISPLAY\_STYLE\_T style);

void UnityAssertGreaterOrLessOrEqualNumber(const UNITY\_INT threshold,

const UNITY\_INT actual,

const UNITY\_COMPARISON\_T compare,

const char \*msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_DISPLAY\_STYLE\_T style);

void UnityAssertEqualIntArray(UNITY\_INTERNAL\_PTR expected,

UNITY\_INTERNAL\_PTR actual,

const UNITY\_UINT32 num\_elements,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_DISPLAY\_STYLE\_T style,

const UNITY\_FLAGS\_T flags);

void UnityAssertBits(const UNITY\_INT mask,

const UNITY\_INT expected,

const UNITY\_INT actual,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber);

void UnityAssertEqualString(const char\* expected,

const char\* actual,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber);

void UnityAssertEqualStringLen(const char\* expected,

const char\* actual,

const UNITY\_UINT32 length,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber);

void UnityAssertEqualStringArray( UNITY\_INTERNAL\_PTR expected,

const char\*\* actual,

const UNITY\_UINT32 num\_elements,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_FLAGS\_T flags);

void UnityAssertEqualMemory( UNITY\_INTERNAL\_PTR expected,

UNITY\_INTERNAL\_PTR actual,

const UNITY\_UINT32 length,

const UNITY\_UINT32 num\_elements,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_FLAGS\_T flags);

void UnityAssertNumbersWithin(const UNITY\_UINT delta,

const UNITY\_INT expected,

const UNITY\_INT actual,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_DISPLAY\_STYLE\_T style);

void UnityAssertNumbersArrayWithin(const UNITY\_UINT delta,

UNITY\_INTERNAL\_PTR expected,

UNITY\_INTERNAL\_PTR actual,

const UNITY\_UINT32 num\_elements,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_DISPLAY\_STYLE\_T style,

const UNITY\_FLAGS\_T flags);

#ifndef UNITY\_EXCLUDE\_SETJMP\_H

UNITY\_NORETURN void UnityFail(const char\* message, const UNITY\_LINE\_TYPE line);

UNITY\_NORETURN void UnityIgnore(const char\* message, const UNITY\_LINE\_TYPE line);

#else

void UnityFail(const char\* message, const UNITY\_LINE\_TYPE line);

void UnityIgnore(const char\* message, const UNITY\_LINE\_TYPE line);

#endif

void UnityMessage(const char\* message, const UNITY\_LINE\_TYPE line);

#ifndef UNITY\_EXCLUDE\_FLOAT

void UnityAssertFloatsWithin(const UNITY\_FLOAT delta,

const UNITY\_FLOAT expected,

const UNITY\_FLOAT actual,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber);

void UnityAssertEqualFloatArray(UNITY\_PTR\_ATTRIBUTE const UNITY\_FLOAT\* expected,

UNITY\_PTR\_ATTRIBUTE const UNITY\_FLOAT\* actual,

const UNITY\_UINT32 num\_elements,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_FLAGS\_T flags);

void UnityAssertFloatSpecial(const UNITY\_FLOAT actual,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_FLOAT\_TRAIT\_T style);

#endif

#ifndef UNITY\_EXCLUDE\_DOUBLE

void UnityAssertDoublesWithin(const UNITY\_DOUBLE delta,

const UNITY\_DOUBLE expected,

const UNITY\_DOUBLE actual,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber);

void UnityAssertEqualDoubleArray(UNITY\_PTR\_ATTRIBUTE const UNITY\_DOUBLE\* expected,

UNITY\_PTR\_ATTRIBUTE const UNITY\_DOUBLE\* actual,

const UNITY\_UINT32 num\_elements,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_FLAGS\_T flags);

void UnityAssertDoubleSpecial(const UNITY\_DOUBLE actual,

const char\* msg,

const UNITY\_LINE\_TYPE lineNumber,

const UNITY\_FLOAT\_TRAIT\_T style);

#endif

/\*-------------------------------------------------------

\* Helpers

\*-------------------------------------------------------\*/

UNITY\_INTERNAL\_PTR UnityNumToPtr(const UNITY\_INT num, const UNITY\_UINT8 size);

#ifndef UNITY\_EXCLUDE\_FLOAT

UNITY\_INTERNAL\_PTR UnityFloatToPtr(const float num);

#endif

#ifndef UNITY\_EXCLUDE\_DOUBLE

UNITY\_INTERNAL\_PTR UnityDoubleToPtr(const double num);

#endif

/\*-------------------------------------------------------

\* Error Strings We Might Need

\*-------------------------------------------------------\*/

extern const char UnityStrOk[];

extern const char UnityStrPass[];

extern const char UnityStrFail[];

extern const char UnityStrIgnore[];

extern const char UnityStrErrFloat[];

extern const char UnityStrErrDouble[];

extern const char UnityStrErr64[];

extern const char UnityStrErrShorthand[];

/\*-------------------------------------------------------

\* Test Running Macros

\*-------------------------------------------------------\*/

#ifndef UNITY\_EXCLUDE\_SETJMP\_H

#define TEST\_PROTECT() (setjmp(Unity.AbortFrame) == 0)

#define TEST\_ABORT() longjmp(Unity.AbortFrame, 1)

#else

#define TEST\_PROTECT() 1

#define TEST\_ABORT() return

#endif

/\* This tricky series of macros gives us an optional line argument to treat it as RUN\_TEST(func, num=\_\_LINE\_\_) \*/

#ifndef RUN\_TEST

#ifdef \_\_STDC\_VERSION\_\_

#if \_\_STDC\_VERSION\_\_ >= 199901L

#define UNITY\_SUPPORT\_VARIADIC\_MACROS

#endif

#endif

#ifdef UNITY\_SUPPORT\_VARIADIC\_MACROS

#define RUN\_TEST(...) RUN\_TEST\_AT\_LINE(\_\_VA\_ARGS\_\_, \_\_LINE\_\_, throwaway)

#define RUN\_TEST\_AT\_LINE(func, line, ...) UnityDefaultTestRun(func, #func, line)

#endif

#endif

/\* If we can't do the tricky version, we'll just have to require them to always include the line number \*/

#ifndef RUN\_TEST

#ifdef CMOCK

#define RUN\_TEST(func, num) UnityDefaultTestRun(func, #func, num)

#else

#define RUN\_TEST(func) UnityDefaultTestRun(func, #func, \_\_LINE\_\_)

#endif

#endif

#define TEST\_LINE\_NUM (Unity.CurrentTestLineNumber)

#define TEST\_IS\_IGNORED (Unity.CurrentTestIgnored)

#define UNITY\_NEW\_TEST(a) \

Unity.CurrentTestName = (a); \

Unity.CurrentTestLineNumber = (UNITY\_LINE\_TYPE)(\_\_LINE\_\_); \

Unity.NumberOfTests++;

#ifndef UNITY\_BEGIN

#define UNITY\_BEGIN() UnityBegin(\_\_FILE\_\_)

#endif

#ifndef UNITY\_END

#define UNITY\_END() UnityEnd()

#endif

#ifndef UNITY\_SHORTHAND\_AS\_INT

#ifndef UNITY\_SHORTHAND\_AS\_MEM

#ifndef UNITY\_SHORTHAND\_AS\_NONE

#ifndef UNITY\_SHORTHAND\_AS\_RAW

#define UNITY\_SHORTHAND\_AS\_OLD

#endif

#endif

#endif

#endif

/\*-----------------------------------------------

\* Command Line Argument Support

\*-----------------------------------------------\*/

#ifdef UNITY\_USE\_COMMAND\_LINE\_ARGS

int UnityParseOptions(int argc, char\*\* argv);

int UnityTestMatches(void);

#endif

/\*-------------------------------------------------------

\* Basic Fail and Ignore

\*-------------------------------------------------------\*/

#define UNITY\_TEST\_FAIL(line, message) UnityFail( (message), (UNITY\_LINE\_TYPE)(line))

#define UNITY\_TEST\_IGNORE(line, message) UnityIgnore( (message), (UNITY\_LINE\_TYPE)(line))

/\*-------------------------------------------------------

\* Test Asserts

\*-------------------------------------------------------\*/

#define UNITY\_TEST\_ASSERT(condition, line, message) do {if (condition) {} else {UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), (message));}} while(0)

#define UNITY\_TEST\_ASSERT\_NULL(pointer, line, message) UNITY\_TEST\_ASSERT(((pointer) == NULL), (UNITY\_LINE\_TYPE)(line), (message))

#define UNITY\_TEST\_ASSERT\_NOT\_NULL(pointer, line, message) UNITY\_TEST\_ASSERT(((pointer) != NULL), (UNITY\_LINE\_TYPE)(line), (message))

#define UNITY\_TEST\_ASSERT\_EMPTY(pointer, line, message) UNITY\_TEST\_ASSERT(((pointer[0]) == 0), (UNITY\_LINE\_TYPE)(line), (message))

#define UNITY\_TEST\_ASSERT\_NOT\_EMPTY(pointer, line, message) UNITY\_TEST\_ASSERT(((pointer[0]) != 0), (UNITY\_LINE\_TYPE)(line), (message))

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT8(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_INT8 )(expected), (UNITY\_INT)(UNITY\_INT8 )(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT16(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_INT16)(expected), (UNITY\_INT)(UNITY\_INT16)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT32(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_INT32)(expected), (UNITY\_INT)(UNITY\_INT32)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT8(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(expected), (UNITY\_INT)(UNITY\_UINT8 )(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT16(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_UINT16)(expected), (UNITY\_INT)(UNITY\_UINT16)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT32(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_UINT32)(expected), (UNITY\_INT)(UNITY\_UINT32)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX8(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_INT8 )(expected), (UNITY\_INT)(UNITY\_INT8 )(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX16(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_INT16)(expected), (UNITY\_INT)(UNITY\_INT16)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX32(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_INT32)(expected), (UNITY\_INT)(UNITY\_INT32)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32)

#define UNITY\_TEST\_ASSERT\_EQUAL\_CHAR(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(UNITY\_INT8 )(expected), (UNITY\_INT)(UNITY\_INT8 )(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR)

#define UNITY\_TEST\_ASSERT\_BITS(mask, expected, actual, line, message) UnityAssertBits((UNITY\_INT)(mask), (UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line))

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_INT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_INT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 )(threshold), (UNITY\_INT)(UNITY\_INT8 )(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_INT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT16)(threshold), (UNITY\_INT)(UNITY\_INT16)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_INT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT32)(threshold), (UNITY\_INT)(UNITY\_INT32)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_UINT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_UINT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_UINT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_UINT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_HEX8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_HEX16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_HEX32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_CHAR(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 )(threshold), (UNITY\_INT)(UNITY\_INT8 )(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_INT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_INT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 )(threshold), (UNITY\_INT)(UNITY\_INT8 )(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_INT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT16)(threshold), (UNITY\_INT)(UNITY\_INT16)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_INT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT32)(threshold), (UNITY\_INT)(UNITY\_INT32)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_UINT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_UINT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_UINT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_UINT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_HEX8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_HEX16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_HEX32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_CHAR(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 )(threshold), (UNITY\_INT)(UNITY\_INT8 )(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_INT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_INT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 )(threshold), (UNITY\_INT)(UNITY\_INT8 )(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_INT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT16)(threshold), (UNITY\_INT)(UNITY\_INT16)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_INT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT32)(threshold), (UNITY\_INT)(UNITY\_INT32)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_UINT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_UINT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_UINT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_UINT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_HEX8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_HEX16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_HEX32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_CHAR(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 )(threshold), (UNITY\_INT)(UNITY\_INT8 )(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_INT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT) (threshold), (UNITY\_INT) (actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_INT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 ) (threshold), (UNITY\_INT)(UNITY\_INT8 ) (actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_INT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT16) (threshold), (UNITY\_INT)(UNITY\_INT16) (actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_INT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT32) (threshold), (UNITY\_INT)(UNITY\_INT32) (actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_UINT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT) (threshold), (UNITY\_INT) (actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_UINT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_UINT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_UINT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_HEX8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_HEX16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_HEX32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_CHAR(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 ) (threshold), (UNITY\_INT)(UNITY\_INT8 ) (actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_INT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT) (threshold), (UNITY\_INT) (actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_INT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 )(threshold), (UNITY\_INT)(UNITY\_INT8 ) (actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_INT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT16)(threshold), (UNITY\_INT)(UNITY\_INT16) (actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_INT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT32)(threshold), (UNITY\_INT)(UNITY\_INT32) (actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_UINT(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT) (threshold), (UNITY\_INT) (actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_UINT8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_UINT16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_UINT32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_HEX8(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT8 )(threshold), (UNITY\_INT)(UNITY\_UINT8 )(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_HEX16(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT16)(threshold), (UNITY\_INT)(UNITY\_UINT16)(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_HEX32(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_UINT32)(threshold), (UNITY\_INT)(UNITY\_UINT32)(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_CHAR(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(UNITY\_INT8 )(threshold), (UNITY\_INT)(UNITY\_INT8 ) (actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR)

#define UNITY\_TEST\_ASSERT\_INT\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin( (delta), (UNITY\_INT) (expected), (UNITY\_INT) (actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT)

#define UNITY\_TEST\_ASSERT\_INT8\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT8 )(delta), (UNITY\_INT)(UNITY\_INT8 ) (expected), (UNITY\_INT)(UNITY\_INT8 ) (actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8)

#define UNITY\_TEST\_ASSERT\_INT16\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT16)(delta), (UNITY\_INT)(UNITY\_INT16) (expected), (UNITY\_INT)(UNITY\_INT16) (actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16)

#define UNITY\_TEST\_ASSERT\_INT32\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT32)(delta), (UNITY\_INT)(UNITY\_INT32) (expected), (UNITY\_INT)(UNITY\_INT32) (actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32)

#define UNITY\_TEST\_ASSERT\_UINT\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin( (delta), (UNITY\_INT) (expected), (UNITY\_INT) (actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT)

#define UNITY\_TEST\_ASSERT\_UINT8\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT8 )(delta), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT8 )(expected), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT8 )(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8)

#define UNITY\_TEST\_ASSERT\_UINT16\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT16)(delta), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT16)(expected), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT16)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16)

#define UNITY\_TEST\_ASSERT\_UINT32\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT32)(delta), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT32)(expected), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT32)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32)

#define UNITY\_TEST\_ASSERT\_HEX8\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT8 )(delta), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT8 )(expected), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT8 )(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8)

#define UNITY\_TEST\_ASSERT\_HEX16\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT16)(delta), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT16)(expected), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT16)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16)

#define UNITY\_TEST\_ASSERT\_HEX32\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT32)(delta), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT32)(expected), (UNITY\_INT)(UNITY\_UINT)(UNITY\_UINT32)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32)

#define UNITY\_TEST\_ASSERT\_CHAR\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((UNITY\_UINT8 )(delta), (UNITY\_INT)(UNITY\_INT8 ) (expected), (UNITY\_INT)(UNITY\_INT8 ) (actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR)

#define UNITY\_TEST\_ASSERT\_INT\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin( (delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_INT8\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT8 )(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_INT16\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT16)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_INT32\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT32)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_UINT\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin( (delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_UINT8\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT16)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_UINT16\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT16)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_UINT32\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT32)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_HEX8\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT8 )(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_HEX16\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT16)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_HEX32\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT32)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_CHAR\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT8 )(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), ((UNITY\_UINT32)(num\_elements)), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_PTR(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_PTR\_TO\_INT)(expected), (UNITY\_PTR\_TO\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_POINTER)

#define UNITY\_TEST\_ASSERT\_EQUAL\_STRING(expected, actual, line, message) UnityAssertEqualString((const char\*)(expected), (const char\*)(actual), (message), (UNITY\_LINE\_TYPE)(line))

#define UNITY\_TEST\_ASSERT\_EQUAL\_STRING\_LEN(expected, actual, len, line, message) UnityAssertEqualStringLen((const char\*)(expected), (const char\*)(actual), (UNITY\_UINT32)(len), (message), (UNITY\_LINE\_TYPE)(line))

#define UNITY\_TEST\_ASSERT\_EQUAL\_MEMORY(expected, actual, len, line, message) UnityAssertEqualMemory((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(len), 1, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT8\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT16\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT32\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT8\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT16\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT32\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX8\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX16\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX32\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_PTR\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_POINTER, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_STRING\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualStringArray((UNITY\_INTERNAL\_PTR)(expected), (const char\*\*)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_MEMORY\_ARRAY(expected, actual, len, num\_elements, line, message) UnityAssertEqualMemory((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(len), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_CHAR\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_INT(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT) (expected), (UNITY\_INT\_WIDTH / 8)), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_INT8(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT8 )(expected), 1), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT8, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_INT16(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT16 )(expected), 2), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT16, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_INT32(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT32 )(expected), 4), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT32, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_UINT(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT) (expected), (UNITY\_INT\_WIDTH / 8)), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_UINT8(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_UINT8 )(expected), 1), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT8, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_UINT16(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_UINT16)(expected), 2), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT16, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_UINT32(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_UINT32)(expected), 4), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT32, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_HEX8(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT8 )(expected), 1), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX8, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_HEX16(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT16 )(expected), 2), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX16, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_HEX32(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT32 )(expected), 4), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX32, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_PTR(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_PTR\_TO\_INT) (expected), (UNITY\_POINTER\_WIDTH / 8)), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_POINTER, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_STRING(expected, actual, num\_elements, line, message) UnityAssertEqualStringArray((UNITY\_INTERNAL\_PTR)(expected), (const char\*\*)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_MEMORY(expected, actual, len, num\_elements, line, message) UnityAssertEqualMemory((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(len), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_CHAR(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT8 )(expected), 1), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_CHAR, UNITY\_ARRAY\_TO\_VAL)

#ifdef UNITY\_SUPPORT\_64

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT64(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT64(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX64(expected, actual, line, message) UnityAssertEqualNumber((UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT64\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT64\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX64\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray((UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_INT64(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT64)(expected), 8), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_UINT64(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_UINT64)(expected), 8), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_HEX64(expected, actual, num\_elements, line, message) UnityAssertEqualIntArray(UnityNumToPtr((UNITY\_INT)(UNITY\_INT64)(expected), 8), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64, UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_INT64\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((delta), (UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64)

#define UNITY\_TEST\_ASSERT\_UINT64\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((delta), (UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64)

#define UNITY\_TEST\_ASSERT\_HEX64\_WITHIN(delta, expected, actual, line, message) UnityAssertNumbersWithin((delta), (UNITY\_INT)(expected), (UNITY\_INT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_INT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_UINT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64)

#define UNITY\_TEST\_ASSERT\_NOT\_EQUAL\_HEX64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_NOT\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_INT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_UINT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_HEX64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_GREATER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_INT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_UINT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_HEX64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_GREATER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_INT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_UINT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_HEX64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_SMALLER\_THAN, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_INT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_UINT64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_HEX64(threshold, actual, line, message) UnityAssertGreaterOrLessOrEqualNumber((UNITY\_INT)(threshold), (UNITY\_INT)(actual), UNITY\_SMALLER\_OR\_EQUAL, (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64)

#define UNITY\_TEST\_ASSERT\_INT64\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT64)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_INT64, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_UINT64\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT64)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_UINT64, UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_HEX64\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UnityAssertNumbersArrayWithin((UNITY\_UINT64)(delta), (UNITY\_INTERNAL\_PTR)(expected), (UNITY\_INTERNAL\_PTR)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_DISPLAY\_STYLE\_HEX64, UNITY\_ARRAY\_TO\_ARRAY)

#else

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT64(expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT64(expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX64(expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_EQUAL\_INT64\_ARRAY(expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_EQUAL\_UINT64\_ARRAY(expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_EQUAL\_HEX64\_ARRAY(expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_INT64\_WITHIN(delta, expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_UINT64\_WITHIN(delta, expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_HEX64\_WITHIN(delta, expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_INT64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_UINT64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_GREATER\_THAN\_HEX64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_INT64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_UINT64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_GREATER\_OR\_EQUAL\_HEX64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_INT64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_UINT64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_THAN\_HEX64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_INT64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_UINT64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_SMALLER\_OR\_EQUAL\_HEX64(threshold, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_INT64\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_UINT64\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#define UNITY\_TEST\_ASSERT\_HEX64\_ARRAY\_WITHIN(delta, expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErr64)

#endif

#ifdef UNITY\_EXCLUDE\_FLOAT

#define UNITY\_TEST\_ASSERT\_FLOAT\_WITHIN(delta, expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_EQUAL\_FLOAT(expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_EQUAL\_FLOAT\_ARRAY(expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_FLOAT(expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_INF(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NEG\_INF(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NAN(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_DETERMINATE(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NOT\_INF(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NOT\_NEG\_INF(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NOT\_NAN(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NOT\_DETERMINATE(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrFloat)

#else

#define UNITY\_TEST\_ASSERT\_FLOAT\_WITHIN(delta, expected, actual, line, message) UnityAssertFloatsWithin((UNITY\_FLOAT)(delta), (UNITY\_FLOAT)(expected), (UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line))

#define UNITY\_TEST\_ASSERT\_EQUAL\_FLOAT(expected, actual, line, message) UNITY\_TEST\_ASSERT\_FLOAT\_WITHIN((UNITY\_FLOAT)(expected) \* (UNITY\_FLOAT)UNITY\_FLOAT\_PRECISION, (UNITY\_FLOAT)(expected), (UNITY\_FLOAT)(actual), (UNITY\_LINE\_TYPE)(line), (message))

#define UNITY\_TEST\_ASSERT\_EQUAL\_FLOAT\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualFloatArray((UNITY\_FLOAT\*)(expected), (UNITY\_FLOAT\*)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_FLOAT(expected, actual, num\_elements, line, message) UnityAssertEqualFloatArray(UnityFloatToPtr(expected), (UNITY\_FLOAT\*)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_INF(actual, line, message) UnityAssertFloatSpecial((UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_INF)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NEG\_INF(actual, line, message) UnityAssertFloatSpecial((UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NEG\_INF)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NAN(actual, line, message) UnityAssertFloatSpecial((UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NAN)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_DETERMINATE(actual, line, message) UnityAssertFloatSpecial((UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_DET)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NOT\_INF(actual, line, message) UnityAssertFloatSpecial((UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NOT\_INF)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NOT\_NEG\_INF(actual, line, message) UnityAssertFloatSpecial((UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NOT\_NEG\_INF)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NOT\_NAN(actual, line, message) UnityAssertFloatSpecial((UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NOT\_NAN)

#define UNITY\_TEST\_ASSERT\_FLOAT\_IS\_NOT\_DETERMINATE(actual, line, message) UnityAssertFloatSpecial((UNITY\_FLOAT)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NOT\_DET)

#endif

#ifdef UNITY\_EXCLUDE\_DOUBLE

#define UNITY\_TEST\_ASSERT\_DOUBLE\_WITHIN(delta, expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_EQUAL\_DOUBLE(expected, actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_EQUAL\_DOUBLE\_ARRAY(expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_DOUBLE(expected, actual, num\_elements, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_INF(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NEG\_INF(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NAN(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_DETERMINATE(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NOT\_INF(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NOT\_NEG\_INF(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NOT\_NAN(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NOT\_DETERMINATE(actual, line, message) UNITY\_TEST\_FAIL((UNITY\_LINE\_TYPE)(line), UnityStrErrDouble)

#else

#define UNITY\_TEST\_ASSERT\_DOUBLE\_WITHIN(delta, expected, actual, line, message) UnityAssertDoublesWithin((UNITY\_DOUBLE)(delta), (UNITY\_DOUBLE)(expected), (UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line))

#define UNITY\_TEST\_ASSERT\_EQUAL\_DOUBLE(expected, actual, line, message) UNITY\_TEST\_ASSERT\_DOUBLE\_WITHIN((UNITY\_DOUBLE)(expected) \* (UNITY\_DOUBLE)UNITY\_DOUBLE\_PRECISION, (UNITY\_DOUBLE)(expected), (UNITY\_DOUBLE)(actual), (UNITY\_LINE\_TYPE)(line), (message))

#define UNITY\_TEST\_ASSERT\_EQUAL\_DOUBLE\_ARRAY(expected, actual, num\_elements, line, message) UnityAssertEqualDoubleArray((UNITY\_DOUBLE\*)(expected), (UNITY\_DOUBLE\*)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_ARRAY)

#define UNITY\_TEST\_ASSERT\_EACH\_EQUAL\_DOUBLE(expected, actual, num\_elements, line, message) UnityAssertEqualDoubleArray(UnityDoubleToPtr(expected), (UNITY\_DOUBLE\*)(actual), (UNITY\_UINT32)(num\_elements), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_ARRAY\_TO\_VAL)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_INF(actual, line, message) UnityAssertDoubleSpecial((UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_INF)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NEG\_INF(actual, line, message) UnityAssertDoubleSpecial((UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NEG\_INF)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NAN(actual, line, message) UnityAssertDoubleSpecial((UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NAN)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_DETERMINATE(actual, line, message) UnityAssertDoubleSpecial((UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_DET)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NOT\_INF(actual, line, message) UnityAssertDoubleSpecial((UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NOT\_INF)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NOT\_NEG\_INF(actual, line, message) UnityAssertDoubleSpecial((UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NOT\_NEG\_INF)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NOT\_NAN(actual, line, message) UnityAssertDoubleSpecial((UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NOT\_NAN)

#define UNITY\_TEST\_ASSERT\_DOUBLE\_IS\_NOT\_DETERMINATE(actual, line, message) UnityAssertDoubleSpecial((UNITY\_DOUBLE)(actual), (message), (UNITY\_LINE\_TYPE)(line), UNITY\_FLOAT\_IS\_NOT\_DET)

#endif

/\* End of UNITY\_INTERNALS\_H \*/

#endif