# Misc

Ubuntu color pallete

<https://design.ubuntu.com/brand/colour-palette>

<https://design.ubuntu.com/brand/colour-palette>

<https://linux.die.net/man/1/rsync>

### GTest (Google unit test)

<https://github.com/google/googletest>

# C++

Creating and Using a Dynamic Link Library (C++)

<https://msdn.microsoft.com/en-us/library/ms235636.aspx>

DLLs in Visual C++

<https://msdn.microsoft.com/en-us/library/1ez7dh12.aspx>

Int-To-Type

<http://www.drdobbs.com/genericprogramming-mappings-between-type/184403750>

<https://stackoverflow.com/questions/29518736/how-to-build-type2int-from-int2type>

Dynamic Load

<https://stackoverflow.com/questions/8696653/dynamically-load-a-function-from-a-dll>

<https://msdn.microsoft.com/en-us/library/windows/desktop/ms686944(v=vs.85).aspx>

<https://msdn.microsoft.com/en-us/library/ms810279.aspx>

[C++ String Compare Performance](https://tinodidriksen.com/2011/02/cpp-string-compare-performance/)

<https://tinodidriksen.com/2011/02/cpp-string-compare-performance/>

C++ Language

<http://en.cppreference.com/w/cpp/language>

Linking an Executable to a DLL

<https://msdn.microsoft.com/en-us/library/9yd93633.aspx>

<https://msdn.microsoft.com/en-us/library/k70yt3e2(v=vs.100).aspx>

dllexport, dllimport

<https://msdn.microsoft.com/en-us/library/3y1sfaz2.aspx>

def file

<https://msdn.microsoft.com/en-us/library/hyx1zcd3.aspx>

<https://msdn.microsoft.com/en-us/library/d91k01sh.aspx>

User defined Macros (PATH)

<https://msdn.microsoft.com/en-us/library/f2t8ztwy(v=vs.90).aspx>

Specifying Custom Build Tools

<https://msdn.microsoft.com/en-us/library/hefydhhy.aspx>

Memory leaks and debugging

https://msdn.microsoft.com/en-us/library/x98tx3cf(v=vs.100).aspx

<https://stackoverflow.com/questions/3202520/c-memory-leak-testing-with-crtdumpmemoryleaks-does-not-output-line-numb>

<https://stackoverflow.com/questions/8718758/using-crtdumpmemoryleaks-to-display-data-to-console>

// declare memory stare variable

\_CrtMemState state;

...

// create a checkpoint to for current memory state

\_CrtMemCheckpoint(&state);

... do stuff ...

// report differences

\_CrtMemDumpAllObjectsSince(&state);

This routine will dump all allocations since the checkpoint. IT can be wrapped around a function call, loaded on startup and when exiting, etc. I have also used it in a DLL in DllMain process attach/detach.

Also handy when combined with \_CrtSetReportMode, \_CrtSetReportFile, etc.

<https://msdn.microsoft.com/en-us/library/aa246768(v=vs.60).aspx>

Atomic flag

<http://en.cppreference.com/w/cpp/atomic/atomic_flag>

<https://stackoverflow.com/questions/11217501/global-variable-in-namespace-values-differ-in-threads>

Atomic operations library

<http://en.cppreference.com/w/cpp/atomic>

Filesystem library

<http://en.cppreference.com/w/cpp/filesystem>

Misc

<https://stackoverflow.com/questions/8317010/how-to-declare-and-define-global-variables-in-order-to-access-them-from-all-head>

### Boost

Boost.Lockfree

<http://www.boost.org/doc/libs/1_64_0/doc/html/lockfree.html>

<http://www.boost.org/doc/libs/1_64_0/>

Filesystem

<http://www.boost.org/doc/libs/1_63_0/libs/filesystem/example/tut4.cpp>

<http://www.boost.org/doc/libs/1_63_0/libs/filesystem/doc/tutorial.html>

Multiprecision

<http://www.boost.org/doc/libs/1_63_0/libs/multiprecision/doc/html/boost_multiprecision/tut/ints/cpp_int.html>

<http://www.boost.org/doc/libs/1_63_0/libs/multiprecision/doc/html/index.html>

Matrix

<http://www.boost.org/doc/libs/1_63_0/libs/numeric/ublas/doc/index.html>

<http://www.boost.org/doc/libs/1_63_0/libs/numeric/ublas/doc/matrix_expression.html#1MatrixExpression>

<http://www.boost.org/doc/libs/1_63_0/libs/numeric/ublas/doc/operations_overview.html>

Bind

<http://rakafon.blogspot.com/2009/04/boostbind.html>

Multi index

<https://stackoverflow.com/questions/39510143/how-to-use-create-boostmulti-index>

C++ 17

<https://isocpp.org/std/status>

https://en.wikipedia.org/wiki/C%2B%2B17

<https://stackoverflow.com/questions/38060436/what-are-the-new-features-in-c17>

# Gtest

https://github.com/google/googletest

<https://github.com/google/googletest/blob/master/README.md>

<https://stackoverflow.com/questions/12558327/google-test-in-visual-studio-2012>

How to build

<https://github.com/google/googletest/blob/master/googletest/README.md>

Very important compilation notes

<https://github.com/google/googletest/blob/master/googletest/README.md>

Google Mock

<https://github.com/google/googletest/blob/master/googlemock/README.md>

**Choosing a TR1 Tuple Library**

Some Google Test features require the C++ Technical Report 1 (TR1) tuple library, which is not yet available with all compilers. The good news is that Google Test implements a subset of TR1 tuple that's enough for its own need, and will automatically use this when the compiler doesn't provide TR1 tuple.

Usually you don't need to care about which tuple library Google Test uses. However, if your project already uses TR1 tuple, you need to tell Google Test to use the same TR1 tuple library the rest of your project uses, or the two tuple implementations will clash. To do that, add

-DGTEST\_USE\_OWN\_TR1\_TUPLE=0

to the compiler flags while compiling Google Test and your tests. If you want to force Google Test to use its own tuple library, just add

-DGTEST\_USE\_OWN\_TR1\_TUPLE=1

to the compiler flags instead.

If you don't want Google Test to use tuple at all, add

**-DGTEST\_HAS\_TR1\_TUPLE=0 (set this for the compilation for VS2012)**

and all features using tuple will be disabled.

Also

<https://github.com/appveyor/ci/issues/742>

<https://msdn.microsoft.com/en-us/library/bdscwf1c.aspx>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SHORT MEMO \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ASSERTIONS

https://github.com/google/googletest/blob/master/googletest/docs/Primer.md

Fatal assertion

ASSERT\_TRUE( condition ) ; condition is true

ASSERT\_FALSE( condition ) ; condition is false

Nonfatal assertion

EXPECT\_TRUE( condition ) ; condition is true

EXPECT\_FALSE( condition ) ; condition is false

---------------------------------------------------------------

BINARY COMPARISION

Fatal assertion

ASSERT\_EQ( val1 , val2 ); val1 == val2

ASSERT\_NE( val1 , val2 ); val1 != val2

ASSERT\_LT( val1 , val2 ); val1 < val2

ASSERT\_LE( val1 , val2 ); val1 <= val2

ASSERT\_GT( val1 , val2 ); val1 > val2

ASSERT\_GE( val1 , val2 ); val1 >= val2

Nonfatal assertion

EXPECT\_EQ( val1 , val2 ); val1 == val2

EXPECT\_NE( val1 , val2 ); val1 != val2

EXPECT\_LT( val1 , val2 ); val1 < val2

EXPECT\_LE( val1 , val2 ); val1 <= val2

EXPECT\_GT( val1 , val2 ); val1 > val2

EXPECT\_GE( val1 , val2 ); val1 >= val2

---------------------------------------------------------------

STRING COMPARISON

The assertions in this group compare two C strings.

If you want to compare two string objects,

use EXPECT\_EQ , EXPECT\_NE , and etc instead.

Fatal assertion

ASSERT\_STREQ( str1 , str2 ); the two C strings have the same content

ASSERT\_STRNE( str1 , str2 ); the two C strings have different content

ASSERT\_STRCASEEQ( str1 , str2 ); the two C strings have the same content, ignoring case

ASSERT\_STRCASENE( str1 , str2 ); the two C strings have different content, ignoring case

Nonfatal assertion

EXPECT\_STREQ( str1 , \_str\_2 ); the two C strings have the same content

EXPECT\_STRNE( str1 , str2 ); the two C strings have different content

EXPECT\_STRCASEEQ( str1 , str2 ); the two C strings have the same content, ignoring case

EXPECT\_STRCASENE( str1 , str2 ); the two C strings have different content, ignoring case

Note that "CASE" in an assertion name means that case is ignored.

\*STREQ\* and \*STRNE\* also accept wide C strings ( wchar\_t\* ).

If a comparison of two wide strings fails, their values will be printed as UTF-8 narrow strings.

A NULL pointer and an empty string are considered different.

\*/

///////////////////////////////////////////////////////////////////////////////

// TESTS

<https://stackoverflow.com/questions/12076072/how-to-run-specific-test-cases-in-googletest>

[Sign up](https://stackoverflow.com/users/signup?ssrc=hero&returnurl=%2fusers%2fstory%2fcurrent&amp;utm_source=stackoverflow.com&amp;utm_medium=dev-story&amp;utm_campaign=signup-redirect)

## How to run specific test cases in GoogleTest

You should set a flag filter, like

::testing::GTEST\_FLAG(filter) = "Test\_Cases1\*";

such that your main function becomes something like

int main(int argc, char \*\*argv) {

::testing::InitGoogleTest(&argc, argv);

::testing::GTEST\_FLAG(filter) = "Test\_Cases1\*";

return RUN\_ALL\_TESTS();

}

