

FINALLY, AFTER 8 MONTHS OF DEVELOPMENT, THE SYSTEM IS WHAT WE EXPECTED. NOW I NEED YOU TO TURN IT ACCESSIBLE ONLINE



WHAT?? BUT IT WASN'T DESIGNED THIS WAY. IT WAS BUILT TO RUN ONLY ON DESKTOP!



OK. SO YOU ONLY HAVE TO RECOMPILE IT AS A WEBSITE...



**PROGRAMMER'S  
LIFE**

```
real story;  
string sender = "@notsodeep";
```

#180



August meetup

vienna.cpp user group

# Organizational Stuff

# Organizational Stuff

- `vienna_cpp_user_group.members() > 60`

The screenshot shows the Facebook page for the Vienna C++ User Group. The header features the group's logo, which consists of a stylized 'C++' symbol and the text 'vienna.cpp user group'. Below the header is a navigation bar with links for Home, Members, Sponsors, Photos, Pages, Discussions, and More. On the right side of the navigation bar are links for Group tools and My profile. The main content area is divided into three columns. The left column contains the group's location (Vienna, Austria), founding date (Apr 28, 2016), and buttons for 'About us...' and 'Invite friends'. Below these buttons, the text 'C++ enthusiasts' is followed by a circled number '63' and a small '1' below it. The middle column has a 'Welcome!' section with a 'Schedule a new Meetup' button, followed by a calendar view showing 'Upcoming 1', 'Past', and 'Calendar'. The main event listed is 'August Meetup: "Undefined Behavior"' at Metalab, on Tue Aug 30 at 7:00 PM. The right column is titled 'What's new' and shows two recent RSVPs: Hans-Peter Schrei and Florian Henke, both for the August Meetup.

# Organizational Stuff

- poll.cpp of the month
  - One poll one month before the next meeting, question will be decided on the meeting
- posted on [reddit/r/cpp](https://www.reddit.com/r/cpp)

# Organizational Stuff

- **September Meetup**
  - **DateTimeLocation:**  
**27/09 @7pm @metalab.lib**
  - **Possible topic:**  
**C++ Interview Questions**

# Organizational Stuff

- **vienna.cpp user group std lib meeting**
- **DateTimeLocation:  
04/10 @7pm @metalab.lib**
- **Topic:  
std::min(...)**

# **C++ News**



# C++ News

- **opt-in header-only libraries**

## opt-in header-only libraries

6 august 2016

[c++](#) [c++14](#) [library](#) [cmake](#) [api](#) [tutorial](#) [article](#)

An interesting **header-only** logging library, *spdlog*, was recently posted on Reddit. A [comment by Sean Middleditch](#) sparked an interesting discussion on the thread:

"Header only" is an anti-feature. Fast compiles are important. PCHes only fix a fraction of the problems of header bloat. Avoiding 15 minutes of setup to get a library building/precompiled in exchange for months of lost productivity waiting for slow builds is a pretty bad trade off.

Here are some of the replies:

"Honestly, I lost months of productivity trying to handle the compilation of twenty libraries on X different platforms with Y different link flags, with X and Y both > 10"

"Respectfully, I disagree. Header-only libraries save a lot of setup stage time and they were a real breeze to work with when it comes down to cross-platform development. Some may argue that CMake solves that problem, but I would argue that CMake is a problem itself."

Another reply is particularly interesting, and inspired me to write this article:

[http://vittorioromeo.info/index/blog/2016\\_cpp\\_library\\_configuration\\_api.html](http://vittorioromeo.info/index/blog/2016_cpp_library_configuration_api.html)

really nice for quick setups, small and test projects (tutorial, workshops...), or having

# C++ News

- **opt-in header-only libraries with CMake**

```
1  cmake_minimum_required(VERSION 3.3)
2
3  project(example_lib)
4
5  # define the library
6
7  set(library_srcs
8      example_lib/library/module0/module0.cpp
9      example_lib/library/module1/module1.cpp
10 )
11
12 add_library(library_static STATIC ${library_srcs})
13 add_library(library_shared SHARED ${library_srcs})
14
15 add_library(library_iface INTERFACE)
16 target_compile_definitions(library_iface
17     INTERFACE LIBRARY_HEADER_ONLY
18 )
19
20 set(installed_srcs
21     include/example_lib/library/module0/module0.cpp
22     include/example_lib/library/module1/module1.cpp
23 )
24 add_library(library_srcs INTERFACE)
25 target_sources(library_srcs INTERFACE
26     $<INSTALL_INTERFACE:${installed_srcs}>
27 )
28
29 # install and export the library
30
31 install(DIRECTORY
32     example_lib/library
33     DESTINATION
34     include/example_lib
35 )
36 install(FILES
```

<https://steveire.wordpress.com/2016/08/09/opt-in-header-only-libraries-with-cmake/>

```
39     DESTINATION
40     include/example lib
```

# C++ News

- **Mangrove: MongoDB ODM**

mongodb-labs / mangrove

Watch 9 Star 7 Fork 1

Code Issues 0 Pull requests 1 Pulse Graphs

Mangrove: the MongoDB C++ ODM <http://mongodb-labs.github.io/mangrove>

41 commits 10 branches 0 releases 2 contributors

Branch: master New pull request Find file Clone or download

Adam Chelminski CXXODM-19 Add support for mongocxx::options:: in active-record save()...	Latest commit 29472f2 25 days ago
build	initial commit 3 months ago
buildscripts	CXXODM-44 Rename everything to Mangrove, except the repo a month ago
etc	CXXODM-46 CXXODM-48 CXXODM-14 Add README, Doxyfile, and Hugo-powered ... a month ago
examples	CXXODM-62 Include cereal headers for supported STL containers 28 days ago
hugo	CXXODM-58 Add support for ` \$where ` operator in builder. 24 days ago
src	CXXODM-19 Add support for mongocxx::options:: in active-record save()... 19 days ago
.clang-format	initial commit 3 months ago
.gitignore	CXXODM-46 CXXODM-48 CXXODM-14 Add README, Doxyfile, and Hugo-powered ... a month ago
.travis.yml	CXXC
.ycm_extra_conf.py	initial commit 3 months ago

<https://github.com/mongodb-labs/mangrove>

# C++ News

- CppCheck 1.75 released

## Cppcheck

A tool for static C/C++ code analysis

[Home](#) | [Wiki](#) | [Forum](#) | [Issues](#) | [Developer Info](#) | [Online Demo](#) | [Project page](#)

[Download](#) | [Features](#) | [News](#) | [Documentation](#) | [Support](#) | [Contribute](#)

**Cppcheck** is a [static analysis tool](#) for C/C++ code. Unlike C/C++ compilers and many other analysis tools it does not detect syntax errors in the code. Cppcheck primarily detects the types of bugs that the compilers normally do not detect. The goal is to detect only real errors in the code (i.e. have zero false positives).

## Download

**Download Now!**

Version 1.75 for Windows  
(64-bit)

**Download Now!**

Version 1.75 for Windows  
(32-bit)

## Clients and plugins

Cppcheck is integrated with many popular development tools. For instance:

- **CLion** - [Cppcheck plugin](#)
- **Code::Blocks** - *integrated*

<http://cppcheck.sourceforge.net>

# C++ News

- **Qt Lite**



Nils  
Christian  
Roscher-  
Nielsen

## Introducing the Qt Lite project—Qt for any platform, any thing, any size

Published Thursday August 18th, 2016

[29 Comments](#)

Posted in [Biz Circuit & Dev Loop](#)

We believe in a future of great software and hardware, developed together, delivered quickly, and that you can have fun in the process. Embedded development should be just as simple as all other software development, and you should immediately see the result of your ideas running on your device.

The amount of devices and things surrounding us are rapidly increasing, becoming more intelligent and requiring software that runs on a greater variety of hardware—everything from IoT devices with or without a screen, smart watches through to high end smart TVs and industrial grade PCs. As the requirements and the world of software development is changing so does Qt. We have taken action and are now unveiling the Qt Lite Project. This is a whole range of changes to Qt, allowing you to strip Qt down and bring you need in order to create your device for more or less any platform and regardless of size. Qt Lite is neither a separate product nor a fork of Qt— it is an evolution

Join us at #QtWS16

Qt World Summit 2016 Oct 18-20 – Reasons to join  
**SAN FRANCISCO**

LAST YEAR EVENT INFO:	<b>40 COUNTRIES</b>
<b>850</b> ATTENDEES	
<b>60</b> SPEAKERS	
<b>50k</b> SOCIAL MEDIA FOLLOWERS	
EXECUTIVE MANAGEMENT Developers & Team Lead PRODUCT MANAGERS <b>and many more</b>	
QT ECOSYSTEM:	NETWORK WITH OTHERS LIKE:
<b>1M+</b> GLOBAL USERS	
	NASA Rimac HARMAN Ulstein <b>FORMLABS</b> Red Hat BLUESCAPE Intel ABB Magneti Marelli <b>LG</b>

**Qt World Summit**  
@QtWorldSummit

Be inspired by the inevitable//desired, connected//disruptive. See what's Built with Qt [hubs.ly/H042zyt0](https://hubs.ly/H042zyt0)

4:22 PM - 18 Aug 2016

<http://blog.qt.io/blog/2016/08/18/introducing-the-qt-lite-project-qt-for-any-platform-any-thing-any-size/>



# C++ News

- **range-v3 on MSVC on GitHub**

## Visual C++ Team Blog

### Range-v3 on MSVC is Available on GitHub

August 23, 2016 by [EricMittlelette](#) // 10 Comments



f 0 | t 31 | in 8

We are delighted to announce that the Visual C++ Team just published an implementation of range-v3 on the [Microsoft GitHub repo](#).

This contribution comes hot on the heels of our recent work to improve [expression SFINAE](#) on our Visual Studio 2015 Update 3 VC++ compiler . This is the first implementation of the Range TS running in MSVC. In "Ranges for the Standard Library, Revision 1" (N4128) Ranges are defined like this: "A *range* is an object that refers to a sequence of elements, conceptually similar to a pair of iterators. One prime motivation for ranges is to give users a simpler syntax for calling algorithms. Rather than this:

```
std::vector<int> v { /*...*/ };
std::sort( v.begin(), v.end() );
Ranges would give us a pithier syntax:
std::sort( v );"
```

More information on range-v3 [here](#)

Supporting Ranges is important to our C++ team, as exemplified not only by us hiring one of the key ma the investments we continue to make in this feature across our entire compiler (from parsing to code gen



Share This Post



Search this blog  Search all blogs

Tags

[Android Announcement](#)  
[Announcements](#) [ATL Bug Info](#)  
[Build C++](#) [c++/CX](#) [C++0x](#) [c++11](#)  
[C++ language Channel 9](#)

<https://blogs.msdn.microsoft.com/vcblog/2016/08/23/range-v3-on-msvc-is-available-on-github/>

# C++ News

- range-v3 on MSVC on GitHub



**Eric Niebler**

@ericniebler



 Folgen

Want a better compiler? Write a popular library that the compiler fails utterly to compile and wait. @visualc [github.com/Microsoft/Range...](https://github.com/Microsoft/Range-3)  
#cpp

 Übersetzung anzeigen



**Microsoft/Range-V3-VS2015**

Range-V3-VS2015 - A fork of the popular range v3 C++ library with support for the Visual S

<https://twitter.com/ericniebler/status/768195533840146432?lang=de>

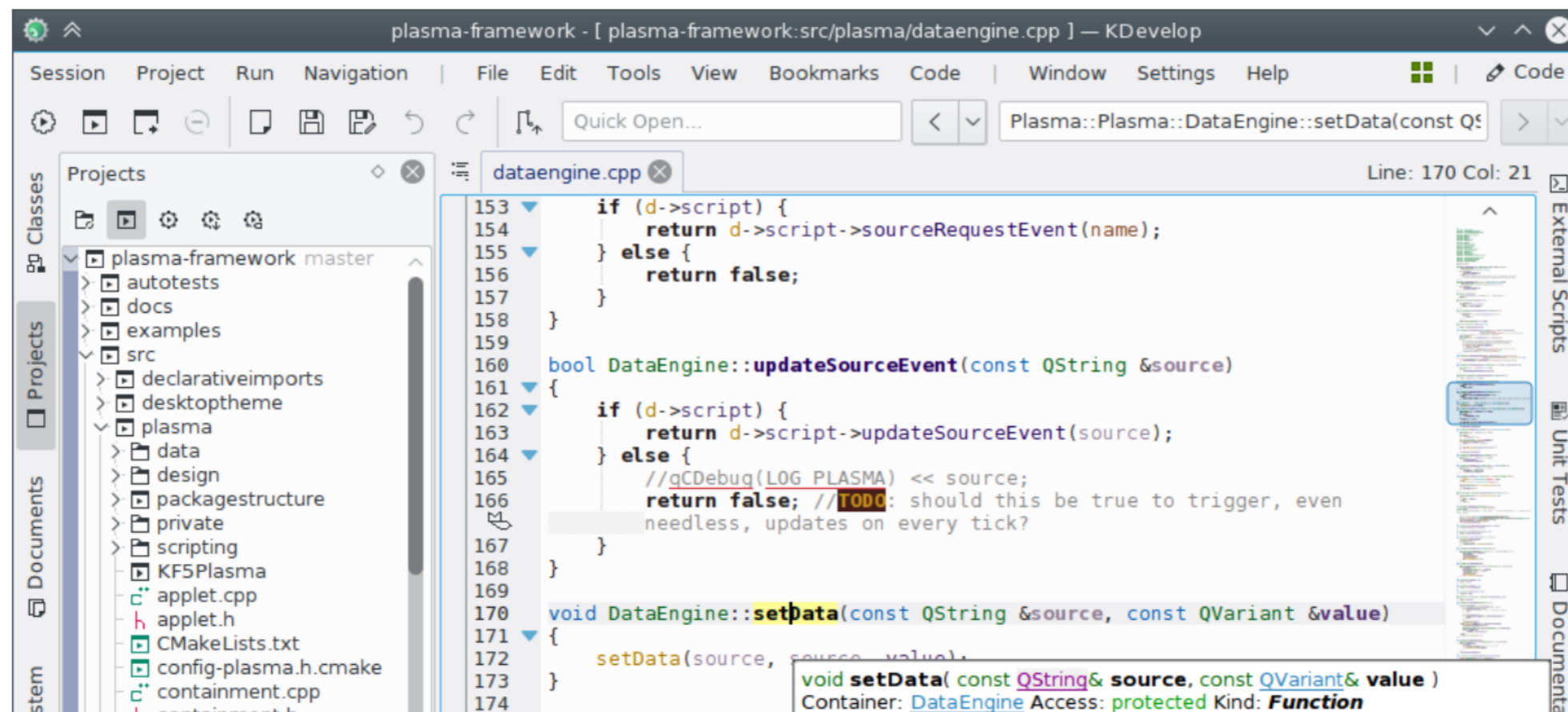
# C++ News

- KDevelop 5.0.0 released

## What's new in KDevelop 5.0?

August 23, 2016 | Tags: kdevelop, kde, kdevelop-release

Almost two years after the release of KDevelop 4.7, we are happy to announce the [immediate availability of KDevelop 5.0!](#)



<http://kfunk.org/2016/08/23/whats-new-in-kdevelop-5-0/>



# Other (Tech) News



Questions Jobs **Documentation Beta** Tags Users Badges Ask Question

Documentation is now in Beta! [Take the tour](#), and [join us in chat](#).

C++ Language **dashboard** all topics

Q search C++ Language [Request Topic](#)

## OVERVIEW

[Introduction to C++](#)

- 73 topics
- 58 topic requests
- 44 proposed changes
- 11 improvement requests

## RECENT CONTRIBUTIONS

- Jarod42 created [Iteration over an enum](#)
- honk requested [std::mutex](#)

## My Drafts

No drafts in C++ Language

[Create New Topic](#)

**summary** requested topics need improvement proposed changes activity my drafts

**58 Requested Topics** No notifications **top** newest handled dismissed

- 28** [Design pattern implementation in C++](#)  
*Implementation of any design pattern using the C++ language.*  
requested by Sean 33.9k  
- 3 comments [Create Topic](#) [Dismiss](#)
- 14** [Atomic Types](#)  
*Information about the different types, synchronization, memory accesses.*  
requested by Pyves 271 [Create Topic](#) [Dismiss](#)
- 11** [Constructors / Destructors](#)  
*Default Constructor, Copy Constructor, Move Constructor, Destructor, etc.*  
requested by A. Sarid 1,479  
- 3 comments [Create Topic](#) [Dismiss](#)

**11 Need Improvement** No notifications **recent** popular worst

- [Enumeration](#)  
*missing examples* [Improve](#)
- [Using a function object consumer](#)  
*unclear* [Improve](#)
- [Creating a std::thread](#)  
*unclear* [Improve](#)
- [User-Defined Literals](#)  
*missing examples* [Improve](#)
- [Sorting](#)  
*missing examples* [Improve](#)