



Latest and Greatest in Visual Studio for C++ Developers

Marian Luparu @mluparu
Sy Brand @TartanLlama

@VisualC
<https://aka.ms/cpp>



Mission of the C++ product team at Microsoft

Make the lives of all C++ developers on the planet better

Our
agenda
today

1. by participating in the creation of the **C++ Standards**
2. by investing in the Microsoft Visual C++ (**MSVC**) Compiler & Libraries
3. by simplifying C++ library acquisition via **Vcpkg**
4. by improving the **Visual Studio IDE**
5. by continuing to enhance the C++ extension for **Visual Studio Code**

Visual Studio Code

```
helloworld.cpp
1  #include <iostream>
2
3  int main()
4  {
5      std::cout << "Hello World!" << std::endl;
6      std::cout
7      return 0
8  }
```

IntelliSense suggestions for `std::cout`:

- `count`
- `cout` (selected)
- `_Count_pr`
- `conjunction`
- `conjunction_v`
- `_Atomic_counter_t`
- `_Check_match_counts`
- `_Copy_unchecked`
- `_Copy_unchecked1`
- `_Get_atomic_count`
- `_Ptr_cout`
- `_Ref_count`

File: iostream

#1 most used code editor

Stack Overflow Developer Surveys (2018, 2019)

- Free, open source code editor
- Runs on Windows, macOS, and Linux
- C++ IntelliSense, Debugging, Code browsing



Tue 9/17 15:15 – 15:45

**What's new in Visual Studio
Code for C++ development**

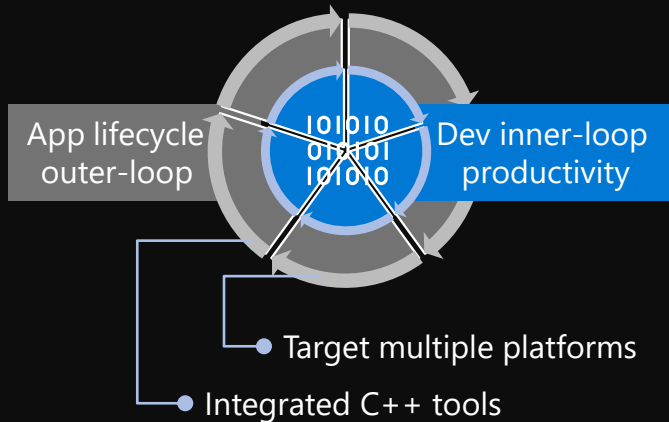
Tara Raj @ Summit 4/5

<https://aka.ms/cpp/code>

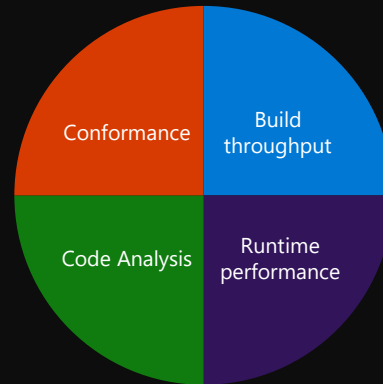
Visual Studio

Agenda

Visual Studio IDE



MSVC Compiler Toolset



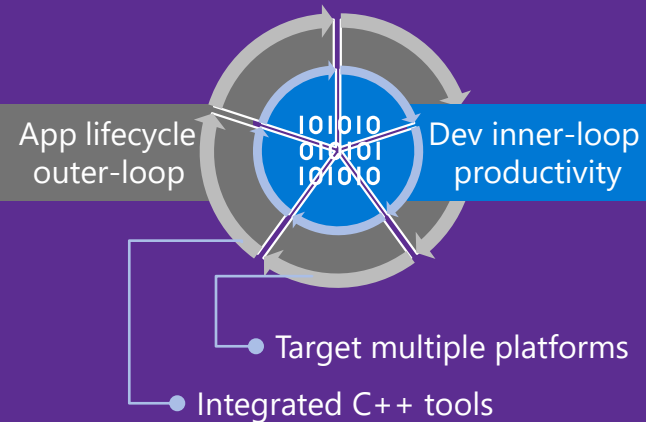
Future of C++ & Tooling



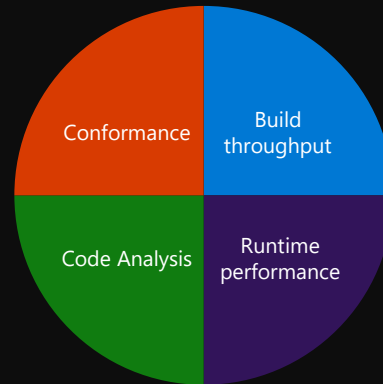
Visual Studio

Agenda

Visual Studio IDE



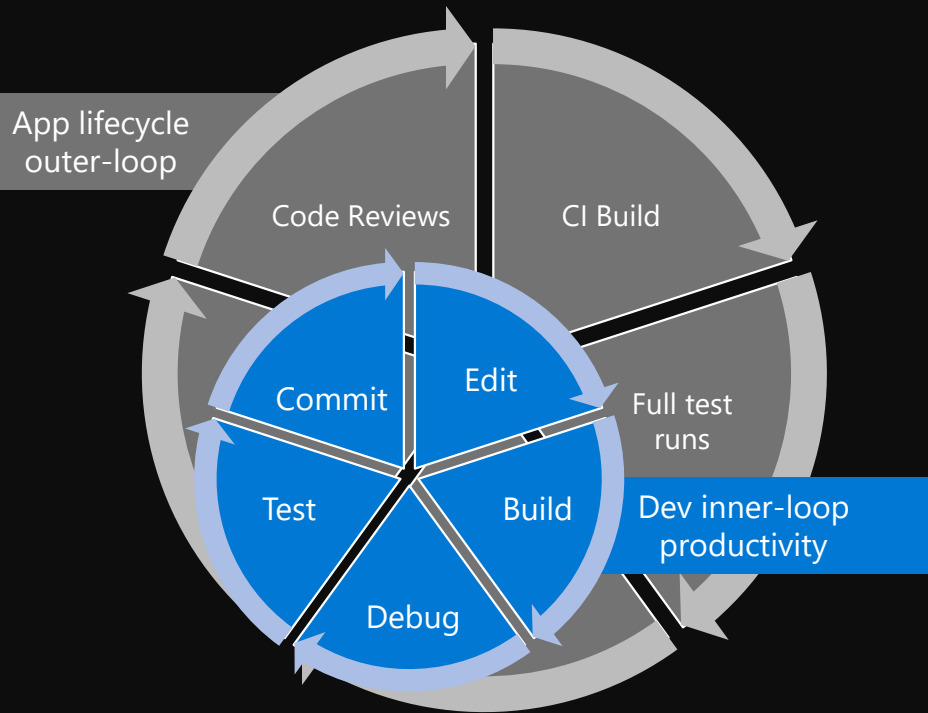
MSVC Compiler Toolset



Future of C++ & Tooling



Visual Studio 2019



Freedom to **target any platform** from one single IDE

ARM/mBed, Android, Cygwin, iOS, Linux, MinGW, UWP, Windows

Easy to get started

Keep your CMake/make/Ninja, No import/conversion to VS solutions, Easy C++ library acquisition

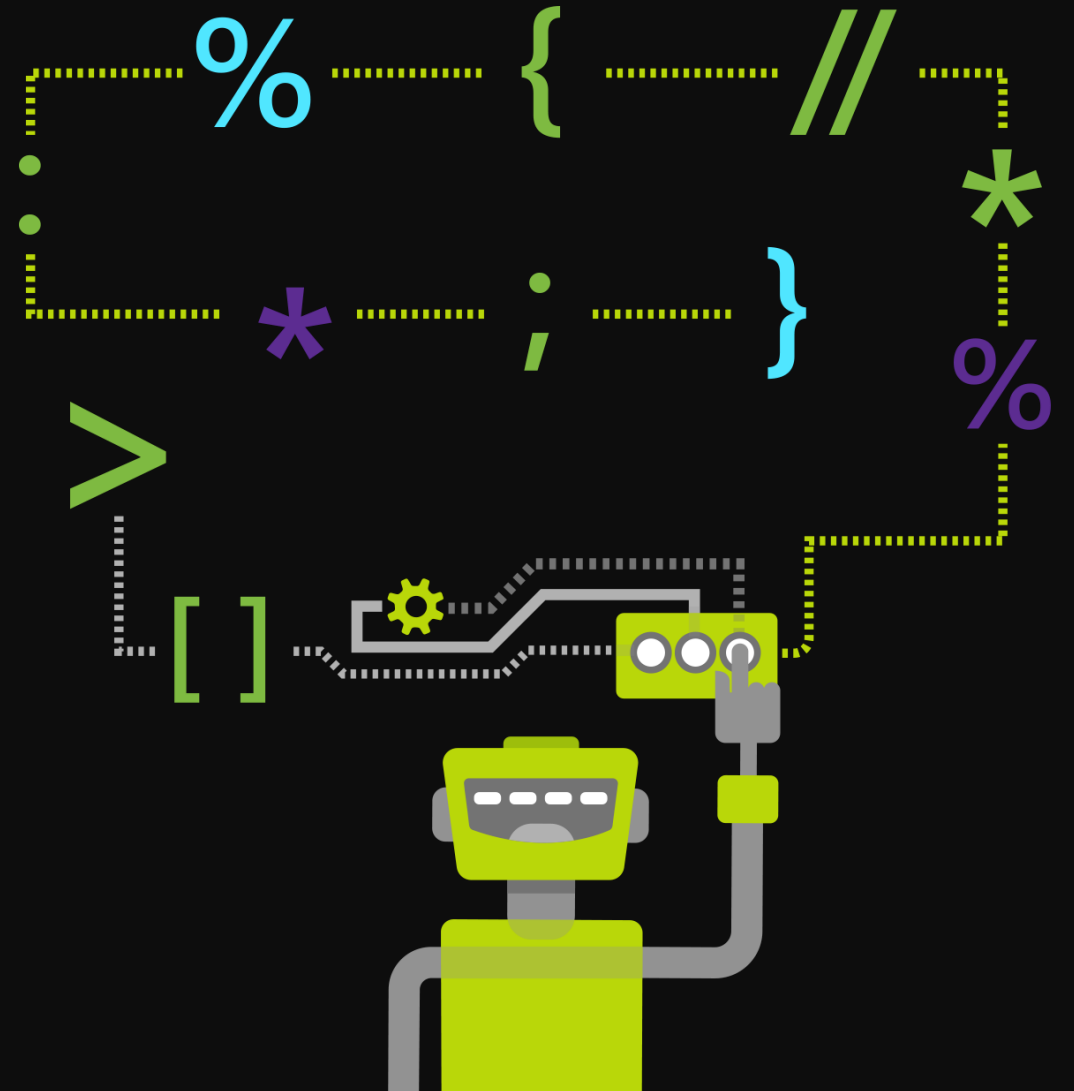
Rich and familiar C++ **code editing and debugging** experiences

IntelliSense, Refactoring, Conditional breakpoints, Debug visualization

Use **your C++ tools** of choice, all **integrated** in the IDE

Clang/LLVM, GCC, Clang-format, Google Test, Boost.Test

Demo



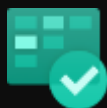


Azure DevOps



Azure Pipelines

CI/CD that works with any language, platform, and cloud. Connect to GitHub or any Git repository and deploy continuously.



Azure Boards

Powerful work tracking with Kanban boards, backlogs, team dashboards, and custom reporting.



Azure Artifacts

Maven, npm, and NuGet package feeds from public and private sources.



Azure Repos

Unlimited cloud-hosted private Git repos for your project. Collaborative pull requests, advanced file management, and more.



Azure Test Plans

All in one planned and exploratory testing solution.

Azure Pipelines is **FREE** for public and private GitHub repos Linux, macOS, and Windows. Unlimited minutes and 10 free parallel jobs for public repositories.

Learn more at <https://aka.ms/cpp/devops>



C++ Library acquisition for Linux, macOS, and Windows

1000+ cross-platform libraries available

A single consistent way to acquire C++ dependencies on all platforms

As simple as "vcpkg install [library_name]"



Recent improvements

Visual Studio and Visual Studio Code integrations

Binary caching for CI scenarios



Wed 9/18 09:00 – 09:30

How to Herd 1,000 Libraries

Robert Schumacher @ Crest 3

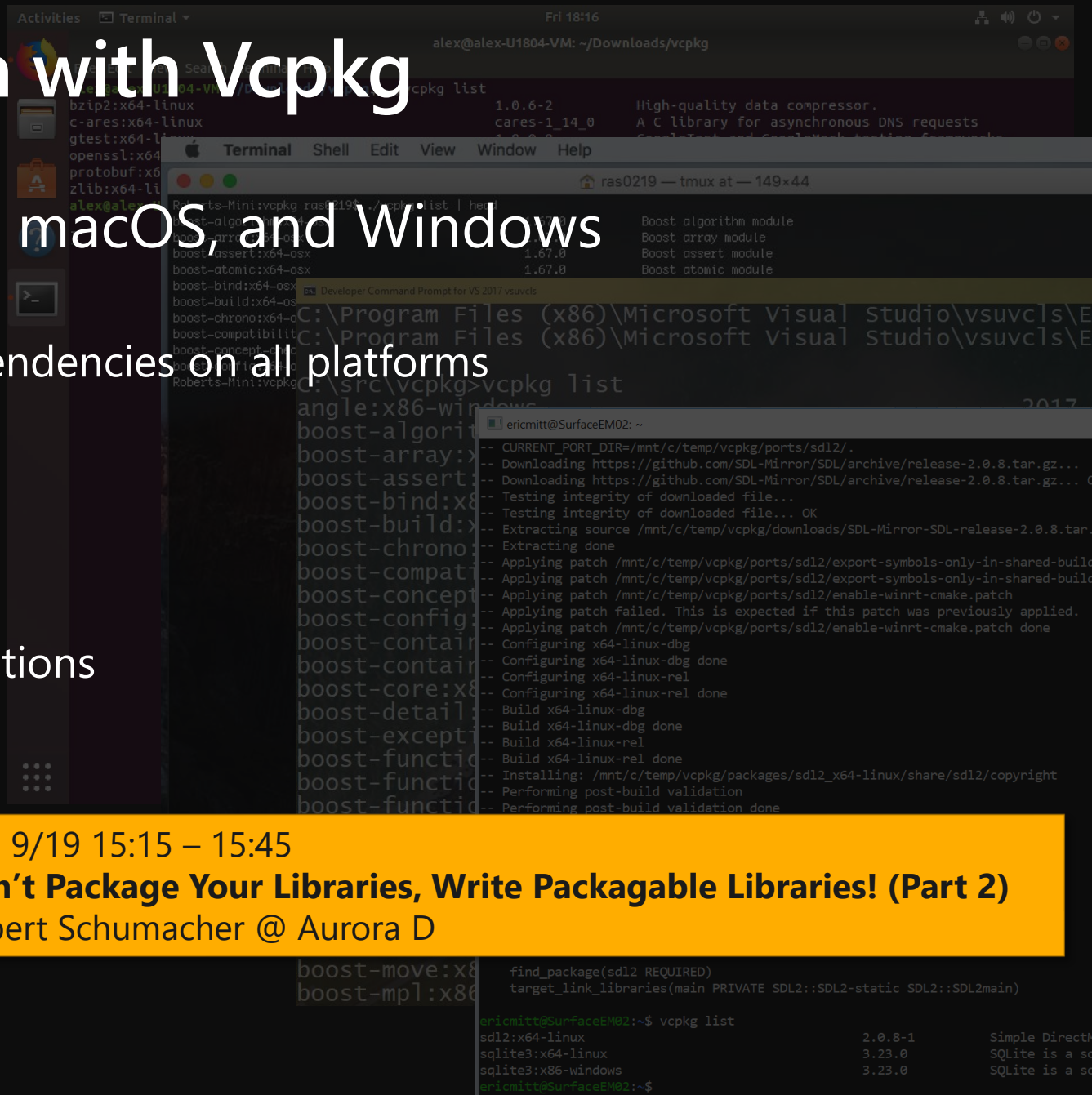


Thu 9/19 15:15 – 15:45

Don't Package Your Libraries, Write Packagable Libraries! (Part 2)

Robert Schumacher @ Aurora D

Learn more at <https://aka.ms/vcpkg>



CMake experience

CMake is a first-class project system in Visual Studio

Familiar edit-build-debug inner-loop

Target Windows, Linux, MinGW and more

Learn more at <https://aka.ms/cmake>

New in Visual Studio 2019

CMake Settings UI

Open existing cache

CMake FileAPI-based integration

C++ library acquisition via Vcpkg

Code Analysis squiggles and quick actions

Clang/LLVM integration

Clang 8.0.1

Acquired through VS Installer

Installation details

☒ C++ Clang tools for Windows (8.0.1 - x64/x86)

Clang-cl integration on Windows

Easy to use with both MSBuild and CMake projects, when targeting Windows or Linux

Learn more at <https://aka.ms/cpp/clang>

Announcing

Clang-Tidy integration

- In Visual Studio 2019 release 16.4
- **coming next week**
- Review clang-tidy warnings as editor squiggles & in Error List
- Both MSBuild and CMake integration
- Works with Windows and Linux configurations

Linux targeting

Target any Linux distro

IntelliSense can run in GCC compat mode and can parse remote Linux headers

Build remotely via MSBuild or CMake

Debug local or remote targets

Learn more at <https://aka.ms/vslinux>

New in Visual Studio 2019

Local WSL (Windows subsystem for Linux) integration

ASAN integration

Open existing remote cache

C++ library acquisition via Vcpkg

Use Clang or GCC

Separate build & debug on different remote machines

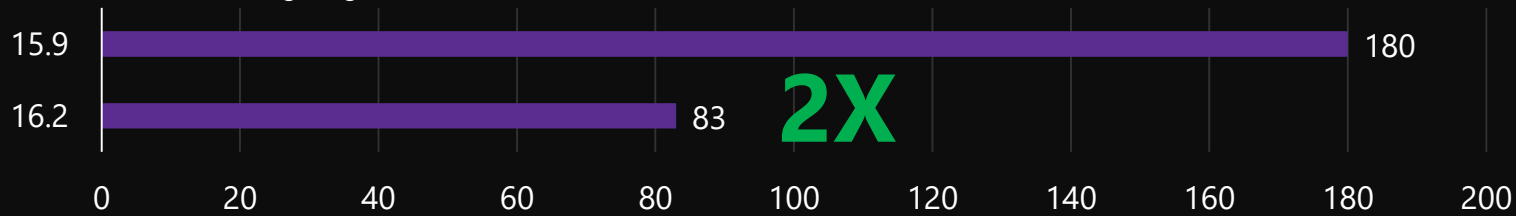
Logging for remote connections

IDE Performance

Improvements in VS 2019

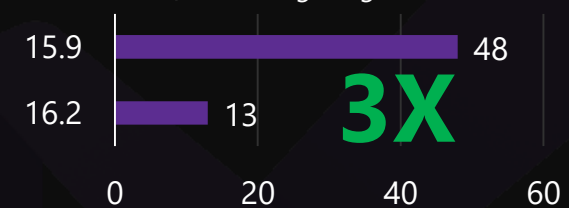
Open CMake-based folder → Full C++ IntelliSense available (seconds)

LLVM folder, targeting Windows



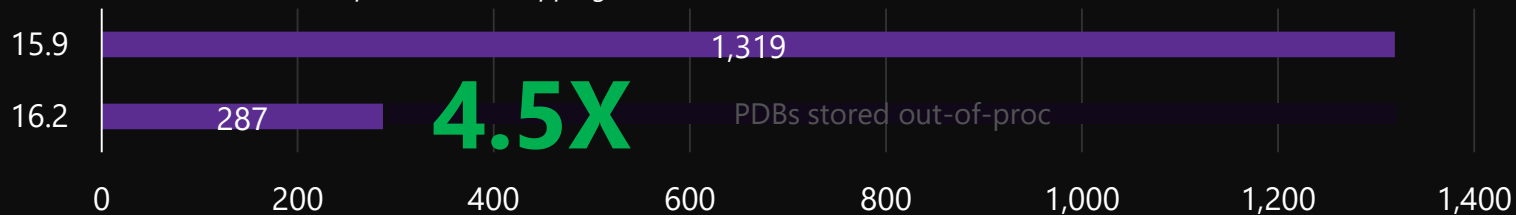
CMake targeting Linux: Rsync (seconds)

LLVM folder, targeting Linux



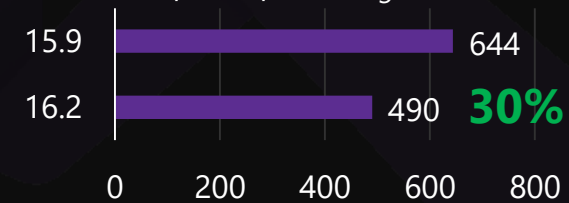
C++ Debugger memory consumption (MB, devenv.exe)

Gears of War 4, breakpoints hit & stepping for 5 minutes



C++ IntelliSense Ready (ms)

LLVM folder, file editing



C++ Productivity Features

New in Visual Studio 2019

Template IntelliSense

Find all instantiations

Code Analysis squiggles in editor

New Quick Actions

Missing #include

Replace NULL with nullptr

Initialize variable

IntelliCode on by default

Live Share

New Semantic Colorization

Call Stack – Template Argument Filtering

IntelliSense Member List filtering on type qualifiers

Toggle line comments

Quick Info colorization

Quick Info links to Online Docs



Tue 9/17 15:50 – 16:10

**Upgrade from “permissive C++” to
“modern C++” with Visual Studio 2019**

Nick Uhlenhuth @ Crest 3

Visual Studio Live Share

 **New** in Visual Studio 2019

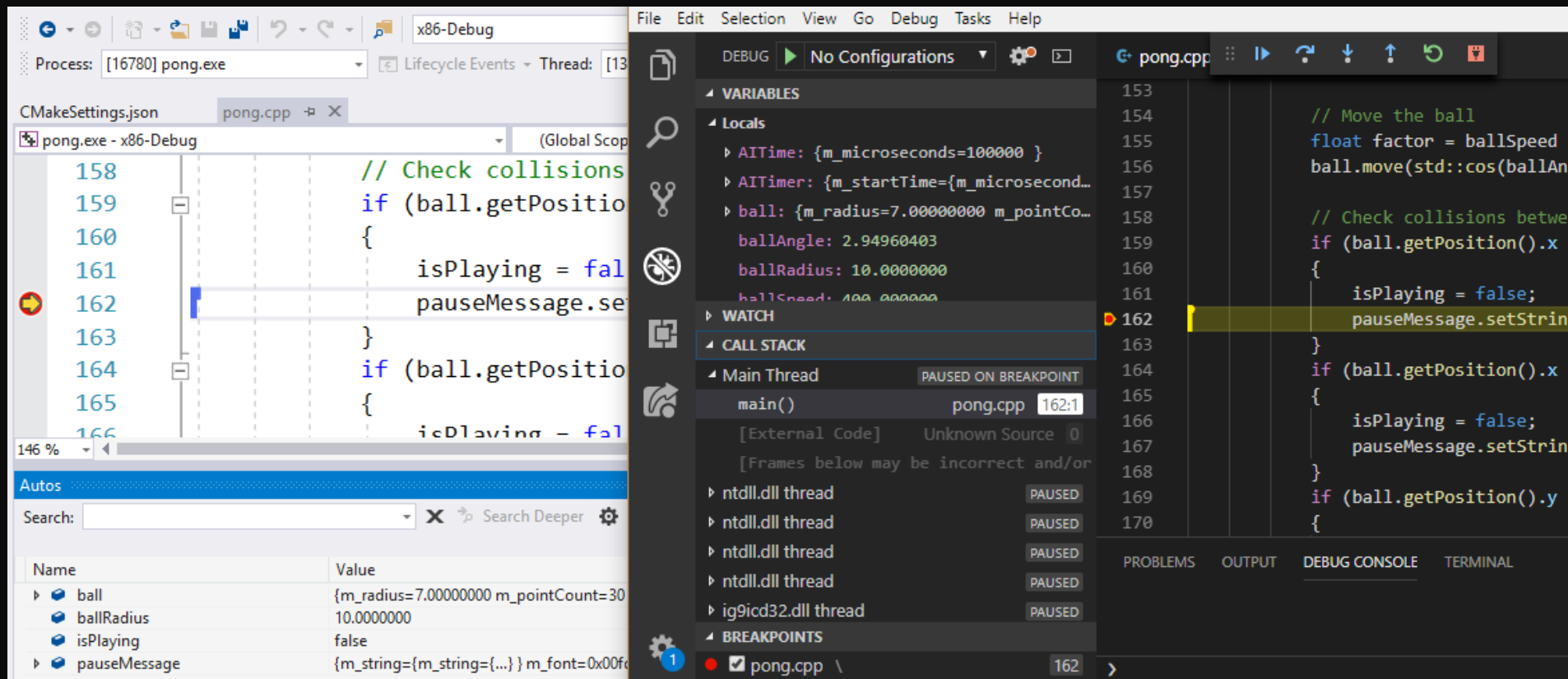
Easily work together, in any way you need

Collaborative editing

Collaborative debugging

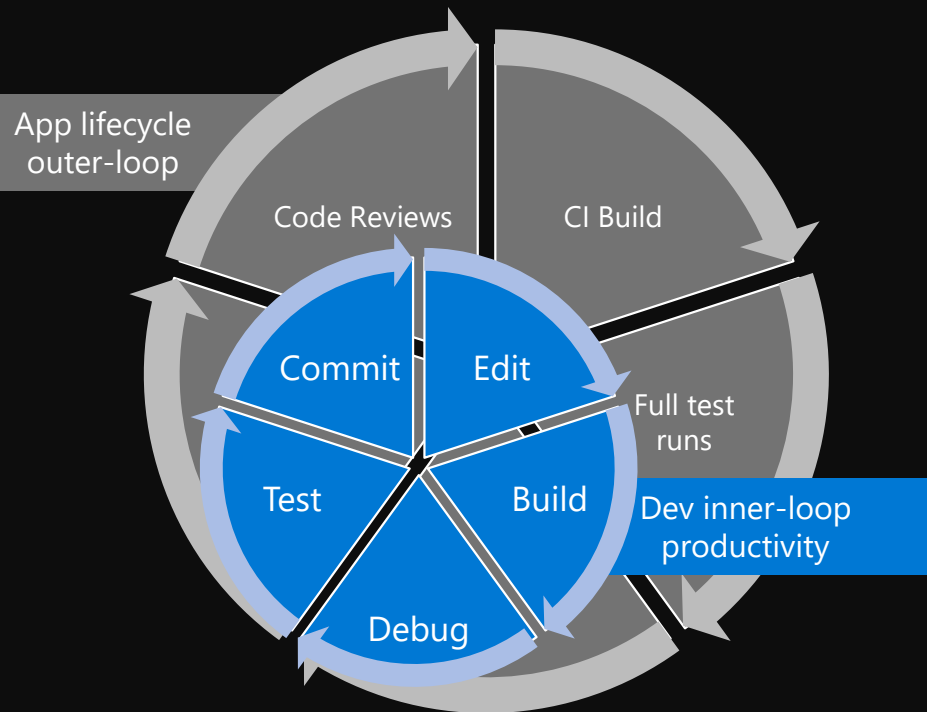
No hassle sharing for any app, on any OS

Visual Studio or Visual Studio Code
macOS, Linux, or Windows



Learn more at <https://aka.ms/cpp/liveshare>

Visual Studio 2019



Freedom to **target any platform** from one single IDE

ARM/mBed, Android, Cygwin, iOS, Linux, MinGW, UWP, Windows

Easy to get started

Keep your CMake/make/Ninja, No import/conversion to VS solutions,
Easy C++ library acquisition

Rich and familiar C++ **code editing and debugging**
experiences

IntelliSense, Refactoring, Conditional breakpoints, Debug visualization

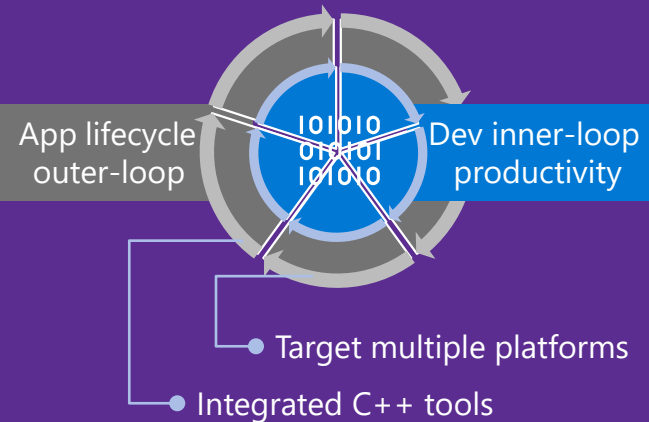
Use **your C++ tools** of choice, all **integrated** in the IDE

Clang/LLVM, Clang-Tidy, GCC, Clang-format, Google Test, Boost.Test

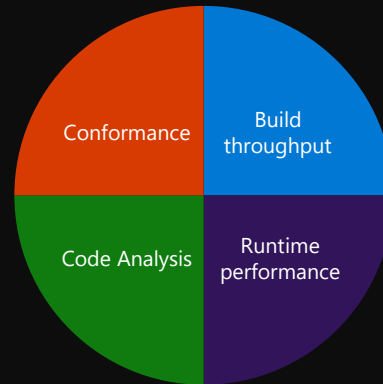
Visual Studio

Agenda

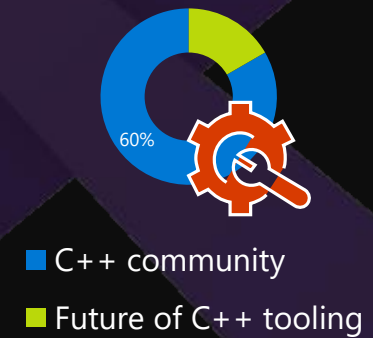
Visual Studio IDE



MSVC Compiler Toolset



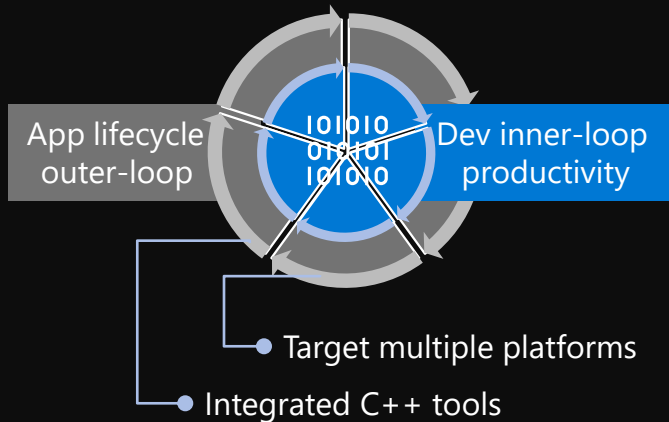
Future of C++ & Tooling



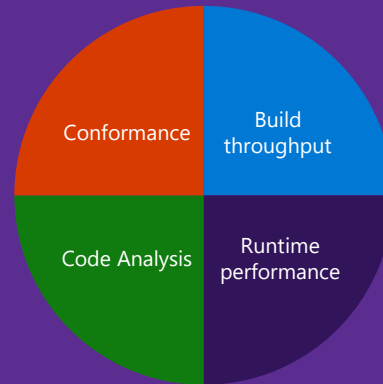
Visual Studio

Agenda

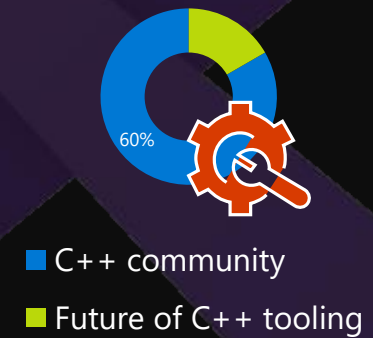
Visual Studio IDE



MSVC Compiler Toolset



Future of C++ & Tooling



Visual Studio 2019

✓ C++98*

* with /permissive-

✓ C++11

✓ C++14

✓ C++17*

* preprocessor is experimental

* most complete Standard Library implementation

✓ C++20

under /std:c++latest

✓ C++ standards conformance

With version 15.7,

Visual Studio 2017 achieves

C++ standards conformance

- supporting all C++11/14/17 compiler features,
- and including two-phase name lookup and expression SFINAE,
- and the most complete C++17 library implementation

For more details, visit <https://aka.ms/msvcconformance>



Conformance

✓ C++98*

* with /permissive-

✓ C++11

✓ C++14

✓ C++17*

* preprocessor is experimental

✓ C++20

under /std:c++latest

Announcing

STL is now C++17 complete

MSVC's Standard Library implementation is C++17 complete

Complete <charconv> available in Visual Studio 2019 version 16.4



Thu 9/19 16:45 – 17:45

**Floating-Point charconv: Making Your Code
10x Faster With C++17's Final Boss**

Stephan T. Lavavej @ Aurora D



Tue 9/17 15:50 – 16:10

C++ Standard Library "Little Things"

Billy O'Neal @ Crest 4/5

Conformance

✓ C++98*

* with /permissive-

✓ C++11

✓ C++14

✓ C++17*

* preprocessor is experimental

✓ C++20

under /std:c++latest



Mon 9/16 15:15 – 16:15

Programming with C++ Modules: Guide for the Working Programmer

Gabriel Dos Reis @ Aurora C

C++20 progress

- **Coroutines** (partial support) under /await
- **Modules** (partial support) under /experimental:module
- **<=> three-way comparison** operator (partial support)
- Feature-test macros
- remove_cvref
- Prevent aggregate-init with user-declared or deleted constructors

Conformance

✓ C++98*

* with /permissive-

✓ C++11

✓ C++14

✓ C++17*

* preprocessor is experimental

✓ C++20

under /std:c++latest

✓ Concepts

Announcing

C++20 Concepts are feature complete

Available in Visual Studio 2019 version 16.3
Under /std:c++latest

Validated with concept-enabled **ranges-v3** and **cmcSTL2** libraries

Learn more at <https://aka.ms/cpp/concepts>

Conformance

✓ C++98*

* with /permissive-

✓ C++11

✓ C++14

✓ C++17*

* preprocessor is experimental

✓ C++20

under /std:c++latest

✓ Concepts

Announcing

STL is now Open Source

Available today at <https://github.com/Microsoft/STL>

Development will happen in the **open Track** and/or **contribute** to STL's conformance progress

STL will be under a **friendly OSS license**
No licensing changes for existing VS users

Learn more at <https://aka.ms/cpp/stl>

Conformance

Visual Studio 2019 Linker Improvements

16.0

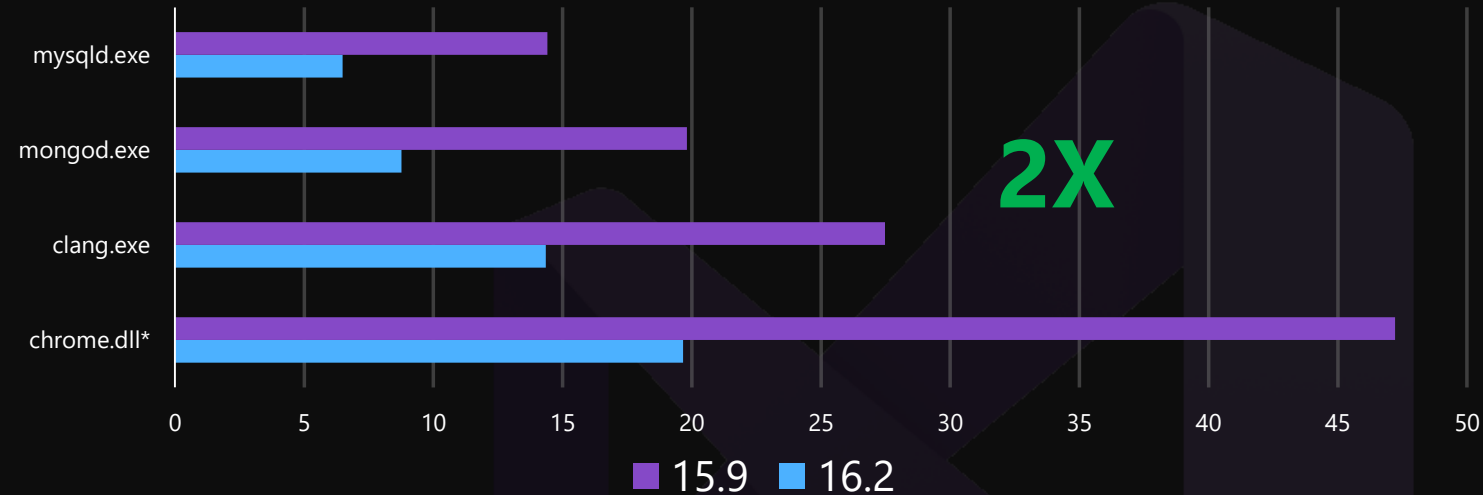
- Debug info pruning in compiler
- Type and global symbol hash changes in PDB

16.2

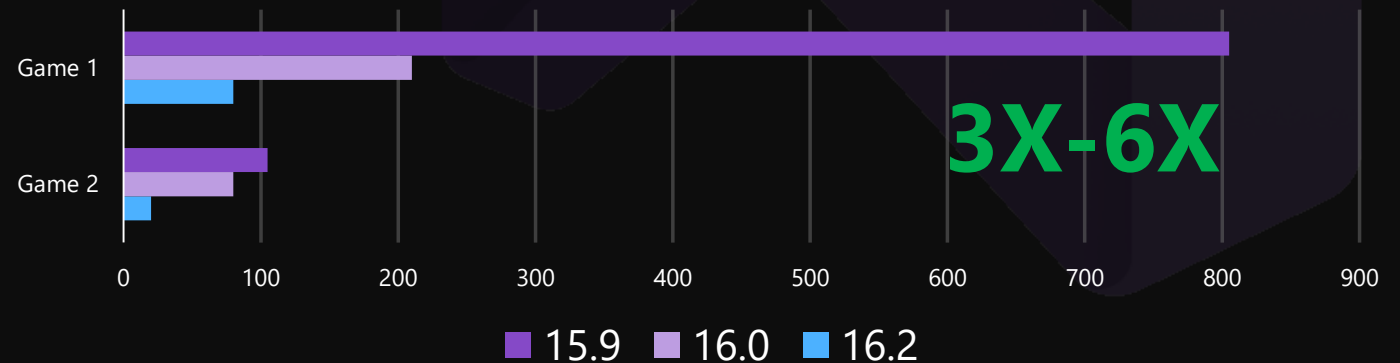
- Better cache performance & memory utilization
- Aggressive memoization of intermediate data

Build
throughput

OSS Projects **/debug:fastlink** Iteration Build Times (s)



AAA Game Link **/debug:full** Build Times (s)

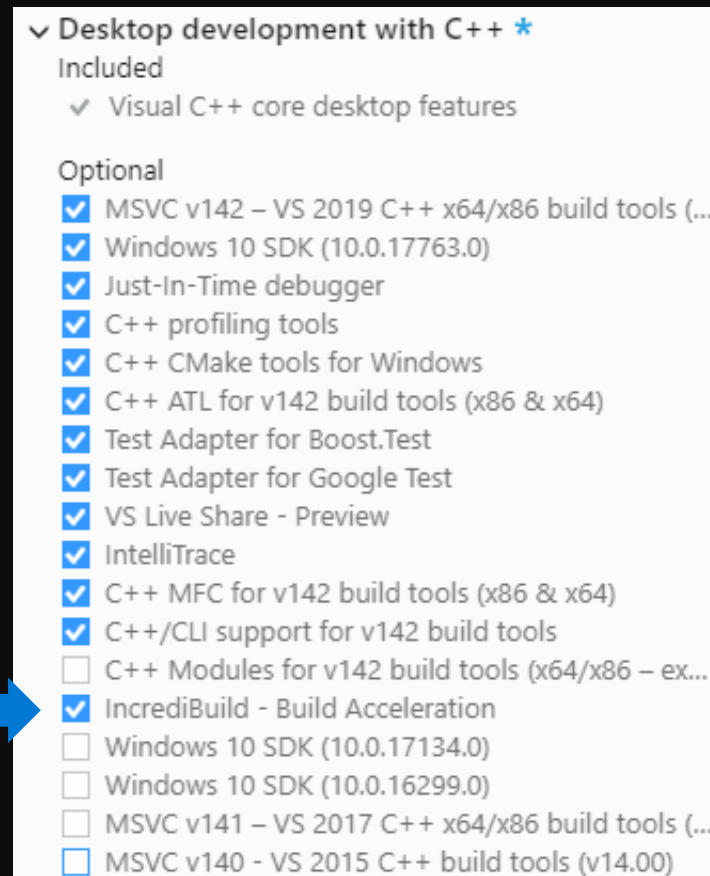


Visual Studio 2019 Build Acceleration

Speed up your local builds on up to 16 cores for free with **IncrediBuild**

Learn more at <https://aka.ms/cpp/ib>

Build
throughput



*Unreal Engine Infiltrator demo***Frame duration
%age improvement****VS 2019 vs. VS 2017****VS 2019 16.2 vs. VS 2017***Average*

0.7%

2.5%

Largest

2.8%

4-6%

More optimizer improvements in release mode

Augment SSA optimizer to understand vector (SIMD) code

Control-statement optimization improvements

Improvements to memset generation

Additional removals of redundant copies

OpenMP SIMD extension – with `/openmp:experimental`

C++ exception handling data size is 60% smaller, leading up to ~20% binary reduction when heavy exceptions are used – with `/d2FH4`

More aggressive inlining than the default `/Ob2` under `/O2` – with `/Ob3`

Learn more at <https://aka.ms/cpp/runtimeperf>

Concurrency checks

Use-after-move check off-by-default, part of C++ Core Check Experimental Rules

Coroutine checks

Lifetime profile checks off-by-default, part of C++ Core Check Lifetime Rules

Concurrency check example

```
bool HandleRequest(int Id, Request* request)
{
    EnterCriticalSection(&cs_);
    if (cache_.find(Id) != cache_.end())
        return false; // C26115: Failing to release lock 'this->cs_' in function
    cache_[Id] = request;
    LeaveCriticalSection(&cs_);
}
```



Wed 9/18 14:00 – 15:00

Lifetime analysis for everyone

Gábor Horváth and Matthias Gehre @ Aurora D



Code
Analysis

Coming soon

Address Sanitizer (ASAN) in MSVC

programming tool that detects memory corruption bugs such as buffer overflows or accesses to a dangling pointer (use-after-free).

Updated ASAN Library for Windows
MSVC Compiler support for ASAN
MSBuild & CMake integration
Debugger ASAN integration



```
delete[] array;  
return array[argc];
```

Exception Unhandled

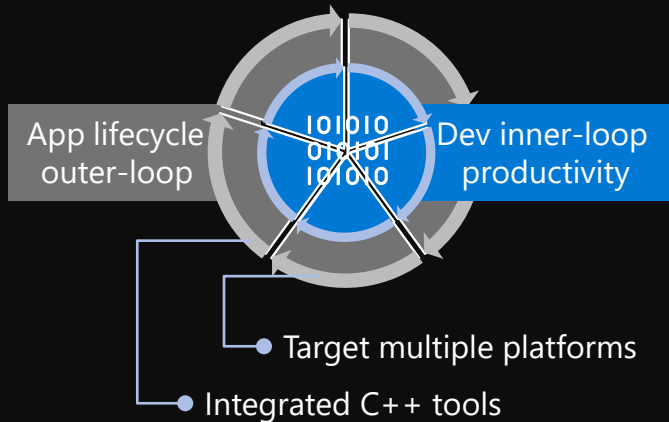
AddressSanitizer: heap-use-after-free on address 0x614000000044

Wed 9/18 14:00 – 15:00
C++ Sanitizers and Fuzzing for the Windows Platform Using New Compilers, Visual Studio, and Azure
Jim Radigan @ Summit 8/9

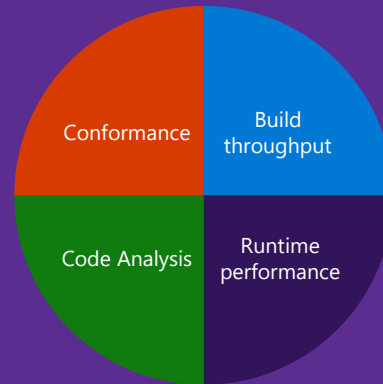
Visual Studio

Agenda

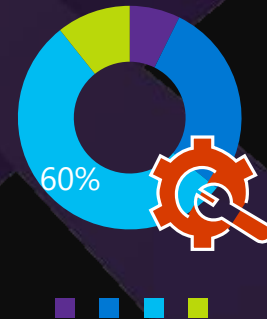
Visual Studio IDE



MSVC Compiler Toolset



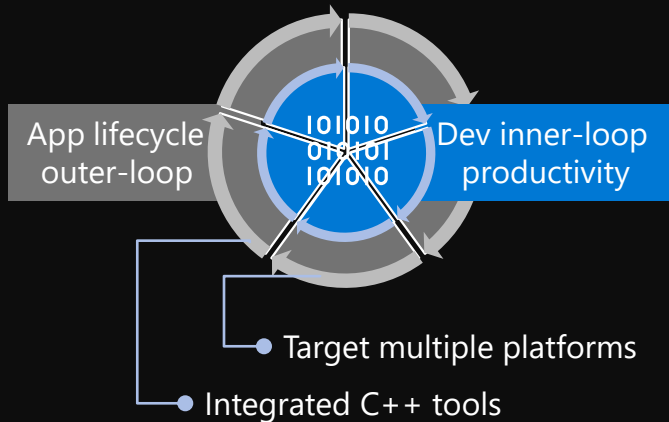
Future of C++ & Tooling



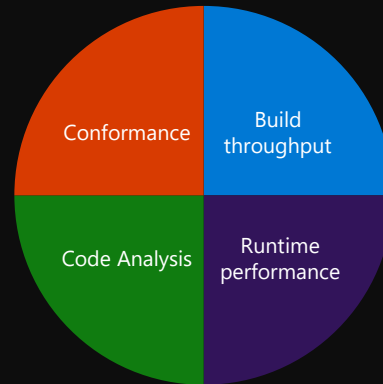
Visual Studio

Agenda

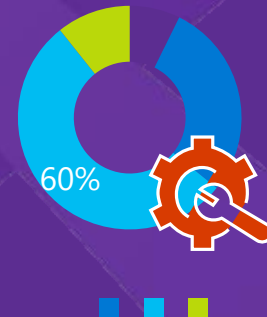
Visual Studio IDE



MSVC Compiler Toolset

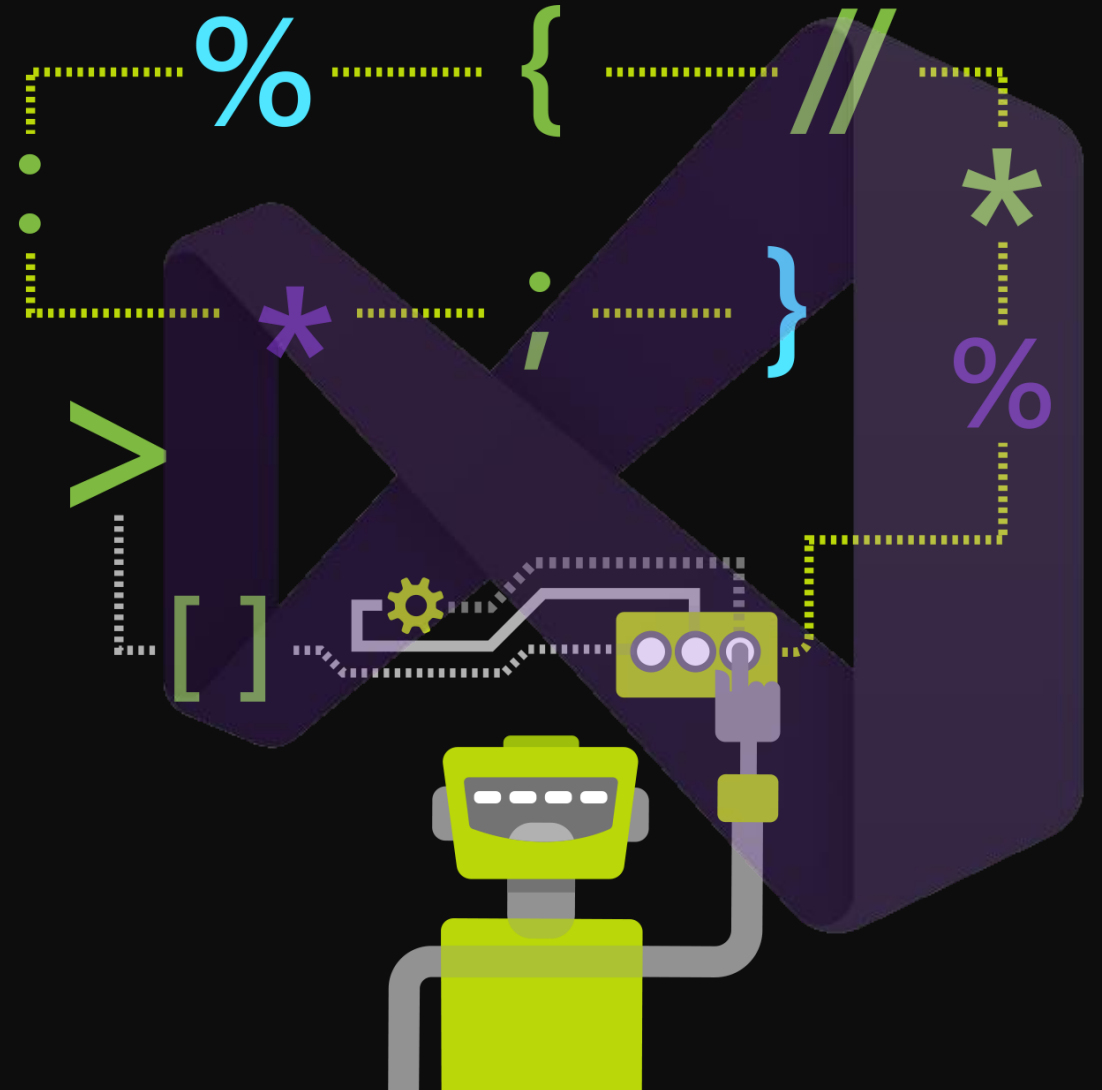


Future of C++ & Tooling



A Sneak Peek at the Future of Visual Studio

Demo



Developer Community – Most Popular Suggestions

Compiler Explorer libraries

```
1  #include <ctre.hpp>
2
3  static constexpr auto pattern = ctll::fixed_string("number of votes: [0-9]+");
4
5  bool example() {
6      constexpr bool matches = ctre::match<pattern>("number of votes: 48");
7      return matches;
8  }
```

A ▾

☐ 11010

☒ ./a.out

☒ .LX0: ☐

```
1  bool example(void) PROC
2      mov     al, 1
3      ret     0
4  bool example(void) ENDP
```


Developer Community – Most Popular Suggestions

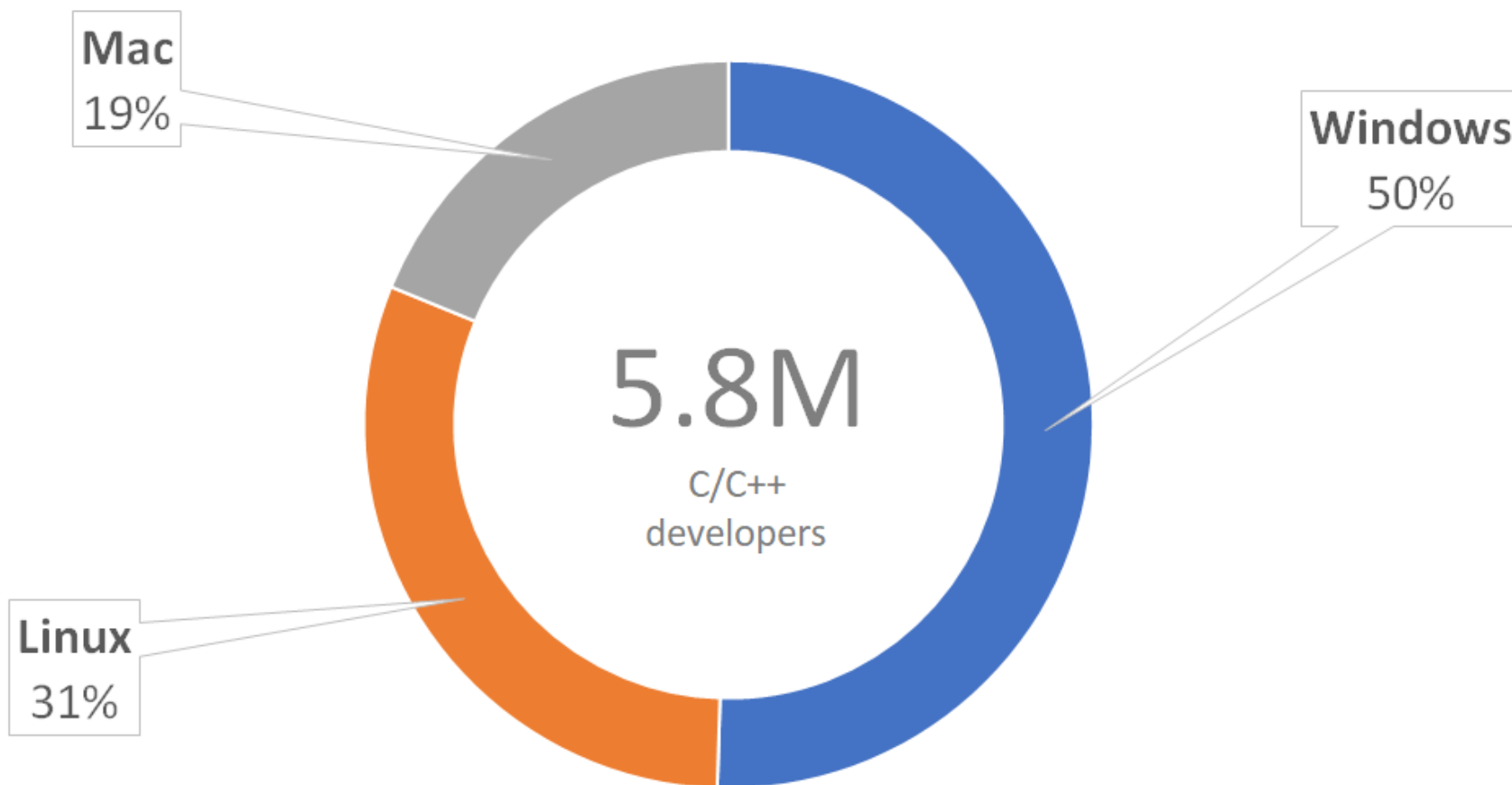
More OpenMP Features - SIMD

```
#pragma omp simd  
for (int i = 0; i < count; i++)  
{  
    a[i] = b[i] + 1;  
}
```





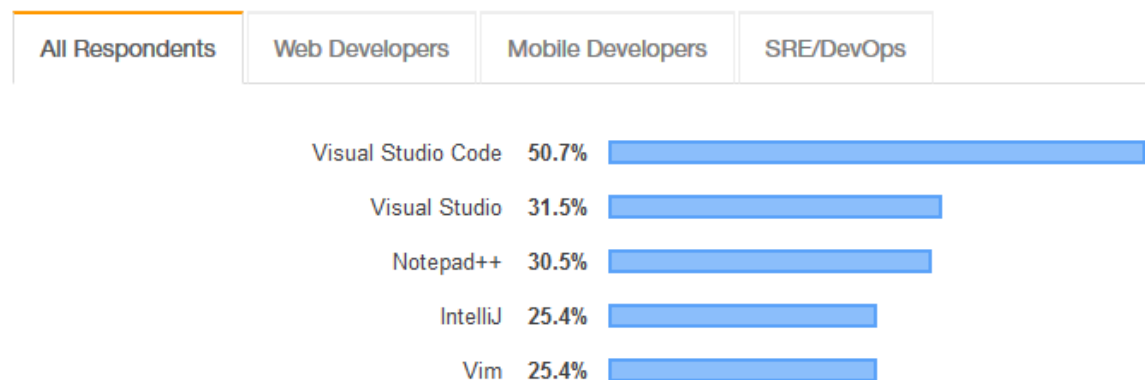
The C++ Community



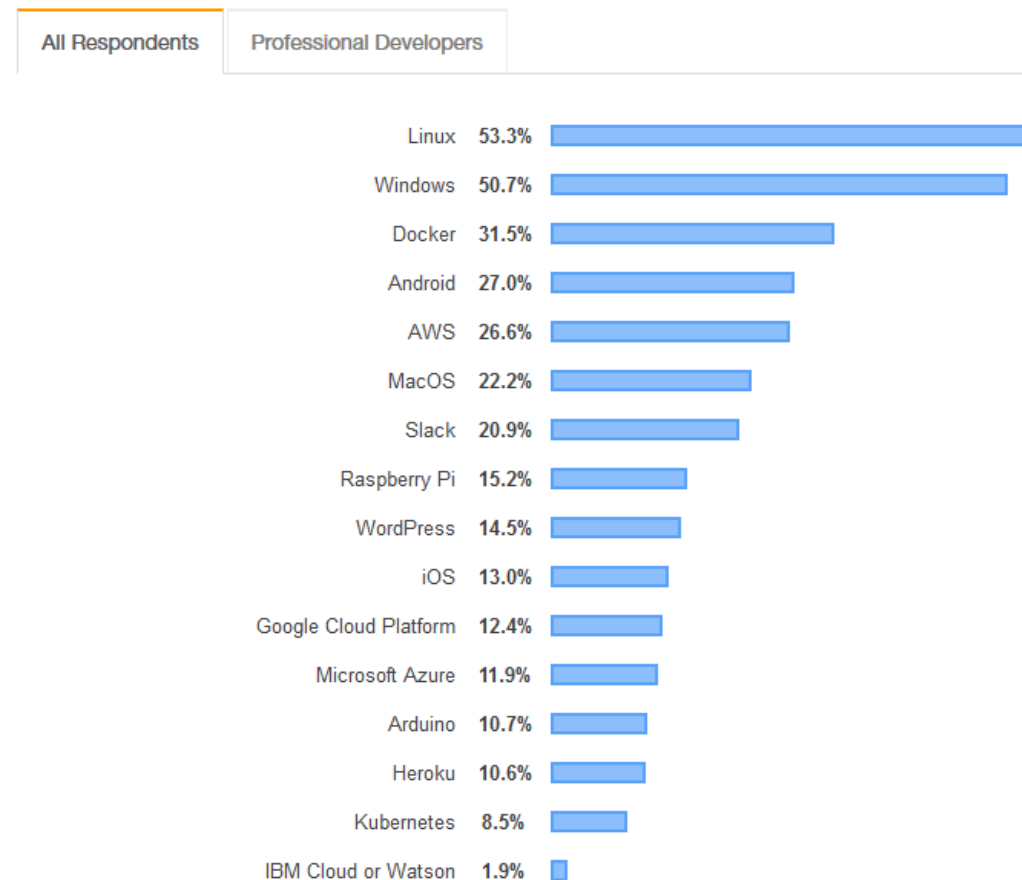
The C++ Community



Most Popular Development Environments



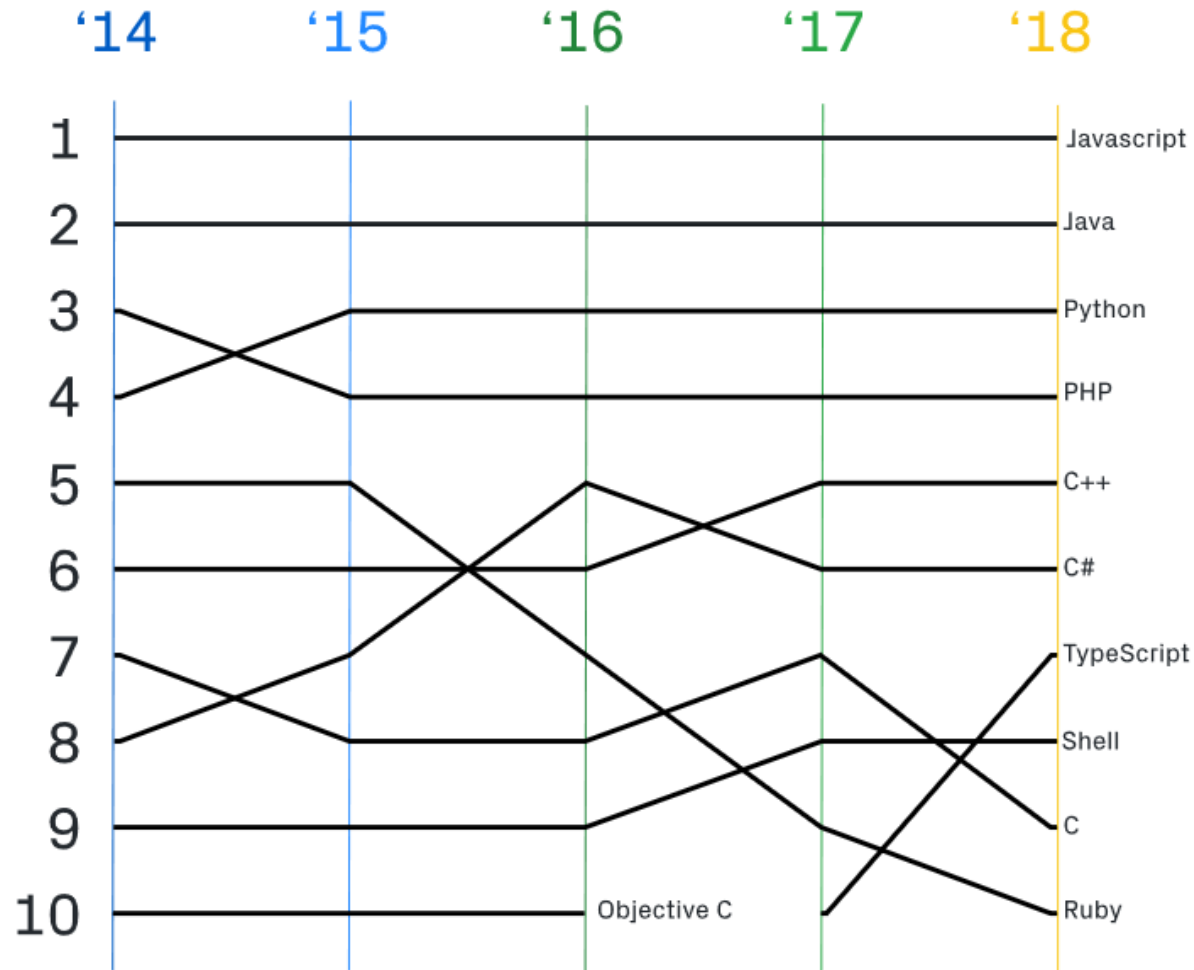
Platforms





The C++ Community

Most used languages by year

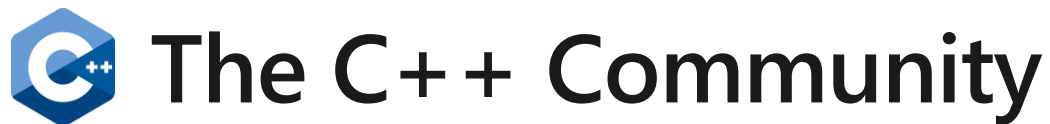


The State of the

Octoverse

Fastest growing languages

Growth in contributors	
1 Kotlin	2.6x
2 HCL	2.2x
3 TypeScript	1.9x
4 PowerShell	1.7x
5 Rust	1.7x
6 CMake	1.6x
7 Go	1.5x
8 Python	1.5x
9 Groovy	1.4x
10 SQLPL	1.4x



The C++ Community





Mission of the C++ product team at Microsoft

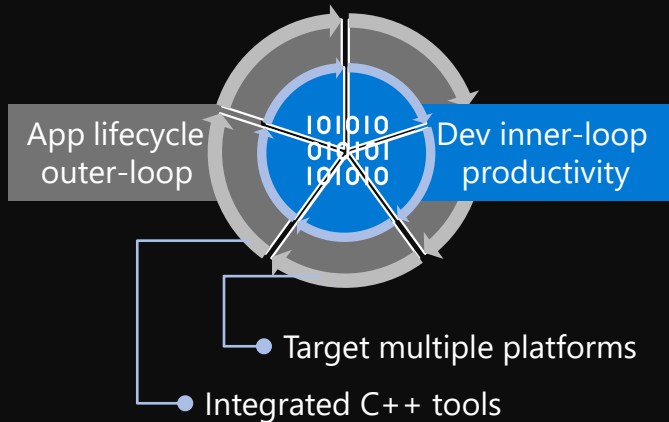
Make the lives of all C++ developers on the planet better

1. by participating in the creation of the **C++ Standards**
2. by investing in the Microsoft Visual C++ (**MSVC**) Compiler & Libraries
3. by simplifying C++ library acquisition via **Vcpkg**
4. by improving the **Visual Studio IDE**
5. by continuing to enhance the C++ extension for **Visual Studio Code**

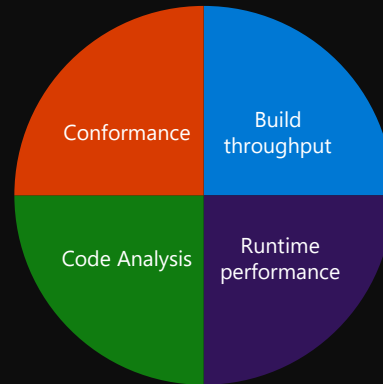
Visual Studio

Agenda

Visual Studio IDE



MSVC Compiler Toolset



Future of C++ & Tooling



Happy Coding!

Thank you

Other talks from Microsoft

Monday 16th

14:00 – 15:00: [Hello World From Scratch](#) by Sy Brand and Peter Bindels
15:15 – 16:15: [Programming with C++ Modules: Guide for the Working Programmer](#) by Gabriel Dos Reis
16:45 – 17:45: [Latest & Greatest in Visual Studio 2019 for C++ Developers](#) by Sy Brand and Marian Luparu

Tuesday 17th

15:15 – 15:45: [What's New in Visual Studio Code for C++ Development – Remote Development, IntelliSense, Build/Debug, vcpkg, and More!](#) by Tara Raj
15:50 – 16:10: [\(Ab\)using Compiler Tools Summit](#) by Réka Kovács
15:50 – 16:10: [C++ Standard Library "Little Things"](#) by Billy O'Neal
15:50 – 16:10: [Upgrade from "permissive C++" to "modern C++" with Visual Studio 2019](#) by Nick Uhlenhuth

Wednesday 18th

09:00 – 09:30: [How to Herd 1,000 Libraries](#) by Robert Schumacher
14:00 – 15:00 [C++ Sanitizers and Fuzzing for the Windows Platform Using New Compilers, Visual Studio, and Azure](#) by Jim Radigan
14:00 – 15:00 [Lifetime analysis for everyone](#) by Gábor Horváth and Matthias Gehre
16:45 – 17:45: [Killing Uninitialized Memory: Prot](#) by Joe Bialek and Shayne Hiet-Block

Thursday 19th

15:15 – 15:45: [Don't Package Your Libraries, Write Packagable Libraries! \(Part 2\)](#) by Robert Schumacher
16:45 – 17:45: [Floating-Point charconv: Making Your Code 10x Faster With C++17's Final Boss](#) by Stephan T. Lavavej

Friday 20th

16:15 – 18:00: [De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable \("Simplifying C++" #6 of N\)](#) by Herb Sutter



Monday 9/16 16:45
Summit 4/5

Latest and Greatest in Visual Studio for C++ Developers

Marian Luparu @mluparu
Sy Brand @TartanLlama

@VisualC
<https://aka.ms/cpp>