Italian C++ Community

www.italiancpp.org

The joys and trials of writing cross-platform C++ code Let tools help!

Marc Goodner

Italian C++ Conference 2019
June 15, Milan

Thank you to Italian C++ Sponsors!

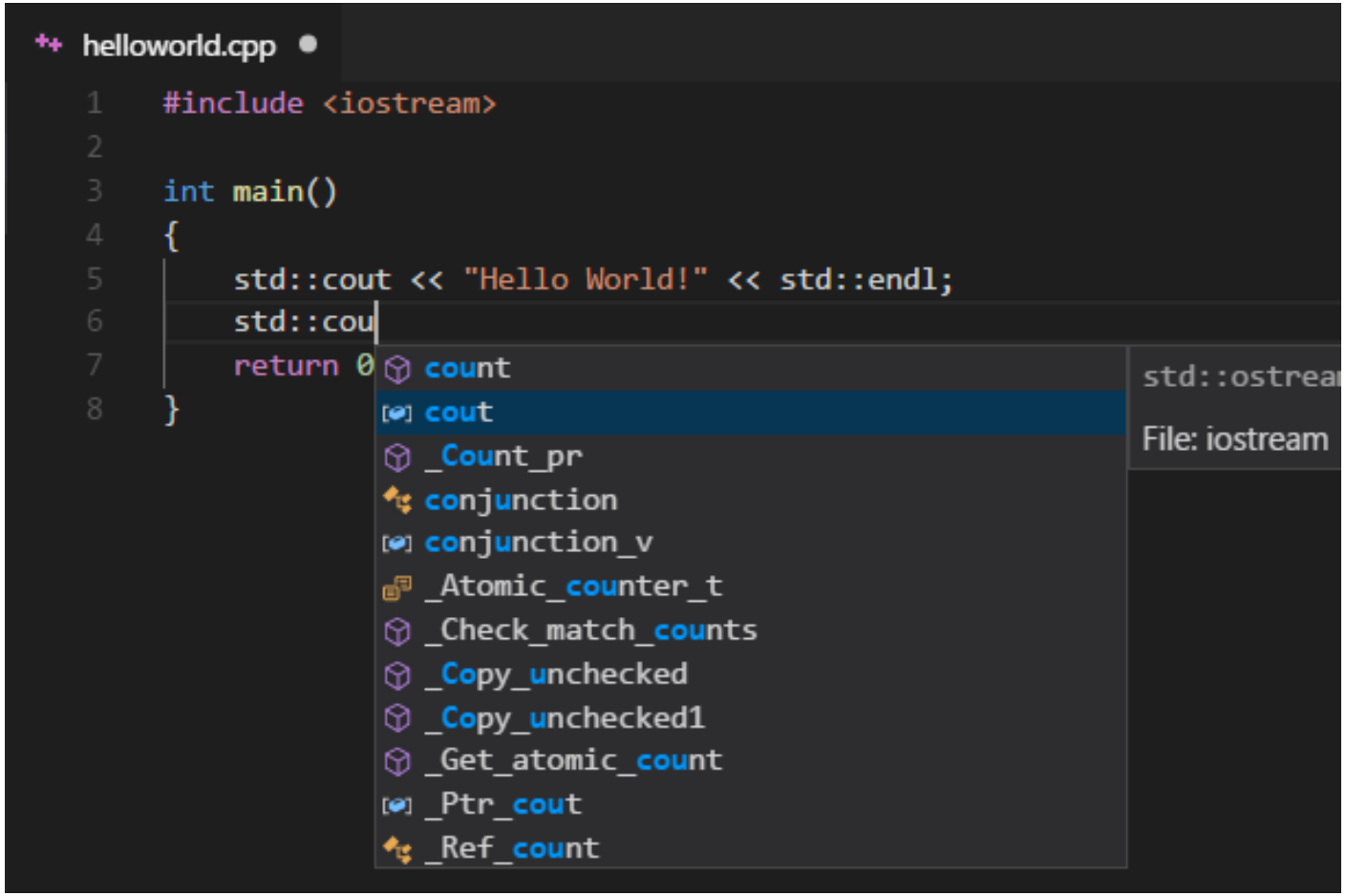
HOST

PATRON

Community Crumbs

SPONSORS

Visual Studio Code



Free, open source code editor

Runs on Windows, macOS, and Linux

#1 most used code editor *(StackOverflow Developer Surveys 2018, 2019)*

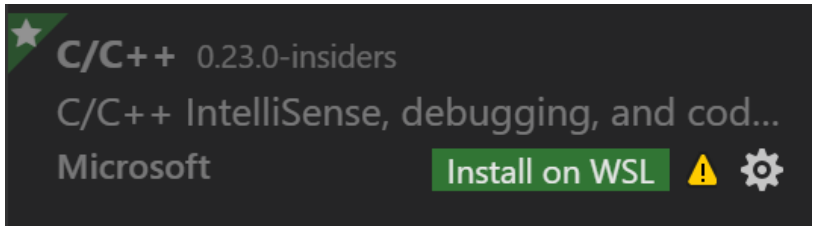
Microsoft C/C++ extension available:

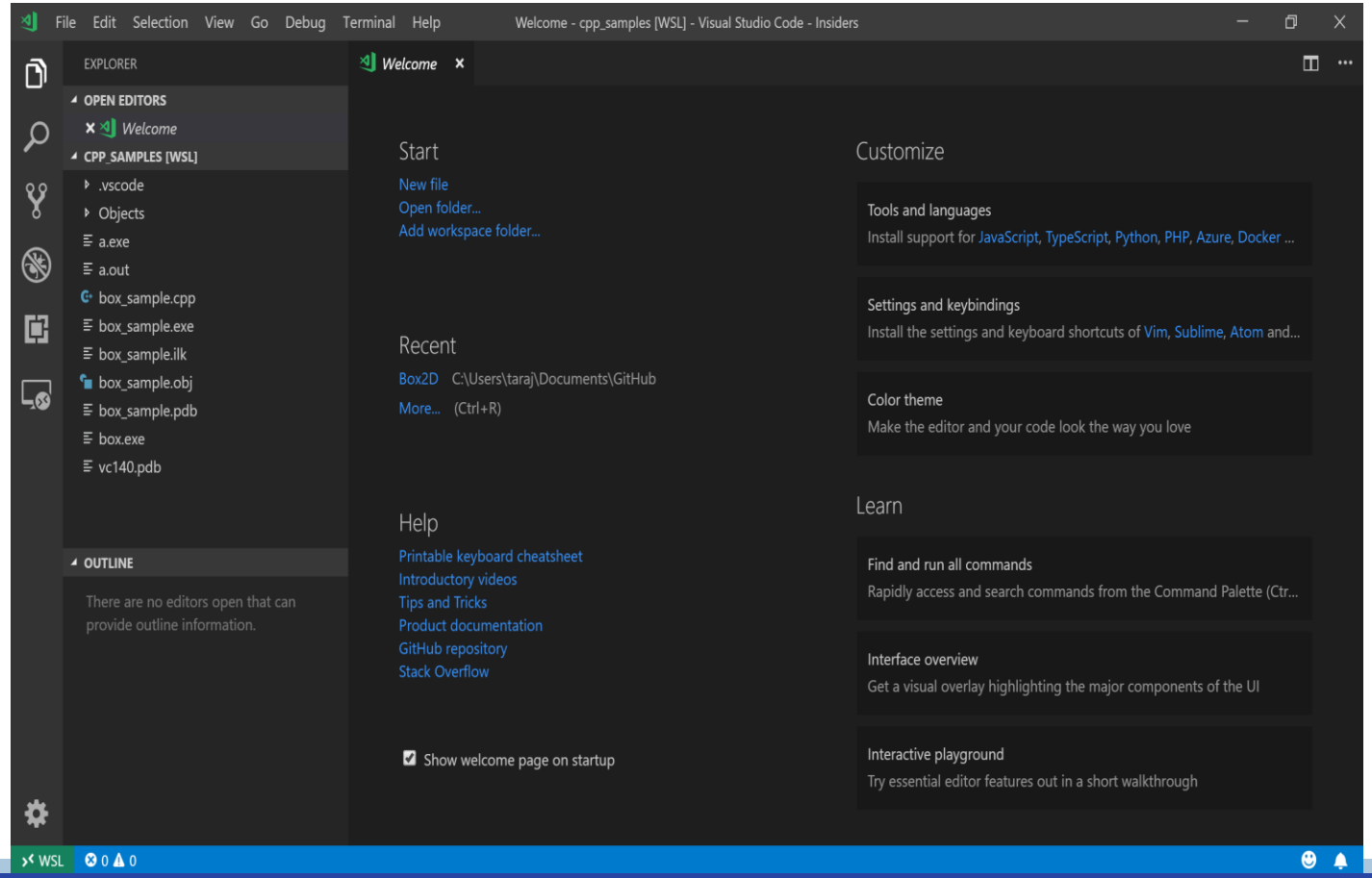
- IntelliSense
- Debugging
- Code browsing

https://code.visualstudio.com/

Remote Development with Visual Studio Code

NOW AVAILABLE with the C/C++ extension

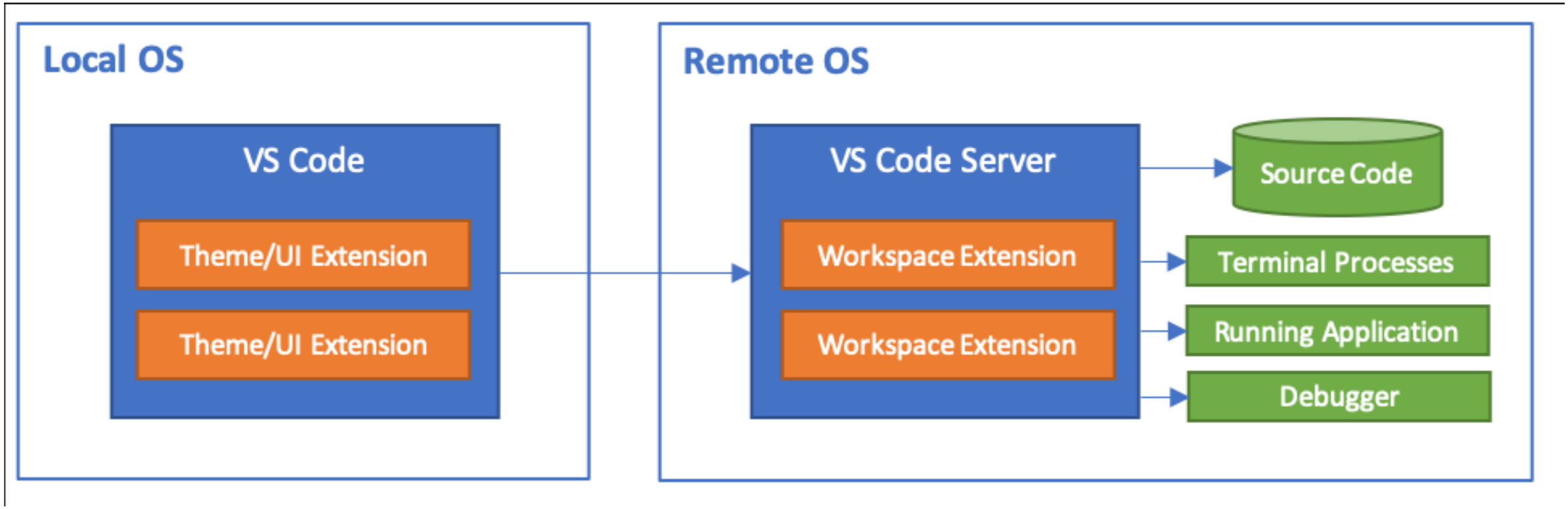




You can

- Easily develop your C/C++ programs on the same operating system you are deploying to
- Sandbox your development environment
- Use runtimes not available on your local OS
- Access an existing environment from multiple locations
- Debug an application running somewhere else

Remote Development – How it Works



Italian C++ Community

www.italiancpp.org

Demo

Remote Cross-platform development

Italian C++ Conference 2019
June 15, Milan

Key takeaways from VS Code Remote demo

With minimal setup you can **enable VS Code to work with remote environments** from your host

Remember to install the **C++ extension** and any others you need into the VS Code remote instance

Key takeaways from Visual Studio Code demo

- Visual Studio Code's C/C++ extension comes with features designed to improve your productivity
- It is easy to configure, build, run, and debug your first program
- You can use Visual Studio Code on Linux, macOS, as well as Windows

Visual Studio Code with C/C++ extension

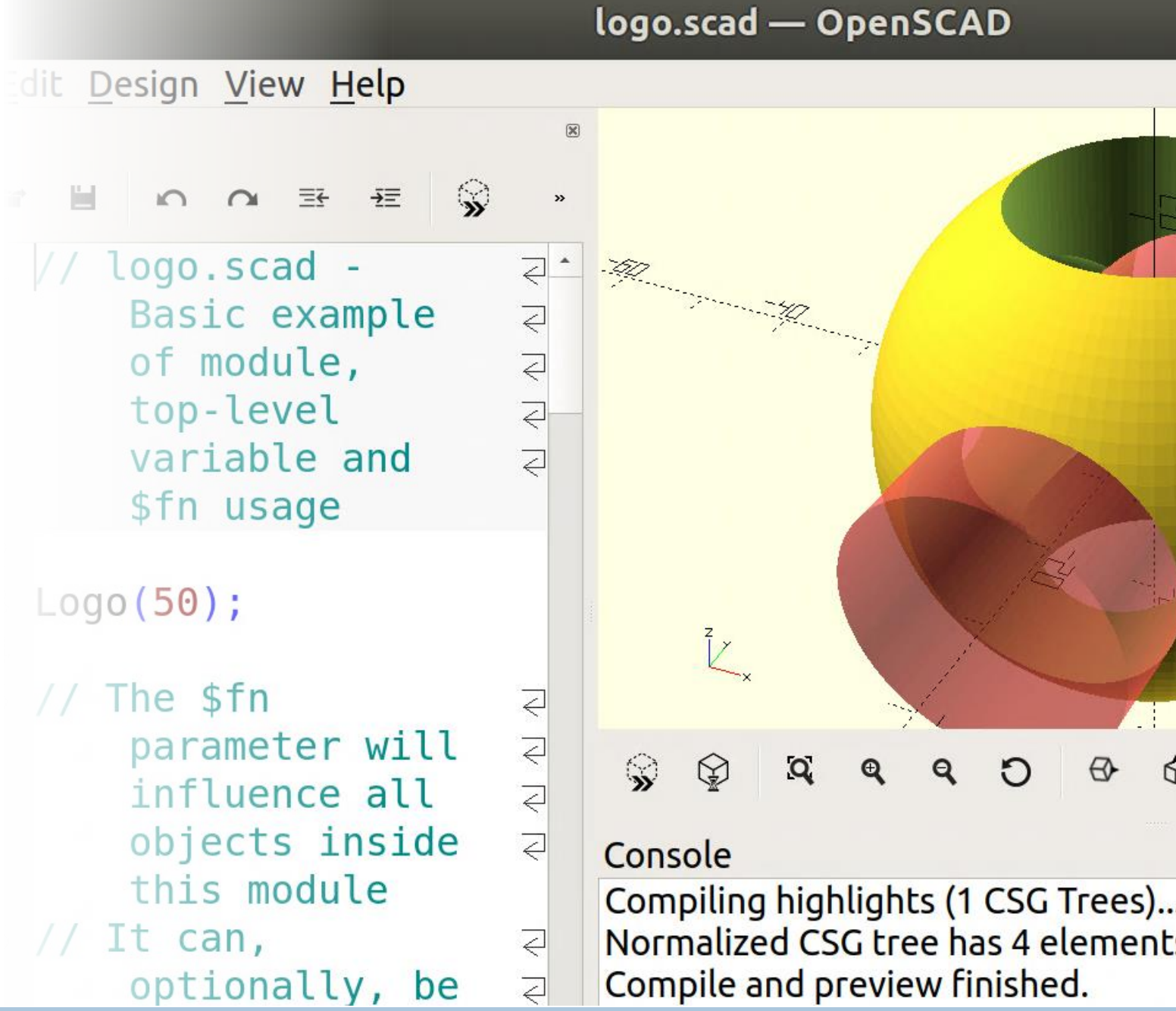
Existing Functionality

- Edit, build, run, debug
- Command palette
- IntelliSense completion
- Go to Definition

NEW in Visual Studio Code (Jan – May)

- Doc comments for hover, completion, and signature help
- Improved member function completion
- Improved IntelliSense performance
- Build and debug active file
- Configuration Settings Editor
- Squiggles for malformed configurations

Let’s look at working together when you can’t build on the target…



Italian C++ Community

www.italiancpp.org

Demo

Cross-platform collaboration

Italian C++ Conference 2019
June 15, Milan

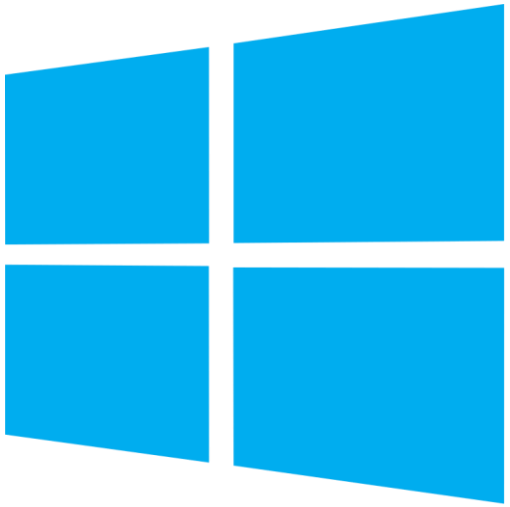
Key takeaways from OpenSCAD demo

There are a lot of cross platform strategies, not every one that targets Windows runs on Windows

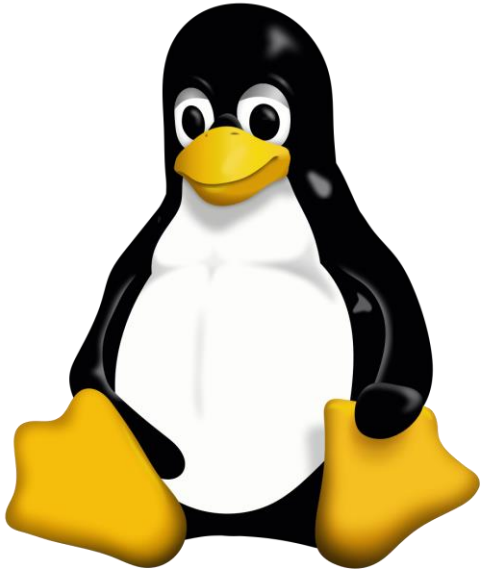
Use **LiveShare** to collaborate with peers across platforms and Visual Studio and Visual Studio code

WSL is a swiss army knife of getting things done with Linux from Windows

Visual Studio: One IDE for multiple platforms



Windows



Linux
Remote machine or WSL

Italian C++ Community

www.italiancpp.org

Demo

Getting Started with Visual Studio 2019 for Windows and Linux

Italian C++ Conference 2019
June 15, Milan

Key takeaways from WSL demo

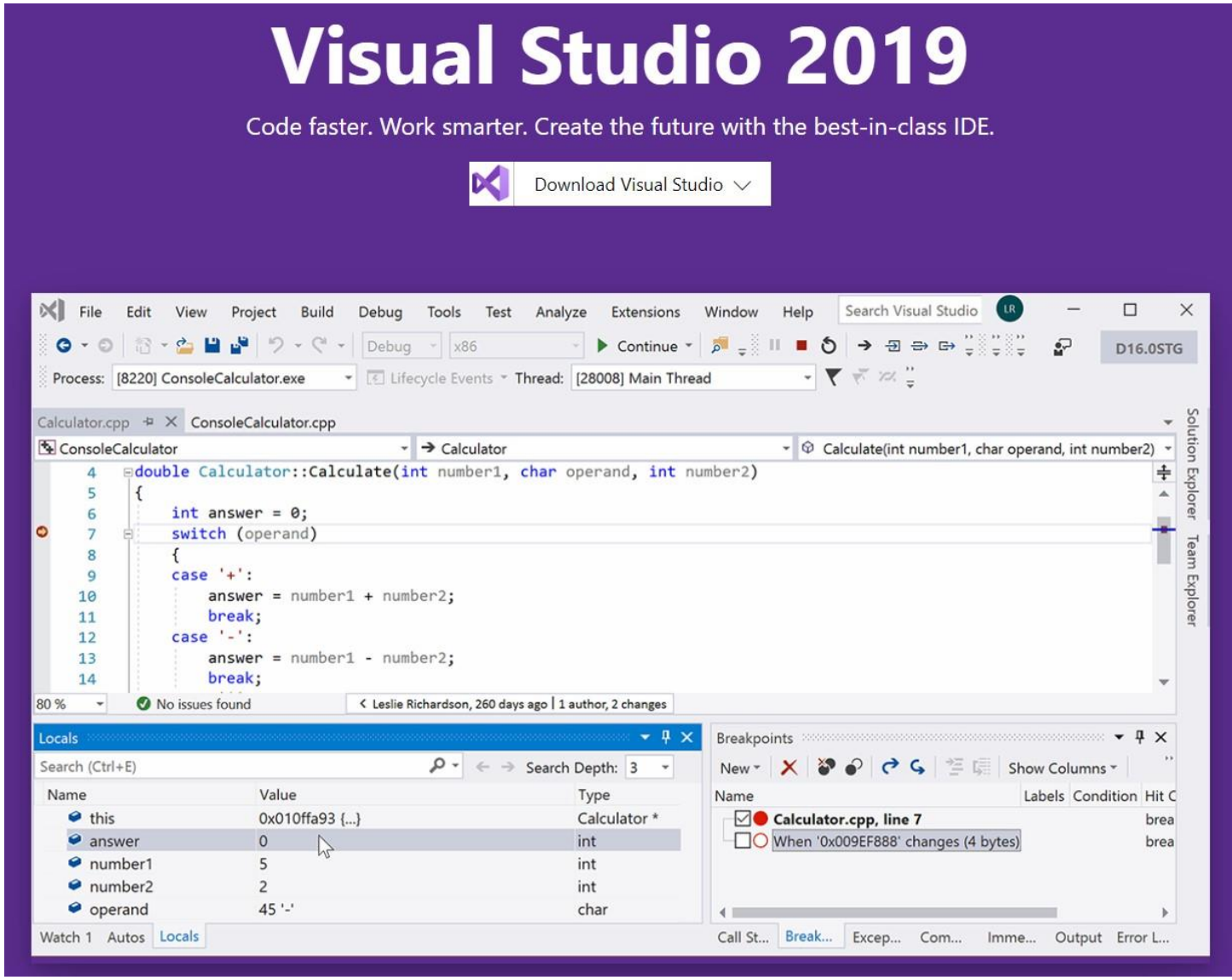
Doing cross-platform development? Try **CMake** with Visual Studio.

You can **target both Windows and Linux** seamlessly from **Visual Studio** on Windows.

Try out the **Windows Subsystem for Linux integration** for an easy way to get started without needing a separate Linux system.

SSH connect to a remote Linux system when you're ready to move to a larger project.

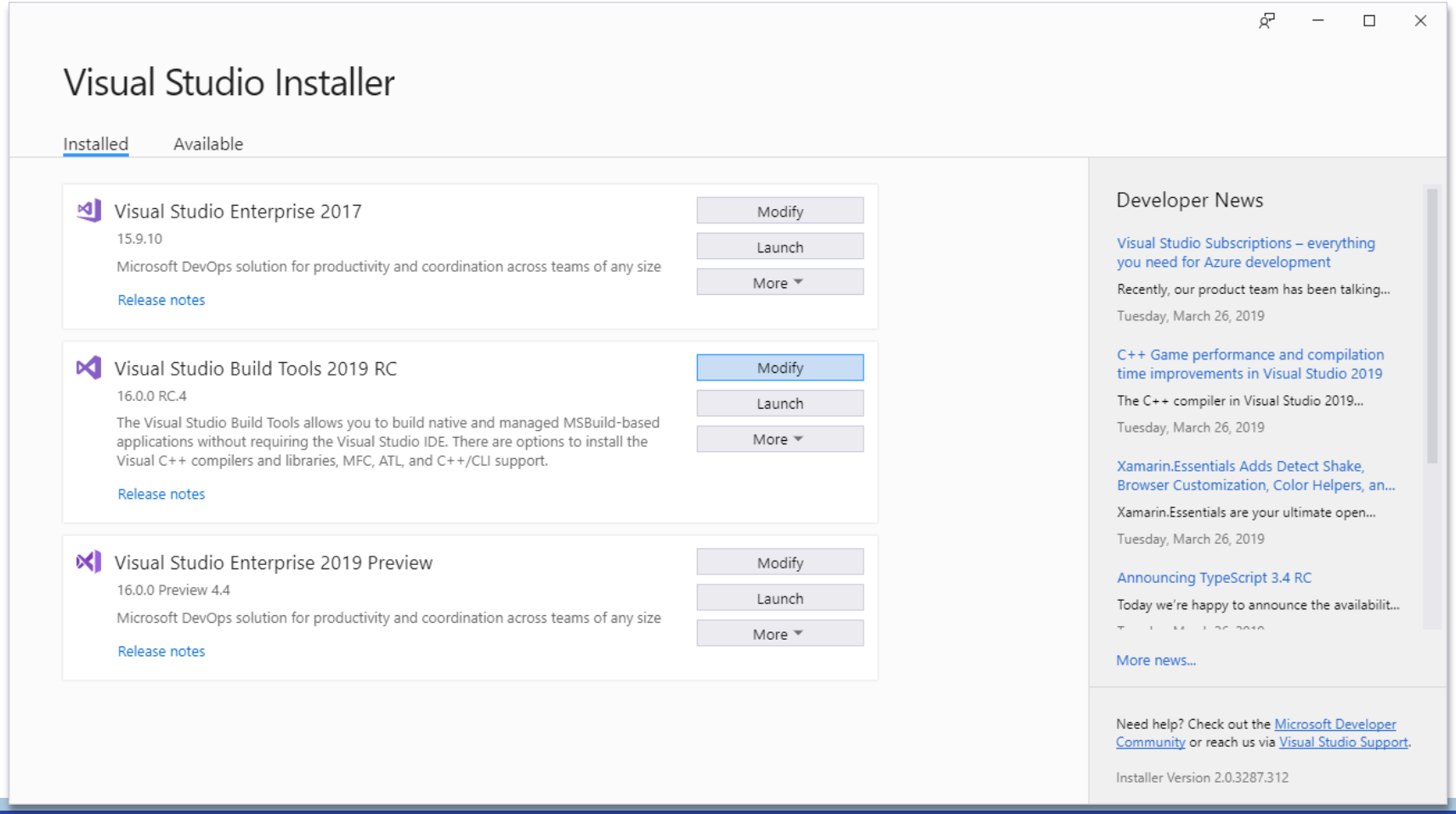
Visual Studio 2019 now available



Pain-free upgrades

Learn more at https://aka.ms/cpp/upgrade/2019

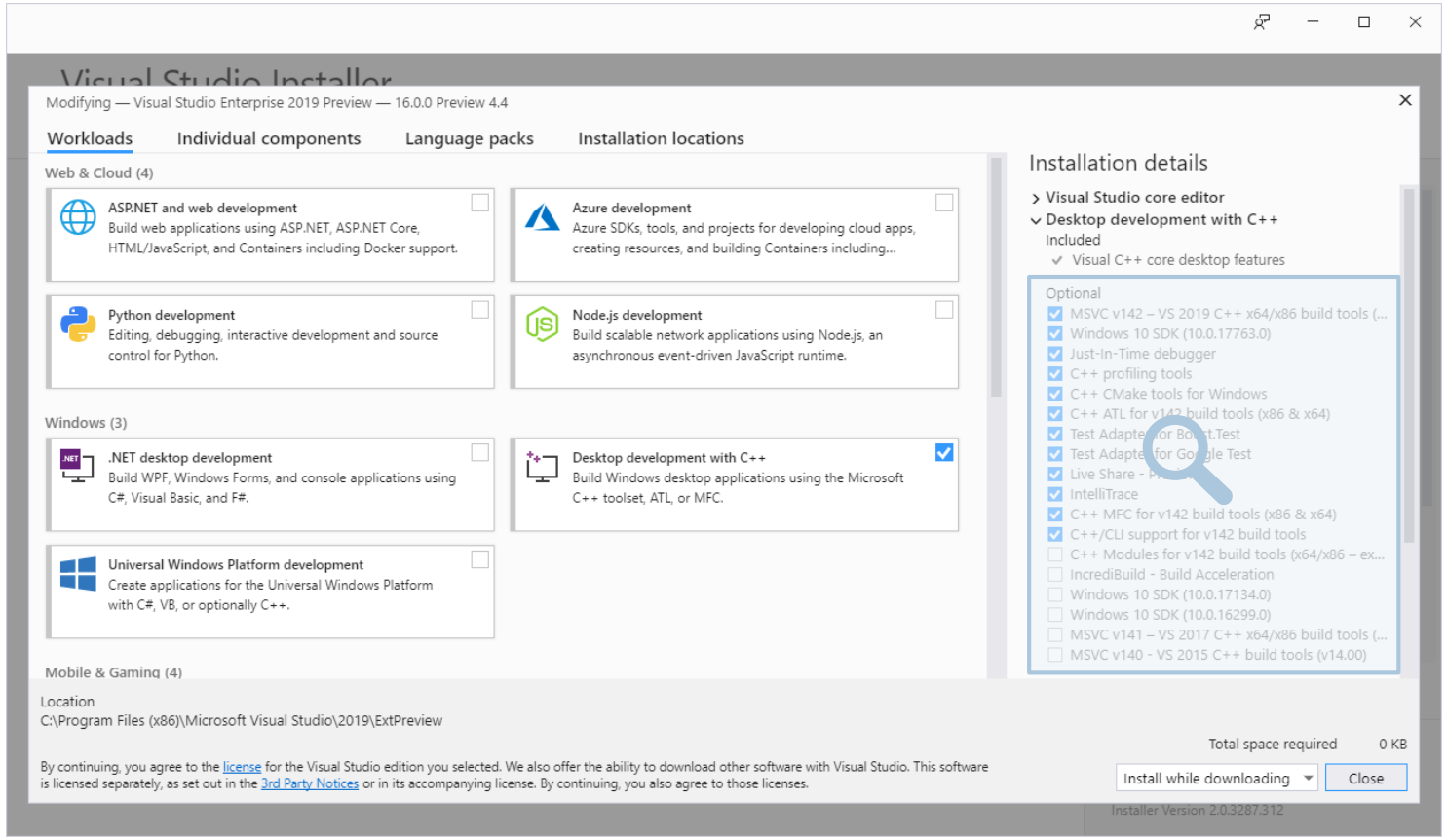
1. Install the latest IDE **side-by-side** with your current one

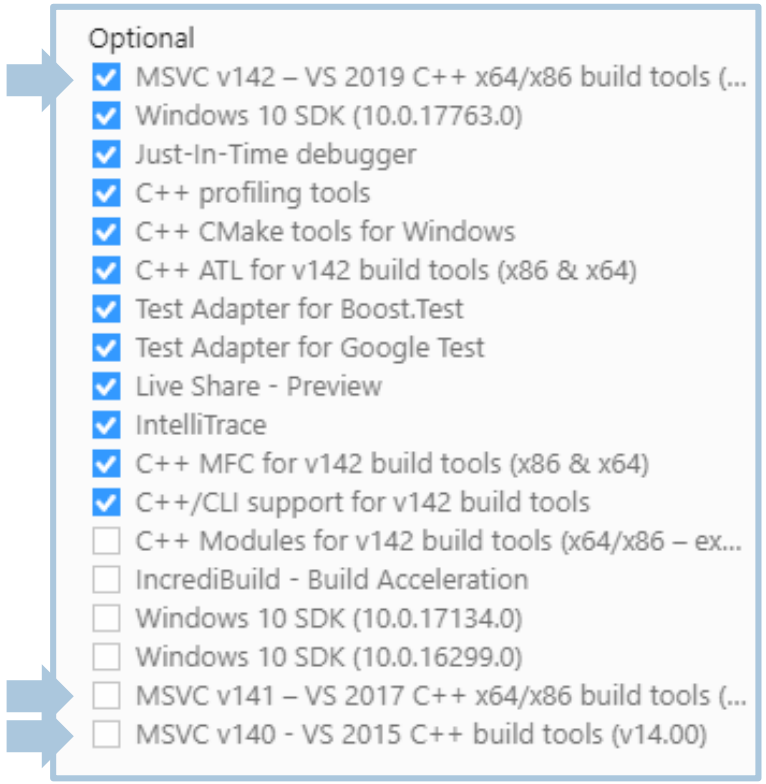


Pain-free upgrades

Learn more at https://aka.ms/cpp/upgrade/2019

1. Install the latest IDE **side-by-side** with your current one
2. Use **your preferred MSVC toolset** inside the Visual Studio 2019 IDE





Pain-free upgrades

Learn more at https://aka.ms/cpp/upgrade/2019

1. Install the latest IDE **side-by-side** with your current one
2. Use **any MSVC toolset** inside the Visual Studio 2019 IDE
3. Maintain **binary compatibility** with 3rd party binaries when you upgrade to the latest MSVC toolset

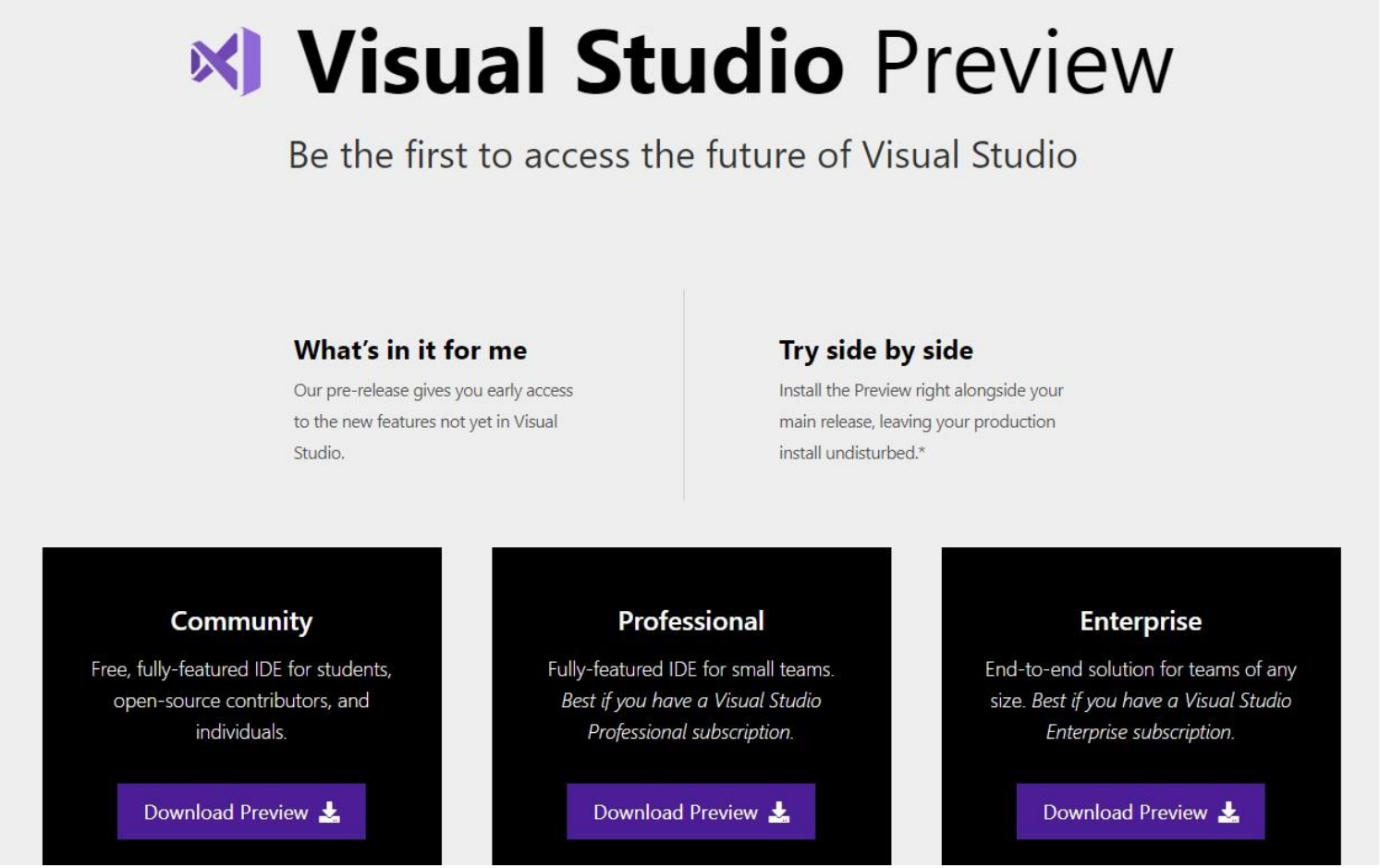
Pain-free upgrades

Learn more at https://aka.ms/cpp/upgrade/2019

1. Install the latest IDE **side-by-side** with your current one
2. Use **any MSVC toolset** inside the Visual Studio 2019 IDE
3. Maintain **binary compatibility** with 3rd party binaries when you upgrade to the latest MSVC toolset
4. Access the **full collection of OSS libraries** available in vcpkg

```
C:\src> git clone https://github.com/Microsoft/vcpkg.git & cd vcpkg
C:\src\vcpkg> .\bootstrap-vcpkg.bat
C:\src\vcpkg> .\vcpkg integrate install
C:\src\vcpkg> .\vcpkg install curl
```

Preview upcoming Visual Studio 2019 features

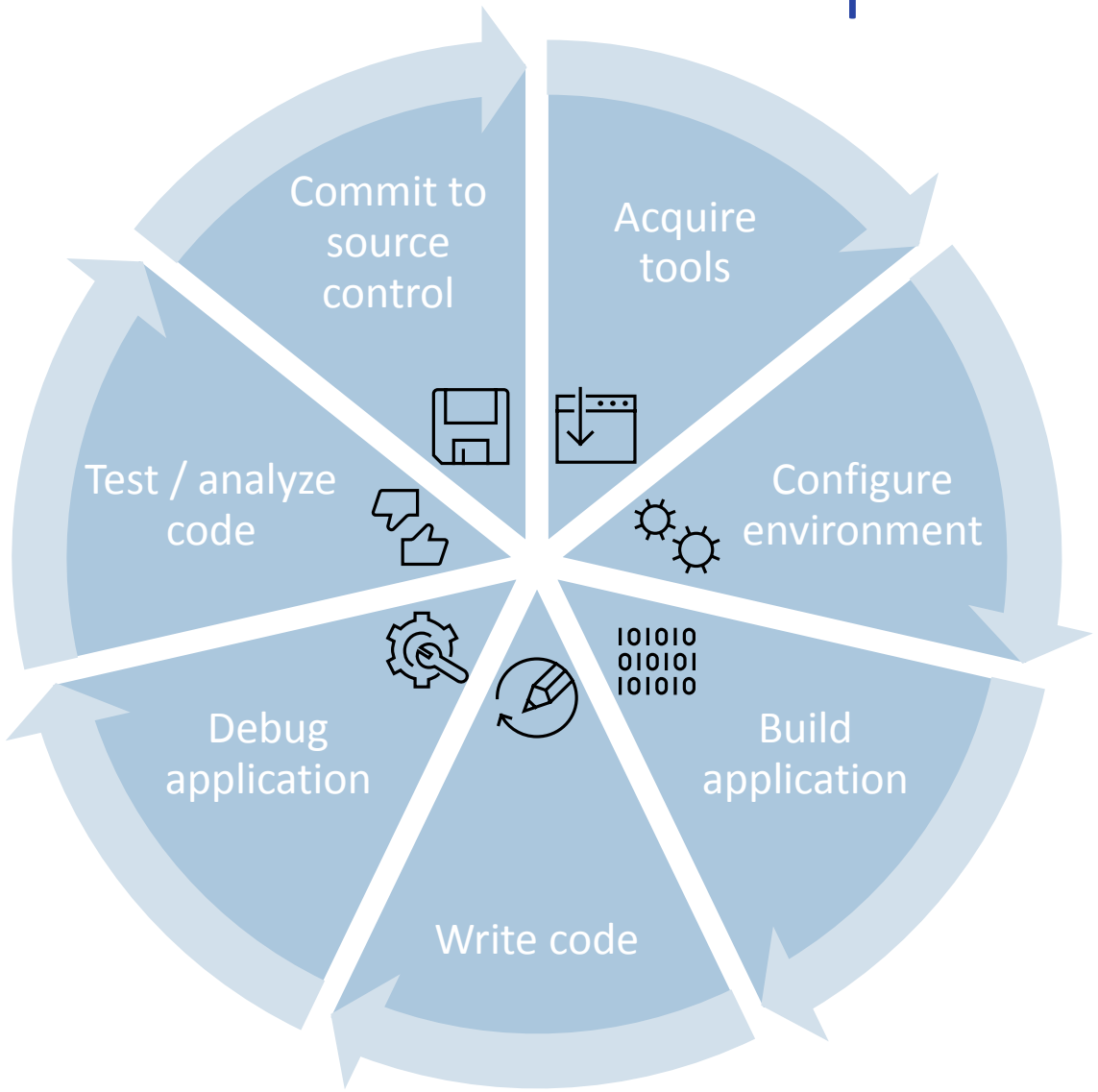


16.2 Preview 2 now available

Access the latest features as soon as they're available

https://visualstudio.microsoft.com/vs/preview/

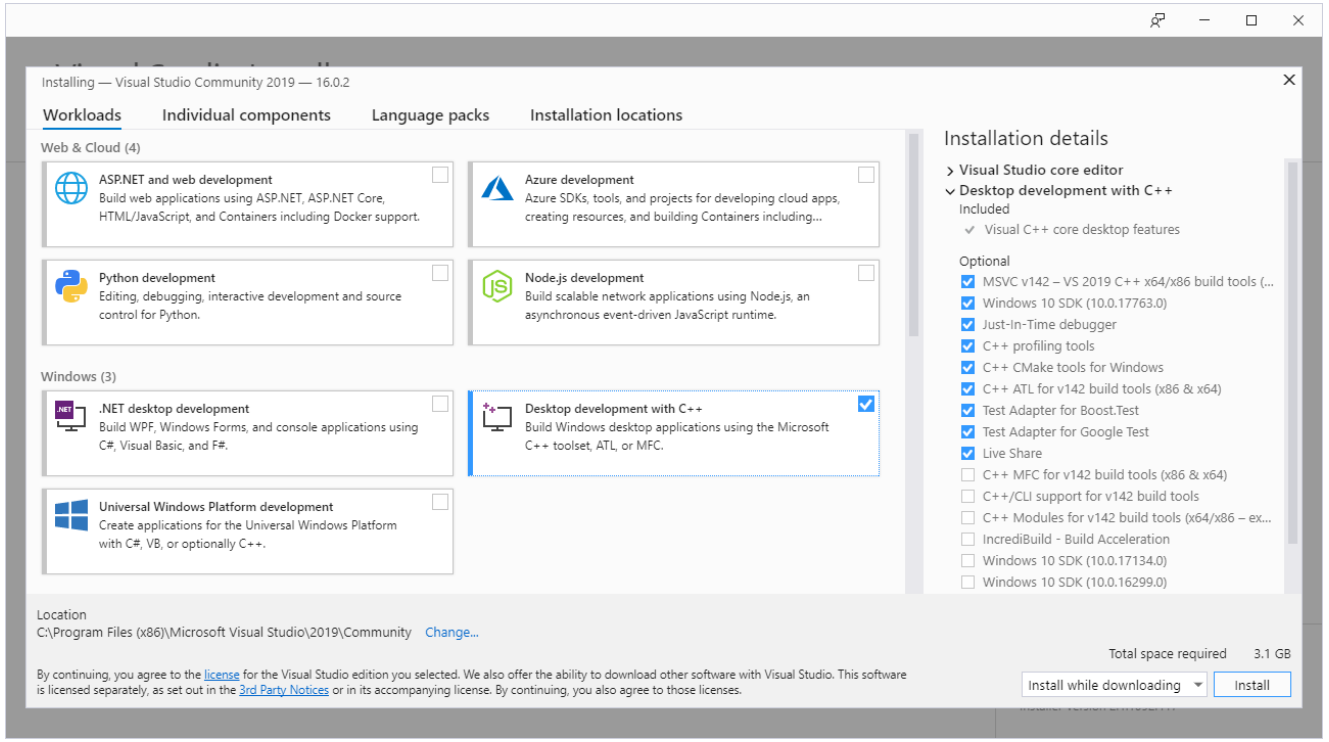
A software development workflow



with **Visual Studio 2019**

1. Acquire tools

Visual Studio 2019 (now available!)



Powerful productivity features

Built-in build tools

Revolutionary debugger

Highly extensible

Minimal configuration

https://visualstudio.microsoft.com/

1. Acquire tools

Vcpkg

Cross-platform C++ library manager

Operating system	# of libraries in vcpkg catalog (x64)
Windows	924
macOS	664
Linux	697

To get started:

```
> git clone https://github.com/Microsoft/vcpkg.git
> cd vcpkg

PS> .\bootstrap-vcpkg.bat
Linux:~/$ ./bootstrap-vcpkg.sh
```

Hundreds of libraries in catalog

https://github.com/Microsoft/vcpkg

2. Configure environment

Feature Area	Existing Functionality	**NEW** to Visual Studio 2019
Build system support	Native MSBuild and CMake support Bring your own build system in Open Folder	CMake 3.14 support Improved CMake configuration performance
Library integration	Vcpkg integration for MSBuild	Vcpkg integration for CMake
Codebase configuration	MSBuild project templates CMake Project template	CMake Settings Editor CMake config error squiggles Separate build and deploy targets
Connecting to remote systems	Remote Linux IntelliSense Remote build, run, and debug	Windows Subsystem for Linux integration Logging for remote connections

2. Configure environment

Time to load code (IntelliSense + CMake generation):

Visual Studio 2017 version 15.9	Visual Studio 2019 version 16.1	Improvement
2:58 mins	**1:26 mins**	**2x**

Tested with LLVM project + CMake

3. Build application

Conformance	Security
MSVC is the best compiler for targeting Windows.	
Performance	Code analysis

3. Build application

MSVC v142 toolset now available

- Binary compatible with v141 & v140 toolsets
- C++17 standard conformant compiler with `/permissive-` switch and experimental preprocessor
- Most complete C++17 standard library implementation

MSVC toolsets shipped with Visual Studio 2019		
First shipped in	**Toolset name**	**Minor versions**
Visual Studio 2015	v140	v14.00
Visual Studio 2017	v141	14.16
Visual Studio 2019	v142	v14.20+

- Preview in-progress C++ language features with `/std:c++latest` switch
- OpenMP SIMD extension with `/openmp:experimental`

3. Build application

MSVC v142 performance improvements

Frame duration improvement	VS 2019 vs. VS 2017
Average	0.7%
Largest	2.8%

Code runtime

Unreal Engine Infiltrator demo

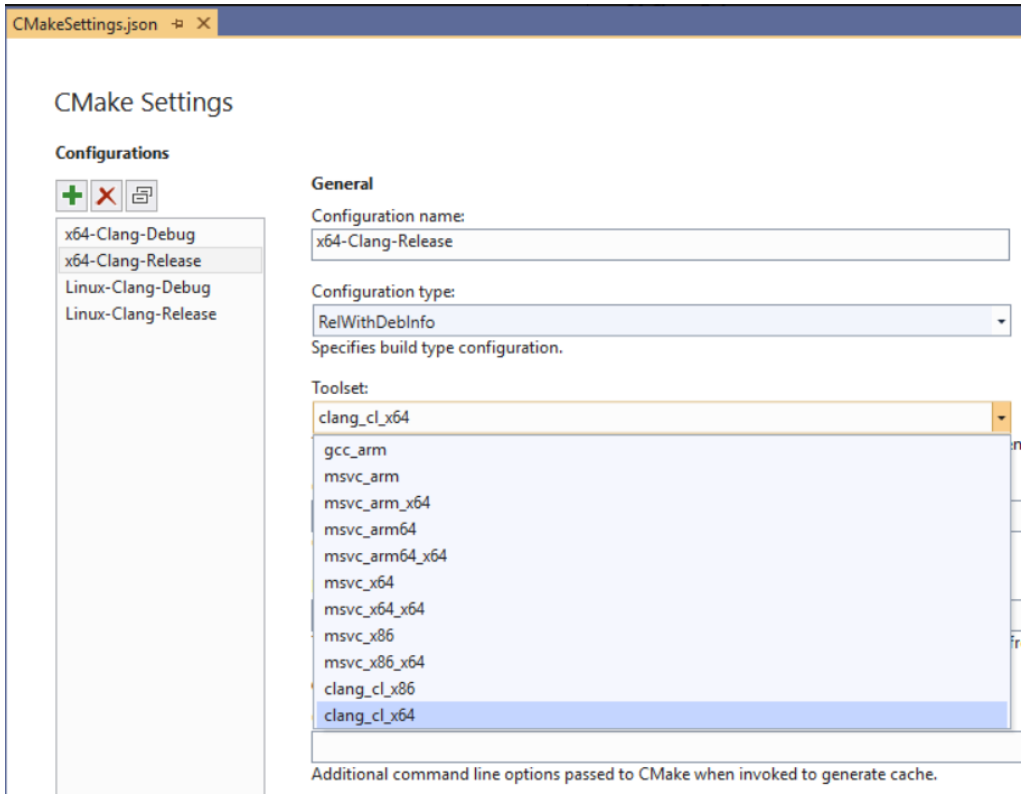
Link time

Unreal Engine AAA game (s)

Debug configuration	VS 2017 (15.9)	VS 2019 (16.0)	Speedup
/DEBUG:full	392.1s	163.3s	2.40x
/DEBUG:fastlink	72.3s	31.2s	2.32x

Clang/LLVM Support in Visual Studio 2019

Edit, build, and debug CMake projects with Clang and Visual Studio



- MSBuild support coming soon
- Default Clang build configurations available
- Install Clang for Windows from Visual Studio installer…

  …or bring your own

Let’s look at a larger codebase…

Italian C++ Community

www.italiancpp.org

Demo

Cross-platform development and productivity features with Visual Studio 2019

Italian C++ Conference 2019
June 15, Milan

Key takeaways from SuperTux demo

You don't need to compile all your 3rd party libraries by hand. Get them through **vcpkg**, our cross-platform library acquisition tool.

Take advantage of Visual Studio 2019 **productivity** features whether you're targeting Windows or Linux, using MSBuild or CMake.

Use Visual Studio's **rich debugging features** on Windows while targeting your Linux machine.

4. Write code

Existing Functionality

- Go To (Ctrl+T)
- Find All References
- Rename
- Go To Definition (F12)
- Extract Function
- View Call Hierarchy
- IntelliSense completion
- Platform-specific IntelliSense

NEW to Visual Studio 2019

- Template IntelliSense (find all instantiations)
- IntelliCode in-box
- Live Share in-box
- Quick Info colorization
- Online search links from Quick Info
- Improved VS Search
- New Quick Fixes
  - Add missing #include, add “using namespace”
  - Add missing semicolon, * to & and & to *
  - NULL to nullptr, uninitialized variable, uninitialized memory
- Go to Document with F12
- CMake in-editor documentation

5. Debug application

Existing Functionality

- Remote debugging
- Run to Click
- Conditional breakpoints
- Data breakpoints
- Watch, Autos, Locals windows

NEW to Visual Studio 2019

- 64-bit out of process debugging
- Just My Code debugging (now for CMake)
- Search in Watch, Autos, Locals windows

6. Test / analyze code

Existing Functionality

- Google Test support
- Boost.Test support
- Test Explorer

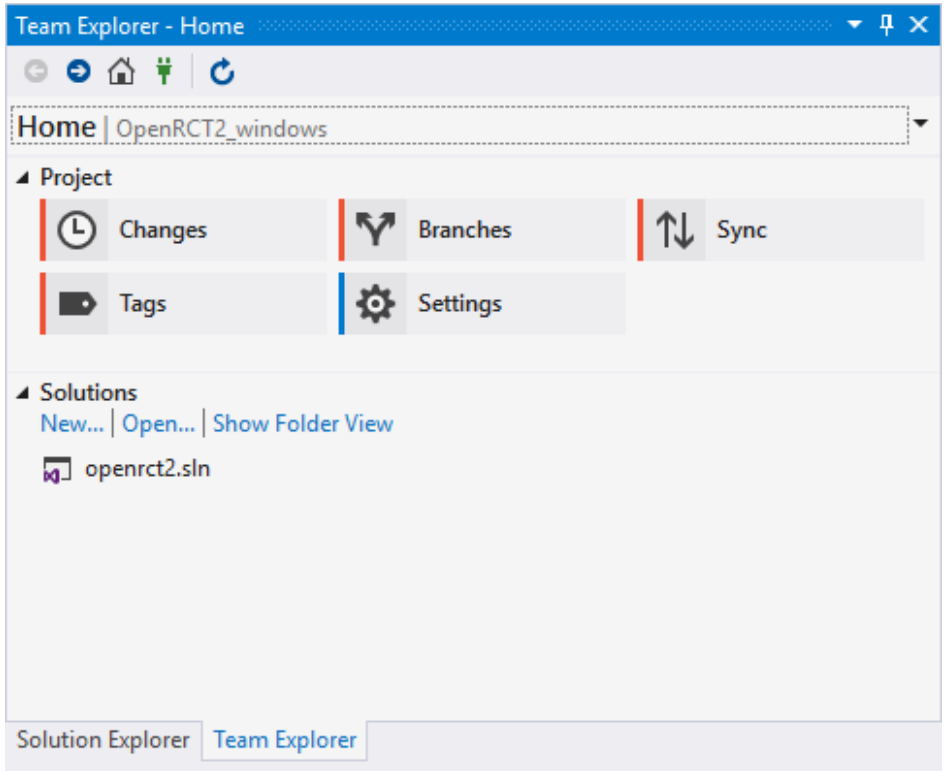
NEW to Visual Studio 2019

- Background Code Analysis
- New C++ core checker rules
- New lifetime and concurrency check rules
- Document Health Indicator
- New Test Explorer window experience
- ASan support for Linux

7. Commit to source control

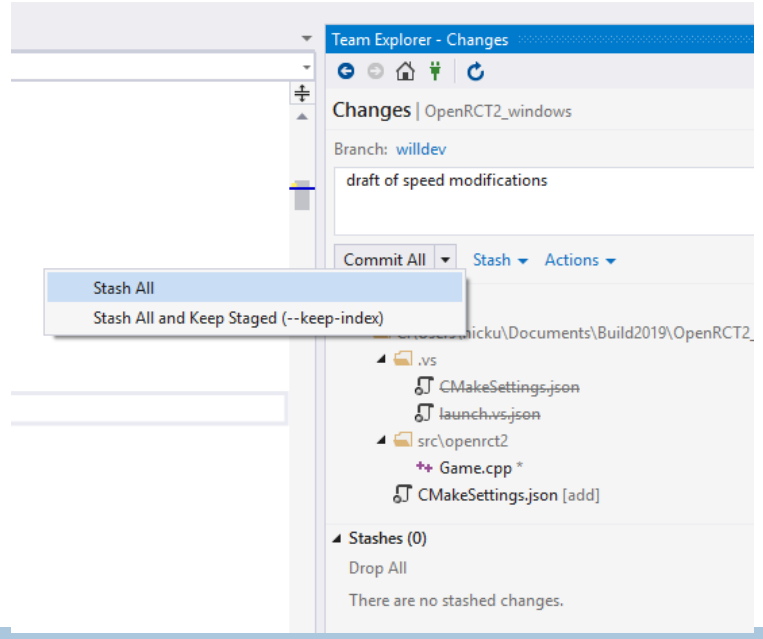
Existing Functionality

- Team Explorer
- Git integration



NEW

- Git stash support
- Pull Requests extension aka.ms/PullRequestsForVS



Summary: Recommended development paths

Development Stage	On Windows Targeting Windows	On Windows Targeting Windows+Linux	On Linux or Mac
Acquiring	Visual Studio + vcpkg	Visual Studio + vcpkg	VS Code + vcpkg
Configuring	CMake Settings Editor or MSBuild project properties	CMake Settings Editor + WSL or Connection Manager	JSON or UI config files for IntelliSense; JSON for build and debug
Building	MSVC	MSVC (Windows) + Clang or gcc (Linux)	CMake or custom + Clang or GCC
Writing	Visual Studio editor	Visual Studio editor	VS Code editor
Debugging	Visual Studio debugger	Visual Studio debugger + gdb for Linux	VS Code with gdb or lldb
Testing/Analyzing	Test Adapters for Google Test/Boost.Test, VS Code Analysis tools	Test Adapters for Google Test/Boost.Test, VS Code Analysis tools, ASan	Third party tools
Committing	Visual Studio source control	Visual Studio source control	VS Code git integration

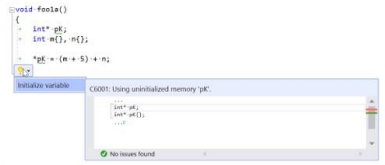
Keep up with the news on the C++ Team Blog

C++ Team Blog

C++ tutorials, C and C++ news, and information about Visual Studio, Visual Studio Code, and Vcpkg from the Microsoft C++ team.

Filter by:

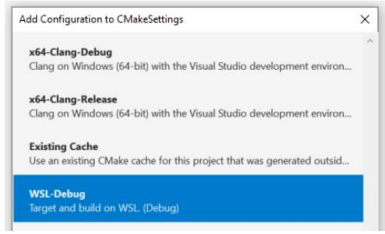
Search C++ Team Blog



New code analysis quick fixes for uninitialized memory (C6001) and use bef...

eli fessler May 7, 2019

In the latest Preview release of Visual Studio 2019 version 16.1, we've added two quick fixes to the Code Analysis experience focused around uninitialized variable checks. These quick fixes are available via the Quick Actions (lightbulb) menu on relevant lines, accessed by hovering over the line or squiggle,



C++ with Visual Studio 2019 and Windows Subsystem for Linux (WSL)

Erika Sweet May 6, 2019

In Visual Studio 2019 version 16.1 Preview 3 we have added native support for using C++ with the Windows Subsystem for Linux (WSL). WSL lets you run a lightweight Linux environment directly on Windows, including most command-line tools, utilities, and applications.

https://devblogs.microsoft.com/cppblog/

Domande?

Thank you to Italian C++ Sponsors!

HOST

PATRON

Community Crumbs

SPONSORS

Grazie!