C++ Productivity in Visual Studio 2017

Augustin Popa, Program Manager – aupopa@microsoft.com

Marian Luparu, Principal Program Manager – mluparu@microsoft.com

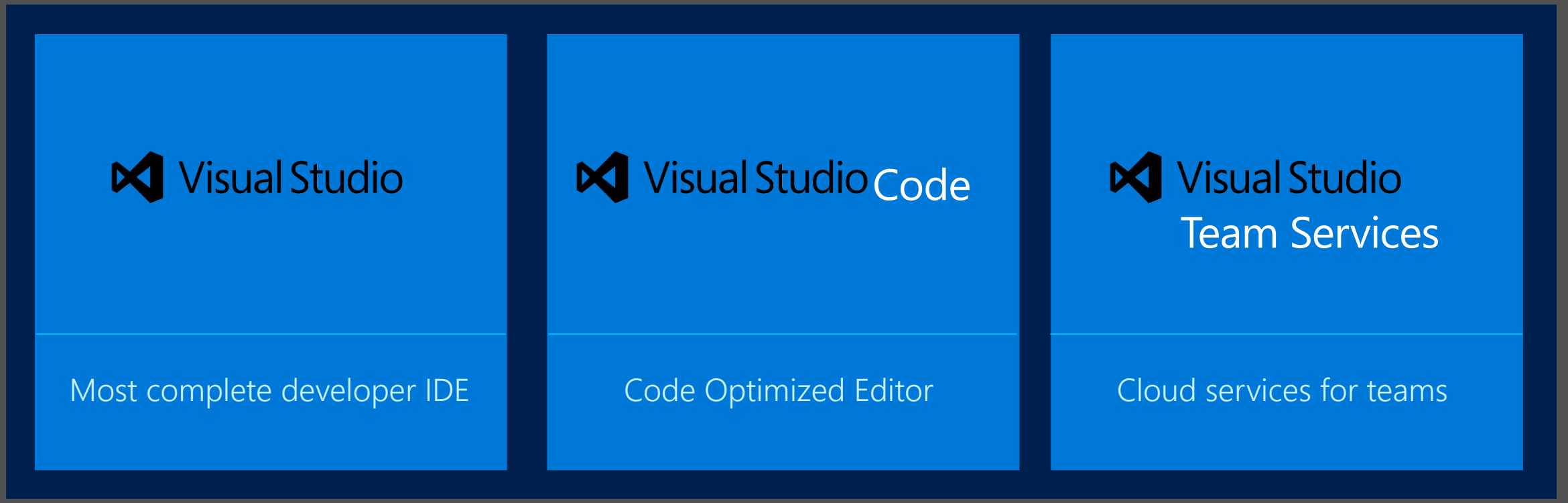
New User Study

We’re running a user study for developers who haven’t used Visual Studio before

1 hr of your time here in Redmond…

Get a $150 Visa gift card for participating!

The Visual Studio family



Best-in-class tools for every developer and every team

Tools and services for any developer, any app

Windows

Linux

iOS

Android

.NET C++ JavaScript PHP

Python Node.js R Cordova Unity

Breaking Down Visual Studio 2017

Acquisition

Performance

Productivity

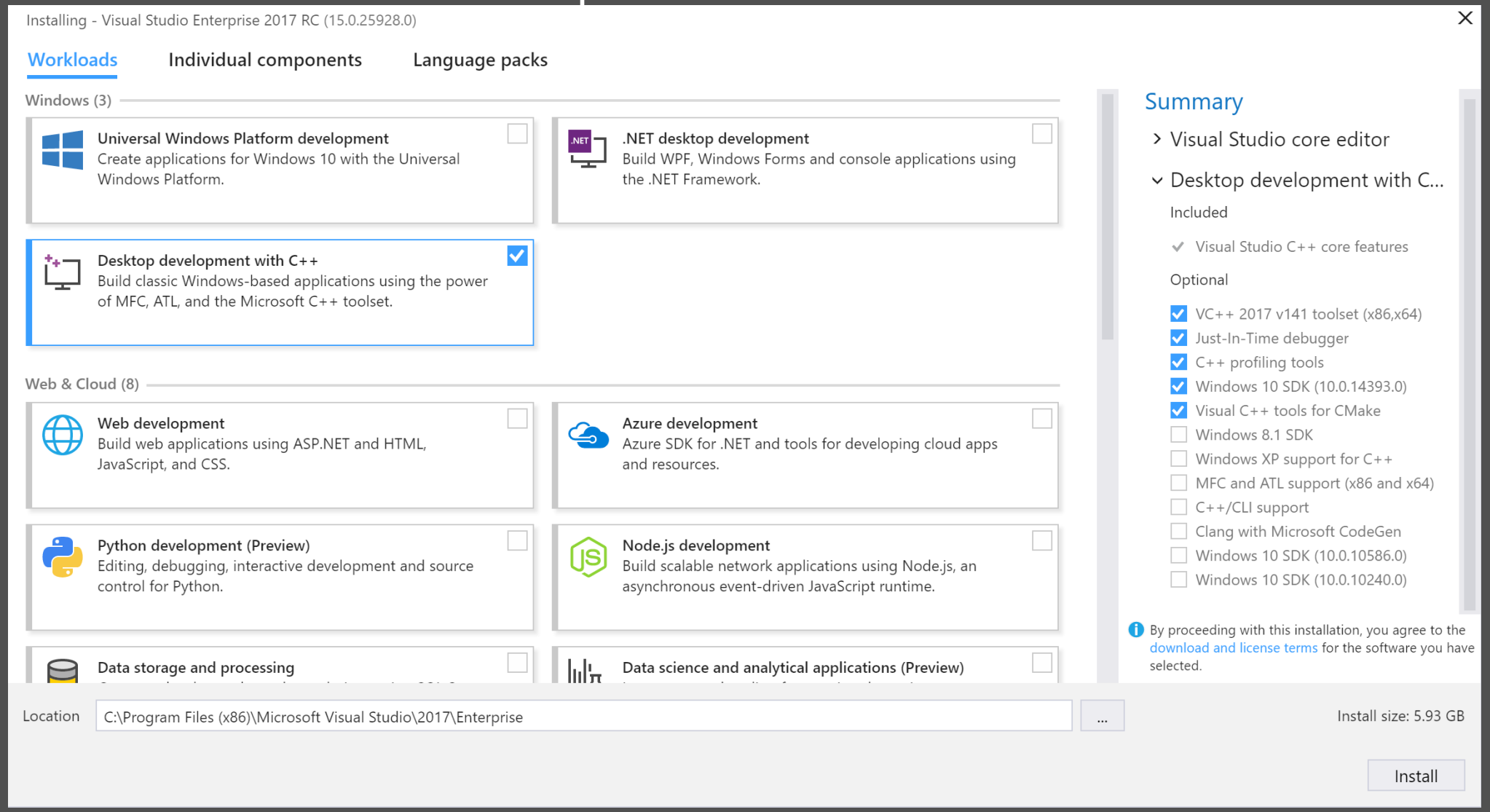
Breaking Down Visual Studio 2017

Acquisition

Performance

Productivity

New Installation Experience



aka.ms/vcinstaller

C++ Workloads

Pick and choose your C++ scenario!
5 workloads relevant to C++

- Desktop Development with C++ (5.76 GB)
- Linux Development with C++ (1.13 GB)
- Game Development with C++ (5.02 GB)
- Mobile Development with C++ (10.97 GB)
- Universal Windows App Development (8.58 GB)

aka.ms/vcinstaller

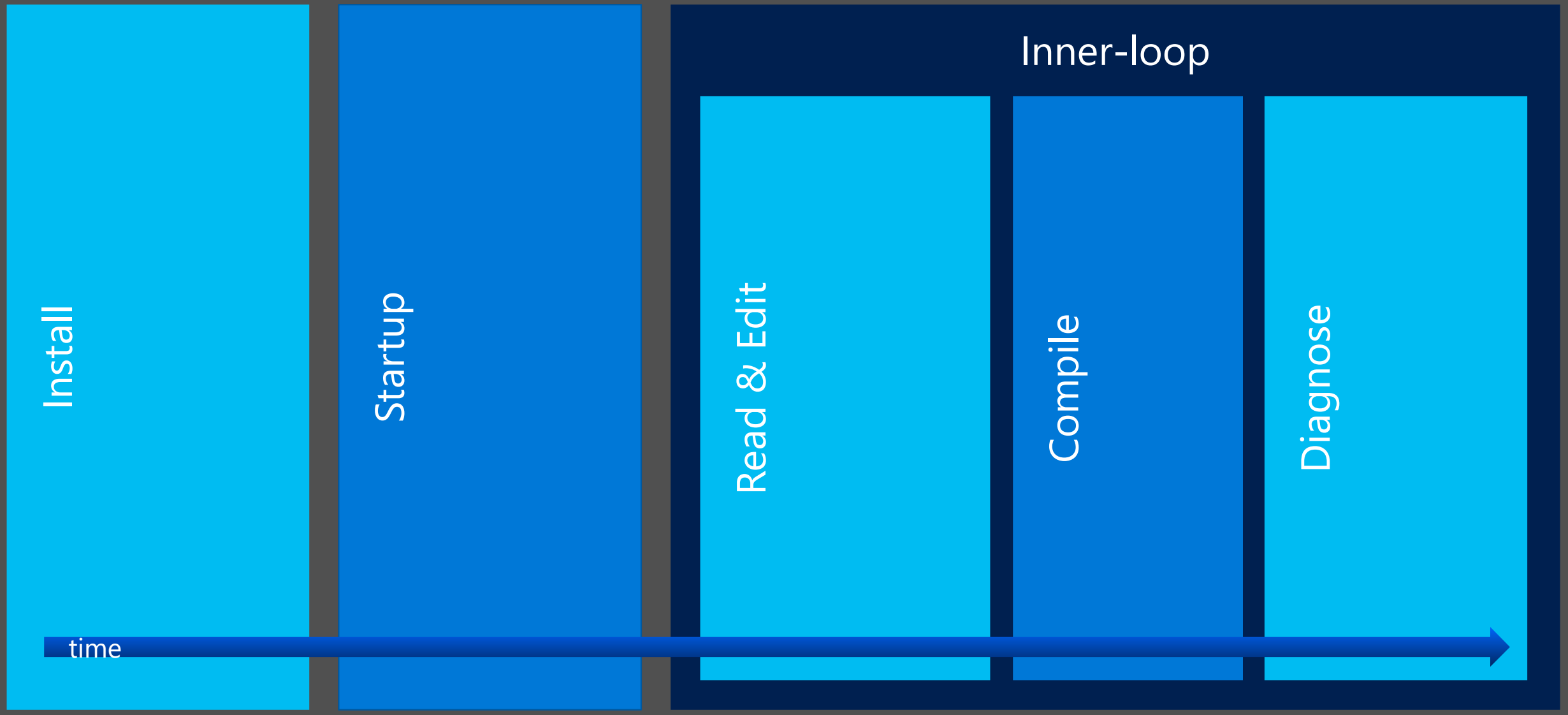
Visual Studio 2017 –Overview

Acquisition

Performance

Productivity

Faster Visual Studio



Faster

Install

Startup

Read & Edit

Faster Startup Times, Less Memory Usage

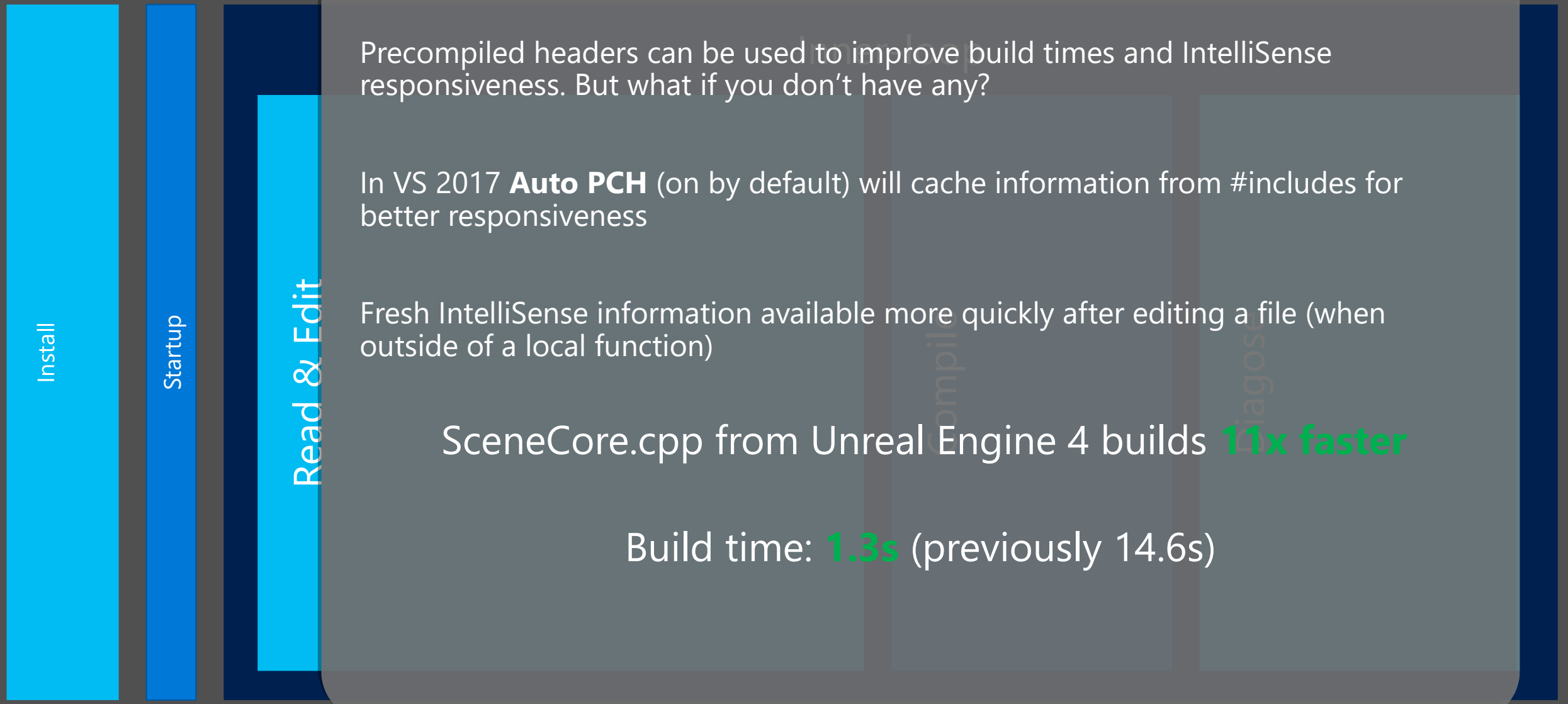
Start the IDE in an instant, load your projects in seconds, use less memory

Dramatic improvements in **solution load time (both 1st time and subsequent)**
Substantial reduction in IDE **Private Working Set** & **Virtual Memory** usage

Chromium - 4600 Solution Items	VS 2015 Update 3	VS 2017	Change
Time - First Solution Open (s)	1,213	182	**-85%**
Time – Subsequent Solution Open (s)	1,211	68	**-94%**
Private Working Set (MB)	2,293	804	**-65%**
Virtual Memory (MB)	3,066	1,302	**-58%**

aka.ms/vcperf

Faster



Auto Pre-Compiled Headers (Auto PCH)

Better IntelliSense Responsiveness

Precompiled headers can be used to improve build times and IntelliSense responsiveness. But what if you don't have any?

In VS 2017 **Auto PCH** (on by default) will cache information from #includes for better responsiveness

Fresh IntelliSense information available more quickly after editing a file (when outside of a local function)

SceneCore.cpp from Unreal Engine 4 builds **11x faster**

Build time: **1.3s** (previously 14.6s)

Faster

Install

Startup

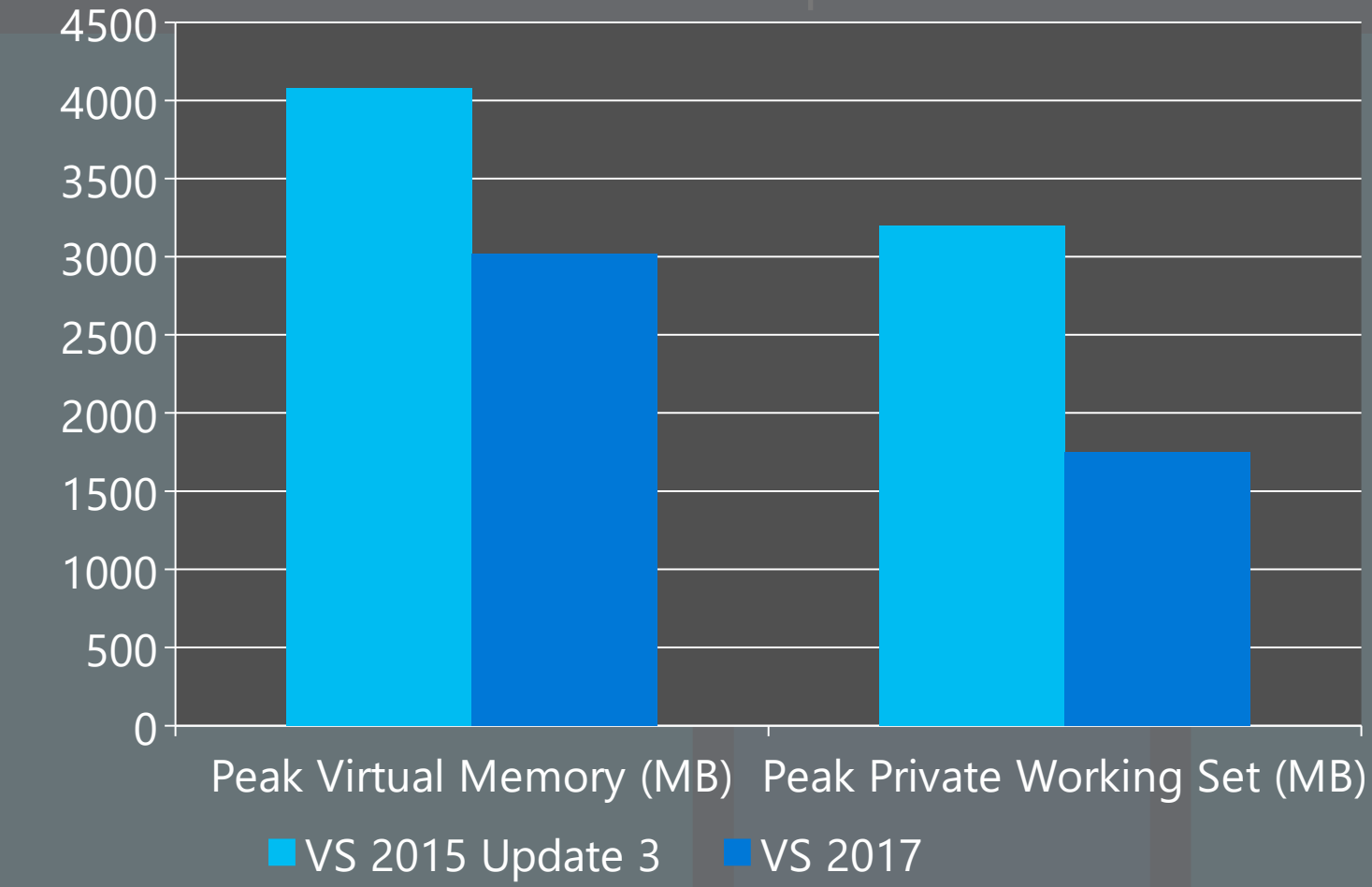
Read & Edit

Inner-loop

Debugger Memory Reductions

Keep your machine from choking while looking for bugs in your code

Data shown below is for Unreal Engine



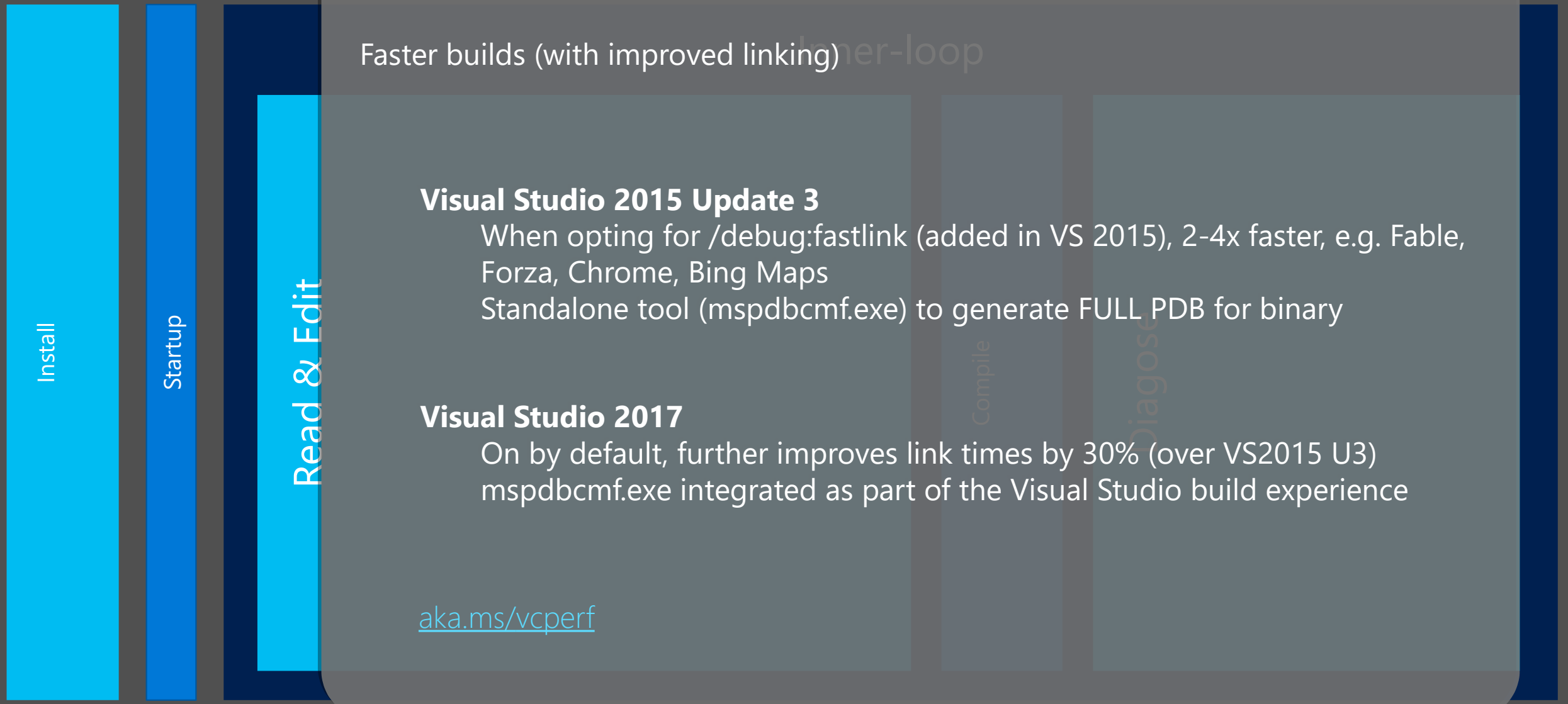
aka.ms/vcperf

Faster

Install

Startup

Read & Edit



Build Throughput Improvements

Get faster to debug your code

Faster builds (with improved linking)

Visual Studio 2015 Update 3

When opting for /debug:fastlink (added in VS 2015), 2-4x faster, e.g. Fable, Forza, Chrome, Bing Maps

Standalone tool (mspdbcmf.exe) to generate FULL PDB for binary

Visual Studio 2017

On by default, further improves link times by 30% (over VS2015 U3)

mspdbcmf.exe integrated as part of the Visual Studio build experience

aka.ms/vcperf

Visual Studio 2017 –Overview

Acquisition

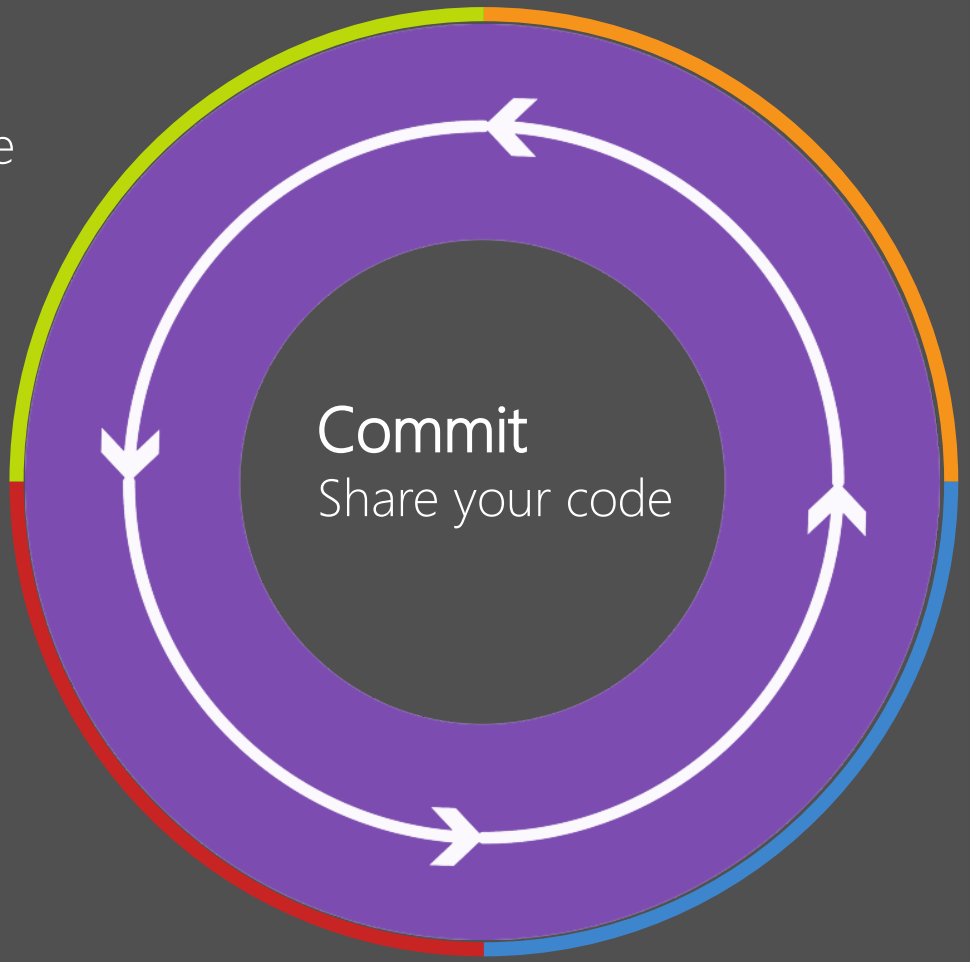
Performance

Productivity

Developer Inner Loop Productivity

Read & Edit
Create and update your code

Diagnose
Fix and optimize your code



Compile
Build your code

Test
Validate your code

Demo

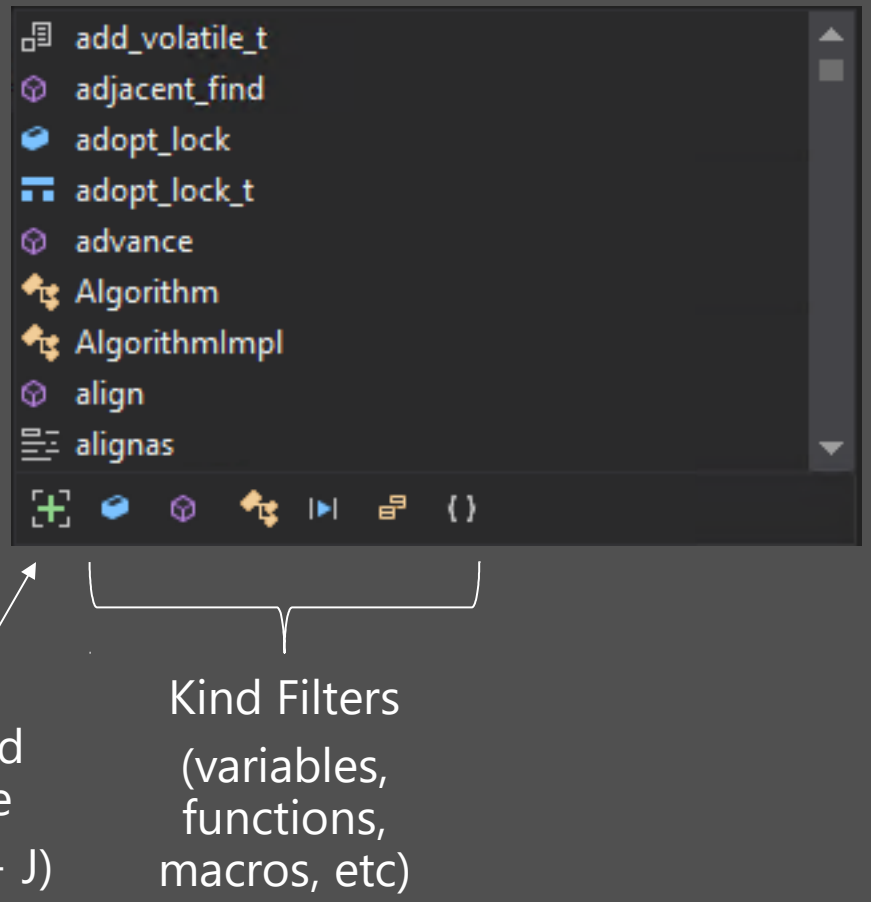
Code Editing & Navigation

IntelliSense

Predictive IntelliSense

- Predictive Scope
- Accessibility Scope
- Text Matching Scope
- No Scope

IntelliSense “Kind” Filters

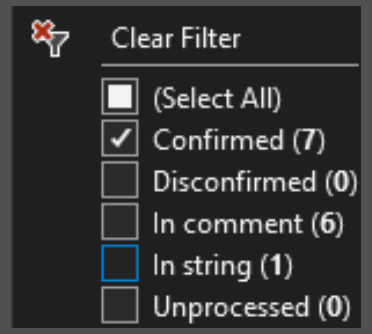


aka.ms/predictiveintellisense

Find All References

Filter Results

(available for most columns)



Group By…

- Project then Definition
- Definition Only
- Definition then Project
- Definition then Path
- Definition, Project then Path

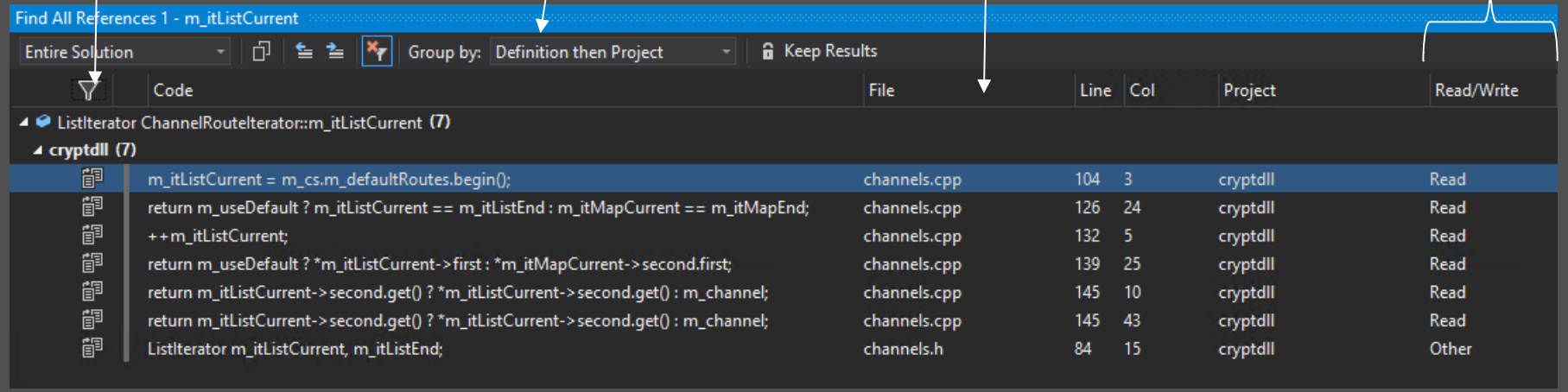
Sort Results

By any column by clicking it

Distinguish

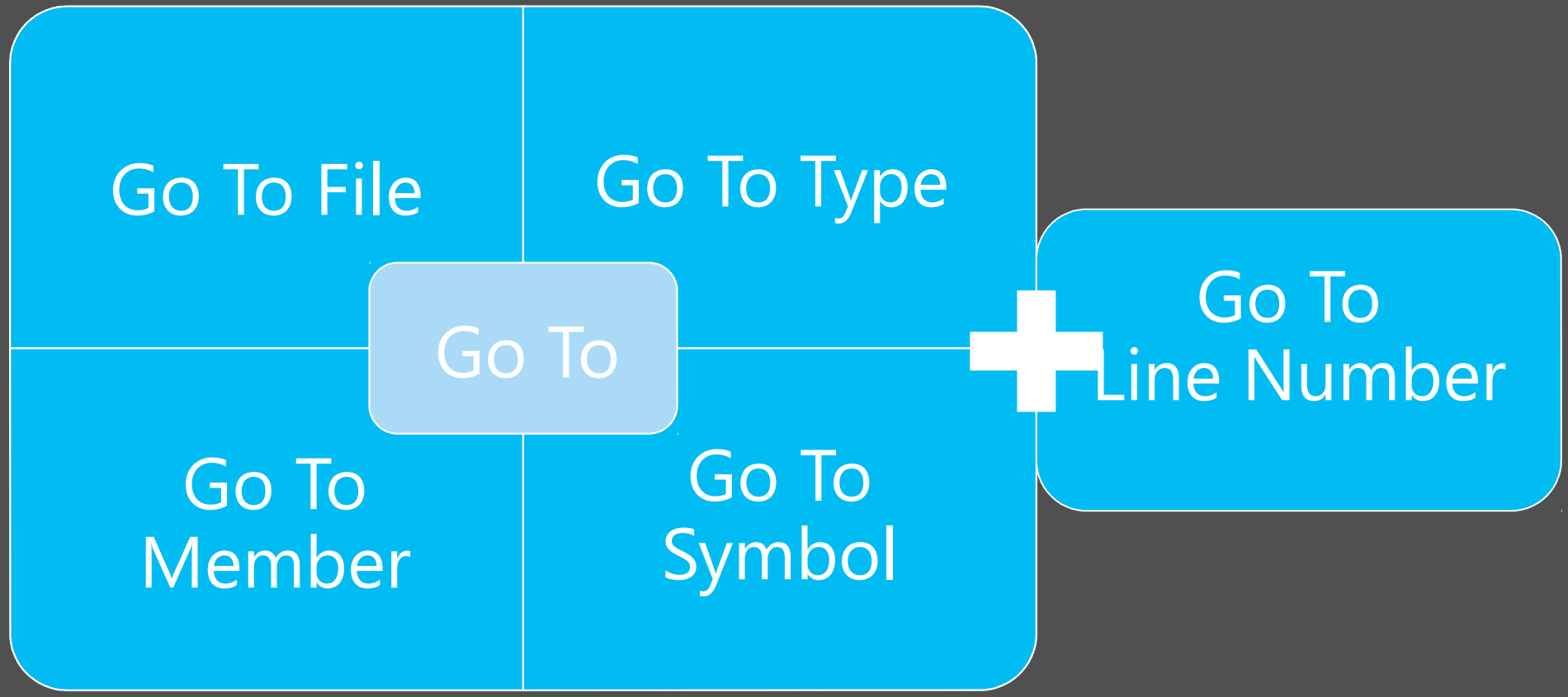
Reads vs. Writes

Incremental Population



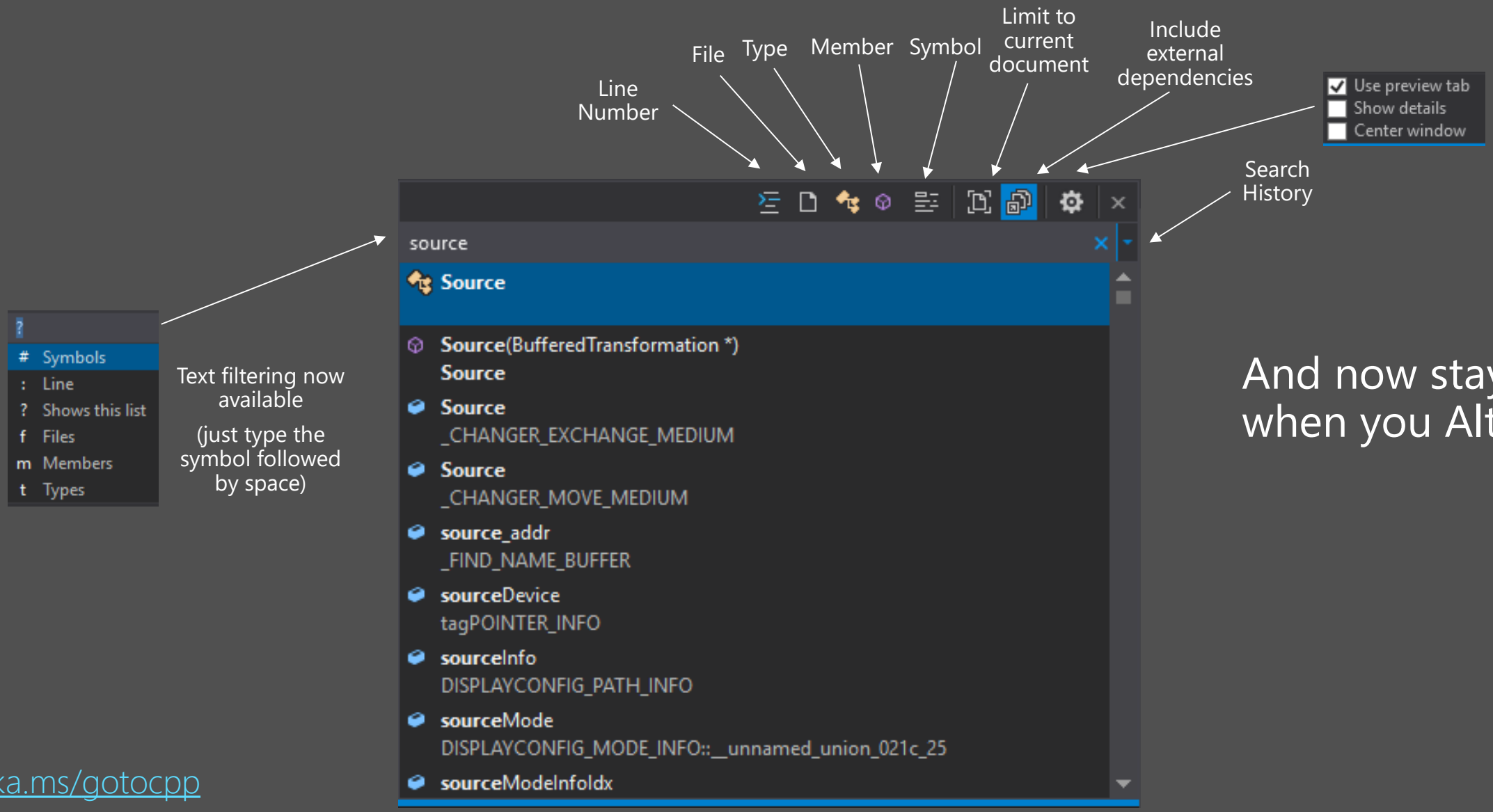
aka.ms/farcpp

Go To (previously Navigate To)



aka.ms/gotocpp

Go To (previously Navigate To)



And now stays open when you Alt + Tab

aka.ms/gotocpp

Support for .editorconfig

Code style enforcement for teams at any level of your project

.editorconfig

```
# See http://EditorConfig.org
root = true

[*]
trim_trailing_whitespace = true

[*.{c,cpp,h,idl,cs}]
indent_style = space
indent_size = 4

```

aka.ms/vs2017productivity

editorconfig.org

CMake Support & Open Folder

“Open Folder” for C++

Ideal for non-MSBuild projects

e.g. any github project using CMake, make or other C++ build systems

Easy to get started

`devenv.exe <directory>`

or “File > Open > Folder...” (Ctrl+Alt+Shift+O)

Read & edit C++ code

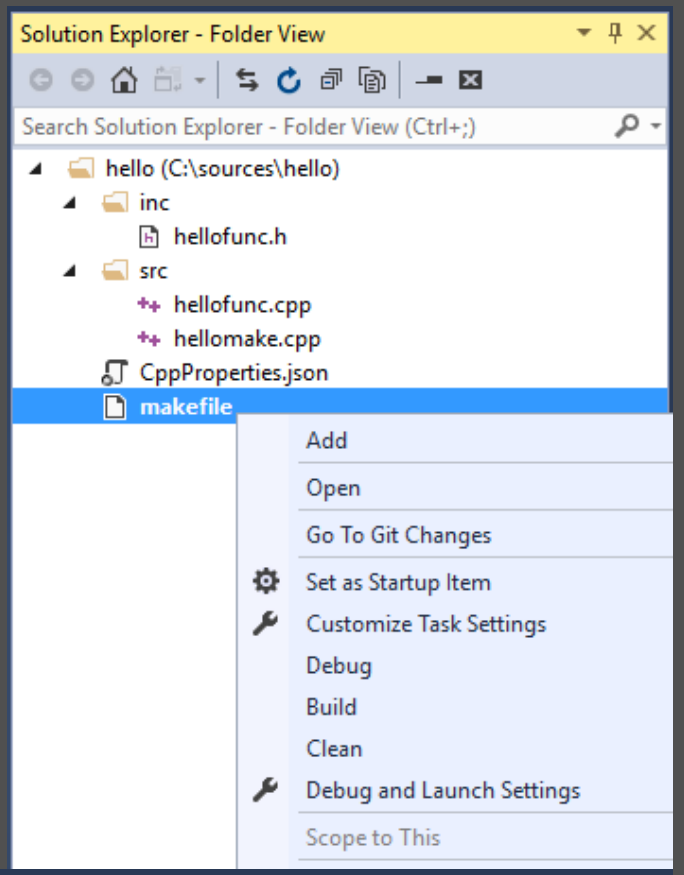
All C++ navigation and IntelliSense features

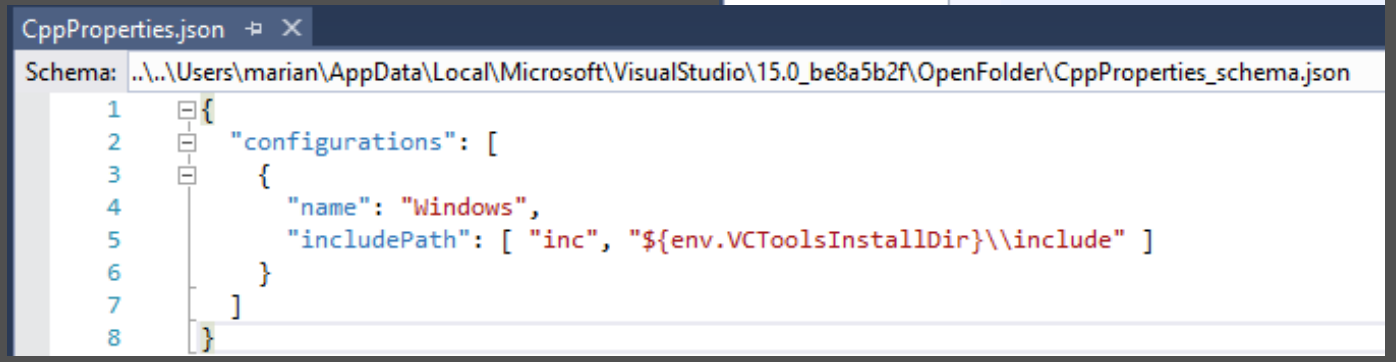
Build & debug C++ binaries

Flexible integration of external build processes

Familiar Visual Studio C++ debug experience

aka.ms/openfolder/cpp





Tools for CMake

Skip the command line

Open CMake folders directly, no VS solution

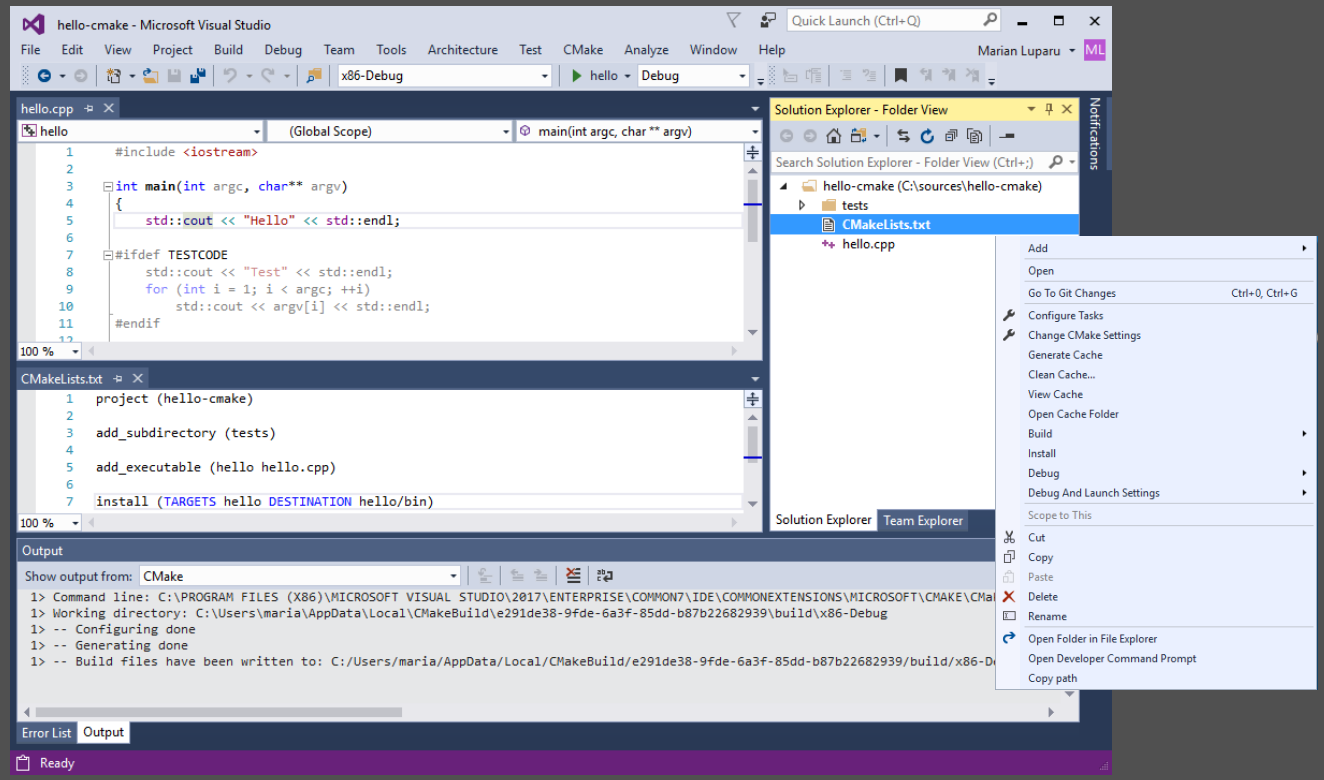
CMake-focused experience

Decluttered folder-based view

CMake-specific configuration

CMake-driven environment

C++ code editing, building and debugging



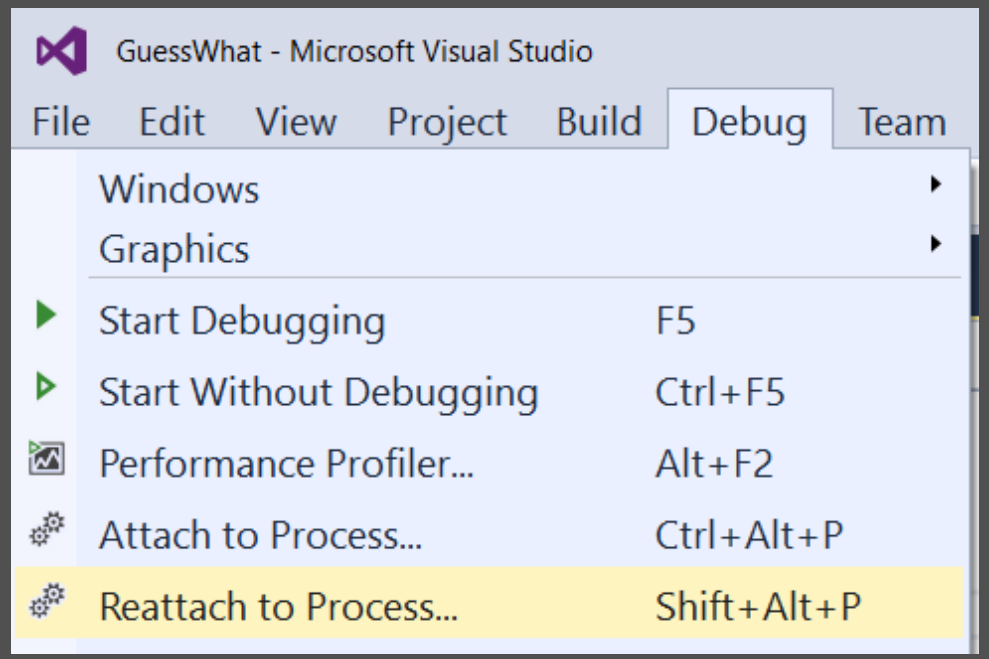
aka.ms/cmake

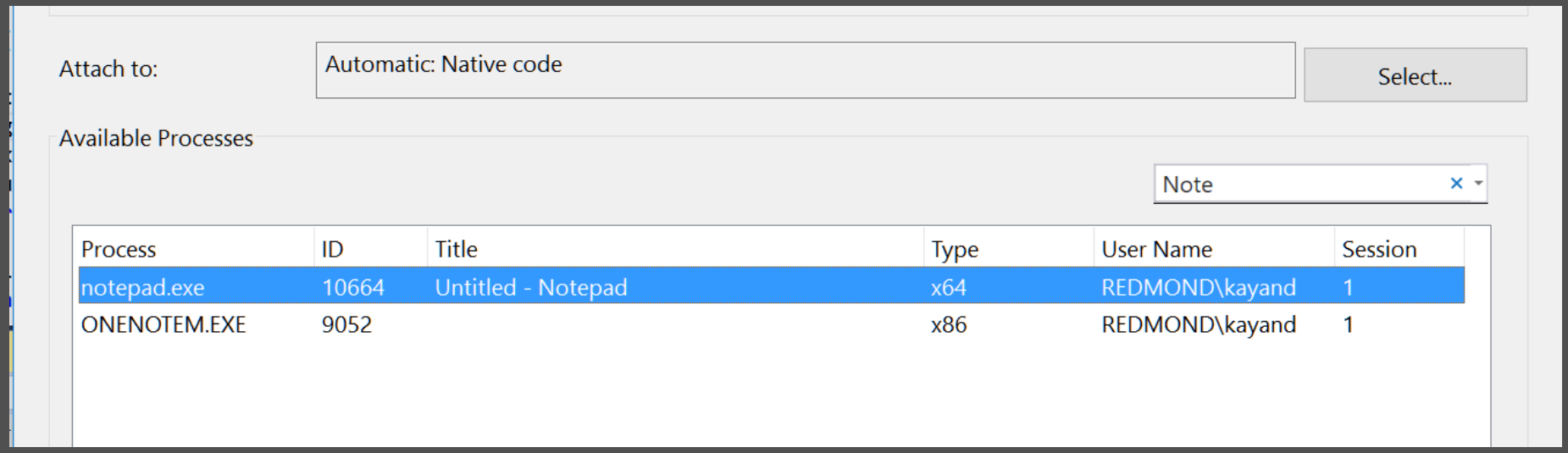
Debugging / Diagnostics

Reattach to Process

Debug Menu | Reattach to Process…

Filter Search Box in Attach to Process



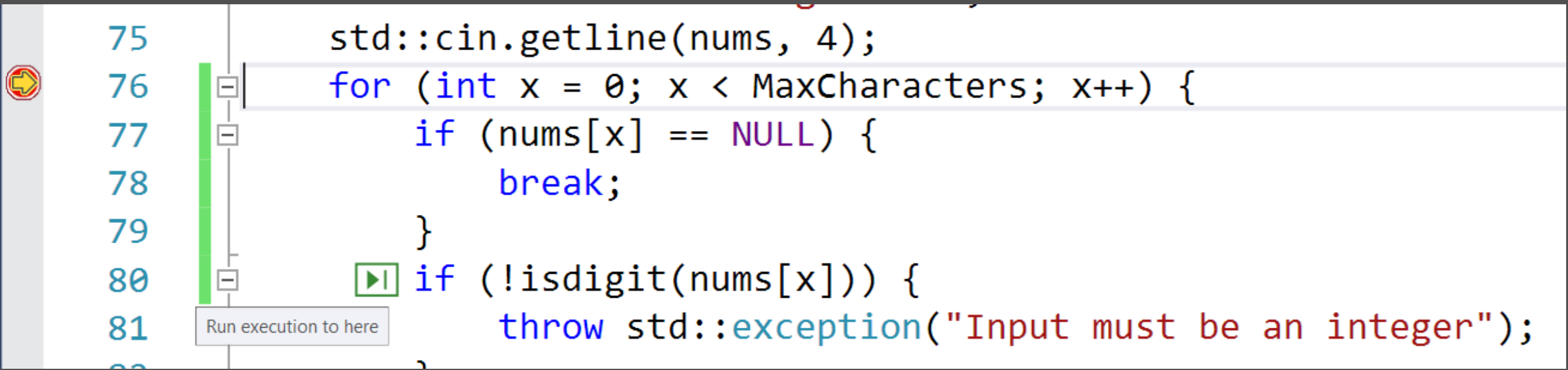


aka.ms/reattach2process

Run to Click

- Run to Cursor at the Push of a Button
  - Hit a Breakpoint while debugging
  - Click the Glyph
  - Execution Runs to that line

aka.ms/run2click



Exception Helper

Non-Modal

Shows what is Null

aka.ms/exceptionhelper

```
61          if (value.is_object())
62          {
63              guess_result->ServerResult->Result = (GuessResult)value.at(L"result").as_integer();
64          }
65          return guess_result;
66      }).get();
67  }
68
69  int GetUserGuess() {
70      int number;
71      const int MaxCharacters = 4;
72      char nums[MaxCharacters];
73
```

Exception Thrown

Exception thrown: read access violation.

guess_result->**ServerResult** was 0xFFFFFFFFFFFFFFFF.

Copy Details

Exception Settings

Q & A and More Info

- Visual C++ Blog: aka.ms/visualcpp
- Why you will love Visual Studio 2017: aka.ms/vs2017love

Contact us directly:

- aupopa@microsoft.com
- mluparu@microsoft.com

Let us know if you're a new user and want to participate in a UX study!
$150 gratuity provided!