Microsoft

DISCOVER
what's new
Visual Studio 2017

Advanced IntelliSense
Code faster with the new IntelliSense prediction algorithm. Better filtering and highlighting helps you find what you're looking for, especially in APIs and code you aren't familiar with.

Fast code navigation
Navigate your code to quickly find types and open files. Identify everywhere a type is referenced in your codebase.

visualstudio.com
https://aka.ms/vsposter

Live code analysis
In-line visualizations identify errors in your code so you can get immediate feedback on code quality as you type. Be more consistent with best coding practices, adhering to team-defined style rules, or applying one of our many refactorings, code actions, and code fixes.

Connected services
Connecting your app to the cloud is now even easier. We've got your back, whether you want to store data in Azure, integrate with Office 365, or use HockeyApp to manage your mobile application.

Synchronize your environment
Have a set of extensions you like to use on all your development workstations? It's easier than ever to roam your settings, preferences, and extensions across all your signed-in machines.

Xamarin Profiler
Xamarin Profiler is a must for profiling mobile apps and optimizing their performance. Optimize memory allocation, identify reference cycles, and understand the bottlenecks in your code. Part of Visual Studio Enterprise.

New Start page
Get to code and start working faster. From creating a new project to cloning from source control to picking up where you left off in a previous session, the start experience has been redesigned with you in mind.

In addition to the core languages, Visual Studio 2017 also has colorization, symbolic search, lightweight IntelliSense, and snippets for:

Swift	Go
PHP	Rust
Objective-C	Ruby
Java	and 20 more

Open folder
Open and work with source code repos without needing to first create a solution or project.

Live Unit Testing
Because every second counts. Get immediate feedback on your code changes by automatically running unit tests in the background. Part of Visual Studio Enterprise.

Live Unit Testing

BLAZE your trail
Select a workload for streamlined options, or mix, match, and make your own rules

Workloads

Workloads you can install and configure
- Windows 10/UWP (Universal Windows Platform): desktop, HoloLens, mobile, Surface Hub, and Xbox
- Visual Studio extensions
- Azure
- Office and SharePoint
- Data storage and processing
- .NET: desktop
- .NET Core: cross-platform

- ASP.NET: web
- C++: desktop, Linux, and mobile
- DirectX, Unreal, and Cocos2d: games
- JavaScript and Apache Cordova: mobile
- Node.js: web
- Unity: games
- Xamarin: mobile

Lower impact
- Installs fewer system components than Visual Studio 2015
- Automatically downloads the latest updates
- Install less by default, install faster

CRUISE through debugging

Debugging and support
Better debugging? Yes, please!

```
Loaded += delegate
{
    _framesPerSecondStopwatch.Start();
    RecomputeAndRedraw();
}
```

Run To Click
Make temporary breakpoints a thing of the past. Now, you can run to a line of code while the Visual Studio 2017 debugger is paused.

Exception Helper
Exception debugging getting you worked up? The new Visual Studio 2017 exception helper gives you the information you need to fix it and keep coding.

JUMP AHEAD with new extensions

Extensibility
Make Visual Studio your own
Visual Studio 2017 is powerful right out of the box, but it doesn't cover everything. Make Visual Studio the perfect IDE for your needs by adding custom tool windows, additional commands, new functionality in the code editor, or support for a new language. Visit https://aka.ms/extendvs for ideas, samples, and walkthroughs to get you started.

.NET Core CLI reference

C:\source> dotnet [host-options] [command] [arguments] [common-options]

.NET COMMANDS:
new
Initialize .NET projects.
restore
Restore dependencies specified in the .NET project.
build
Builds a .NET project.
publish
Publishes a .NET project for deployment (including the runtime).
run
Compiles and immediately executes a .NET project.

test
Runs unit tests using the test runner specified in the project.
pack
Creates a NuGet package.
migrate
Migrates a project.json based project to a msbuild based project.
clean
Clean build output(s).
sln
Modify solution (SLN) files.

PROJECT MODIFICATION COMMANDS:
add
Add items to the project
remove
Remove items from the project
list
List items in the project

DESIGNED FOR DEVELOPERS

DevOps

Redgate ReadyRoll Core Edition
Stuck in a SQL bottleneck? Redgate Data Tools extend DevOps to databases. Use advanced IntelliSense-style code completion to develop migration scripts, manage changes using source control, and safely automate deployments. Part of Visual Studio Enterprise.

PowerShell Tools for Visual Studio
A rich interactive environment geared toward DevOps, including a command explorer and language service.

Continuous Delivery Tools for Visual Studio
Your delivery pipeline needs to work, period. Visual Studio's CI/CD tools get your ASP.NET 4 and ASP.NET Core project updates to your customers. Fast.

Docker
Restarting your container after every code change takes time. Docker integration enables you to develop, run, and validate your ASP.NET Core apps, without the restarts. Now you can use that time for something else. Like coding.

.NET Standard

Get in the merge lane: all your .NET APIs, together

Multi-platform development is really coming together. The Microsoft .NET Standard Library unifies APIs for desktop apps, mobile apps, games, and cloud services.

.NET FRAMEWORK
WPF · WINDOWS FORMS
ASP.NET

.NET CORE
UWP · ASP.NET CORE · WINDOWS

XAMARIN
iOS · ANDROID

APP MODELS

.NET STANDARD LIBRARY
One library to rule them all

COMMON INFRASTRUCTURE
COMPILERS · LANGUAGES · RUNTIME COMPONENTS

.NET STANDARD 2.0
XML SERIALIZATION · NETWORKING · IO · THREADING · CORE

XLinq, XML Document, XPath, XSD, XSL
BinaryFormatter, Data Contract, XML
Sockets, Http, Mail, WebSockets
Files, Compression, MMF
Threads, Thread Pool, Tasks
Primitives, Collections, Reflection, Interop, Linq

Navigate your CODE

Visual Studio Mobile Center
Mission control for your mobile apps. Make data-based decisions to give your users the faster release cycles and higher-quality apps they want and deserve. https://aka.ms/vsmc

Visual Studio Team Services
Share code, track work, and ship software for any language. It's all designed for continuous integration to make your existing Agile and GitHub workflows better. https://aka.ms/vsts

Default keyboard shortcuts

SEARCH AND NAVIGATION
Quick Launch	Ctrl+Q
Go to All	Ctrl+T or Ctrl+Comma
Go to Type / File / Member / Symbol	Ctrl+[L/T/F/M/S]
Navigate Backward / Forward	Ctrl+Hyphen / Ctrl+Shift+Hyphen
Quick Find / Quick Replace	Ctrl+F / Ctrl+H
Find / Replace in Files	Ctrl+Shift+F / Ctrl+Shift+H
Go to Definition	F12
Peek Definition	Alt+F12
Go to Implementation	Ctrl+F12
Find All References	Shift+F12
Go to Next Result in List (Find Results, Error List, etc.)	F8 / Shift+F8 (forward / back)
Go to Next Error	Ctrl+Shift+F12

EDITING AND REFACTORING
Quick Actions / Refactoring Suggestions	Alt+Enter or Ctrl+Period
IntelliSense Code Completion	Ctrl+Space
Parameter Info / Signature Help	Ctrl+Shift+Space
Method Info	Ctrl+K, Ctrl+I
Comment	Ctrl+K, Ctrl+C
Uncomment	Ctrl+K, Ctrl+U
Delete Line (without copying it)	Ctrl+Shift+L
Paste from Clipboard Ring (Paste from buffer of previously copied items)	Ctrl+Shift+V
Move Code Up / Down	Alt+Up arrow / Alt+Down arrow
Format Document	Ctrl+K, Ctrl+D
Format Selection	Ctrl+K, Ctrl+F
Surround with (../if/try/foreach)	Ctrl+K, Ctrl+S
Remove and Sort Usings	Ctrl+R, Ctrl+G
Rename	Ctrl+R, Ctrl+R
Encapsulate Field	Ctrl+R, Ctrl+E
Remove Parameters	Ctrl+R, Ctrl+V
Reorder Parameters	Ctrl+R, Ctrl+O
Extract Method	Ctrl+R, Ctrl+M
Extract Interface	Ctrl+R, Ctrl+I

DEBUGGING AND TESTING
Debug	F5
Run (without debugging)	Ctrl+F5
Stop Debugging	Shift+F5
Pause Debugging	Ctrl+Alt+Break
Toggle Breakpoint	F9
Step Over	F10
Step Into	F11
Step Out	Shift+F11
Run All Tests	Ctrl+R, A
Debug All Tests	Ctrl+R, Ctrl+A
Run Tests in Context	Ctrl+R, T
Debug Tests in Context	Ctrl+R, Ctrl+T
Execute Code Snippet in Interactive Window	Ctrl+E, E

WINDOW MANAGEMENT
Open Tool Windows:	
Properties	F4
Solution Explorer	Ctrl+Alt+L
Output Window	Ctrl+Alt+O
Error List	Ctrl+\, E
Team Explorer	Ctrl+\, Ctrl+M
Breakpoints	Ctrl+Alt+B
Immediate Window	Ctrl+Alt+I
Cycle Through Tool Windows	Alt+F6 / Shift+Alt+F6 (forward / back)
Access any Open Tool Window and Documents	Ctrl+Tab (hold Ctrl), Up / Down / Left / Right arrows
Close Current Tool Window	Shift+Esc
Go to Document to the Left / Right	Ctrl+Alt+Page up / Ctrl+Alt+Page down
Go to Most Recently /	Ctrl+Tab /
Least Recently Accessed Document	Ctrl+Shift+Tab
Select Active File in Solution Explorer	Ctrl+[, S
Keep Preview Window Open	Ctrl+Alt+Home
Full Screen (max window size / reduced menus)	Shift+Alt+Enter

Want to change your keyboard mappings quickly? Ctrl+Q, then type **keyboard**

Give us your feedback
Got a suggestion, bug report, or compliment? This icon is your direct line to the Visual Studio team. We hope you use it a lot.

Language improvements

C# 7

Tuples
Lightweight, unnamed types with multiple public fields.
```
private static (int Max, int Min) Range(IEnumerable<int> numbers)
{
    int min = int.MaxValue;
    int max = int.MinValue;
    foreach (var n in numbers)
    {
        min = (n < min) ? n : min;
        max = (n > max) ? n : max;
    }
    return (max, min);
}

var numbers = new int[] { 1, 2, 3, 100, 180, 2250 };
var range = Range(numbers);
Console.WriteLine($"The numbers supplied range from {range.Min} to {range.Max}.");
```

More expression-bodied members
Constructors, finalizers, and get and set accessors can now be authored with expressions.
```
// Expression-bodied constructor
public ExpressionResourceExample(string label) => this.Label = label;

// Expression-bodied finalizer
~ExpressionMembersExample() => Console.Error.WriteLine("Finalized!");

private string label;

// Expression-bodied get / set accessors
public string Label
{
    get => label;
    set => this.label = value ?? "Default label";
}
```

throw expressions
Increased support for throw expressions in code constructs
```
private ConfigResource loadedConfig =
LoadConfigResourceOrDefault() ??
    throw new InvalidOperationException("Could not load config");
```

Pattern matching
Branching logic based on arbitrary types and member values.
```
switch (shape)
{
    case Triangle t:
        area = t.Base * 2 * t.Height;
        break;
    case Rectangle r:
        area = r.Width * r.Height;
        break;
    case Circle c:
        area = Math.PI * c.Radius * c.Radius;
        break;
    default:
        throw new ArgumentException(
            message: "shape is not a recognized shape",
            paramName: nameof(shape));
}
```

out variables
Simplified methods by declaring out variables inline.
```
if (int.TryParse(input, out int result))
    Console.WriteLine(result);
else
    Console.WriteLine("Could not parse input");
```

Numeric literal syntax
New tokens for improved readability of numeric constants.
```
public const int Sixteen = 0b0001_0000;
public const int ThirtyTwo = 0b0010_0000;
public const int SixtyFour = 0b0100_0000;
public const int OneHundredTwentyEight = 0b1000_0000;
```

MORE C# GOODIES
Local functions: Limit scope and visibility by nesting functions inside other functions.
Generalized async return types: Methods declared with async can now return more than just Task and Task<T>.

F#

- .NET Standard and .NET Core support
- struct representation of types
- New Result type
- Improvements to the F# compiler and FSharp.Core
- New editor based on Roslyn Workspaces
- Community contributions and bug fixes

TypeScript

Downlevel async functions – Use async/await and target ES5/ES3 without any other tools.
Non-nullable types – New strict null checking mode makes it easier to detect erroneous use of null and undefined.
Simplified declaration file acquisition – Get declaration files using npm.
Object type – Match any types except primitive types.

Visual Basic

Tuples
Need to return multiple values from a method without using out parameters? Visual Studio 2017 has you covered. Use tuples to easily pass around groups of values.
Binary literals
No more mentally mapping decimal and hexadecimal numbers to their binary bit patterns. Native support for binary in Visual Basic and cosmetic enhancements will make reading large literals much easier in Visual Studio 2017.

Git commands

CREATE REPOSITORIES
Start a new repository or obtain one from an existing URL
> git init [project-name] Creates a new local repository with the specified name
> git clone [url] Downloads a project and its entire version history

MAKE CHANGES
Review edits and craft a commit transaction
> git status Lists all new or modified files to be committed
> git diff Shows file differences not yet staged
> git ls-files --others --ignored --exclude-standard Lists all ignored files in this project
> git add [file] Snapshots the file in preparation for versioning
> git diff --staged Shows file differences between staging and the last file version

SAVE FRAGMENTS
Shelve and restore incomplete changes
> git stash Temporarily stores all modified tracked files
> git stash pop Restores the most recently stashed files
> git stash list Lists all stashed changesets
> git stash drop Discards the most recently stashed changeset

REVIEW HISTORY
Browse and inspect the evolution of project files
> git log Lists version history for the current branch

> git checkout [branch-name] Switches to the specified branch and updates working directory
> git merge [branch-name] Combines the specified branch's history into the current branch
> git branch -d [branch-name] Deletes the specified branch

REFACTOR FILE NAMES
Relocate and remove versioned files
> git rm [file] Deletes the file from the working directory and stages the deletion
> git mv [file-original] [file-renamed] Changes the file name and prepare it for commit

GROUP CHANGES
Name a series of commits and combine completed efforts
> git branch Lists all local branches in the current repository
> git branch [branch-name] Creates a new branch

> git diff [first-branch]...[second-branch] Shows content differences between two branches
> git log Outputs metadata and content changes of the specified commit

REDO COMMITS
Erase mistakes and craft replacement history
> git reset [commit] Undoes all commits after [commit], preserving changes locally
> git reset --hard [commit] Discards all history and changes back to the specified commit

SYNCHRONIZE CHANGES
Register a remote URL and exchange repository history
> git merge [remote]/[branch] Combines the remote branch into the current local branch
> git merge [remote] [branch] Uploads all local branch commits to GitHub
> git pull Downloads bookmark history and incorporates changes

Azure 2.0 CLI

INSTALL AZURE 2.0 CLI
From a Windows machine with Python installed:
C:\git> pip install azure-cli
Basic command syntax
C:\git> az command [options]

DEPLOY A WEBSITE
Login to Azure and set the deployment user for Azure web services:
C:\git> az login
C:\git> az appservice web deployment user set --user-name <username> --password <password>
Create a new resource group:
C:\git> az appservice list-locations
C:\git> az group create --name "location" --name my-group

Create a new web app with a unique name <app>:
C:\git> az appservice web create --name <app> --resource-group my-group --plan appservice-plan
Git clone your website:
C:\git> git clone https://github.com/user/mysite.git
C:\git> cd mysite
Configure local Git deployment for your ASP.NET Core web app:
C:\git> az appservice web source-control config-local-git --name <app> --resource-group my-group
Add the URL in the JSON as a Git remote for your local repository (here called 'azure' for simplicity):
C:\git> git remote add azure https://<user>@<app>.scm.azurewebsites.net/<app>.git
Deploy your website to your Azure app like you would push any code with Git:
C:\git\mysite> git push azure master

CREATE A VIRTUAL MACHINE
Create a resource group
C:\git> az appservice list-locations
C:\git> az group create --location "location" --name my-group
Show VM images available for creation:
C:\git> az vm image list --resource-group my-group --name myVM --image UbuntuLTS --generate-ssh-keys
Create a new virtual machine, create SSH keys if not present:
C:\git> az vm create --resource-group my-group --name myVM --image UbuntuLTS --generate-ssh-keys
Open port 80 to allow web traffic to host:
C:\git> az vm open-port --port 80 --resource-group my-group --name myVM

C++

Pain-free upgrade, improved performance, and support for any C++ codebase
Get more details at https://aka.ms/cpp2017

- 'Open Folder' of any C++ code (including CMake) without converting to MSBuild
- Write cross-platform C++ code for Windows, Linux, Android, iOS, and Windows

- Create C++ games using Unreal, Cocos, or DirectX
- Upgrade and maintain library compatibility with third-party library dependencies built with the Visual Studio 2015 toolset

- Upgrade your third-party open-source libraries automatically with the new C++ library manager
- Use the Visual Studio 2015 toolchain from the Visual Studio 2017 IDE

- Write better code than ever by following the latest C++ standards
- Gain insights into your code with improved compiler diagnostics and code analysis

- Take advantage of the latest security advancements in C++ runtime
- Get better runtime performance of your code with the latest C++ compiler