

POWER UP VISUAL STUDIO TO OPTIMIZE YOUR DEVELOPMENT

by Nadia Comanici

PEAK IT - an AgileHub Event
08 december 2018

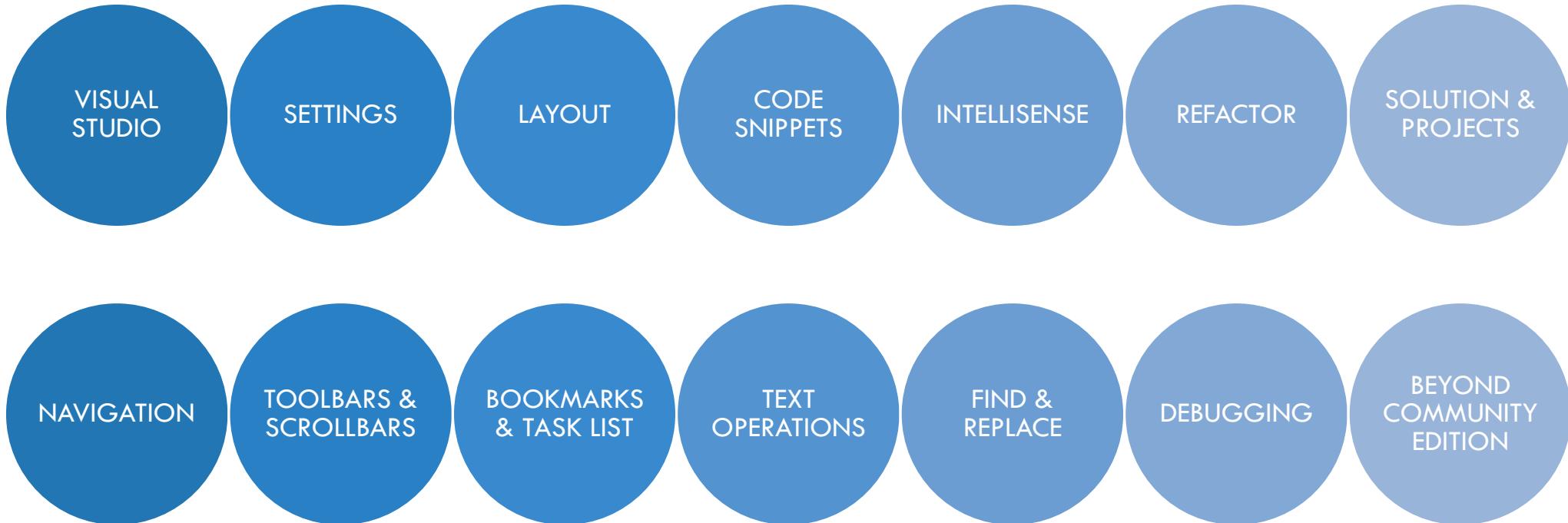


WORKSHOP RECORDING

You can view the recording from this workshop at:
<https://www.youtube.com/watch?v=mdM7p-MAc8o>



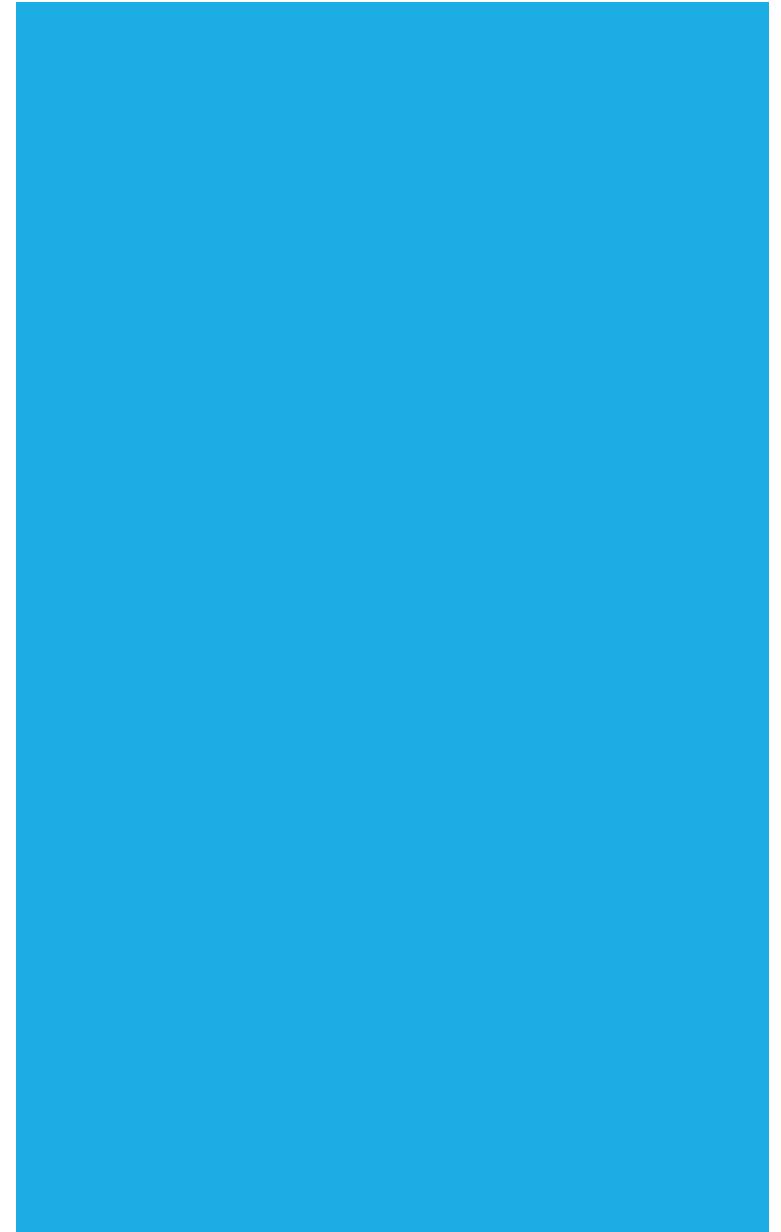
TOPICS



INSPIRATION FOR THE COURSE

- 10 years of Personal Experience
- <https://app.pluralsight.com/library/courses/visual-studio-2015-playbook>
- <https://app.pluralsight.com/library/courses/visual-studio-2015-essentials-power-user>

VISUAL STUDIO



VISUAL STUDIO - EDITIONS

Editia	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise
Description	<ul style="list-style-type: none">- Free- Replaces VS Express- Ideal version for learning Visual Studio	<ul style="list-style-type: none">- For smaller teams	<ul style="list-style-type: none">- For larger teams and larger projects > 250 PCs or > 1 Million \$
In addition to the previous version	<ul style="list-style-type: none">- Code Metrics- Static Code Analysis- Unit Testing- Peek Definition- Refactoring- One-Click Web Deployment	<ul style="list-style-type: none">- CodeLens- Integrare cu TFS	<ul style="list-style-type: none">- Architectural Layer Diagrams- IntelliTrace- Code Map Debugger Integration- .NET Memory Dump Analysis- Testing tools- Live Unit Testing- Code Coverage- Xamarin Profiler & Inspector

More: <https://visualstudio.microsoft.com/vs/compare>

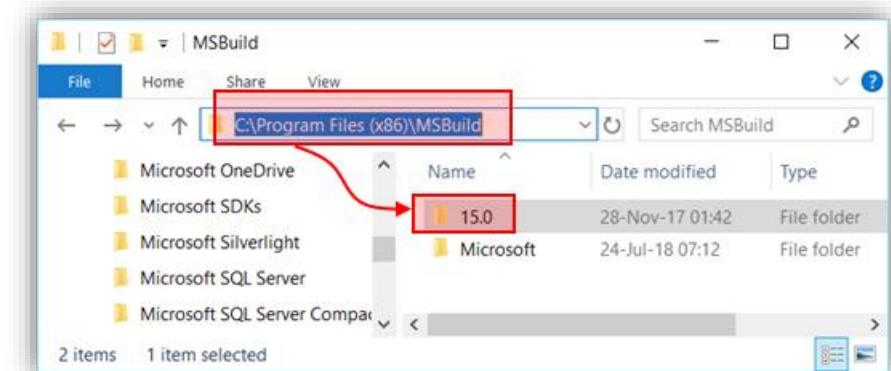
VISUAL STUDIO - VERSIONS

For Visual Studio, the internal code and the release year do not match

Visual Studio [year]	Internal Code (Version)
VS 2005	v 8
VS 2008	v 9
VS 2010	v 10
VS 2012	v 11
VS 2013	v 12
VS 2015	v 14
VS 2017	v 15

- Check version:

1. Visual Studio install location



2. The .sln file contains the minimum version to open the solution

A screenshot of the Visual Studio IDE showing an open solution file 'AutomatedTellerMachine.sln'. The code editor displays the following content:

```
1 Microsoft Visual Studio Solution File, Format Version 12.00
2 # Visual Studio 15
3 VisualStudioVersion = 15.0.26430.14
4 MinimumVisualStudioVersion = 10.0.40219.1
5
6 Project ("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") =
7 "AutomatedTellerMachine",
8 "AutomatedTellerMachine\AutomatedTellerMachine.csproj",
```

A red box highlights the version information in the code.

NAMING CONVENTION

Settings	Properties
<ul style="list-style-type: none">- For Visual Studio- Applies to all the solutions and all projects that you open, such as:<ul style="list-style-type: none">- Keyboard shortcuts- Panel layout- Toolbars- Text size and color- Are automatically saved when closing VS	<ul style="list-style-type: none">- For the current project/solution, such as<ul style="list-style-type: none">- Namespace- Executable name- Output folder

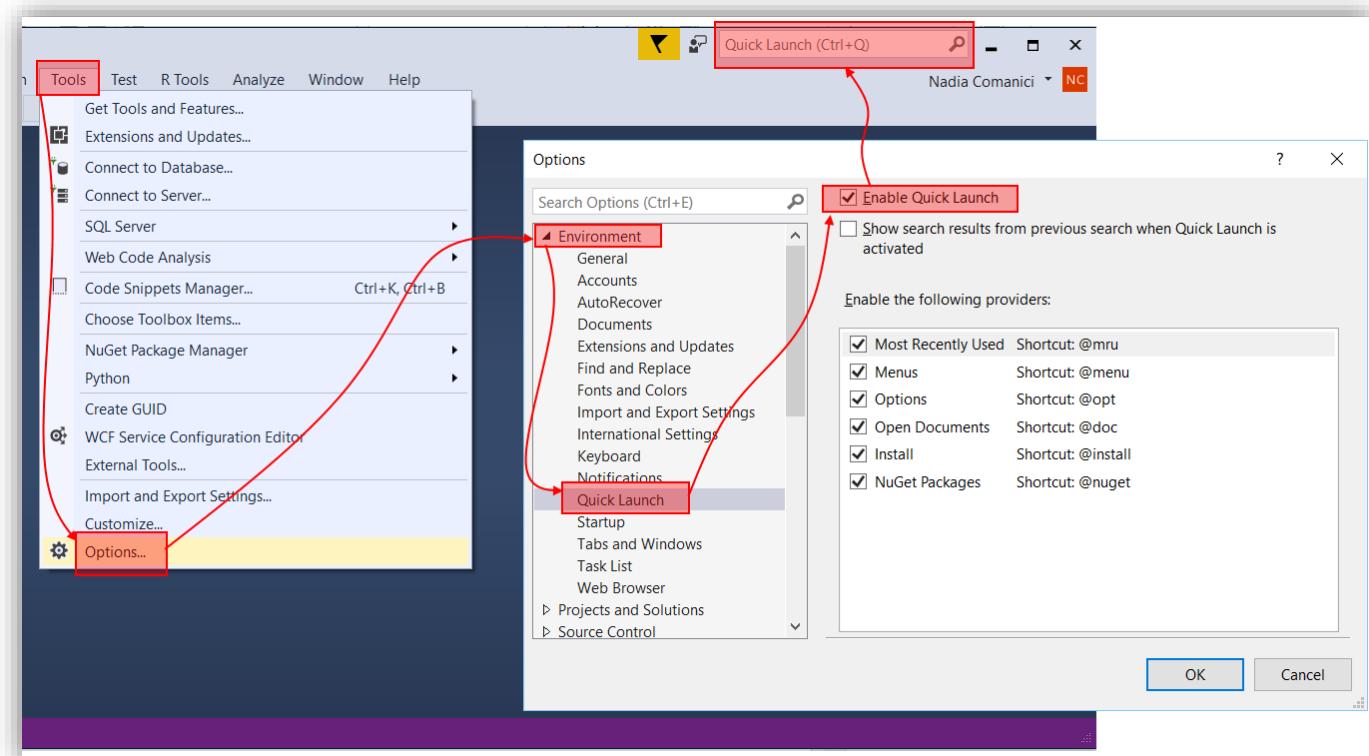
QUICK LAUNCH

Enable/disable:

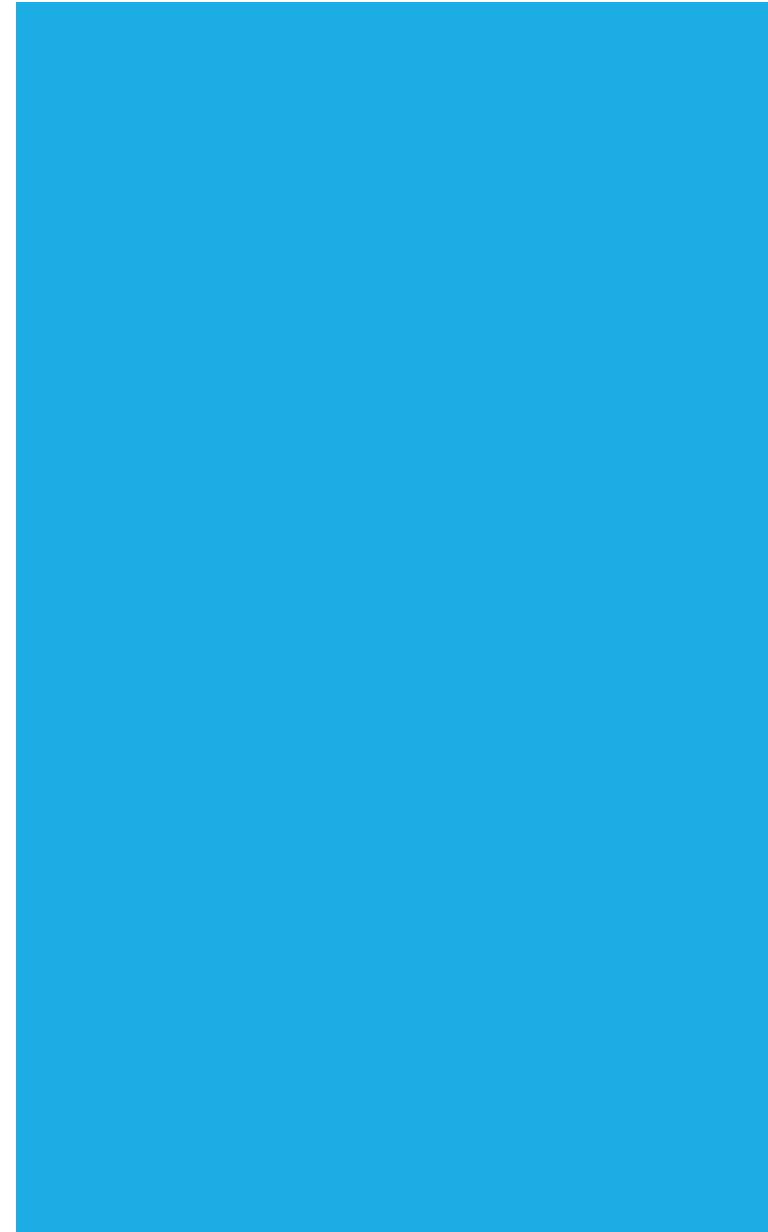
- Tools → Options → Environment → Quick Launch

Usage:

- Ctrl + Q
- Search box in top upper corner



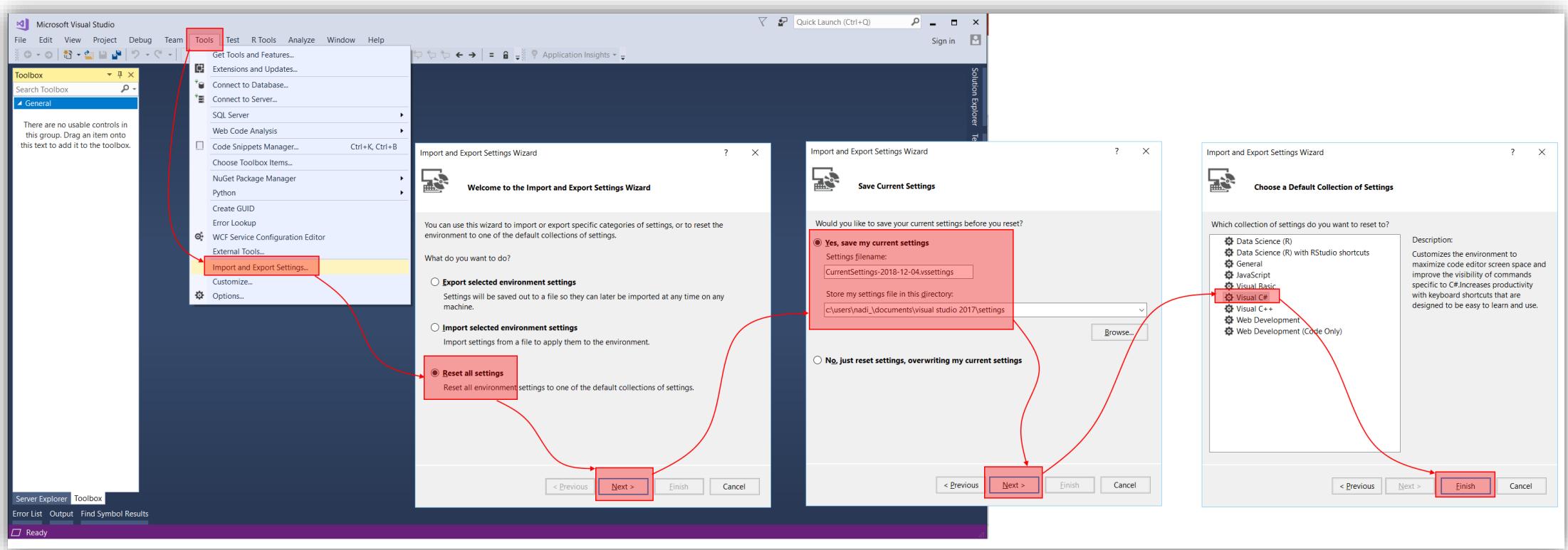
SETTINGS



ENVIRONMENT SETTINGS - RESET

Tools → Export/Import Settings → Reset All Settings → Visual C#

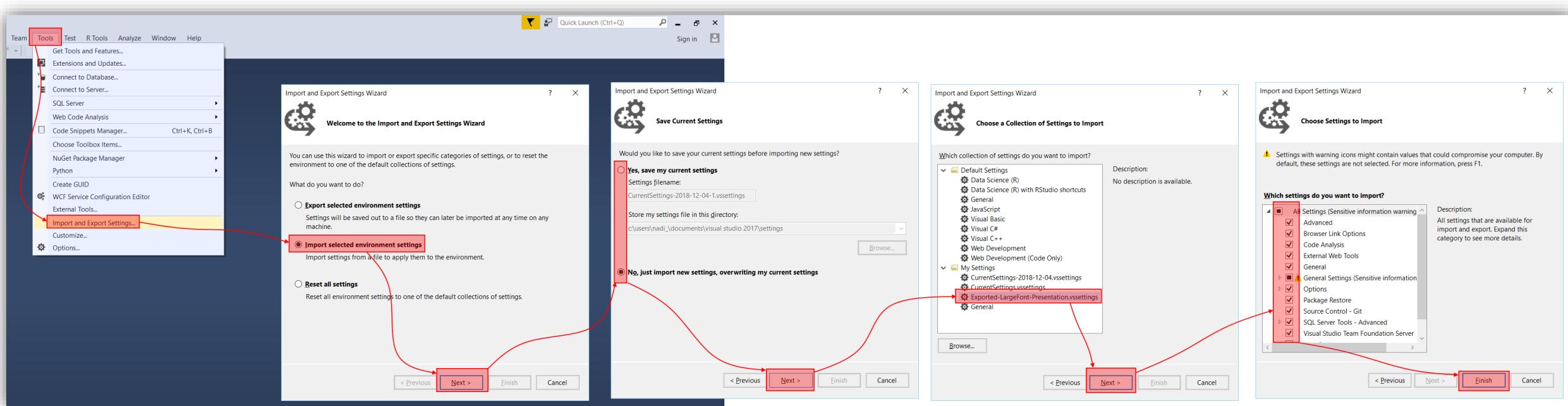
- Prompts you to save current settings



ENVIRONMENT SETTINGS - IMPORT

Tools → Export/Import Settings → Import Settings

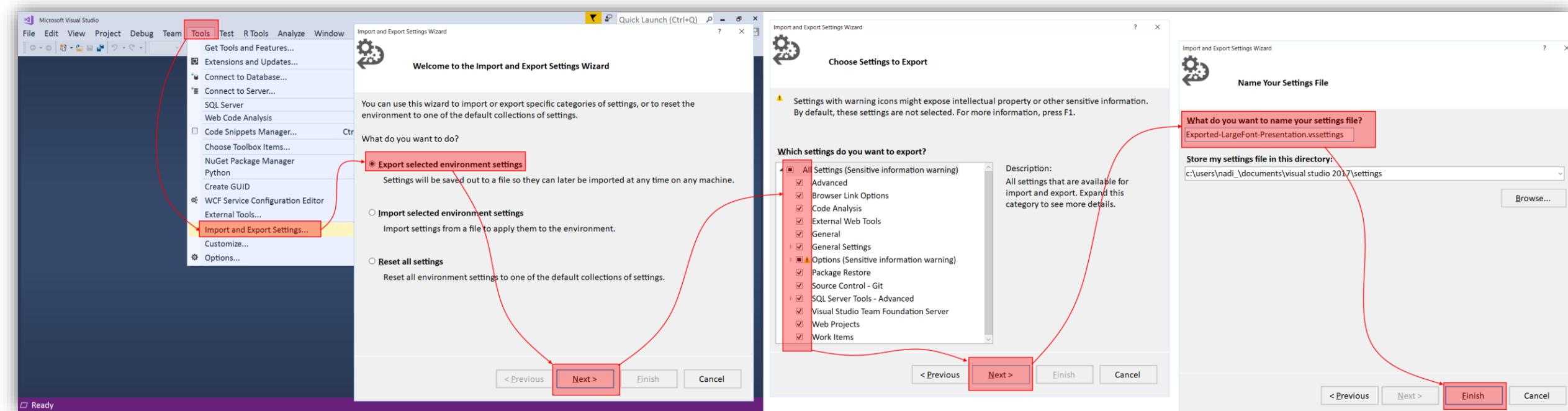
- Prompts you to save current settings
- You can choose which settings to import and which not to save



ENVIRONMENT SETTINGS - EXPORT

Tools → Export/Import Settings → Export Settings

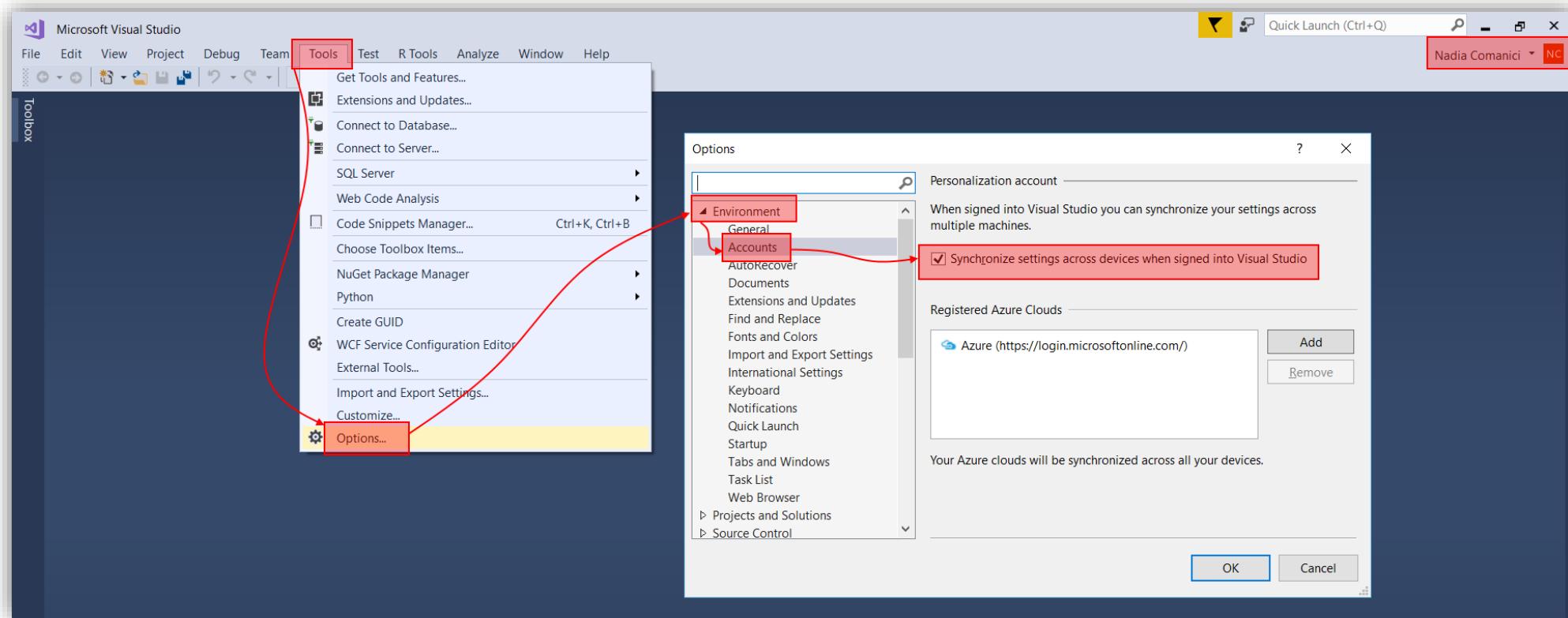
- You can choose which settings to save and which not to save



SYNC SETTINGS

If logged in with Microsoft account can sync across multiple PCs

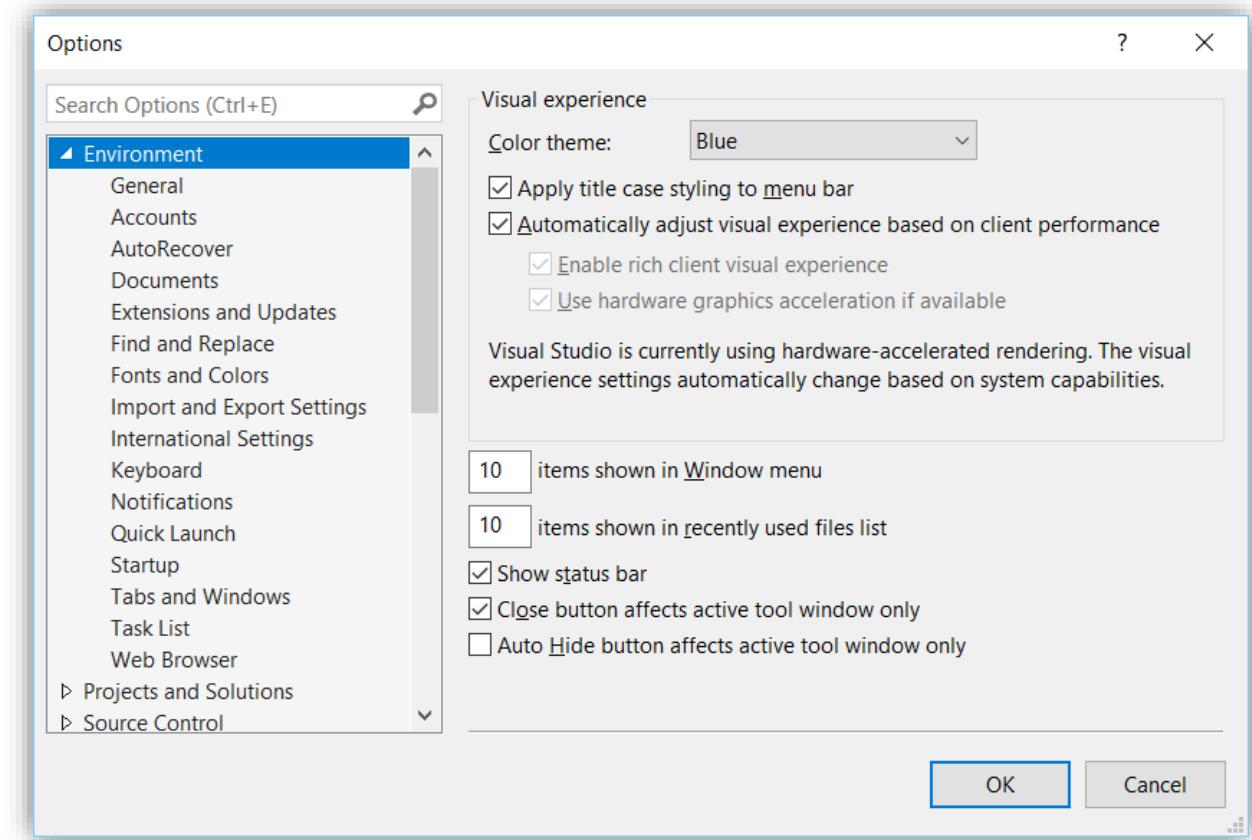
- Tools → Options → Environment → Account → Sync settings accros devices when logged in



ENVIRONMENT SETTINGS

Tools → Options → Environment

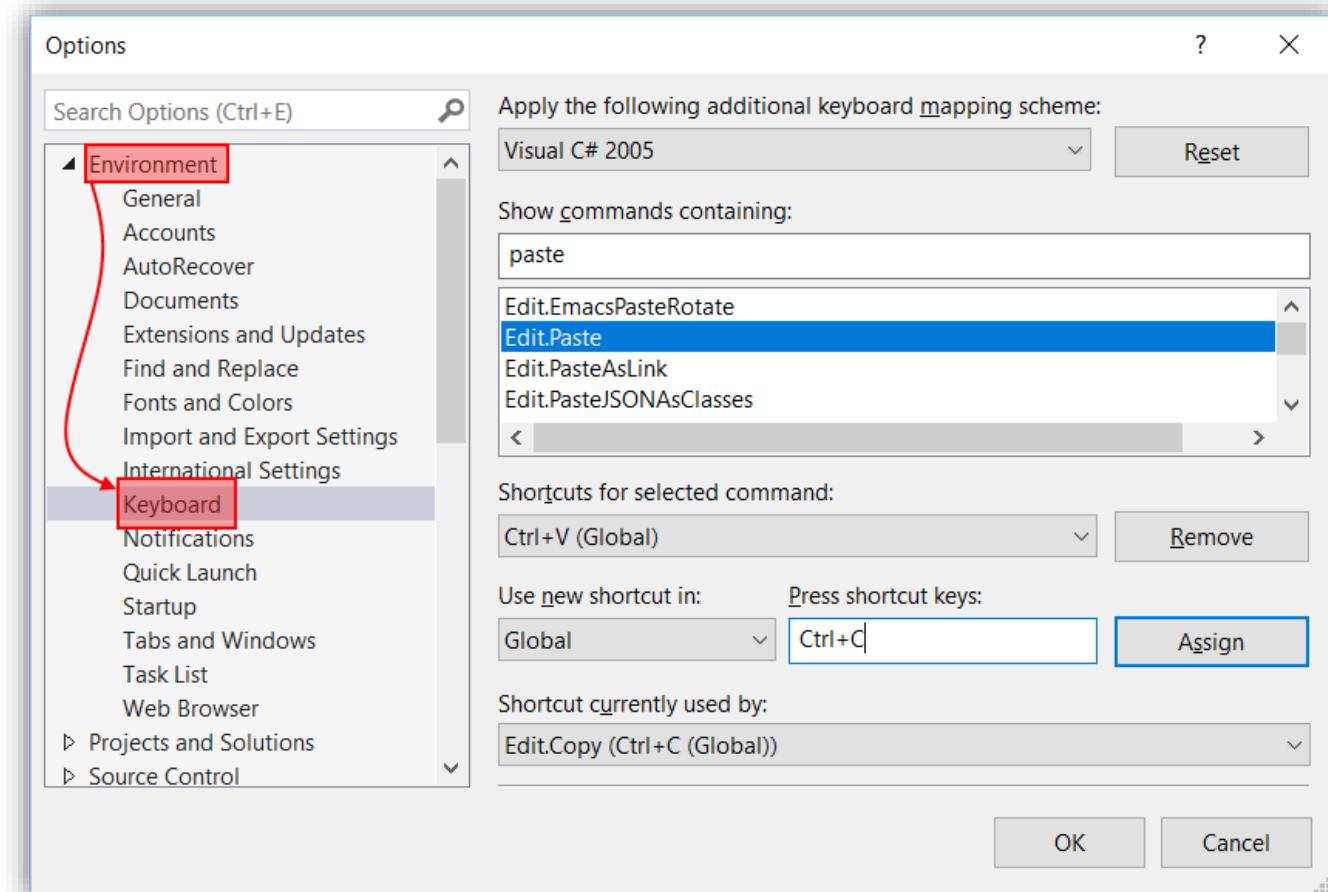
- General
 - Themes
 - Items Shown in Window menu / recently used
 - Show Status Bar
- AutoRecover
 - Save AutoRecovery every X minutes
- Fonts and Colors
 - Set different font size or color
- Documents
 - Detect Changes
 - Edit read-only files
 - Misc folder
- Quick Launch
 - Enable / disable
- Startup
 - Choose what to do at startup
 - Download at ... Min
 - Customize startup page
- Tabs and windows
 - Pinned tabs



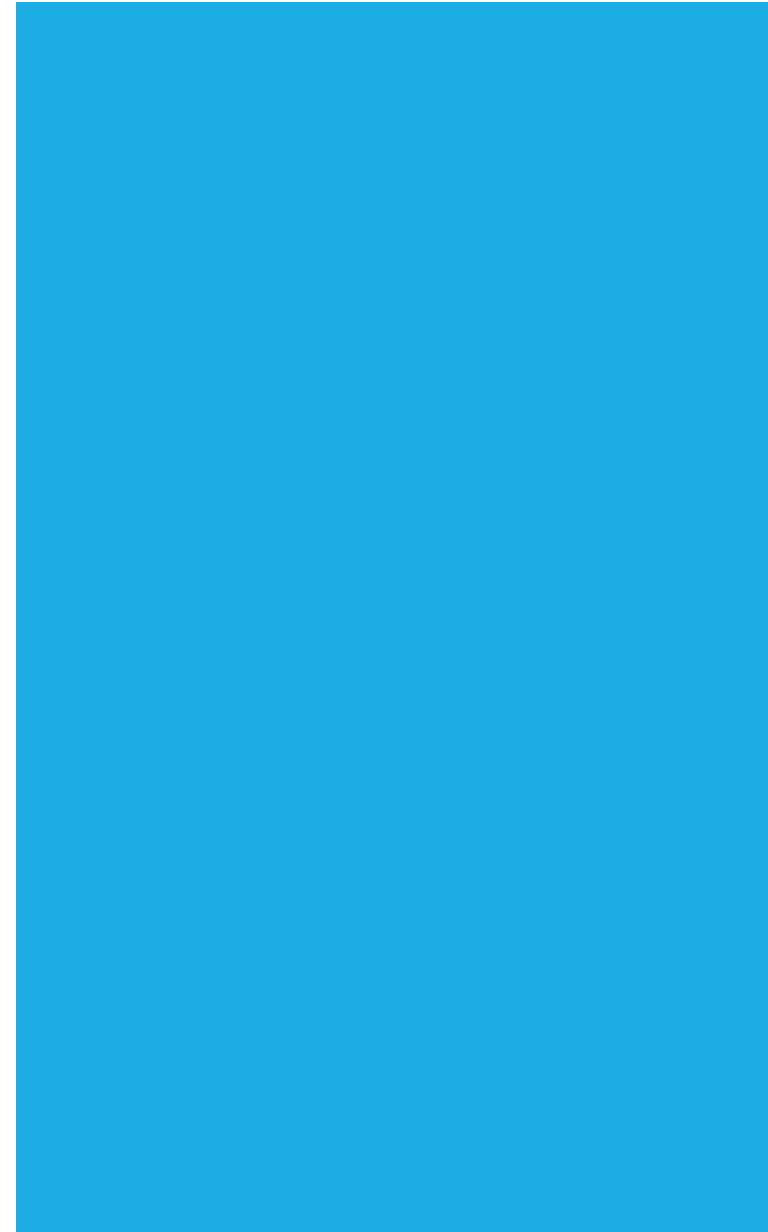
KEYBOARD SETTINGS

- Can see the shortcut for a command in:
 - Menu items
 - Quick Launch
 - Tools → Options → Environment → Keyboard

- Mapping scheme
 - Type part of the command
 - If you want to change the shortcut, it checks and shows you if that combination is already used for
 - Reset mapping scheme

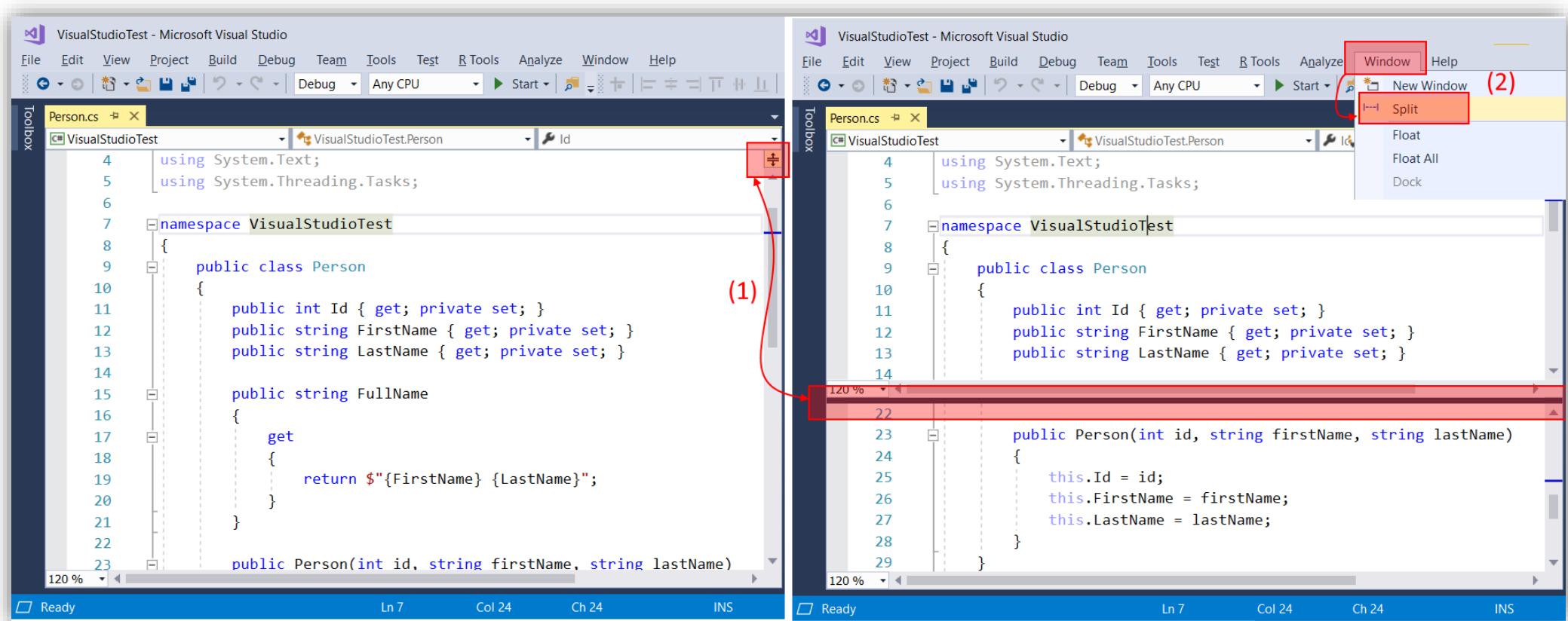


LAYOUT



FILE - SPLIT IN 2 VERTICAL PARTS

1. Icon in top right corner of the file → drag down to split (or up to remove split)
2. Window → Split

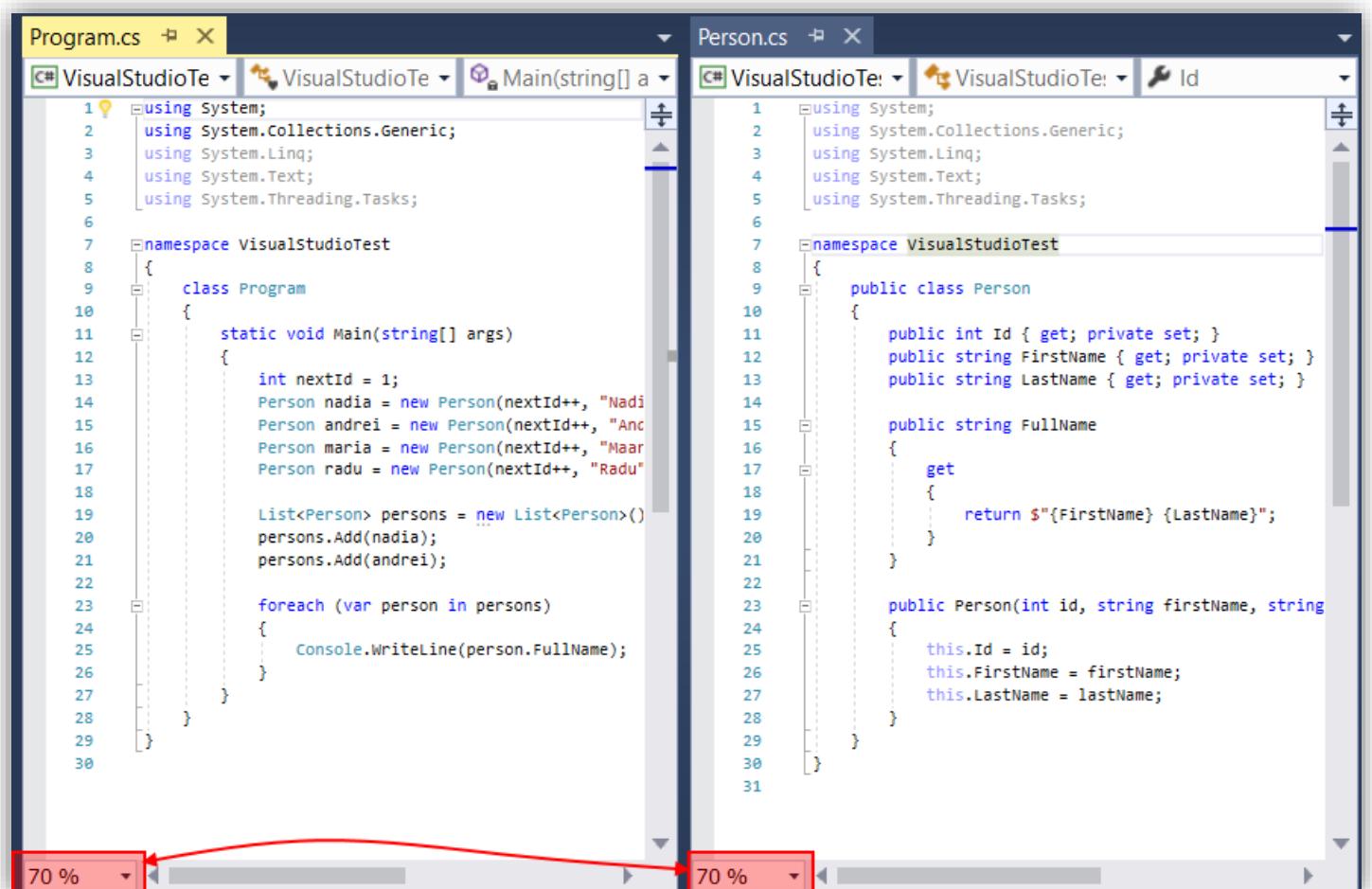


FILE - ZOOM

- The same zoom for all opened files

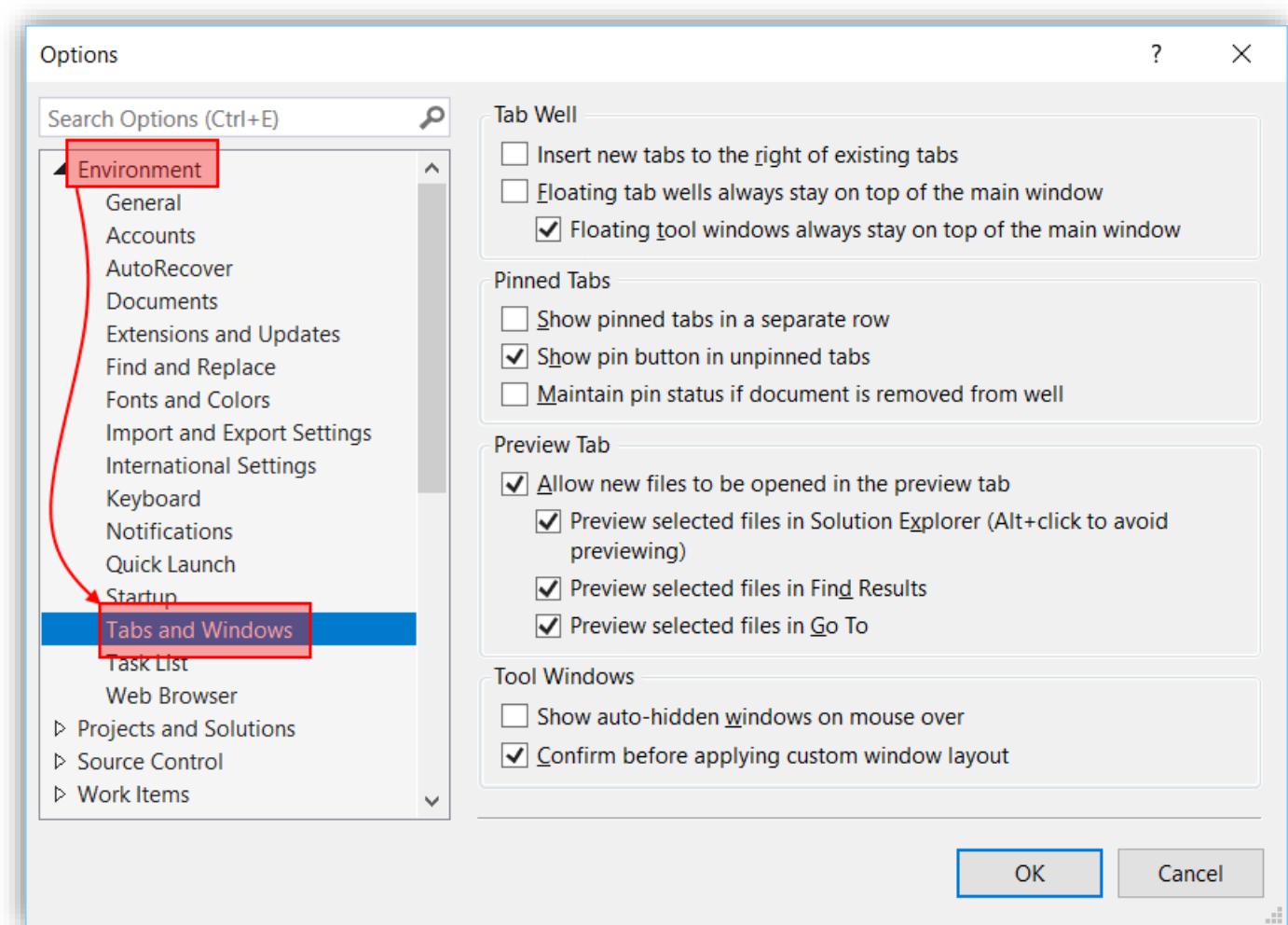
- Modify:

- Type zoom value
- Select from dropdown
- Ctrl + Mouse Scroll



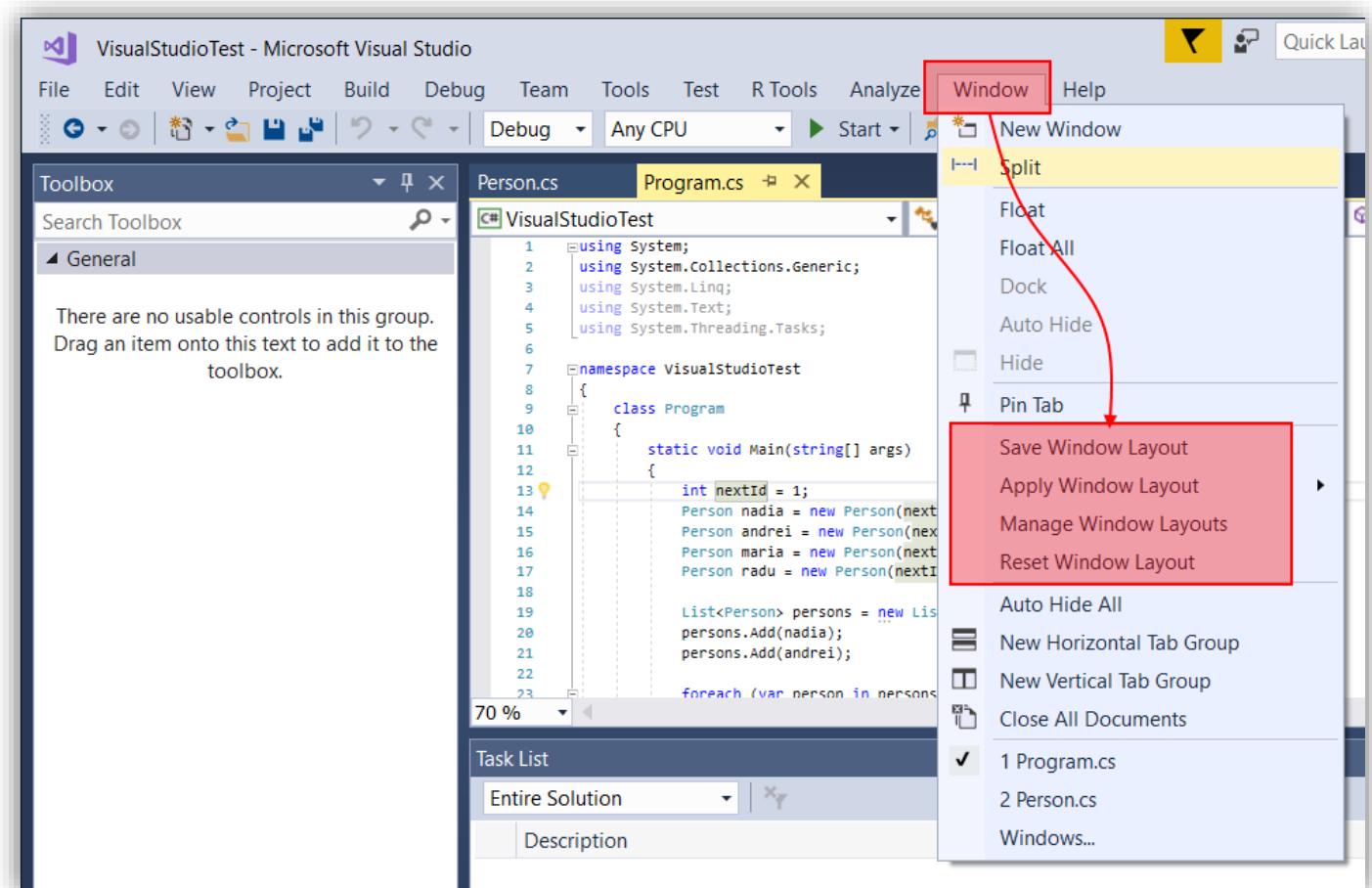
WINDOWS LAYOUT

- Move with Drag & Drop
- Float (Ctrl + move)
- Dock
- Dock as tab document
- Hide
- Auto-hide
- Pin
- Move to another monitor
- Tab Groups
 - Ctrl + Click – Select multiple tabs and move in a new window



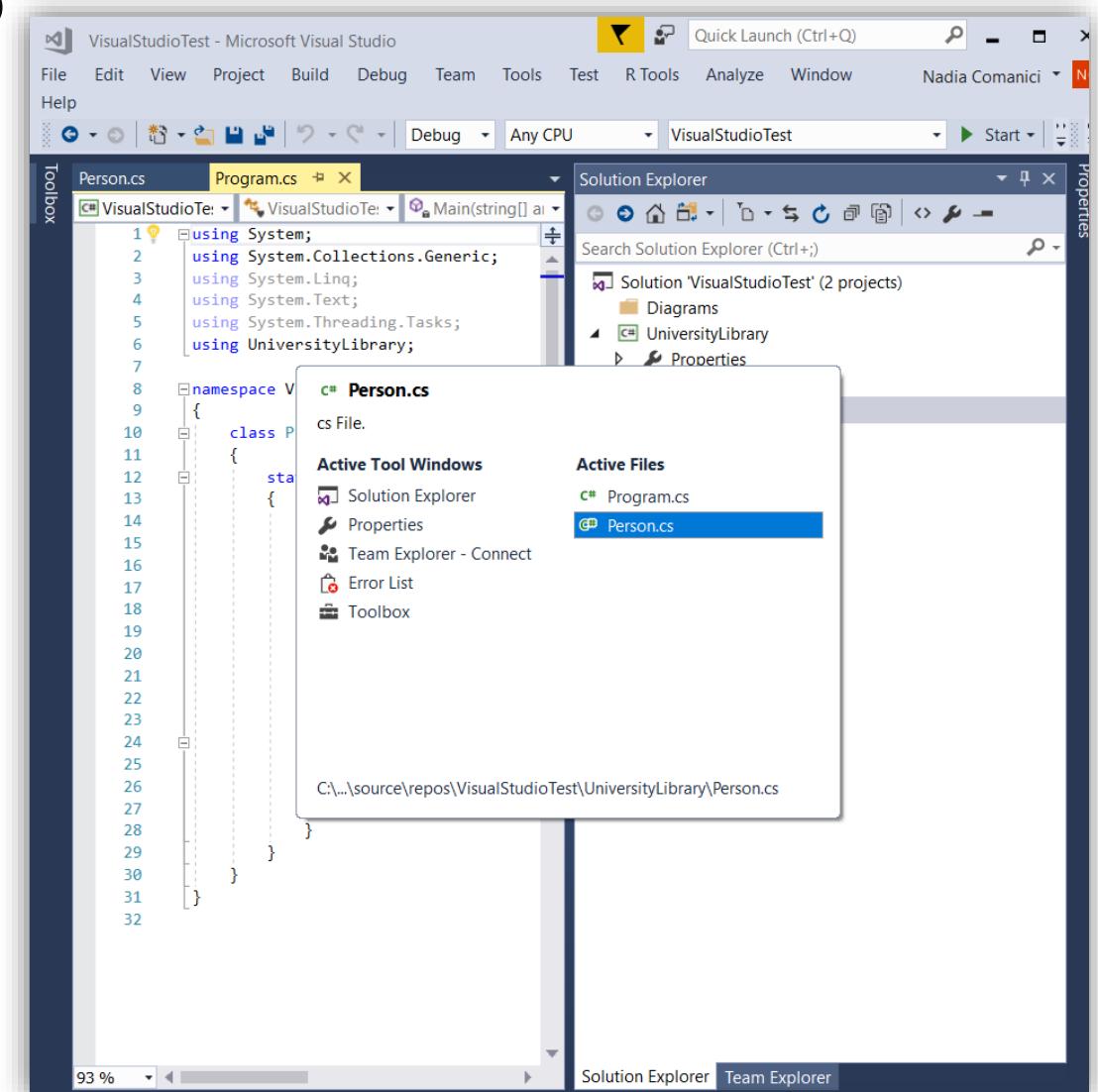
MANAGE LAYOUTS

- Different layouts for
 - Editing
 - Debugging
- Working with layouts
 - Window
 - Save Layout
 - Give a name
 - Apply Layout
 - Select from saved layouts
 - Manage layouts
 - Up to 10 layouts
 - Reset Layout



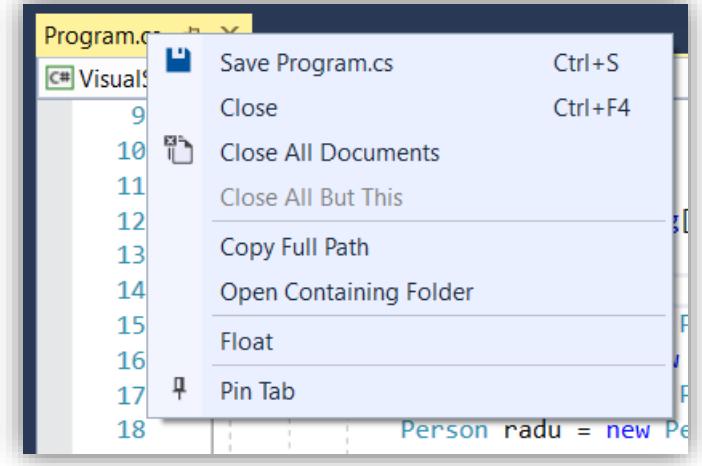
NAVIGATION BETWEEN TABS

- Ctrl + Tab = navigation between tabs (active document)
- Use left/right arrow keys from tool window

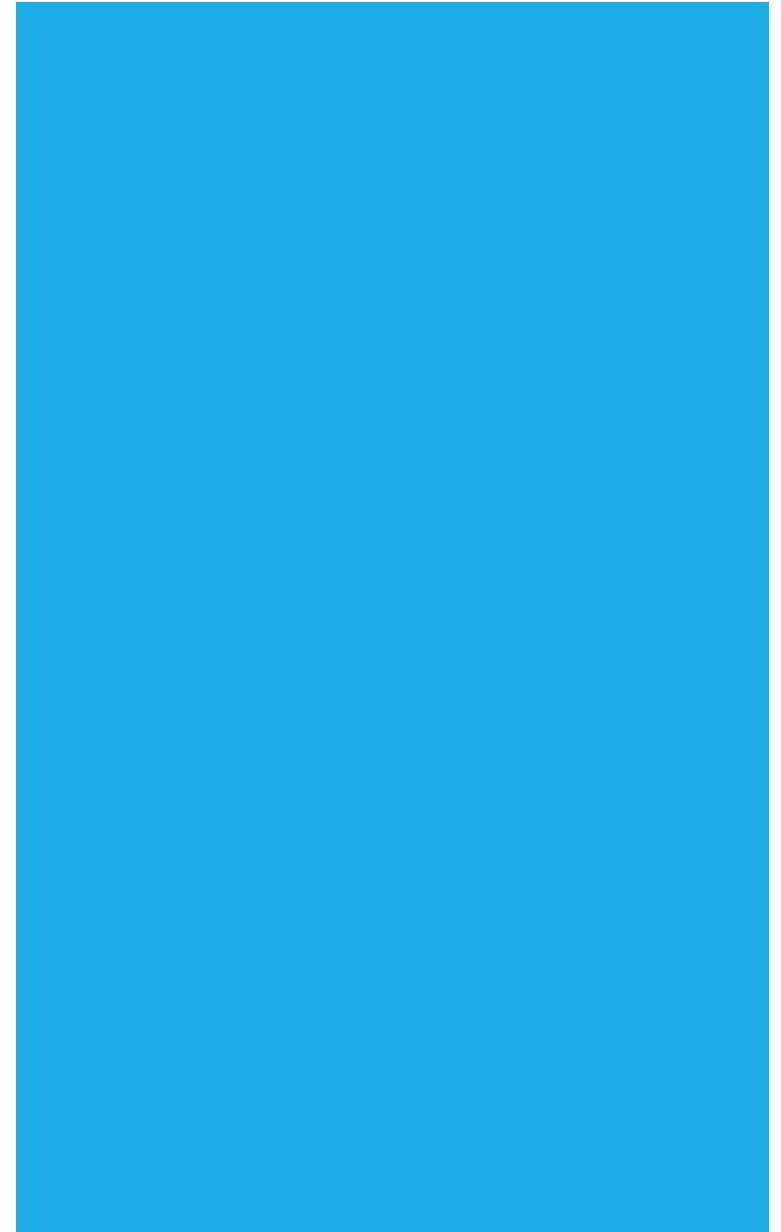


TAB CONTEXTUAL MENU

- Right click on a Tab
 - Close Operations:
 - Close
 - Close all documents
 - Close all but this
 - When closing a file, undo/redo history is lost
- Copy Full Path
- Open Containing Folder
- Float
- Pin

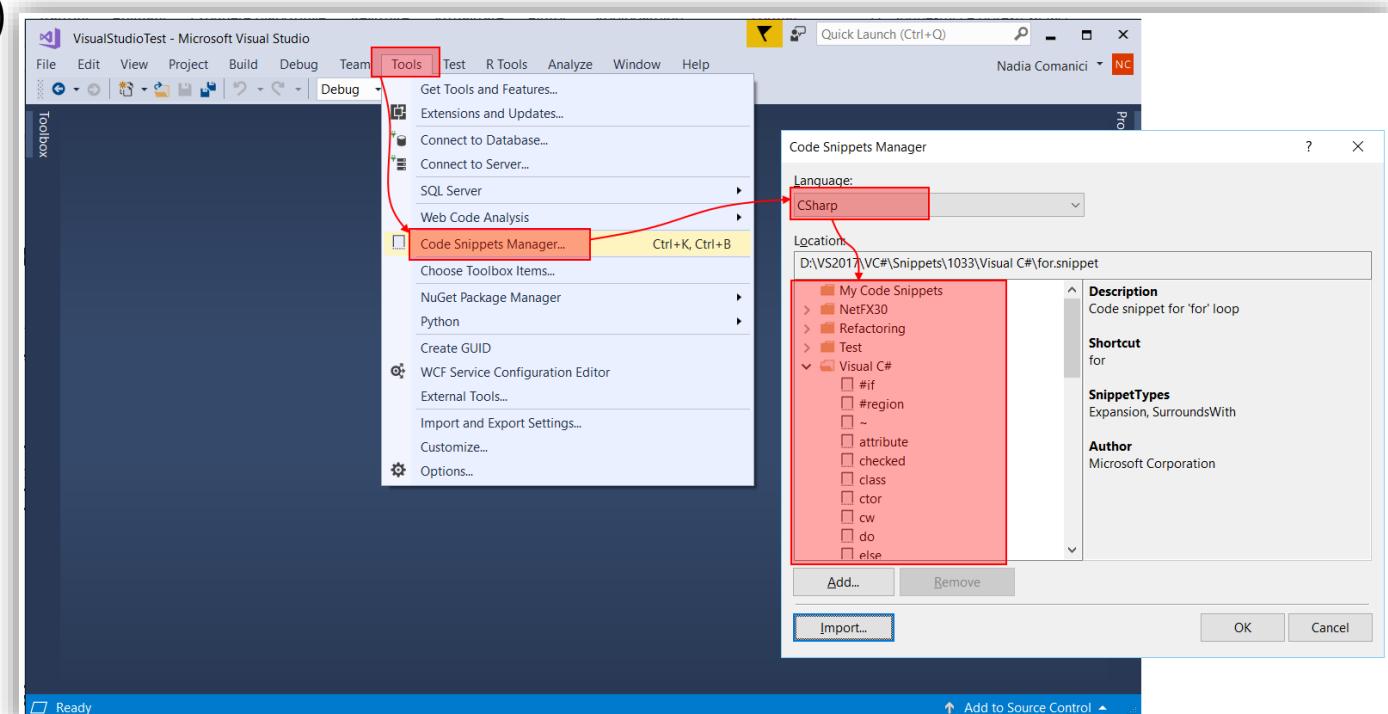


CODE SNIPPETS



CODE SNIPPETS

- Tools → Code Snippets Manager → Language
- Use existing snippets
 - „for”/„prop” + tab + tab → generated snippet
 - Right click → insert snippet
 - Use Tab to cycle between placeholders (yellow)
- Add and import snippets



CREATE CODE SNIPPETS

```
<?xml version="1.0" encoding="utf-8"?>
<CodeSnippets xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
<CodeSnippet Format="1.0.0">
  <Header>
    <SnippetTypes>
      <SnippetType>Expansion</SnippetType>
    </SnippetTypes>
    <Title>Prop-PublicGet-PrivateSet</Title>
    <Author>[USERNAME]</Author>
    <Description></Description>
    <HelpUrl></HelpUrl>
    <Shortcut>propP</Shortcut>
  </Header>
  <Snippet>
    <Declarations>
      <Literal Editable="true">
        <ID>.PropertyType</ID>
        <ToolTip>Property Type</ToolTip>
        <Default>.PropertyType</Default>
        <Function></Function>
      </Literal>
      <Literal Editable="true">
        <ID>PropertyName</ID>
        <ToolTip>Property Name</ToolTip>
        <Default>PropertyName</Default>
        <Function></Function>
      </Literal>
    </Declarations>
    <Code Language="csharp" Delimiter="$"><![CDATA[public $.PropertyType$ $PropertyName$ { get; private set; }]]></Code>
  </Snippet>
</CodeSnippet>
</CodeSnippets>
```

➤ In C:\Users\[User]\Documents\Visual Studio 2017\Code Snippets

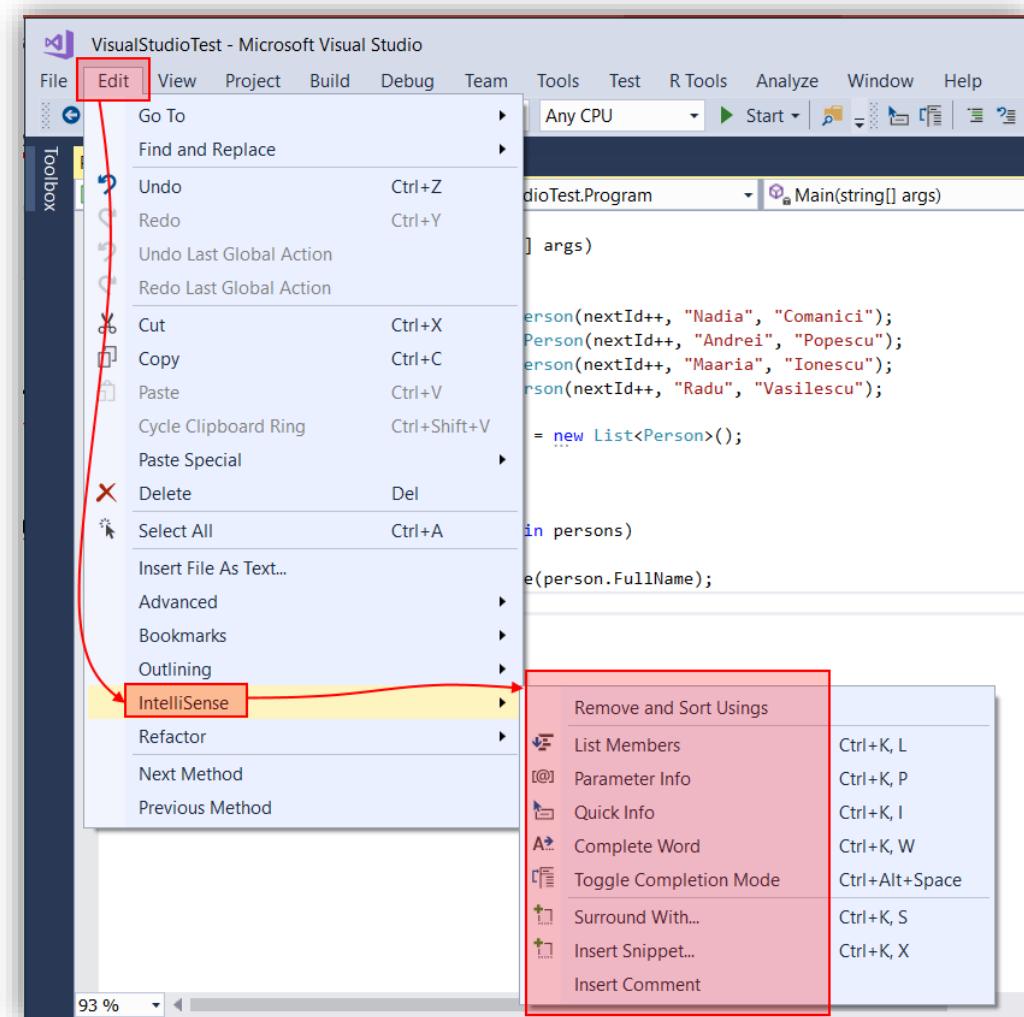
INTELLISENSE

INTELLISENSE

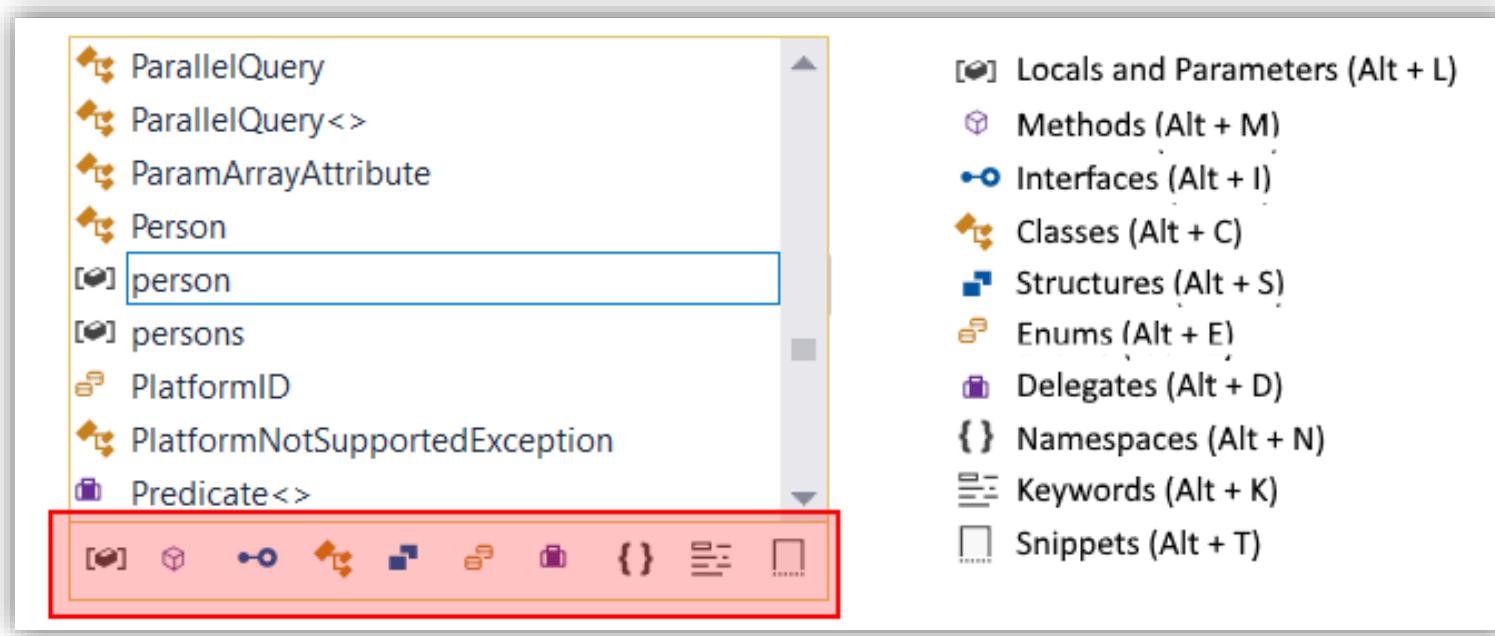
Shortcut	Action
Enter Punctuation Shift + Space	Start typing and to complete word using intellisense suggestion
Ctrl + J Ctrl + Space	Get it back
Hold Ctrl	Make panel semitransparent
Esc	Close

INTELLISENSE

- Edit → Intellisense
 - Remove and Sort usings
 - Toggle completion mode
 - Surround with ...

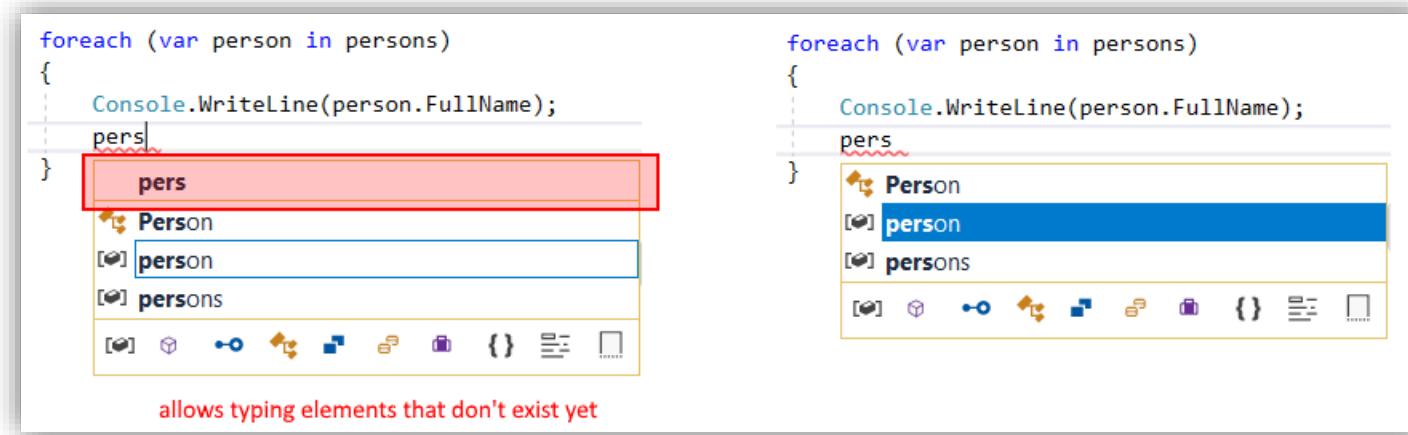


INTELLISENSE – FILTER TOOLBAR



AGGRESSIVE INTELLIGENCE

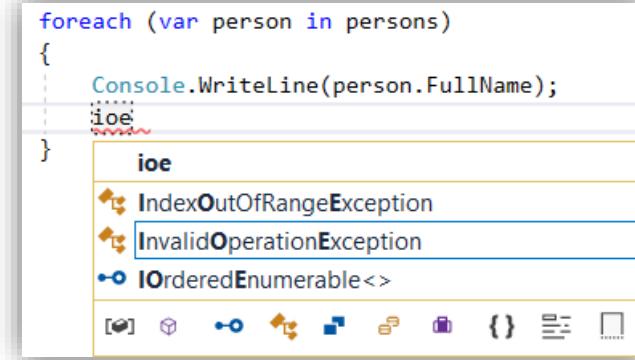
- Edit → Intellisense → Toggle Completion Mode
 - On = automatically completes the first match in list
 - Off = can add names that don't exist yet
- Switch between modes: Ctrl + Alt + Space



ADVANCED INTELLISENSE

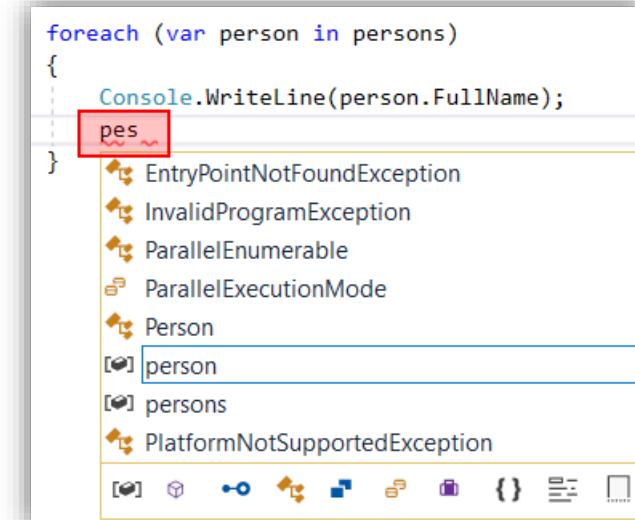
➤ Word part intellisense

- Starting with 2015
- Type upper case letters to get the object



➤ Fuzzy search

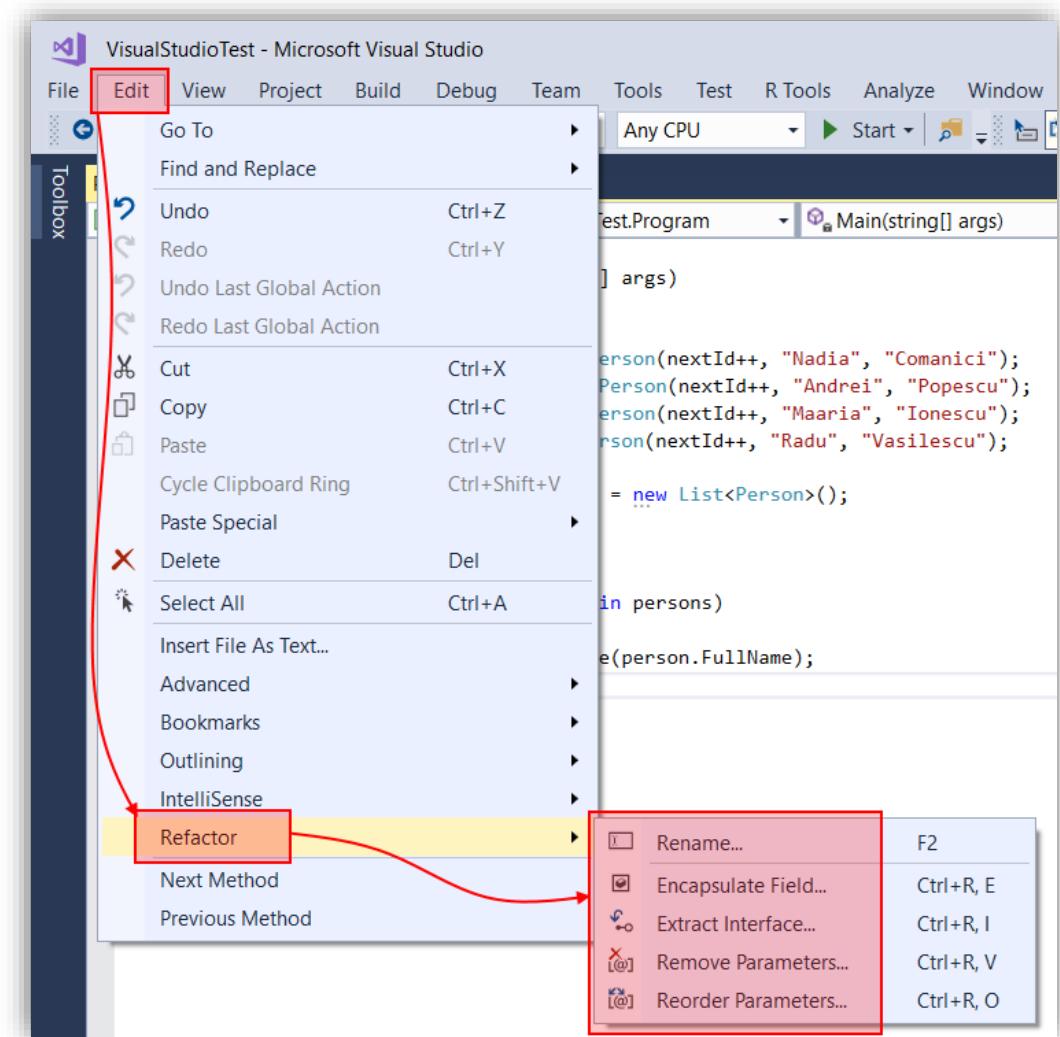
- Useful for misspellings



REFACTOR

REFACTOR

- **Rename**
 - Replaces all occurrences and shows the results of the replacement
- **Encapsulate fields**
 - Generates basic get and set for a field
- **Extract interface**
 - Creates an interface based on the current class and adds it in a new file
- **Remove parameters**
 - Removes parameters from the method definition and all occurrences of this method
- **Reorder parameters**
 - Changes the order of the parameters in method definition and all occurrences of this method



SOLUTION & PROJECTS



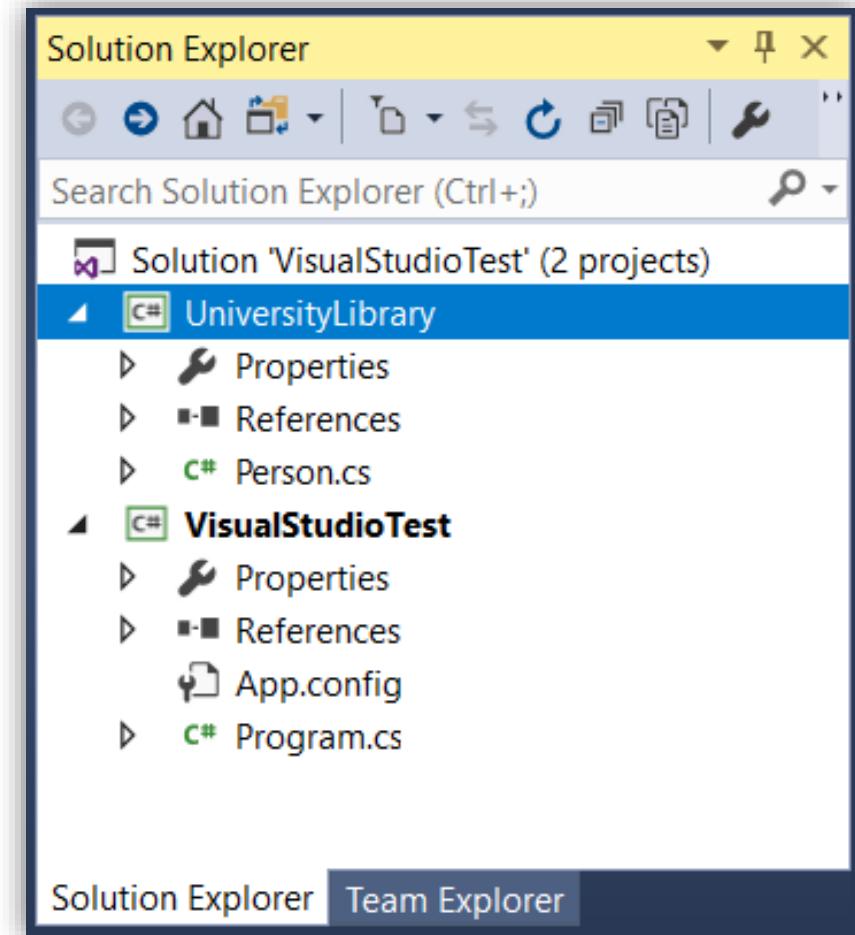
SOLUTIONS AND PROJECTS

➤ Each solution

- Can contain one or more projects
- Has a .sln file that contains information about:
 - The projects are inside it
 - Minimum Visual Studio Version
 - Platforms and configuration per solution

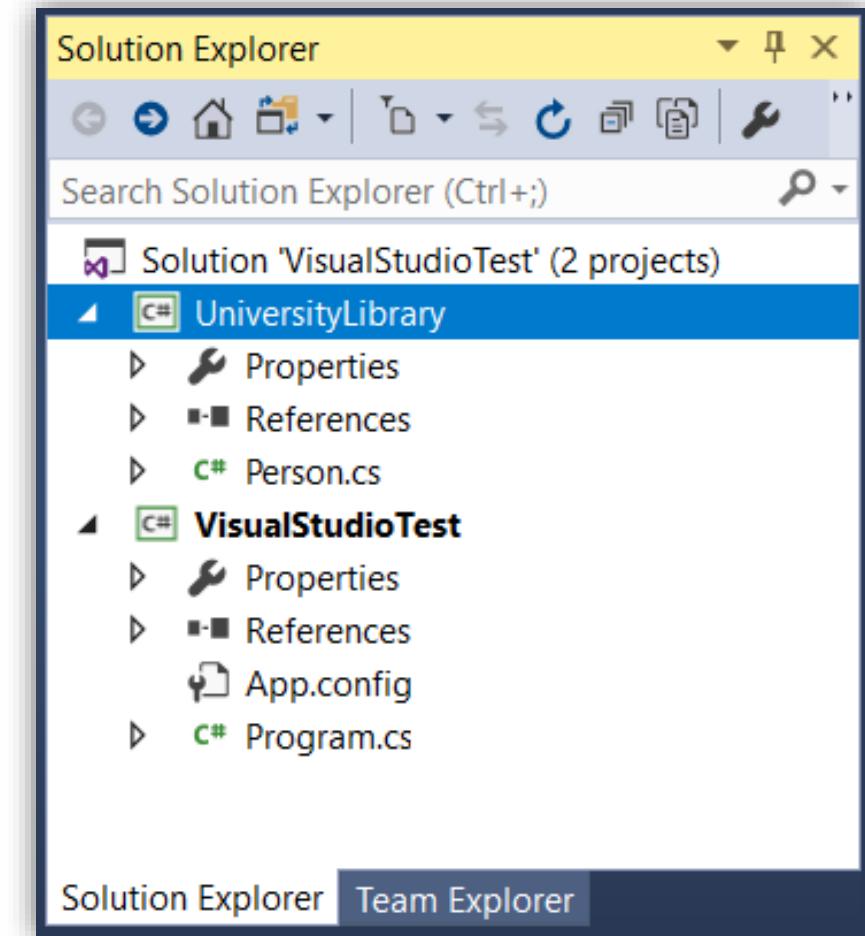
➤ Each project

- Has a .csproj file contains information about
 - The files and references inside a project
 - Platforms and Configurations per project



SOLUTION EXPLORER

Shortcut	Action
Ctrl + Alt + L View -> Solution Explorer	Open Solution Explorer
Ctrl + ;	Filter files/references
Hold Ctrl and Click	Multiple file selection



ADDING REFERENCES AND USINGS

Adding a reference to the project	Using in File
<ul style="list-style-type: none">- At project level- If there is no reference added, we cannot use the classes from that assembly- Even if a solution has multiple projects, they don't know about each other, so you have to add the references manually- Ctrl + . = Add Reference	<ul style="list-style-type: none">- At file level, at the begining of the file- Define the used namespaces<ul style="list-style-type: none">- Gray – unused in this file- Black – used in this file- Allows using a class without specifying its full qualified name each time- Ctrl + R, Ctrl + G = Sort and remove usings in a file

SOLUTIONS AND PROJECT FOLDERS

➤ Solution Folders

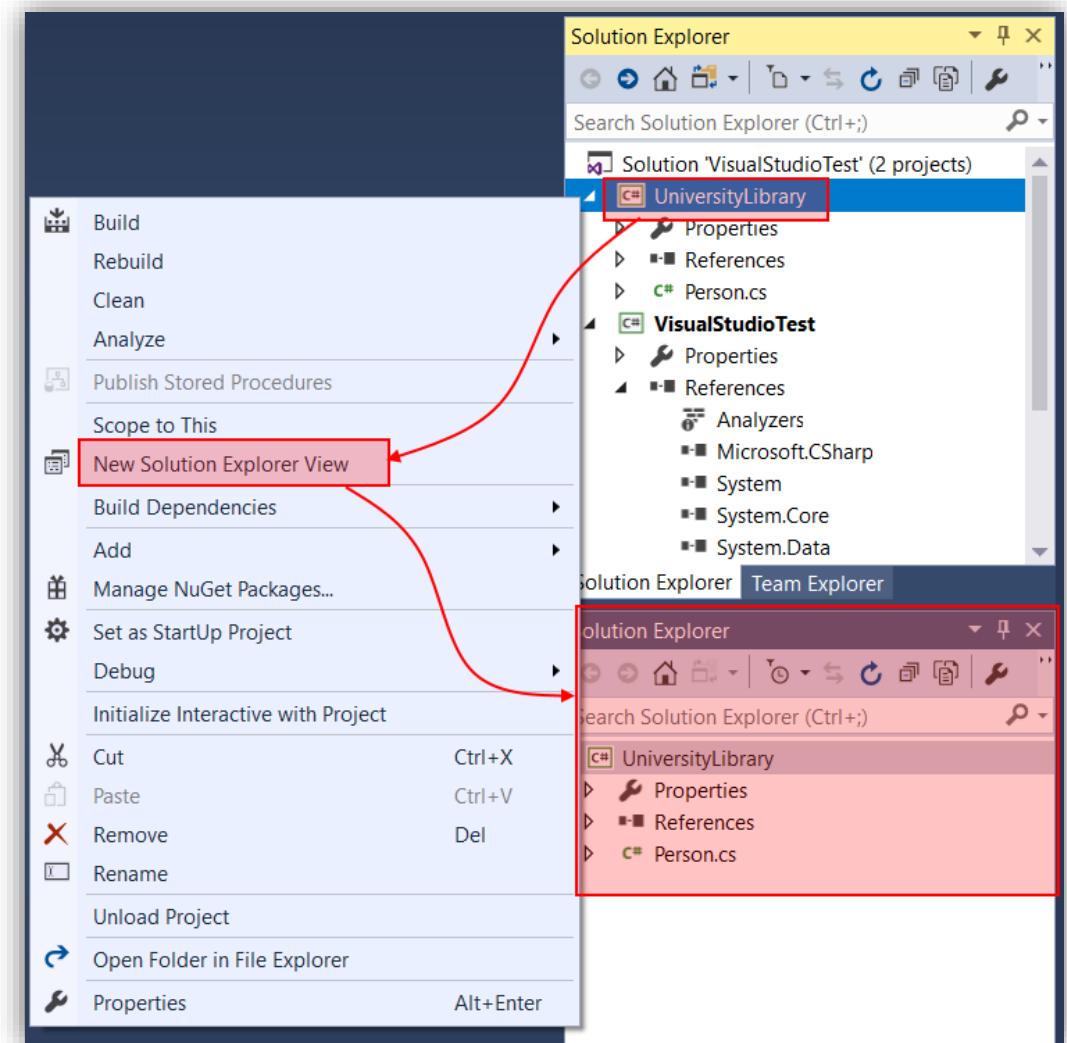
- Virtual folders, not on disk
- Groups parts of the app

➤ Project Folders

- Real folders on disc

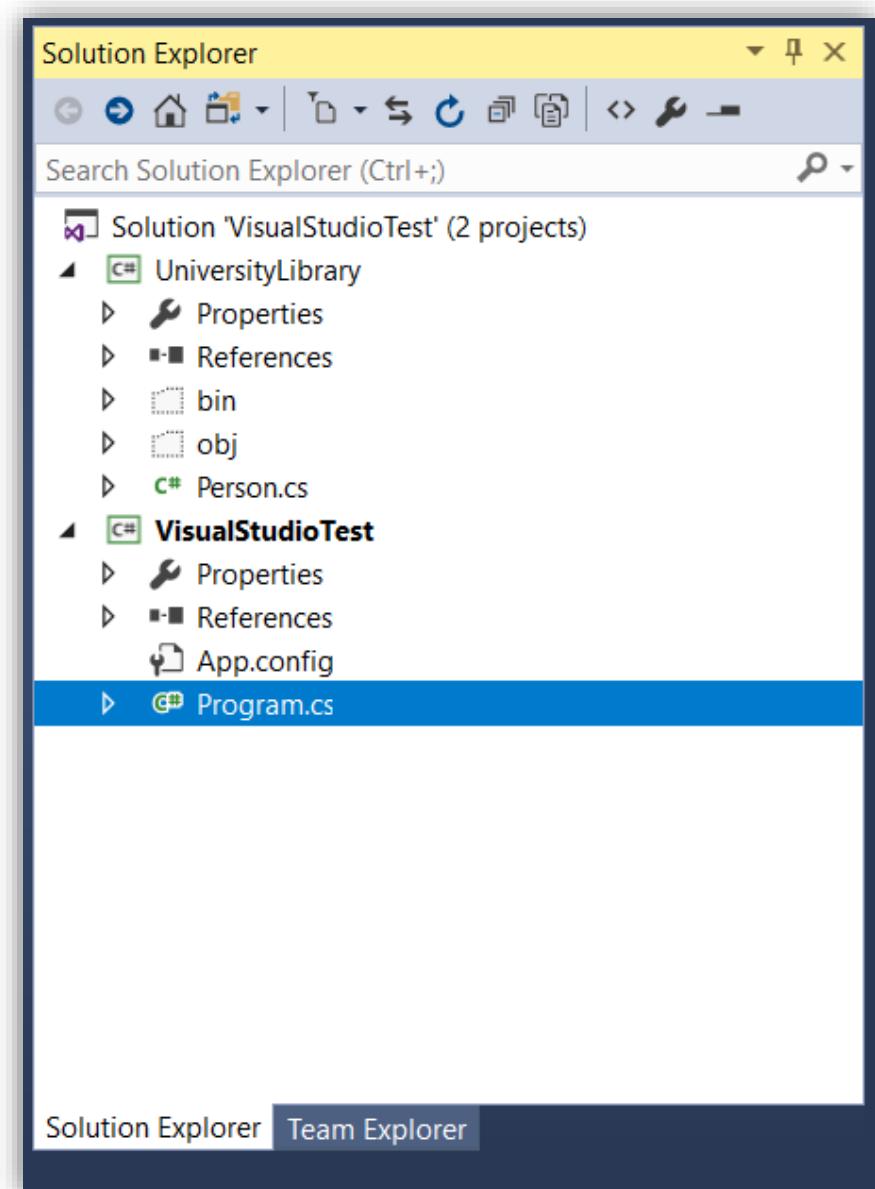
MULTIPLE SOLUTION EXPLORERS

- Right click a file → new Solution Explorer view
- Home = root of the view
- You can have a Solution Explorer for each project, and each SE has its own history
- Arrows to navigate
 - Back
 - Forward
- Dependency graph



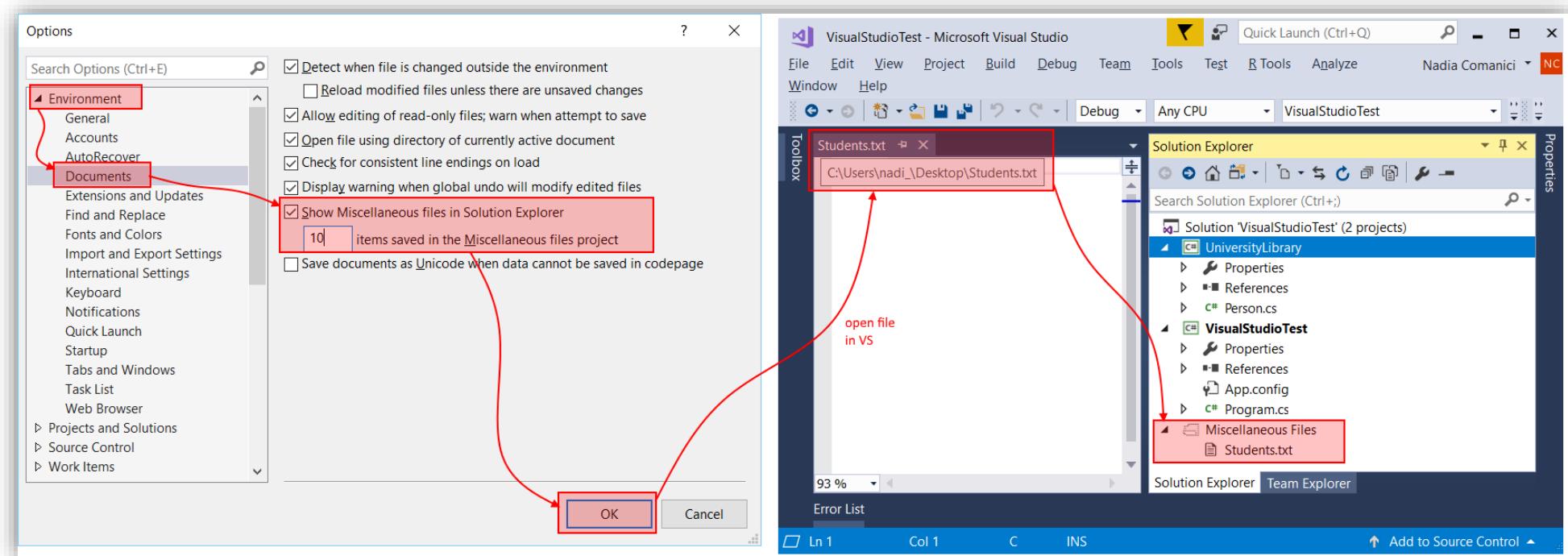
SOLUTION EXPLORER OPTIONS

- Filter
 - Pending changes
 - Open files
- Sync with Active Document
 - Always highlight the active file in SE
 - Enable/disable in Tools → Options → Projects & Solutions → General → Track active items in SE
- Refresh
- Collapse All
- Show All Files
- View Code
- Properties
- Preview selected items



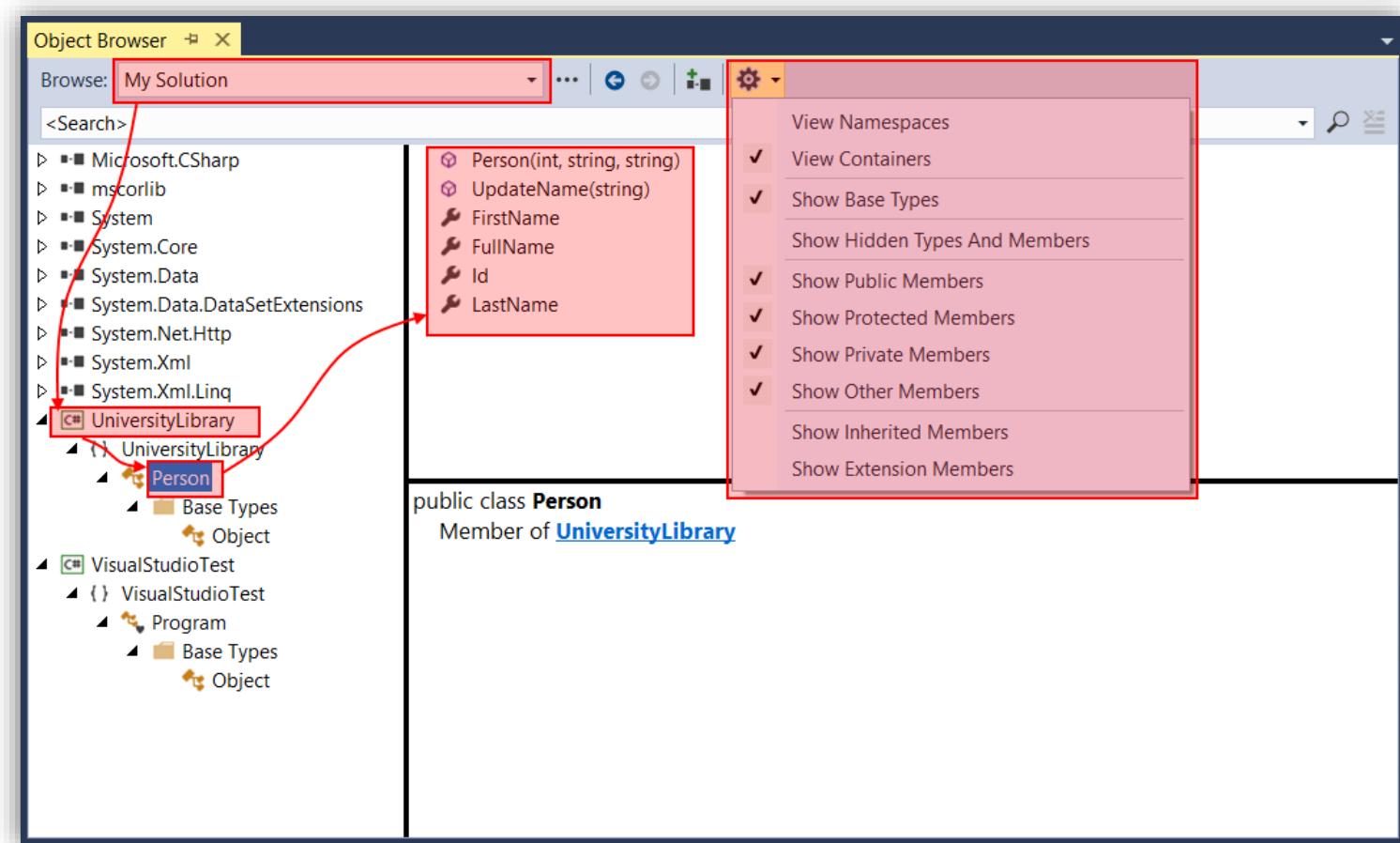
MISC FOLDER

- Files that are not part of your solution, but associated with your code
- Enable Misc Folder:
 - Tools → Options → Environment → Documents → Show Miscellaneous files in Solution Explorer → Number of Saved documents
- Useful when debugging from another project



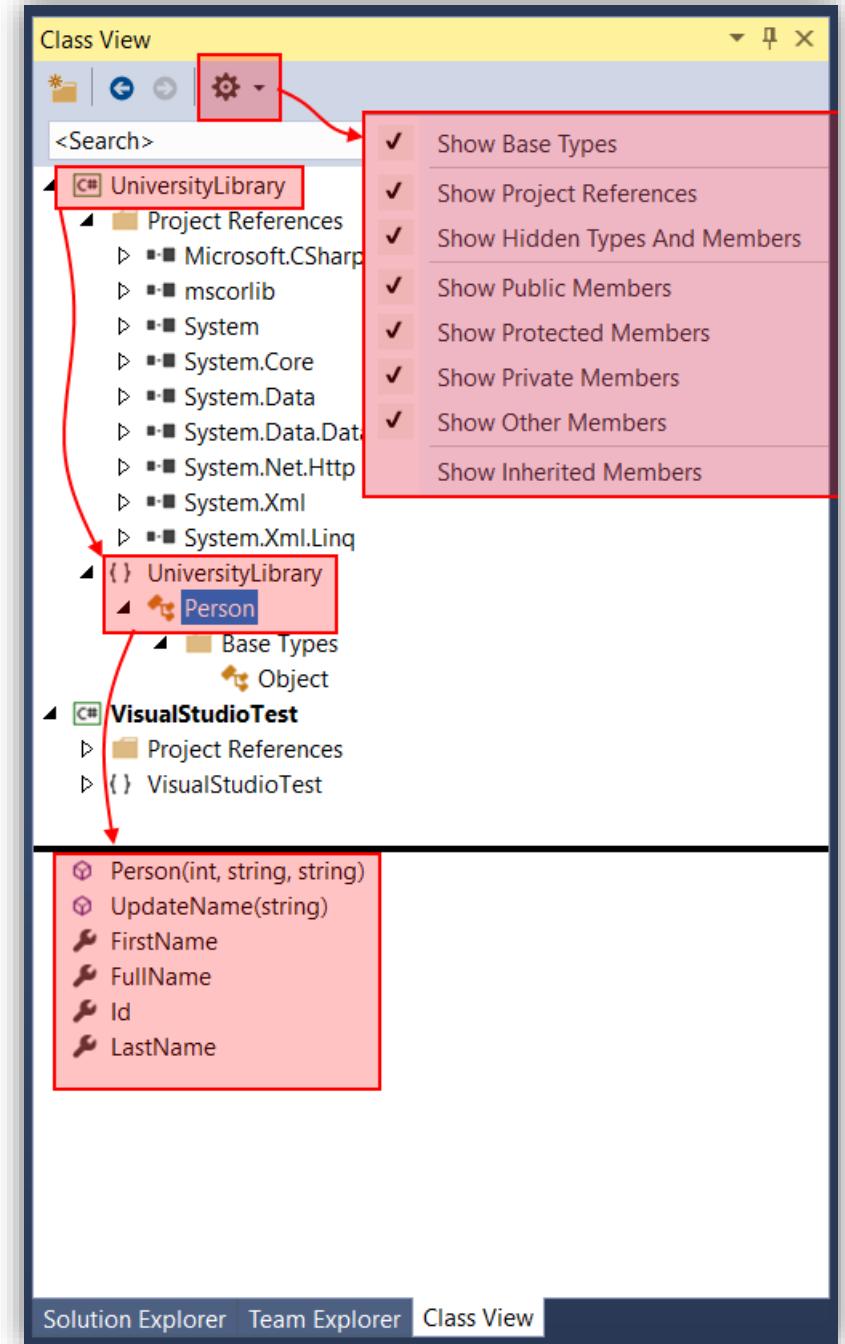
OBJECT EXPLORER

- Choose what to browse
 - Current solution
 - .Net frameworks
 - COM dlls
 - Etc
- Filter by search string
- Filter displayed results by what members to show



CLASS VIEW

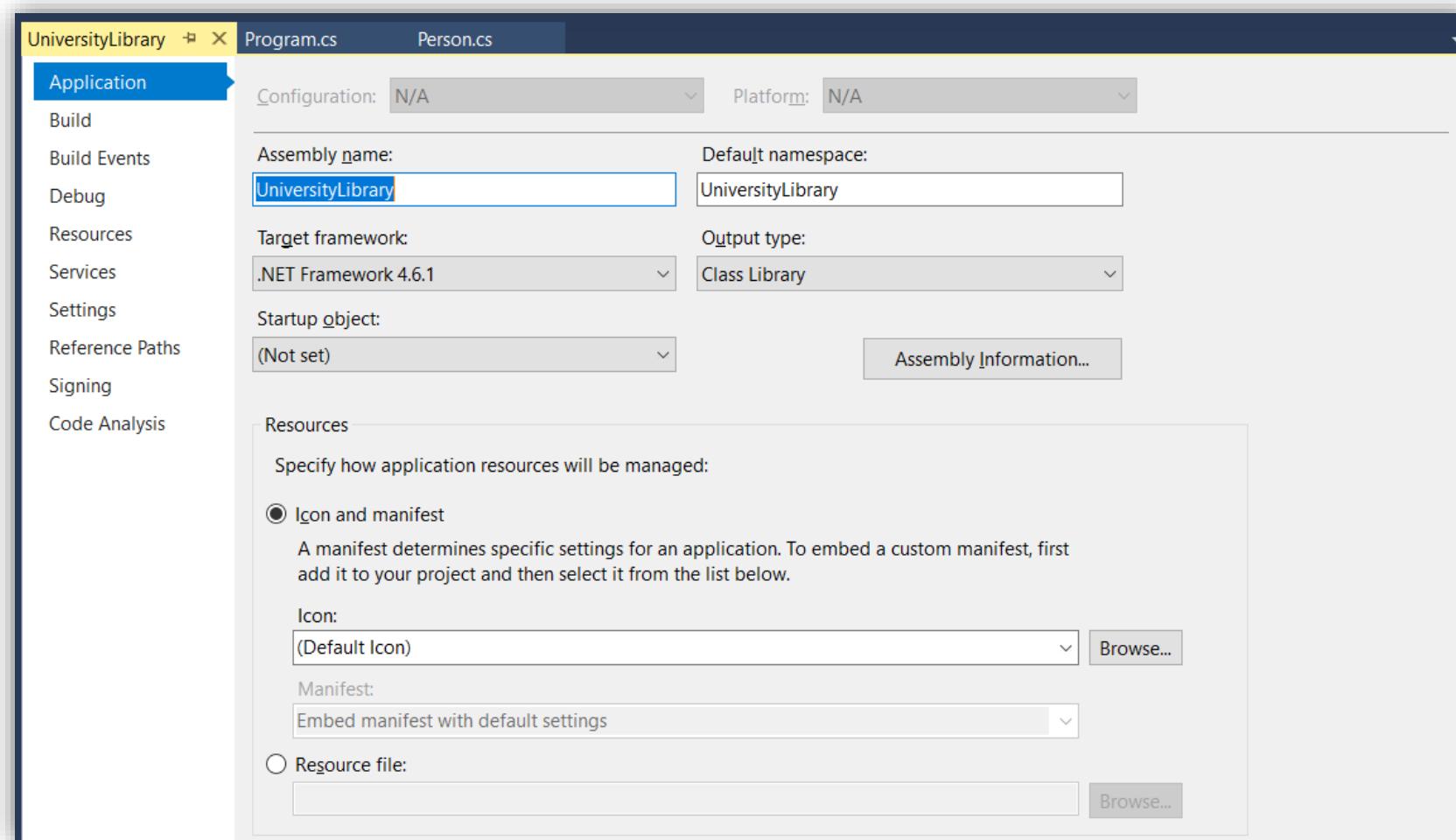
- Useful for
 - Partial classes
 - Multiple classes in same file
- Inheritance Hierarchy
- Can filter displayed results by what members to show (public, private, protected etc)
- Icons: <https://docs.microsoft.com/en-us/visualstudio/ide/class-view-and-object-browser-icons?view=vs-2017>



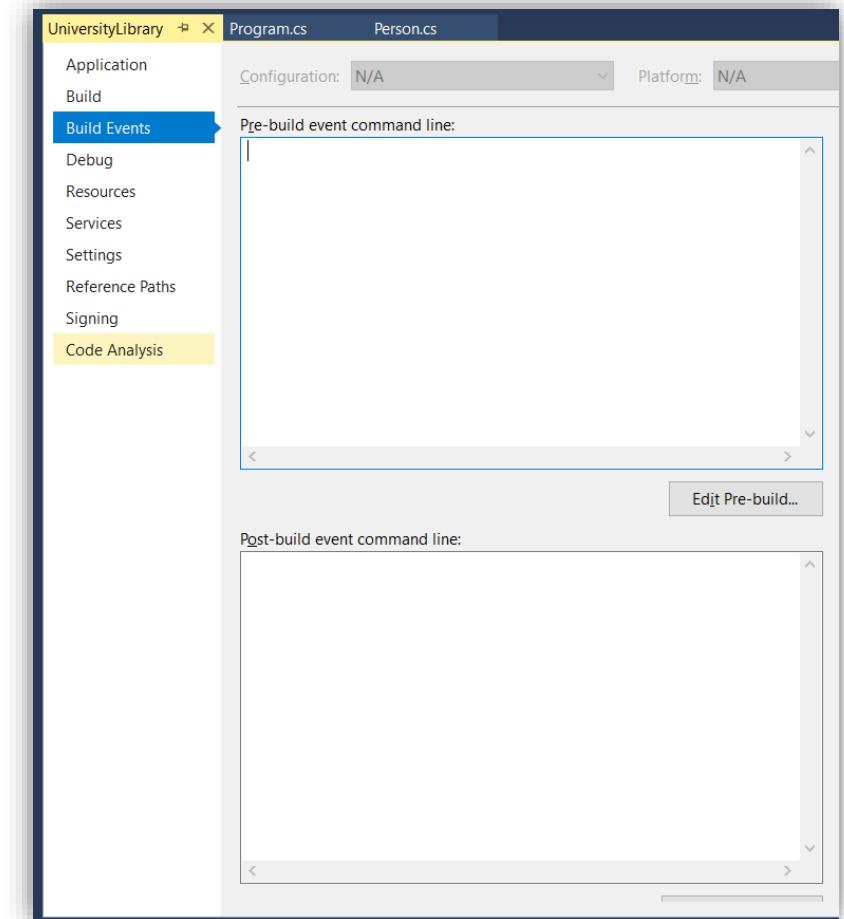
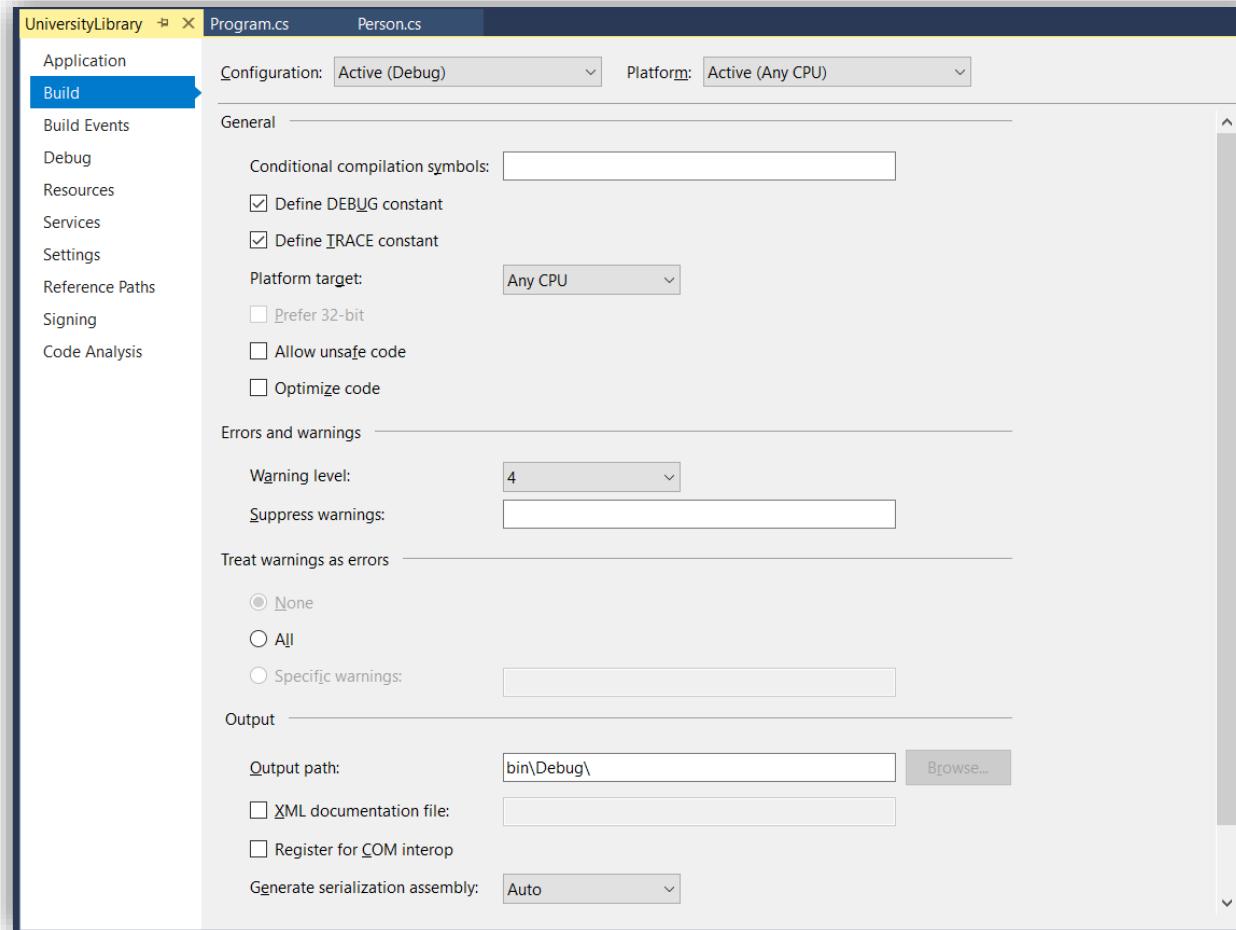
PROJECT – CONTEXTUAL MENU

- Build operations:
 - Clean
 - Build
 - Rebuild
 - Batch build
- Generated Folders:
 - Obj
 - Bin
- Project Dependencies
 - Project Build Order
- Set as startup Project
- Open Folder in File Explorer

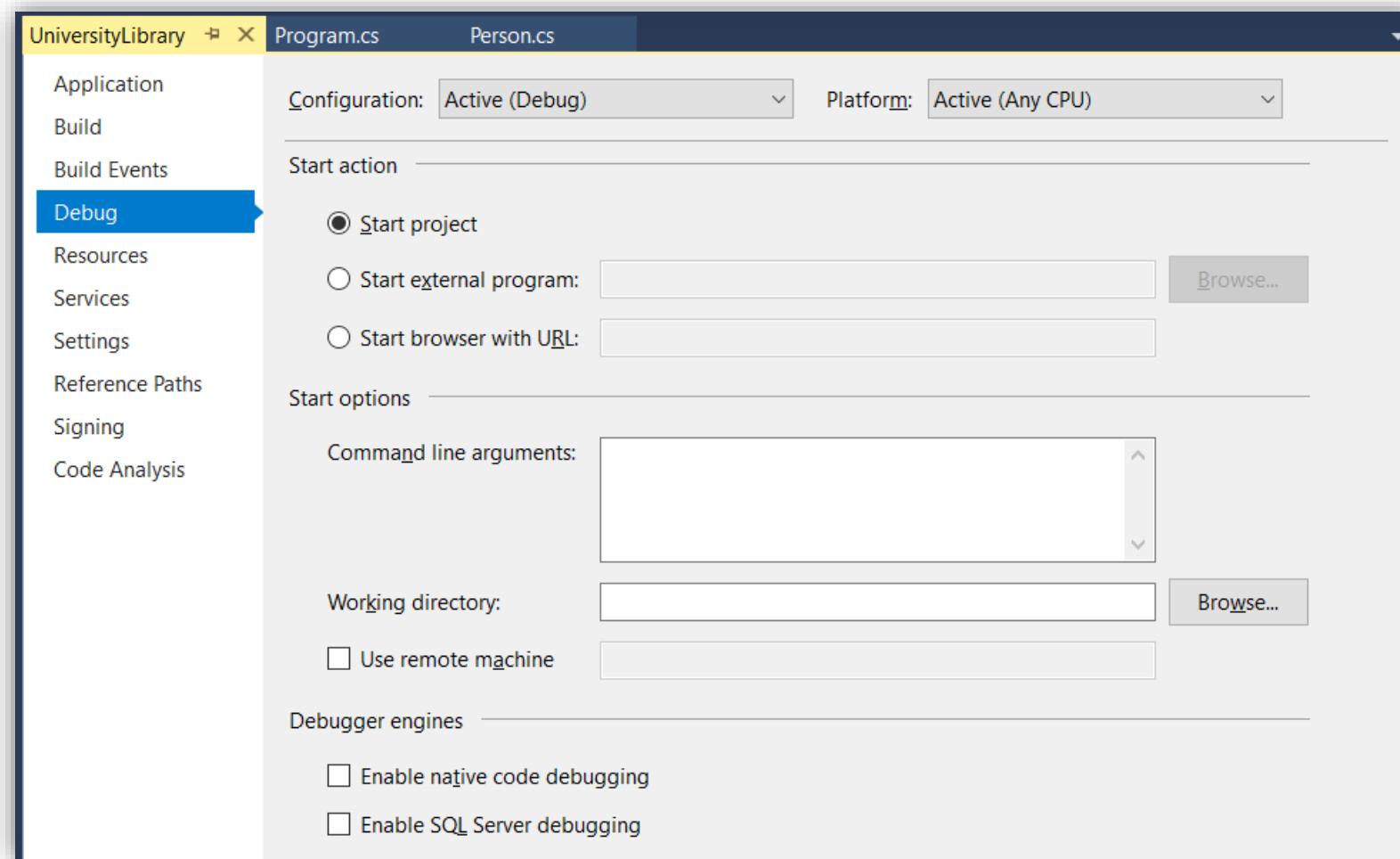
PROJECT PROPERTIES – APPLICATION



PROJECT PROPERTIES – BUILD & BUILD EVENTS

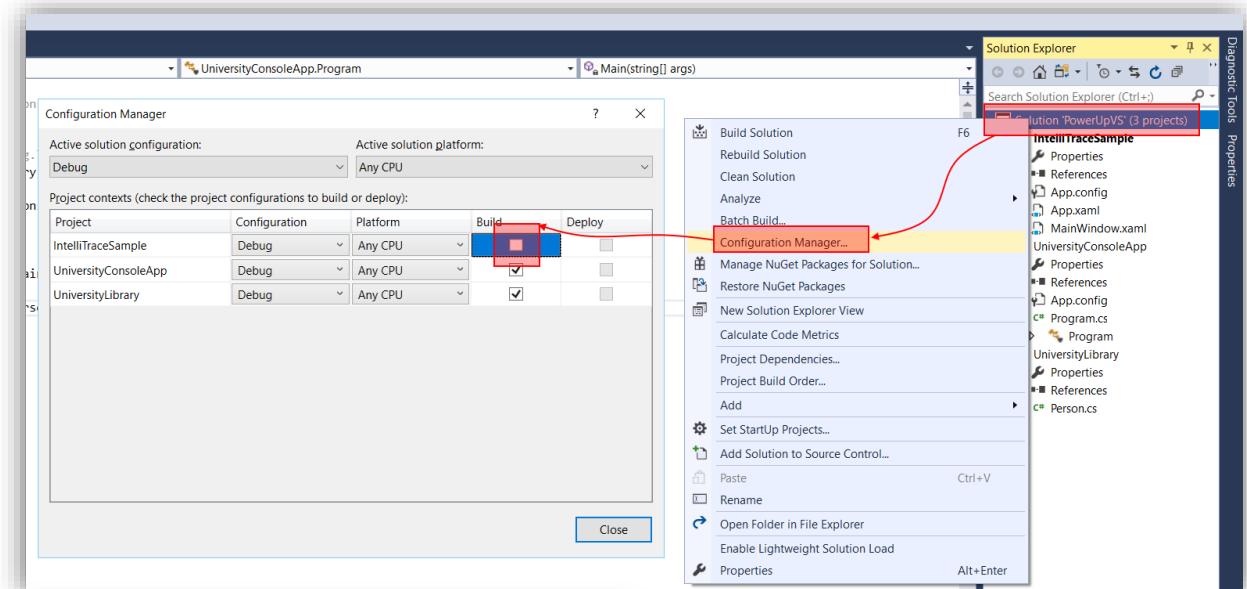


PROJECT PROPERTIES – DEBUG

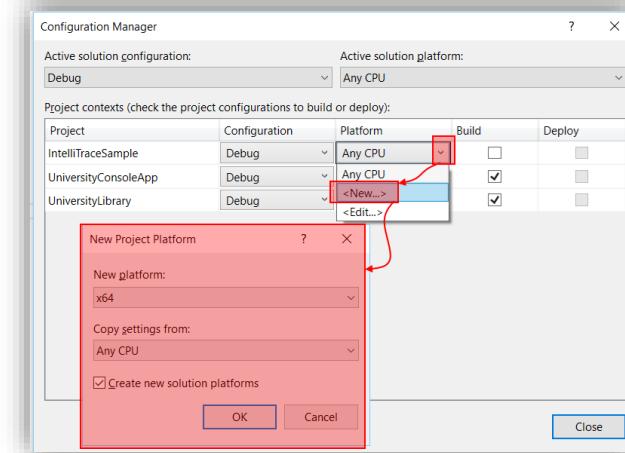


CONFIGURATION MANAGER

- Building only certain project in solution

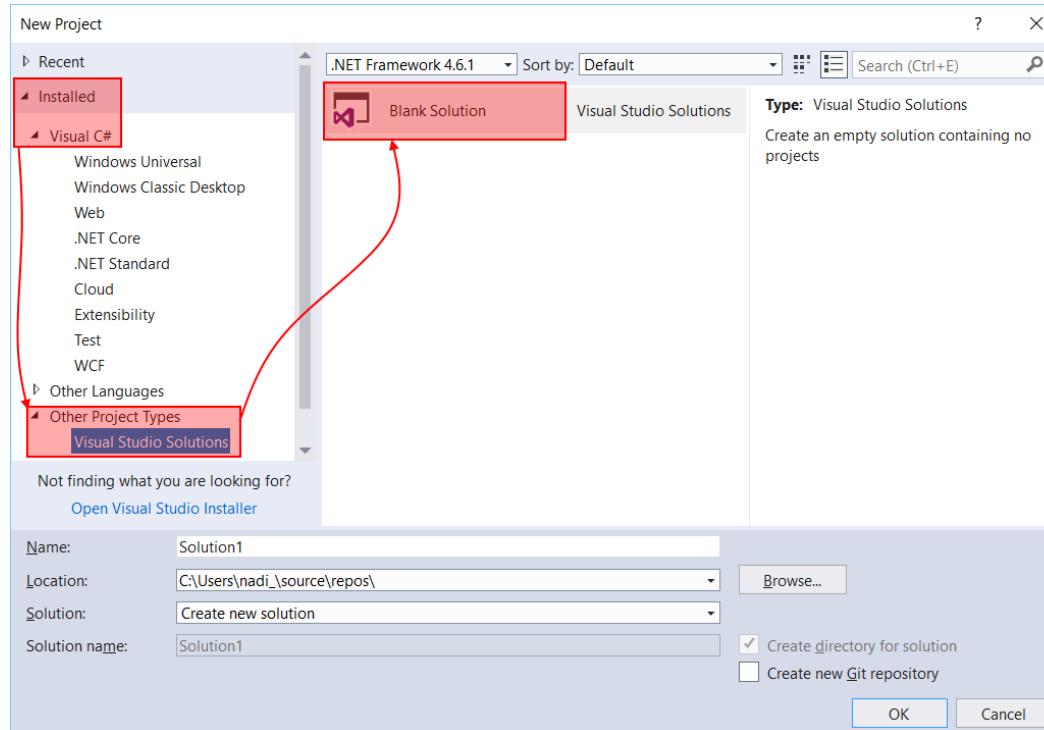


- Define platforms and configurations
 - for projects
 - for solutions



CREATING AN EMPTY SOLUTION

- File → New → Project
- Installed → Templates → Other Project Types → Visual Studio Solutions → Blank Solution



RENAME A PROJECT

1. Right click a project → Rename with new name
2. Right click the project → Properties → Application
 - Change Assembly Name (the name of the dll/executable)
 - Change Default Namespace
3. Replace the old namespace in all the files inside the solution
4. Rename Project Folder on disk
 - Right click the project → Open Folder in File Explorer
 - Close Visual Studio
 - In File Explorer, rename the folder and the .csproj file
 - In File Explorer, open the .sln file with Notepad and set the correct name to the .csproj file

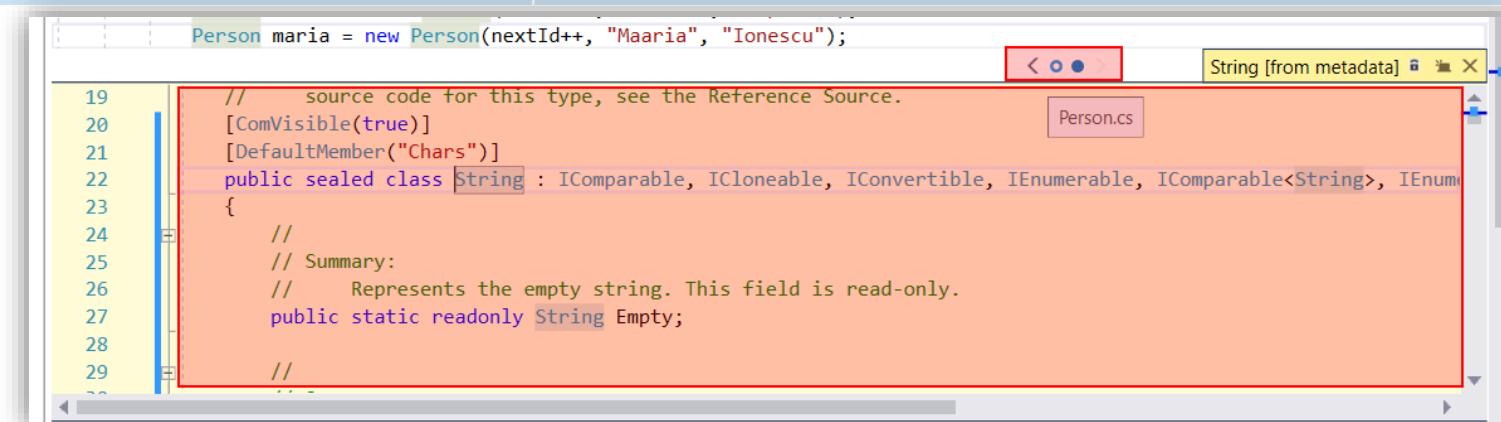
NAVIGATION

NAVIGATION

Shortcut	Action
F12 Right Click → Go To Definition	Go To Definition
Ctrl + F12 Right Click → Go To Implementation	Go To Implementation
Ctrl + -	Navigate Back
Ctrl + Shift + -	Navigate Forward
Shift + F12 Ctrl + K, Ctrl + R Right Click → Find All References	Find All References

NAVIGATION - PEEK

Shortcut	Action
Alt + F12	Peek definition <ul style="list-style-type: none">• Changes in peek are updated in the file too• Go deeper → backwards with Alt + F12 again
Ctrl + Alt + -	Back in peek navigation
Ctrl + Alt + +	Next in peek
Esc	Close Peek
Shift + Esc	Move between peek edit and doc



NAVIGATION BAR

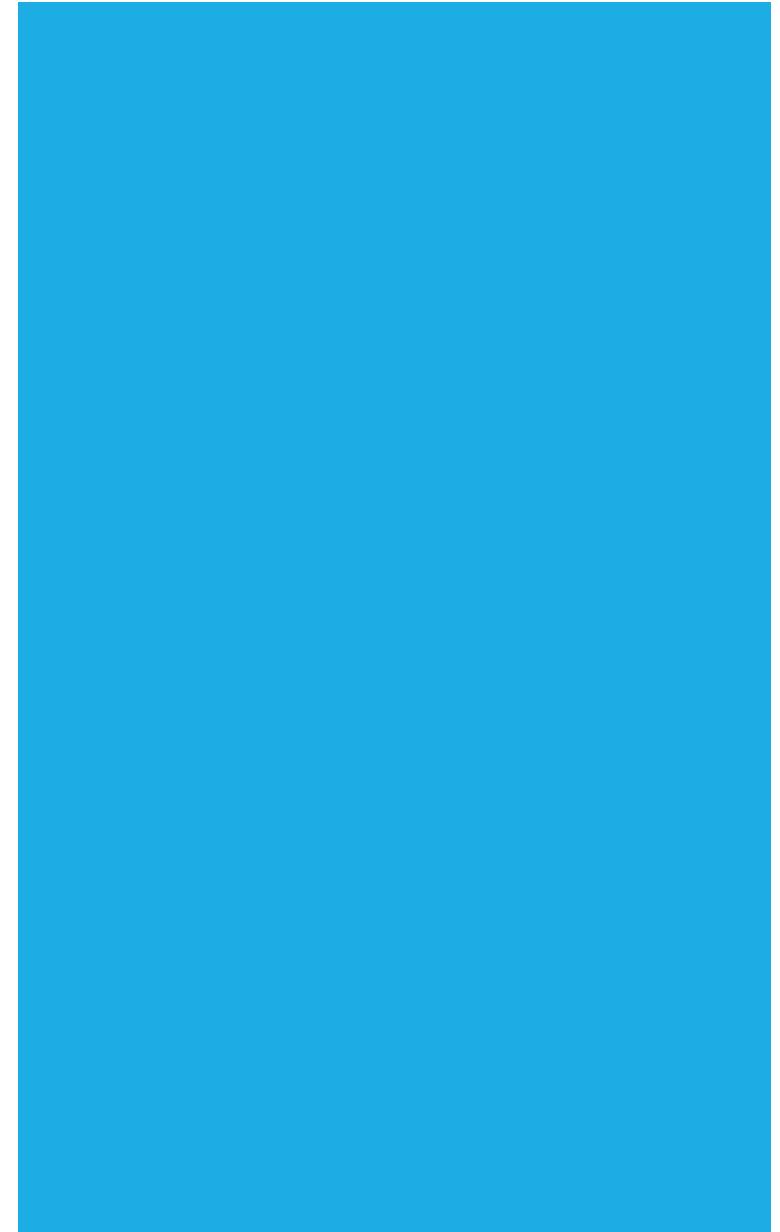
- Has 3 Levels
 - Project in which the current file is
 - Classes in this file
 - Lists of all fields/properties/methods etc inside the selected class
 - Enable/disable Navigation bar
 - Options → Text Editor → All languages → General → Navigation bar



The screenshot shows a Visual Studio code editor with the file "Person.cs" open. The code defines a class "Person" with properties "FirstName", "FullName", "Id", and "LastName", and methods "Person(int id, string firstName, string lastName)" and "UpdateName(string newLastName)". A tooltip is displayed over the "Id" property, listing its members: FirstName, FullName, Id (selected), LastName, Person(int id, string firstName, string lastName), and UpdateName(string newLastName). The "Id" member is highlighted with a red border.

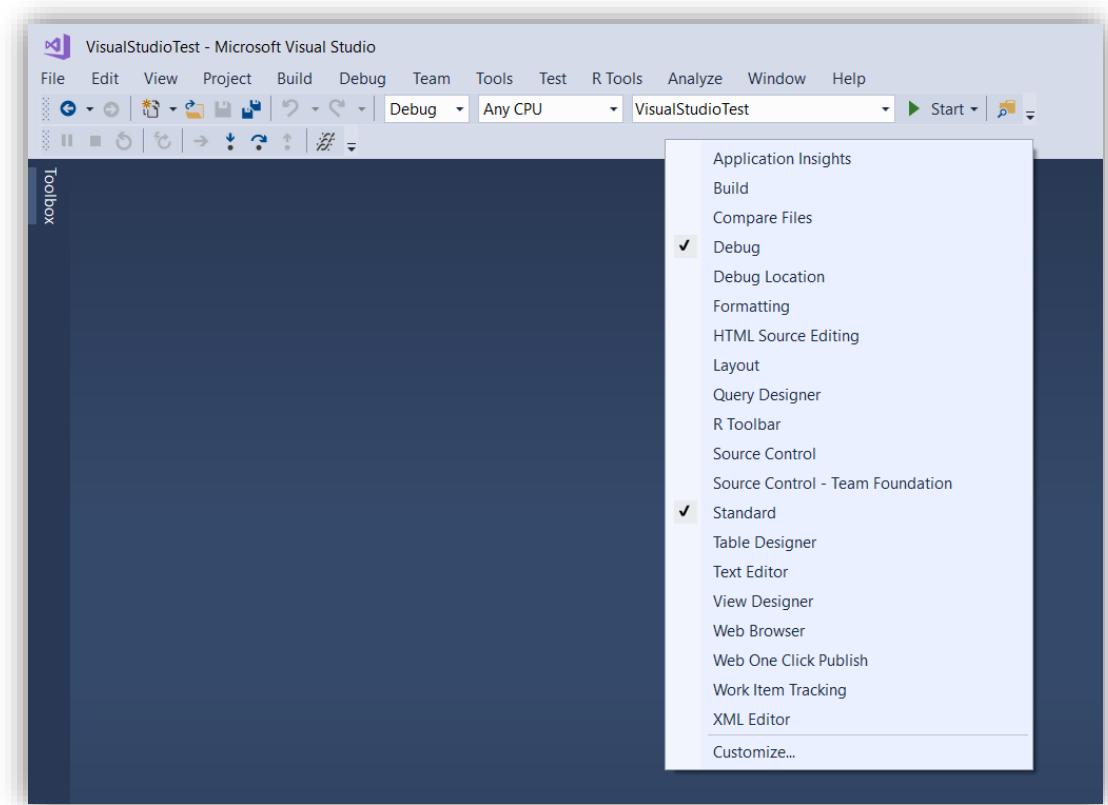
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace UniversityLibrary
8  {
9      public class Person
10     {
11         public string FirstName { get; set; }
12         public string FullName { get; set; }
13         public int Id { get; set; } // Selected in tooltip
14         public string LastName { get; set; }
15
16         public Person(int id, string firstName, string lastName)
17         {
18             Id = id;
19             FirstName = firstName;
20             LastName = lastName;
21         }
22
23         public void UpdateName(string newLastName)
24         {
25             LastName = newLastName;
26         }
27     }
28 }
```

TOOLBARS



TOOLBARS

- Add/Remove default toolbars
- Customize toolbars – show/hide buttons



SCROLLBARS

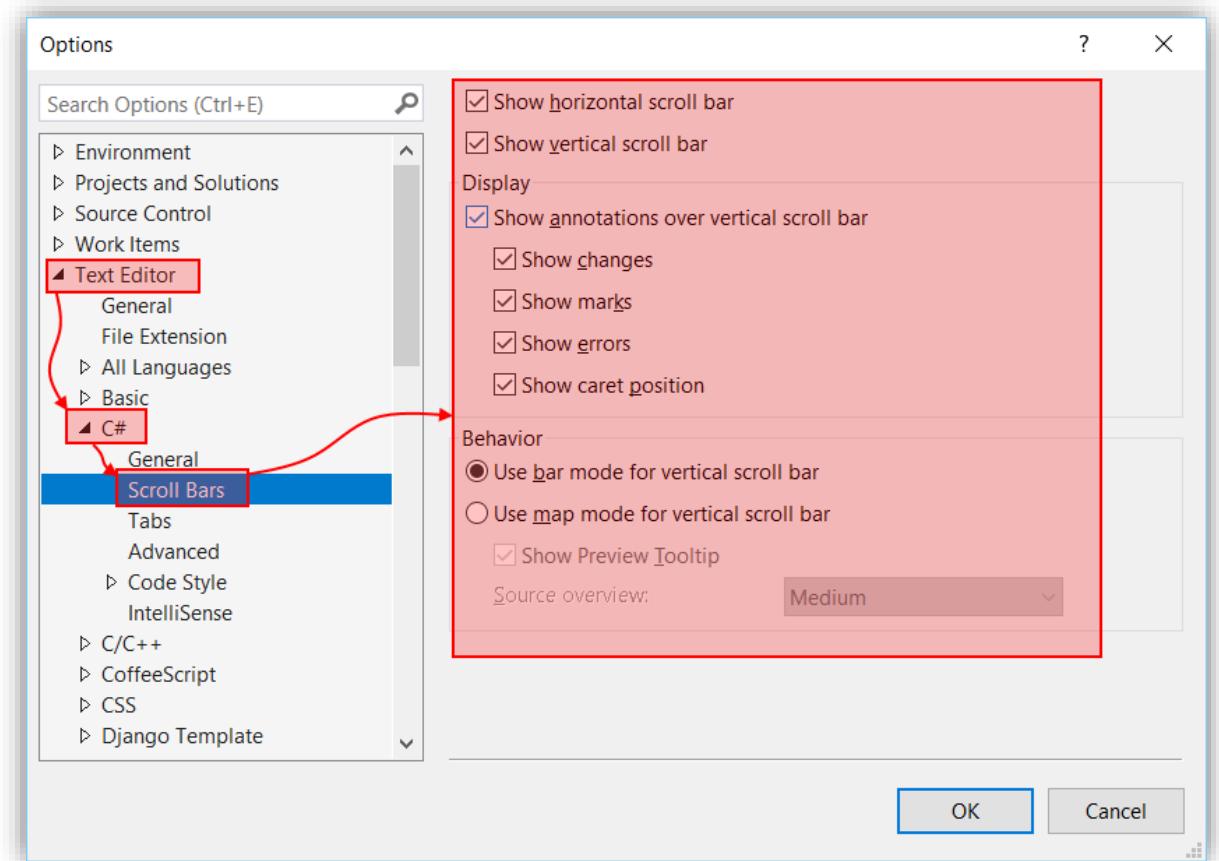
SCROLL BAR

➤ Enable/disable

- Tools → Options → Text Editor → C# → Scroll Bars

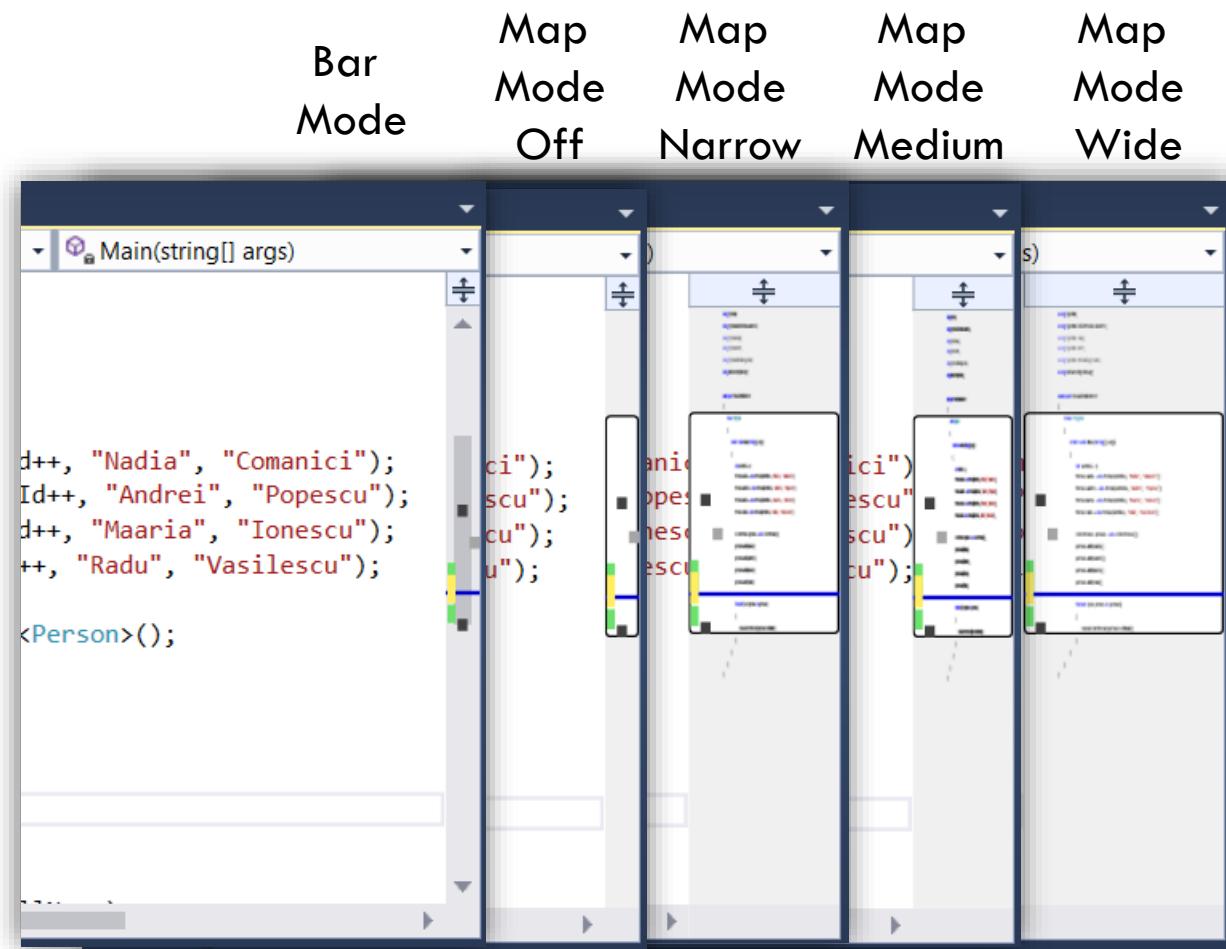
➤ Can be:

- Bar Mode
- Map Mode
 - Wide or narrow
 - Colored Columns for info



SCROLLBAR MODES

- Blue line = current line
- Grey scroll = part of the file in view
- Has 3 columns:
 - Column 1:
 - Yellow = Unsaved changes
 - Green = Saved changes
 - Column 2:
 - Orange = Find Results
 - Dark Red = Breakpoints
 - Black = Bookmarks
 - Column 3:
 - Light Red = errors
 - Light Green = warnings



BOOKMARKS AND TASK LIST

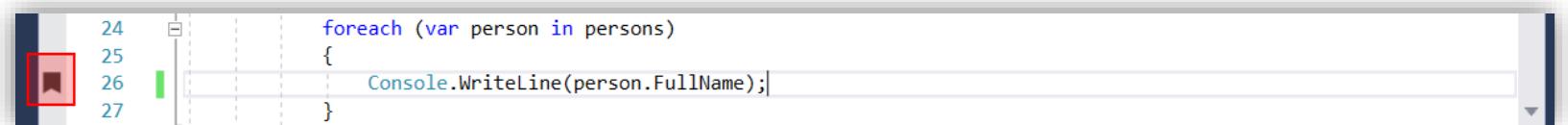
BOOKMARKS

➤ Add a bookmark:

- Ctrl + K + K

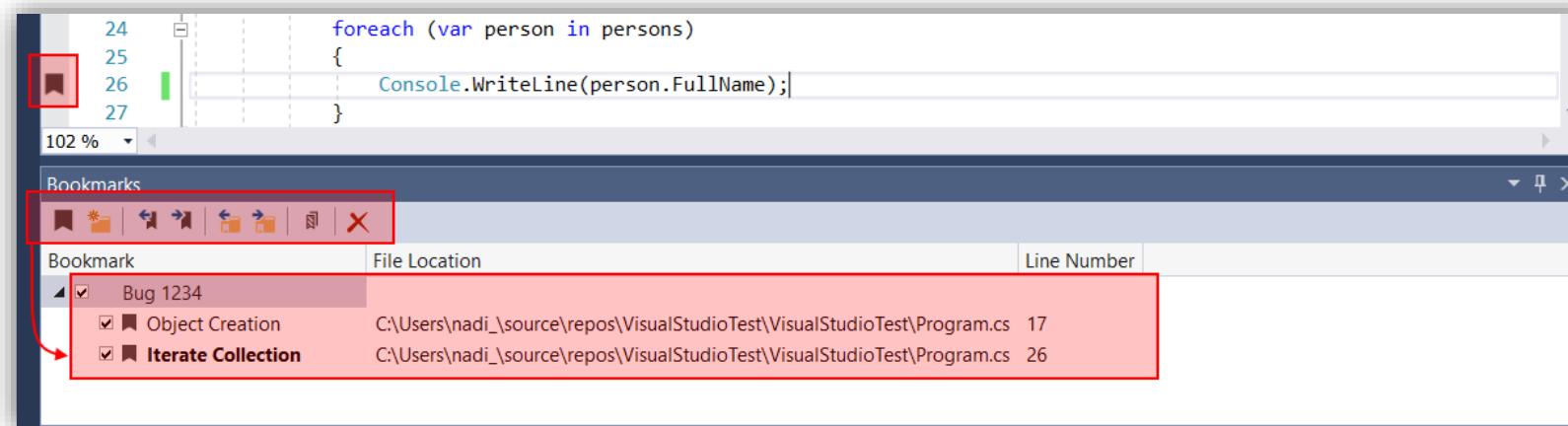
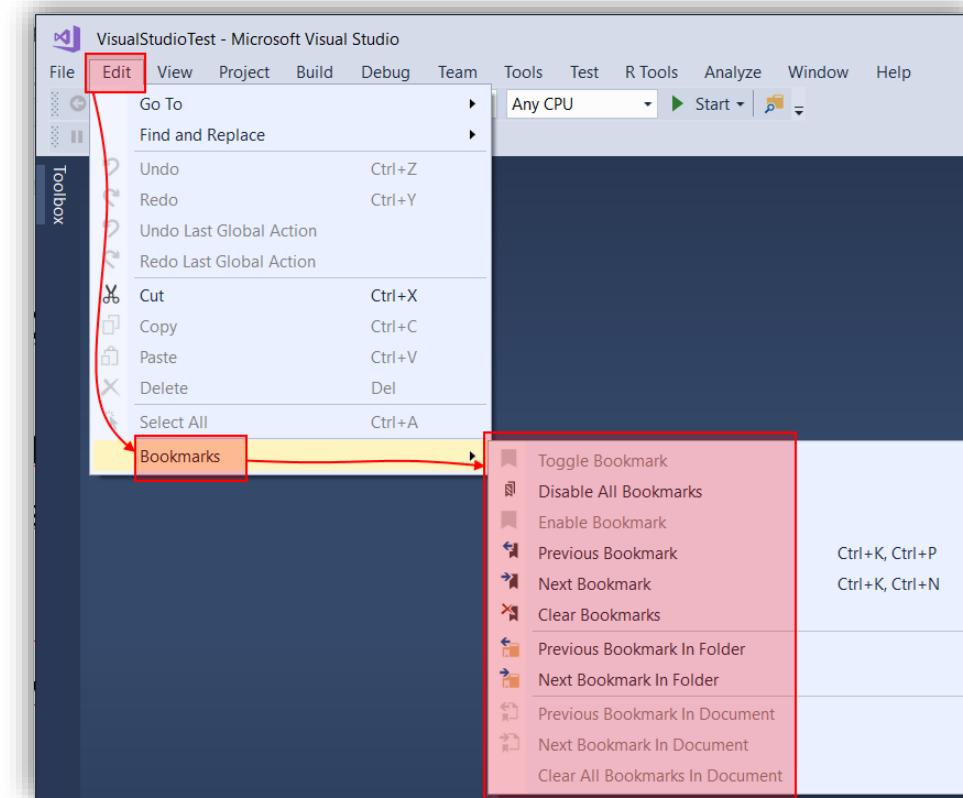
➤ Stored in .suo file

- Not transferable to others



BOOKMARKS WINDOW

- Show Bookmarks Window
 - View → Other Windows → Bookmarks Window
- Define folders
- Give names to bookmarks



TASK LIST

- Show Bookmarks Window
 - View → Task List
- Can specify messages inside a comment and view these messages later in a specific window
- Use keywords:
 - TODO
 - HACK
 - UNDONE
- Filtering:
 - Search by string
 - Entire Solution
 - Current Project
 - Current Document
 - Opened Documents

The screenshot shows a code editor window with C# code and a Task List window below it.

```
50
37
38
39    catch (Exception ex)
{
    // TODO: display a more specific message
    MessageBox.Show(ex.Message, "An error has occurred", MessageBoxButton.OK, MessageBoxImage.Error);
}
41
42
43 }
```

The Task List window is titled "Task List" and shows the following entry:

Description	Project	File	Line
TODO: display a more specific message	IntelliTraceSample	MainWindow.xaml.cs	39

At the bottom of the Task List window, there are links: Error List ..., Call Hierarchy, Bookmarks, and Output.

TEXT OPERATIONS

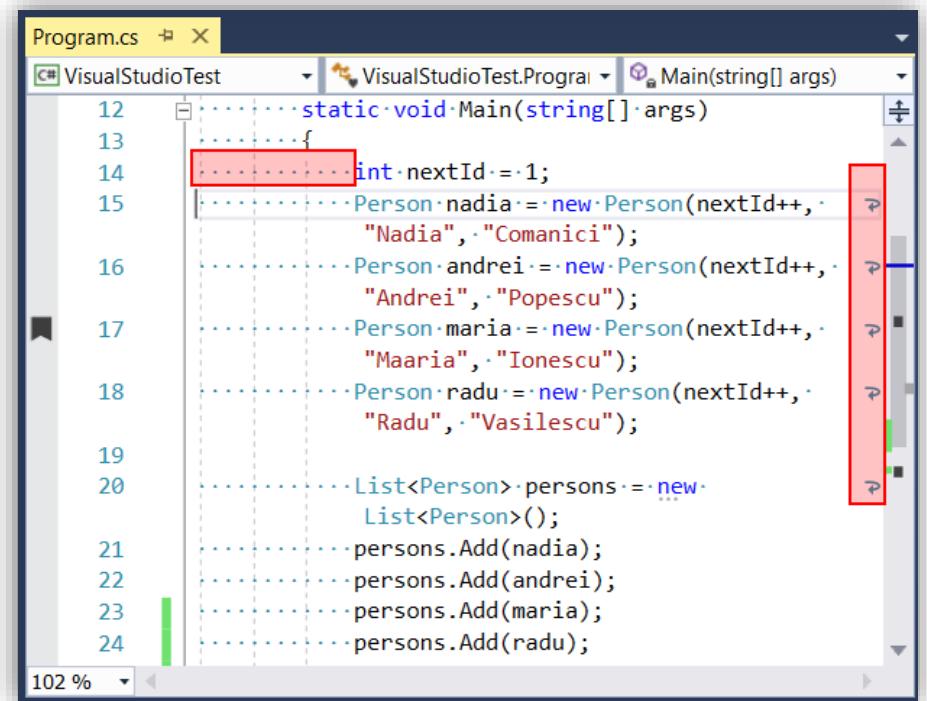
TEXT DISPLAY OPTIONS

➤ Word Wrap

- Enable in Edit → Advanced → Word Wrap
- Ctrl + E + W

➤ Show White Space

- Enable in Edit → Advanced → View White Space
- Ctrl + R + W

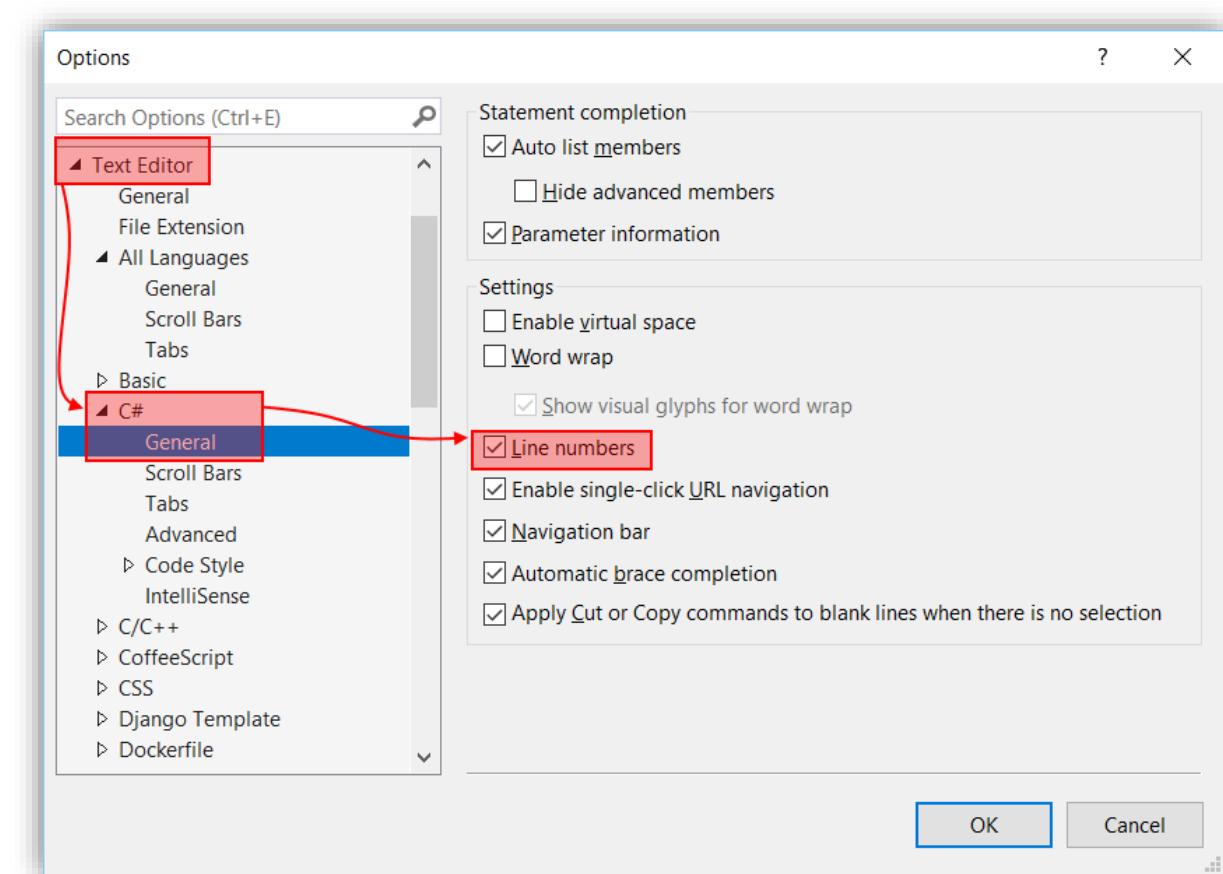


A screenshot of the Visual Studio code editor showing a C# file named Program.cs. The code defines a static void Main method that creates four Person objects: nadia, andrei, maria, and radu, and adds them to a list of persons. The code is displayed with word wrap enabled, as indicated by the red boxes around the lines of code. The white space around the code is also highlighted with red boxes, demonstrating the 'View White Space' feature.

```
12     static void Main(string[] args)
13     {
14         int nextId = 1;
15         Person nadia = new Person(nextId++, "Nadia", "Comanici");
16         Person andrei = new Person(nextId++, "Andrei", "Popescu");
17         Person maria = new Person(nextId++, "Maaria", "Ionescu");
18         Person radu = new Person(nextId++, "Radu", "Vasilescu");
19
20         List<Person> persons = new List<Person>();
21         persons.Add(nadia);
22         persons.Add(andrei);
23         persons.Add(maria);
24         persons.Add(radu);
```

SHOW LINE NUMBERS

➤ Enable in Text Editor → C# → General



CURSOR MOVEMENT

Shortcut	Action
Ctrl + ↓	Scroll Down
Ctrl + ↑	Scroll Up
Ctrl + ←	Move to the begining of next word
Ctrl + →	Move to the end of this word
Ctrl + PageUp	Move to the top of the current display in window
Ctrl + PageDown	Move to the bottom of the current display in window

TEXT SELECTION

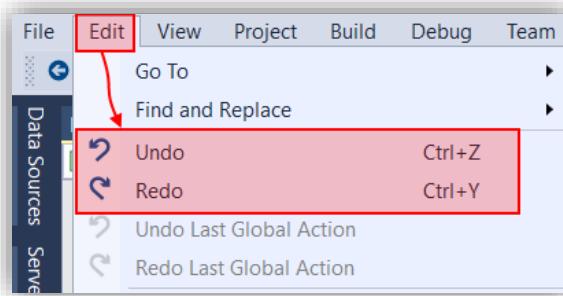
- Selection means having:
 - A start position (anchor)
 - An final active position
- The changes apply to the final active position
- Try:
 - Shift + ←
 - Shift + ↑
 - Ctrl + Shift + →
 - Ctrl + Shift + PageDown

TEXT SELECTION

Shortcut	Action
Ctrl + K + A	Switch anchor
Shift + left/right arrow	Select character
Ctrl + Shift + left/right arrow	Select word
Shift + PageUp/PageDown	Select up or down, on the same column
Place caret at { and then Ctrl + Shift +]	Select entire block of code

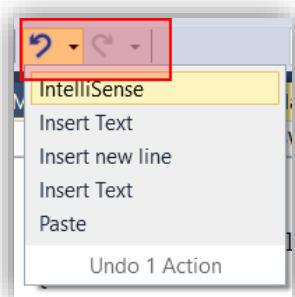
UNDO/REDO BUFFERS

- Meniul Edit → Undo/Redo:



Shortcut	Action
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + Shift + Z	

- Toolbar:



- Undo/Redo stacks are kept for each file and will be deleted once the file is closed

KEYBOARD SHORTCUTS

Shortcut	Action
Ctrl + C	Copy
Ctrl + X (with selection)	Cut
Ctrl + X (no selection)	Delete entire line
Ctrl + L	Delete entire line(s)
Ctrl + V	Paste
Ctrl + Shift + V	Cycle clipboard

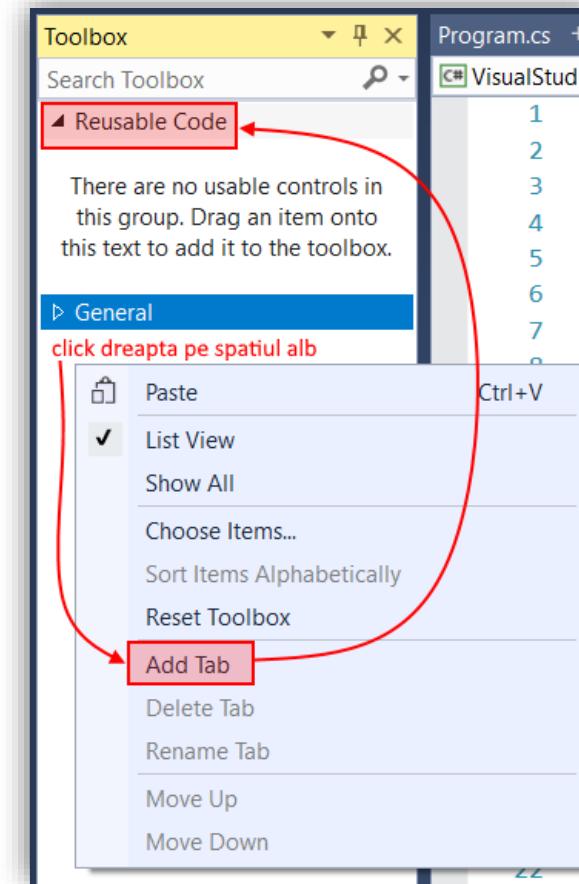
Shortcut	Action
Shift + Alt + T	Transpose line
Alt + Up	
Alt + Down	
Ctrl + T	Transpose letters

Shortcut	Action
Ctrl + U	Lower case letter
Ctrl + Shift + U	Upper case letter

Shortcut	Action
Ctrl + E + D	Format Document
Ctrl + K + D	
Ctrl + E + F	Format Selection

REUSING CODE - TOOLBOX

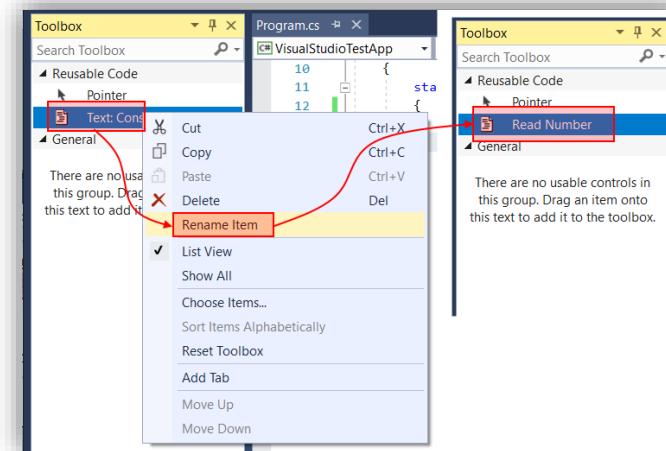
- **General tab**
 - Default in Toolbox
- **Add tab**
 - Right Click on Toolbox
 - Add Tab
- **Rename Tab**
 - Right Click on Tab
 - Rename Tab
- **Delete Tab**
 - Right Click on the Tab
 - Delete Tab



REUSING CODE - TOOLBOX

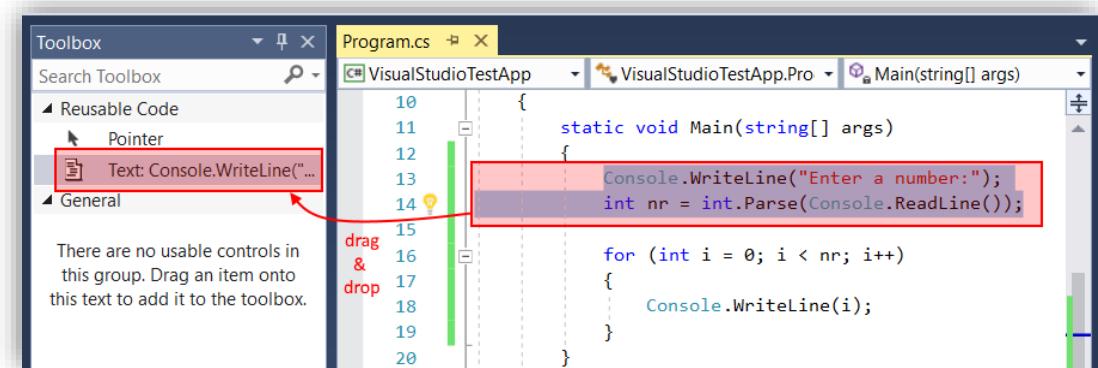
➤ Add code

- Select code
- Drag & Drop in Toolbox, under the desired tab



➤ Rename code in toolbox

- Right click on the item in toolbox
- Rename Item



➤ Other:

- Reorder tabs and items inside a tab
- Moving an item from a tab to another

COLUMN SELECTION

- Alt + Shift + Click
- Alt + Shift + Up/Down Arrow

```
int nextId = 1;
Person nadia = new Person(nextId++, "Nadia", "Comanici");
Person andrei = new Person(nextId++, "Andrei", "Popescu");
Person maria = new Person(nextId++, "Maaria", "Ionescu");
Person radu = new Person(nextId++, "Radu", "Vasilescu");
```

OUTLINE

Shortcut	Action
Ctrl + M + O	Collapse to definition (in a file)
Ctrl + M + L	Expand (in file)
Ctrl + M + H	Hide selected lines of code (without region)
Ctrl + M + U	Remove hide
Edit → Outlining Right Click → Outlining	Show outline menu

REGIONS

- Start with `#region NameOfRegion`
- Ends with `#endregion`
- Can be collapsed/expanded

The diagram illustrates the state of a code editor showing a class definition with a region:

```
public class Person
{
    #region Properties
    public int Id { get; private set; }
    public string FirstName { get; private set; }
    public string LastName { get; private set; }
    #endregion
}
```

The region from line 11 to line 15 is highlighted with a red box. A red arrow points from the line 11 marker to the start of the region, and another red arrow points from the end of the region back to the line 15 marker.

Below this, a second code block shows the same class definition, but the region has been collapsed:

```
public class Person
{
    [+] Properties
}
```

The word "Properties" is enclosed in a red box, indicating it is the collapsed state of the region. A red arrow points from the line 11 marker to the start of the collapsed region.

FIND AND REPLACE

FIND (CTRL + F)

- Match Case
- Match Whole Word
- Use Regular Expressions
- Search in:
 - Current Block
 - Selection
 - Current Document
 - All Opened Documents
 - Current Project
 - Entire Solution

A screenshot of the Visual Studio IDE showing the 'Program.cs' file. The code defines a 'Person' class and a 'Main' method. The word 'Person' is highlighted in orange across multiple occurrences in the code. The status bar at the bottom shows '102 %'. The search interface in the top right corner shows the search term 'person' and options for 'Current Document'.

```
9
10 class Program
11 {
12     static void Main(string[] args)
13     {
14         int nextId = 1;
15         Person nadia = new Person(nextId++, "Nadia", "Comanici");
16         Person andrei = new Person(nextId++, "Andrei", "Popescu");
17         Person maria = new Person(nextId++, "Maaria", "Ionescu");
18         Person radu = new Person(nextId++, "Radu", "Vasilescu");
19
20         List<Person> persons = new List<Person>();
21         persons.Add(nadia);
22         persons.Add(andrei);
23         persons.Add(maria);
24         persons.Add(radu);
25
26         foreach (var person in persons)
```

Shortcut	Action
F3	Next Result
Shift + F3	Previous Result
Select string and Ctrl + F3	Open find with selected string

FIND IN FILES (CTRL + SHIFT + F)

➤ Look in:

- Current Block
- Current Document
- All Opened Documents
- Entire Solution
- Current Project
- Choose search folders

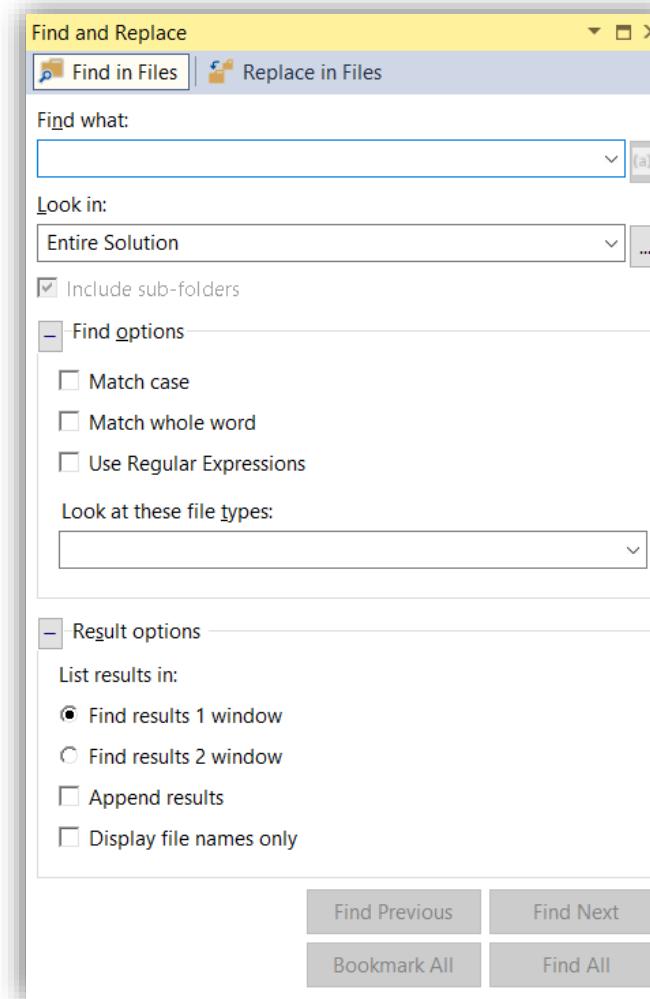
➤ Find Options

- Match case
- Match whole word
- Use Regular Expressions

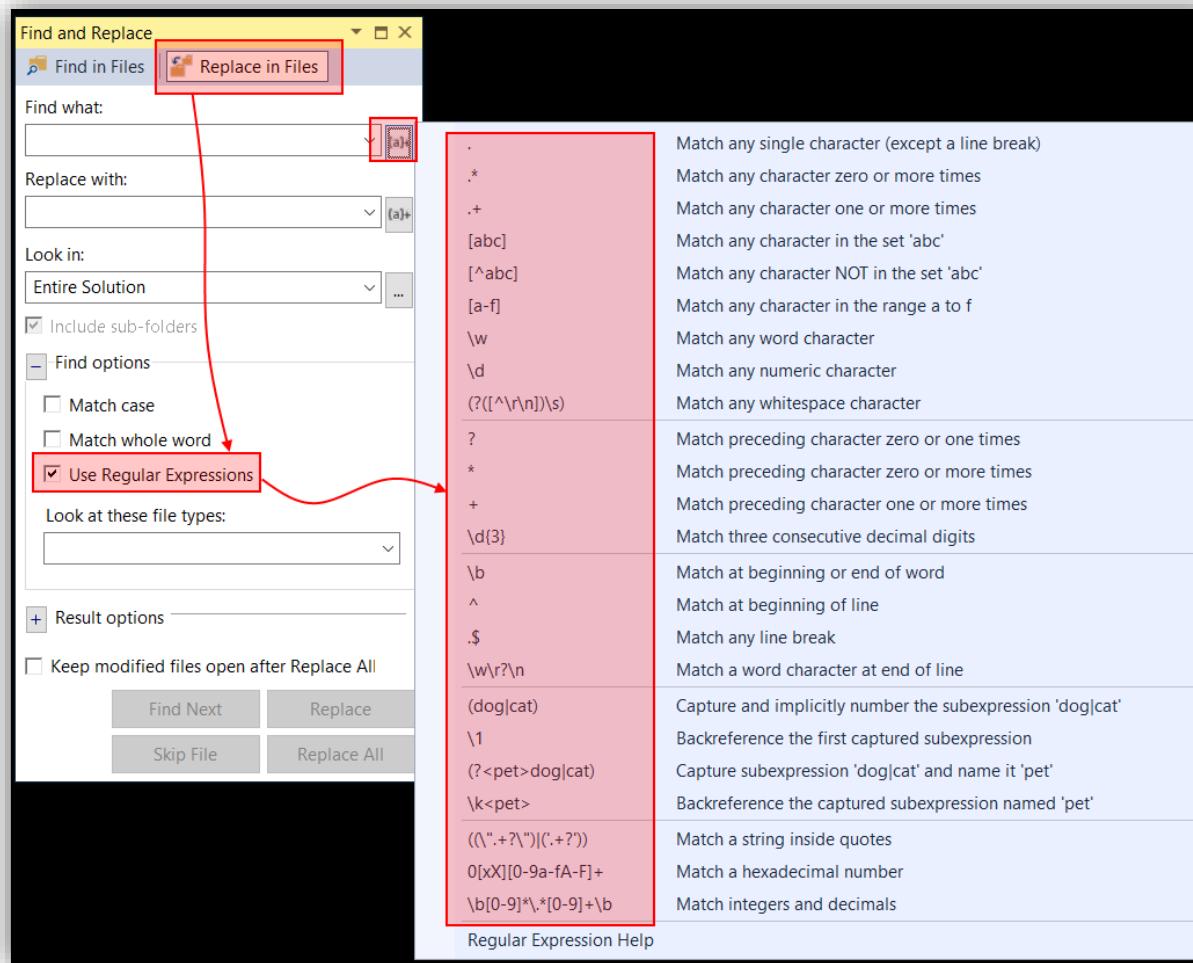
➤ Look at these file types – extensions

➤ Results Options

- Display in Results 1 Window or Results 2 Window
- Append Results
- Display file names only

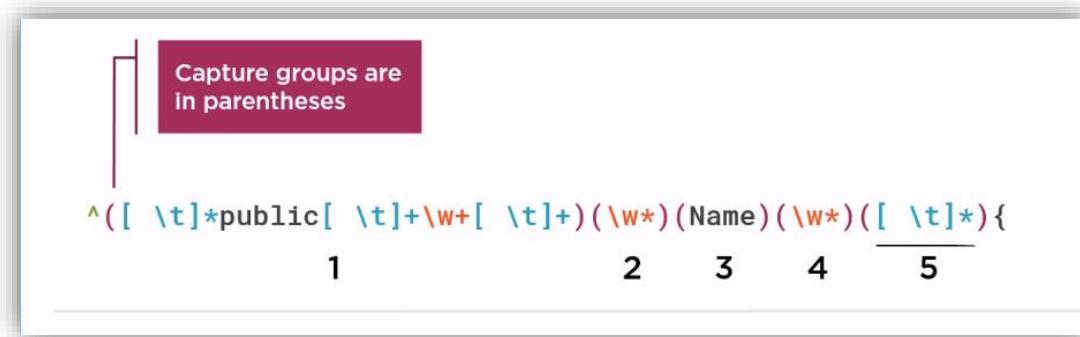


REGEX SEARCH AND REPLACE



REGEX CAPTURE GROUP

- Capture group
 - A set of characters that are treated together in a regex replacement
 - The group is specified between ()
 - Usage in replace: \$1, \$2
- First use „Find” before calling „Replace”



DEBUGGING

ERROR LIST

➤ Ctrl + Shift + F12 = Navigate between errors

➤ Filter:

- By solution/opened documents/current project
- By errors/warnings/notifications
- By Build/Intellisense

➤ Search by keyword

➤ Filter by columns

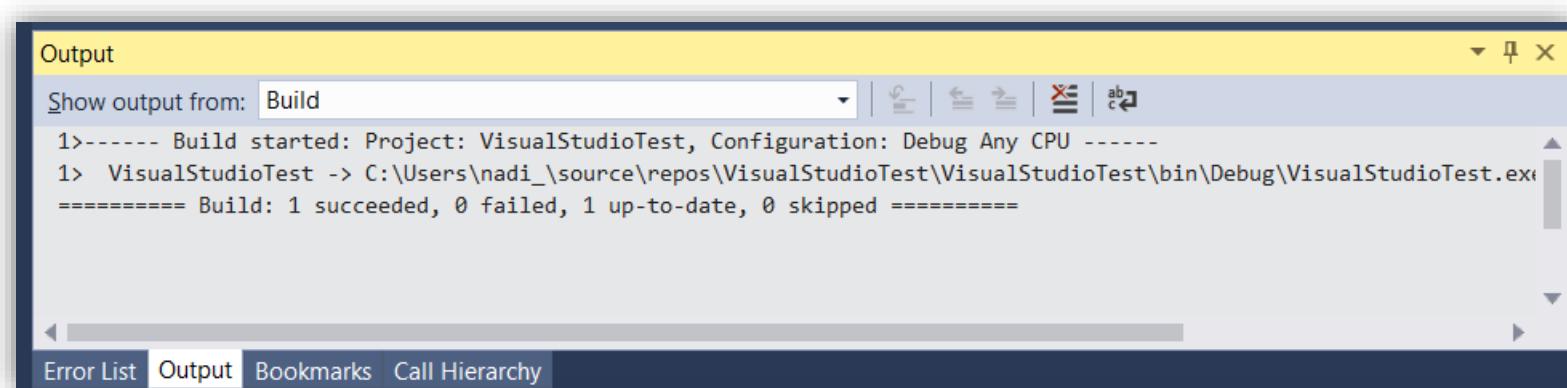
➤ Right Click header

- Show Columns
- Grouping

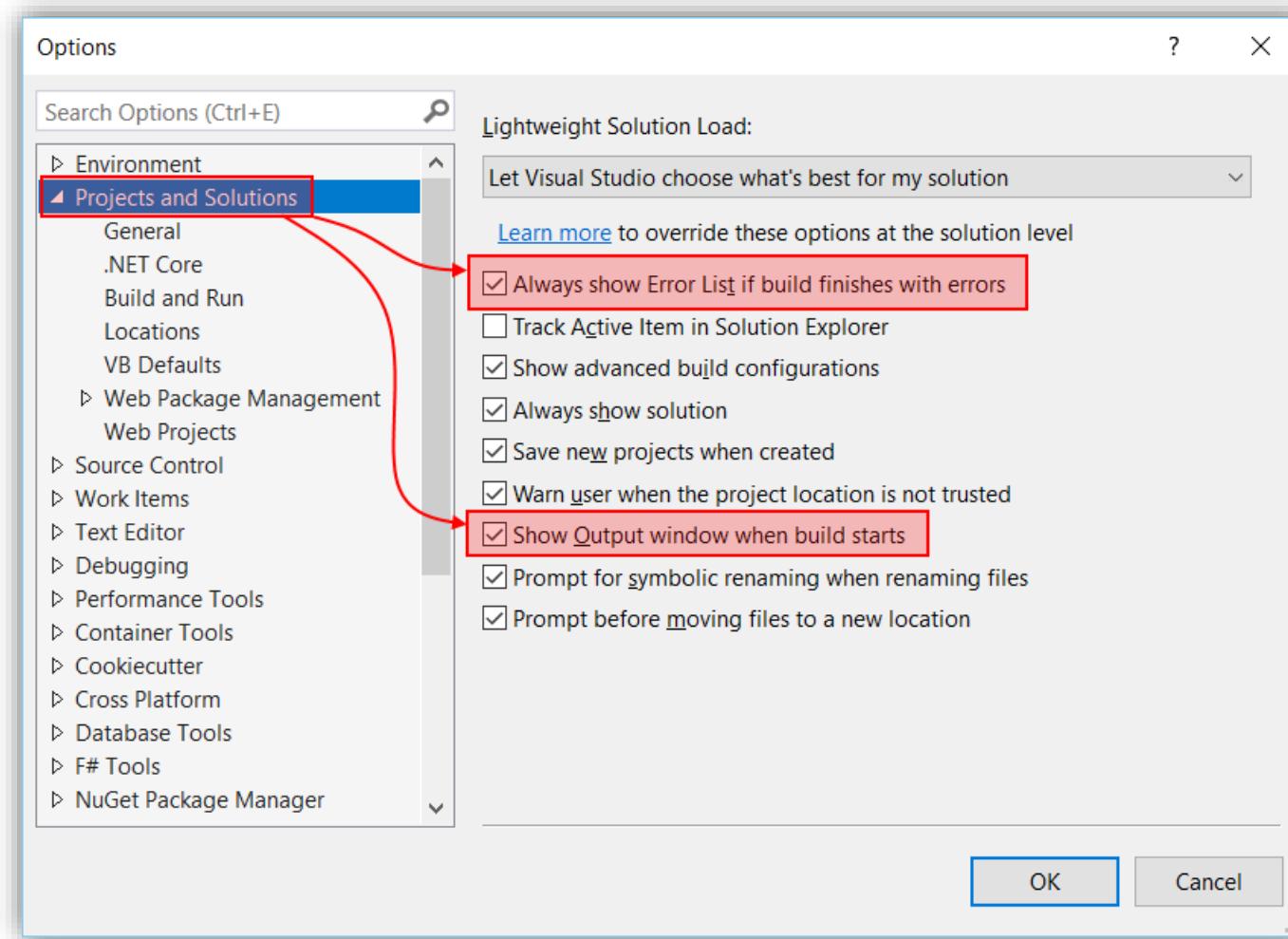
Error List						
	Code	Description	Project	File	Line	Suppression St...
▲	CS0103 (1)	CS0103 The name 'person' does not exist in the current context	VisualStudioTest	Program.cs	26	Active
▲	CS0246 (2)	CS0246 The type or namespace name 'Perso' could not be found (are you missing a using directive or an assembly reference?)	VisualStudioTest	Program.cs	15	Active
▲	CS0246	CS0246 The type or namespace name 'Perso' could not be found (are you missing a using directive or an assembly reference?)	VisualStudioTest	Program.cs	18	Active
▲	IDE0028 (1)	IDE0028 Collection initialization can be simplified	VisualStudioTest	Program.cs	20	Active

OUTPUT WINDOW

- Stream of operations that are taking place
 - Build
 - Build order
 - Other
- Can search inside it = Ctrl + F
- Clickable



ERROR AND OUTPUT WINDOW

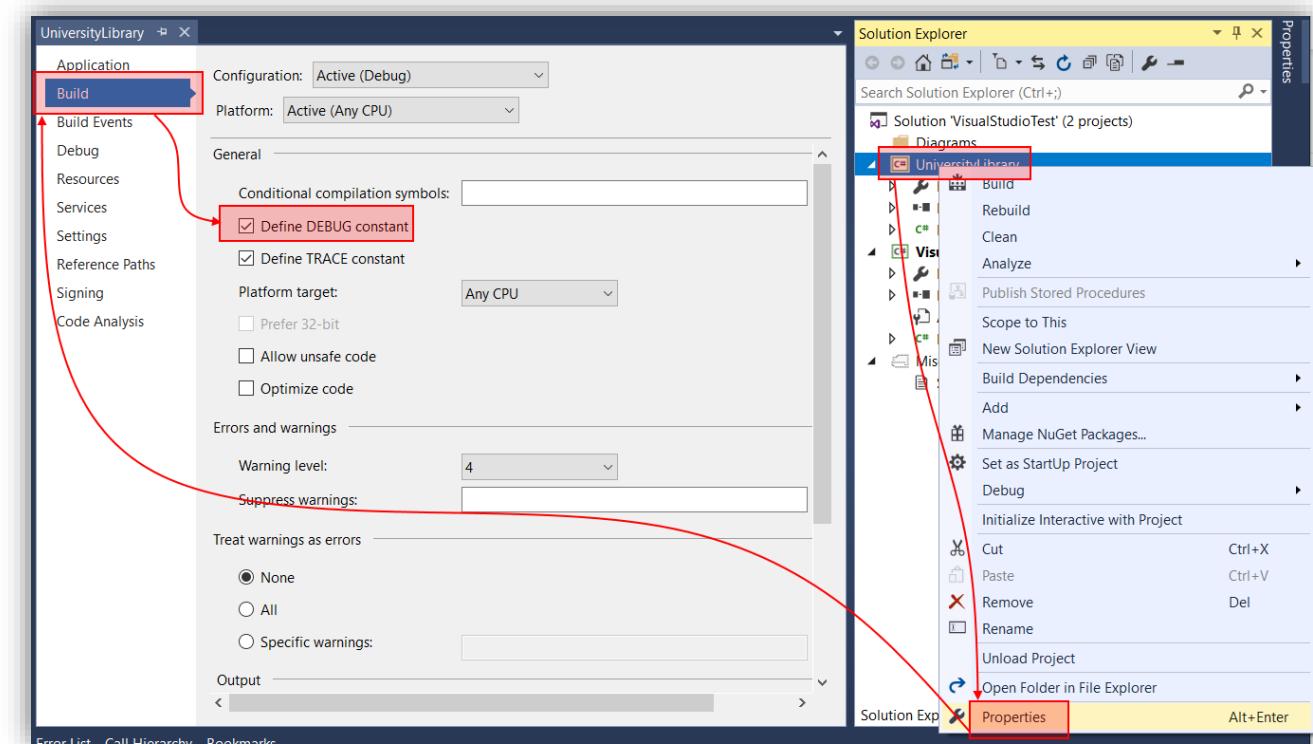


DEBUG CONSTANT AND DEBUG CLASS

- Debug class can be used in code, but only when the DEBUG constant is present
 - Debug constant can be enabled/disabled in project properties

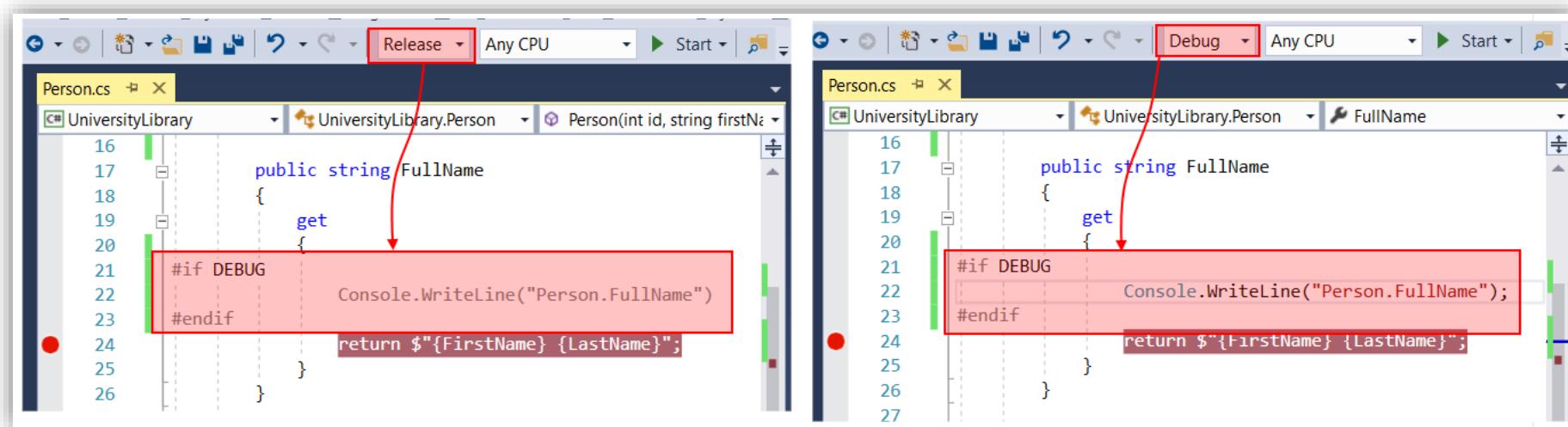
➤ Alternatives:

- Trace – old and inefficient
- Microsoft Event Tracing (ETW)

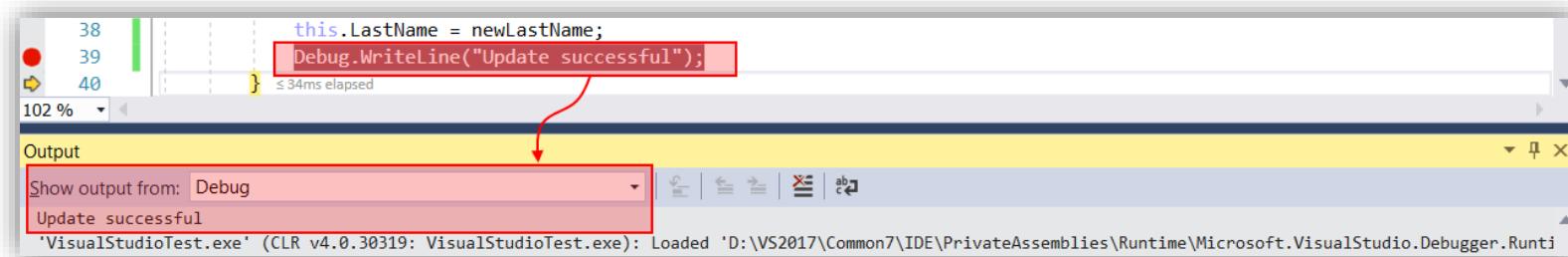


DEBUG CONSTANT AND DEBUG CLASS

➤ Usage of debug constant

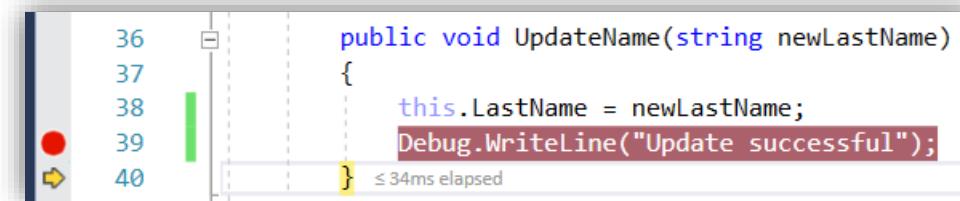


➤ Usage of Debug class → writes messages in Output Window



SETTING BREAKPOINTS & DEBUGGING

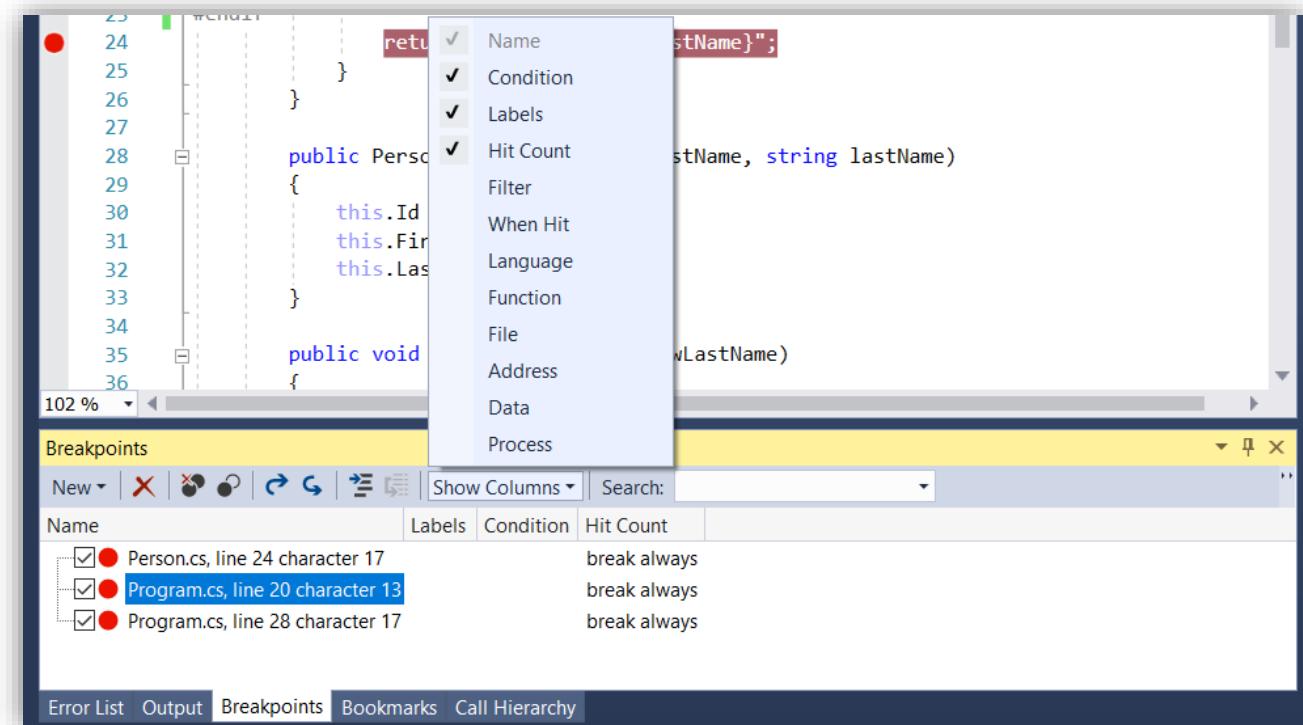
1. Click in the left margin
2. F9 or right click menu
3. Right click in the callstack
4. Function breakpoint
 - Debug menu -> New breakpoint
 - Breakpoint dialog
5. Run to cursor



Shortcut	Action
F9	Toggle breakpoint
F10	Step over the current line of code
F11	Step Into
Shift + F11	Step out of current method
Ctrl + Shift + F10	Set Next Statement Or grab the yellow arrow and moving it to the next statement you want

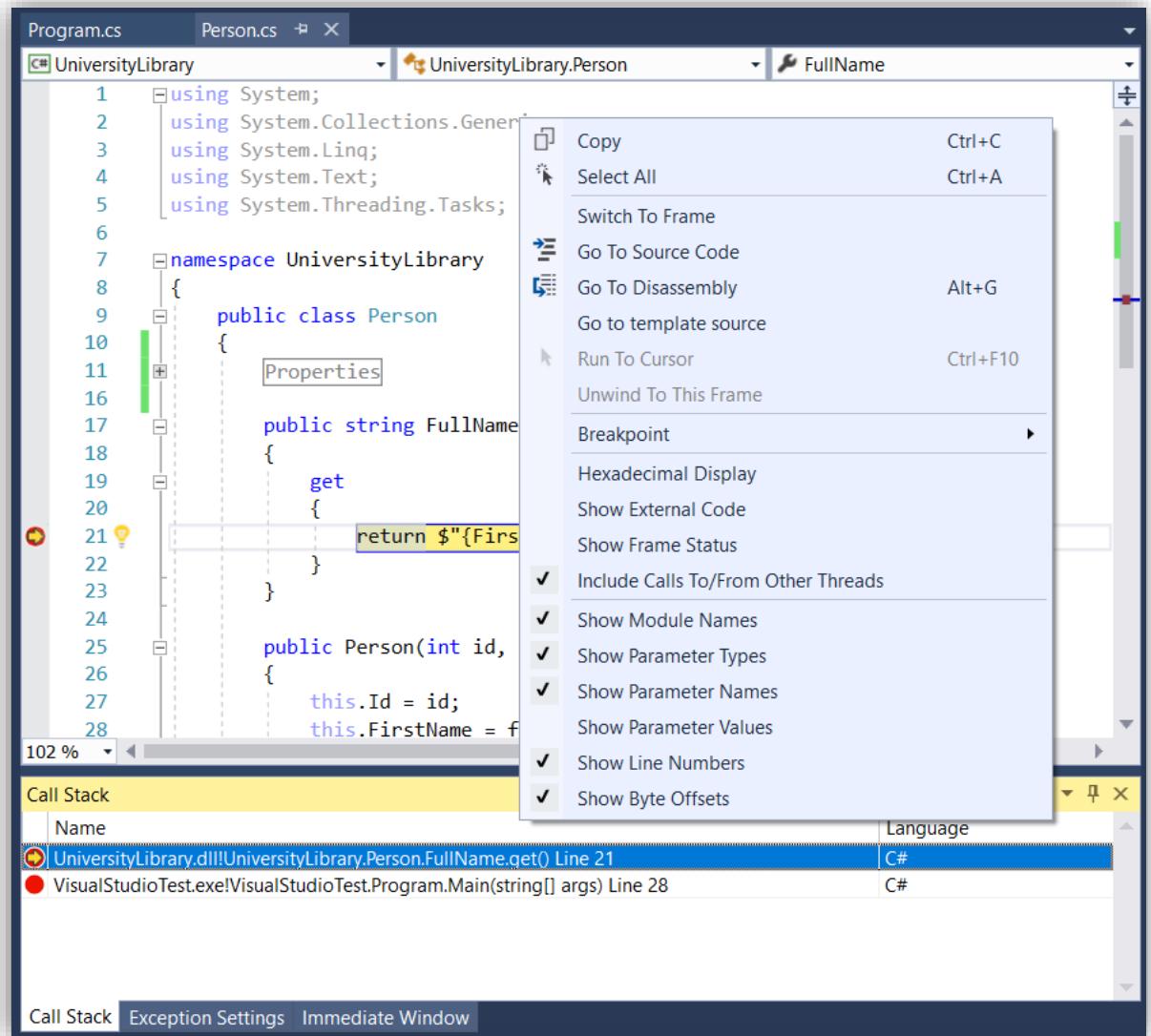
BREAKPOINTS DIALOG

- Search – filter what is displayed
- Delete or disable breakpoints
- Customize Displayed Columns
- Add labels
 - For example – breakpoints for Bug X
- Export breakpoints
 - The found breakpoints to a different person, to continue
 - XML format
- Search by keyword



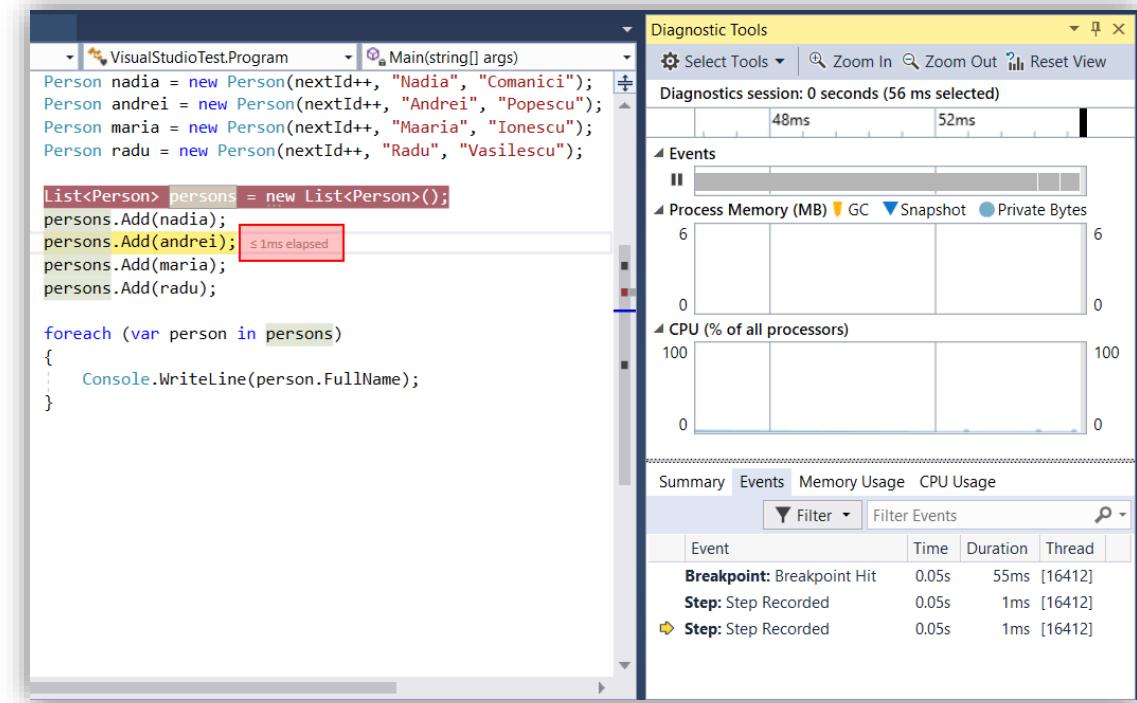
CALL STACK

- Right click → options which column to display
- Can navigate to each step in the call stack and evaluate the value of members



DIAGNOSTIC TOOLS

- Tools → Options → Debugging → General
 - Enable Diagnostic Tools while debugging
- Can see the history of debugging and go to the code
 - Previous exception
 - Duration for each step
- PerfTips
 - Shows the duration of executing that line
- Take snapshot
- See usage for
 - Process memory
 - CPU



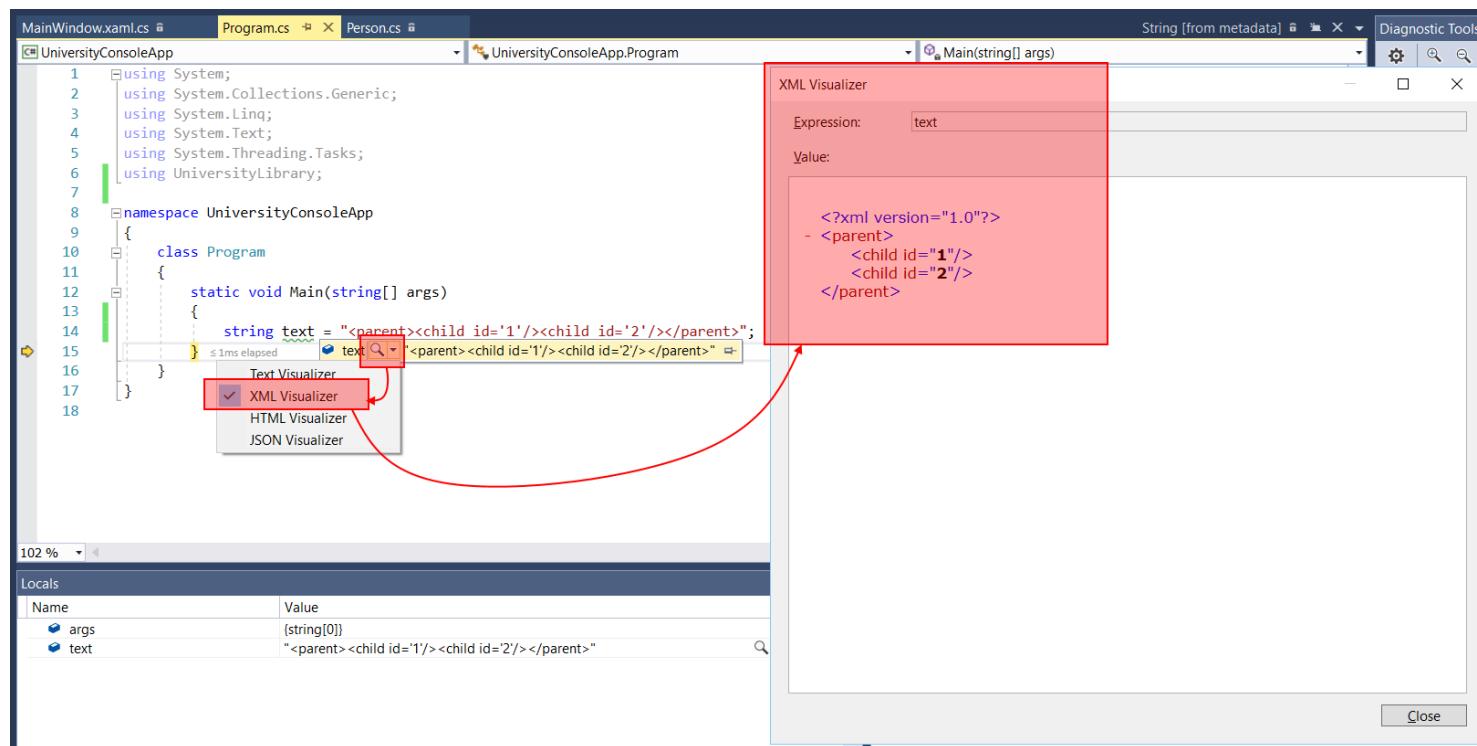
LOCALS, AUTOS & WATCH

Window	Description
Locals	All the variables in the current function
Autos	All the variables involved in: <ul style="list-style-type: none">- The current line- The previous executed line- The next line that will be executed
Watch	There are 4 watch windows, in which you can add variables
Quick Watch	A popup that allows to evaluate a variable or its contents

- Each variable can be edited in these windows
- A modified variable will have its value colored in red

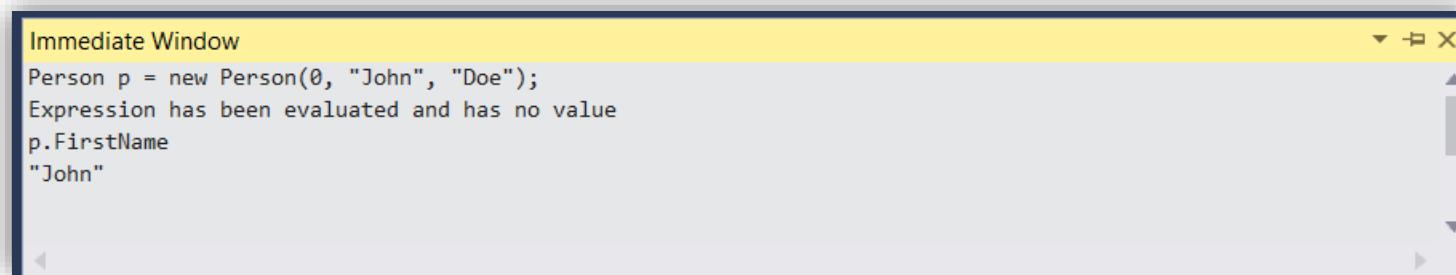
VISUALIZERS

- You can view a certain object (such as a string) in different formats
- There are default visualizers
- You can create custom visualizers



IMMEDIATE WINDOW

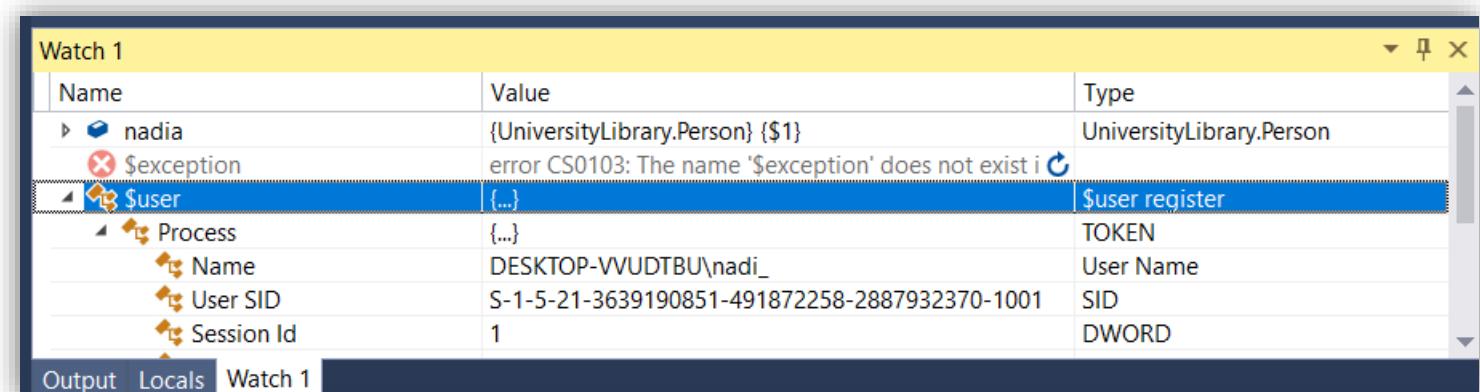
You can write and execute instructions, as you would write normal code



The screenshot shows the Immediate Window of a debugger. The window title is "Immediate Window". The content displays the following text:
Person p = new Person(0, "John", "Doe");
Expression has been evaluated and has no value
p.FirstName
"John"

OBJECT ID

- Lets you access data when the variable pointing to it is not in scope
- Right click variable in watch window → make object Id → \$1
- You don't have intellisense, but you can write properties and they will be evaluated
 - \$1.FirstName
- Can add these object Ids in other debugging windows or conditional breakpoints
- Special objects
 - \$exception
 - \$user



EXCEPTION WINDOW

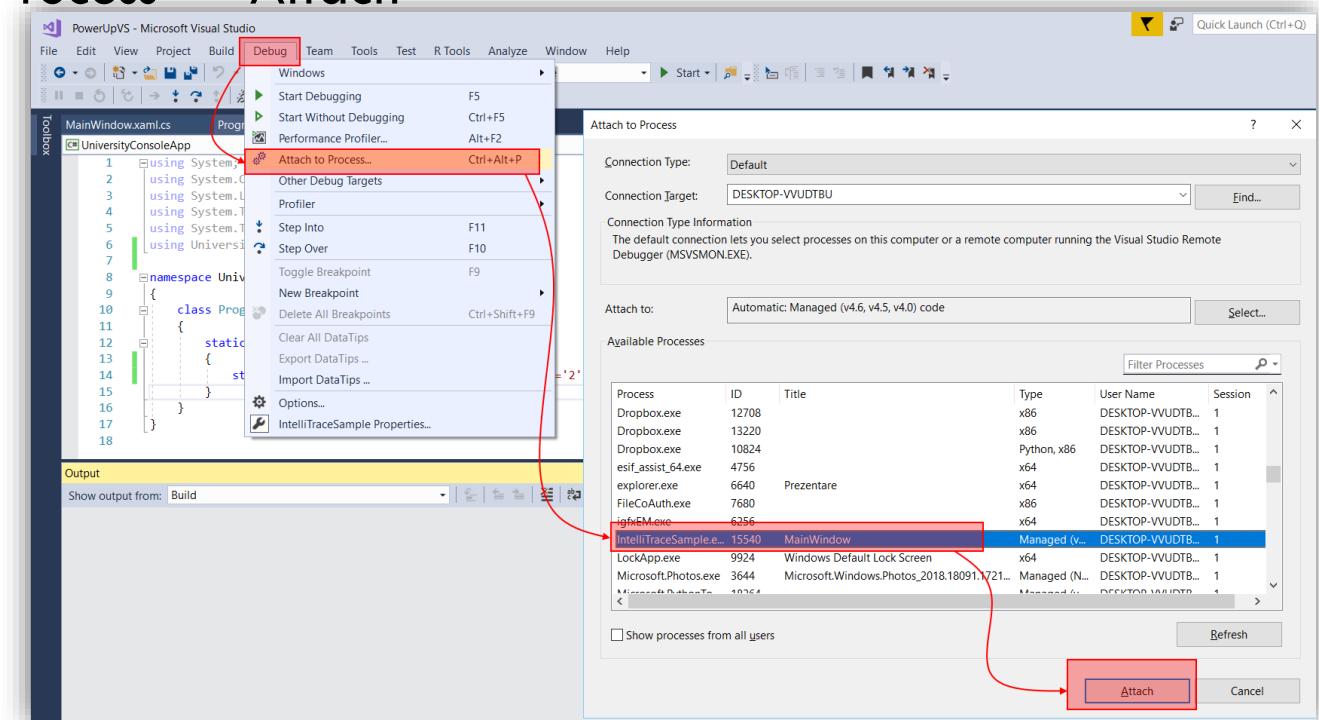
- \$exception = pseudo variable for last thrown exception
- Exception settings
 - Remove unnecessary groups
 - Can reset to default to get the others back
- If checked, debugger breaks on the line where the exception occurred
 - Very useful when you know an exception is thrown, but don't know where it comes from
- Continue when unhandled in user code
 - Will not stop if that code does not belong to the user, but is in the callstack

CONDITIONAL BREAKPOINTS

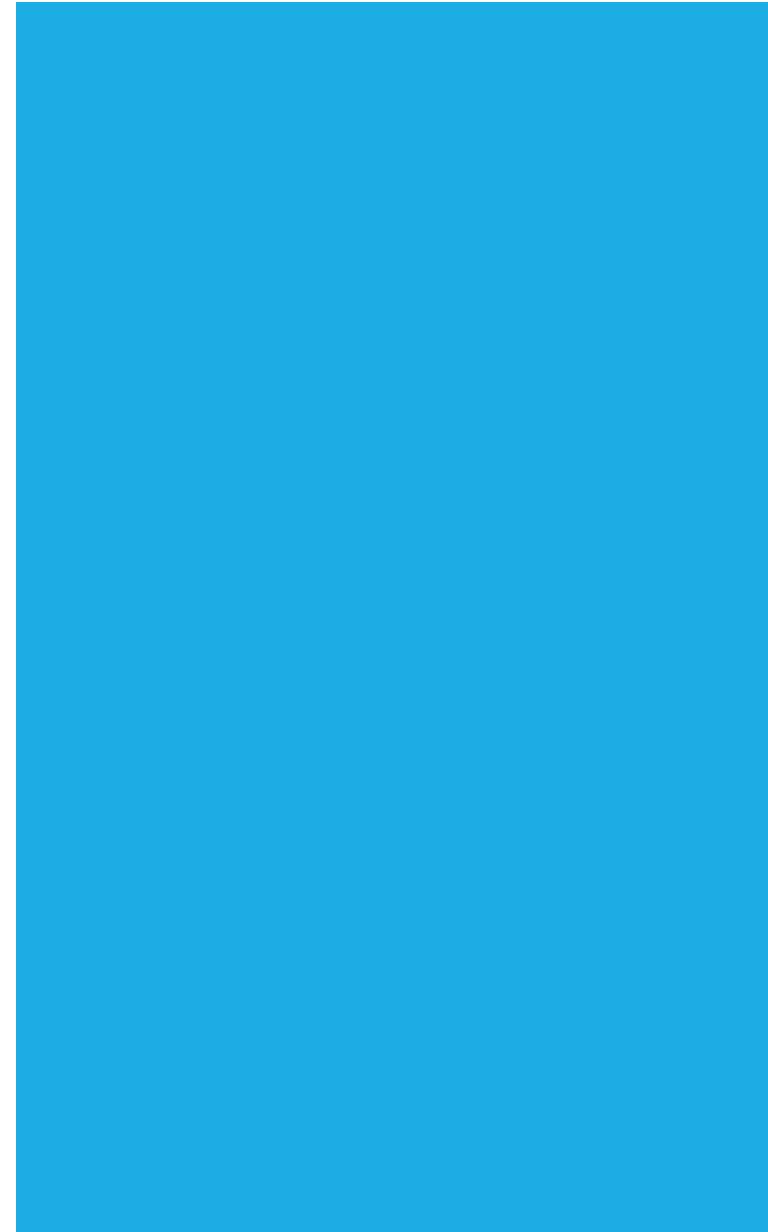
- Included in undo/redo buffer
- You can use objectIDs or a variable created in the immediate window
- Conditions:
 - Conditional Expression
 - Hit Count
 - Put 9999999 to see exactly which one (Breakpoints Window -> Hit count column)
 - Filters
 - When stopping on a single thread
- Multiple conditions, but only one of each type
- Action:
 - Logging to the output window
 - Uncheck „continue execution” to break

ATTACH TO PROCESS

- If you are already started the application without debug, you can still attach visual studio to it and debug without restarting the application again
- Debug → Attach to Process → Select Process → Attach



BEYOND COMMUNITY EDITION



CODELENS

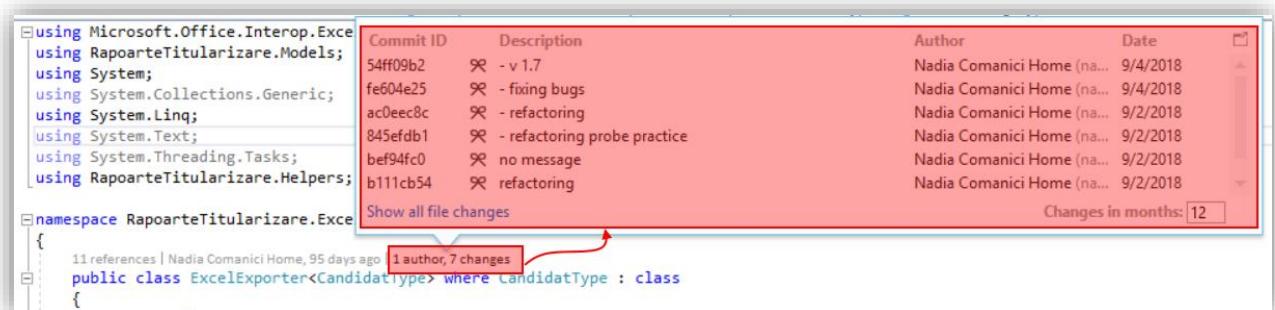
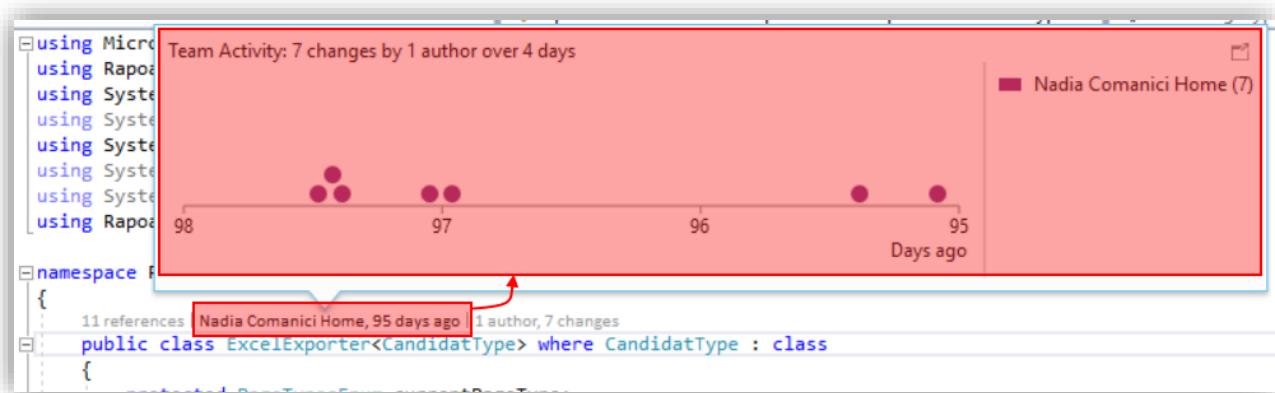
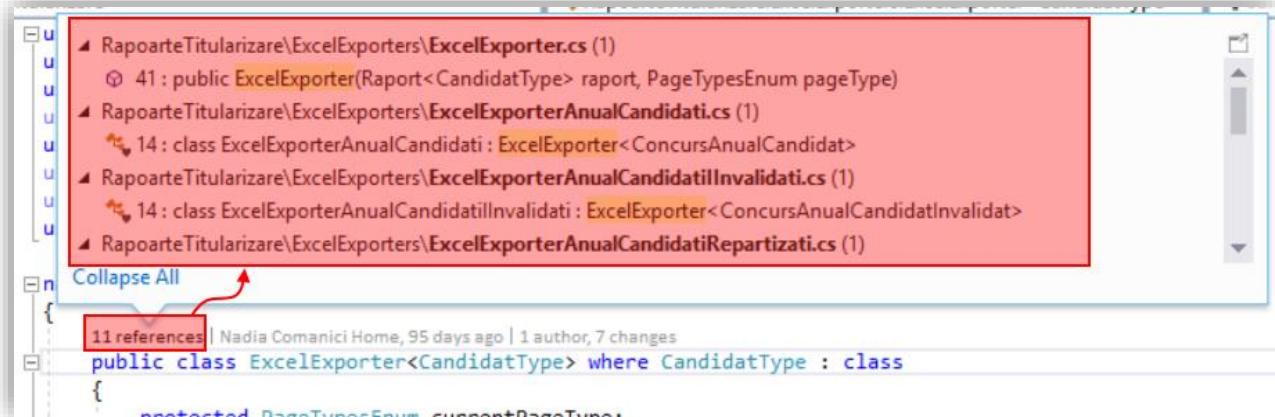
➤ Enable/disable:

- Text Editor → All Languages → Code Lens

➤ Code above the method

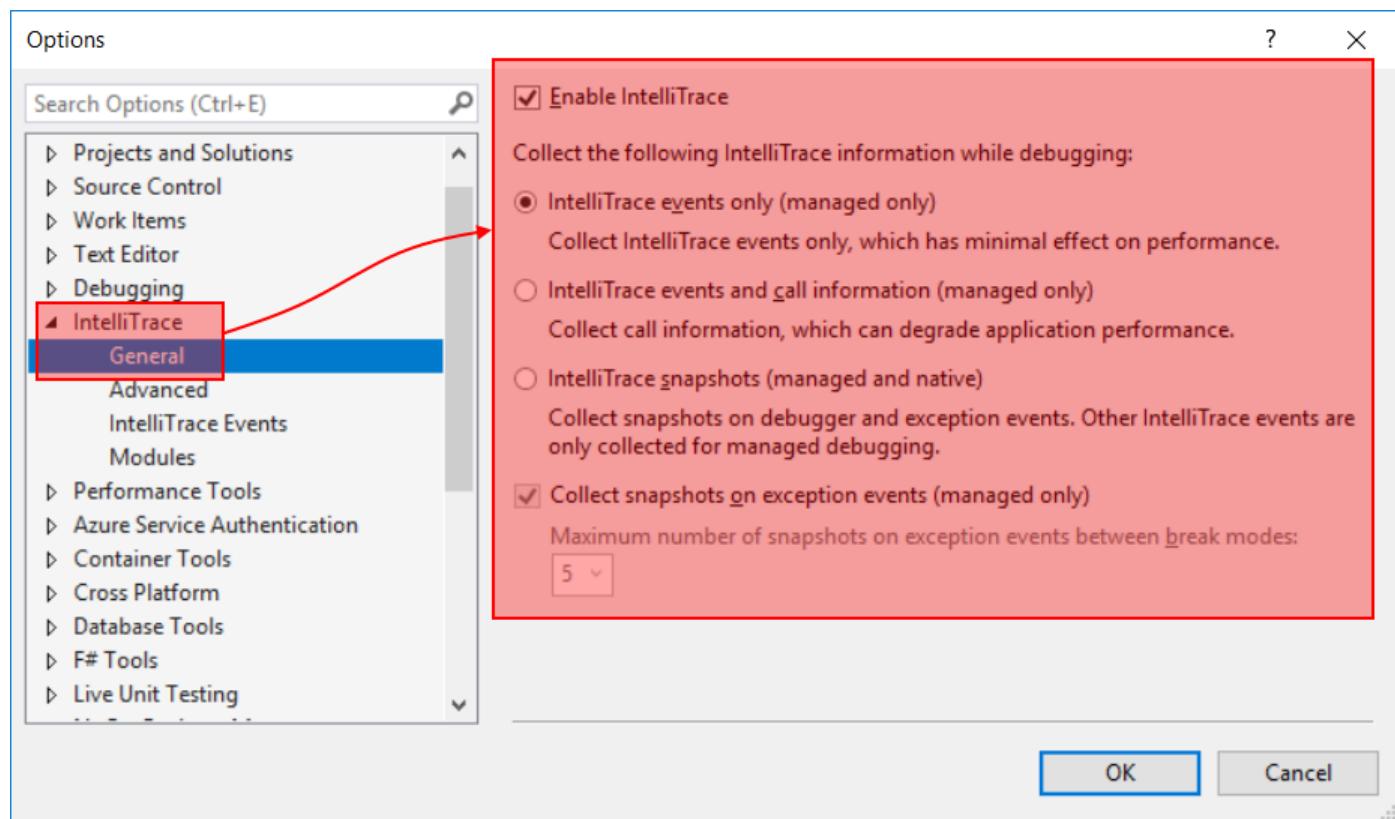
➤ You can see:

- References
- Who made changes
- When changes were done
- Commit description



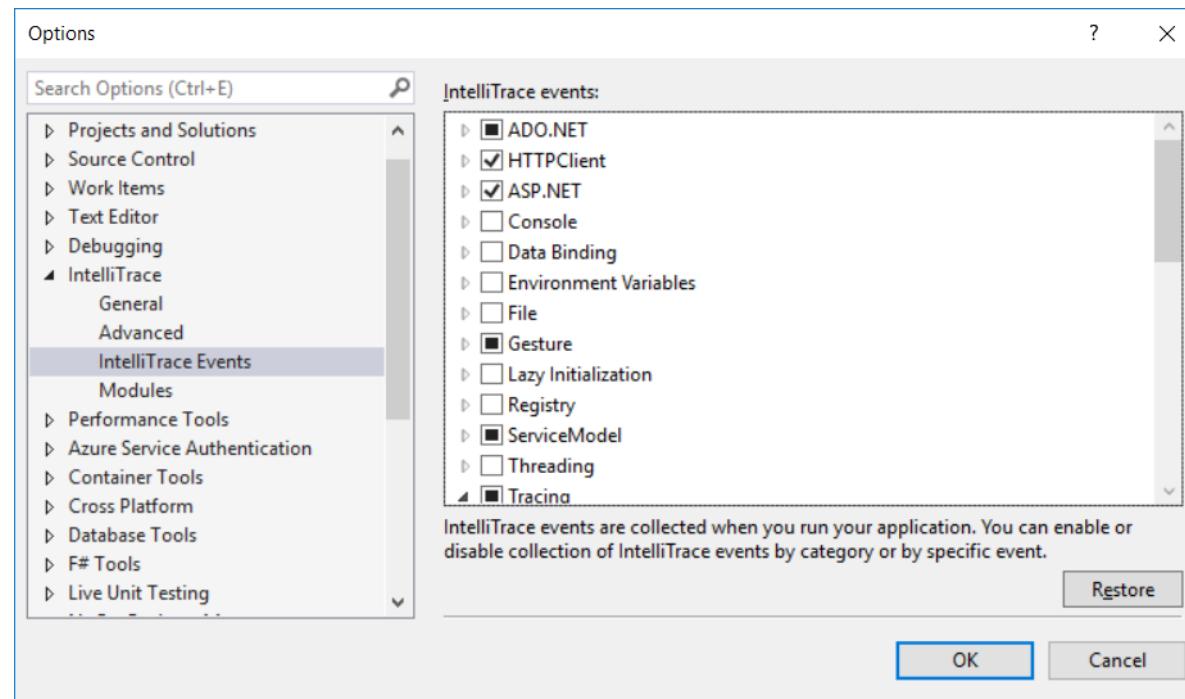
INTELLITRACE

- If you want to fix a bug
 - You debug and reproduce the bug
 - It records a trace file while you are reproducing the bug
 - You can come back afterwards and break into the trace, move back/forward
- Enable/Disable
 - Tools → Options → IntelliTrace → General
- Can specify what to trace
- Affects performance



INTELLITRACE

Be sure the event you want to trace is selected for tracing

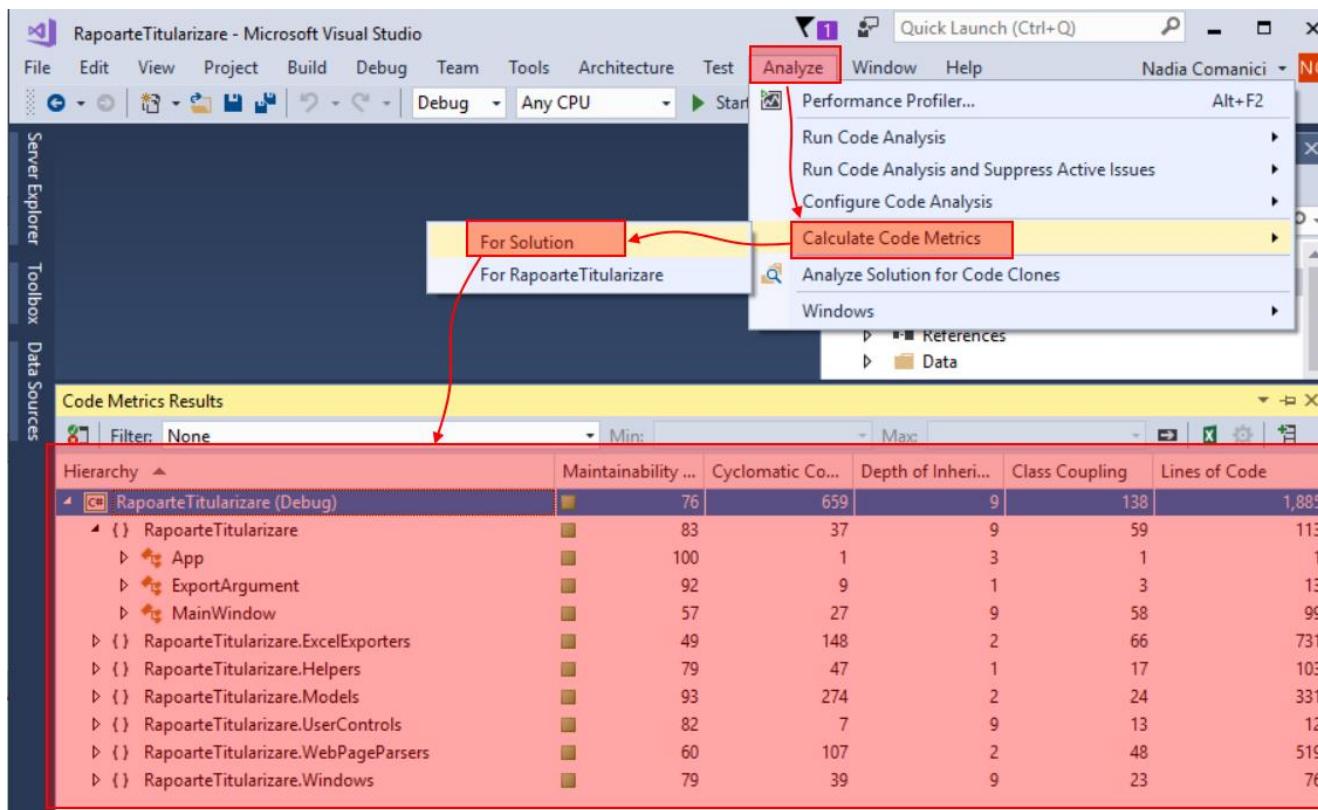


INTELLITRACE

- You can use the Historically Debug feature to go back/forward in the trace and identify what happened
- IntelliTrace saves itrace files, which you can load from another computer and historically debug

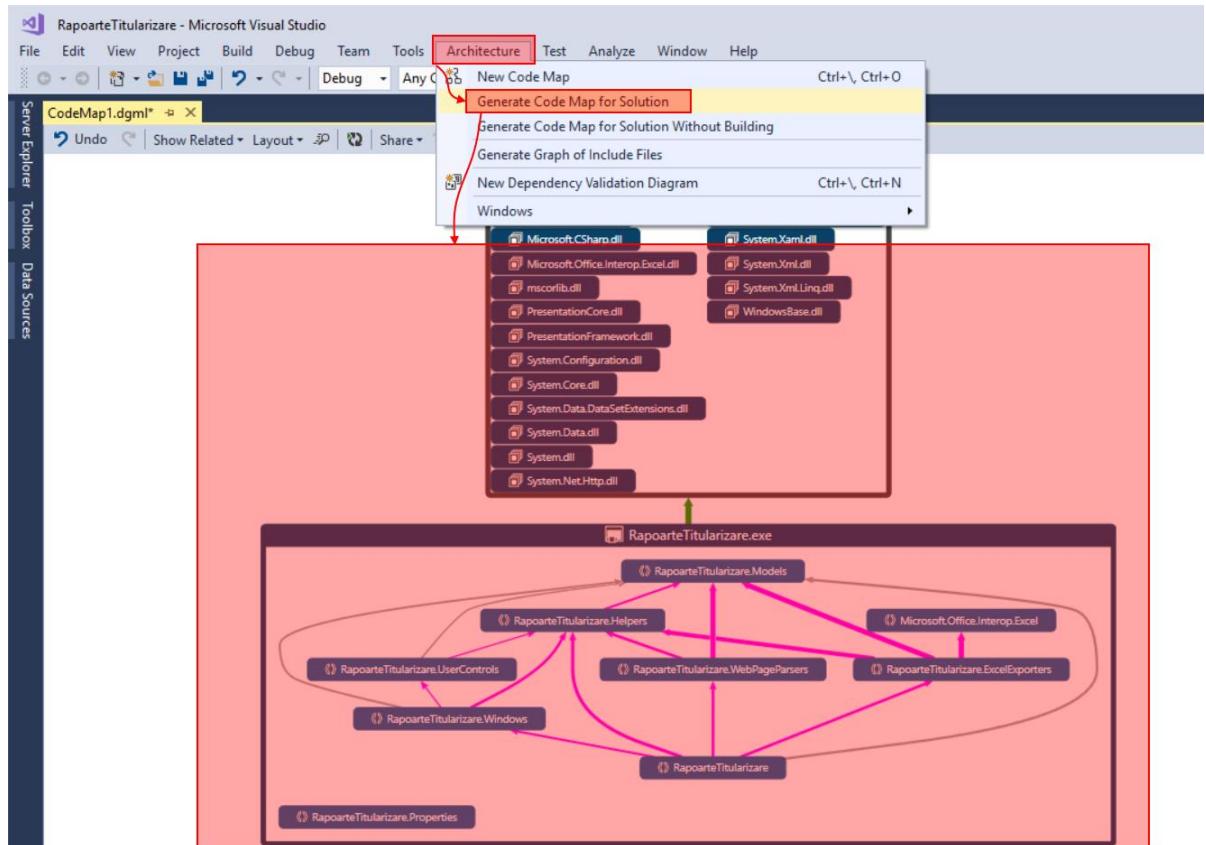
CODE METRICS

- More: <https://docs.microsoft.com/en-us/visualstudio/code-quality/code-metrics-values>



CODE MAP

- Can create a diagram with all the elements inside the solution and their dependencies
- Generates a dgml file, which is a graph
- Can generate
 - While debugging
 - For entire solution
 - Architecture → Generate Code Map for Solution
- Add comments on any element
- Share
 - Save as image
 - Save as XPS
 - Save as diagram in project



QUESTIONS

Contact: nadi_comanici@yahoo.com



FEEDBACK

<https://goo.gl/forms/1kSzCEcMVhTi6bWb2>

