Visual Studio Tips

Take some time to make your visual studio experience efficient.

1. Shortcuts

<table>
<thead>
<tr>
<th>Action</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go to definition</td>
<td>F12</td>
</tr>
<tr>
<td>Find all references of a variable</td>
<td>Shift F12</td>
</tr>
<tr>
<td>Go to declaration</td>
<td>Ctrl F12</td>
</tr>
<tr>
<td>Go to previous location or forward location</td>
<td>Ctrl hyphen or Ctrl Shift Hyphen</td>
</tr>
<tr>
<td>Turn on Intellisense</td>
<td>Ctrl Space</td>
</tr>
<tr>
<td>Navigate backwards/forwards</td>
<td>Use buttons on toolbar</td>
</tr>
<tr>
<td>Show Error List</td>
<td>Ctrl backslash, E</td>
</tr>
<tr>
<td>Show Call Stack (when debugging)</td>
<td>Ctrl Alt C</td>
</tr>
<tr>
<td>Toggle to Full Screen</td>
<td>Shift Alt Enter</td>
</tr>
<tr>
<td>Find</td>
<td>Ctrl F</td>
</tr>
<tr>
<td>Find in Files</td>
<td>Ctrl Shift F</td>
</tr>
<tr>
<td>Find Symbol Results</td>
<td>Ctrl Alt F12</td>
</tr>
<tr>
<td>Go to Line number</td>
<td>Ctrl G</td>
</tr>
<tr>
<td>Surround with (if, region, for, try, …)</td>
<td>Ctrl K, Ctrl S</td>
</tr>
<tr>
<td>Switch between tabs</td>
<td>Ctrl Tab (back), Ctrl Shift Tab (forward)</td>
</tr>
<tr>
<td>Toggle Word wrap</td>
<td>Ctrl R</td>
</tr>
<tr>
<td>Quickly navigate forward and backwards in the go to definition stack.</td>
<td>Ctrl+Shift+7 Ctrl+Shift+8</td>
</tr>
<tr>
<td>Comment and uncomment lines of code (note to comment you need the entire Ctrl K Ctrl C)</td>
<td>Ctrl K Ctrl C (to comment) Ctrl K Ctrl U (to uncomment)</td>
</tr>
<tr>
<td>Incremental Search</td>
<td>Ctrl I</td>
</tr>
</tbody>
</table>

2. Drag and drop code snippets

The Toolbox (Ctrl+Alt+X) window has multiple tabs. You can drag and drop code onto this window and copy it elsewhere. Some tabs do not allow dropping code into them; those that allow will have the appropriate icon.

3. Previous cursor positions

Ctrl -  This cycles you through the code positions you visited.

Ctrl Shift -  to navigate in the opposite direction.

4. Incremental search
To incrementally search for text as you type, first press Ctrl+i. Then type the word you want to search. Hit backspace to clear a character and enter to finish. Pressing F3 after this will work as usual, i.e. search for the next occurrence of previous search. Works without bringing up a search window.

5. Matching brace/comment/region/quote

Ctrl ] takes you to the matching brace. It also takes you to the matching comment, region or quote depending on what is at the cursor now.

6. Vertical block selection

Press Alt and then select the area you want with your mouse. Normally selection is horizontal (taking whole lines). This selects vertically.

7. Closing/Showing support windows

There are a bunch of necessary/useful windows in the Visual Studio IDE like Properties (F4), Solution Explorer (Ctrl Alt L), Output Window (Ctrl Alt O). However, they take up a lot of space. An easy way around this is to use the auto hide feature.

Open the window you want to hide. Right click on its title and choose Auto Hide. The window will dock into the margin whenever your mouse is not hovering over it. This is really convenient as then they only take up room when you are using them.

8. Tab groups - group code editor windows

If you have many source code windows open, you can group them logically using tab groups. Right click the tab of the code window and choose New Horizontal Tab Group. This will move the window into a split window, allowing you to see both files. You can add more files to this new tab group and also move files back to the previous group by choosing Move To Previous Tab Group (by right clicking the tab of the code window).

9. Auto-complete

Press Ctrl Space or Alt RightArrow to auto-complete the word. Intellisense suggestions may pop up a window if there is more than one possibility.

10. Intellisense suggestions window

Press Ctrl Shift Space to bring up the Intellisense suggestions window. When giving parameters for functions, I often need to escape the suggestions window to check another part of code. To bring it back, I used to delete a comma and then type it again; but this is easier.

10. Word wrap
To make it so a line wraps within your window:

Tools|Options|Text Editor|All Languages|General|Word Wrap

If you want to set this option for only one language, then choose the appropriate language instead of All Languages.

11. Line numbering

Tools|Options|Text Editor|All Languages|General|Line numbers.

If you want to set this option for only one language, then choose the appropriate language instead of All Languages.

12. Bookmarks

Bookmarks are available through Edit|Bookmarks. Bookmarks allow you to mark places in your code that you would want to come back to.

- Create/Remove Bookmark – Ctrl K, Ctrl K
- Move to next bookmark – Ctrl K, Ctrl N
- Move to previous bookmark – Ctrl K, Ctrl P
- Clear all bookmarks – Ctrl K, Ctrl L

13. Code Formatting

- Auto-format selection – Ctrl K, Ctrl F. It works best if (when formatting a method) the opening brace is on the same line as the method header.
- Convert to lower case – Ctrl U
- Convert to upper case – Ctrl Shift U
- Comment selection – Ctrl K, Ctrl C
- Uncomment selection – Ctrl K, Ctrl U

14. Outlining

I like this feature that allows me to hide code that is irrelevant to what I'm currently working on.

- Fold/Unfold the current code block – Ctrl M, Ctrl M
- Unfold all – Ctrl M, Ctrl L
- Stop outlining – Ctrl M, Ctrl P
- Fold all – Ctrl M, Ctrl O

15. Build and Debug

- Build – Ctrl Shift B
- Run – Ctrl F5
• Debug - F5
• Cycle through build errors - F8

16. Open header files

Right-click any filename in your source-code (i.e. in a string expression or #include line) and click "Open Document". Visual Studio will open the corresponding file in the IDE. This is nice if you wonder what things are defined in each header file.

17. Generate Browse Information

Select Generate Browse Info under the C/C++ project settings tab, for any source-files that you want to be included. This is very useful for searching for function definitions and declarations – just right-click a function / variable and select "Goto Definition Of..." and the IDE will take you straight there.

18. Refactoring – You have to buy Ref++ for refactoring support in C++. C# has it for free. This will be useful to you in the future.

I 'd bet you would agree that a significant part of any engineer's day is spent re-working existing code to become in some way 'better.' In many cases, 'better code' is a subjective term, but some common improvements include: adhering to OO-best practices, increasing type safety, improving performance, and increasing code readability and maintainability.

Refactoring is a formal and mechanical process, used to modify existing code in such a way that it does indeed become 'better' while preserving the program's intended functionality. In addition to improving a program's overall design, the refactoring process tends to yield code which is far easier to maintain and extend in the long run.


• Extract Method This allows you to define a new method based on a selection of code statements.
• Encapsulate Field Turns a public field into a private field encapsulated by a .NET property.
• Extract Interface Defines a new interface type based on a set of existing type members.
• Reorder Parameters Provides a way to reorder member arguments.
• Remove Parameters As you would expect, this refactoring removes a given argument from the current list of parameters.
• Rename This allows you to rename a code token (method name, field, local variable, and so on) throughout a project.
• Promote Local Variable to Parameter Moves a local variable to the parameter set of the defining method.