# Shortcuts

|  |  |  |
| --- | --- | --- |
| **Groups of Functions** | **Functions** | **Shortcuts** |
| **Edit** | Indent | Tab |
| Re-indent | Shift + Tab |
| Toggle single comment | Ctrl + / |
| Toggle block comment | Shift + Alt + A 🡪 change to Ctrl + Shift + / |
| Set multiple cursors | **Each time**: Hold Ctrl while clicking at different positions to add different cursors simultaneously  **All-in-one**: Black out the text, then press Shift + Alt + I. To move left / right / beginning / ending, press Left / Right / Home / End (to blackout while moving, combine these keys with Shift). |
| Delete a whole word | Ctrl + Delete / Backspace |
| Delete line(s) | Ctrl + Shift + K |
| Duplicate line(s) | Shift + Alt + ↑ / ↓ |
| Copy / cut line(s) | Ctrl + C / X |
| Move line(s) up / down | Alt + ↑ / ↓ |
| Show completion | Ctrl + Space |
| **Navigation** | Quick jump | **Jump to file**: Ctrl + P  **Jump to line**: Ctrl + P, then type : (or Ctrl + G)  **Jump to method**: Ctrl + P, then type @  **Jump back**: Alt + ←  **Jump forward**: Alt + →  **Jump to last edit**: Ctrl + K + Q --> change to Alt + -  **Jump to bracket**: Ctrl + Shift + \ --> change to Alt + ]  **Jump to Definition**: F12  **Jump to Declaration**: Ctrl + F12  **Open Peek Definition**: Alt + F12  **Open Peek Declaration**: Ctrl + Alt + F12 |
| Select occurrences of current word | **Each time**: Ctrl + D  **All-in-one**: Ctrl + F2  **Redo**: Ctrl + U |
| Go to:   * Beginning of a line * Ending of a line * Beginning of a file * Ending of a file | * Home * End * Ctrl + Home * Ctrl + End |
| Switch between files | Ctrl + Tab |
| **View** | Toggle Sidebar | Ctrl + B﻿ |
| Open error and warning window | Ctrl + Shift + M |
| **Others** | Save all | Ctrl + K + S --> change to Ctrl + Shift + S |
| Show Preference / Setting | Ctrl + , |
| Scroll fast with mouse wheel | Alt + Scroll |

# Extensions

## C/C++

### [C/C++](https://code.visualstudio.com/docs/languages/cpp)

Adds language support for C/C++, including:

Language services:

* Syntax highlight
* Code formatting (clang-format)
* Snippet
* Go to definition/declaration
* Peek definition/declaration
* Class/method/symbol navigation
* Signature help
* Quick info (hover)
* Error squiggles
* IntelliSense (c\_cpp\_properties.json)

Debugging:

* Windows (PDB, MinGW/Cygwin), Linux and macOS applications
* Line by line code stepping
* Breakpoints (including conditional and function breakpoints)
* Variable inspection
* Multi-threaded debugging support
* Core dump debugging support
* Executing GDB or MI commands directly when using 'C++ (GDB/LLDB)' debugging environment

### [Doxygen Documentation Generator](https://marketplace.visualstudio.com/items?itemName=cschlosser.doxdocgen)

Creates and updates Doxygen-style documentation comments in code.

### Cppcheck

Cppcheck is a static analysis tool for C/C++ code. Unlike C/C++ compilers and many other analysis tools, it doesn't detect syntax errors. Instead, Cppcheck detects types of bugs that the compilers normally fail to detect.

Common types of bugs Cppcheck can detect: uninitialized/unused variables and functions, out of bounds, exception safety, memory leaks, obsolete functions, invalid usage of STL, etc.

Cppcheck guide: [http://cppcheck.sourceforge.net/manual.pdf](https://www.youtube.com/redirect?q=http%3A%2F%2Fcppcheck.sourceforge.net%2Fmanual.pdf&redir_token=1rAcDj9HICj81pqUg5flpmwo5Ph8MTU0NjUyMjcxMEAxNTQ2NDM2MzEw&stzid=UgwXf60YkcI5ugBscnV4AaABAg&event=comments)

## C#

### [C#](https://marketplace.visualstudio.com/items?itemName=ms-vscode.csharp)

Adds language support for C#, including:

Language services:

* Code formatting
* Snippet
* Go to definition
* Peek definition
* Find all references
* Class/method/symbol navigation
* Quick info (hover)
* Error squiggles/Quick fixes/Suggestions
* Add and show/hide using
* IntelliSense
* CodeLens
* Support for project.json and csproj projects on Windows, macOS and Linux

Debugging:

* Support for .NET Core (CoreCLR). NOTE: Mono debugging is not supported. Desktop CLR debugging has limited support.

### [C# XML Documentation Comments](https://marketplace.visualstudio.com/items?itemName=k--kato.docomment)

Generate XML documentation comments for VS Code (built-in feature in VS IDE).

## Python

### [Python](https://marketplace.visualstudio.com/items?itemName=ms-python.python)

Adds language support for C#, including: IntelliSense, linting, debugging, code navigation, code formatting, Jupyter notebook support, refactoring, variable explorer, test explorer, snippets, and more!

### [autoDocstring](https://marketplace.visualstudio.com/items?itemName=njpwerner.autodocstring)

Quickly generate a docstring snippet (with briefs, parameters, return values, decorators, errors, etc.) that can be tabbed through.

## Markdown

### [reStructuredText](https://docs.restructuredtext.net/)

This plugin provides rich reStructuredText language support for VS Code, including:

* Syntax highlighting
* Code snippets
* Live preview
* Section builder
* Linter
* IntelliSense

### [Text Tables](https://marketplace.visualstudio.com/items?itemName=RomanPeshkov.vscode-text-tables)

Help create and format table easily.

### [PlantUML](https://marketplace.visualstudio.com/items?itemName=jebbs.plantuml)

This plugin provides rich support for PlantUML in VS Code, including:

* Preview Diagram (Alt-D)
  + Instant preview
  + Auto update
  + Zoom & scroll
* Export Diagrams
  + Export to images
  + Generate URLs.
* Editing Supports
  + Format PlantUML code
  + Syntax highlight
  + Snippets
  + Auto Include
  + Symbol List support

Supported formats: \*.wsd, \*.pu, \*.puml, \*.plantuml, \*.iuml

## Visual Enhancements

### [Monokai](https://marketplace.visualstudio.com/items?itemName=AndreyVolosovich.monokai-st3) ST3

This theme is exactly the same Monokai's theme of Sublime Text but for Visual Studio Code instead.

### [Text Marker (Highlighter)](https://marketplace.visualstudio.com/items?itemName=ryu1kn.text-marker)

Highlight selected text for better view.

### [TODO Highlight](https://marketplace.visualstudio.com/items?itemName=wayou.vscode-todo-highlight)

Highlight TODO, FIXME and your own defined annotations within your code.

### [Bracket Pair Colorizer 2](https://marketplace.visualstudio.com/items?itemName=CoenraadS.bracket-pair-colorizer-2)

Match brackets with colours. The user can define which tokens to match and which colours to use.

## Others

### [Comment Translate](https://marketplace.visualstudio.com/items?itemName=intellsmi.comment-translate)

Uses the Google Translate API to translate comments among different languages.

### [Code Spell Checker](https://marketplace.visualstudio.com/items?itemName=streetsidesoftware.code-spell-checker)

Check spelling mistakes for both code and comments.

### [Bookmarks](https://marketplace.visualstudio.com/items?itemName=alefragnani.Bookmarks)

Mark any line of code. Also, support quick moving between multiple bookmarks.

### [Save As Snippet](https://marketplace.visualstudio.com/items?itemName=artydeveloperduck.save-as-snippet)

Convert code blocks to snippet on the fly.

Alternative: <https://snippet-generator.app/>

### [Box Code](https://marketplace.visualstudio.com/items?itemName=PabloAlmonte.boxcode)

Save fragments of your code to use it at another time.

OTHERs

C# Snippets: https://marketplace.visualstudio.com/items?itemName=jorgeserrano.vscode-csharp-snippets

Collect some helpful snippets and create my own snippets using Save as snippet

Live Share: https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare

Maybe for future uses

.NET Core Extension Pack: https://marketplace.visualstudio.com/items?itemName=doggy8088.netcore-extension-pack

Good references to explorer many helpful extensions for C#

C# Extensions: https://marketplace.visualstudio.com/items?itemName=jchannon.csharpextensions

Use it when coding to explore its functions

Better Align: https://marketplace.visualstudio.com/items?itemName=wwm.better-align

Cursor Align: https://marketplace.visualstudio.com/items?itemName=yo1dog.cursor-align

Comment Aligner: https://marketplace.visualstudio.com/items?itemName=huangbaoshan.comment-aligner

Use it when coding to explore its functions

Comment Bars: https://marketplace.visualstudio.com/items?itemName=zfzackfrost.commentbars

Use it when coding to explore its functions

Enhanced Syntax Highlighting for C#

# c\_cpp\_properties.json

Sample:

{

"configurations": [

{

"name": "Win32",

"defines": [

"\_DEBUG",

"UNICODE"

],

"intelliSenseMode": "clang-x64",

"compilerPath": "C:/MinGW/bin/gcc.exe",

"includePath": [

"${workspaceFolder}",

"C:/MinGW/lib/gcc/mingw32/6.3.0/include/c++",

"../../Common/"

],

"browse": {

"path": [

"${workspaceFolder}",

"C:/MinGW/lib/gcc/mingw32/6.3.0/include/c++",

"../../Common/"

],

"limitSymbolsToIncludedHeaders": true,

"databaseFilename": ""

}

}

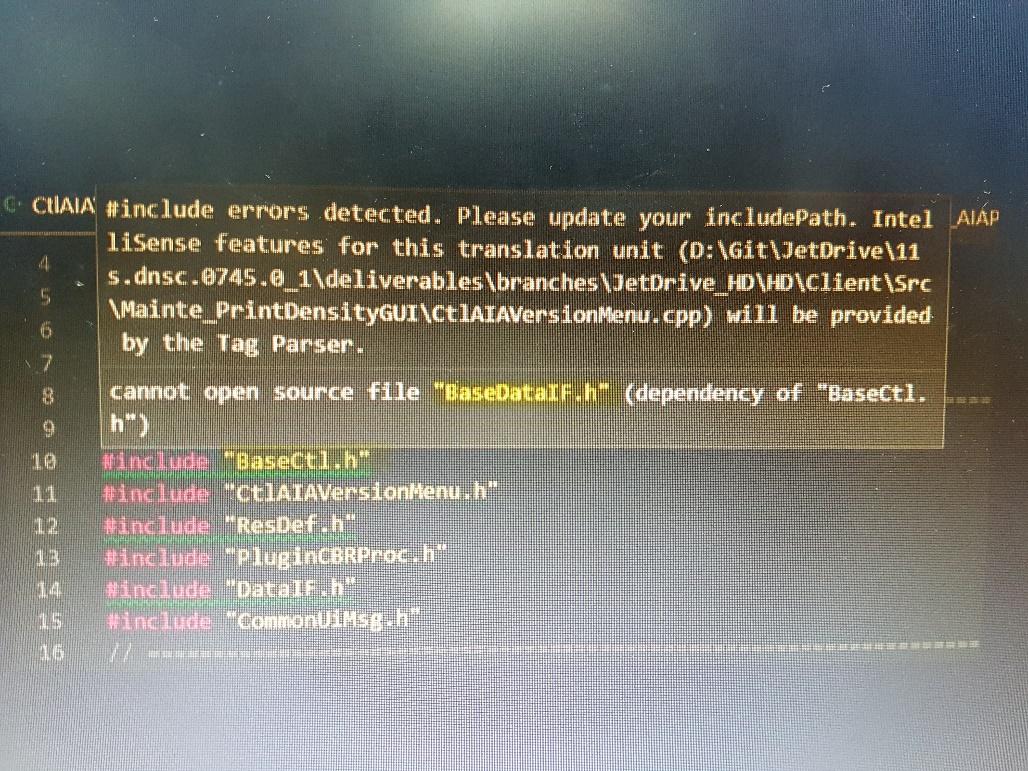
],

"version": 4

}

**Note:**

* "*includePath*" --> resolve underlines in headers
* "*path*" --> should have all paths as "*includePath*" to be able to use "go to definition" and other similar features in VS Code
* "*C:/MinGW/lib/gcc/mingw32/6.3.0/include*" = "*C:\\MinGW\\lib/gcc\\mingw32\\6.3.0\\include*"
* Don’t mistake between *BaseDataIF.h* and *BaseCtl.h* as below picture:



**Tips:**

1. In some cases, VS Code can detect the path for libraries. So, no need to find libraries manually with Windows Explorer. Instead, click on the **bulk** icon and VS Code will automatically include the path for these libraries.

2. If working with Visual Studio IDE before, should copy all libraries added to “Additional Include Directories” in “Property Pages > C/C++ -> General”

3. **Recursive search** of includePath and Global IntelliSense settings: Read [here](https://devblogs.microsoft.com/cppblog/visual-studio-code-cc-extension-may-2018-update-intellisense-configuration-just-got-so-much-easier).

# Tips

## Snippets

<https://code.visualstudio.com/docs/editor/userdefinedsnippets>

## Code Folding

Ctrl + Shift + P to open the command palette -> type “fold” then choose any folding option

More details: <https://code.visualstudio.com/docs/editor/codebasics#_folding>

## File Comparison

In the Explorer, right click to select a file and choose “Select for Compare” from the context menu.

Right click to select another file to which you want to compare, and choose “Compare with <previously selected file for compare>”

Compare windows will appear in a *side by side view* by default. You can switch to *inline view* via *More Actions…* setting on the right corner of the VS Code window.