Visual Studio Code Setup

This is an extended guide on how to setup Visual Studio Code to use Intellisense for HPL3 and SOMA. Visual Studio Code is a lightweight notepad-esque version of Microsoft's full IDE Visual Studio. As Visual Studio Code is still under active development, some steps of this guide may need to be altered, so feel free to do so as the need arises.

Note: Even though Visual Studio Code is a cross-platform program, HPL3 is designed to be modded on a Windows machine. As such, the instructions in this guide assume that you are on a Windows OS. (Linux users who have successfully gotten the HPL3 modding tools may use this guide as a base to get Intellisense working on their platform.)

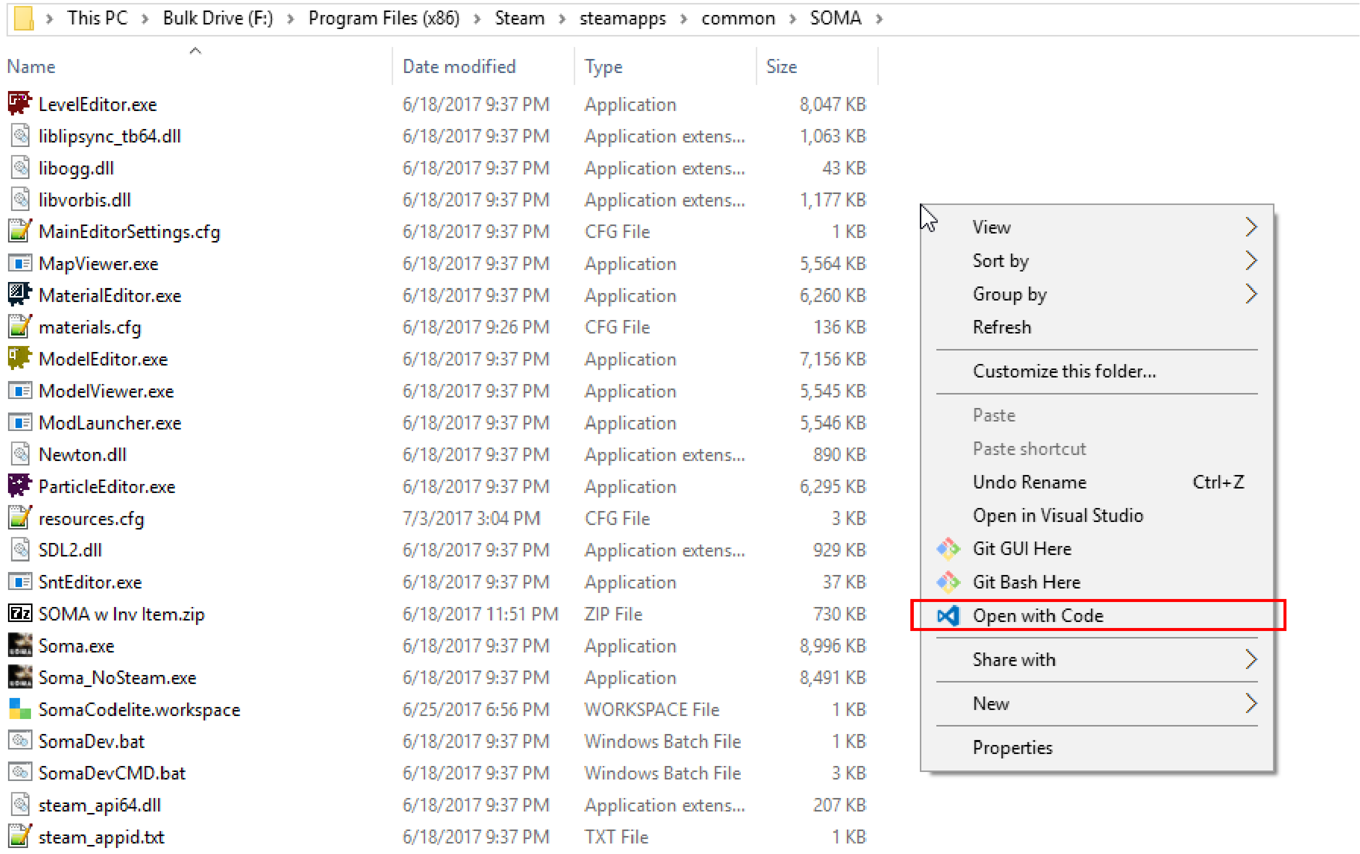
Install VS Code

You will need to download and install VS Code. You can do so from Microsoft's VS Code website. Simply choose your desired platform and download the installer. Follow the instructions given in the installer to install VS Code on your computer.

Open SOMA Folder in VS Code

VS Code has the capability of opening an entire folder into its internal directory tree. Doing this gives you the benefit to be able to quickly and easily navigate to files in that folder and in subsequent folders. It also grants added benefits for inter-file relationships, such as C++ linking capabilities.

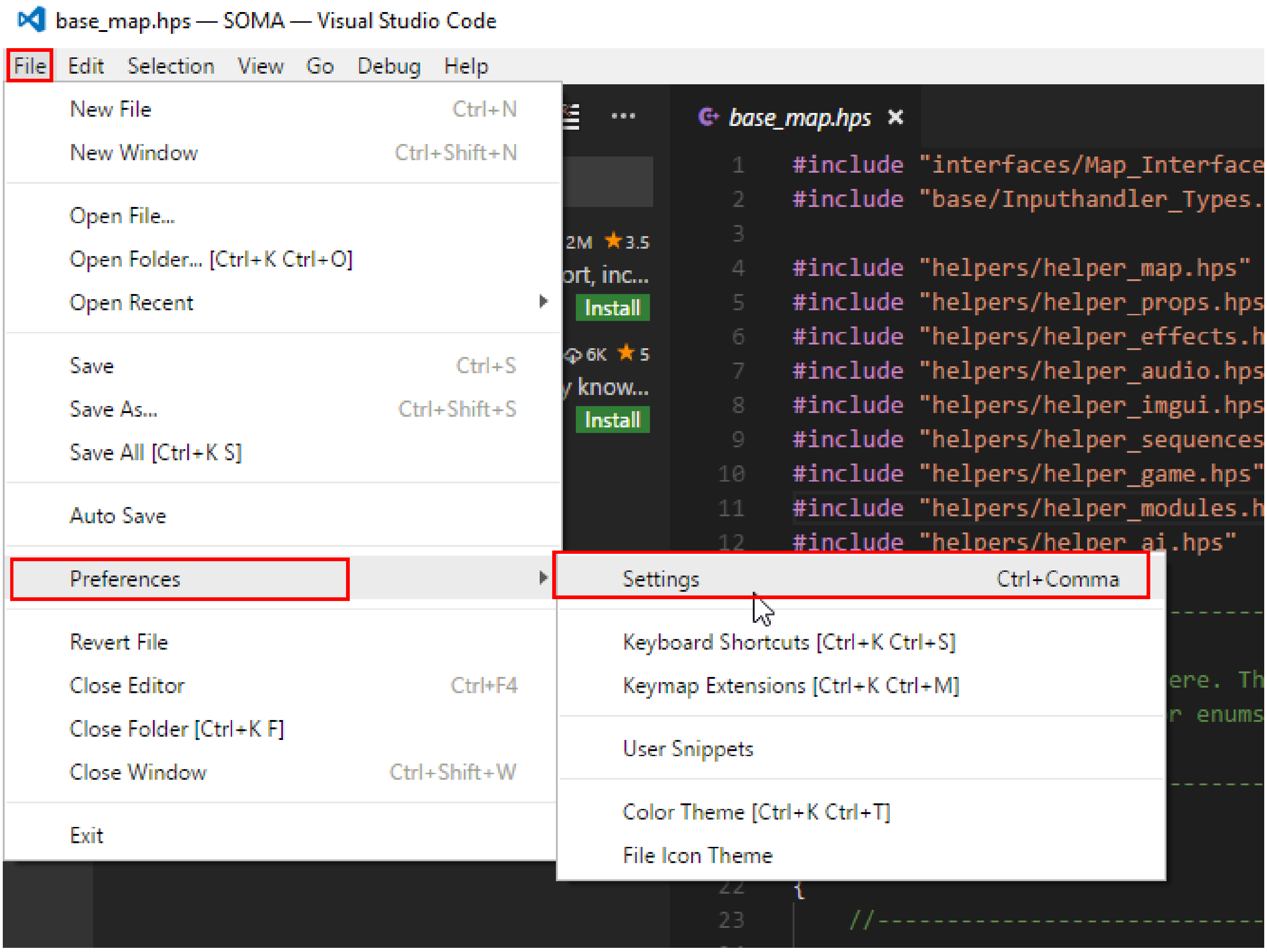
To do this, simply navigate to your SOMA installation folder and right-click on the window background. (i.e. Don't click on any file or folder within the SOMA folder.) Then select the “Open with Code” option from the drop down menu:



Name	Date modified	Type	Size
LevelEditor.exe	6/18/2017 9:37 PM	Application	8,047 KB
liblipsync_tb64.dll	6/18/2017 9:37 PM	Application extens...	1,063 KB
libogg.dll	6/18/2017 9:37 PM	Application extens...	43 KB
libvorbis.dll	6/18/2017 9:37 PM	Application extens...	1,177 KB
MainEditorSettings.cfg	6/18/2017 9:37 PM	CFG File	1 KB
MapViewer.exe	6/18/2017 9:37 PM	Application	5,564 KB
MaterialEditor.exe	6/18/2017 9:37 PM	Application	6,260 KB
materials.cfg	6/18/2017 9:26 PM	CFG File	136 KB
ModelEditor.exe	6/18/2017 9:37 PM	Application	7,156 KB
ModelViewer.exe	6/18/2017 9:37 PM	Application	5,545 KB
ModLauncher.exe	6/18/2017 9:37 PM	Application	5,546 KB
Newton.dll	6/18/2017 9:37 PM	Application extens...	890 KB
ParticleEditor.exe	6/18/2017 9:37 PM	Application	6,295 KB
resources.cfg	7/3/2017 3:04 PM	CFG File	3 KB
SDL2.dll	6/18/2017 9:37 PM	Application extens...	929 KB
SntEditor.exe	6/18/2017 9:37 PM	Application	37 KB
SOMA w Inv Item.zip	6/18/2017 11:51 PM	ZIP File	730 KB
Soma.exe	6/18/2017 9:37 PM	Application	8,996 KB
Soma_NoSteam.exe	6/18/2017 9:37 PM	Application	8,491 KB
SomaCodelite.workspace	6/25/2017 6:56 PM	WORKSPACE File	1 KB
SomaDev.bat	6/18/2017 9:37 PM	Windows Batch File	1 KB
SomaDevCMD.bat	6/18/2017 9:37 PM	Windows Batch File	3 KB
steam_api64.dll	6/18/2017 9:37 PM	Application extens...	207 KB
steam_appid.txt	6/18/2017 9:37 PM	TXT File	1 KB

Associate HPS files with C++

The first thing to do is to associate HPL3 script files with VS Code's C++ syntax highlighting engine. To do so, you need to open VS Code's preferences file:



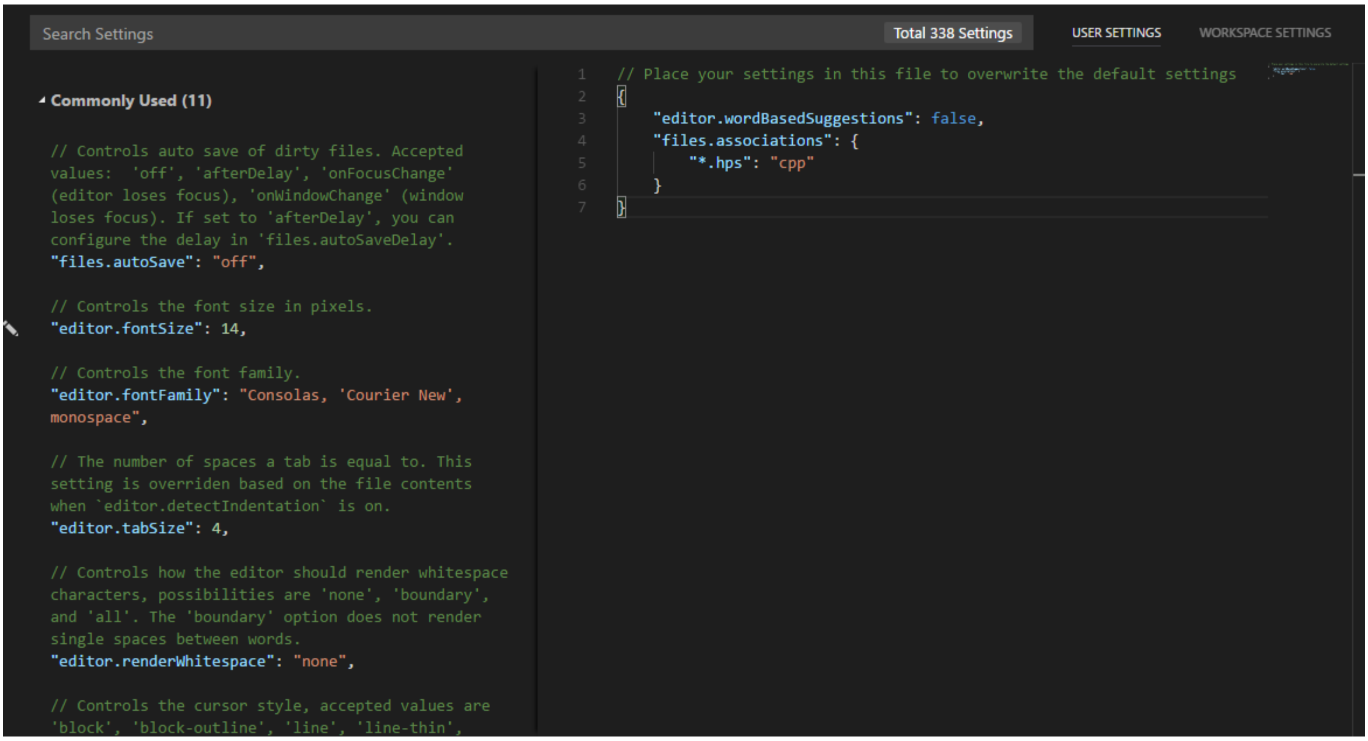
```
#include "interfaces/Map_Interface
#include "base/Inputhandler_Types.

#include "helpers/helper_map.hps"
#include "helpers/helper_props.hps
#include "helpers/helper_effects.h
#include "helpers/helper_audio.hps
#include "helpers/helper_imgui.hps
#include "helpers/helper_sequences
#include "helpers/helper_game.hps"
#include "helpers/helper_modules.h
#include "helpers/helper_ai.hps"
```

In VS Code, settings are saved in a JSON file. The window on the left is the VS Code default settings, while the window on the right is a JSON file for creating and saving user configurations.

To associate `.hps` files with C++, simply copy the following text into the user preferences window (on the right):

```
"files.associations": {
    "*.hps": "cpp"
}
```

Afterwards, your preferences file should look something like this:

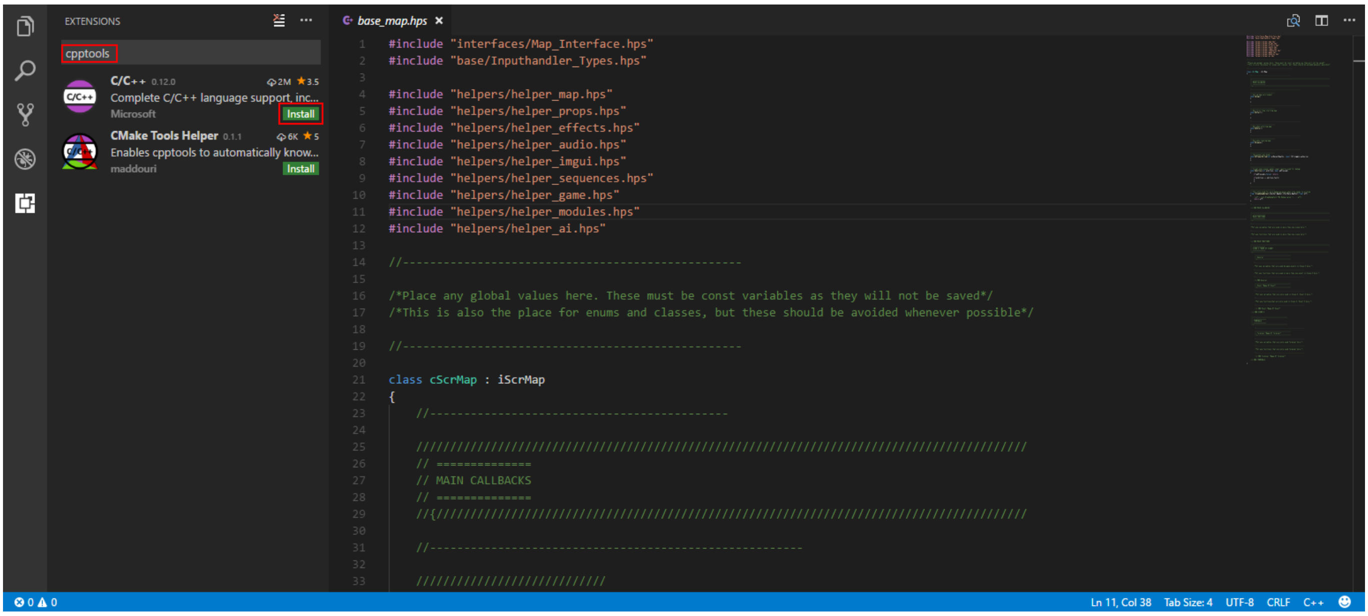


```
// Controls auto save of dirty files. Accepted
values:  'off', 'afterDelay', 'onFocusChange'
(editor loses focus), 'onWindowChange' (window
loses focus). If set to 'afterDelay', you can
configure the delay in 'files.autoSaveDelay'.
"files.autoSave": "off",

// Controls the font size in pixels.
"editor.fontSize": 14,

// Controls the font family.
"editor.fontFamily": "Consolas, 'Courier New',
monospace",

// The number of spaces a tab is equal to. This
setting is overriden based on the file contents
when `editor.detectIndentation` is on.
"editor.tabSize": 4,

// Controls how the editor should render whitespace
characters, possibilities are 'none', 'boundary',
and 'all'. The 'boundary' option does not render
single spaces between words.
"editor.renderWhitespace": "none",

// Controls the cursor style, accepted values are
'block', 'block-outline', 'line', 'line-thin',
```

```
// Place your settings in this file to overwrite the default settings
{
    "editor.wordBasedSuggestions": false,
    "files.associations": {
        "*.hps": "cpp"
    }
}
```

Install CPPTools extension

Next, you will need to get the C/C++ extension for VS Studio. To do this, go to the Extensions tab on the left side of the window. In the search bar, search for “C/C++”. The extension you want should appear at the top of the list (it's authored by Microsoft). After you find it, click the green “Install” button:

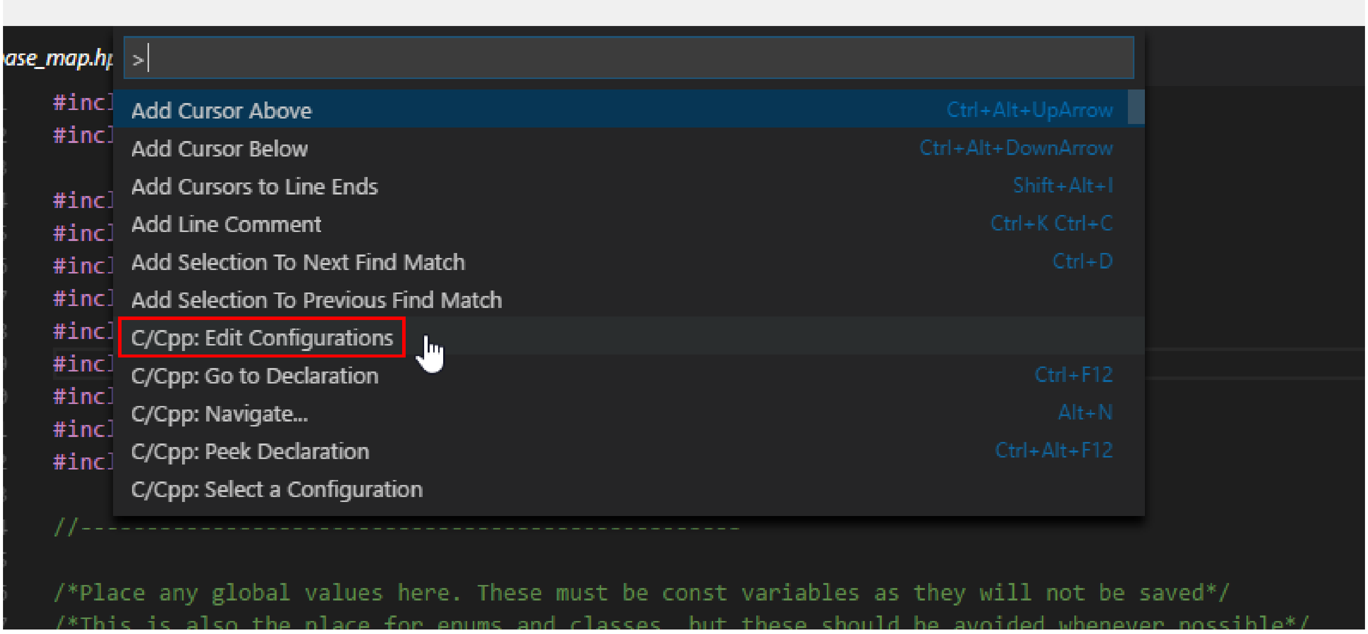


```
#include "interfaces/Map_Interface.hps"
#include "base/Inputhandler_Types.hps"

#include "helpers/helper_map.hps"
#include "helpers/helper_props.hps"
#include "helpers/helper_effects.hps"
#include "helpers/helper_audio.hps"
#include "helpers/helper_imgui.hps"
#include "helpers/helper_sequences.hps"
#include "helpers/helper_game.hps"
#include "helpers/helper_modules.hps"
#include "helpers/helper_ai.hps"

//----------------------------------------------------

/*Place any global values here. These must be const variables as they will not be saved*/
/*This is also the place for enums and classes, but these should be avoided whenever possible*/

//----------------------------------------------------

class cScrMap : iScrMap
{
    //------------------------------------------

    //////////////////////////////////////////////////////////////////////////////////////////
    // ============
    // MAIN CALLBACKS
    // ============
    //{//////////////////////////////////////////////////////////////////////////////////////

    //-------------------------------------------------------

    //////////////////////////
```

(After you install it, the window will need to be reloaded twice. The first time is after installing the extension, and the second time is after the extension downloads and installs its needed dependencies. Simply click on the “Reload” button as it appears and VS Code will restart itself for you, retaining your open files.)

Configure The HPL3 Profile

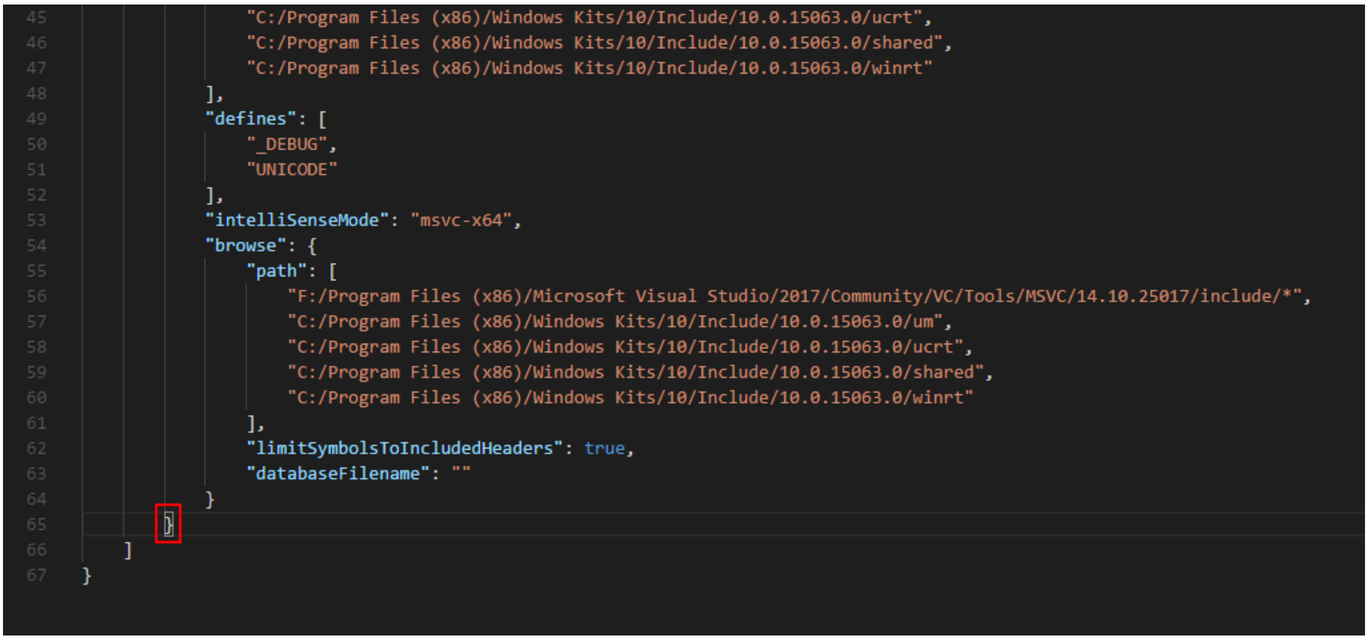
Now you need to congifure the extension to use the HPL3 script files for its Intellisense. To do this, open the Command Palette by pressing “Ctrl-Shift-P”. From there, find and select the “C/Cpp: Edit Configurations” option:


ase_map.hp
>
Add Cursor Above Ctrl+Alt+UpArrow
Add Cursor Below Ctrl+Alt+DownArrow
Add Cursors to Line Ends Shift+Alt+I
Add Line Comment Ctrl+K Ctrl+C
Add Selection To Next Find Match Ctrl+D
Add Selection To Previous Find Match
C/Cpp: Edit Configurations
C/Cpp: Go to Declaration Ctrl+F12
C/Cpp: Navigate... Alt+N
C/Cpp: Peek Declaration Ctrl+Alt+F12
C/Cpp: Select a Configuration
#incl

```
//-------------------------------------------------

/*Place any global values here. These must be const variables as they will not be saved*/
/*This is also the place for enums and classes, but these should be avoided whenever possible*/
```

On this page, you will see a lot of existing JSON text. This was automatically generated by the extension, and there's no need to change it. Instead, we are going to add a new profile for the Intellisense to use in our projects.

Scroll to the bottom and put the cursor after the closing bracket of the last profile. (It's the third bracket from the bottom.)



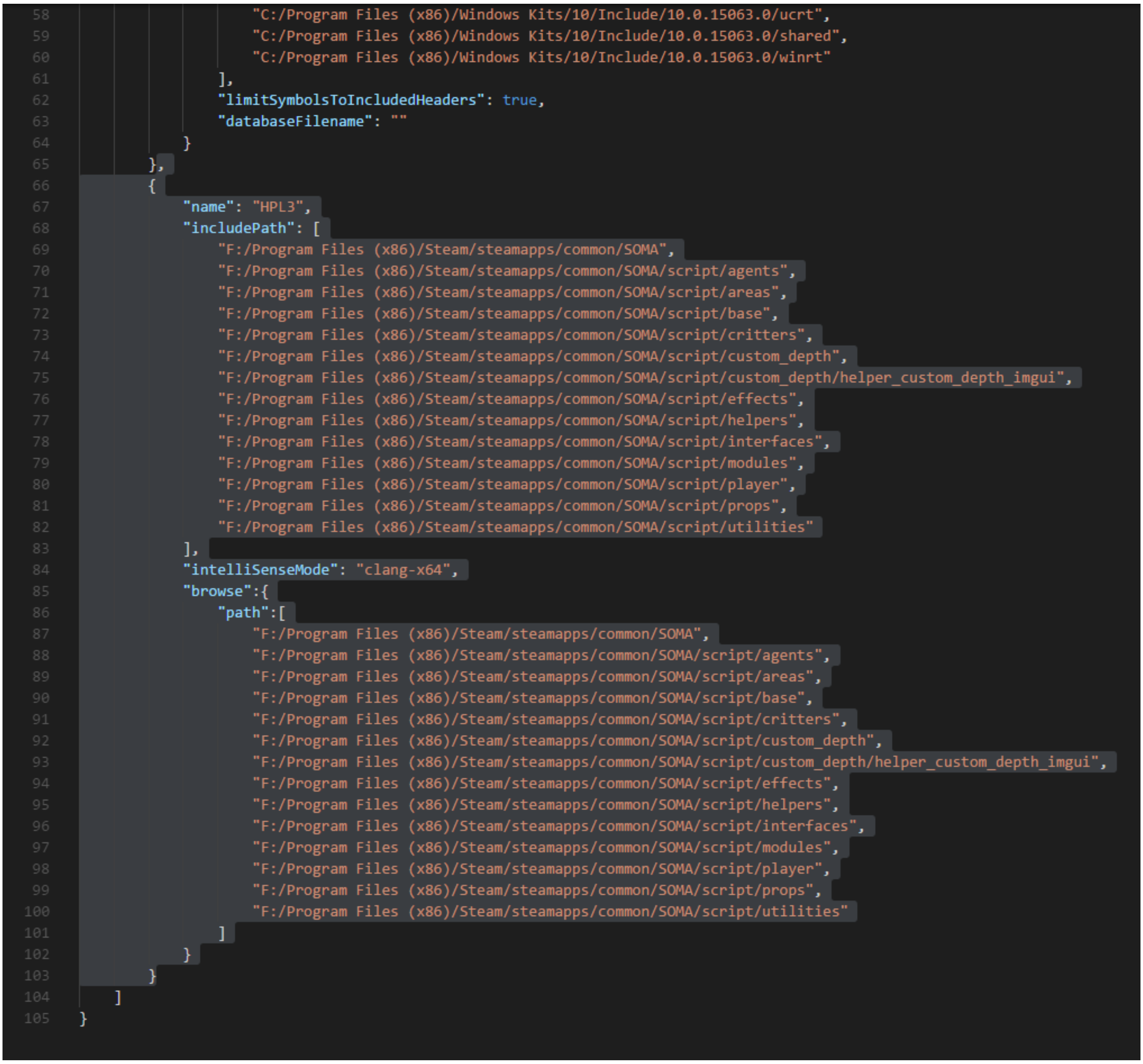
```
                "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/ucrt",
                "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/shared",
                "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/winrt"
            ],
            "defines": [
                "_DEBUG",
                "UNICODE"
            ],
            "intelliSenseMode": "msvc-x64",
            "browse": {
                "path": [
                    "F:/Program Files (x86)/Microsoft Visual Studio/2017/Community/VC/Tools/MSVC/14.10.25017/include/*",
                    "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/um",
                    "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/ucrt",
                    "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/shared",
                    "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/winrt"
                ],
                "limitSymbolsToIncludedHeaders": true,
                "databaseFilename": ""
            }
        }
    ]
}
```

After that bracket, add a comma, add a new line (press Enter), then paste the following text:

```
{
    "name": "HPL3",
    "includePath": [
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA",
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/agents",
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/areas",
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/base",
        "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/critters",
        "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/custom_depth",
        "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/custom_depth/helper_custom_depth_im
gui",
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/effects",
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/helpers",
        "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/interfaces",
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/modules",
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/player",
        "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/props",
        "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/utilities"
    ],
    "intelliSenseMode": "clang-x64",
    "browse":{
        "path":[
            "F:/Program Files (x86)/Steam/steamapps/common/SOMA",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/agents",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/areas",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/base",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/critters",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/custom_depth",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/custom_depth/helper_custom_depth_im
gui",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/effects",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/helpers",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/interfaces",
            "F:/Program Files
(x86)/Steam/steamapps/common/SOMA/script/modules",
            "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/player",
            "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/props",
            "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/utilities"
        ]
    }
}
```

(Change the path in each one to the folder where your installation of SOMA is located, obviously.)

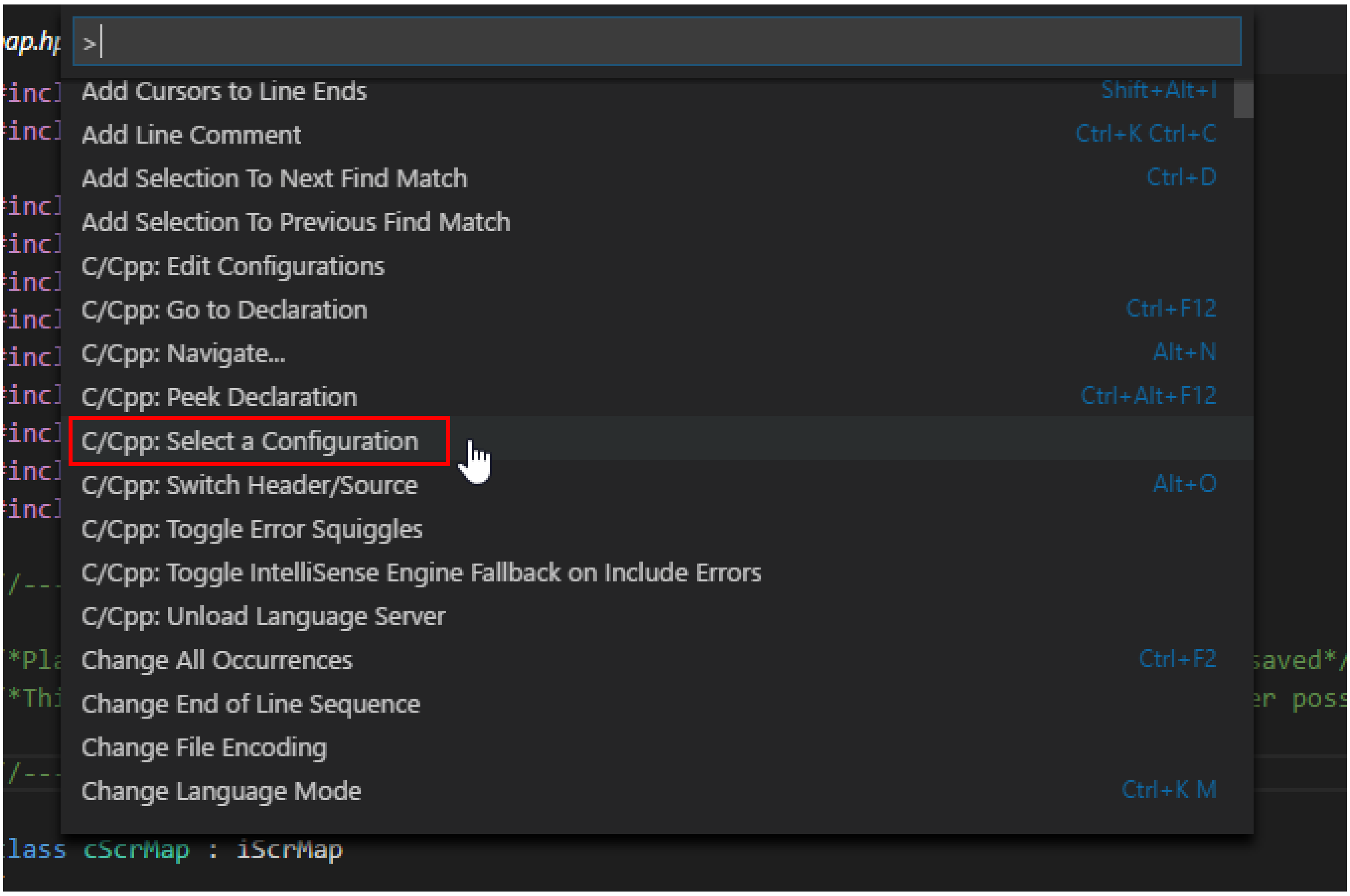
Afterwards, the text should look like this (the new text is highlighted):



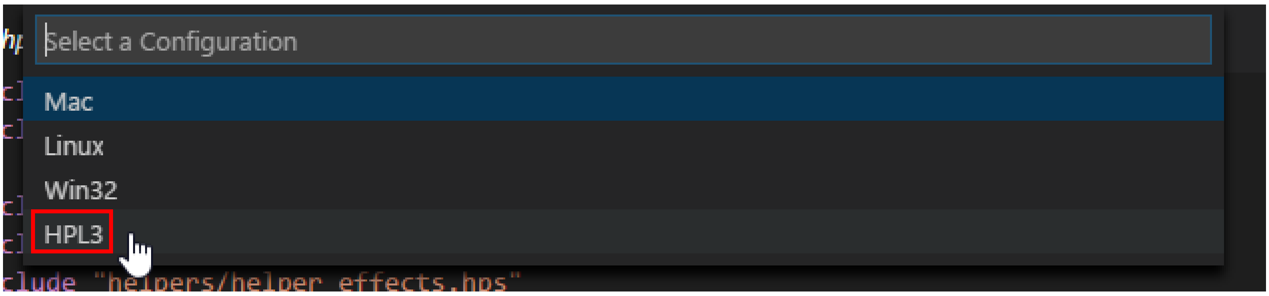
```
                    "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/ucrt",
                    "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/shared",
                    "C:/Program Files (x86)/Windows Kits/10/Include/10.0.15063.0/winrt"
                ],
                "limitSymbolsToIncludedHeaders": true,
                "databaseFilename": ""
            }
        },
        {
            "name": "HPL3",
            "includePath": [
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/agents",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/areas",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/base",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/critters",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/custom_depth",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/custom_depth/helper_custom_depth_imgui",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/effects",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/helpers",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/interfaces",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/modules",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/player",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/props",
                "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/utilities"
            ],
            "intelliSenseMode": "clang-x64",
            "browse":{
                "path":[
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/agents",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/areas",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/base",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/critters",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/custom_depth",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/custom_depth/helper_custom_depth_imgui",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/effects",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/helpers",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/interfaces",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/modules",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/player",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/props",
                    "F:/Program Files (x86)/Steam/steamapps/common/SOMA/script/utilities"
                ]
            }
        }
    ]
}
```

(If your text has any red squiggles in it, then you copied the text into the wrong place. Compare your text to the image above to make sure it looks the same.)

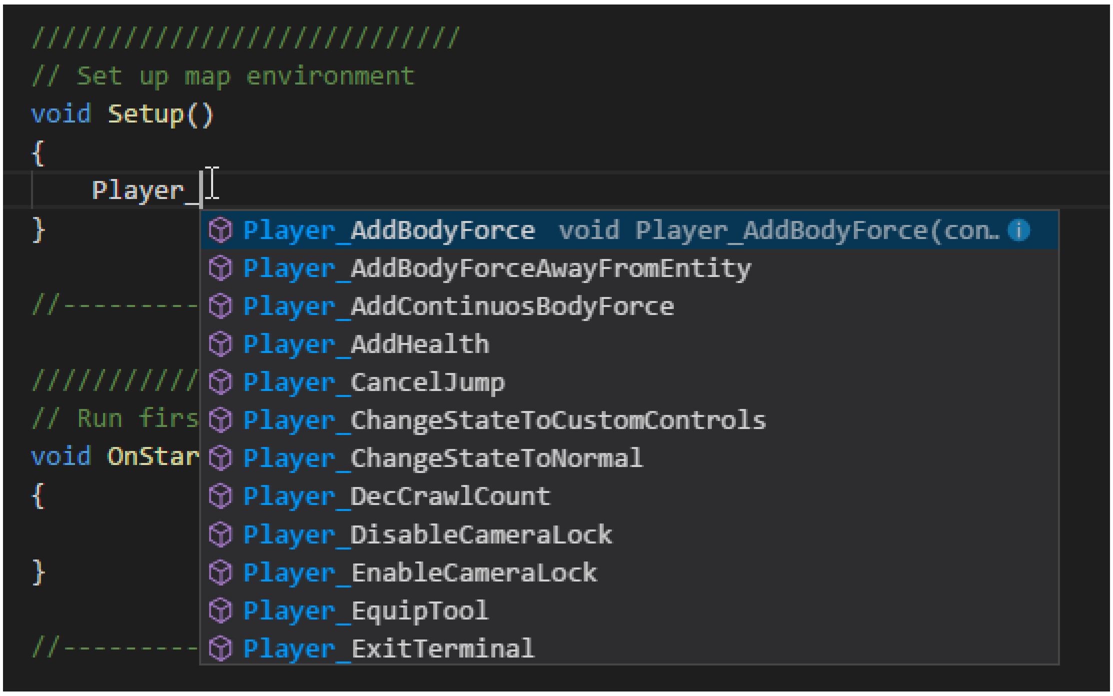
Now your profile is ready to go. Save and close the file, then go back to your map script. Open the Command Palette again ("Ctrl-Shift-P") and navigate to "C/Cpp: Select a Configuration":


>
Add Cursors to Line Ends Shift+Alt+I
Add Line Comment Ctrl+K Ctrl+C
Add Selection To Next Find Match Ctrl+D
Add Selection To Previous Find Match
C/Cpp: Edit Configurations
C/Cpp: Go to Declaration Ctrl+F12
C/Cpp: Navigate... Alt+N
C/Cpp: Peek Declaration Ctrl+Alt+F12
C/Cpp: Select a Configuration
C/Cpp: Switch Header/Source Alt+O
C/Cpp: Toggle Error Squiggles
C/Cpp: Toggle IntelliSense Engine Fallback on Include Errors
C/Cpp: Unload Language Server
Change All Occurrences Ctrl+F2
Change End of Line Sequence
Change File Encoding
Change Language Mode Ctrl+K M
lass cScrMap : iScrMap

Under the drop-down menu, select "HPL3":



And that's it! You're done! Go ahead and test it out in your map script:



```
////////////////////////////
// Set up map environment
void Setup()
{
    Player_
}

//---------

///////////
// Run firs
void OnStar
{

}

//---------
```

```
Player_AddBodyForce  void Player_AddBodyForce(con…
Player_AddBodyForceAwayFromEntity
Player_AddContinuosBodyForce
Player_AddHealth
Player_CancelJump
Player_ChangeStateToCustomControls
Player_ChangeStateToNormal
Player_DecCrawlCount
Player_DisableCameraLock
Player_EnableCameraLock
Player_EquipTool
Player_ExitTerminal
```

Optional: Launch SOMA from VS Code

Another useful thing you can do with Visual Studio Code is to set Custom Debug Configuration. With that in mind, we can make VS Code launch the SomaDev.bat (or any other custom bat) file through the debugger.

Go to the Debug Panel by pressing “Ctrl+Shift+D”. On the Debug tab, press on the configuration button and select C/C++ (Windows).

Override the existing code with the following:

```
{

// Use IntelliSense to learn about possible attributes.

// Hover to view descriptions of existing attributes.

// For more information, visit:
https://go.microsoft.com/fwlink/?linkid=830387

"version": "0.2.0",

"configurations": [
{

      "name": "SOMA",

      "type": "cppvsdbg",

      "request": "launch",

      "program": "${workspaceFolder}/SomaDevCMD.bat",
      "args": [],

      "stopAtEntry": false,

      "cwd": "${workspaceFolder}",

      "environment": [],

      "externalConsole": true

    }

  ]

}
```

You can add more configurations by pressing on the “Add Configuration” Button at the bottom right and selecting again C/C++ Windows (Launch).

Now, just select the debugger you want and press on the green arrow to launch it.

From:
https://wiki.frictionalgames.com/ - **Frictional Game Wiki**

Permanent link:
https://wiki.frictionalgames.com/hpl3/community/scripting/visual_studio_code_setup

Last update: **2018/03/24 19:36**