

# CodeLite - How to get started

1. [Download the latest CodeLite release.](#)
2. Launch CodeLite and configure:
  - \* Settings→Code Completion→Display and Behavior: Activate “Keep function signature unformatted”.
  - Settings→Code Completion→ctags→Search Paths: Add a path to “redist/script”.
  - Settings→Code Completion→ctags→Advanced: Add \*.hps to file types list.
  - Settings→Code Completion→Colouring: Enable colour local variables.
  - (optional) Settings→Code Completion→Colouring: Enable colour workspace tags, also enable enumerator but leave the rest with default setting.
  - (optional) Settings→Code Completion: Configure the rest to your liking. For example, auto-show & case sensitive enabled.
  - Settings→Colours and Fonts→Customize, select C++ and in Global Settings add \*.hps to the File Extension list.
3. Create a new workspace. Do this by going to Workspace menu and select “New Workspace...”. Pick C++ Workspace. Name is as you like and set the workspace path as you like.
4. Create a new project in the workspace. Do this by right-clicking on the workspace in the list to the left and choose the Create New Project option. Select others → Non-code project. Name as you like and leave rest as default.
5. Add one folder to the project. Right-click on the project folder and choose New Virtual Folder. Name it \_api (you can create others later when you know more).
6. Time to add files to the project.
  - \* Right-click on \_api and Add An Existing File, add the file “SOMA GAME FOLDER LOCATION/hps\_api.hps” (might need to launch the game executable one time to have this file created).
  - Right-click on the **project** in the list to the left, pick Import Files From Directory. Navigate to the SOMA GAME FOLDER and select it. In the next window, check the boxes for maps and scripts and make sure that the \*.hps file extension is present in the list of extensions to import. Click OK.

That should be it, you should now have three top folders in your project and a lot of sub-folders with various script files. Hps files should be color coded and parsed for completion as you open them in the project. If they do not, you can re-tag the workspace, project or the single file to see if it helps.

Start typing something, for example Entity (depending on configuration this can be case sensitive!) and a drop down should appear showing all the functions that begin with Entity and the documentation for each selected function. You can force things to appear using these shortcuts:

- ctrl-space will bring up the completion drop down.
- ctrl-shift-space will bring up hints when inside the () of a function.

## Issues & Info

- Write Class@ object and not Class @object, the later will screw up coloring.
- A bug currently can make some words loose coloring when you launch codelite. Open the Tags settings, click OK and the re-tag the file or project to fix it.
- If you loose all coloring and can't get it back, delete the .tags file saved in the same folder as

your workspace. Then relaunch codelite and re-tag workspace, save file to get it working.

- Parameters with `&in` will break coloring of parameter, but will have no other side-effects.

## Extra Info

Info about Settings→Colours and Fonts→Customize→Styles→Edit Lexer keyword sets, Set 0 to Set 4:

- Set 0 User configured list of keywords not already parsed. Can be edited.
- Set 1 Unknown.
- Set 2 Doxygen keywords, words here will be colored in comments using `/**` codeblocks when prepended with `@` ie `@var int myInt`. Can be edited.
- Set 3 Used for the setting “Colour workspace tags”, automatically created (not visibly). Can NOT be edited.
- Set 4 Used for the setting “Colour local variables”, automatically created (not visibly). Can NOT be edited.

From:

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

Permanent link:

<https://wiki.frictionalgames.com/hpl3/3rdparty/codelite?rev=1443725137>

Last update: **2015/10/01 19:45**

