Document Title : Procedure to build AceXtreme SDK Samples in Visual Studio – 2019
Document No : AN0002
Author : Ganesh Okade, RTips Technologies
Last edited : Oct 05, 2024
Introduction **:** This document explains the steps for creating a Visual Studio C++ console project to rebuild the samples provided by DDC's AceXtreme 'C' SDK (BU-69092S0). Steps described are for Visual Studio 2019 but may be followed for other versions as well. The document is applicable to BU-69092S0 ver 4.9.x and above for building 64-bit versions of the sample code.

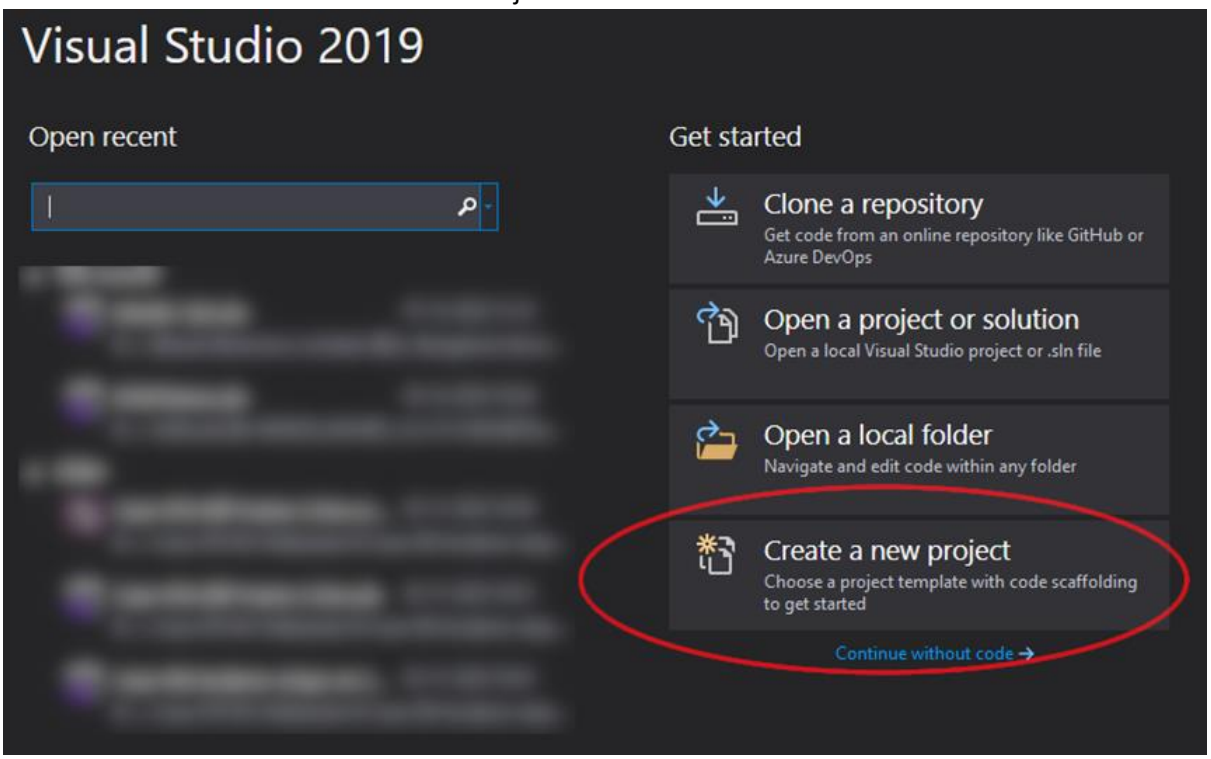
The same procedure may be followed to build your own AceXtreme 'C' SDK based applications as well.

Pre-requisites **:**

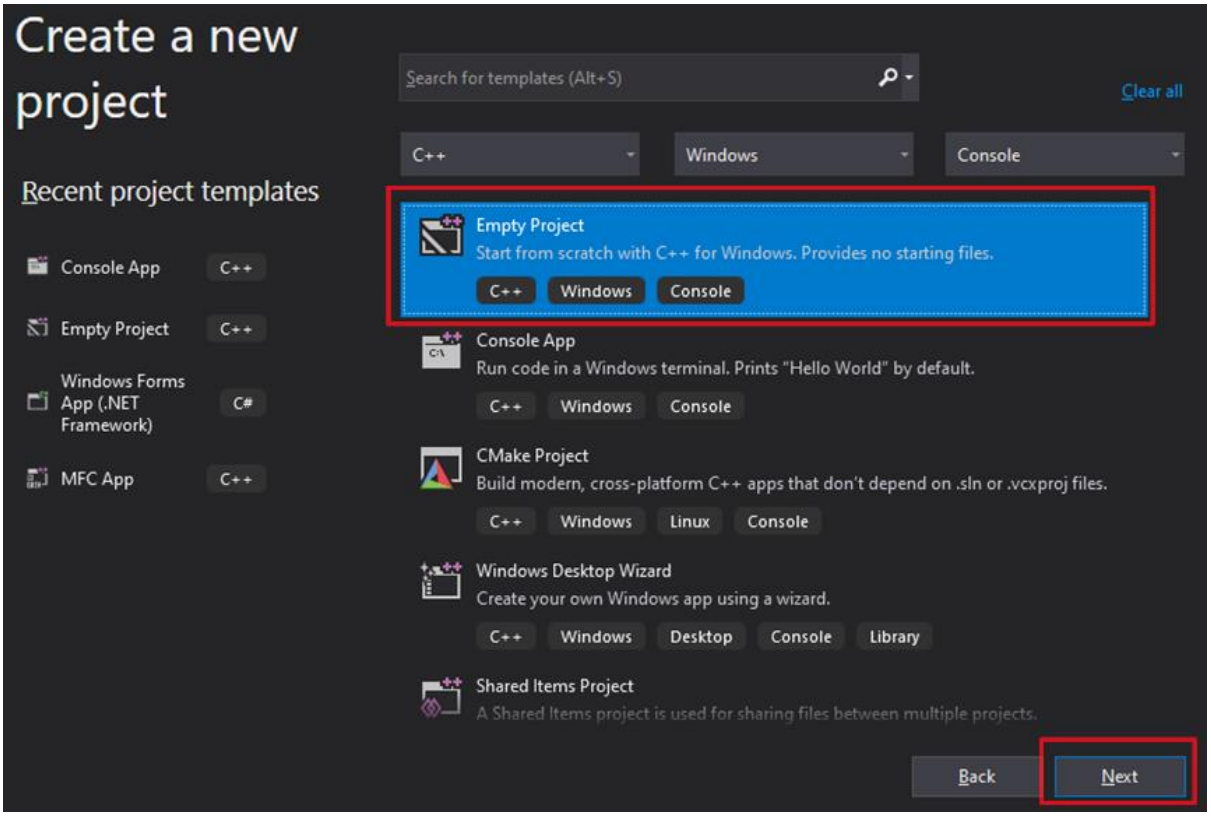
a. DDC's AceXtreme 'C' SDK for Windows must be installed. The P/N is BU-69092S0-110
b. Visual Studio 2019 is installed.
c. Although the steps described are in detail, familiarity in using Visual Studio will greatly help.

Procedure **:**

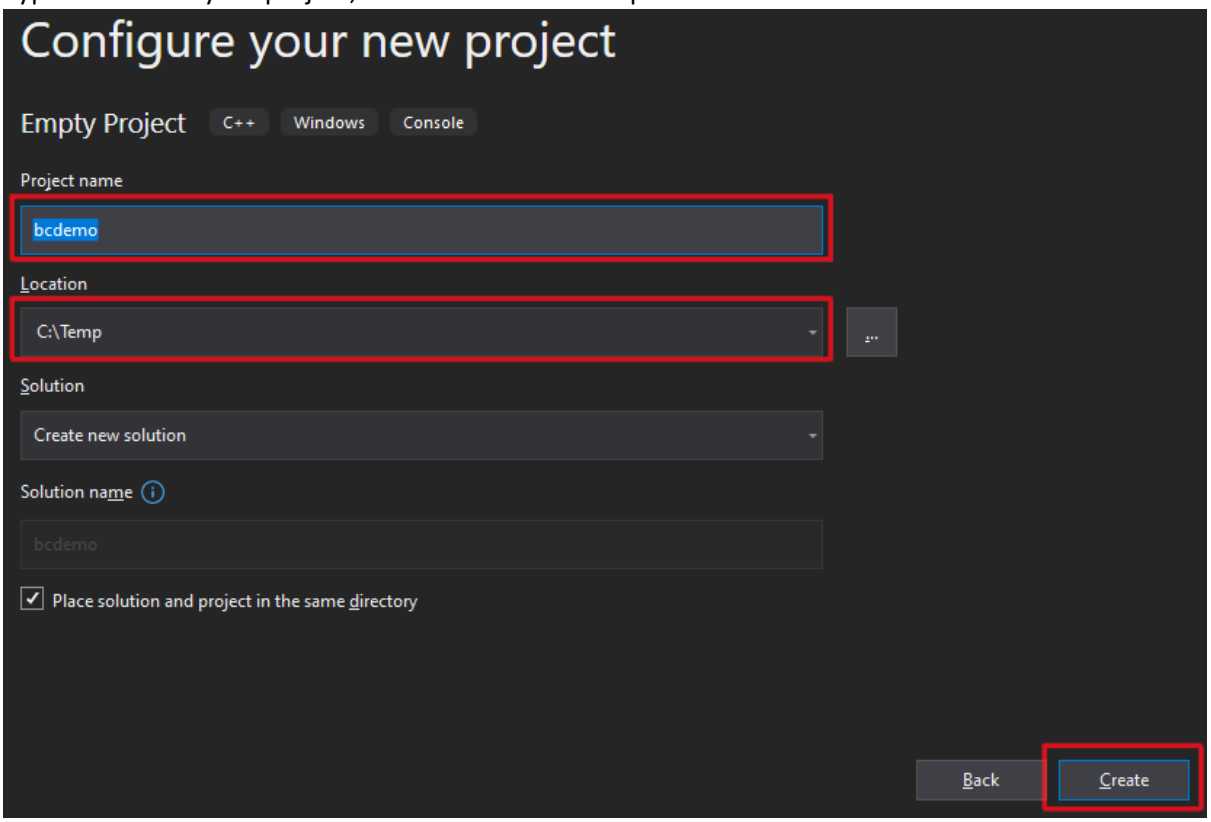
1) Start Visual Studio. Select "Create New Project".



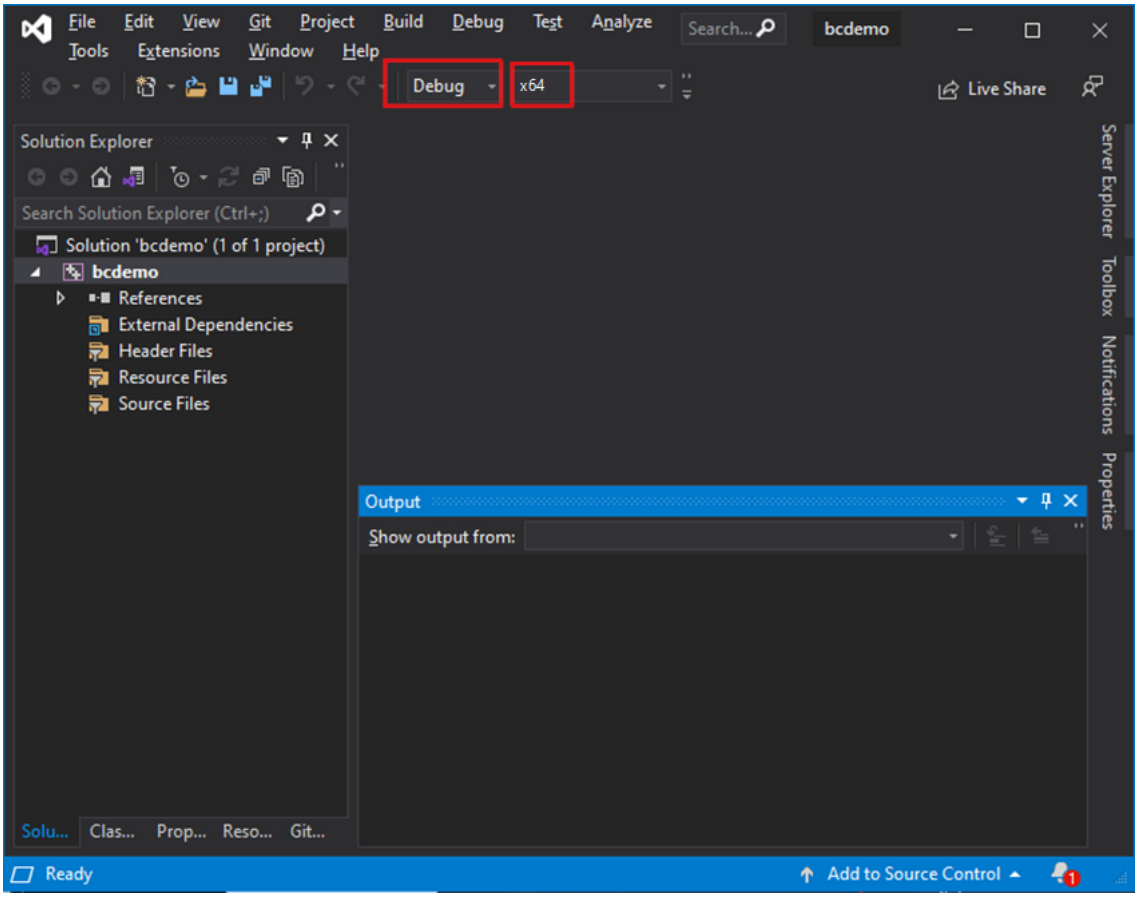
2) Select "Empty Project" and press "Next"



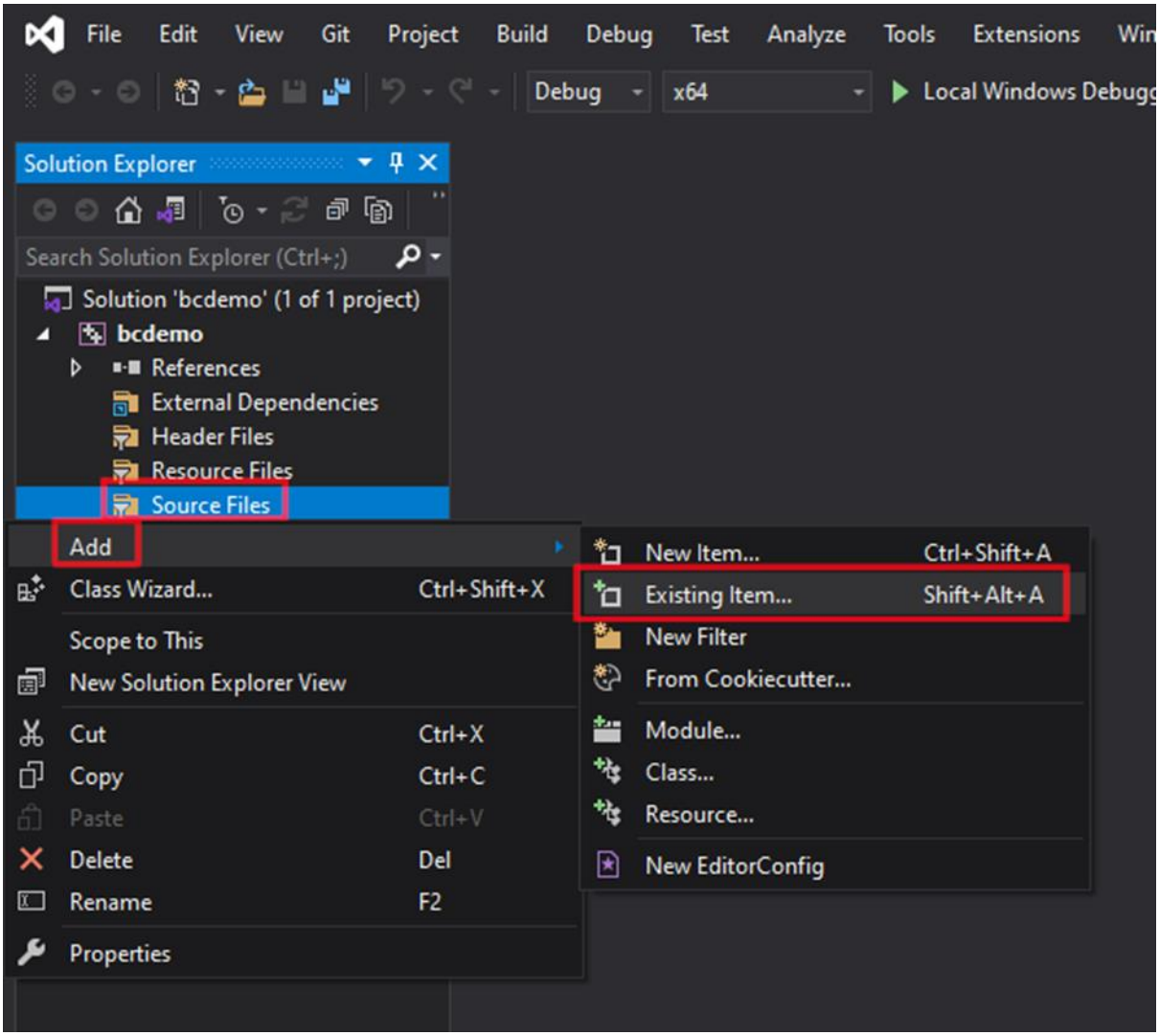
3) Type a name for your project, choose a location and press "Create".



4) The project is created as below. Change the configuration to "Debug" and platform to "x64".



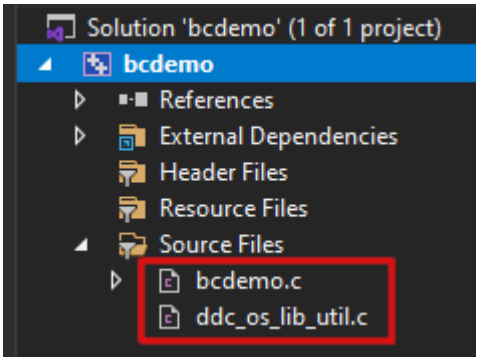
5) You may now add the sample .c file to the project. To do this right click on "Source Files", choose "Add" and "Existing Item…".



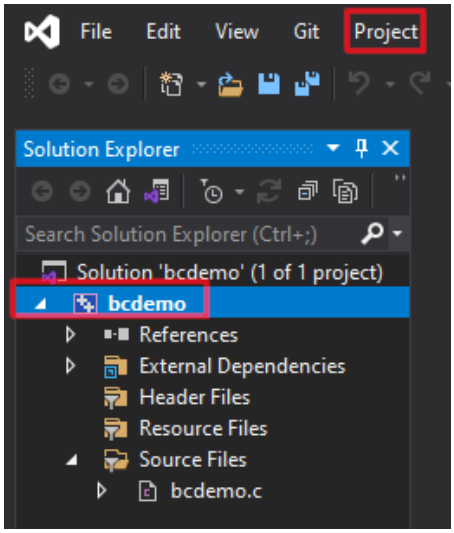
6) Browse to the folder containing the sample and choose the .c file.

7) Similarly, add the following file to the project "C:\DDC\aceXtremeSDKv4.9.5\drivers\acex\src\os\lib_support\ddc_os_lib_util.c". This file is needed for DDC samples which call some functions like DDCPressAKey() etc. If these functions are not used in your application, this step may be skipped.

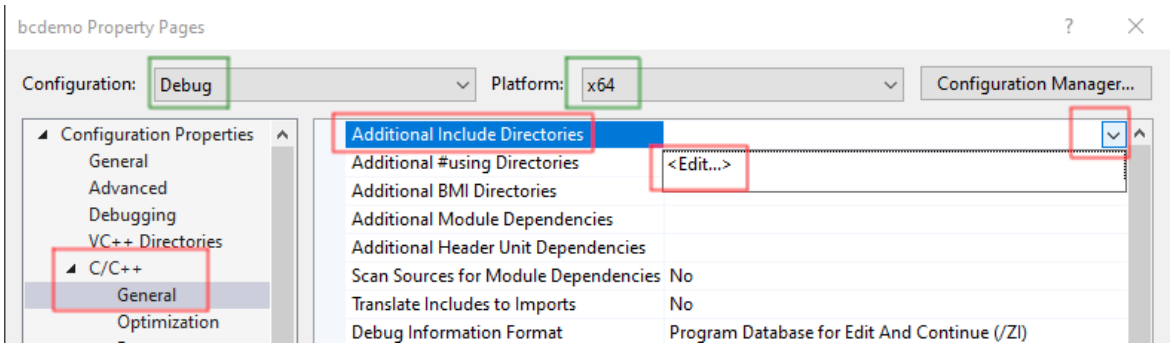
8) The added files appear as below.



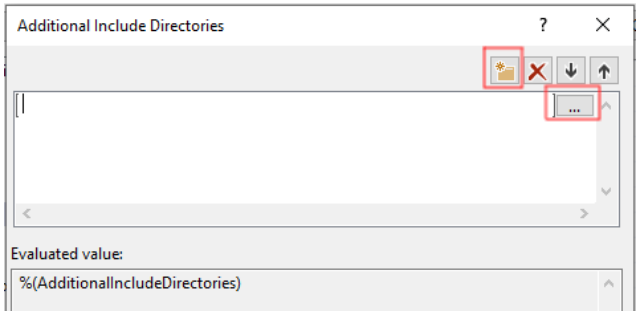
9) Open Project Properties by clicking on project name and then choosing menu "Project->Properties" or pressing Alt+F7.



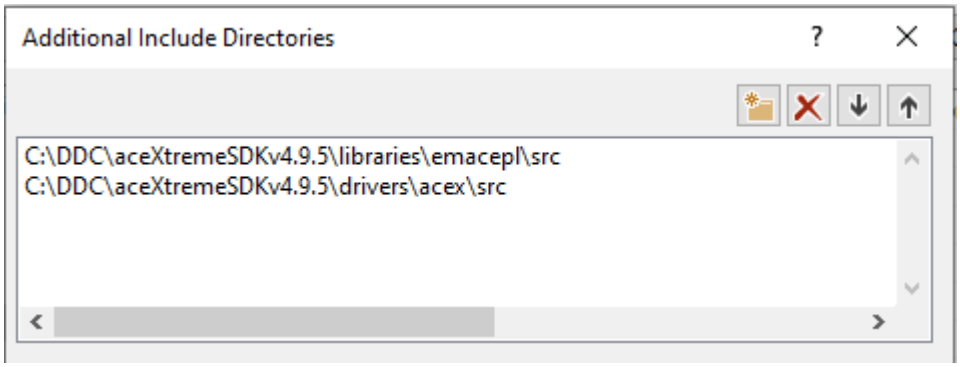
10) In Project Properties select Configuration as Debug, Platform as x64, click on "C/C++->General" and under "Additional Include Directories" click the down arrow and then "Edit".



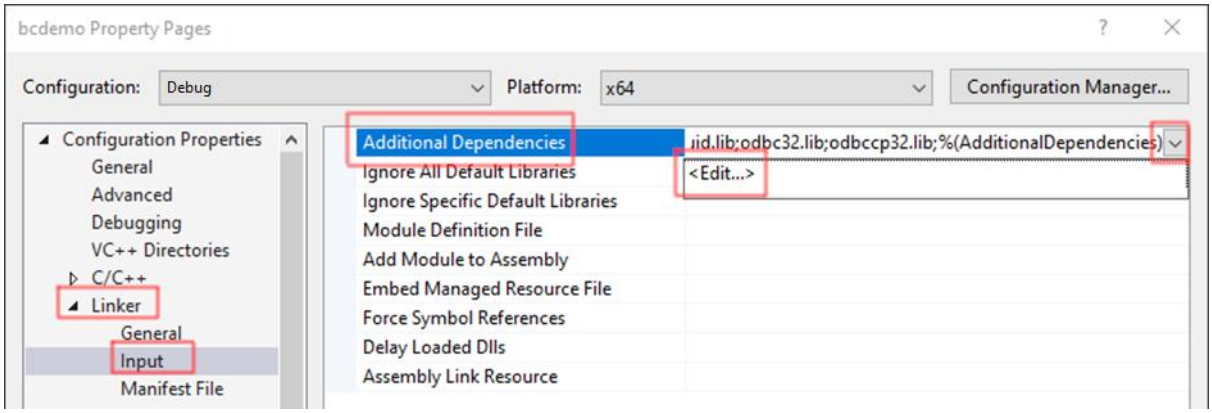
11) The "Additional Include Directories" pop up appears. Here click the folder icon ("New Line") and the *browse* button. A "Select Directory" popup appears. Browse and select the following folder (note replace *C:\DDC* with the location at which the AceXtreme SDK was installed and *aceXtremeSDKv4.9.5* with the name of your AceXtreme SDK folder:
C:\DDC\aceXtremeSDKv4.9.5\libraries\emacepl\src



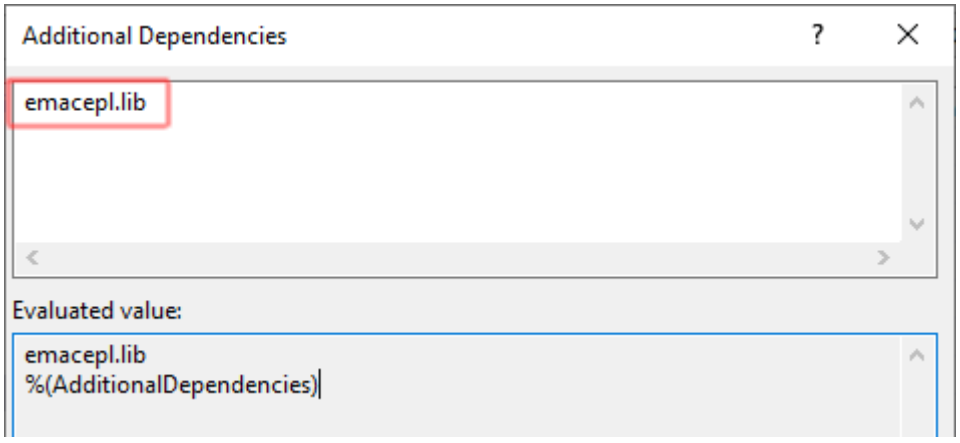
12) Repeat above step to add *C:\DDC\aceXtremeSDKv4.9.5\drivers\acex\src.* The two added directories appear as below.



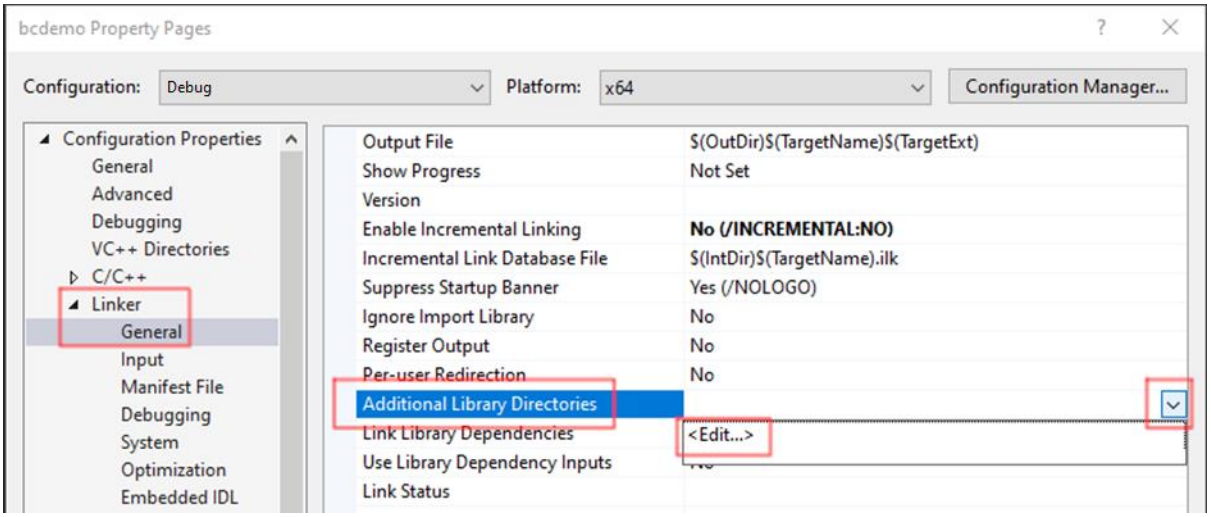
13) In Project Properties Window select "Configuration Properties->Linker->Input". Under "Additional Dependencies" click the drop-down arrow and click "Edit"



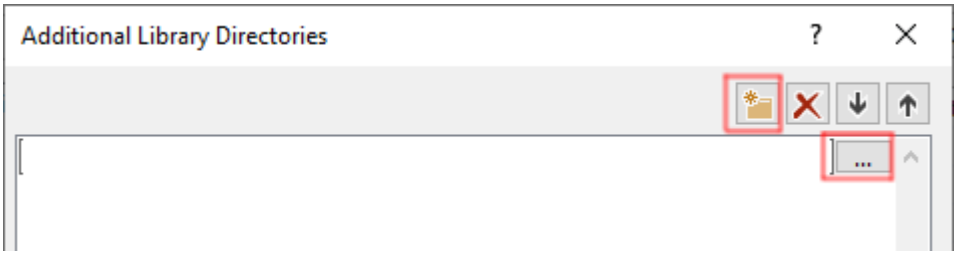
14) This brings up the "Additional Dependencies" popup. Enter "emacepl.lib" and click OK.



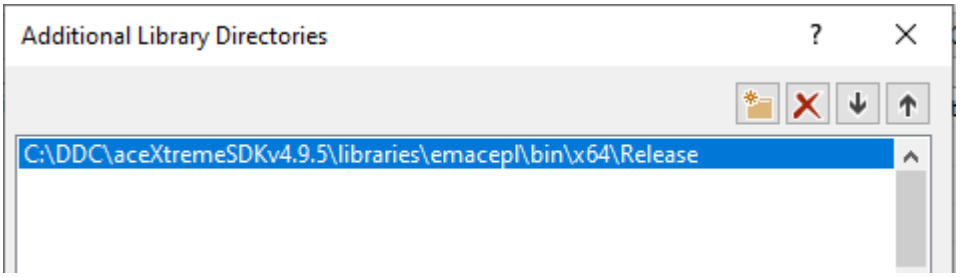
15) In Project Properties Window select "Configuration Properties->Linker->General". Under "Additional Library Directories" click the drop-down arrow and click "Edit".



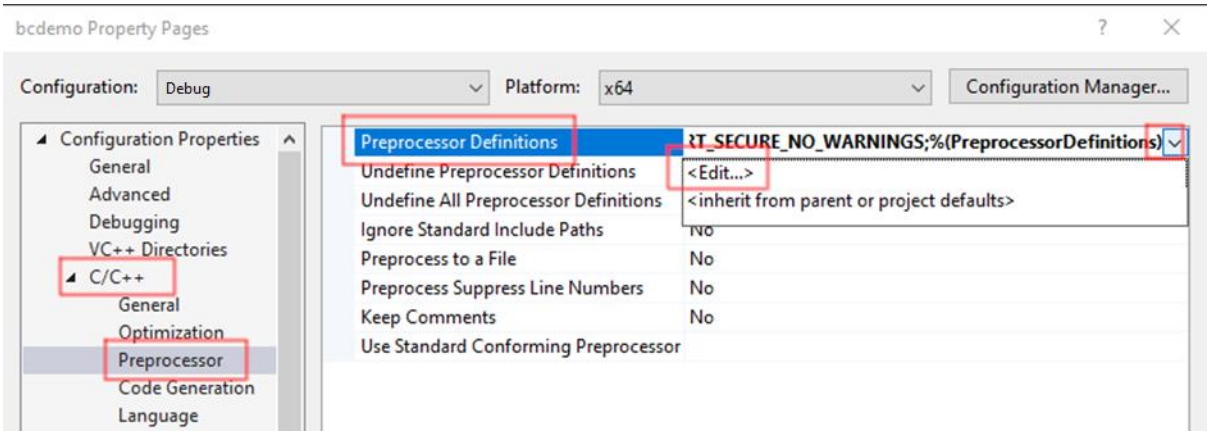
16) The "Additional Library Directories" pop up appears. Here click the folder icon ("New Line") and the *browse* button. A "Select Directory" popup appears. Browse and select the following folder (note replace *C:\DDC* with the location at which the AceXtreme SDK was installed and *aceXtremeSDKv4.9.5* with the name of your AceXtreme SDK folder:
C:\DDC\aceXtremeSDKv4.9.5\libraries\emacepl\bin\x64\Release



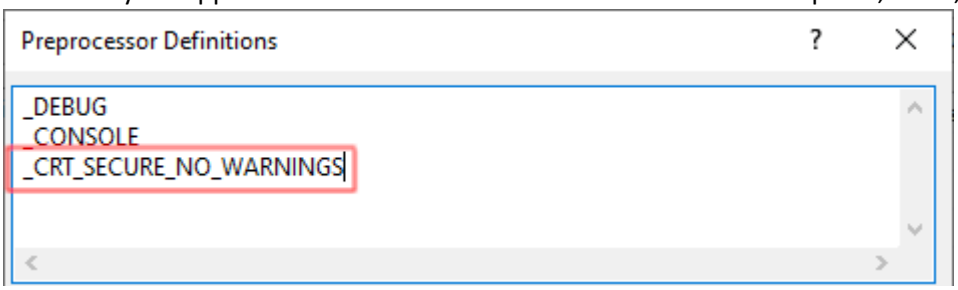
17) After folder selection above, the "Additional Library Directories" will appear as below. Click OK to close this popup.



18) In Project Properties dialog box, select C/C++ -> Preprocessor. Under "Preprocessor Definitions" click the drop-down arrow and click "Edit".



19) The "Preprocessor Definitions" pop up appears. In the last line add a new entry with text "_CRT_SECURE_NO_WARNINGS" as below and click OK to close the popup. This step is only needed if your application used unsafe versions of 'C' functions like printf, scanf, sprintf etc.



20) This completes all settings to be made in Project Properties. In the Project Properties dialog box press "Apply" and then OK to close it.

21) In *ddc_os_lib_util.c* add the following lines after `#include "ddc_os_lib_util.h"`. This step may be skipped if *ddc_os_lib_util.c* is not added into your project.

```
#define kbhit _kbhit
#define getch _getch
```

22) Build the project (F7) and verify that there are no errors.