Introduction

Something has gone wrong while using a TI compiler. You need to submit a test case so TI can reproduce the problem. This article shows you how to do that.

If you experience a problem while linking, this article does not apply to you. A different (not yet written) article addresses that situation.

In Video Form

For those who prefer, this video (https://www.youtube.com/watch?v=d-hB8i534C0) shows the typical way CCS users submit a compiler test case.

What Must be Submitted

1. Preprocessed source file
2. Compiler Version
3. Compiler Options Used

This page is mostly about creating and handling the preprocessed source file. Please do not forget the compiler version and options used. That information must be supplied, or the problem cannot be addressed.

Preprocess the Source File

The problem source file probably includes many other header files. Preprocessing means you don't have to manually track down those header files (which often include yet other header files) and somehow submit them too. After preprocessing, there is only one file to submit.

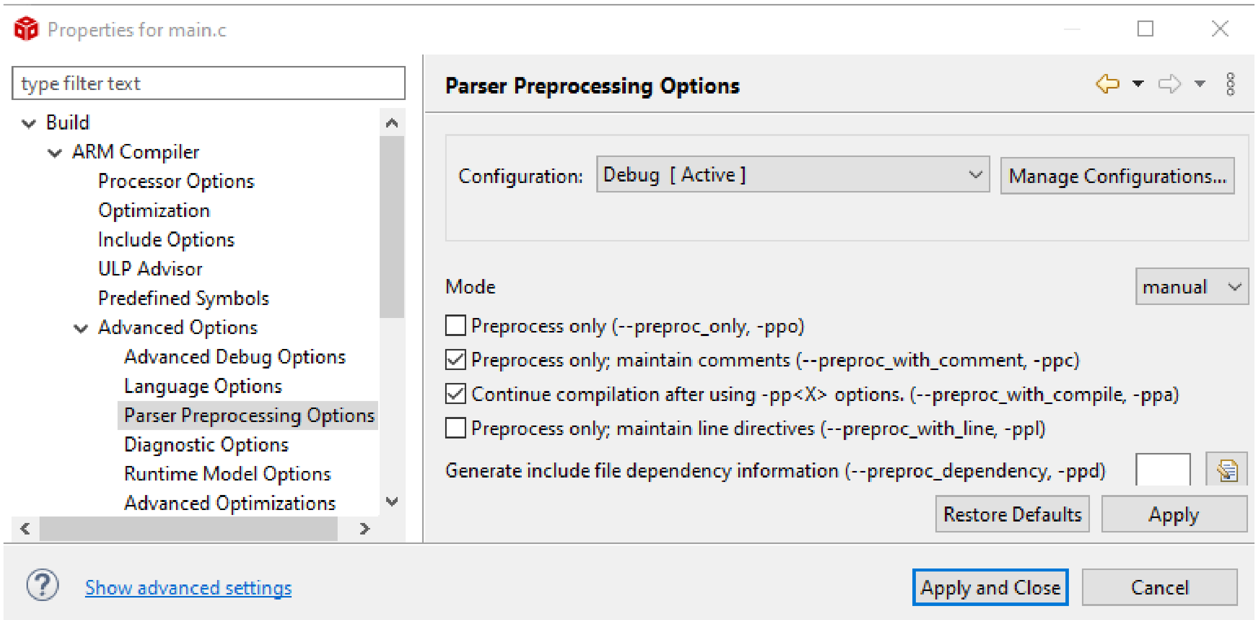
The exact details for creating a preprocessed file depend on how you do your build. This article discusses two different methods. One method uses CCS. The other uses a simple makefile. Use these descriptions only as a general guide, and not a set of precise instructions.

CCS Method

These directions presume the problem source file is part of a project you build within CCS. These directions are mostly accurate for CCS versions 6.x and higher. These particular directions were tested with CCS version 10.1.

- From the Project Explorer view, right-click on the file name and select **Show Build Settings ...**

- Navigate the directory like path on the left to something like **Build/ARM Compiler/Advanced Options**
- Highlight **Parser Preprocessing Options**
- Change the **Mode** drop-down box from **automatic (default)** to **manual**
- Check the box for **Preprocess only; maintain comments**
- Check the box for **Continue compilation after using -pp<X> options**
- Click **Apply and Close**



Parser Preprocessing Options

Configuration: Debug [Active]

Mode: manual

☐ Preprocess only (--preproc_only, -ppo)

☑ Preprocess only; maintain comments (--preproc_with_comment, -ppc)

☑ Continue compilation after using -pp<X> options. (--preproc_with_compile, -ppa)

☐ Preprocess only; maintain line directives (--preproc_with_line, -ppl)

Generate include file dependency information (--preproc_dependency, -ppd)

Now you are ready to build the file.

- Right-click on the file name and select **Build Selected File(s)**

A preprocessed file is created with the same base name as the source file, and the extension changed to .pp. For example, my_source.pp. This file is found in the same directory as the original source file. This is the file to send to TI. It is an ordinary text file. Don't be alarmed if it starts with many blank lines. That is not unusual. Note you can use the usual Windows operations to copy and paste this file from CCS.

You might see a .pp file in another directory, like Debug or Release. Do **not** send in that file. It is an automatically generated file used by CCS. Be sure you send in the .pp file found in the same directory as the source file.

Feel free to leave the above build option changes in place, or undo them.

Makefile Method

This section presumes you build with a makefile, and you have identified a single source file as the problem.

Start with this simple makefile. Presume iup.c is the problem file.

```
#-----------------------------------------------------------------------------
# Name the object files
#-----------------------------------------------------------------------------
OBJS := icdct.obj isbt.obj iup.obj iwinm.obj mhead.obj towave.obj wavep.obj

#-----------------------------------------------------------------------------
# Develop C_OPTS: The compiler build options
#-----------------------------------------------------------------------------
# Optimization level 2
C_OPTS := $(C_OPTS) --opt_level=2
# Build for C6600 processors
C_OPTS := $(C_OPTS) -mv6600
# Extra SW pipeline info
C_OPTS := $(C_OPTS) --debug_software_pipeline
# See C source like comments in assembly output
C_OPTS := $(C_OPTS) --src_interlist

#-----------------------------------------------------------------------------
# Link build rule
#-----------------------------------------------------------------------------
mp3.out : $(OBJS)
    cl6x -z $(OBJS) -o=$@ lnk.cmd

#-----------------------------------------------------------------------------
# Compile build rule
#-----------------------------------------------------------------------------
%.obj : %.c
    cl6x $(C_OPTS) $<
```

The first step is to add two options for preprocessing. Add the following lines to the C_OPTS part of the makefile.

```
# Preprocessing which preserves comments
C_OPTS := $(C_OPTS) --preproc_with_comment
# Compilation continues normally after preprocessing
C_OPTS := $(C_OPTS) --preproc_with_compile
```

Next, rebuild only iup.c. Here is one way to do that.

```
% rm iup.obj
% gmake iup.obj
cl6x --opt_level=2 -mv6600 --debug_software_pipeline --src_interlist --preproc_with_comment --preproc_with_compile iup.c
```

The file iup.pp appears in the current directory. It is an ordinary text file. Don't be alarmed if your .pp file starts with many blank lines. That is not unusual. Note iup.pp is the only file you need to send to TI.

Attach Preprocessed File to a Forum Post

The most common way to notify TI about a compiler problem is by posting to the E2E forum (http://e2e.ti.com). The typical pattern is to post an initial description of the problem. TI responds by requesting a test case, with a pointer to this page. You prepare the test case, and now it is time for the next step. Reply to the last post on the forum thread about your problem. Use the paper clip icon to attach the preprocessed file. Note the forum only accepts a very small set of file types, and .pp is not one of them. So, add the extension .txt to the file. This forms a file name like my_source.pp.txt.

Protecting Intellectual Property

This section discusses ways to improve (not guarantee) protection of your source code.

The methods used above preserve the comments in the source code. If you use --preproc_only instead of --preproc_with_comments, then comments do not appear in the .pp file.

Rather than attaching the code to a forum post, you can send it privately to the TI employee who requested the test case. While on the forum, hover your mouse over the screen name or avatar of the TI employee. A box will pop up. Click on **Send a private message**. In the message compose interface which comes up, use the paper clip icon to attach the preprocessed file. Just like when attaching to a forum post, you need to add the file extension .txt to it.

Compiler Version

When you send the preprocessed file to TI, be sure to also indicate the version of the compiler used. Note this is **not** the same as the version of CCS.

This short video (https://www.youtube.com/watch?v=vnetm9247y0) shows one way to see the compiler version.

Another way is to inspect the linker map file. The version is shown in the first few lines. In a CCS project, in the Debug or Release folder (or whatever the name of the build configuration is), you will find the file name of project.map. Double click to open it and see the version number.

Compiler Options Used

In addition to the preprocessed file and the compiler version, also show the build options used, exactly as they are seen by the compiler. Always copy-and-paste the text of the options. Do not use a screen shot.

If you build with CCS, copy-and-paste the build command for the problem source file from the Console view. If requested, or if it is easier, supply the full contents of the Console view. Use the **Copy Build Log** icon to save everything to a text file. Be sure to use the file extension

.txt. Then attach that file to your next post the same as you attach the preprocessed file.

Resources

TI Code Composer Studio Product Page (https://www.ti.com/tool/CCStudio)
Related Technical Documents (https://software-dl.ti.com/ccs/esd/documents/ccs_documentation-overview.html)
TI E2E Technical Forums (https://e2e.ti.com)