



# ***TI Compiler Qualification Kit***

## **Coverage Testing User's Guide**

---

### **1. Introduction**

Application of the TI Compiler Qualification Kit (QKIT) may require generating coverage data in order to verify that all portions of the compiler tool chain used for compiling application code are also exercised by validation testing. TI C/C++ Compilers with instrumentation for collecting coverage data can be downloaded from the same tool folder where your QKIT installer was obtained.

NOTE: The data collection files you send TI (\*.cov files) only include coverage data for the compiler. No coverage data is generated or collected for the application code.

### **2. Coverage Testing**

Process flow for collecting coverage data is as follows.

Download the instrumented compiler for your compiler version from the same tool folder where your QKIT installer was obtained.

Unzip the "bincov" folder into a temporary location. This folder contains instrumented versions of all compiler tool chain executables along with coverage data collection files (\*.cov files) for each executable.

NOTE: Some executables will not have an associated \*.cov file. These executables are based off the LLVM compiler infrastructure and will collectively gather coverage information in the llvm.cov or armlvm.cov data collection file.

Rename the original compiler installation folder from "bin" to "bin.old".

Create a new "bin" folder within your compiler installation folder and copy in the executables and \*.cov coverage files from the "bincov" folder.

NOTE: Reset coverage data collection by copying in the \*.cov files from the original "bincov" folder.

NOTE: The directory of the instrumented compiler tools must be writable. An instrumented compiler will contain a corresponding .cov file for every compiler executable. As the instrumented compiler executes for coverage, data will be written to these .cov files.

Set the environment variable COVFILELIST. This is a comma-separated list of the paths to all the compiler executable .cov files.

NOTE: For Windows environment variable limitations see:

<https://www.bullseye.com/help/env-windows.html>

### Commandline:

Using cmd.exe and command line execution of the compiler, then use below to set COVFILELIST within your existing cmd.exe console:

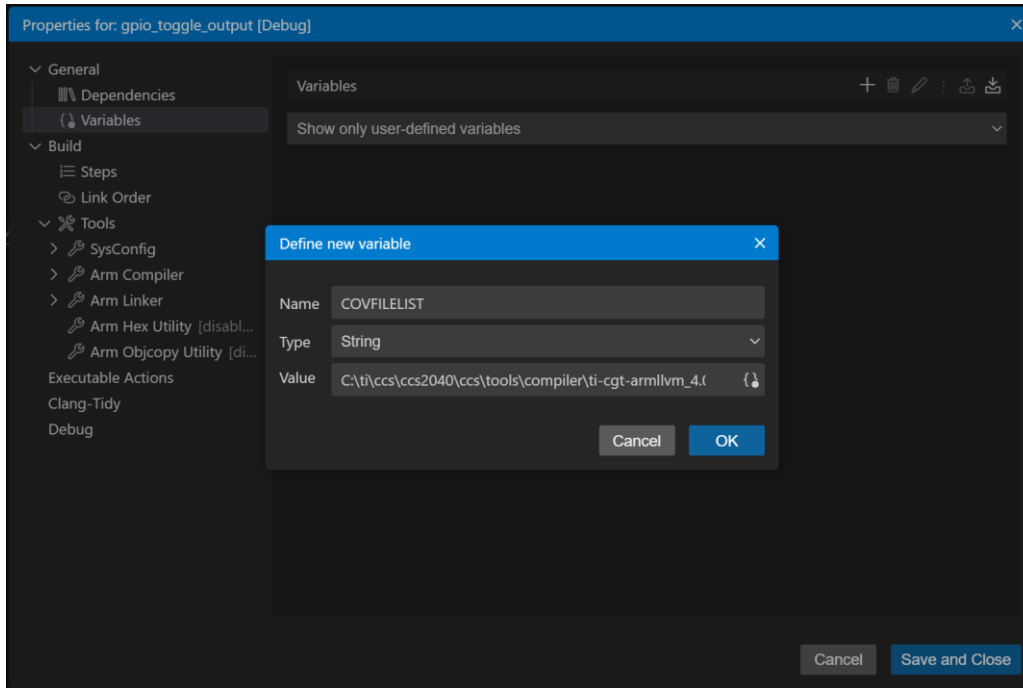
```
set
COVFILELIST=C:\ti\ccs\ccs2040\ccs\tools\compiler\ti-cgt-
armllvm_4.0.4.LTS\bin\armllvm.cov,C:\ti\ccs\ccs2040\ccs\tools\compiler\ti-cgt-
armllvm_4.0.4.LTS\bin\tiarmasm.cov,C:\ti\ccs\ccs2040\ccs\tools\compiler\ti-cgt-
armllvm_4.0.4.LTS\bin\tiarmdis.cov,C:\ti\ccs\ccs2040\ccs\tools\compiler\ti-cgt-
armllvm_4.0.4.LTS\bin\tiarmhex.cov,C:\ti\ccs\ccs2040\ccs\tools\compiler\ti-cgt-
armllvm_4.0.4.LTS\bin\tiarmlibinfo.cov,C:\ti\ccs\ccs2040\ccs\tools\compiler\ti-cgt-
armllvm_4.0.4.LTS\bin\tiarmlnk.cov,C:\ti\ccs\ccs2040\ccs\tools\compiler\ti-cgt-
armllvm_4.0.4.LTS\bin\tiarmofd.cov,C:\ti\ccs\ccs2040\ccs\tools\compiler\ti-cgt-
armllvm_4.0.4.LTS\bin\tiarmstrip.cov
```

### CCS 20.4 and later:

To set at the Project Level (Specific Project Only):

- Right-click your project in the Project Explorer and select Properties.
- Navigate to General → Variables.
- Click the “+” Add button and add entry for COVFILELIST as shown below.

NOTE: use the file names for \*.cov files in bin folder.



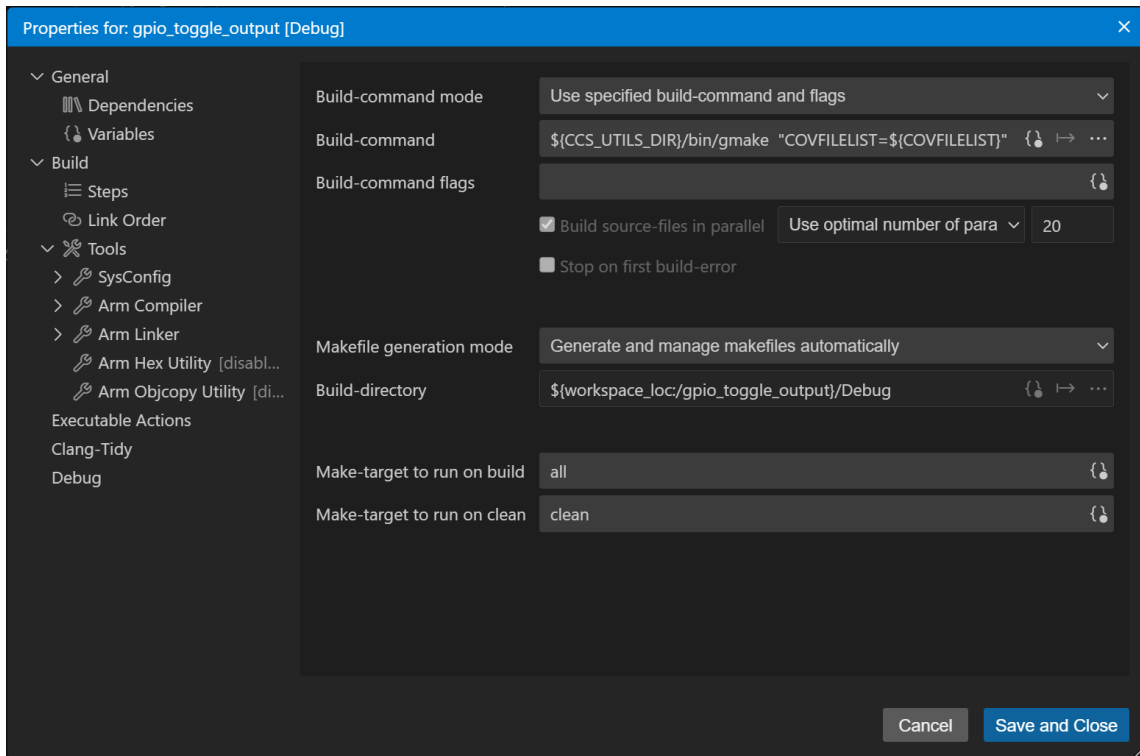
- Click Save and Close.

Next, update the Build-command to use above setting:

- Right-click your project in the Project Explorer and select Properties.
- Navigate to Build → Variables.
 

Build-command mode	Use default build-command and common flags
Build-command	<code>\${CCS_UTILS_DIR}/bin/gmake</code>
- Update to below settings:
 

Build-command mode	Use specified build-command and flags
Build-command	<code>\${CCS_UTILS_DIR}/bin/gmake "COVFILELIST=\${COVFILELIST}"</code>

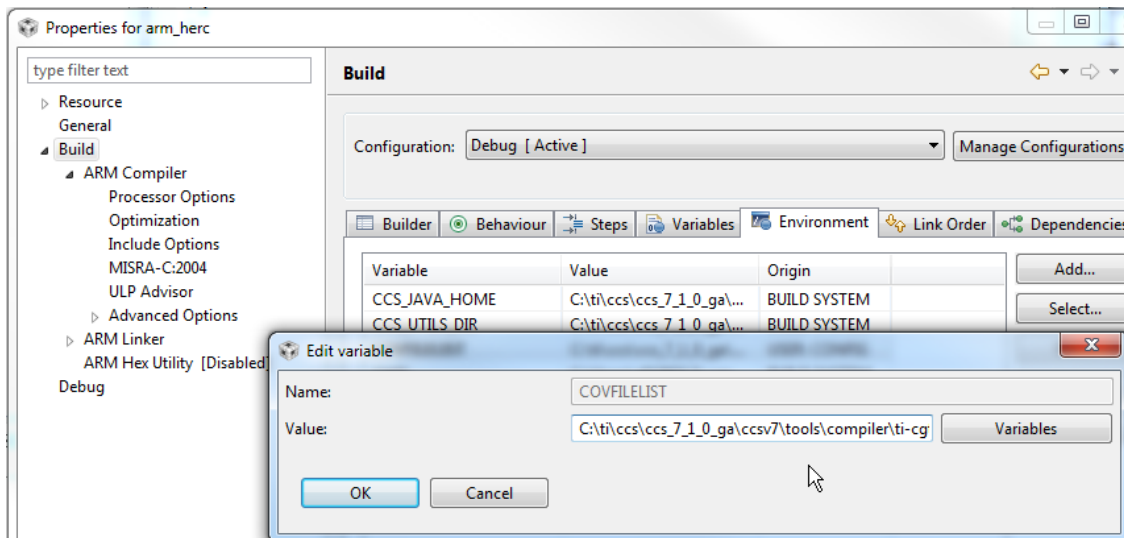


- Click on tab for “Save and Close”

### CCS 12 and earlier:

If using CCS 12, then use below steps to set COVFILELIST:

- Open “Project properties”
- Click on “Build”
- Click on tab for “Environment”
- Click on “Add” and add COVFILELIST



Build your application. Note that the time stamps for \*.cov files update each time the instrumented executables are exercised.

Once coverage data collection is complete, rename the "bin" folder to "bin.cov" and rename the original "bin.old" folder to "bin" to restore the original un-instrumented compiler executables.

After building your application with coverage, email the following to TI for analysis:

- zip file containing all \*.cov files
- include list of compiler and linker options used for compiling your application
- email to: safeti\_cqkit\_data@ti.com

TI will perform a comparison of function coverage for your submitted coverage data files as compared to coverage data files collected during compiler release validations. TI will send a summary report.