Contents

[1 References 3](#_Toc456873572)

[2 ARM Cortex R4 4](#_Toc456873573)

[2.1 Make Process 4](#_Toc456873574)

[2.2 C/C++ Build 4](#_Toc456873575)

[2.2.1 ARM/Thumb Mode Compile Settings 4](#_Toc456873576)

[2.2.2 File Optimization Settings 5](#_Toc456873577)

[2.2.3 Excluded Files 8](#_Toc456873578)

[2.2.3.1 Release Build Configuration 8](#_Toc456873579)

[2.2.3.2 Fault Injection Build Configuration 9](#_Toc456873580)

[2.2.4 Project Properties 11](#_Toc456873581)

[2.2.4.1 Resources 11](#_Toc456873582)

[2.2.4.1.1 Linked Resources 11](#_Toc456873583)

[2.2.4.1.1.1 Path Variables 11](#_Toc456873584)

[2.2.4.1.1.2 Linked Resources 12](#_Toc456873585)

[2.2.4.1.2 Resource Filters 12](#_Toc456873586)

[2.2.4.2 General 13](#_Toc456873587)

[2.2.4.3 Build 14](#_Toc456873588)

[2.2.4.3.1 ARM Compiler 17](#_Toc456873589)

[2.2.4.3.1.1 Processor Options 17](#_Toc456873590)

[2.2.4.3.1.2 Optimization Options 18](#_Toc456873591)

[2.2.4.3.1.3 Debug Options 19](#_Toc456873592)

[2.2.4.3.1.4 Include Options 20](#_Toc456873593)

[2.2.4.3.1.5 MISRA-C 2004 20](#_Toc456873594)

[2.2.4.3.1.6 Advanced Options 21](#_Toc456873595)

[2.2.4.3.1.6.1 Language Options 21](#_Toc456873596)

[2.2.4.3.1.6.2 Parser Preprocessing Options 23](#_Toc456873597)

[2.2.4.3.1.6.3 Predefined Symbols 23](#_Toc456873598)

[2.2.4.3.1.6.4 Diagnostic Options 24](#_Toc456873599)

[2.2.4.3.1.6.5 Runtime Model Options 26](#_Toc456873600)

[2.2.4.3.1.6.6 Advanced Optimizations 28](#_Toc456873601)

[2.2.4.3.1.6.7 Entry/Exit Hook Options 29](#_Toc456873602)

[2.2.4.3.1.6.8 Library Function Assumptions 30](#_Toc456873603)

[2.2.4.3.1.6.9 Assembler Options 30](#_Toc456873604)

[2.2.4.3.1.6.10 File Type Specifier 32](#_Toc456873605)

[2.2.4.3.1.6.11 Directory Specifier 32](#_Toc456873606)

[2.2.4.3.1.6.12 Default File Extensions 33](#_Toc456873607)

[2.2.4.3.1.6.13 Command Files 34](#_Toc456873608)

[2.2.4.3.2 ARM Linker 35](#_Toc456873609)

[2.2.4.3.2.1 Basic Options 35](#_Toc456873610)

[2.2.4.3.2.2 File Search Path 36](#_Toc456873611)

[2.2.4.3.2.3 Advanced Options 37](#_Toc456873612)

[2.2.4.3.2.3.1 Command File Preprocessing 37](#_Toc456873613)

[2.2.4.3.2.3.2 Diagnostics 37](#_Toc456873614)

[2.2.4.3.2.3.3 Linker Output 39](#_Toc456873615)

[2.2.4.3.2.3.4 Symbol Management 40](#_Toc456873616)

[2.2.4.3.2.3.5 Runtime Environment 41](#_Toc456873617)

[2.2.4.3.2.3.6 Link Optimization 42](#_Toc456873618)

[2.2.4.3.2.3.7 Miscellaneous 42](#_Toc456873619)

[2.2.4.3.2.3.8 Dynamic Linking Support Options 43](#_Toc456873620)

[2.2.5 Post Build 45](#_Toc456873621)

[3 NHET 46](#_Toc456873622)

[3.1 Make Process 46](#_Toc456873623)

[3.2 Tool Settings 46](#_Toc456873624)

[3.2.1 Assembler 46](#_Toc456873625)

[4 Appendix 47](#_Toc456873626)

[4.1 Debug Output Files 47](#_Toc456873627)

[5 Revision Control Log 48](#_Toc456873628)

# References

The following references were used in assistance with the creation of this document and the rationale for some of the configuration settings.

1. ARM Optimizing C/C++ Compiler v4.6 (spnu151f.pdf)
2. Cortex-R4 and Cortex-R4F Revision r1p3 Technical Reference Manual (DDI0363E\_cortexr4\_r1p3\_trm.pdf)
3. Texas Instruments Assembly Language Tools Enhanced High-End Timer (NHET) Assembler User’s Guide (spnu490.pdf)
4. AUTOSAR realtime operating system MICROSAR OS TMS570 SafeContext Technical Reference v5.00 (TechnicalReference\_MicrosarOS\_TMS570SafeContext.pdf)
5. ARM Assembly Language Tools (spnu118j.pdf)

# ARM Cortex R4

## Make Process

This project uses a generated Makefile to perform the build process. The makefile generator included with the CCSv5 installation is used to generate the Makefile for the build.

## C/C++ Build

### ARM/Thumb Mode Compile Settings

All files compiled in this project are compiled in thumb mode with the exception of the files listed in Table 1: ARM Mode Files.

|  |  |
| --- | --- |
| **File** | **Rationale** |
| TMS570\_Startup\src | |
| AppStartup.c | This file contains \_c\_init00 which is vectored to by the reset vector. At the moment, it is understood that every function directly entered by a vector from the vector table should be compiled in ARM mode. |
| sys\_core.asm | The .arm directive in the source is a preprocessor to compile in ARM mode. |
| sys\_memory.asm | The .arm directive in the source is a preprocessor to compile in ARM mode. |
| fiqintvect.asm | ISR reference to Isr\_ESHM, which is in ARM mode. |
| TMS570\_uDiag\src | |
| Cd\_uDiagESM.c | Contains Isr\_ESHM which is a category 1 interrupt and §8.3 of [4] states “Note: Category 1 interrupt functions must always be compiled for 32bit ARM mode!” |
| Cd\_uDiagVIM.c | Present configuration. No real need for this to be in ARM mode. |
| dabort.asm | The .arm directive in the source is a preprocessor to compile in ARM mode. |
| pabort.asm | The .arm directive in the source is a preprocessor to compile in ARM mode. |
| undefinst.asm | The .arm directive in the source is a preprocessor to compile in ARM mode. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\Os | |
| osekasm.asm | §11 of [4] requires all Os assembly files to be compiled in ARM mode |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\VStdLib | |
| vstdlib.c | Unknown where requirement is for ARM mode compilation. It fails on the inline assembly when compiled in Thumb mode. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\Wdg | |
| Wdg\_TMS570LS3x.c | Present configuration. No real need for this to be in ARM mode. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\WdgIf | |
| WdgIf.c | Present configuration. No real need for this to be in ARM mode. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\WdgM | |
| WdgM.c | Present configuration. No real need for this to be in ARM mode. |
| WdgM\_Checkpoint.c | Present configuration. No real need for this to be in ARM mode. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\CDD\BasicSysSrvc | |
| Interrupts.c | §6.7.2 of [1] requires all compiler intrinsic functions to be compiled in ARM mode. Interrupts.c is using compiler intrinsic calls to enable/disable FIQ. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\CDD\MtrCtrl | |
| MtrCtrl\_Irq.c | Motor control ISR is a category 1 interrupt and §8.3 of [4] states “Note: Category 1 interrupt functions must always be compiled for 32bit ARM mode!” |
| CmMtrCurr\src | |
| Sa\_CmMtrCurr.c | Present configuration. No real need for this to be in ARM mode. |
|  |  |

Table 1: ARM Mode Files

### File Optimization Settings

All files compiled in this project are compiled with *Optimization Level* set to two (2) and *Optimization for Speed* set to three (3). Files that do not use these default optimization levels are shown in Table 2: Optimized Files.

|  |  |  |  |
| --- | --- | --- | --- |
| **File** | **Optimization Level** | **Optimize for Speed** | **Rationale** |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\Os | | | |
| osekasm.asm | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\VStdLib | | | |
| vstdlib.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\WdgIf | | | |
| WdgIf.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\BSW\WdgM | | | |
| WdgM.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\CDD\BasicSysSrvc | | | |
| Interrupts.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| GM\_C1XX\_EPS\_TMS570\SwProject\Source\CDD\MtrCtrl | | | |
| MtrCtrl\_Irq.c | 3 | 5 | Motor control ISR is a frequently triggered ISR with a strict deadline, so fast processing is mandatory. |
| ePWM\src\ | | | |
| Cd\_Nhet1.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| ePWM.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| DigMSB\src\ | | | |
| Sa\_DigMSB.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| DigMSBSigCorr\src | | | |
| Sa\_DigMSBSigCorr.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| SVDrvr\_CM\src | | | |
| PwmCdd.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| MtrCtrl\_CM\src | | | |
| Ap\_PICurrCntrl.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| Ap\_TrqCanc.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| MtrVel\_Digi\src | | | |
| Sa\_MtrVel3.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| TMS570\_Startup\src | | | |
| sys\_core.asm | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| AppStartup.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| sys\_memory.asm | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |
| Dma/src/ | | | |
| Dma.c | 3 | 5 | Present configuration. No rationale for the optimized level or speed. |

Table 2: Optimized Files

### Excluded Files

The following tables show the files that are excluded from each build configuration that is configured in the project.

#### Release Build Configuration

|  |  |
| --- | --- |
| **File** | **Rationale** |
| TMS570\_Startup\src | |
| BootStartup.c | Not required by the application |
| sys\_startup.c | Not required by the application |
| sys\_pmu.asm | Not required by the application |

Table 3: Release Build Configuration Excluded Files

#### 

### Project Properties

The following sections describe the settings used within the project properties for the Resource, General, and Build sections.

The values listed in the “cfg” columns in the following sections refer to the following:

* **A** – Default for All build configurations
* **R** – The Release build configuration
* **D** – The Debug build configuration
  + Not Implemented – CPU use with all debug info enabled produces an executable not capable of meeting deadlines and is not capable of approximating a “Release” build system at this time.
* **U** – CPU Usage Metrics Build
  + Not implemented – Metrics builds are performed by running a set of scripts in the Tools folder located within the project. The script will add the files needed to perform a metrics build and is compiled using the release build configuration.
* **P** – The Profiling build configuration
  + Not Implemented TBD – according to compiler guide the debug info should not be enabled when profiling.
* **F** – Fault Injection build configuration

#### Resources

##### Linked Resources

The following resources are used within Code Composer Studio for linking to directories within the host system.

###### Path Variables

The path variables defined below specify locations in the file system. The variables can include other path variables with the syntax “${VAR}.”

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Cfg** | **Value** | **Rationale** |
| CCS\_BASE\_ROOT | A | << Install Path >> | Value is automatically determined. |
| CCS\_INSTAL\_ROOT | A | << Install Path >> | Value is automatically determined. |
| CG\_TOOL\_ROOT | A | << Install Path >> | Value is automatically determined. |
| ECLIPSE\_HOME | A | << Install Path >> | Value is automatically determined. |
| PARENT\_LOC | A | << Install Path >> | Value is automatically determined. |
| PROJECT\_LOC | A | << Install Path >> | Value is automatically determined. |
| WORKSPACE\_LOC | A | << Install Path >> | Value is automatically determined. |
| CM\_ROOT | A | ${PROJECT\_LOC}\..\.. | Defines the base folder for the project which contains all of the software components included for a build. The path variable is used with the linked resources to bring the components into the project. |

###### Linked Resources

The linked resources tab provides all the link paths within the project. The image below shows a project using a path variable for linked resources included in the project.

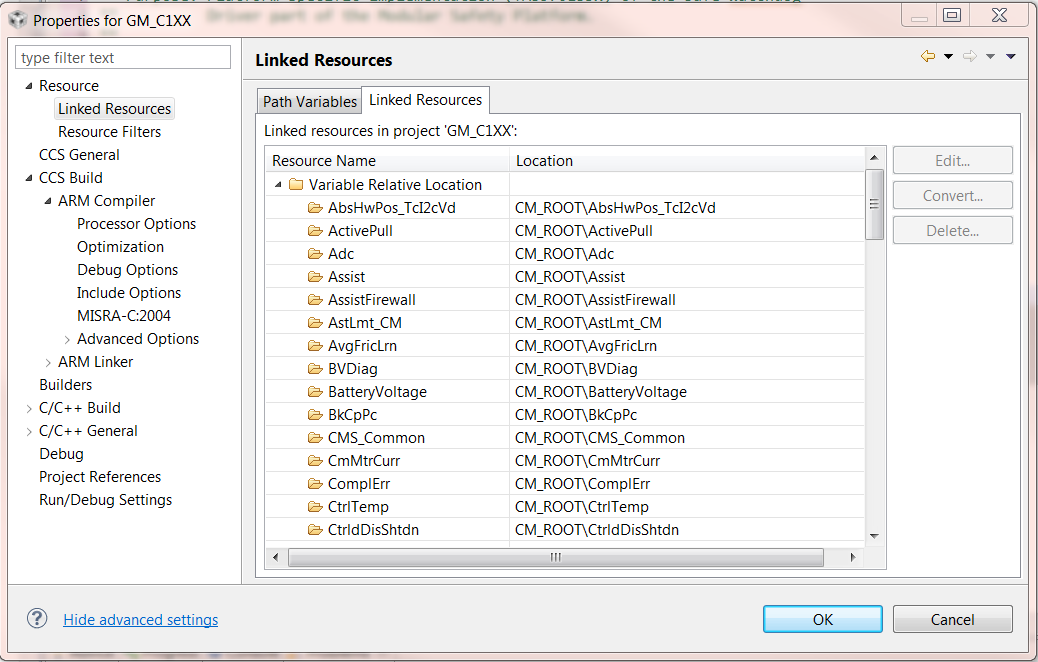


Figure 1: Linked Resources Defined in a Project

##### Resource Filters

The following resource filters are applied to the build environment.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Cfg** | **Value** | **Rationale** |
| Include Only: | | | |
| *[None]* |  |  |  |
| Exclude All: | | | |
| UTP Folders | A | *Filter Type:* Exclude All  *Applies To:* Folders and All children (recursive)  *File and Folder Attributes:* “Name matches utp” | Removes access to the UTP folders in each software component from Code Composer Studio’s view of the project tree. The UTP folders are used for unit testing and not used as part of the build. |

#### General

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Device | | | | |
| Variant | A | TMS570LS20216SPGE |  | This is the microprocessor selected for the C1xx Bev for EA3.0 |
| Advanced settings | | | | |
| Device endianness | A | be32 | -be32 | ARM is migrating future devices to big-endian, so big-endian is the desired setting for this project to be compatible with future devices. The option be32 is selected to provide a big-endian program with big-endian encoded instructions. |
| Compiler version | A | TI v4.9.5 |  | TI v4.9.5 contains all the features necessary to support EPS production programs at this time. |
| Output Format | A | eabi (ELF) | --abi=eabi | The “eabi” format option is required to support use of floating point. |
| Linker command file | A | TMS570LS20216SPGE |  | Linker file with all memory map necessary for linking the application |
| Runtime support library | A | rtsv7R4\_T\_be\_v3D16\_eabi.lib |  | Run-time support (rts) library that supports the following:  *v7R4*: ARM7 Cortex R4 core architecture *T*: Thumb mode compilation *be*: Big-endian library *v3D16*: VFP support for version 3d16 and the cortex R4 implements the VFPv3-D16 architecture per §1.2 of [2] *eabi*: Application binary interface (ELF) and is required for floating-point |

#### Build

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Builder | | | | |
| Use default build command | R | Not Selected |  | Custom build command defined in the next line. |
|  | F | Selected |  | Uses options defined in other settings to generate command. |
| Build Command | R | ${CCS\_UTILS\_DIR}/bin/gmake -j8 -s -j 8 | --j8 -s -j 8 | -s: Silent operation, do not print the commands as they are executed. This option was included to help save compile time by generating less output to the console window in Code Composer Studio.  -j 8: Specifies the number of jobs, or commands, to run simultaneously (8). |
|  | F | ${CCS\_UTILS\_DIR}/bin/gmake-j 8 | -j 8 | -j 8: Specifies the number of jobs, or commands, to run simultaneously (8). |
| Generate Makefiles automatically | A | Selected |  | Uses the built-in makefile generator of Code Composer Studio to generate the makefile for the compiler. |
| Behavior – Build Settings | | | | |
| Stop on first build Error | A | Selected |  | Build execution stops on first build error |
| Use Parallel build | A | Selected (use parallel jobs = 8) |  | See above for option -j 8. |
| Behavior – Build Settings | | | | |
| Build on resource save (Auto Build) | A | Selected *Make build Target:* all |  | Default settings |
| Build (Incremental Build) | A | Selected *Make build Target:* all |  | Default settings |
| Clean | A | Selected *Make build Target:* clean |  | Default settings |
| Steps – Pre-build Steps | | | | |
| Command | A |  |  |  |
| Description | A |  |  |  |
| Steps – Pre-build Steps | | | | |
| Command | A | "${PROJECT\_ROOT}/postbuild.bat" "${BuildArtifactFileBaseName}" |  | Runs a batch that is maintained by Nexteer. The batch file performs a few tasks at the end of the build, including: Run checksum and ECC over the created out file, generating files needed for PDX container creation to support customer tools, creating an hex format output of the file, and performing an archiving script to zip up the output deliverables, source files, and documentation. |
| Description | A | Create flash image: Intel-HEX |  |  |
| Build Variables | | | | |
|  | A | |  |  |  | | --- | --- | --- | | Name | Type | Value | | CM\_ROOT | Directory | ${PROJECT\_ROOT}/../.. | |  | CM\_ROOT is used to link the components to the project. |
| Environment | | | | |
|  | A | |  |  |  | | --- | --- | --- | | Name | Value | Origin | | CCS\_JAVA\_HOME | << Install Path >> | BUILD\_SYSTEM | | CCS\_UTILS\_DIR | << Install Path >> | BUILD\_SYSTEM | | CWD | << Install Path >> | BUILD\_SYSTEM | | PWD | << Install Path >> | BUILD\_SYSTEM | |  | Values are determined by the install path of Code Composer, the works space, and the compiler tools. These are populated automatically by Code Composer Studio. |
| Link Order | | | | |
|  | A |  |  | No link order is currently required. |
| Dependencies | | | | |
|  | A |  |  | No dependencies are currently required. |

##### ARM Compiler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Command | A | "${CG\_TOOL\_CL}" |  | Default settings |
| Command-line pattern | A | ${command} ${flags} ${inputs} |  | Default settings |

###### Processor Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Target processor version | A | 7R4 | -mv7R4 | TMS570 has an ARM7 Cortex R4 core |
| Designate code state, 16-bit(thumb) or 32-bit | A | 16  *(See Table 1: ARM Mode Files for files requiring 32)* | --code\_state=16 | 16 is the default selection for EPS project for thumb mode. Table 1: ARM Mode Files shows all files which are being compiled in 32 bit mode. |
| Specify floating point support | A | VFPv3D16 | --float\_support=vfpv3d16 | The Cortex R4 implements the VFPv3-D16 architecture per §1.2 of [2] |
| Application binary interface | A | eabi | --abi=eabi | eabi is required when using floating point unit |
| Little endian code | A | Not Selected | --little\_endian, -me | ARM is migrating away from the little-endian architecture, so Nexteer should not use little-endian in the design. |

###### Optimization Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Optimization | | | | |
| Optimization level | A | 2 | -O2 | Level 2 selected because it is the default level of optimization and is also the level of optimization used on the 28x platform.  Level 3 optimization adds program-level optimizations that are not desired at this time because level causes op-code changes in the final integrated application, that are not present during a unit test build.  Files that are compiled with a different optimization level are documented in Table 2: Optimized Files. |
| Optimize for speed | A | 3 | --opt\_for\_speed=3 | Level 3 enables optimizations geared towards improving the code performance/speed with a low risk of worsening of impacting code size.  Files that are compiled with a different optimization for speed level are documented in Table 2: Optimized Files. |

###### Debug Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Debugging model | R | Full symbolic debug | --symbdebug:drawf, -g | DWARF debug format is required when using floating point for the EABI support. Skeletal build not used because the debugging information was removed for all static symbols, only global symbols were available. |
|  | F | Symbolic debug for program analysis | --symbdebug:skeletal | An old option setting that was not updated to use the same settings as the release build option. The fault injection build is a special case build used for testing. |
| Optimize fully in the presence of debug directives | R | Selected | --optimize\_with\_debug | Combined with the full symbolic debug, the compiled binary will contain debugging information needed for system development but it will be optimized to reduce the impact to the binary execution.  In the past there were issues with having full symbolic debug enabled with the optimization level disabled and would cause performance issues. |
|  | F | Not Selected |  | To our knowledge, with the skeletal build, the debugging symbols available should be almost identical. |
| Specify DWARF version | A | 2 | --symdebug:dwarf\_version=2 | DWARF 2 format is being selected because there is a known errata with compiler when DWARF 3 is selected. |

###### Include Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Specify a preinclude file | A | None specified |  | All include files are strictly specified in the source files. This option is not used. |
| Add dir to #include search path | A |  |  | The file attached shows the current list of includes as of 29July13. |

###### MISRA-C 2004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
|  | A | None specified | --check\_misra | QAC is used during EPS software development for official MISRA conformance checking. MISRA checking is not enabled in the project due to the additional warnings that are produced. |

###### Advanced Options

Language Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| C/C++ Dialect | A | None selected |  | Embedded C++ is not being used in this project.  K&R compatibility should not be used. This project is required to be ANSI/ISO C compliant. |
| Language mode | A | None selected |  | Leaving this blank provides the default level of normal ANSI/ISO mode, which provides warnings for all non-ANSI/ISO features used in the program. All warnings are justified at the end of each build, so if a non-ANSI/ISO feature is required the warning will be justified in the warning justification document. |
| Create precompiled header file as named | A | None specified | --create\_pch | Not using precompiled header file option at this time. It can be used to decrease build times. |
| Disable intrinsic functions | A | Not Selected | --no\_intrinsics, -pn | Although direct use of intrinsic functions must be avoided in all portable files (e.g. AUTOSAR compliant) the use of an intrinsic in a macro replacement defined in a target specific file could be allowed, so intrinsic functions are left as enabled. |
| Multibyte character support | A | Not Selected | --multibyte\_chars, -pc | Default is to disable support. There is no known reason to enable this support for this project. |
| Allow extern C functions to propagate exceptions | A | Not Selected | --extern\_c\_can\_throw | Default is to disable support. This option is related to C++ function external linkage to C files. The project is written only in C, so support for this is not needed. |
| Precompiled header file for compilation | A | None specified | --use\_pch | Not using precompiled header file option at this time. It can be used to decrease build times. |
| Enable C++ exception handling | A | Not Selected | --exceptions | C++ language is not used in this project. |
| Automatically create/use precompiled headers | A | Not Selected | --pch | Not using precompiled header file option at this time. It can be used to decrease build times. |
| Output xref listing to .crl file | A | Not Selected | --gen\_acp\_xref, -px | Cross-reference listing file not required for common debugging. Disabled by default. |
| Specify precompiled header path | A | None specified | --pch\_dir | Not using precompiled header file option at this time. It can be used to decrease build times. |
| Floating Point mode | A | None selected | --fp\_mode=relaxed | Relaxed can result in faster code with some loss of accuracy and does not conform with ISO. Strict should be selected to ensure ISO conformance. Current configuration uses relaxed. |
| Do not delete unreferenced static variables | A | Not Selected | --keep\_unneeded\_statics | In general, keeping unreferenced static variables needlessly allocates memory. This is not currently necessary in the project. |
| Support C++ run-time type information | A | Not Selected | --rtti, -rtti | C++ language is not used in the project. |
| Treat C files as C++ files | A | Not Selected | --cpp\_default, -fg | C++ language is not used in the project. |
| Enable support for GCC extensions | A | Not Selected | --gcc | This option enables GCC features that are not found in the ANSI C standard. Leaving disabled to comply with the ANSI C standard. |
| Ignore inline keyword | A | Selected | --no\_inlining, -pi | Although the inline keyword is not part of the ANSI/ISO standard, it has been used in past projects when required. If used, then the inlining functionality should be performed. |
| Info on considered unused precompiled headers | A | Not Selected | --pch\_verbose | Not using precompiled header file option at this time. It can be used to decrease build times. |
| Do static early template instantiation | A | Not Selected | --static\_template\_instantiation | Templates are a feature of the C++ language. C++ is not being used in this project. |
| Program mode compilation | A | Not Selected | --program\_level\_compile, -pm | This option is only applicable when using opt\_level=3. This project is currently using 2. |
| Output raw listing to .rl file | A | Not Selected | --getn\_acp\_raw, -pl | Raw listing file is not archived as part of an official release. In an effort to reduce the Debug build time; this option is disabled because it is not normally needed when developing. It can be enabled in a developer’s local build if a particular need arises. |

Parser Preprocessing Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Mode | A | Automatic |  | Default option. There is no reason at this time to change to manual mode. |

Predefined Symbols

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Pre-define NAME | A | APPLICATION | --define=APPLICATION | Required for Application build of Vector AUTOSAR |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Undefine NAME | A | None specified |  | Not required |

Diagnostic Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Quiet Level | A | Verbose | --quiet, -q | Suppress common compiler non-diagnostic output. Remarks, errors and warnings will still be generated, as well as feature specific status.  Verbose mode is only required under rare debugging circumstances, in which case it can be enabled manually when the need arises. |
| Verbose diagnostics | A | Not Selected | --verbose\_diagnostics, -pdv | When using the CCSv5 environment, the Error window links directly to the source line of the error. The additional information provided by this is not normally required for debugging. |
| Treat diagnostic <id> as warning | A | 225 | --diag\_warning, -pdsw | Treats implicit functions are warnings. |
| Set error limit to <count> | A | Not specified | --set\_error\_limit, -pdel | The default is 100 when this is not specified. |
| Generate user information file | A | Not Selected | --gen\_aux\_user\_info, -b | Unclear what this file exactly contains. The default behavior is being kept. |
| Output diagnostic to .err file | A | Not Selected | --write\_diagnostics\_file, -pdf | There is no need for this file at this time. The same information is displayed in the standard output window. |
| Treat diagnostic <id> as remark | A | None specified | --diag\_remark, -pdsr | No need at this time to alter the default error/warning reporting of the compiler. |
| Treat diagnostic <id> as error | A | None specified | --diag\_error, -pdse | No need at this time to alter the default error/warning reporting of the compiler. |
| Emit diagnostic identifier numbers | A | Selected | --display\_error\_number, -pden | Displaying the diagnostic number generally doesn’t provide any additional useful information, however enabled in present configuration. |
| Suppress warnings | A | Not Selected | --no\_warnings, -pdw | All warning should be listed for debug and also for logging in the build output for the release. |
| Treat warnings as errors | A | Not Selected | --emit\_warnings\_as\_errors, -pdew | At this time certain warnings are allowed for a released build. All warnings are justified in a warnings document provided with the release. |
| Suppress diagnostic <id> | A | 232 | --diag\_suppress, -pds | 232 – Non-Standard Type for a bitfield is suppressed due to the large number of remarks generated for TI register definition structures. This diagnostic remark is not considered critical and is being disabled to allow the diagnostic output to be meaningful and not cluttered with this remark. |
| Issue remarks | A | Not Selected | --issue\_remarks, -pdr | All remarks should be listed for debug and also for logging in the build output for the release. |
| Print version numbers for each tool | A | Not Selected | --tool\_version, -version | Tool versions are suppressed for Debug build because they clutter the standard output window and are very rarely required to solve a problem when debugging. |

Runtime Model Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Place each function in a separate subsection | A | None selected | --gen\_func\_subsections, -ms | Default behavior is off. In general this optimization would not produce any benefit for the EPS application with properly designed libraries. This is being disabled until a need for it arises. |
| Use linker-generated file to remove dead functions | A | None specified | --use\_dead\_funcs\_list | Benefit from use of this option requires building the software twice. Additionally, the quality tools used during the development process should identify all dead functions and appropriate actions should be taken by the build manager/developers to remove the dead functions. |
| Generate unaligned loads and stored | A | None selected | --unaligned\_access | Since faults can be generated when accessing memory out of alignment, the feature is turned off. |
| Designate enum type | A | packed | --enum\_type | Default is packed for EABI. This project is using floating point which requires EABI. |
| Force alignment of structures to <bytecount> bytes | A | None specified | --alight\_structs | There is no known reason why structure alignment would be necessary in this project. |
| Reserve as global register | A | None specified | --global\_register, -r | There is no need to reserve a register for global use at this time in the EPS application. |
| Allow reassociation of sat arithmetic | A | None selected | --sat\_reassoc | It is not clear what this option does when it is enabled. It is also not clear if off is the default. None specified is the default and until understanding this parameter is the selection. |
| Compile for power profiling | A | Not Selected | --profile:power | Disable support of the TI profiling tool for Release builds because the profiler support causes additional code size overhead. |
| Enable 16 bit code | A | Selected  (See Table 1: ARM Mode Files for files requiring 32) | --thumb\_state, -mt | Thumb is the default selection for EPS project in order to reduce the memory footprint. Table 1: ARM Mode Files shows all files which are being compiled in 32 bit mode. |
| Allow reassociation of FP arithmetic | A | None selected | --fp\_reassoc | This option is automatically set when the fp\_mode is specified. The fp\_mode is specified as strict in this document, which causes this option to be set to “No”. “Yes” will cause ANSI violations. |
| Enums may be char/short, instead of int | A | Not Selected | --small\_enum, --small-enum | The default selection is not selected, however it is unclear how this option differs from the enum\_type packed option. |
| Enable dynamic stack overflow checking | A | Not Selected | --stack\_overflow\_check, -mo | There is not enough information in [1] to determine what effect this option has on the compiled code. The default is disabled. The stack overflow checks included with the OSEK OS should provide the required overflow detection capability, so usage of this option seems unlikely. |
| Specify whether constants can be embedded in code sections | A | Not Selected | --embedded\_constants | Can only be turned off for Cortex M devices. |
| Generates SIMD instructions targeting Neon | A | Not Selected | --neon | Option is not applicable to TMS570.  Per ARM website “NEON technology is a 128 bit SIMD (Single Instruction, Multiple Data) architecture extension for the ARM [Cortex™-A series](http://www.arm.com/products/processors/cortex-a/index.php) processors”. The TMS 570 uses a Cortex™-R series processor. |
| Specify how to treat plain chars | A | unsigned | --plain\_char=unsigned | Treating chars as unsigned is the default behavior and is the behavior of the previous EPS project. |
| Specify length of maximum branch chain | A | Not specified | --max\_branch\_chain, -ab | Branch chaining is supported in 16-BIS mode only, which is currently not enabled. The default length is 10. Leaving this setting unspecified until it becomes applicable. |
| Chars signed by default | A | Not Selected | --signed\_chars, -mc | This option seems to be redundant with the –plain\_char option. |
| No dual state support | A | Not Selected | --disable\_dual\_state, -md | This option is only applicable when using the TIABI. It is not applicable to the EABI used for this project (See Application Binary Interface selection option rationale) |
| Prevent generation of branch chains in Thumb mode | A | Not Selected | --disable\_branch\_chaining | 16-bit code is not enabled at the moment, so this option has no effect. If 16-bit code is being used, then the usage of branch chains could be disabled in time critical code where execution time is more important that code density. |
| Compile for breakpoint-based profiling | A | Not Selected | --profile:breakpt | Disable the profiling option to allow full optimization of the compiled software. |

Advanced Optimizations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Generate optimized source interlisted assembly | A | Not Selected | --optimizer\_interlist, -os | Enabling this option may prevent some optimizations from occurring. For the Release build all optimizations for the selected optimization level are desired to be applied.  In an effort to reduce the Debug build time, this option is disabled because it is not normally needed when developing. It can be enabled in a developer’s local build if a particular need arises. |
| Specify call assumptions when optimizing | A | None selected | --call\_assumptions, -op | This option is not applicable, unless using optimization level 3 (--opt\_level=3). This project is using optimization level 2 (See Optimization Level selection for more details) |
| Specify threshold for automatic inlining | A | Not Specified | --auto\_inline, -oi | When no auto inline threshold is specified, the decision to inline a function at a particular call-site is based on an algorithm that attempts to optimize benefit and cost. |
| Inline functions only called once | A | Not Selected | --single\_inline | Enabled during release build to save the calling/return overhead when source level debugging is less likely to be required. |
| Remove entry/exit hooks from the inlined functions | A | No | --remove\_hooks\_when\_inlining | Entry/Exit hooks are not being used in this project, so this option does not apply. It is being left at is default value. (See Entry/Exit Hook Options for more info) |
| Generate Optimizer information file at level[0-2] | A | None selected | --gen\_opt\_info, -on | This option only applies when Optimization Level 3 is selected (--opt\_level=3). This project is using Optimization Level 2 (see Optimization Level setting for more info) |
| Assume called funcs hidden aliases(rare) | A | No | --aliased\_variables, -ma | Based on the description of this option in §3.4 of [1], this worst-case aliasing assumption does not need to be applied for this software project. |

Entry/Exit Hook Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Insert this call at the entry to each function | A | Not specified | --entry\_hook | No entry hook is currently being specified for this software project. |
| Pass caller’s name or address to entry hook | A | Not selected | --entry\_parm | No entry hook is currently being specified for this software project. |
| Insert this call at the exit from each function | A | Not specified | --exit\_hook | No exit hook is currently being specified for this software project. |
| Pass caller’s name or address to exit hook | A | Not selected | --exit\_parm | No exit hook is currently being specified for this software project. |

Library Function Assumptions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Level of printf support required | A | None selected | --printf\_support | This option does not apply because printf support is not required in this project, thus the library is not linked in. |
| File does not define any RTS library func(def.) | A | Not Selected | --std\_lib\_func\_not\_defined, -ol2 | Per §3.2.1 of [1] this option is only applicable when using Optimization level 3 (--opt\_level=3). This project is using Optimization level 2. |
| File redefines an RTS library function. | A | Not Selected | --std\_lib\_func\_redefined, -ol0 | Per §3.2.1 of [1] this option is only applicable when using Optimization level 3 (--opt\_level=3). This project is using Optimization level 2. |
| File contains an RTS library function. | A | Not Selected | --std\_lib\_func\_defined, -ol1 | Per §3.2.1 of [1] this option is only applicable when using Optimization level 3 (--opt\_level=3). This project is using Optimization level 2. |

Assembler Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Keep the generated assembly language (.asm) file | A | Selected | --keep\_asm, -k | .asm files are not saved as part of the release build process and are generally not referenced after a build. unless debugging, so they are not generated for release build.  They are also not generated by the Debug build by default because they are generally not used and clutter the directory. If they are required for Debugging a specific problem, then this option should be enabled in the developer’s workspace. |
| Source interlist | A | None selected | --src\_interlist | Source interlisted files are only used for debug.  In an effort to reduce the Debug build time, this option is disabled because it is not normally needed when developing. It can be enabled in a developer’s local build if a particular need arises. |
| Generate listing file | A | Not Selected | --asm\_listing, -al | listing files are not saved as part of the release build process and are generally not referenced, unless debugging, so they are not generated for release build.  In an effort to reduce the Debug build time, this option is disabled because it is not normally needed when developing. It can be enabled in a developer’s local build if a particular need arises. |
| Keep local symbols in output file | A | Not Selected | --output\_all\_syms, -as | This option puts labels in the symbol table. At this time the benefit from that the additional label information provides is not know to be used, therefore this option is left at its default value so as to not increase the size of the output file. |
| Do no generate .clink for .const sections | A | Not Selected | --no\_const\_clink | The default behavior is to remove unreferenced global constant arrays at link time.  **CAUTION:** A potential issue that could arise from this optimization is that if any part identification information is intended to be defined in the software and never referenced, the constant data will be optimized out. |
| Simulate source ‘.copy filename’ | A | None specified | --copy\_file, -ahc | There is no need to insert special assembly instructions into the beginning of the assembly files. |
| Symbol names are not case-significant | A | Not Selected | --syms\_ignore\_case, -ac | Default behavior is that symbol names are case significant. There is no reason to ignore symbol case. |
| Use unified assembly language | A | Not Selected | --ual | This option doesn’t apply to the TMS570 because it is using an ARMv7 core for which UAL syntax is the default. Leaving this value at its default of not selected. |
| Undefine assembly symbol NAME | A | None specified | --asm\_undefine, -au | This project does not require undefining assembly defines. |
| Simulate source ‘.include filename’ | A | None specified | --include\_file, -ahi | This project does not require including a generic include into all assembly files. |
| Pre-define assembly symbol NAME | A | None specified | --asm\_define, -ad | This project does not require defining assembly defines. |
| Generate first-level assembly include file list | A | Not Selected | --asm\_includes, -api | This file is not required for the release build and is not foreseen to be used for typical debugging. |
| Generate cross reference file | A | Not Selected | --cross\_reference, -ax | This file is not required for the release build and is not foreseen to be used for typical debugging. |
| Generate assembly dependency information | A | Not specified | --asm\_dependency, -apd | This file contains information suitable for input to a standard make utility. At the moment, the make process is controlled by the managed make process, so this file is not required. |

File Type Specifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| File is a C file (default for .c/no ext) | A | None specified | --c\_file, -fc | Default behavior of .c as a C source file is desired for this project. |
| File is an assembly file (default for .asm) | A | None specified | --asm\_file, -fa | Default behavior of .asm as a assembly source file is desired for this project. |
| File is an object file (default for .obj) | A | None specified | --obj\_file, -fo | Default behavior of .obj as an object file is desired for this project. |
| File is a C++ file (default for .C .cpp. cc) | A | None specified | --cpp\_file, -fp | This option does not apply to this project since C++ is not being used. Default behavior is acceptable for this project. |

Directory Specifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Mode | A | Automatic |  | Default is Automatic Mode. There is no reason to change from the default. |
| Temporary file directory | A | None specified | --temp\_directory, -ft | Default is current directory. There is no reason to change from the default. |
| Assembly file directory | A | None specified | --asm\_directory, -fs | Default is current directory. There is no reason to change from the default. |
| Preprocessor file directory | A | None specified | --pp\_directory | Default is current directory. There is no reason to change from the default. |
| Compilation output file name, can override | A | None specified | --output\_file, -fe | Default is current directory. There is no reason to change from the default. |
| Object file directory | A | None specified | --obj\_directory, -fr | Default is current directory. There is no reason to change from the default. |
| Listing/xref file directory | A | None specified | --list\_directory, -ff | Default is Object file directory. There is no reason to change from the default. |

Default File Extensions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Extension for object files | A | None specified | --obj\_extension, -eo | Default is .obj and is the desired extension. |
| Extension for listing files | A | None specified | --listing\_extension, -es | Default is .lst and is the desired extension. |
| Extension for C++ files | A | None specified | --cpp\_extension, -ep | Default is .cpp. C++ is not used in this project, so this setting doesn’t matter. |
| Extension for assembly files | A | None specified | --asm\_extension, -ea | Default is .asm and is the desired extension. |
| Extension for C files | A | None specified | --c\_extension, -ec | Default is .c and is the desired extension. |

Command Files

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Command Files | | | | |
| Read options from specified file | A | None specified | --cmd\_file, -@ | Currently all command line options are being specified via the CCSv5 Properties dialog selection and appending the commands from an additional command file to the compiler execution command is not necessary. |

##### ARM Linker

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Command | A | "${CG\_TOOL\_CL}" |  | Default values |
| Command-line pattern: | A | ${command} ${flags} ${output\_flag} ${output} ${inputs} |  | Default values |

###### Basic Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Basic Options | | | | |
| Specify output file name | A | "EPS\_GM\_C1xx\_dbg.out" | --output\_file, -o= | See EPS Naming conventions. |
| Set C system stack size | A | Not specified | --stack\_size, -stack | The .stack section is specified as a number of sub-stack sections in the linker command file to allocate space for each of the TMS570 operating mode stacks. For this reason, it is not necessary to specify the stack size via this command option. |
| Input and output sections listed into <file> | A | "EPS\_GM\_C1xx\_dbg.map" | --map\_file, -m | See EPS Naming conventions. |
| Heap size for C/C++ dynamic memory allocation | A | 0 | --heap\_size, -heap | Dynamic memory allocation is not performed in this project. |

###### File Search Path

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| File Search Path | | | | |
| Include library file or command file as input | A | "rtsv7R4\_T\_be\_v3D16\_eabi.lib"  "F021\_API\_CortexR4\_BE\_v3D16.lib" | --library, -l | Run-time support library for the TMS570 ARM core compile in big-endian mode. This library supplies the definitions for memcpy, memmove, memset, \_\_aeabi\_idivmod, etc ., that are required for building files in the project. |
| Add <dir> to library search path | A | "${CG\_TOOL\_ROOT}/lib"  "${CG\_TOOL\_ROOT}/include"  "${workspace\_loc:/${ProjName}/Fls/src}" | -search\_path, -i | At the moment the RTS library installed with the tool chain selected is used directly from the install directory.  Metrics include required for support in the linker command file for build constants that change with metrics settings. |
| Reread libraries; resolve backward references | A | Selected | --reread\_libs, -x | Enabling this option provides a more robust way of resolving references in by removing the linker operation order dependency. |
| Search libraries in priority order | A | Not Selected | --priority, -priority | This option does not apply to this project. Libraries designed for use in this project will not contain definitions for the same function. |
| Disable automatic RTS selection | A | Selected |  | Automatic RTS selection is not desired for this project. The project should be structured so that a change in library selection is controlled by a change request. |

###### Advanced Options

Command File Preprocessing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Pre-define preprocessor macro \_name\_to\_value\_ | R | Not Specified |  |  |
| F | Not Specified |  |  |
| Undefine preprocessor macro \_name\_ | A | Not specified | --undefine | Preprocessor macro expressions are not used within the linker command file for this project. |
| Don’t use C preprocessor for command files | A | Not Selected | --disable\_pp | C preprocessor is not used for command files in this project. |

Diagnostics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Verbose diagnostics |  | Not Selected | --verbose\_diagnostics | Default setting |
| Treat diagnostic <id> as warning | A | None specified | --diag\_warning | No need at this time to alter the default error/warning reporting of the linker. |
| Set error limit to <count> | A | Not specified | --set\_error\_limit | The default error limit for the linker is 100. This is acceptable for this project. |
| Treat diagnostic <id> as remark | A | None specified | --diag\_remark | No need at this time to alter the default error/warning reporting of the linker. |
| Treat diagnostic <id> as error | A | None specified | --diag\_error | No need at this time to alter the default error/warning reporting of the linker. |
| Warn if an unspecified output section is created | A | Selected | --warn\_sections, -w | Warnings for unspecified output sections should be issued so that the appropriate action can be taken. All output sections should be explicitly allocated in this project. |
| Emit diagnostic identifier numbers | A | Not Selected | --disaypl\_error\_number | Displaying the diagnostic number generally doesn’t provide any additional useful information, unless one would like to see the diagnostic number for purposes of altering the diagnostic reaction. |
| Suppress warnings | A | Not Selected | --no\_warnings | All warning should be listed for debug and also for logging in the build output for the release. |
| Treat warnings as errors | A | Not Selected | --emit\_warnings\_as\_errors, -pdew |  |
| Suppress diagnostic <id> | A | None specified | --diag\_suppress | No need at this time to alter the default error/warning reporting of the linker. |
| Issue remarks | A | Selected | --issue\_remarks | All remarks should be listed for debug and also for logging in the build output for the release. |
| Don’t demangle symbol names in diagnostics | A | Not Selected | --no\_demangle | Name demangleing is applicable to C++ source, which is not used in this project. |

Linker Output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Detailed link information data-base into <file> | A | "EPS\_GM\_C1xx\_dbg\_linkInfo.xml" | --xml\_link\_info, -xml\_link\_info | Default setting. |
| Produce absolute executable object file (default) | A | Not Selected | --absolute\_exe | Presently no need for this output. |
| Display attribute settings in map file | A | Not specified | --mapfile\_contents | Default setting |
| List removed dead functions into file | A | Not specified | --generate\_dead\_funcs\_list | Default setting |
| Create a ROM object | A | Not Selected | --rom | No need for this option. |
| Produce a relocatable output module | A | Not Selected | --relocatable, -r | No need for this option. |

Symbol Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Strip symbol table and line number entries | A | Not Selected | --no\_symtable, -s | Default setting |
| Do not localize symbols matching pattern | A | Not specified | --globalize | Default setting |
| Scan libraries for duplicate symbol definitions | A | Not Selected | --scan\_libraries, -scanlibs | Default setting |
| Add <sym> to symbol table as an unresolved symbol | A | Not specified | --undef\_sym, -u | No symbols to add as an unresolved symbol. |
| Make the symbols matching pattern local | A | Not specified | --localize | Default setting |
| Make all global symbols static | A | Not Selected | --make\_static, -h | Default setting |
| Specify program entry point for the output module | A | Not specified | --entry\_point, -e | No entry point required. |
| Exclude symbols matching pattern from hiding | A | Not specified | --unhide | No requirements to unhide any symbols |
| Specify symbols/sections to be retained by linker | A | Not specified | --retain | No symbols or sections that need to be retained |
| Don’t make global symbol static if | A | Not specified | --make\_global, -g | Nothing required to be added in this section |
| Hide symbols matching pattern | A | Not specified | --hide | No symbols required to be hidden |
| No type merging in symbolic debugging information | A | Not Selected | --no\_sym\_merge, -b | Default setting. |

Runtime Environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Initialization model | A | Link using ROM auto initialization model (--rom\_model, -c) | --rom\_model, -c | The linker defines cinit as the starting address of the .cinit section. The C boot routine uses this symbol as the starting point for autoinitialization. Using the ram\_model option sets the cinit to -1 which indicates that the initialization tables are not in memory, so no initialization is performed at run time which is not desired. |
| ARM big endian modes | A | Link big-endian code in be-32 format | --be32 | Selected to provide a big-endian program with big-endian encoded instructions. |
| Default fill value for holes in output sections | A | Not specified | --file\_value, -f | Default value. |
| Generate far call trampolines | A | None selected | --trampolines | No need for this feature at this time, left to default value. |
| Set C arg/argv memory size | A | Not specified | --arg\_size, -args | No need for this feature at this time, left to default value. |

Link Optimization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Eliminate sections not needed in the executable | A | Off | --unused\_section\_elimination | Sections should not be removed that are defined; therefore this option is turned off. |
| Aggressively reduce the size of the DWARF information | A | Off | --compress\_dwarf | For ELF object files, the --compress\_dwarf option eliminates duplicate information that could not be removed through the use of ELF COMDAT  groups. Since the headers are only required to use what is referenced by the source file, this option is not useful for builds and thus disabled. |
| Compress ELF sections copied using copy tables | A | Not specified | --copy\_compression | No compression is required, therefore left blank. |
| Compress ELF C style auto initialization data | A | Not specified | --cinit\_compression | No compression is required, therefore left blank. |

Miscellaneous

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Miscellaneous | | | | |
| Add <function> to preferred placement order list | A | Not specified | --preferred\_order | No order preferred. |
| Disable conditional linking ignore .clink | A | Not Selected | --disable\_clink, -j | According to page 86 in [5], “under the ELF EABI model, the linker assumes that all sections are eligible for removal via conditional linking. Therefore, the .clink directive has no effect under EABI.” |
| Strict compatibility checking | A | None selected | --strict\_compatibility | Using the default value to disable the check due to the possibility of false compatibility errors when linking in object files built with an older toolset, or  with object files built with another compiler vendor's toolset. |
| Zero initialize ELF uninitialized sections | A | Off | --zero\_init | Set to off otherwise initialization to zero of all RAM with unspecified init values will be generated into the cinit table. This is not necessary as the hardware RAM initialization will perform this step. |
| Select trampoline minimization algorithm | A | None selected | --minimize\_trampolines | Default setting. |
| Minimum space between non-adjacent trampolines | A | Not specified | --trampoline\_min\_spacing | Default setting. |

Dynamic Linking Support Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Cfg** | **Value** | **Cmd Line Opt** | **Rationale** |
| Add directory to end of dynamic library search path | A | Not specified | --runpath | Feature is not required currently for this project. |
| Export specified symbol | A | Not specified | --export | Feature is not required currently for this project. |
| Specify ELF shared object file name | A | Not specified | --soname | Feature is not required currently for this project. |
| Force all import references to bind during static linking | A | None selected | --forced\_static\_binding | Feature is not required currently for this project. |
| Add directory to beginning of dynamic library search path | A | Not specified | --rpath | Feature is not required currently for this project. |
| Specify the symbol name of the initialization code | A | Not specified | --init | Feature is not required currently for this project. |
| Specify the symbol name of the finalization code | A | Not specified | --fini | Feature is not required currently for this project. |
| Import specified symbol | A | Not specified | --import | Feature is not required currently for this project. |
| Generate dynamic executable or dynamic library | A | None selected | --dynaimc | Feature is not required currently for this project. |

### Post Build

Upon completion of the build a set of activities is executed on the build outputs via execution of the postbuild.bat file launched by the CCS build process.

|  |  |  |
| --- | --- | --- |
| **Utility Name** | **Description** | **Installation Instructions** |
| hex470 | TI provided tool used to generate a hex/s19 record file required as input by the other utilites | Archived in the project under Tools |
| nowECC.exe | TI provided tool used to generate ECC data for the Flash memory. | Archived in the project under Tools |
| SWE Generator | BMW provided tool to generate the BMW required programming files. | Archived in the project under Tools |
| Hexview | Vector provided utility used to fill the Hex file image | Archived in the project under Tools |
| Common Checksum Tool (CCT) | Nexteer developed checksum calculation utility | Archived in project under Tools |
| WinZip® Command Line Support Add-On | This utility adds command line interface support to WinZip.  WinZip is assumed to be installed on all computers. If is inot, then it must also be installed. | Download from http://www.winzip.com/downcl.htm  PATH variable must contain the directory in which the add on was installed |

Table 5: Required Post Build Utilities

# NHET

## Make Process

This NHET component consists of two source files (\*.het) and thus the make process is controlled via a batch file that performs the following steps:

1. Invoke the hetp Assembler for all HET source files (\*.het)
2. Copy the \*.h output files to the appropriate directory. The \*.c output files are already in the correct folder location.

## Tool Settings

### Assembler

The following commands are performed as part of step 1 of the batch file described in the earlier section.

*hetp -hc32 -n0 -v2 Nhet\_Prog.het*

*hetp -hc32 -n1 -v2 Nhet2\_Prog.het*

|  |  |
| --- | --- |
| **Cmd Line Opt** | **Rationale** |
| -hc32 | This option produces the .c and .h files necessary to integrate the NHET project with the main project. Also this option must be used with -nx (see below) |
| -nx  *-n0 and -n1 from the commands above* | Specifies the x-th HET module in the device. Also this option must be used with -hc32. The 65nm Gladiator parts contain two NHET modules. |
| -v2 | Support enabled for additional instructions compared to NHET version 1. Required for 65nm Gladiator parts that contain NHET version 2. |

# Appendix

## Debug Output Files

This section lists the optional Debug Output file options that are not selected in the normal build configurations.

|  |  |
| --- | --- |
| File Type |  |
| Cross-Reference Listing Information |  |
| Raw Listing File |  |

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1.0 | Initial Creation for C1xx Bev | 08/08/16 | CP |