TEXAS INSTRUMENTS

# Closed Defects in Release

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| CODEGEN-2010 | Decomp error involving packed class | Fixed | ARM_5.2.0B1 | ARM_5.2.9 |  | Fixed an issue that could result in a "Decomposition error" abort of the compiler. |
| CODEGEN-1976 | Value of __cplusplus is wrong | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | If possible, use the -ps or --strict_ansi options. This mode will use the strict definition of __cplusplus, which is 199711L. | Our parser mimicked G++ behavior for the value of this macro in relaxed ANSI mode. This reproduced a bug in G++ versions v.4.7 and v.4.3 that has since been fixed. |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| CODEGEN-1703 | Designated initializer plus struct hack hangs compiler | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | Avoiding using string constants to initialize objects with flexible array members. Instead, use a brace-initialized array. For example: struct { int a; char b[]; } mystruct = {0, {'h', 'e', 'l', 'l', 'o'} }; | Fixed a compiler hang caused by initializing flexible array members with string constants. struct {int a; char b[]; } mystruct = {0, "hello"} /* Would cause the compiler to hang and/or crash */ |
| CODEGEN-1640 | MISRA 19.1 misreported: #include statements should only be preceded by other preprocessor directives | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | No practical workaround | |
| CODEGEN-1634 | MISRA 7.1 misreported: octal tokens in token paste | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | No practical workaround | |
| CODEGEN-1632 | MISRA 15.2 misreported: switch clause unconditional break | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | Put an explicit break at the end of the compound statement | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| CODEGEN-1555 | Incorrect result for ullong expression passed to abs | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | Use llabs instead of abs | abs(x-y) may compute an incorrect result when x and y are unsigned long long variables. |
| CODEGEN-1438 | Using isnan (and friends) in a custom object library built with 5.2.x compiler means that library cannot be linked with the | Fixed |  | ARM_5.2.9 | The work around is to rebuild any object files that generated undefined symbols (_FDclass/_Fdpcomp/_FDsign and similar) using a more recent 15.x or 16.x release version of compiler |  |
| CODEGEN-1429 | Software pipelined loop generates different results than loop not pipelined | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | Avoid unsigned expressions in subscripts or in computing subscripts, or compile with -o1 or -o0. |  |
| SDSCM00052814 | TMS570LC4357 ECC algorithm not supported | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | There is a backdoor in the implementation that you can use to realize this new algorithm. Use this algorithm | Some devices, such as TMS570LC4357 Hercules, have a FLASH bank in high memory, and ECC uses the high bits of the address |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| | | | | | specification: ECC { algo_name : address_mask = 0xffffffff hamming_mask = /* address */ 0x53aaa750, 0xeb45d688, 0xa6d54da8, 0x9e353c68, 0x7e0cfc18, 0xfe03fc00, 0x01fffc00, 0xfe0003f8, /* upper */ 0x2E4B2E4B, 0x57155715, 0x99A699A6, 0xE338E338, 0xFCC0FCC0, 0x00FF00FF, 0xFF0000FF, 0xFF0000FF, /* lower */ 0xD1B4D1B4, 0x57155715, 0x99A699A6, 0xE338E338, 0xFCC0FCC0, 0x00FF00FF, 0xFF0000FF, | in the ECC calculation. Therefore, we need to extend the participating address bits mask in the linker-generated ECC handling. |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| | | | | | 0x00FFFF00, parity_mask = 0x0c mirroring = F021 } If you value your sanity, do not ask where the constants come from. | |
| CODEGEN-1333 | Structure assignment causes compiler to fail with INTERNAL ERROR: Decomposition error | Fixed | ARM_5.2.7 | ARM_5.2.8 | Replace struct assignments involving packed structures with a memcpy() call to copy the contents of the RHS of the struct assign to the LHS. | |
| CODEGEN-1320 | Bit field operations mishandled | Fixed | ARM_5.2.7 | ARM_5.2.9 | | |
| CODEGEN-1311 | Initializer involving empty braces for union member leads to INTERNAL ERROR: ILLEGAL INITIALIZATION | Fixed | ARM_5.2.7 | ARM_5.2.8 | | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00052734 | Internal error in trampoline generation when user defines data symbol named "signal" | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | It is a user error to have two different objects in the system with the same name. Rename one of the objects. | |
| SDSCM00052864 | struct field in register is not sign-extended to 32 bits before 32-bit compare | Fixed | ARM_5.2.0 | ARM_5.2.8 | Declare the local struct as volatile | The compiler can place small structs in registers for fast access. In this case, the compiler fails to sign-extend a 16-bit struct member to 32-bits before a 32-bit comparison. This can happen for structs 32 bits or less, with any integer field 16 bits or less. |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00052786 | TI ARM compiler fails to build library | Fixed | ARM_5.2.7 | ARM_5.2.8 | Put the CCS utils sub-directories (c:/ti/ccsv6/utils/bin and c:/ti/ccsv6/utils/cygwin, for example) at the front of your PATH. Another alternative is to invoke mklib from inside a bash console (like MKS, for example). | |
| SDSCM00052888 | Illegal instruction inside IT sequence causes assembler to abort with segmentation fault | Fixed | ARM_5.2.0B1 | ARM_5.2.8 | | |
| SDSCM00052869 | Incorrect opcode for LDMED and LDMFA | Fixed | ARM_5.2.0 | ARM_5.2.8 | None | |
| SDSCM00052861 | NULL deref when using --minimize_trampolines=postorder | Fixed | ARM_5.2.0 | ARM_5.2.8 | As a workaround, use the option --trampolines=off and do not use the option --minimize_trampolines | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00052854 | Compiler crashes when MISRA checking is enabled and an empty struct initializer is used | Fixed | ARM_5.2.0B1 | ARM_5.2.8 | Use {0}, not {}, to initialize the struct | |
| SDSCM00052833 | Linker INTERNAL ERROR with object files with DWARF information compiled by IAR compiler | Fixed | ARM_5.2.0B1 | ARM_5.2.8 | Add option --compress_dwarf=off | |
| SDSCM00052805 | Decomposition error on while(*ptr++) where ptr points to volatile | Fixed | ARM_5.2.0B1 | ARM_5.2.8 | No workaround. | |
| SDSCM00052723 | Compiler seg faults with use of --opt_level=4 | Fixed | ARM_5.2.0B1 | ARM_5.2.7 | Use optimization level 3 or below. | |
| SDSCM00052699 | Compiler discards write to a volatile local struct member | Fixed | ARM_5.2.0B1 | ARM_5.2.8 | No workaround. | |
| SDSCM00052688 | Compiler fails to accept __MRC(15, ... | Fixed | ARM_5.2.0 | ARM_5.2.8 | No workaround. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00052565 | On MSP432, giving #pragma vector= non function name arguments causes segfault | Fixed | ARM_5.2.0B1 | ARM_5.2.7 | Use symbol names as arguments to #pragma vector=. | |
| SDSCM00052529 | Optimizer assumes enum variable can only have value of enumeration constants | Fixed | ARM_5.2.0B1 | ARM_5.2.7 | Don't make enumerated types with constants only for 0 and 1 if other values may be used. Either add a third value, or use values different than 0 and 1, or avoid using values that aren't the enumeration constants. | |
| SDSCM00052430 | 64-bit CRC values are truncated to 32-bits in linker map file and linker XML info file | Fixed | ARM_5.2.0B1 | ARM_5.2.7 | Use the Linux toolset to generate the linker map file and linker XML info file, or just ignore the reported CRC value in those files. | |
| SDSCM00052397 | Optimizer crashes on unreachable integer divide-by-zero | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | None. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00052374 | COMDAT functions may be incorrectly specialized when using -o4 | Fixed | ARM_5.2.0B1 | ARM_5.2.7 | None. | |
| SDSCM00052339 | demangler --output option does not work at all | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | armdem file.asm > file.dis | |
| SDSCM00052301 | Linker crashes when --ecc:ecc_error is used | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | None. | |
| SDSCM00052281 | ARM Parser Segfaults on OSX at Template Class | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | None. | |
| SDSCM00052265 | Flexible array with initializer not handle correctly | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | None. | |
| SDSCM00052257 | Use of --opt_level=4 causes link to fail with message symbol "name" redeclared with incompatible type | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | Change all anonymous members of the types involved to have an explicit name. | |
| SDSCM00052238 | Strong function symbol removed from linked ICODE with -o4 | Fixed | ARM_5.2.0B1 | ARM_5.2.7 | Use --opt_level=3 or below. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00052216 | M0 little-endian double multiply incorrectly gives +Inf | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | The source code to the RTS library can be found in a directory location similar to C:₩ti₩ccsv6₩tools₩compiler₩ti-cgt-arm_5.2.4₩lib₩src . Modify the file fd_mul16.asm as follows and add it to your project: .if .TMS470_BIG_DOUBLE POP {res1, op1e} ; Grab that stuff we stored off .else POP {res1} ; Grab that stuff we stored off POP {op1e} ; Grab that stuff we stored off .endif | |
| SDSCM00052185 | In certain cases at opt_level=2 or higher, compiler gets load and store instructions in the wrong order | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | Retain "volatile" in the cast, or remove it from the definition. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00052144 | RTS libraries do not get automatically built if the library is missing (Mac OS X) | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | Copy the libraries from a Windows/Linux machine. | |
| SDSCM00052140 | math.h should not have a using statement for abs | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | Include cmath instead of math.h. | |
| SDSCM00052114 | Windows Stack Size for Parser Should be Raised | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | There are at least two options: 1. On Windows, if you have Visual Studio available, the stack for an executable may be raised with editbin. For example, to raise the stack to 8MB for the acpia6x parser, you would use: editbin /stack:8388608 acpia6x.exe 2. Use the linux toolchain. The default stack is much higher. Even if that limit is reached, the soft limit for the stack size of user | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| | | | | | applications may be raised with ulimit. For example, to raise the stack to 16MB, you would use: ulimit -S -s 16384 (This could be placed in the startup script, such as .bashrc) | |
| SDSCM00052083 | Error correcting codes may not be generated for sections beyond the first in a memory region | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | Place each section that needs ECC in a separate memory region. | |
| SDSCM00052014 | Cannot read files with inode > 4B | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | None. | |
| SDSCM00051958 | Linker incorrectly splits text section. Section .text.1 is too long. It overwrites the trampoline at the start of .text.2 | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | Disabling split placement of the section will avoid this issue. Otherwise --disable_early_consolidation will also avoid this behavior. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00051870 | Underflow on conversion from double to float gives Inf instead of 0.0 | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | Avoid conversions from double to float, or use the 16-bit mode library. The 16-bit mode library is not available on Thumb 2 only devices such as Cortex-M3. | |
| SDSCM00051848 | Calling function that returns a structure causes compiler to fail with: INTERNAL ERROR: no match for CALL | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | None. | |
| SDSCM00051825 | Standard library headers may cause errors with the --float_operations_allowed option | Fixed | ARM_5.2.0 | ARM_5.2.5 | Do not use the --float_operations_allowed option. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00051809 | "if (P) X=A; else X=B;" may compile incorrectly when A and B are 0/1 and P is 0/1 with a side effect | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | Typically such an IF is equivalent to "X=P" or "X=!P"; write it that way to avoid the problem. Adding statements to either arm of the IF can also help. | |
| SDSCM00051762 | Dot expression adding align(64) computes hugely incorrect size | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | None. | |
| SDSCM00051740 | AP modified before V4 saved, potentially corrupting the stack pointer | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | None. | |
| SDSCM00051709 | incompatible wchar_t types diagnostic gives wrong type for one file | Fixed | ARM_5.2.0B1 | ARM_5.2.4 | Ignore the settings reported for the size of wchar_t | |
| SDSCM00051708 | Loop downcounter may be mistakenly typed too small | Fixed | ARM_5.2.0B1 | ARM_5.2.4 | Lower the level of optimization. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00051636 | The --rom option should not be exposed for any target except C6000 | Fixed | ARM_5.2.0 | ARM_5.2.4 | None. | |
| SDSCM00051629 | Compiler seg faults in C++ copy constructor call | Fixed | ARM_5.2.0B1 | ARM_5.2.4 | In some cases, using a different ABI option may avoid this error. However, the most safe option is to avoid the use of a copy constructor. | |
| SDSCM00051621 | Redeclaring extern "C" function as static causes parser to hang | Fixed | ARM_5.2.0B1 | ARM_5.2.4 | Use consistent function declarations. | |
| SDSCM00051602 | Parser may seg fault on template arguments with ! operator | Fixed | ARM_5.2.0 | ARM_5.2.4 | Using the --strict_ansi option will avoid this behavior. If this is not possible, changing template arguments of the form !foo to 0==foo will also avoid this behavior. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00051485 | Incorrect reordering of nested op= with ++ | Fixed | ARM_5.2.0B1 | ARM_5.2.9 | Don't embed "X op= Y++" under another assignment operator. | An expression like "a += b += c++" will produce the wrong answer. The problem is specific to sub-expressions of the form "X op= Y++" that occur under another assignment operator. |
| SDSCM00051472 | Conditionals that use shift operator may be optimized out | Fixed | ARM_5.2.0B1 | ARM_5.2.3 | This behavior may be avoided by lowering the optimization level to -o1 or -o0. Moving the conditional check result to a variable so that the shift is not in the conditional may in some cases also avoid this behavior. | |
| SDSCM00051463 | Automatic library resolution fails when a a load image file is included in the link | Fixed | ARM_5.2.0B1 | ARM_5.2.3 | Specify the specific library to link against instead of using libc.a. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00051412 | Compiler generates illegal assembly code | Fixed | ARM_5.2.0 | ARM_5.2.3 | The use of the constant 0x80000000 may cause the bug. Exchanging the constant with a smaller value will avoid the issue. If not possible, raising the optimization level may avoid the issue. | |
| SDSCM00051312 | Linker mistakenly issues output section warning diagnostic for sections pre-bound to hard coded addresses | Fixed | ARM_5.2.0B1 | ARM_5.2.2 | Ignore the warning | |
| SDSCM00051241 | Output section splitting creates a zero-length section | Fixed | ARM_5.2.0B1 | ARM_5.2.2 | 1) Do not split the section in question.<br>2) Change the zero-length section so that it has at least one more byte. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00051238 | The Cortex-R4 silicon defect for VMRSEQ APSR_nzcv, FPSCR is incorrectly warned about if the destination is an ARM register | Fixed | ARM_5.2.0B1 | ARM_5.2.2 | None | |
| SDSCM00051212 | MSP 4.4.0 changes to _lock.h declaration of _nop(void) instead of _nop() breaking driverlib | Fixed | ARM_5.2.0B1 | ARM_5.2.0 | Update _lock.h in the rts source folder, rename the library you are using, and it will automatically be rebuilt during next compile. To find compiler install location, click on project properties and click on: Resource->Linked Resources The "Path Variable" will have an entry for: CG_TOOL_ROOT c:₩install_location₩ccsv6₩tools₩compiler₩msp430_4.3.4 Within above folder look for: .₩lib₩src₩_lock.h | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| | | | | | .₩include₩_lock.h Edit both of above files and change below line 47: _CODE_ACCESS void _nop(void); to remove void argument and instead be: _CODE_ACCESS void _nop(); Rename the library you are using with suffix .old: .₩lib₩rtsx_lc_sd_eabi.lib change to a temp name .₩lib₩rtsx_lc_sd_eabi.lib.old Rebuild your project and the rts should automatically rebuild. | |
| SDSCM00051128 | Alias missed on address of local passed as function argument | Fixed | ARM_5.2.0B1 | ARM_5.2.0 | Use the -ma (--aliased_variables) option. | |
| SDSCM00051112 | Missing __aeabi_localeconv | Fixed | ARM_5.2.0B1 | ARM_5.2.5 | Link with another vendor's toolset | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00051110 | __aeabi_{i,l}div0 should be extern, not static | Fixed | ARM_5.2.0B1 | ARM_5.2.3 | Use a different vendor's RTS library. | |
| SDSCM00051097 | Output section splitting creates a zero-length section | Fixed | ARM_5.2.0B1 | ARM_5.2.0 | 1) Do not split the section in question. 2) Change the zero-length section so that it has at least one more byte. | |
| SDSCM00051039 | 64-bit divide routine does not return results as documented | Fixed | ARM_5.2.0B1 | ARM_5.2.0 | Don't rely on integer division by zero to return any reasonable value. | |
| SDSCM00050992 | The optimizer should not make a symbol an alias if it has either the location or code_section pragmas applied. | Fixed | ARM_5.2.0B1 | ARM_5.2.2 | Available workarounds are to: 1. Reduce the optimization level to -o2. 2. If using a COFF abi, you can move the functions into separate files. 3. Add an __asm("NOP") to the function which will prevent an alias from being created. | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00050973 | The function __aeabi_uidiv is missing in the RTS | Fixed | ARM_5.2.0B1 | ARM_5.2.0 | Link with another vendor's linker, or add a function named __aeabi_uidiv which takes two unsigned int arguments and calls __aeabi_uidivmod | |
| SDSCM00050959 | SIGSSEGV when using % in an option argument | Fixed | ARM_5.2.0B1 | ARM_5.2.3 | Don't use any printf-special characters like % in a compiler argument | |
| SDSCM00050834 | Assembler doesn't generate relocation entry for weak symbol | Fixed | ARM_5.2.0B1 | ARM_5.2.7 | Use --gen_func_subsections. | |
| SDSCM00050764 | Bound function has incorrect address in DWARF debug info | Fixed | ARM_5.2.0B1 | ARM_5.2.1 | Set a breakpoint at the bound function and run to it, rather than attempting to step into it. | |
| SDSCM00050005 | Compiler mistakenly issues MISRA diagnostic 12.9 for a float point type | Fixed | ARM_5.2.0B1 | ARM_5.2.8 | | |

| ID | Summary | State | Reported In Release | Target Release | Workaround | Release Notes |
|---|---|---|---|---|---|---|
| SDSCM00049912 | Float NaN converted to double incorrectly turned into Inf | Fixed | ARM_5.2.0B1 | ARM_5.2.6 | Manually check for NaN values where they are possible if software floating point conversions are required. | |
| SDSCM00046409 | __ldrex pointer parameter should be "volatile void*" | Fixed | ARM_5.2.0B1 | ARM_5.2.7 | None. | |
| CODEGEN-26 | mklib generates segmentation fault on MacOS X | Fixed | ARM_5.2.7 | ARM_5.2.8 | Make RTS library using a different host (Linux or PC) , then copy the newly created RTS object library into the install area on the Mac. | The mklib in v5.2.7 CGT generates segmentation fault on MacOS X. The error occurs when mklib is invoked both manually via command line as well as automatically via CCS build. |

**Generated on Thu Feb 2 16:39:38 2017**