PARASOFT C/C++test® Report

Builtin configuration: **MISRA C 2023 (MISRA C 2012)**

2025-05-19T10:57:54-04:00

STATIC ANALYSIS

Project Name	Tasks				Files		Lines	
	suppressed	qfix	total	per 10,000 lines	checked	total	checked	total
Timer	0	0	203	5205	8	8	390	390
Total [0:00:31]	**0**	**0**	**203**	**5205**	**8**	**8**	**390**	**390**

All Tasks by Category

[1] **Rule 8.6 (Required) An identifier with external linkage shall have exactly one external definition** (MISRAC2012-RULE_8_6)

[1] An identifier with external linkage shall have external definition (MISRAC2012-RULE_8_6-b-2)

[1] **Rule 11.5 (Advisory) A conversion should not be performed from pointer to void into pointer to object** (MISRAC2012-RULE_11_5)

[1] A conversion should not be performed from pointer to void into pointer to object (MISRAC2012-RULE_11_5-a-4)

[4] **Rule 8.4 (Required) A compatible declaration shall be visible when an object or function with external linkage is defined** (MISRAC2012-RULE_8_4)

[4] A declaration shall be visible when an object or function with external linkage is defined (MISRAC2012-RULE_8_4-a-2)

[2] **Rule 11.9 (Required) The macro NULL shall be the only permitted form of integer null pointer constant** (MISRAC2012-RULE_11_9)

[1] Literal zero (0) shall not be used as the null-pointer-constant (MISRAC2012-RULE_11_9-a-2)

[1] Use NULL instead of literal zero (0) as the null-pointer-constant (MISRAC2012-RULE_11_9-b-2)

[6] **Rule 8.7 (Advisory) Functions and objects should not be defined with external linkage if they are referenced in only one translation unit** (MISRAC2012-RULE_8_7)

[6] Functions and objects should not be defined with external linkage if they are referenced in only one translation unit (MISRAC2012-RULE_8_7-a-4)

[2] **Dir 4.12 (Required) Dynamic memory allocation shall not be used** (MISRAC2012-DIR_4_12)

[2] Dynamic heap memory allocation shall not be used (MISRAC2012-DIR_4_12-a-2)

[22] **Rule 8.2 (Required) Function types shall be in prototype form with named parameters** (MISRAC2012-RULE_8_2)

[5] Identifiers shall be given for all of the parameters in a function prototype declaration (MISRAC2012-RULE_8_2-a-2)

[17] Function types shall be in prototype form (MISRAC2012-RULE_8_2-c-2)

[1] **Rule 10.4 (Required) Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category** (MISRAC2012-RULE_10_4)

[1] Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category (MISRAC2012-RULE_10_4-a-2)

[26] **Rule 7.4 (Required) A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char"** (MISRAC2012-RULE_7_4)

[26] A string literal shall not be modified (MISRAC2012-RULE_7_4-a-2)

[1] **Rule 18.1 (Required) A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand** (MISRAC2012-RULE_18_1)

[1] Avoid accessing arrays out of bounds (MISRAC2012-RULE_18_1-a-2)

[1] **Rule 10.3 (Required) The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category** (MISRAC2012-RULE_10_3)

[1] The value of an expression shall not be assigned to an object of a different essential type category (MISRAC2012-RULE_10_3-b-2)

[3] **Rule 21.6 (Required) The Standard Library input/output functions shall not be used** (MISRAC2012-RULE_21_6)

[3] The input/output functions from the 'cstdio' and 'cwchar' libraries should not be used (MISRAC2012-RULE_21_6-a-2)

[1] **Rule 21.7 (Required) The atof, atoi, atol and atoll functions of <stdlib.h> shall not be used** (MISRAC2012-RULE_21_7)

[1] The 'atof', 'atoi', 'atol' and 'atoll' functions from the 'stdlib.h' or 'cstdlib' library should not be used (MISRAC2012-RULE_21_7-a-2)

[4] **Rule 21.1 (Required) #define and #undef shall not be used on a reserved identifier or reserved macro name** (MISRAC2012-RULE_21_1)

[4] Do not #define or #undef identifiers with names which start with underscore (MISRAC2012-RULE_21_1-a-2)

[2] **Rule 21.3 (Required) The memory allocation and deallocation functions of <stdlib.h> shall not be used** (MISRAC2012-RULE_21_3)

[2] Dynamic heap memory allocation shall not be used (MISRAC2012-RULE_21_3-a-2)

[1] **Rule 13.3 (Advisory) A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator** (MISRAC2012-RULE_13_3)

[1] A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects (MISRAC2012-RULE_13_3-a-4)

[1] **Rule 2.7 (Advisory) A function should not contain unused parameters** (MISRAC2012-RULE_2_7)

[1] There should be no unused parameters in functions (MISRAC2012-RULE_2_7-a-4)

[5] **Rule 2.8 (Advisory) A project should not contain unused object definitions** (MISRAC2012-RULE_2_8)

[5] A project should not contain unused local variables (MISRAC2012-RULE_2_8-c-4)

[1] **Dir 4.8 (Advisory) If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden** (MISRAC2012-DIR_4_8)

[1] If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden (MISRAC2012-DIR_4_8-a-4)

[1] **Dir 4.7 (Required) If a function returns error information, then that error information shall be tested** (MISRAC2012-DIR_4_7)

[1] Always check the returned value of non-void function (MISRAC2012-DIR_4_7-b-2)

[22] **Dir 4.6 (Advisory) typedefs that indicate size and signedness should be used in place of the basic numerical types** (MISRAC2012-DIR_4_6)

[22] typedefs should be used in place of the basic types (MISRAC2012-DIR_4_6-b-4)

[2] **Dir 4.1 (Required) Run-time failures shall be minimized** (MISRAC2012-DIR_4_1)

[1] Avoid accessing arrays out of bounds (MISRAC2012-DIR_4_1-a-2)

[1] Avoid null pointer dereferencing (MISRAC2012-DIR_4_1-b-2)

[1] **Rule 8.13 (Advisory) A pointer should point to a const-qualified type whenever possible** (MISRAC2012-RULE_8_13)

[1] A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object (MISRAC2012-RULE_8_13-a-4)

[17] **Rule 1.5 (Required) Obsolescent language features shall not be used** (MISRAC2012-RULE_1_5)

[17] Function types shall be in prototype form (MISRAC2012-RULE_1_5-c-2)

[3] **Rule 15.5 (Advisory) A function should have a single point of exit at the end** (MISRAC2012-RULE_15_5)

[3] A function shall have a single point of exit at the end of the function (MISRAC2012-RULE_15_5-a-4)

[1] **Rule 14.3 (Required) Controlling expressions shall not be invariant** (MISRAC2012-RULE_14_3)

[1] Avoid conditions that always evaluate to the same value (MISRAC2012-RULE_14_3-ac-2)

[2] **Rule 14.4 (Required) The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type** (MISRAC2012-RULE_14_4)

[2] Tests of a value against zero should be made explicit, unless the operand is effectively Boolean (MISRAC2012-RULE_14_4-a-2)

[19] **Rule 21.10 (Required) The Standard Library time and date functions shall not be used** (MISRAC2012-RULE_21_10)

[2] The standard header files <time.h> or <ctime> shall not be used (MISRAC2012-RULE_21_10-a-2)

[10] The time handling functions and macros of the library <time.h> shall not be used (MISRAC2012-RULE_21_10-b-2)

[7] The types defined in the library <time.h> shall not be used (MISRAC2012-RULE_21_10-c-2)

[15] **Rule 17.3 (Mandatory) A function shall not be declared implicitly** (MISRAC2012-RULE_17_3)

[15] Functions shall always have visible prototype at the function call (MISRAC2012-RULE_17_3-a-1)

[35] **Rule 17.7 (Required) The value returned by a function having non-void return type shall be used** (MISRAC2012-RULE_17_7)

[35] The value returned by a function having non-void return type shall be used (MISRAC2012-RULE_17_7-a-2)

[15] **Severity 1 - Highest**

[15] Functions shall always have visible prototype at the function call (MISRAC2012-RULE_17_3-a-1)

[147] **Severity 2 - High**

[1] Avoid accessing arrays out of bounds (MISRAC2012-DIR_4_1-a-2)

[1] Avoid null pointer dereferencing (MISRAC2012-DIR_4_1-b-2)

[2] Dynamic heap memory allocation shall not be used (MISRAC2012-DIR_4_12-a-2)

[1] Always check the returned value of non-void function (MISRAC2012-DIR_4_7-b-2)

[1] The value of an expression shall not be assigned to an object of a different essential type category (MISRAC2012-RULE_10_3-b-2)

[1] Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category (MISRAC2012-RULE_10_4-a-2)

[1] Literal zero (0) shall not be used as the null-pointer-constant (MISRAC2012-RULE_11_9-a-2)

[1] Use NULL instead of literal zero (0) as the null-pointer-constant (MISRAC2012-RULE_11_9-b-2)

[1] Avoid conditions that always evaluate to the same value (MISRAC2012-RULE_14_3-ac-2)

[2] Tests of a value against zero should be made explicit, unless the operand is effectively Boolean (MISRAC2012-RULE_14_4-a-2)

[35] The value returned by a function having non-void return type shall be used (MISRAC2012-RULE_17_7-a-2)

[1] Avoid accessing arrays out of bounds (MISRAC2012-RULE_18_1-a-2)

[17] Function types shall be in prototype form (MISRAC2012-RULE_1_5-c-2)

[4] Do not #define or #undef identifiers with names which start with underscore (MISRAC2012-RULE_21_1-a-2)

[2] The standard header files <time.h> or <ctime> shall not be used (MISRAC2012-RULE_21_10-a-2)

[10] The time handling functions and macros of the library <time.h> shall not be used (MISRAC2012-RULE_21_10-b-2)

[7] The types defined in the library <time.h> shall not be used (MISRAC2012-RULE_21_10-c-2)

[2] Dynamic heap memory allocation shall not be used (MISRAC2012-RULE_21_3-a-2)

[3] The input/output functions from the 'cstdio' and 'cwchar' libraries should not be used (MISRAC2012-RULE_21_6-a-2)

[1] The 'atof', 'atoi', 'atol' and 'atoll' functions from the 'stdlib.h' or 'cstdlib' library should not be used (MISRAC2012-RULE_21_7-a-2)

[26] A string literal shall not be modified (MISRAC2012-RULE_7_4-a-2)

[5] Identifiers shall be given for all of the parameters in a function prototype declaration (MISRAC2012-RULE_8_2-a-2)

[17] Function types shall be in prototype form (MISRAC2012-RULE_8_2-c-2)

[4] A declaration shall be visible when an object or function with external linkage is defined (MISRAC2012-RULE_8_4-a-2)

[1] An identifier with external linkage shall have external definition (MISRAC2012-RULE_8_6-b-2)

[41] **Severity 4 - Low**

[22] typedefs should be used in place of the basic types (MISRAC2012-DIR_4_6-b-4)

[1] If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden (MISRAC2012-DIR_4_8-a-4)

[1] A conversion should not be performed from pointer to void into pointer to object (MISRAC2012-RULE_11_5-a-4)

[1] A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects (MISRAC2012-RULE_13_3-a-4)

[3] A function shall have a single point of exit at the end of the function (MISRAC2012-RULE_15_5-a-4)

[1] There should be no unused parameters in functions (MISRAC2012-RULE_2_7-a-4)

[5] A project should not contain unused local variables (MISRAC2012-RULE_2_8-c-4)

[1] A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object (MISRAC2012-RULE_8_13-a-4)

[6] Functions and objects should not be defined with external linkage if they are referenced in only one translation unit (MISRAC2012-RULE_8_7-a-4)

Tasks by Author

Author	Tasks	
	suppressed	total
syoung	0	203

Test Parameters

C++test: -config c++test.builtin://MISRA C 2023 (MISRA C 2012) -report C:\Users\syoung\Documents\CppTestEclipse\cpptest\examples\Timer\reports