

**MISRA C:2012 GUIDELINES FOR THE USE  
OF THE C LANGUAGE IN CRITICAL  
SYSTEMS(MARCH 2013) | CODESONAR®  
6.2**

**INCLUDING MISRA C:2012 AMENDMENT 1 ADDITIONAL  
SECURITY GUIDELINES FOR MISRA C:2012 (APRIL 2016)**



## INTRODUCTION

The MISRA C:2012 standard aims to foster safety, reliability, and portability of programs written in ISO C for embedded systems. It is used in a wide range of industries, including automotive, aero-space, medical devices, and industrial control.

CodeSonar 6.2 includes a large number of warning classes that support checking for the MISRA C:2012 guidelines. Every CodeSonar warning report includes the numbers of any MISRA C:2012 rules and directives that are closely mapped to the warning's class. (The close mapping for a warning class is the set of categories—including MISRA C:2012 rule and directive numbers—that most closely match the class, if any).

You can configure CodeSonar to enable and disable warning classes mapped to specific MISRA C:2012 rules and directives, or use build presets to enable all warning classes that are closely mapped to any MISRA C:2012 rules and directives. In addition, you can use the CodeSonar search function to find warnings related to specific MISRA C:2012 rules or directives, or to any MISRA C:2012 rule or directive.

For more information on MISRA C:

<https://www.misra.org.uk/MISRAHome/tabid/181/Default.aspx>

The following table contains CodeSonar classes that are closely mapped to specific MISRA C:2012 rules and directives.

Note-All CodeSonar MISRA mappings are close.

GrammaTech is a leading global provider of application testing (AST) solutions used by the world's most security conscious organizations to detect, measure, analyze and resolve vulnerabilities for software they develop or use. The company is also a trusted cybersecurity and artificial intelligence research partner for the nation's civil, defense, and intelligence agencies.

CodeSonar and CodeSentry are registered trademarks of GrammaTech, Inc.  
© GrammaTech, Inc. All rights reserved.



MISRA C:2012 ID	Closely Mapped CodeSonar 6.2 Classes
1.2	C++ Comment in CGNU Extension GNU Typeof Microsoft Extension
1.3	Float Division By Zero Negative Shift Amount Null Pointer Dereference Type Underrun Buffer Underrun Buffer Overrun Pointer Past End of Object Dangerous Function Cast Uninitialized Variable Division By Zero Type Overrun Shift Amount Exceeds Bit Width Pointer Before Beginning of ObjectUnterminated C String
2.1	Unexercised Call Unexercised ComputationUnexercised Conditional Unexercised Control FlowUnexercised Data Flow Unreachable Call Unreachable Computation Unreachable Conditional Unreachable Control FlowUnreachable Data Flow
2.2	Function Call Has No Effect Unused Value Useless Assignment
2.3	Unused Type
2.4	Unused Tag
2.5	Unused Macro
2.6	Unused Label
2.7	Unused Parameter
3.1	/* in Comment // in Comment
3.2	Line Splicing in Comment

4.1	Unterminated Escape Sequence
4.2	Trigraph
4.9	Function-Like Macro

4.11	Addition Overflow of Allocation Size Arctangent Domain Error Argument Too High Argument Too Low Floating Point Domain ErrorFloating Point Range Error Gamma on Zero Integer Overflow of Allocation SizeLogarithm on Negative Value Logarithm on Zero MAX_PATH Exceeded Raises FE_INVALID Undefined Power of Zerocosh on High Number cosh on Low Number sqrt on Negative Value
4.13	chroot without chdir
4.14	Tainted Environment Variable
5.1	Non-distinct Identifiers: External Names
5.2	Non-distinct Identifiers: Same Scope
5.3	Non-distinct Identifiers: Nested Scope
5.4	Non-distinct Identifiers: Macro/Macro
5.5	Non-distinct Identifiers: Macro/Other
5.6	Non-unique Identifiers: Typedef
5.7	Non-unique Identifiers: Tag
5.8	Library Function Override Non-unique Identifiers: External Name
5.9	Library Function Override Non-unique Identifiers: Internal Name
6.1	Bit-field Signedness Not ExplicitInappropriate Bit-field Type
6.2	Bit-field Too Short
7.1	Octal Constant
7.2	Missing Literal Suffix
7.3	Confusing Literal Suffix
7.4	Non-const String Literal
8.2	Incomplete Function Prototype
8.3	Global Variable Declared with Different TypesInconsistent Function Declarations Inconsistent Object Declarations

8.5	Missing External Declaration Multiple Declarations of a Global Multiple External Declarations
8.6	Missing External Definition Multiple External Definitions
8.7	Scope Could Be File Static Scope Could Be Local Static
8.8	Scope Could Be File Static Scope Could Be Local Static
8.9	Scope Could Be Local Static
8.10	Inline Function Not static
8.11	Extern Array Without Size
8.12	Inconsistent Enumerator Initialization
8.13	Pointed-to Type Could Be const
8.14	Restrict Qualifier Used
9.1	Uninitialized Variable
9.2	Missing Braces in Initialization
9.3	Partially Uninitialized Array
9.4	Over-initialized Element
9.5	Unspecified Array Size with Designator Initialization
10.1	Inappropriate Operand Type
10.2	Inappropriate Character Arithmetic
10.3	Inappropriate Assignment Type Cast Alters Value Coercion Alters Value
10.4	Mismatched Operand Types
10.5	Inappropriate Cast Type
10.6	Expression Value Widened by Assignment
10.7	Expression Value Widened by Other Operand
10.8	Inappropriate Cast Type: Expression
11.1	Conversion from Function Pointer Conversion to Function Pointer Dangerous Function Cast Function Pointer Conversion
11.2	Conversion: Pointer to Incomplete
11.3	Cast: Object Pointers
11.4	Conversion: Pointer/Integer
11.5	Conversion: Void Pointer to Object Pointer
11.6	Cast: Arithmetic Type/Void Pointer
11.7	Cast: Non-integer Arithmetic Type/Object Pointer

11.8	Cast Removes const Qualifier Cast Removes volatile Qualifier
11.9	Coercion: Integer Constant to Pointer



12.1	Missing Parentheses
12.2	Negative Shift Amount Shift Amount Exceeds Bit Width
12.3	Use of Comma Operator
12.4	Coercion Alters Value
13.1	Side Effects in Initializer List
13.3	Side Effects in Expression with Decrement Side Effects in Expression with Increment
13.4	Assignment Result in Expression
13.5	Side Effects in Logical Operand
13.6	Side Effects in sizeof
14.1	Float-typed Loop Counter
14.2	Malformed for-loop Condition Malformed for-loop Initialization Malformed for-loop Step Missing for-loop Step Missing for-loop Termination
14.3	Redundant Condition
14.4	Condition Is Not Boolean
15.1	Goto Statement
15.2	Backwards goto
15.3	Label Not In Enclosing Block
15.4	Multiple Abnormal Loop Exits
15.5	Misplaced Return Statement Multiple Return Statements
15.6	Body Is Not Compound Statement
15.7	Missing Final else
16.1	Malformed switch Statement
16.2	Misplaced case
16.3	Missing break
16.4	Missing default
16.5	Misplaced default
16.6	Too Few Cases in switch
16.7	Boolean switch Expression
17.1	Use of <stdarg.h> Feature
17.2	Recursion
17.3	Implicit Function Declaration
17.4	Missing Return Statement
17.5	Array Parameter Mismatch



17.6	Static Array Parameter
17.7	Ignored Return Value
17.8	Modified Parameter
18.1	Buffer Overrun Buffer Underrun Pointer Before Beginning of Object Pointer Past End of Object Tainted Buffer Access Type Overrun Type Underrun
18.2	Buffer Overrun Buffer Underrun Pointer Before Beginning of Object Pointer Past End of Object Tainted Buffer Access Type Overrun Type Underrun Tainted Buffer Access Pointer Before Beginning of Object Subtraction of Unrelated Pointers
18.3	Pointed-to Type Could Be const Comparison of Unrelated Pointers
18.4	Pointer Arithmetic
18.5	Too Much Indirection in Declaration
18.6	Return Pointer to Local
18.7	Declaration of Flexible Array Member
18.8	Declaration of Variable Length Array
19.1	Overlapping Memory Regions
19.2	Union Type
20.1	Code Before #include
20.2	Dangerous Include File Name
20.3	Malformed #include
20.4	Macro Name is C Keyword
20.5	Macro Undefined of Reserved Name Use of #undef
20.6	Preprocessing Directives in Macro Argument
20.7	Macro Parameter Not Parenthesized
20.8	Non-Boolean Preprocessor Expression
20.9	Undefined Macro in #if
20.10	Macro Uses # Operator
20.11	## Follows # Operator
20.13	Invalid Preprocessor Directive
20.14	No Matching #endif No Matching #if

21.1	Macro Definition of Reserved Name Macro Undefinition of Reserved Name
------	--



21.2	Declaration of Reserved Name
21.3	Dynamic Allocation After Initialization Use of <stdlib.h> Allocator/Deallocator MacroUse of <stdlib.h> Allocator/Deallocator
21.4	Use of <setjmp.h> Use of longjmp Use of setjmp
21.5	Use of <signal.h>
21.6	Use of <stdio.h> Input/Output Use of <stdio.h> Input/Output Macro Use of <wchar.h> Input/Output Use of <wchar.h> Input/Output Macro
21.7	Use of atof Use of atoi Use of atol Use of atoll
21.8	Use of abort Use of exit Use of system
21.9	Use of bsearch Use of qsort
21.10	Potential Timebomb Use of <time.h> Time/Date Function
21.11	Use of <tgmath.h>
21.12	Use of <fenv.h> Exception Handling Function
21.13	Negative Character Value

21.17	Buffer Overrun Buffer Underrun Type Overrun Type Underrun Use of strcat Use of strcmp Use of strcpy Use of strlen Truncation of Size Multiplication Overflow of Size Use of strstr Use of strpbrk Use of strchr Use of strchr Use of strcoll Use of strspn Use of strcspn
21.18	Addition Overflow of Size Subtraction Underflow of SizeUnreasonable Size ArgumentUnreasonable Size ArgumentAddition Overflow of Size
21.19	Use of getenv
22.1	Leak
22.4	Write to Read Only File
22.5	FILE* Dereference
22.6	Use After Close
22.7	Negative Character Value
D.4.3	Inline Assembly Code Mixed Assembly and Code
D.4.4	Commented-out Code
D.4.5	Typographically Ambiguous Identifiers
D.4.6	Basic Numerical Type Used
D.4.7	Ignored Return Value
D.4.12	Dynamic Allocation After Initialization

D.4.1	Leak Negative file descriptor Double Close Negative Character Value Use of abort cosh on Low Number Arctangent Domain Error Float Division By Zero Floating Point Domain Error Null Pointer Dereference Type Underrun Buffer Underrun cosh on High NumberArgument Too Low Buffer Overrun Double Unlock Logarithm on Negative Value Uninitialized Variable Floating Point Range Error Type Overrun Double Lock Double Free Free Null Pointer sqrt on Negative Value Logarithm on Zero Argument Too High Gamma on Zero
D.4.13	Socket In Wrong State
D.4.14	Command Injection Format String InjectionIgnored Return Value LDAP Injection Library InjectionSQL Injection Tainted Allocation Size Tainted Configuration SettingTainted Filename Tainted Network AddressTainted Write Untrusted Library Load Untrusted Network Host Untrusted Network Port Untrusted Process Creation