

MISRA HIS Subset

The MISRA Hersteller Initiative Software (HIS) Subset.

The MISRA checks used by the Hersteller Initiative Software (HIS) metrics:

- NOMV: Number of MISRA HIS Subset violations
- NOMVPR: Number of MISRA violations per rule

Enable this configuration in the background or run it manually and sync the results to the Violation Browser to calculate the metric(s).

Checks

Check ID	Check Name	Supported	Category
MISRA12_1.1	1.1 The program shall contain no violations of the standard C syntax and constraints, and shall not exceed the implementation's translation limits	Yes	Required
MISRA12_1.2	1.2 Language extensions should not be used (Partial)	Yes	Advisory
MISRA12_1.3	1.3 There shall be no occurrence of undefined or critical unspecified behaviour (Partial)	Yes	Required
MISRA12_2.1	2.1 A project shall not contain unreachable code	Yes	Required
MISRA12_2.2	2.2 There shall be no dead code (Partial)	Yes	Required
MISRA12_3.1	3.1 The character sequences /* and // shall not be used within a comment	Yes	Required
MISRA12_4.1	4.1 Octal and Hexadecimal Sequences	Yes	Required
MISRA12_4.2	4.2 Trigraphs should not be used	Yes	Advisory
MISRA12_5.3	5.3 Shadowed Identifiers	Yes	Required
MISRA12_5.6	5.6 A typedef name shall be a unique identifier	Yes	Required
MISRA12_5.7	5.7 A tag name shall be a unique identifier	Yes	Required
MISRA12_6.2	6.2 Signed single-bit named bit-fields	Yes	Required
MISRA12_7.1	7.1 Octal constants shall not be used	Yes	Required
MISRA12_8.1	8.1 Types shall be explicitly specified	Yes	Required
MISRA12_8.2	8.2 Use Named Parameters and Prototype Form	Yes	Required
MISRA12_8.3	8.3 All declarations of an object or function shall use the same names and type qualifiers	Yes	Required

Check ID	Check Name	Supported	Category
MISRA12_8.4	8.4 A compatible declaration shall be visible when an object or function with external linkage is defined	Yes	Required
MISRA12_8.5	8.5 An external object or function shall be declared once in one and only one file	Yes	Required
MISRA12_8.6	8.6 An identifier with external linkage shall have exactly one external definition	Yes	Required
MISRA12_8.7	8.7 Functions and objects should not be defined with external linkage if they are referenced in only one translation unit	Yes	Advisory
MISRA12_8.8	8.8 Use the static keyword for internal linkage	Yes	Required
MISRA12_8.9	8.9 Objects shall be local if only accessed from one function	Yes	Advisory
MISRA12_8.13	8.13 A pointer should point to a const-qualified type whenever possible	Yes	Advisory
MISRA12_9.1	9.1 The value of an object with automatic storage duration shall not be read before it has been set	Yes	Mandatory
MISRA12_9.2	9.2 The initializer for an aggregate or union shall be enclosed in braces	Yes	Required
MISRA12_10.1	10.1 Operands shall not be of an inappropriate essential type	Yes	Required
MISRA12_10.2	10.2 Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations	Yes	Required
MISRA12_10.3	10.3 The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category	Yes	Required
MISRA12_10.4	10.4 Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category	Yes	Required
MISRA12_10.6	10.6 The value of a composite expression shall not be assigned to an object with wider essential type	Yes	Required
MISRA12_10.7	10.7 Implicit Casts of Operations	Yes	Required
MISRA12_10.8	10.8 The value of a composite expression shall not be cast to a different essential type category or a wider essential type	Yes	Required
MISRA12_11.1	11.1 Conversions shall not be performed	Yes	Required

Check ID	Check Name	Supported Category	
	between a pointer to a function and any other type		
MISRA12_11.3	11.3 A cast shall not be performed between a pointer to object type and a pointer to a different object type	Yes	Required
MISRA12_11.8	11.8 A cast shall not remove any const or volatile qualification from the type pointed to by a pointer	Yes	Required
MISRA12_12.1	12.1 The precedence of operators within expressions should be made explicit	Yes	Advisory
MISRA12_12.2	12.2 The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left hand operand.	Yes	Required
MISRA12_12.3	12.3 The comma operator shall not be used.	Yes	Advisory
MISRA12_12.4	12.4 Evaluation of constant expressions should not lead to unsigned integer wrap-around	Yes	Advisory
MISRA12_13.2	13.2 The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders	Yes	Required
MISRA12_13.4	13.4 The result of an assignment operator should not be used	Yes	Advisory
MISRA12_13.5	13.5 The right hand operand of a logical && or operator shall not contain persistent side effects	Yes	Required
MISRA12_13.6	13.6 The operand of the sizeof operator shall not contain any expression which has potential side effects	Yes	Mandatory
MISRA12_14.1	14.1 A loop counter shall not have essentially floating type	Yes	Required
MISRA12_14.2	14.2 A for loop shall be well-formed	Yes	Required
MISRA12_15.1	15.1 The goto statement should not be used	Yes	Advisory
MISRA12_15.2	15.2 The goto statement shall jump to a label declared later in the same function	Yes	Required
MISRA12_15.3	15.3 Goto Into or Between Blocks	Yes	Required
MISRA12_15.6	15.6 The body of an iteration-statement or a selection-statement shall be a compound-statement	Yes	Required
MISRA12_15.7	15.7 All if ... else if constructs shall be	Yes	Required

Check ID	Check Name	Supported Category	
	terminated with an else statement		
MISRA12_16.2	16.2 A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement	Yes	Required
MISRA12_16.3	16.3 An unconditional break statement shall terminate every switch-clause	Yes	Required
MISRA12_16.4	16.4 Every switch statement shall have a default label	Yes	Required
MISRA12_16.5	16.5 A default label shall appear as either the first or the last switch label of a switch statement	Yes	Required
MISRA12_16.6	16.6 Every switch statement shall have at least two switch-clauses	Yes	Required
MISRA12_16.7	16.7 Switch Boolean	Yes	Required
MISRA12_17.1	17.1 The features of <stdarg.h> shall not be used	Yes	Required
MISRA12_17.2	17.2 Functions shall not call themselves, either directly or indirectly	Yes	Required
MISRA12_17.3	17.3 A function shall not be declared implicitly	Yes	Mandatory
MISRA12_17.4	17.4 Always return a value in non-void functions	Yes	Required
MISRA12_18.1	18.1 A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand (Partial)	Yes	Required
MISRA12_18.2	18.2 Subtraction between pointers shall only be applied to pointers that address elements of the same array	Yes	Required
MISRA12_18.3	18.3 The relational operators >, >=, < and <= shall not be applied to objects of pointer type except where they point into the same object	Yes	Required
MISRA12_18.4	18.4 The +, -, += and -= operators should not be applied to an expression of pointer type	Yes	Advisory
MISRA12_18.5	18.5 Declarations should contain no more than two levels of pointer nesting	Yes	Advisory
MISRA12_18.6	18.6 The address of an object with automatic storage shall not be copied to another object that persists after the first	Yes	Required

Check ID	Check Name	Supported Category	
	object has ceased to exist		
MISRA12_19.1	19.1 An object shall not be assigned or copied to an overlapping object (Partial)	Yes	Mandatory
MISRA12_20.2	20.2 Invalid Header Name	Yes	Required
MISRA12_20.3	20.3 The #include directive shall be followed by either a <filename> or "filename" sequence	Yes	Required
MISRA12_20.4	20.4 Keyword Macros	Yes	Required
MISRA12_20.6	20.6 Tokens that look like a preprocessing directive shall not occur within a macro argument	Yes	Required
MISRA12_20.7	20.7 Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses	Yes	Required
MISRA12_20.9	20.9 All identifiers used in the controlling expression of #if or #elif preprocessing directives shall be #define'd before evaluation	Yes	Required
MISRA12_20.10	20.10 The # and ## operators should not be used	Yes	Advisory
MISRA12_20.11	20.11 A macro parameter immediately following a # operator shall not immediately be followed by a ## operator	Yes	Required
MISRA12_21.1	21.1 #define and #undef shall not be used on a reserved identifier or reserved macro name	Yes	Required
MISRA12_21.2	21.2 Reserved Identifiers or Macros	Yes	Required
MISRA12_21.3	21.3 The memory allocation and deallocation functions of <stdlib.h> shall not be used	Yes	Required
MISRA12_21.4	21.4 The standard header file <setjmp.h> shall not be used	Yes	Required
MISRA12_21.5	21.5 Standard Header signal.h	Yes	Required
MISRA12_21.6	21.6 C Standard Library I/O Functions	Yes	Required
MISRA12_21.7	21.7 The atof, atoi, atol and atoll functions of <stdlib.h> shall not be used	Yes	Required
MISRA12_21.8	21.8 The library functions abort, exit, getenv and system of <stdlib.h> shall not be used	Yes	Required
MISRA12_21.10	21.10 The Standard Library time and date functions shall not be used	Yes	Required
MISRA12_DIR_1.1	Directive 1.1 Any implementation-defined	No	Required

Check ID	Check Name	Supported Category	
	behaviour on which the output of the program depends shall be documented and understood		
MISRA12_DIR_4.4	Directive 4.4 Sections of code should not be "commented out"	Yes	Advisory
MISRA12_DIR_4.6	Directive 4.6 Typedefs that indicate size and signedness should be used in place of the basic numerical types	Yes	Advisory
MISRA12_DIR_4.7	Directive 4.7 If a function returns error information, then that error information shall be tested	Yes	Required
MISRA12_DIR_4.9	Directive 4.9 A function should be used in preference to a function-like macro where they are interchangeable	Yes	Advisory
MISRA12_DIR_4.11	Directive 4.11 The validity of values passed to library functions shall be checked (Partial)	Yes	Required
MISRA12_DIR_4.12	Directive 4.12 Dynamic Memory Allocation	Yes	Required