

RuleChecker



This page was automatically generated and should not be edited.



The information on this page was provided by outside contributors and has not been verified by SEI CERT.



The table below can be re-ordered, by clicking column headers.

Tool Version: 19.04

Checker	Guideline
alignof-side-effect	EXP44-C. Do not rely on side effects in operands to sizeof, _Alignof, or _Generic
alloc-without-cast	MEM02-C. Immediately cast the result of a memory allocation function call into a pointer to the allocated type
alloc-without-sizeof	EXP09-C. Use sizeof to determine the size of a type or variable
array-index-range-constant	ARR30-C. Do not form or use out-of-bounds pointers or array subscripts
array-size-global	ARR02-C. Explicitly specify array bounds, even if implicitly defined by an initializer
assignment-conditional	EXP45-C. Do not perform assignments in selection statements
assignment-to-non-modifiable-lvalue	EXP40-C. Do not modify constant objects
assignment-to-non-modifiable-lvalue	MSC40-C. Do not violate constraints
bad-function	ERR07-C. Prefer functions that support error checking over equivalent functions that don't
bad-function	WIN01-C. Do not forcibly terminate execution
bitfield-type	INT12-C. Do not make assumptions about the type of a plain int bit-field when used in an expression
bitop-type	INT13-C. Use bitwise operators only on unsigned operands
bitop-type	INT16-C. Do not make assumptions about representation of signed integers
cast-pointer-void-arithmetic-implicit	MSC40-C. Do not violate constraints
chained-comparison	EXP13-C. Treat relational and equality operators as if they were nonassociative
char-sign-conversion	STR34-C. Cast characters to unsigned char before converting to larger integer sizes
compound-ifelse	EXP19-C. Use braces for the body of an if, for, or while statement
compound-loop	EXP19-C. Use braces for the body of an if, for, or while statement
ctype-limits	STR37-C. Arguments to character-handling functions must be representable as an unsigned char
distinct-extern	DCL40-C. Do not create incompatible declarations of the same function or object
empty-body	EXP15-C. Do not place a semicolon on the same line as an if, for, or while statement
empty-parameter-list	DCL20-C. Explicitly specify void when a function accepts no arguments
encoding-mismatch	STR10-C. Do not concatenate different type of string literals
enum-implicit-value	INT09-C. Ensure enumeration constants map to unique values
error-information-unused	ERR33-C. Detect and handle standard library errors
error-information-unused	EXP12-C. Do not ignore values returned by functions
evaluation-order	EXP30-C. Do not depend on the order of evaluation for side effects
evaluation-order	EXP10-C. Do not depend on the order of evaluation of subexpressions or the order in which side effects take place
file-dereference	FIO38-C. Do not copy a FILE object
flexible-array-member-assignment	MEM33-C. Allocate and copy structures containing a flexible array member dynamically
flexible-array-member-declaration	MEM33-C. Allocate and copy structures containing a flexible array member dynamically
float-comparison	FLP02-C. Avoid using floating-point numbers when precise computation is needed
for-loop-float	FLP30-C. Do not use floating-point variables as loop counters
function-name-constant-comparison	EXP16-C. Do not compare function pointers to constant values

function-pointer-integer-cast	INT36-C. Converting a pointer to integer or integer to pointer
function-pointer-integer-cast-implicit	INT36-C. Converting a pointer to integer or integer to pointer
function-pointer-integer-cast-implicit	MSC40-C. Do not violate constraints
function-prototype	DCL07-C. Include the appropriate type information in function declarators
function-return-type	DCL31-C. Declare identifiers before using them
function-return-type	MSC40-C. Do not violate constraints
future-library-use	DCL37-C. Do not declare or define a reserved identifier
generic-selection-side-effect	EXP44-C. Do not rely on side effects in operands to sizeof, _Alignof, or _Generic
global-object-scope	DCL15-C. Declare file-scope objects or functions that do not need external linkage as static
global-object-scope	DCL19-C. Minimize the scope of variables and functions
implicit-function-declaration	DCL31-C. Declare identifiers before using them
implicit-function-declaration	DCL07-C. Include the appropriate type information in function declarators
inappropriate-bool	EXP46-C. Do not use a bitwise operator with a Boolean-like operand
inappropriate-pointer-cast-implicit	MSC40-C. Do not violate constraints
incompatible-function-pointer-conversion	MSC40-C. Do not violate constraints
incompatible-object-pointer-conversion	MSC40-C. Do not violate constraints
initializer-excess	MSC40-C. Do not violate constraints
invalid-array-size	MSC40-C. Do not violate constraints
language-override	DCL37-C. Do not declare or define a reserved identifier
language-override-c99	DCL37-C. Do not declare or define a reserved identifier
literal-assignment	STR05-C. Use pointers to const when referring to string literals
local-object-scope	DCL19-C. Minimize the scope of variables and functions
logop-side-effect	EXP02-C. Be aware of the short-circuit behavior of the logical AND and OR operators
long-suffix	DCL16-C. Use "L," not "l," to indicate a long value
macro-argument-hash	PRE32-C. Do not use preprocessor directives in invocations of function-like macros
macro-final-semicolon	PRE11-C. Do not conclude macro definitions with a semicolon
macro-function-like	PRE00-C. Prefer inline or static functions to function-like macros
macro-parameter-multiplied	PRE12-C. Do not define unsafe macros
macro-parameter-parentheses	PRE01-C. Use parentheses within macros around parameter names
macro-parameter-unused	PRE12-C. Do not define unsafe macros
malloc-size-insufficient	MEM35-C. Allocate sufficient memory for an object
memcmp-with-float	FLP37-C. Do not use object representations to compare floating-point values
memcpy-with-padding	EXP42-C. Do not compare padding data
missing-else	MSC01-C. Strive for logical completeness
mmline-comment	MSC04-C. Use comments consistently and in a readable fashion
multiple-atomic-accesses	CON40-C. Do not refer to an atomic variable twice in an expression
multiple-include	PRE06-C. Enclose header files in an include guard
multiple-volatile-accesses	EXP30-C. Do not depend on the order of evaluation for side effects
multiple-volatile-accesses	EXP10-C. Do not depend on the order of evaluation of subexpressions or the order in which side effects take place
named-declaration-parameter	API08-C. Avoid parameter names in a function prototype
octal-constant	DCL18-C. Do not begin integer constants with 0 when specifying a decimal value
parameter-match	EXP37-C. Call functions with the correct number and type of arguments
parameter-match-type	EXP37-C. Call functions with the correct number and type of arguments
parameter-match-type	MSC40-C. Do not violate constraints
parameter-missing-const	DCL00-C. Const-qualify immutable objects
parameter-missing-const	DCL13-C. Declare function parameters that are pointers to values not changed by the function as const

pointer-cast-alignment	EXP36-C. Do not cast pointers into more strictly aligned pointer types
pointer-integral-cast	INT36-C. Converting a pointer to integer or integer to pointer
pointer-integral-cast-implicit	INT36-C. Converting a pointer to integer or integer to pointer
pointer-integral-cast-implicit	MSC40-C. Do not violate constraints
pointer-qualifier-cast-const	EXP40-C. Do not modify constant objects
pointer-qualifier-cast-const	EXP05-C. Do not cast away a const qualification
pointer-qualifier-cast-const-implicit	EXP40-C. Do not modify constant objects
pointer-qualifier-cast-const-implicit	MSC40-C. Do not violate constraints
pointer-qualifier-cast-const-implicit	EXP05-C. Do not cast away a const qualification
pointer-qualifier-cast-volatile	EXP32-C. Do not access a volatile object through a nonvolatile reference
pointer-qualifier-cast-volatile-implicit	EXP32-C. Do not access a volatile object through a nonvolatile reference
pointer-qualifier-cast-volatile-implicit	MSC40-C. Do not violate constraints
pointer-typedef	DCL05-C. Use typedefs of non-pointer types only
precision-shift-width-constant	INT34-C. Do not shift an expression by a negative number of bits or by greater than or equal to the number of bits that exist in the operand
redeclaration	MSC40-C. Do not violate constraints
reserved-declaration	DCL37-C. Do not declare or define a reserved identifier
reserved-declaration-c99	DCL37-C. Do not declare or define a reserved identifier
reserved-identifier	DCL37-C. Do not declare or define a reserved identifier
restrict	EXP43-C. Avoid undefined behavior when using restrict-qualified pointers
return-implicit	MSC37-C. Ensure that control never reaches the end of a non-void function
scaled-pointer-arithmetic	ARR39-C. Do not add or subtract a scaled integer to a pointer
signal-handler-shared-access	SIG31-C. Do not access shared objects in signal handlers
signal-handler-signal-call	SIG34-C. Do not call signal() from within interruptible signal handlers
signal-handler-unsafe-call	SIG30-C. Call only asynchronous-safe functions within signal handlers
sizeof	EXP44-C. Do not rely on side effects in operands to sizeof, _Alignof, or _Generic
sizeof-array-parameter	ARR01-C. Do not apply the sizeof operator to a pointer when taking the size of an array
sline-splicing	MSC04-C. Use comments consistently and in a readable fashion
smline-comment	MSC04-C. Use comments consistently and in a readable fashion
smline-comment	MSC04-C. Use comments consistently and in a readable fashion
statement-sideeffect	MSC12-C. Detect and remove code that has no effect or is never executed
static-function-declaration	DCL36-C. Do not declare an identifier with conflicting linkage classifications
static-object-declaration	DCL36-C. Do not declare an identifier with conflicting linkage classifications
stdlib-const-pointer-assign	ENV30-C. Do not modify the object referenced by the return value of certain functions
stdlib-limits	FLP32-C. Prevent or detect domain and range errors in math functions
stdlib-macro-ato	MSC24-C. Do not use deprecated or obsolescent functions
stdlib-macro-atoll	MSC24-C. Do not use deprecated or obsolescent functions
stdlib-use-ato	MSC24-C. Do not use deprecated or obsolescent functions
stdlib-use-atoll	MSC24-C. Do not use deprecated or obsolescent functions
stdlib-use-rand	MSC30-C. Do not use the rand() function for generating pseudorandom numbers
stdlib-use-signal	CON37-C. Do not call signal() in a multithreaded program
stdlib-use-system	ENV33-C. Do not call system()
stream-argument-with-side-effects	FIO41-C. Do not call getc(), putc(), getwc(), or putwc() with a stream argument that has side effects
string-literal-modification	STR30-C. Do not attempt to modify string literals
Supported indirectly via MISRA C:2004 rule 6.1.	STR04-C. Use plain char for characters in the basic character set
Supported indirectly via MISRA C:2004 rule 6.1 and MISRA C:2012 rule 10.1.	STR00-C. Represent characters using an appropriate type
Supported indirectly via MISRA C:2004 Rule 17.4.	ARR37-C. Do not add or subtract an integer to a pointer to a non-array object

Supported indirectly via MISRA C:2012 Rule 5.3.	DCL01-C. Do not reuse variable names in subscopes
Supported indirectly via MISRA C:2012 rule 10.1.	STR09-C. Don't assume numeric values for expressions with type plain character
Supported indirectly via MISRA C:2012 Rules 5.1, 5.2, 5.3, 5.4 and 5.5.	DCL23-C. Guarantee that mutually visible identifiers are unique
Supported indirectly via MISRA C:2012 rules 10.1, 10.3 and 10.4.	INT07-C. Use only explicitly signed or unsigned char type for numeric values
Supported via MISRA C:2012 Rules 10.1, 10.3, 10.4, 10.6 and 10.7	INT31-C. Ensure that integer conversions do not result in lost or misinterpreted data
Supported, but no explicit checker	MSC33-C. Do not pass invalid data to the asctime() function
Supported, but no explicit checker	MSC38-C. Do not treat a predefined identifier as an object if it might only be implemented as a macro
switch-clause-break	MSC17-C. Finish every set of statements associated with a case label with a break statement
switch-default	MSC01-C. Strive for logical completeness
switch-label	MSC20-C. Do not use a switch statement to transfer control into a complex block
switch-skipped-code	DCL41-C. Do not declare variables inside a switch statement before the first case label
trigraph	PRE07-C. Avoid using repeated question marks
type-compatibility	DCL40-C. Do not create incompatible declarations of the same function or object
type-compatibility-link	DCL40-C. Do not create incompatible declarations of the same function or object
type-specifier	DCL31-C. Declare identifiers before using them
type-specifier	MSC40-C. Do not violate constraints
undeclared-parameter	DCL31-C. Declare identifiers before using them
undeclared-parameter	MSC40-C. Do not violate constraints
uninitialized-local-read	EXP33-C. Do not read uninitialized memory
universal-character-name-concatenation	PRE30-C. Do not create a universal character name through concatenation
wide-narrow-string-cast	STR38-C. Do not confuse narrow and wide character strings and functions
wide-narrow-string-cast-implicit	STR38-C. Do not confuse narrow and wide character strings and functions