



497c9c4

autware / .clang-tidy

Go to file



yhisaki feat(.clang-tidy): add ExtraArgs (#5439) ✓

3591106 · last week



462 lines (457 loc) · 16.2 KB ·



```

1 Checks: "
2     -*,
3     boost-use-to-string,
4     bugprone-argument-comment,
5     bugprone-assert-side-effect,
6     bugprone-bad-signal-to-kill-thread,
7     bugprone-bool-pointer-implicit-conversion,
8     bugprone-branch-clone,
9     bugprone-copy-constructor-init,
10    bugprone-dangling-handle,
11    bugprone-dynamic-static-initializers,
12    bugprone-exception-escape,
13    bugprone-fold-init-type,
14    bugprone-forward-declaration-namespace,
15    bugprone-forwarding-reference-overload,
16    bugprone-inaccurate-erase,
17    bugprone-incorrect-roundings,
18    bugprone-integer-division,
19    bugprone-lambda-function-name,
20    bugprone-macro-parentheses,
21    bugprone-macro-repeated-side-effects,
22    bugprone-misplaced-operator-in-strlen-in-alloc,
23    bugprone-misplaced-widening-cast,
24    bugprone-move-forwarding-reference,
25    bugprone-multiple-statement-macro,
26    bugprone-not-null-terminated-result,
27    bugprone-parent-virtual-call,
28    bugprone-posix-return,
29    bugprone-signed-char-misuse,
30    bugprone-sizeof-container,
31    bugprone-sizeof-expression,
32    bugprone-string-constructor,
33    bugprone-string-integer-assignment,
34    bugprone-string-literal-with-embedded-nul,
35    bugprone-suspicious-enum-usage,
36    bugprone-suspicious-memset-usage,
37    bugprone-suspicious-missing-comma,
38    bugprone-suspicious-semicolon,
39    bugprone-suspicious-string-compare,
40    bugprone-swapped-arguments,
```

41 bugprone-terminating-continue,
42 bugprone-throw-keyword-missing,
43 bugprone-too-small-loop-variable,
44 bugprone-unchecked-optional-access,
45 bugprone-undefined-memory-manipulation,
46 bugprone-undelegated-constructor,
47 bugprone-unhandled-self-assignment,
48 bugprone-unused-raii,
49 bugprone-unused-return-value,
50 bugprone-use-after-move,
51 bugprone-virtual-near-miss,
52 cppcoreguidelines-avoid-goto,
53 cppcoreguidelines-init-variables,
54 cppcoreguidelines-interfaces-global-init,
55 cppcoreguidelines-macro-usage,
56 cppcoreguidelines-narrowing-conversions,
57 cppcoreguidelines-no-malloc,
58 cppcoreguidelines-pro-bounds-pointer-arithmetic,
59 cppcoreguidelines-pro-type-const-cast,
60 cppcoreguidelines-pro-type-cstyle-cast,
61 cppcoreguidelines-pro-type-member-init,
62 cppcoreguidelines-pro-type-reinterpret-cast,
63 cppcoreguidelines-pro-type-static-cast-downcast,
64 cppcoreguidelines-pro-type-union-access,
65 cppcoreguidelines-slicing,
66 cppcoreguidelines-special-member-functions,
67 google-build-explicit-make-pair,
68 google-build-namespaces,
69 google-build-using-namespace,
70 google-explicit-constructor,
71 google-global-names-in-headers,
72 google-upgrade-googletest-case,
73 hicpp-exception-baseclass,
74 hicpp-multiway-paths-covered,
75 hicpp-no-assembler,
76 hicpp-signed-bitwise,
77 llvm-namespace-comment,
78 misc-definitions-in-headers,
79 misc-misplaced-const,
80 misc-new-delete-overloads,
81 misc-non-copyable-objects,
82 misc-redundant-expression,
83 misc-static-assert,
84 misc-throw-by-value-catch-by-reference,
85 misc-unconventional-assign-operator,
86 misc-uniqueptr-reset-release,
87 misc-unused-alias-decls,
88 misc-unused-parameters,
89 misc-unused-using-decls,
90 modernize-concat-nested-namespaces,
91 modernize-deprecated-headers,
92 modernize-deprecated-ios-base-aliases,
93 modernize-loop-convert,
94 modernize-make-shared,
95 modernize-make-unique,
96 modernize-pass-by-value,

97 modernize-raw-string-literal,
98 modernize-redundant-void-arg,
99 modernize-replace-auto-ptr,
100 modernize-replace-disallow-copy-and-assign-macro,
101 modernize-replace-random-shuffle,
102 modernize-return-braced-init-list,
103 modernize-shrink-to-fit,
104 modernize-unary-static-assert,
105 modernize-use-auto,
106 modernize-use-bool-literals,
107 modernize-use-default-member-init,
108 modernize-use-emplace,
109 modernize-use-equals-default,
110 modernize-use-equals-delete,
111 modernize-use-nodiscard,
112 modernize-use-noexcept,
113 modernize-use-nullptr,
114 modernize-use-override,
115 modernize-use-transparent-functors,
116 modernize-use-uncaught-exceptions,
117 modernize-use-using,
118 openmp-use-default-none,
119 performance-faster-string-find,
120 performance-for-range-copy,
121 performance-implicit-conversion-in-loop,
122 performance-inefficient-algorithm,
123 performance-inefficient-string-concatenation,
124 performance-inefficient-vector-operation,
125 performance-move-const-arg,
126 performance-move-constructor-init,
127 performance-no-automatic-move,
128 performance-no-int-to-ptr,
129 performance-noexcept-move-constructor,
130 performance-trivially-destructible,
131 performance-type-promotion-in-math-fn,
132 performance-unnecessary-copy-initialization,
133 performance-unnecessary-value-param,
134 portability-simd-intrinsics,
135 readability-const-return-type,
136 readability-container-size-empty,
137 readability-convert-member-functions-to-static,
138 readability-delete-null-pointer,
139 readability-else-after-return,
140 readability-function-cognitive-complexity,
141 readability-identifier-naming,
142 readability-inconsistent-declaration-parameter-name,
143 readability-isolate-declaration,
144 readability-make-member-function-const,
145 readability-misleading-indentation,
146 readability-misplaced-array-index,
147 readability-non-const-parameter,
148 readability-redundant-access-specifiers,
149 readability-redundant-control-flow,
150 readability-redundant-declaration,
151 readability-redundant-function-ptr-dereference,
152 readability-redundant-member-init,

```

153     readability-redundant-smartptr-get,
154     readability-redundant-string-cstr,
155     readability-redundant-string-init,
156     readability-simplify-boolean-expr,
157     readability-simplify-subscript-expr,
158     readability-static-accessed-through-instance,
159     readability-static-definition-in-anonymous-namespace,
160     readability-string-compare,
161     readability-uniqueptr-delete-release"
162
163     WarningsAsErrors: "
164         boost-use-to-string,
165         bugprone-dangling-handle,
166         bugprone-fold-init-type,
167         bugprone-inaccurate-erase,
168         bugprone-incorrect-roundings,
169         bugprone-misplaced-widening-cast,
170         bugprone-sizeof-container,
171         bugprone-sizeof-expression,
172         bugprone-string-constructor,
173         bugprone-suspicious-enum-usage,
174         bugprone-suspicious-memset-usage,
175         bugprone-suspicious-missing-comma,
176         bugprone-suspicious-semicolon,
177         bugprone-swapped-arguments,
178         bugprone-unused-raii,
179         bugprone-use-after-move,
180         llvm-namespace-comment,
181         misc-non-copyable-objects,
182         misc-redundant-expression,
183         misc-throw-by-value-catch-by-reference,
184         misc-unused-alias-decls,
185         misc-unused-parameters,
186         misc-unused-using-decls,
187         modernize-deprecated-headers,
188         modernize-redundant-void-arg,
189         modernize-use-bool-literals,
190         modernize-use-emplace,
191         modernize-use-equals-default,
192         modernize-use-equals-delete,
193         modernize-use-nullptr,
194         modernize-use-override,
195         modernize-use-using,
196         performance-faster-string-find,
197         performance-inefficient-algorithm,
198         readability-make-member-function-const,
199         readability-misleading-indentation,
200         readability-misplaced-array-index,
201         readability-string-compare"
202
203     HeaderFilterRegex: ^(!\\usr)(?!\\opt)

```



497c9c4 ▾

autoware / .clang-tidy

↑ Top

Code

Blame

Raw



```
209
210 CheckOptions:
211   - key: bugprone-argument-comment.CommentBoolLiterals
212     value: "0"
213   - key: bugprone-argument-comment.CommentCharacterLiterals
214     value: "0"
215   - key: bugprone-argument-comment.CommentFloatLiterals
216     value: "0"
217   - key: bugprone-argument-comment.CommentIntegerLiterals
218     value: "0"
219   - key: bugprone-argument-comment.CommentNullPtrs
220     value: "0"
221   - key: bugprone-argument-comment.CommentStringLiterals
222     value: "0"
223   - key: bugprone-argument-comment.CommentUserDefinedLiterals
224     value: "0"
225   - key: bugprone-argument-comment.IgnoreSingleArgument
226     value: "0"
227   - key: bugprone-argument-comment.StrictMode
228     value: "0"
229   - key: bugprone-assert-side-effect.AssertMacros
230     value: assert
231   - key: bugprone-assert-side-effect.CheckFunctionCalls
```



```
389     - key: modernize-use-override.FinalSpelling
390       value: final
391     - key: modernize-use-override.IgnoreDestructors
392       value: "0"
393     - key: modernize-use-override.OverrideSpelling
394       value: override
395     - key: modernize-use-transparent-functors.SafeMode
396       value: "0"
397     - key: modernize-use-using.IgnoreMacros
398       value: "1"
399     - key: performance-faster-string-find.StringLikeClasses
400       value: std::basic_string
401     - key: performance-for-range-copy.AllowedTypes
402       value: ""
403     - key: performance-for-range-copy.WarnOnAllAutoCopies
404       value: "0"
405     - key: performance-inefficient-string-concatenation.StrictMode
406       value: "0"
407     - key: performance-inefficient-vector-operation.EnableProto
408       value: "0"
409     - key: performance-inefficient-vector-operation.VectorLikeClasses
410       value: ::std::vector
411     - key: performance-move-const-arg.CheckTriviallyCopyableMove
412       value: "1"
413     - key: performance-move-constructor-init.IncludeStyle
414       value: google
415     - key: performance-no-automatic-move.AllowedTypes
416       value: ""
417     - key: performance-type-promotion-in-math-fn.IncludeStyle
418       value: google
419     - key: performance-unnecessary-copy-initialization.AllowedTypes
420       value: ""
421     - key: performance-unnecessary-value-param.AllowedTypes
422       value: .*Ptr;.*SharedFuture
423     - key: performance-unnecessary-value-param.IncludeStyle
424       value: google
425     - key: portability-simd-intrinsics.Std
426       value: ""
427     - key: portability-simd-intrinsics.Suggest
428       value: "0"
429     - key: readability-function-cognitive-complexity.IgnoreMacros
430       value: "1"
431     - key: readability-else-after-return.WarnOnUnfixable
432       value: "1"
433     - key: readability-identifier-naming.NamespaceCase
```



```
434     value: lower_case
435 - key: readability-identifier-naming.ClassCase
436   value: CamelCase
437 - key: readability-identifier-naming.PrivateMemberSuffix
438   value: _
439 - key: readability-identifier-naming.StructCase
440   value: CamelCase
441 - key: readability-identifier-naming.FunctionCase
442   value: lower_case
443 - key: readability-identifier-naming.VariableCase
444   value: lower_case
445 - key: readability-identifier-naming.GlobalConstantCase
446   value: lower_case
447 - key: readability-identifier-naming.GlobalConstantPrefix
448   value: g_
449 - key: readability-identifier-naming.ConstexprVariableCase
450   value: lower_case
451 - key: readability-inconsistent-declaration-parameter-name.IgnoreMacros
452   value: "1"
453 - key: readability-inconsistent-declaration-parameter-name.Strict
454   value: "0"
455 - key: readability-redundant-smartptr-get.IgnoreMacros
456   value: "1"
457 - key: readability-redundant-string-init.StringNames
458   value: ::std::basic_string
459 - key: readability-simplify-subscript-expr.Types
460   value: ::std::basic_string;::std::basic_string_view;::std::vector;::std::array
461 - key: readability-static-accessed-through-instance.NameSpecifierNestingThreshold
462   value: "3"
```