Subject: TheIDE and clang-tidy
Posted by coolman on Sat, 25 Feb 2023 14:33:54 GMT
View Forum Message <> Reply to Message

Hi,

I played around with clang-tidy and tried running it on TheIDE. I'm attaching a configuration file with the checks commented out so that the compilation can be run without warnings. You can find checks description at Clang-Tidy Checks documentation page.

What do you think, are some of them worth looking at and fixing?

BR, Radek

```
Checks: '-*,
 bugprone-*,
 -bugprone-branch-clone,
 -bugprone-copy-constructor-init,
 -bugprone-easily-swappable-parameters,
 -bugprone-implicit-widening-of-multiplication-result,
 -bugprone-incorrect-roundings,
 -bugprone-integer-division,
 -bugprone-macro-parentheses,
 -bugprone-misplaced-widening-cast,
 -bugprone-move-forwarding-reference,
 -bugprone-narrowing-conversions,
 -bugprone-parent-virtual-call,
 -bugprone-reserved-identifier,
 -bugprone-signed-char-misuse,
 -bugprone-sizeof-expression,
 -bugprone-suspicious-include,
 -bugprone-suspicious-string-compare,
 -bugprone-undefined-memory-manipulation,
 -bugprone-unhandled-self-assignment,
 -bugprone-virtual-near-miss,
 cert-*,
 -cert-dcl37-c,
 -cert-dcl21-cpp,
 -cert-dcl50-cpp,
 -cert-dcl51-cpp,
 -cert-env33-c,
 -cert-err34-c,
 -cert-err09-cpp,
 -cert-err52-cpp,
 -cert-err58-cpp,
 -cert-err61-cpp,
```

-cert-msc30-c,
-cert-msc50-cpp,
-cert-oop54-cpp,
-cert-str34-c,
clang-analyzer-*,
-clang-analyzer-cplusplus.NewDeleteLeaks,
-clang-analyzer-core.CallAndMessage,
-clang-analyzer-core.DivideZero,
-clang-analyzer-core.NonNullParamChecker,
-clang-analyzer-core.NullDereference,
-clang-analyzer-core.uninitialized.Assign,
-clang-analyzer-core.uninitialized.Branch,
-clang-analyzer-core.uninitialized.UndefReturn,
-clang-analyzer-core.UndefinedBinaryOperatorResult,
-clang-analyzer-deadcode.DeadStores,
-clang-analyzer-optin.cplusplus.UninitializedObject,
-clang-analyzer-optin.cplusplus.VirtualCall,
-clang-analyzer-optin.performance.Padding,
-clang-analyzer-cplusplus.NewDelete,
-clang-analyzer-cplusplus.PlacementNew,
-clang-analyzer-security.insecureAPI.DeprecatedOrUnsafeBufferHandling,
-clang-analyzer-security.insecureAPI.strcpy,
clang-diagnostic-*,
concurrency-*,
-concurrency-mt-unsafe,
cppcoreguidelines-*,
-cppcoreguidelines-avoid-c-arrays,
-cppcoreguidelines-avoid-goto,
-cppcoreguidelines-avoid-magic-numbers,
-cppcoreguidelines-avoid-non-const-global-variables,
-cppcoreguidelines-c-copy-assignment-signature,
-cppcoreguidelines-explicit-virtual-functions,
-cppcoreguidelines-init-variables,
-cppcoreguidelines-interfaces-global-init,
-cppcoreguidelines-macro-usage,
-cppcoreguidelines-narrowing-conversions,
-cppcoreguidelines-no-malloc,
-cppcoreguidelines-non-private-member-variables-in-classes,
-cppcoreguidelines-owning-memory,
-cppcoreguidelines-prefer-member-initializer,
-cppcoreguidelines-pro-bounds-array-to-pointer-decay,
-cppcoreguidelines-pro-bounds-constant-array-index,
-cppcoreguidelines-pro-bounds-pointer-arithmetic,
-cppcoreguidelines-pro-type-const-cast,
-cppcoreguidelines-pro-type-cstyle-cast,
-cppcoreguidelines-pro-type-member-init,
-cppcoreguidelines-pro-type-reinterpret-cast,
-cppcoreguidelines-pro-type-static-cast-downcast,

```
 -cppcoreguidelines-pro-type-union-access,
 -cppcoreguidelines-pro-type-vararg,
 -cppcoreguidelines-slicing,
 -cppcoreguidelines-special-member-functions,
 misc-*,
 -misc-definitions-in-headers,
 -misc-no-recursion,
 -misc-non-private-member-variables-in-classes,
 -misc-redundant-expression,
 -misc-throw-by-value-catch-by-reference,
 -misc-unconventional-assign-operator,
 -misc-unused-parameters,
 modernize-*,
 -modernize-avoid-c-arrays,
 -modernize-deprecated-headers,
 -modernize-loop-convert,
 -modernize-pass-by-value,
 -modernize-raw-string-literal,
 -modernize-redundant-void-arg,
 -modernize-return-braced-init-list,
 -modernize-use-auto,
 -modernize-use-bool-literals,
 -modernize-use-default-member-init,
 -modernize-use-equals-default,
 -modernize-use-equals-delete,
 -modernize-use-nullptr,
 -modernize-use-override,
 -modernize-use-transparent-functors,
 -modernize-use-using,
 -modernize-use-trailing-return-type,
 performance-*,
 -performance-for-range-copy,
 -performance-no-int-to-ptr,
 -performance-noexcept-move-constructor,
 -performance-unnecessary-copy-initialization,
 -performance-unnecessary-value-param,
 portability-*,
 -portability-simd-intrinsics,
 -readability-*,
 -readability-convert-member-functions-to-static,
 -readability-uppercase-literal-suffix,
 -readability-braces-around-statements'
WarningsAsErrors: ''
HeaderFilterRegex: '.*\.(h|hpp)$'
AnalyzeTemporaryDtors: false
FormatStyle:     none
```