

# C++ static code analysis: Raw string literals should be used

2 minutes

Since C++11, raw string literals can be used to avoid the need to escape characters in a string.

This rule raises an issue when using a raw string literal would make a string easier to read. For instance, when a non-raw string contains different escaped sequences (among `\'`, `\\`, `\"` and `\?`) or more than two of the same kind.

## Noncompliant Code Example

```
const auto* result = "a?b \\"; // Noncompliant
const auto* regEx = "\\(\\|\\.\\)"; // Noncompliant
const auto* message = "Use \"x\" or \"y\""; // Noncompliant
```

## Compliant Solution

```
const auto* result = R"(a?b ")";
const auto* regEx = R"(\\(|\\.\\))";
const auto* message = R"(Use "x" or "y")";

const auto* twoLines = "one\r\ntwo"; // Compliant, contains \r
const auto* path = "C:\\Program Files\\Microsoft Office\\Office16\\";
// Compliant, raw strings would not improve readability
```

## Exceptions

To preserve readability, this rule ignores strings containing only one character and strings with escaped whitespace or non-printable characters:

- Non-printable characters: `\a \b \f \v \nnn \xnn \unnnn \Unnnnnnnn`
- Tab: `\t`
- Carriage return: `\r`