rules.sonarsource.com

C++ static code analysis: "static_assert" with no message should be used over "static_assert" with empty or redundant message

1 minute

C++11 version of the standard introduced `static_assert(expr, message)` to check that the compile-time constant expression `expr` is true.

C++17 version of the standard has made the second argument `message` optional. This rule flags occurrences of `std::static_assert` where the second argument message is empty or a substring of `expr`.

Noncompliant Code Example

```
template <class T>
T f(T i) {
  static_assert(std::is_integral<T>::value, ""); // Noncompliant, remove the empty string second argument.
  // or
  static_assert(std::is_integral<T>::value, "std::is_integral"); // Noncompliant, remove the redundant second argument.
  // ...
}
```

Compliant Solution

```
template <class T>
T f(T i) {
  static_assert(std::is_integral<T>::value); // Compliant
  static_assert(std::is_integral<T>::value, "Integral required"); // Compliant
  // ...
}
```