rules.sonarsource.com

# C++ static code analysis: "else" statements should be clearly matched with an "if"

1-2 minutes

---

The dangling `else` problem appears when nested `if`/`else` statements are written without curly braces. In this case, `else` is associated with the nearest `if` but that is not always obvious and sometimes the indentation can also be misleading.

This rules reports `else` statements that are difficult to understand, because they are inside nested `if` statements without curly braces.

Adding curly braces can generally make the code clearer (see rule {rule:cpp:S121} ), and in this situation of dangling `else`, it really clarifies the intention of the code.

## Noncompliant Code Example

```
if (a)
  if (b)
    d++;
else     // Noncompliant, is the "else" associated with "if(a)" or "if (b)"? (the answer is "if(b)")
  e++;
```

## Compliant Solution

```
if (a) {
  if (b) {
    d++;
  }
} else { // Compliant, there is no doubt the "else" is associated with "if(a)"
  e++;
}
```

## See

- https://en.wikipedia.org/wiki/Dangling_else