

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#**
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML



C# static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C# code

All rules 409

Vulnerability 34

Bug 76

Security Hotspot 28

Code Smell 271

Quick Fix 52

Tags ▾

Search by name... 🔍

Bug
"NaN" should not be used in comparisons Bug
Conditionally executed code should be reachable Bug
Null pointers should not be dereferenced Bug
For-loop conditions should be true at least once Bug
A "for" loop update clause should move the counter in the right direction Bug
"ToString()" method should not return null Bug
Return values from functions without side effects should not be ignored Bug
Values should not be uselessly incremented Bug
Collections should not be passed as arguments to their own methods Bug
Related "if/else if" statements should not have the same condition Bug
Objects should not be created to be dropped immediately without being used

A conditionally executed single line should be denoted by indentation

Analyze your code

Code Smell Critical ? confusing suspicious

In the absence of enclosing curly braces, the line immediately after a conditional is the one that is conditionally executed. By both convention and good practice, such lines are indented. In the absence of both curly braces and indentation the intent of the original programmer is entirely unclear and perhaps not actually what is executed. Additionally, such code is highly likely to be confusing to maintainers.

Noncompliant Code Example

```
if (condition) // Noncompliant
DoTheThing();

DoTheOtherThing();
SomethingElseEntirely();

Foo();
```





Compliant Solution

```
if (condition)
    DoTheThing();

DoTheOtherThing();
SomethingElseEntirely();

Foo();
```

Available In:
sonarlint | sonarcloud | sonarqube

 Bug
Identical expressions should not be used on both sides of a binary operator  Bug
Loops with at most one iteration should be refactored  Bug
Variables should not be self-assigned  Bug
Constructing arguments of system commands from user input is