

- 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...



"protected" members

Code Smell

Underscores should be used to make large numbers readable

Code Smell

"ToString()" calls should not be redundant

Code Smell

"==" should not be used when "Equals" is overridden

Code Smell

An abstract class should have both abstract and concrete methods

Code Smell

Multiple variables should not be declared on the same line

Code Smell

Culture should be specified for "string" operations

Code Smell

"switch" statements should have at least 3 "case" clauses

Code Smell

break statements should not be used except for switch cases

Code Smell

String literals should not be duplicated

Code Smell

Files should contain an empty newline at the end

Code Smell

Unused "using" should be removed

Code Smell

"nameof" should be used

Analyze your code

Code Smell Critical bad-practice

Because parameter names could be changed during refactoring, they should not be spelled out literally in strings. Instead, use nameof (), and the string that's output will always be correct.

This rule raises an issue when a string in the throw statement contains the name of one of the method parameters.

Noncompliant Code Example

```
void DoSomething(int someParameter, string anotherParam)
{
    if (someParameter < 0)
    {
        throw new ArgumentException("Bad argument", "somePar
    }
    if (anotherParam == null)
    {
        throw new Exception("anotherParam should not be null
    }
}
```

Compliant Solution

```
void DoSomething(int someParameter)
{
    if (someParameter < 0)
    {
        throw new ArgumentException("Bad argument", nameof(s
    }
    if (anotherParam == null)
    {
        throw new Exception($"{nameof(anotherParam)} should
    }
}
```

Exceptions

- The rule doesn't raise any issue when using C# < 6.0.
- When the parameter name is contained in a sentence inside the throw statement string, the rule will raise an issue only if the parameter name is at least 5 characters long. This is to avoid false positives.

Available In:

sonarlint | sonarcloud | sonarqube

A close curly brace should be located at the beginning of a line

 Code Smell

Tabulation characters should not be used

 Code Smell

Methods and properties should be named in PascalCase

 Code Smell

Track uses of in-source issue suppressions

 Code Smell