

- 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

Enumeration types should comply with a naming convention

Analyze your code

Code Smell Minor convention

Shared naming conventions allow teams to collaborate efficiently. This rule checks that all enum names match a provided regular expression.

The default configuration is the one recommended by Microsoft:

- Pascal casing, starting with an upper case character, e.g. BackColor
- Short abbreviations of 2 letters can be capitalized, e.g. GetID
- Longer abbreviations need to be lower case, e.g. GetHtml
- If the enum is marked as [Flags] then its name should be plural (e.g. MyOptions), otherwise, names should be singular (e.g. MyOption)

Noncompliant Code Example

With the default regular expression for non-flags enums: `^[A-Z]{1,3}[a-z0-9]{1,3}*[A-Z]{2}?[a-z0-9]{1,3}$`

```
public enum foo // Noncompliant
{
    FooValue = 0
}
```

With the default regular expression for flags enums: `^[A-Z]{1,3}[a-z0-9]{1,3}*[A-Z]{2}?[a-z0-9]{1,3}$`

```
[Flags]
public enum Option // Noncompliant
{
    None = 0,
    Option1 = 1,
    Option2 = 2
}
```

Compliant Solution

```
public enum Foo
{
    FooValue = 0
}
```

```
[Flags]
public enum Options
{
    None = 0,
    Option1 = 1,
    Option2 = 2
}
```

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

Available In:

sonarlint  | **sonarcloud**  | **sonarqube** 

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
[Privacy Policy](#)