

- 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

Types should be named in PascalCase

Analyze your code

Code Smell Minor ? convention

Shared naming conventions allow teams to collaborate efficiently. This rule checks whether or not type names are using PascalCase. To reduce noise, two consecutive upper case characters are allowed unless they form the whole type name. So, MyXClass is compliant, but XC on its own is not.

Noncompliant Code Example

```
class my_class {...}
class SOMENAME42 {...}
```

Compliant Solution

```
class MyClass {...}
class SomeName42 {...}
```

Exceptions

- The rule ignores types that are marked with `ComImportAttribute` or `InterfaceTypeAttribute`.
- The rule allows for two-letter acronyms in which both letters are capitalized, as shown in the following identifier: `IOStream`
- The rule allows having `'_'` characters in class names inside test projects.

```
class Some_Name__42 {...} // valid in test
class Some_name__42 {...} // still not valid
class Some_Name_XC {...} // invalid because of XC, should be
```

See

- [Microsoft Capitalization Conventions](#)

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