

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C# 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... 🔍

Code Smell
Properties should be preferred
Code Smell
Generics should be used when appropriate
Code Smell
Type names should not match namespaces
Code Smell
Strings should be normalized to uppercase
Code Smell
Exceptions should provide standard constructors
Code Smell
Assemblies should be marked with "NeutralResourcesLanguageAttribute"
Code Smell
Interfaces should not be empty
Code Smell
Enumerations should have "Int32" storage
Code Smell
Generic methods should provide type parameters
Code Smell
Multidimensional arrays should not be used
Code Smell
"static readonly" constants should be "const" instead
Code Smell
Strings or integral types should be

Nested blocks of code should not be left empty

Analyze your code

Code Smell

Major ?

suspicious

Most of the time a block of code is empty when a piece of code is really missing. So such empty block must be either filled or removed.

Noncompliant Code Example





```
for (int i = 0; i < 42; i++){ } // Empty on purpose or missi
```

Exceptions

When a block contains a comment, this block is not considered to be empty.

Available In:

sonarlint | sonarcloud | sonarqube

used for indexers  Code Smell
Parameter names should not duplicate the names of their methods  Code Smell
Track use of "NotImplementedException"  Code Smell
Empty "default" clauses should be removed  Code Smell
Redundant property names should be