

- 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

Fields should be private

Analyze your code

Code Smell Major pitfall

Fields should not be part of an API, and therefore should always be private. Indeed, they cannot be added to an interface for instance, and validation cannot be added later on without breaking backward compatibility. Instead, developers should encapsulate their fields into properties. Explicit property getters and setters can be introduced for validation purposes or to smooth the transition to a newer system.

Noncompliant Code Example

```
public class Foo
{
    public int MagicNumber = 42;
}
```

Compliant Solution

```
public class Foo
{
    public int MagicNumber
    {
        get { return 42; }
    }
}
```

or

```
public class Foo
{
    private int MagicNumber = 42;
}
```





Exceptions

structs are ignored, as are static and const fields in classes.

Further, an issue is only raised when the real accessibility is public, taking into account the class accessibility.

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