

- 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... 🔍

Code Smell

Fields that are only assigned in the constructor should be "readonly"

Code Smell

Static fields should not be used in generic types

Code Smell

Multiline blocks should be enclosed in curly braces

Code Smell

Boolean expressions should not be gratuitous

Code Smell

Types and methods should not have too many generic parameters

Code Smell

Write-only properties should not be used

Code Smell

Exceptions should not be thrown from property getters

Code Smell

Unused type parameters should be removed

Code Smell

Parameters should be passed in the correct order

Code Smell

Two branches in a conditional structure should not have exactly the same implementation

Code Smell

Unused assignments should be removed

Code Smell

"PartCreationPolicyAttribute" should be used with "ExportAttribute"

Analyze your code

Bug Major mef pitfall

The `PartCreationPolicyAttribute` attribute, which is part of the Managed Extensibility Framework (MEF), is used to specify how the exported object will be created. Therefore it doesn't make sense not to export this a class with this attribute using the `ExportAttribute` attribute.

This rule raises an issue when a class is marked as shared with a `PartCreationPolicyAttribute` but lacks a `ExportAttribute`.

### Noncompliant Code Example

```
[PartCreationPolicy(CreationPolicy.Any)] // Noncompliant
public class FooBar : IFooBar
{
}
```

### Compliant Solution

```
[Export(typeof(IFooBar))]
[PartCreationPolicy(CreationPolicy.Any)]
public class FooBar : IFooBar
{
}
```

Available In:

sonarlint | sonarcloud | sonarqube

**Tests should not be ignored**

 Code Smell

**"switch" statements should not have too many "case" clauses**

 Code Smell

**Sections of code should not be commented out**

 Code Smell

**Unused method parameters should be removed**

 Code Smell