

- 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

Property names should not match get methods

Analyze your code

Code Smell Major ? confusing

Properties and Get method should have names that makes them clearly distinguishable.

This rule raises an issue when the name of a public or protected member starts with 'Get' and otherwise matches the name of a public or protected property.

Noncompliant Code Example

```
using System;

namespace MyLibrary
{
    public class Foo
    {
        public DateTime Date
        {
            get { return DateTime.Today; }
        }

        public string GetDate() // Noncompliant
        {
            return this.Date.ToString();
        }
    }
}
```





Compliant Solution

```
using System;

namespace MyLibrary
{
    public class Foo
    {
        public DateTime Date
        {
            get { return DateTime.Today; }
        }

        public string GetDateAsString()
        {
            return this.Date.ToString();
        }
    }
}
```

Available In:

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