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 not have public accessibility

Analyze your code

Code Smell Minor cwe

Public fields in public classes do not respect the encapsulation principle and has three main disadvantages:

- Additional behavior such as validation cannot be added.
- The internal representation is exposed, and cannot be changed afterwards.
- Member values are subject to change from anywhere in the code and may not meet the programmer’s assumptions.

By using private fields and public properties (set and get), unauthorized modifications are prevented. Properties also benefit from additional protection (security) features such as Link Demands.

Note that due to optimizations on simple properties, public fields provide only very little performance gain.

Noncompliant Code Example

```
public class Foo
{
    public int instanceData = 32; // Noncompliant
}
```

Compliant Solution

```
public class Foo
{
    private int instanceData = 32;

    public int InstanceData
    {
        get { return instanceData; }
        set { instanceData = value ; }
    }
}
```

Exceptions

Fields marked as `readonly` or `const` are ignored by this rule.

Fields inside classes or structs annotated with the `StructLayoutAttribute` are ignored by this rule.

See

- MITRE, CWE-493 - Critical Public Variable Without Final Modifier

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