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

Implementations should be provided for "partial" methods

Analyze your code

Code Smell | Minor | suspicious

`partial` methods allow an increased degree of flexibility in programming a system. Hooks can be added to generated code by invoking methods that define their signature, but might not have an implementation yet. But if the implementation is still missing when the code makes it to production, the compiler silently removes the call. In the best case scenario, such calls simply represent cruft, but in they worst case they are critical, missing functionality, the loss of which will lead to unexpected results at runtime.

This rule raises an issue for partial methods for which no implementation can be found in the assembly.

Noncompliant Code Example

```
partial class C
{
  partial void M(); //Noncompliant

  void OtherM()
  {
    M(); //Noncompliant. Will be removed.
  }
}
```

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