

- 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

Control flow statements "if", "switch", "for", "foreach", "while", "do" and "try" should not be nested too deeply

Code Smell

"switch/Select" statements should contain a "default/Case Else" clauses

Code Smell

"if ... else if" constructs should end with "else" clauses

Code Smell

Control structures should use curly braces

Code Smell

Expressions should not be too complex

Code Smell

ASP.NET HTTP request validation feature should not be disabled

Vulnerability

Serialization constructors should be secured

Vulnerability

Calculations should not overflow

Bug

Floating point numbers should not be tested for equality

Bug

Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression

Code Smell

Use a testable date/time provider.

Code Smell

Assertion arguments should be passed in the correct order

Analyze your code

Code Smell Major ? tests suspicious

The standard assertions library methods such as `AreEqual` and `AreSame` in **MSTest** and **NUnit**, or `Equal` and `Same` in **XUnit**, expect the first argument to be the expected value and the second argument to be the actual value. Swap them, and your test will still have the same outcome (succeed/fail when it should) but the error messages will be confusing.

This rule raises an issue when the second argument to an assertions library method is a hard-coded value and the first argument is not.

Noncompliant Code Example

```
Assert.AreEqual(runner.ExitCode, 0, "Unexpected exit code");
```

Compliant Solution

```
Assert.AreEqual(0, runner.ExitCode, "Unexpected exit code");
```

Available In:

sonarlint | sonarcloud | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
[Privacy Policy](#)

Property names should not match get methods

 Code Smell

Locales should be set for data types

 Code Smell

Literals should not be passed as localized parameters

 Code Smell

Operators should be overloaded consistently

 Code Smell