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

The simplest possible condition syntax should be used

Code Smell

Redundant parentheses should not be used

Code Smell

"GC.SuppressFinalize" should not be invoked for types without destructors

Code Smell

Members should not be initialized to default values

Code Smell

Sequential tests should not check the same condition

Code Smell

Redundant modifiers should not be used

Code Smell

Methods and properties that don't access instance data should be static

Code Smell

"Exception" should not be caught when not required by called methods

Code Smell

"sealed" classes should not have "protected" members

Code Smell

Underscores should be used to make large numbers readable

Code Smell

"ToString()" calls should not be redundant

Code Smell

"GetHashCode" should not reference mutable fields

Analyze your code

Bug Minor Quick Fix

`GetHashCode` is used to file an object in a `Dictionary` or `Hashtable`. If `GetHashCode` uses non-`readonly` fields and those fields change after the object is stored, the object immediately becomes mis-filed in the `Hashtable`. Any subsequent test to see if the object is in the `Hashtable` will return a false negative.

Noncompliant Code Example

```
public class Person
{
  public int age;
  public string name;

  public override int GetHashCode()
  {
    int hash = 12;
    hash += this.age.GetHashCode(); // Noncompliant
    hash += this.name.GetHashCode(); // Noncompliant
    return hash;
  }
```

Compliant Solution

```
public class Person
{
  public readonly DateTime birthday;
  public string name;

  public override int GetHashCode()
  {
    int hash = 12;
    hash += this.birthday.GetHashCode();
    return hash;
  }
```

Available In:

sonarlint sonarcloud sonarqube

"==" 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