

- 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... 🔍

System arguments should be used instead of strings

Code Smell

Collection properties should be readonly

Code Smell

Disposable types should declare finalizers

Code Smell

String URI overloads should call "System.Uri" overloads

Code Smell

URI properties should not be strings

Code Smell

URI return values should not be strings

Code Smell

URI Parameters should not be strings

Code Smell

Custom attributes should be marked with "System.AttributeUsageAttribute"

Code Smell

Assemblies should explicitly specify COM visibility

Code Smell

Assemblies should be marked as CLS compliant

Code Smell

"Generic.List" instances should not be part of public APIs

Code Smell

Collections should implement the generic interface

Code Smell

### Static fields should not be updated in constructors

Analyze your code

Code Smell Major ?

Assigning a value to a static field in a constructor could cause unreliable behavior at runtime since it will change the value for all instances of the class. Instead remove the field's static modifier, or initialize it statically.

#### Noncompliant Code Example

```
public class Person
{
    private static DateTime dateOfBirth;
    private static int expectedFingers;

    public Person(DateTime birthday)
    {
        dateOfBirth = birthday; // Noncompliant; now everyone has the same birthday
        expectedFingers = 10; // Noncompliant
    }
}
```

#### Compliant Solution

```
public class Person
{
    private DateTime dateOfBirth;
    private static int expectedFingers = 10;

    public Person(DateTime birthday)
    {
        this.dateOfBirth = birthday;
    }
}
```

Available In: sonarlint | sonarcloud | sonarqube

**Generic event handlers should be used**

 Code Smell

**Event Handlers should have the correct signature**

 Code Smell

**"Assembly.GetExecutingAssembly" should not be called**

 Code Smell

**Arguments of public methods should be validated against null**

 Code Smell