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...

same implementation
Code Smell

Unused assignments should be removed
Code Smell

Tests should not be ignored
Code Smell

"switch" statements should not have too many "case" clauses
Code Smell

Sections of code should not be commented out
Code Smell

Unused method parameters should be removed
Code Smell

Empty arrays and collections should be returned instead of null
Code Smell

Unused private types or members should be removed
Code Smell

Track uses of "FIXME" tags
Code Smell

"Obsolete" attributes should include explanations
Code Smell

Assignments should not be made from within sub-expressions
Code Smell

General exceptions should never be thrown
Code Smell

Serialization event handlers should be implemented correctly

Analyze your code

Bug Major

Serialization event handlers that don't have the correct signature will simply not be called, thus bypassing any attempts to augment the automated de/serialization.

This rule raises an issue when a method marked with one of the following attributes is `public`, `static`, does not return `void`, has type parameters, or does not have a single parameter of type `System.Runtime.Serialization.StreamingContext`:

- `System.Runtime.Serialization.OnSerializingAttribute`
- `System.Runtime.Serialization.OnSerializedAttribute`
- `System.Runtime.Serialization.OnDeserializingAttribute`
- `System.Runtime.Serialization.OnDeserializedAttribute`

Noncompliant Code Example

```
[Serializable]
public class Foo
{
    [OnSerializing]
    public void OnSerializing(StreamingContext context) {} /

    [OnSerialized]
    int OnSerialized(StreamingContext context) {} // Noncomp

    [OnDeserializing]
    void OnDeserializing() {} // Noncompliant should have a

    [OnSerializing]
    public void OnSerializing2<T>(StreamingContext context)

    [OnDeserialized]
    void OnDeserialized(StreamingContext context, string str
}
```

Compliant Solution

```
[Serializable]
public class Foo
{
    [OnSerializing]
    private void OnSerializing(StreamingContext context) {}

    [OnSerialized]
    private void OnSerialized(StreamingContext context) {}

    [OnDeserializing]
    private void OnDeserializing(StreamingContext context) {

    [OnDeserialized]
```

Utility classes should not have public constructors

Code Smell

Local variables should not shadow class fields

Code Smell

Redundant pairs of parentheses should be removed

Code Smell

Inheritance tree of classes should not be too deep

Code Smell

```
    private void OnDeserialized(StreamingContext context) {}
}
```

Available In:

sonarlint | sonarcloud | sonarqube