

- 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 |

Literals should not be passed as localized parameters

Analyze your code

- Code Smell
- Major ?
- localisation pitfall

String literals embedded in the source code will not be localized properly.

This rule raises an issue when a literal string is passed as a parameter or property and one or more of the following cases is true:

- The LocalizableAttribute attribute of the parameter or property is set to true.
- The parameter or property name contains "Text", "Message", or "Caption".
- The name of the string parameter that is passed to a Console.WriteLine or Console.WriteLine method is either "value" or "format".

Noncompliant Code Example

```
using System;
using System.Globalization;
using System.Reflection;
using System.Windows.Forms;

[assembly: NeutralResourcesLanguageAttribute("en-US")]
namespace MyLibrary
{
    public class Foo
    {
        public void SetHour(int hour)
        {
            if (hour < 0 || hour > 23)
            {
                MessageBox.Show("The valid range is 0 - 23.")
            }
        }
    }
}
```

Compliant Solution

```
using System;
using System.Globalization;
using System.Reflection;
using System.Resources;
using System.Windows.Forms;

[assembly: NeutralResourcesLanguageAttribute("en-US")]
namespace MyLibrary
{
    public class Foo
    {
        ResourceManager rm;
        public Foo()
        {
            rm = new ResourceManager("MyLibrary.Resources", Assembly.GetExecutingAssembly());
        }
    }
}
```

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

```
{
    rm = new ResourceManager("en-US", Assembly.GetEx
}

public void SetHour(int hour)
{
    if (hour < 0 || hour > 23)
    {
        MessageBox.Show(
            rm.GetString("OutOfRangeMessage", CultureInf
        )
    }
}
}
```

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](#)