

- 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

### Locales should be set for data types

Analyze your code

Code Smell Major ? localisation

When you create a `DataTable` or `DataSet`, you should set the locale explicitly. By default, the locale for these types is the current culture. For data that is stored in a database or file and is shared globally, the locale should ordinarily be set to the invariant culture (`CultureInfo.InvariantCulture`).

This rule raises an issue when `System.Data.DataTable` or `System.Data.DataSet` instances are created without explicitly setting the locale property (`DataTable.Locale` or `DataSet.Locale`).

#### Noncompliant Code Example

```
using System;
using System.Data;

namespace MyLibrary
{
    public class Foo
    {
        public DataTable CreateTable()
        {
            DataTable table = new DataTable("Customers"); //
            DataColumn key = table.Columns.Add("ID", typeof(

            key.AllowDBNull = false;
            key.Unique = true;
            table.Columns.Add("LastName", typeof(String));
            table.Columns.Add("FirstName", typeof(String));
            return table;
        }
    }
}
```

#### Compliant Solution

```
using System;
using System.Data;
using System.Globalization;

namespace MyLibrary
{
    public class Foo
    {
        public DataTable CreateTable()
        {
            DataTable table = new DataTable("Customers");
            table.Locale = CultureInfo.InvariantCulture;
            DataColumn key = table.Columns.Add("ID", typeof(

            key.AllowDBNull = false;
```

**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

```
        key.Unique = true;
        table.Columns.Add("LastName", typeof(String));
        table.Columns.Add("FirstName", typeof(String));
        return table;
    }
}
```

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](#)