sonar RULES

Products

- 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

Overriding members should do more than simply call the same member in the base class

Analyze your code

Code Smell | Minor | Quick Fix | redundant clumsy

Overriding a method just to call the same method from the base class without performing any other actions is useless and misleading. The only time this is justified is in `sealed` overriding methods, where the effect is to lock in the parent class behavior. This rule ignores overrides of `Equals` and `GetHashCode`.

NOTE: In some cases it might be dangerous to add or remove empty overrides, as they might be breaking changes.

Noncompliant Code Example

```
public override void Method() // Noncompliant
{
  base.Method();
}
```

Compliant Solution

```
public override void Method()
{
  //do something else
}
```

Exceptions

If there is an attribute in any level of the overriding chain, then the overridden member is ignored.

```
public class Base
{
  [Required]
  public virtual string Name { get; set; }
}

public class Derived : Base
{
  public override string Name
  {
    get
    {
      return base.Name;
    }
    set
    {
      base.Name = value;
    }
  }
}
```

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

If there is a documentation comment on the overriding method, it will be ignored:

```
public class Foo : Bar
{
    /// <summary>
    /// Keep this method for backwards compatibility.
    /// </summary>
    public override void DoSomething()
    {
        base.DoSomething();
    }
}
```

Available In:

sonarlint sonarcloud sonarqube