

- 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... 🔍

Parameters should be passed in the correct order

Code Smell

Two branches in a conditional structure should not have exactly the 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

Collection sizes and array length comparisons should make sense

Analyze your code

Bug Major ?

The size of a collection and the length of an array are always greater than or equal to zero. So testing that a size or length is greater than or equal to zero doesn't make sense, since the result is always `true`. Similarly testing that it is less than zero will always return `false`. Perhaps the intent was to check the non-emptiness of the collection or array instead.

Noncompliant Code Example

```
if(collection.Count >= 0){...}

if(enumerable.Count() < 0){...}

if(array.Length >= 0){...}

bool result = array.Length >=0;
```





Compliant Solution

```
if (list.Any()) { ... }

if (list.Count > 0) { ... }

if (array.Length >= 42) { ... }
```

Available In:
sonarlint | **sonarcloud** | **sonarqube**

 Code Smell
General exceptions should never be thrown  Code Smell
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