

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C# 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... 🔍

Code Smell
Assemblies should explicitly specify COM visibility
Code Smell
Assemblies should be marked as CLS compliant
Code Smell
"Generic.List" instances should not be part of public APIs
Code Smell
Collections should implement the generic interface
Code Smell
Generic event handlers should be used
Code Smell
Event Handlers should have the correct signature
Code Smell
"Assembly.GetExecutingAssembly" should not be called
Code Smell
Arguments of public methods should be validated against null
Code Smell
Value types should implement "IEquatable<T>"
Code Smell
Finalizers should not be empty
Code Smell
"[ExpectedException]" should not be used
Code Smell
"[ExpectedException]" should not be used

Multiline blocks should be enclosed in curly braces

Analyze your code

Code Smell Major ? cwe

Curly braces can be omitted from a one-line block, such as with an `if` statement or for loop, but doing so can be misleading and induce bugs.

This rule raises an issue when the whitespacing of the lines after a one line block indicates an intent to include those lines in the block, but the omission of curly braces means the lines will be unconditionally executed once.

Note that this rule considers tab characters to be equivalent to 1 space. If you mix spaces and tabs you will sometimes see issues in code which look fine in your editor but are confusing when you change the size of tabs.

Noncompliant Code Example

```
if (condition)
    FirstActionInBlock();
    SecondAction(); // Noncompliant; executed unconditionally
    ThirdAction();

if(condition) FirstActionInBlock(); SecondAction(); // Nonc

if(condition) FirstActionInBlock(); // Noncompliant
    SecondAction(); // Executed unconditionally

string str = null;
for (int i = 0; i < array.Length; i++)
    str = array[i];
    DoTheThing(str); // Noncompliant; executed only on last a
```

Compliant Solution

```
if (condition)
{
    FirstActionInBlock();
    SecondAction();
}
ThirdAction();


string str = null;
for (int i = 0; i < array.Length; i++)
{
    str = array[i];
    DoTheThing(str);
}
```

See

- MITRE, CWE-483 - Incorrect Block Delimitation

Available In:


this should not be exposed from constructors

 Code Smell

Types should not have members with visibility set higher than the type's visibility

 Code Smell

Fields should be private

 Code Smell

"try" statements with identical "catch" and/or "finally" blocks should be merged

 Code Smell