

- 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... 🔍

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
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
Code Smell
Inheritance tree of classes should not be too deep
Code Smell
Nested blocks of code should not be left empty
Code Smell
Methods should not have too many parameters

All branches in a conditional structure should not have exactly the same implementation

Analyze your code

Bug Major ?

Having all branches in a `switch` or `if` chain with the same implementation is an error. Either a copy-paste error was made and something different should be executed, or there shouldn't be a `switch/if` chain at all.

Noncompliant Code Example

```
if (b == 0) // Noncompliant
{
    DoTheThing();
}
else
{
    DoTheThing();
}

int b = a > 12 ? 4 : 4; // Noncompliant





switch (i) // Noncompliant
{
    case 1:
        DoSomething();
        break;
    case 2:
        DoSomething();
        break;
    case 3:
        DoSomething();
        break;
    default:
        DoSomething();
}
```

Exceptions

This rule does not apply to `if` chains without `else`-s, or to `switch`-es without default clauses.

```
if (b == 0) //no issue, this could have been done on purp
{
    DoSomething();
}
else if (b == 1)
{
    DoSomething();
}
```

Available In:

 Code Smell
Collapsible "if" statements should be merged
 Code Smell
OS commands should not be vulnerable to argument injection attacks
 Vulnerability
Logging should not be vulnerable to injection attacks
 Vulnerability
Empty collections should not be