

Kodstandard - No Silver Bullet

Regions

Regions ska användas för lättare navigation inom koden.

```
/// Region Order
///
/// Member variables
/// Properties
/// Events
/// Constructors
/// Public methods
/// Protected methods
/// Private methods
///
```

Kodexempel

Kommentarer i denna mängd finns endast i exempel. Koden i projektet ska sträva mot att vara självkommenterande och endast använda sig av syftesförklarande kommentarer inom speciellt behov.

- Namn ska alltid vara väl beskrivande. Långa namn undviks inte vid behov.
- All kod ska skrivas på engelska.
- Metoder ska hållas inom rimlig storlek och brytas ner vid behov.
- Hårdkodade värden ska aldrig förekomma.

```
if (true)
{
    //Code
}
else
{
    //Other code
}
```

```

15 namespace Kodstandard
16 {
17     class KodstandardExempel
18     {
19         #region Member variables
20         const int CoolNumber = 3;           //Ska börja med storbokstav
21         readonly int CoolerNumber;         //Ska börja med storbokstav
22
23         int myNumber;                       //Ska börja med my
24         #endregion
25
26         #region Properties
27         public int Number                   //Ska börja med storbokstav
28         {
29             get;
30             private set;
31         }
32         #endregion
33
34         #region Events
35
36         #endregion
37
38         #region Constructors
39         public KodstandardExempel(int aNumber) //Ska börja med storbokstav
40         {                                       //Parametrar ska börja med a
41             CoolerNumber = 5;
42             myNumber = aNumber;
43         }
44         #endregion
45
46         #region Public methods
47         public void PublicMethod(int aNumber)
48         {
49             int normalVariable = 3;          //Ska börja med liten bokstav (Inget a / my för benämning)
50         }
51         #endregion

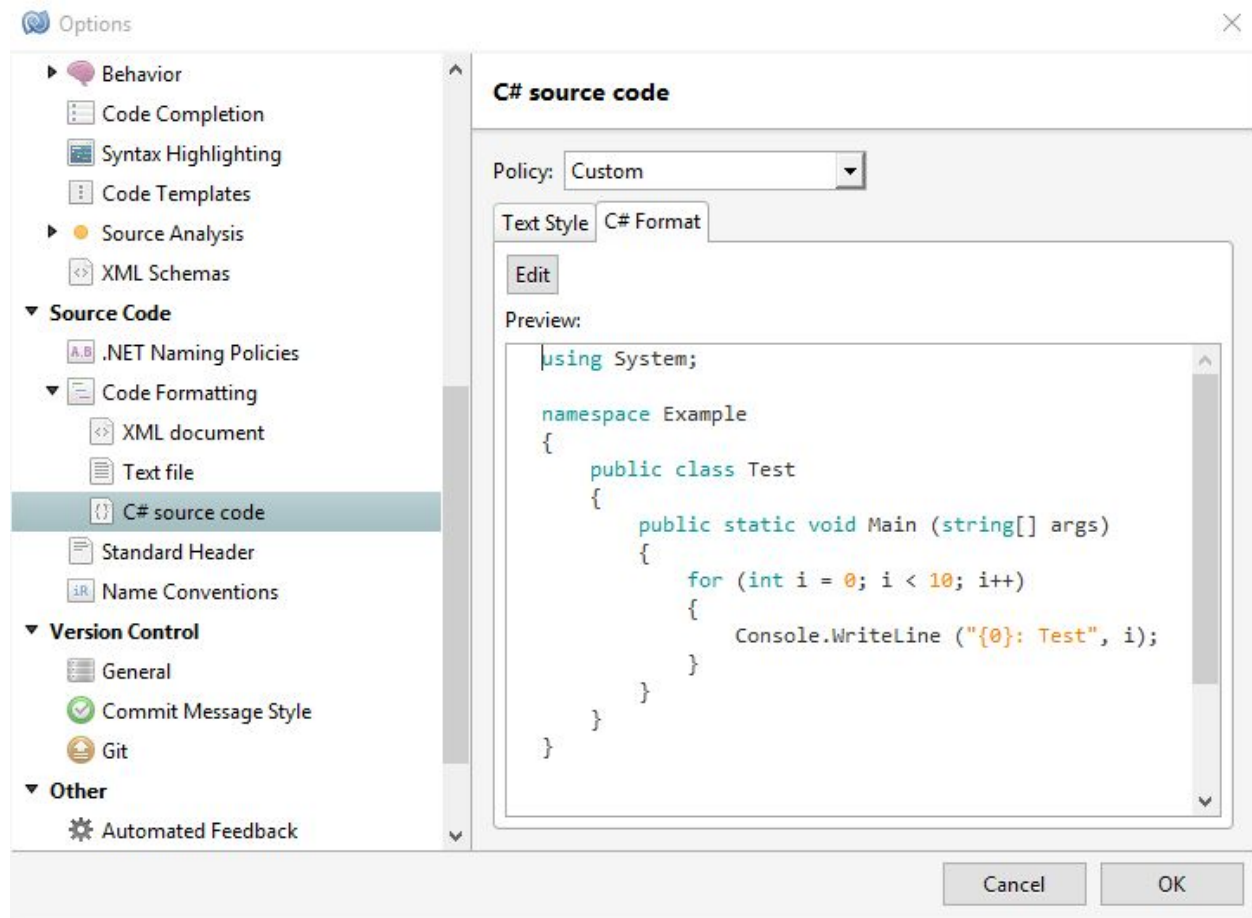
```

```

52
53         #region Protected methods
54         protected void ProtectedMethod(int aNumber)
55         {
56
57         }
58         #endregion
59
60         #region Private methods
61         private void PrivateMethod(int aNumber)
62         {
63
64         }
65         #endregion
66     }
67 }
68

```

Monodevelope



Edit Profile

Category: **Indentation**

Declarations

- ☒ within namespaces
- ☒ within classes
- ☒ within interfaces
- ☒ within structs
- ☒ within enums
- ☒ within methods
- ☒ within properties
- ☒ within events

Statements

- ☒ within blocks
- ☐ Indent 'switch' body
- ☒ Indent 'case' body
- ☒ Indent 'break' statements
- ☒ Indent pre processor directives
- ☐ Indent blocks inside expressions
- ☐ Align embedded statements

Preview:

Cancel OK

Edit Profile

Category: **Braces**

<input checked="" type="checkbox"/> Namespace declaration	Next line
<input checked="" type="checkbox"/> Class declaration	Next line
<input checked="" type="checkbox"/> Interface declaration	Next line
<input checked="" type="checkbox"/> Struct declaration	Next line
<input checked="" type="checkbox"/> Enum declaration	Next line
<input checked="" type="checkbox"/> Method declaration	Next line
<input checked="" type="checkbox"/> Anonymous methods	Next line
<input checked="" type="checkbox"/> Constructor declaration	Next line
<input checked="" type="checkbox"/> Destructor declaration	Next line
<input checked="" type="checkbox"/> Statements	Next line
Property declaration	Next line
<input checked="" type="checkbox"/> Allow automatic property in one line	Force new line
<input checked="" type="checkbox"/> Allow simple property in one line	Force new line
<input checked="" type="checkbox"/> Get declaration	Next line
<input checked="" type="checkbox"/> Allow one line get	Allow one line
<input checked="" type="checkbox"/> Set declaration	Next line
<input checked="" type="checkbox"/> Allow one line set	Allow one line
Event declaration	Next line
<input checked="" type="checkbox"/> Add declaration	Next line
<input checked="" type="checkbox"/> Allow one line add	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Remove declaration	Next line
<input checked="" type="checkbox"/> Allow one line remove	<input checked="" type="checkbox"/>

Preview:

```

class ClassDeclaration
{
    int myProperty;

    int MyProperty
    {
        get { return myProperty; }
        set { myProperty = value; }
    }

    string Simple
    {
        get { ; }
        set { ; }
    }

    int myOtherProperty;

    int MyOtherProperty
    {
        get
        {
            Console.WriteLine ("get myOtherProperty");
            return myOtherProperty;
        }
        set
        {
            if (myOtherProperty != value)
                myOtherProperty = value;
        }
    }
}

```

Cancel OK

Edit Profile

Category: Blank lines

Preview:

Minimum blank lines in compilation unit

Before using declaration: 0

After using declaration: 1

Minimum blank lines in namespaces

Before first declaration: 0

Between type declarations: 1

Minimum blank lines in types

Between field declarations: 0

Between simple event declarations: 0

Between other member declarations: 1

Around region: 1

Inside region: 1

```
// Example
using System;
using System.Collections;

namespace TestSpace
{
    using MyNamespaces;

    class Test
    {
        int a;
        string b;

        public Test (int a, string b)
        {
            this.a = a;
            this.b = b;
        }

        void Print ()
        {
            Console.WriteLine ("a: {0} b : {1}", a, b);
        }
    }

    class MyTest
    {
    }
}
```

Cancel OK

Edit Profile

Category: Wrapping

Preview:

Method declarations

Parameters Do not change

New line after open parentheses Allow both

New line before closing parentheses Allow both

Align to first parameter ☒

Method calls

Arguments Do not change

New line after open parentheses Allow both

New line before closing parentheses Allow both

Align to first argument ☐

Indexer declarations

Parameters Do not change

New line after open parentheses Allow both

New line before closing parentheses Allow both

Align to first parameter ☒

Indexer usage

Arguments Do not change

New line after open bracket Allow both

New line before closing bracket Allow both

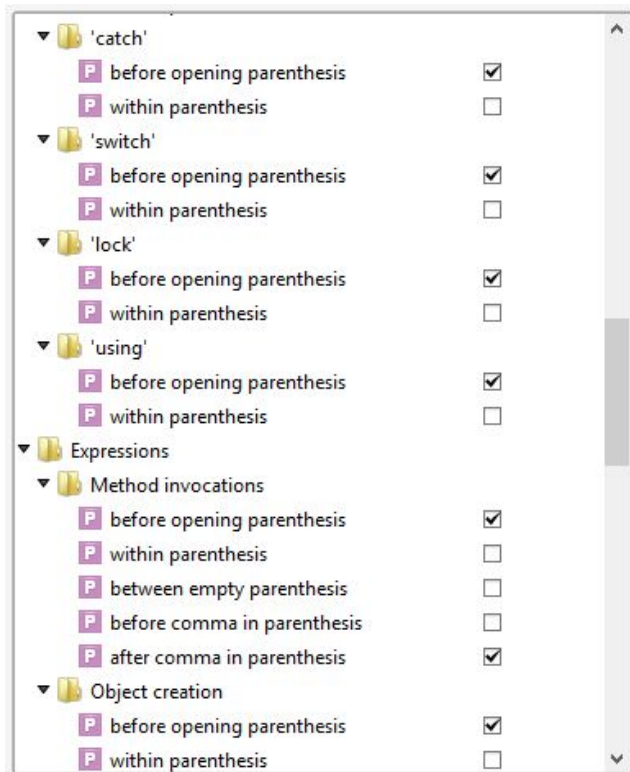
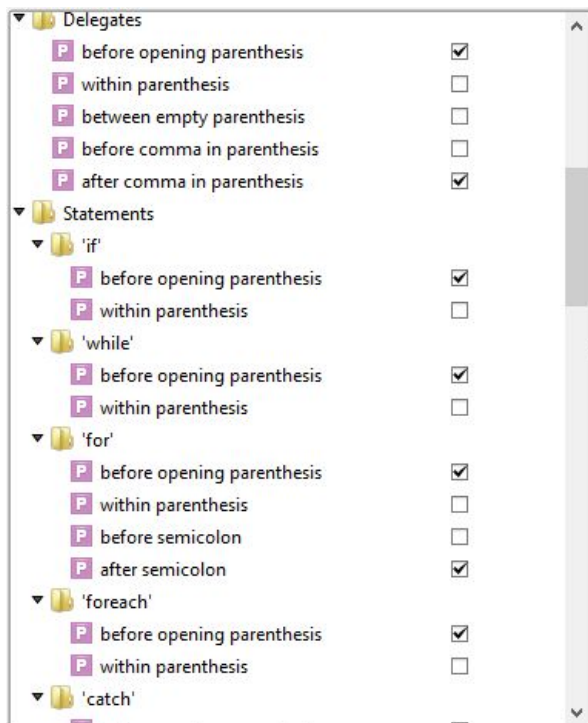
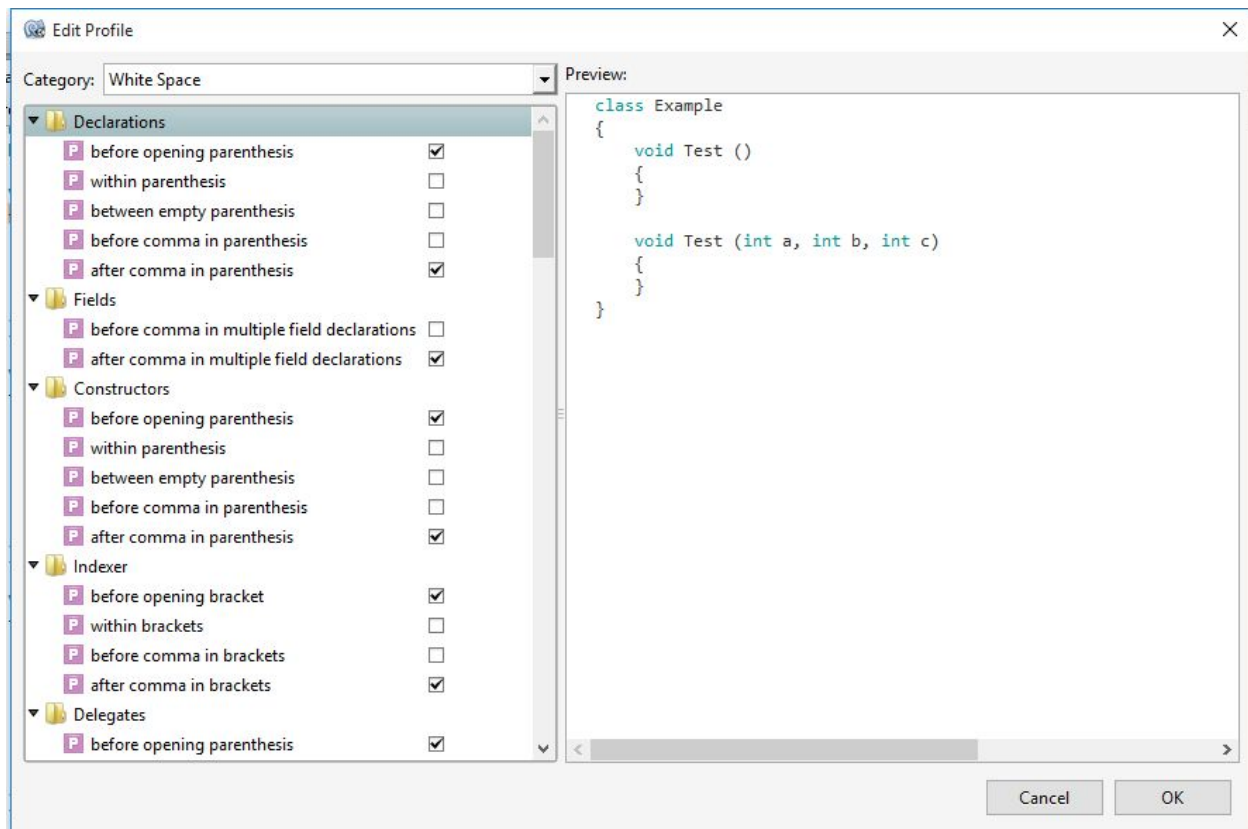
Align to first parameter ☐

```
class ClassDeclaration
{
    public void LongMethodCallInSameLine (int test, string foo,
    {
    }

    public void LongMethodCallInMultiple (
        int test,
        string foo,
        double bar)
    {
    }

    public void LongMethodCallInMultipleCase2 (int test,
                                                string foo,
                                                double bar)
    {
    }
}
```

Cancel OK



▼ Object creation

- ☒ before opening parenthesis
- ☐ within parenthesis
- ☐ between empty parenthesis
- ☐ before comma in parenthesis
- ☒ after comma in parenthesis

▼ Element access

- ☒ before opening bracket
- ☐ within brackets
- ☐ before comma in brackets
- ☒ after comma in brackets

▼ Parentheses

- ☐ within parenthesis

▼ Type cast

- ☐ within parenthesis
- ☐ after type cast

▼ 'sizeof'

- ☐ before opening parenthesis
- ☐ within parenthesis

▼ 'typeof'

- ☐ before opening parenthesis
- ☐ within parenthesis

▼ Around Operators

- ☒ Assignment (=, +=, -=, ...)
- ☒ Logical (&&, ||) operators
- ☒ Equality (==, !=) operators
- ☒ Relational (<, >, <=, >=) operators
- ☒ Bitwise (&, |, ^, ~) operators
- ☒ Additive (+, -) operators
- ☒ Multiplicative (*, /, %) operators
- ☒ Shift (<<, >>) operators
- ☒ Null coalescing (??) operator
- ☐ Unsafe addressof operator (&)
- ☐ Unsafe asterisk operator (*)
- ☐ Unsafe arrow operator (->)

▼ Conditional Operator (?:)

- ☒ before '?'
- ☒ after '?'
- ☒ before ':'
- ☒ after ':'

▼ Array Declarations

- ☐ before opening bracket

▼ Other

- ☐ before semicolon

Edit Profile

Category: New Lines

<input checked="" type="checkbox"/> Place 'else' on new line	Always new line
<input checked="" type="checkbox"/> Place 'else if' on new line	Allow both
<input checked="" type="checkbox"/> Place 'catch' on new line	Always new line
<input checked="" type="checkbox"/> Place 'finally' on new line	Always new line
<input checked="" type="checkbox"/> Place 'while' on new line	Always new line
<input checked="" type="checkbox"/> Place array initializers on new line	Wrap if too long
<input checked="" type="checkbox"/> Place embedded statements on new line	Always new line
<input checked="" type="checkbox"/> Place constructor initializer ':' on new line	Always new line
<input checked="" type="checkbox"/> Place constructor initializer 'base/this' on new line	Always same line

Preview:

```

class ClassDeclaration
{
    public void Test (int i)
    {
        if (i == 5)
        {
            Console.WriteLine ("== 5");
        }
        else if (i > 0)
        {
            Console.WriteLine (">0");
        }
        else if (i < 0)
        {
            Console.WriteLine ("<0");
        }
        else
        {
            Console.WriteLine ("== 0");
        }
    }
}

```

Cancel OK