Coding Conventions

For a good discussion on coding conventions in general take a look at this page.

The following conventions must be followed by any code committed to our repository (which also means that the source files already in the repository do serve as good examples).

Naming

Type, method and property names use Pascal case. Interface names are prefixed with a capital **I** as in **IAstAttribute** .

Public field names also use Pascal case.

Non public field names use Camel case prefixed by an underscore character (_) as in **_assemblyReferences**.

> *const/readonly/final non public fields might also use Pascal case.*

Local variable (including parameter) names use Camel case.

> *Module and private method names might also use Camel case.*

Indentation

Tabs are used for indentation not spaces.

A new block increments the indentation level by 1.

Braces placement (C#)

Braces are always placed in their own line and at the same level of indentation of the previous line.

Single line blocks are always delimited by braces.

C# Example

```
namespace Boo.Lang
{
    using System;
    using System.Collections;
    using System.Text;

    /// <summary>
    /// List.
    /// </summary>
```

```
    [Serializable]
    public class List : IList
    {
        static readonly object[]
EmptyObjectArray = new object[0];

        protected object[] _items;

        protected int _count;

        public List()
        {
            _items = EmptyObjectArray;
            _count = 0;
        }

         // ....
        public List Multiply(int count)
        {
            if (count < 0)
            {
                throw new
ArgumentOutOfRangeException("count");
            }

            object[] items = new
object[_count*count];
            for (int i=0; i<count; ++i)
            {
                Array.Copy(_items, 0, items,
i*_count, _count);
```

```
        }
        return new List(items, true);
    }
    // ...
    int NormalizeIndex(int index)
    {
        if (index < 0)
        {
            index += _count;
        }
        return index;
    }
```

```
        // ...
    }
}
```

boo example

```
class BooEditor(ScrolledWindow):

    static _booSourceLanguage =
SourceLanguagesManager().GetLanguageFromMime
Type("text/x-boo")

    [getter(Buffer)]
    _buffer =
SourceBuffer(_booSourceLanguage,
                                Highlight: true)

    _view = SourceView(_buffer,
                                ShowLineNumbers:
true,
                                AutoIndent:
true,
                                TabsWidth: 4)

    [getter(FileName)]
    _fname as string

    Label:
        get:
            return
```

```
System.IO.Path.GetFileName(_fname) if _fname
            return "unnamed.boo"

    def constructor():
        self.SetPolicy(PolicyType.Automatic,
PolicyType.Automatic)
        self.Add(_view)

    def Open([required] fname as string):
        _buffer.Text =
TextFile.ReadFile(fname)
        _buffer.Modified = false
        _fname = fname

    def SaveAs([required] fname as string):
        TextFile.WriteFile(fname,
```

```
_buffer.Text)
        _fname = fname
        _buffer.Modified = false
```