

Boo Explorer

In the extras directory you're going to find this neat little utility that allows you not only to edit boo scripts with a nice syntax highlighting but also to run them (F5) and/or see their expanded form (CTRL+E).

For Windows Users

booxw can be found in binary form [here](#).

If you want to build it yourself follow these simple steps:

- go to the extras/boox directory
 - type nant
- ⚠ boox requires at least nant 0.85, which is a nightly build version, not a stable version.

Here's a screenshot of boox in action to whet your appetite:

Boo Explorer

File Edit Tools View

F:\Temp\zgen.boo F:\Temp\zconfig.boo (modified) F:\Temp\zctest.boo untitled.boo

```
1 namespace ZEngine.Config
2
3 import Boo.Lang.Compiler.Ast
4
5 class ConfigManager:
6     >> protected def constructor([required] fileName as string):
7         >> fileName.
8         >>
9         >> static def AppendInPlace(ArrayLength Capacity Chars memberName as string, readMeth
10            >> return C
11
12     >> def Bind(className ClearPostNullChar memberName as string, readMeth
13         >> pass
14         >>
15     >> def Save():
16         >> pass
17         >>
18 interface IConfigurable:
19     >> ConfigMgr as ConfigManager:
20         >> set
21
22     >> def Configure()
23
24 class ConfigTypeConverter:
25     >> static def Convert() as object:
26         >> pass
27
28 class ConfigAttribute(Boo.Lang.Compiler.AbstractAstAttribute):
29     >>
30     >> _fieldName as ReferenceExpression
31     >>
32     >> def constructor(fieldName as ReferenceExpression):
33         >> _fieldName = fieldName
34         >>
35     >> override def Apply(node as Node):
36         >> f = node as Field
37         >> if f is null:
38             >> InvalidNodeForAttribute("Field")
```

AppendInPlace System.String.AppendInPlace(System.Char, System.Int32) (...

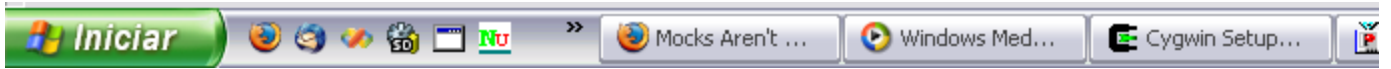
ArrayLength Capacity Chars memberName as string, readMeth

ClearPostNullChar memberName as string, readMeth

Clone CompareTo CopyTo CopyToByteArray EndsWith

Output

Task List Output



For Linux Users

I've started a similar project using Gtk# but it's in a very early stage. If you know or want to learn how to write Gtk# applications this could be a good chance.