Using IronPython From Boo

If you are more familiar with the python standard classes and objects, such as strings and dictionaries, you can use them via IronPython.

In your boo project, add a reference to the IronPython.dll included with IronPython, and then you can use code like this:

```
import IronPython.Objects //for Dict, Str
import IronPython.Modules.__builtin__ //for eval

//using the python dictionary:
//this converts a boo Hash to a python dictionary:
d = Dict({"key1": "value1", "key2": "value2"})
for item in d:
        print item, ":", d[item]
print d.__len__()
print len(d)
print d.get("badkey", "default1")


//using python string methods:
s = Str("firstname lastname")
for word in s.split():
 print Str(word).capitalize()
print s.ToString() //convert back to .NET string class


//eval simple python code
pythonglobals = {"X": 1, "Y": 2}
pythonlocals = null
pythonsrc = "float(X) / Y"
try:
 result = eval(pythonsrc, pythonglobals, pythonlocals)
 print "result of \"${pythonsrc}\" is:", result, "(type:",
result.GetType(), ")"
except e:
 print "Error:", e.Message
```

Running full python scripts is not working:

```
import IronPython.Objects //for Dict, Str
import IronPython.Modules //sys
import IronPython.Modules.__builtin__ //for eval
import IronPython.AST

//compile a python script
pythonsrc = """
from System.Windows.Forms import *
f = Form(Text="Experiment #1")
f.ShowDialog()
"""

sys.path.append(System.Environment.CurrentDirectory)
sys.LoadAssemblyByName("System.Windows.Forms")

topframe = Frame(module())
p = Parser.fromString(pythonsrc)
stmt = p.parseStmt()
code as FrameCode = SnippetMaker.generate(stmt, "input")
code.Run(topframe)
```