Smuggle

Organized Python Pickles

Log, catalogue, and move python objects via pickling

Jeffrey McLarty

2015/02/01

What if we could…

- Easily Pickle Objects
- Easily Catalogue Pickles
- Easily Read the Pickles
- Easily Read the Catalogue

Why?

Reduce Development Time

- Eliminate guessing which details will be needed in error messages (for developer types only)
  - Eg. ErrorMsg = “ERROR : object’s important attribute = {0}”.format(obj.attribute)

Reduce Debugging Time

- Eliminate recreation of error cases in order to simply identify a problem.
- Investigate how an object has changed over time.
  - Logs are great, so are commit diffs, but object differences can shed light on changes extremely quickly.

Reduce Down Time

- Move production objects to new sessions and scripts on development environments.

Vocabulary

- Smuggler ➔ Pickler + Cataloguer of Objects
- Smuggle ➔ The Action of Pickling + Cataloguing
- Contraband ➔ Object getting Pickled + Catalogued
- Payload ➔ Catalogue Reader + Pickle Loader
- Passphrase(s) ➔ Prebuilt python code to load a catalogued pickle

Vocabulary

- Smuggler ➔ Pickler + Cataloguer of Objects
    - [Writer Object]
- Smuggle ➔ The Action of Pickling + Cataloguing
    - [Writing Process]
- Contraband ➔ Object getting Pickled + Catalogued
    - [Any Object (which can be pickled)]
- Payload ➔ Catalogue Reader + Pickle Loader
    - [Reader Object == List of Contraband]
- Passphrase(s) ➔ Prebuilt python code to load a catalogued pickle
    - [Access Code to Object]

Basic Use of Smuggler

MyScript.py

```
HanSolo = Smuggler("C:\Smuggle")
HanSolo.smuggle(foo="foo stuff", boo="boo stuff")
print HanSolo.passphrases()
```

Output

```
import pickle

# boo of type 'str' was smuggled at 22:50:24, 2014/11/15
#   'boo stuff'
boo = pickle.load(open(r"C:\Smuggle\boo-2014-11-15-22-50-24.smug","rb"))

# foo of type 'str' was smuggled at 22:50:24, 2014/11/15
#   'foo stuff'
foo = pickle.load(open(r"C:\Smuggle\foo-2014-11-15-22-50-24.smug","rb"))
```

…for instance, in a traceback

MyScript.py

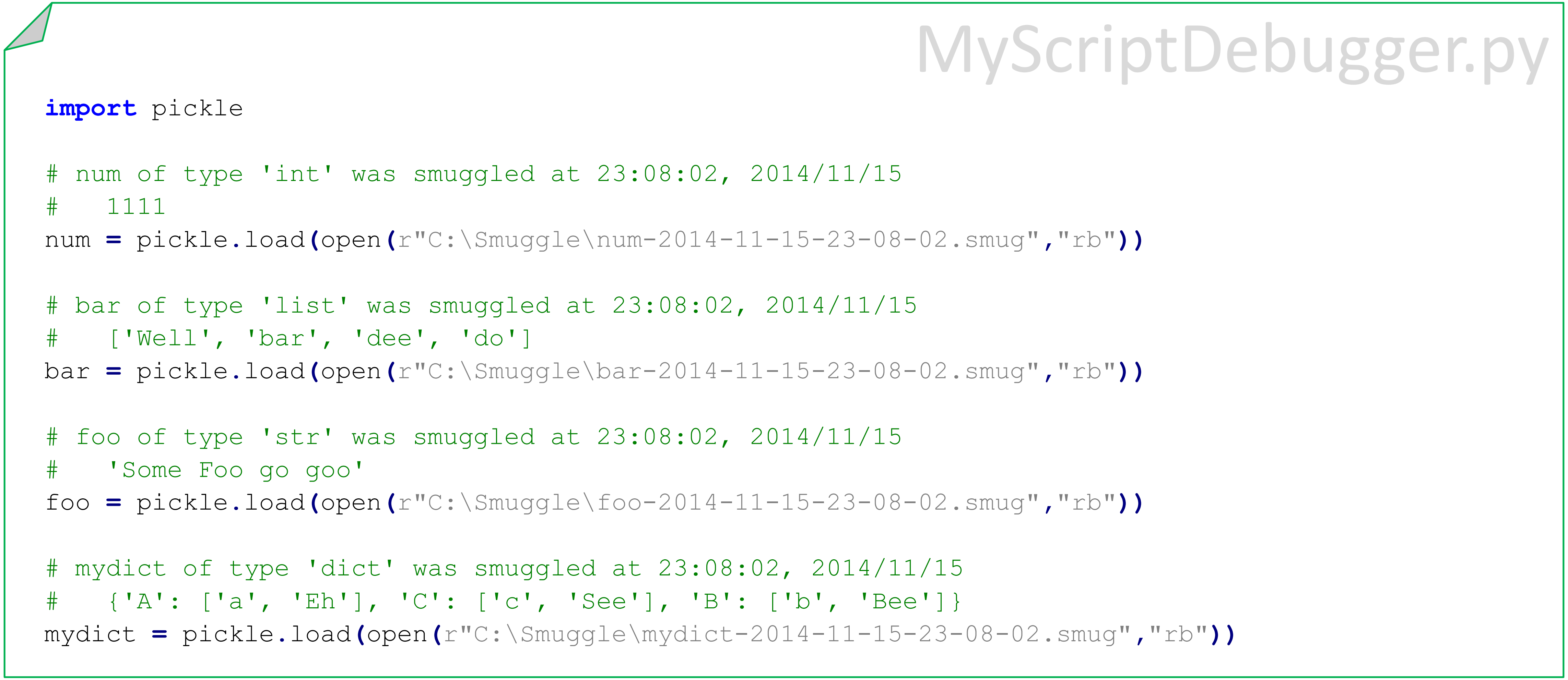
```
HanSolo = Smuggler("C:\Smuggle")
   try:
       foo, bar = "Some Foo go goo", "Well bar dee do"
       adict = {'A' : ['a','Eh'], 'B' : ['b','Bee'], 'C' : ['c','See'] }
       i, j, k = 1, 10, 100
       DoSomethingExceptional()
   except:
       HanSolo.smuggle(foo=foo,            # plain vanilla,
                       bar=bar.split(" "), # hacked up, if needed
                       mydict=adict,       # renamed? sure!
                       num=i+j+k+1000)     # calculation, ok!
       ErrorMsg = "Don't Do Exceptional Things!\n" + HanSolo.passphrases()
       raise Exception(ErrorMsg)
```

…you get the idea…

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\WinPython-32bit-2.7.6.4\python-2.7.6\lib\site-packages\spyderlib\widgets\externalshell\sitecustomize.py", line 540, in runfile
    execfile(filename, namespace)
  File "C:/Users/Jeffrey/Desktop/Projects/smuggle/main2.py", line 210, in <module>
    raise Exception(ErrorMsg)
Exception: Don't Do Exceptional Things!
```

```
import pickle

# num of type 'int' was smuggled at 23:08:02, 2014/11/15
#   1111
num = pickle.load(open(r"C:\Smuggle\num-2014-11-15-23-08-02.smug","rb"))

# bar of type 'list' was smuggled at 23:08:02, 2014/11/15
#   ['Well', 'bar', 'dee', 'do']
bar = pickle.load(open(r"C:\Smuggle\bar-2014-11-15-23-08-02.smug","rb"))

# foo of type 'str' was smuggled at 23:08:02, 2014/11/15
#   'Some Foo go goo'
foo = pickle.load(open(r"C:\Smuggle\foo-2014-11-15-23-08-02.smug","rb"))

# mydict of type 'dict' was smuggled at 23:08:02, 2014/11/15
#   {'A': ['a', 'Eh'], 'C': ['c', 'See'], 'B': ['b', 'Bee']}
mydict = pickle.load(open(r"C:\Smuggle\mydict-2014-11-15-23-08-02.smug","rb"))
```

…then copy + paste:



MyScriptDebugger.py

```python
import pickle

# num of type 'int' was smuggled at 23:08:02, 2014/11/15
#   1111
num = pickle.load(open(r"C:\Smuggle\num-2014-11-15-23-08-02.smug","rb"))

# bar of type 'list' was smuggled at 23:08:02, 2014/11/15
#   ['Well', 'bar', 'dee', 'do']
bar = pickle.load(open(r"C:\Smuggle\bar-2014-11-15-23-08-02.smug","rb"))

# foo of type 'str' was smuggled at 23:08:02, 2014/11/15
#   'Some Foo go goo'
foo = pickle.load(open(r"C:\Smuggle\foo-2014-11-15-23-08-02.smug","rb"))

# mydict of type 'dict' was smuggled at 23:08:02, 2014/11/15
#   {'A': ['a', 'Eh'], 'C': ['c', 'See'], 'B': ['b', 'Bee']}
mydict = pickle.load(open(r"C:\Smuggle\mydict-2014-11-15-23-08-02.smug","rb"))
```

Caught* without a trace?

AdvancedScriptDebug.py

```
MyPayload = Payload("C:\Smuggle")
VarTimeDict = MyPayload.asvartimedict2D()

for varname in VarTimeDict.keys():
    for datetime in VarTimeDict[varname].keys():
        print varname, datetime, VarTimeDict[varname][datetime]
```

Output

```
…
foo 2014-11-15 23:17:12 old Foo go goo
foo 2014-11-15 22:44:36 old Foo go goo
foo 2014-11-15 22:03:35 Some Foo go goo
foo 2014-11-15 22:10:25 Some Foo go goo
foo 2014-11-15 22:50:24 Some Foo go goo
…
```

*pun intended, I love that one…gets me every time.

Access a “Log” of Objects using…

- Passphrases()
- Asvardict()
- Astimevardict1D()
- Astimevardict2D()
- Asvartimedict2D()
- Aslist(objnames)

Catalogue?

Right now

- Just a hardcoded folder + filename convention

Next Steps

- An index file + folder + filename
- Optional SQLite Storage