Working with System Frameworks in Python and Objective-C

by James Barclay

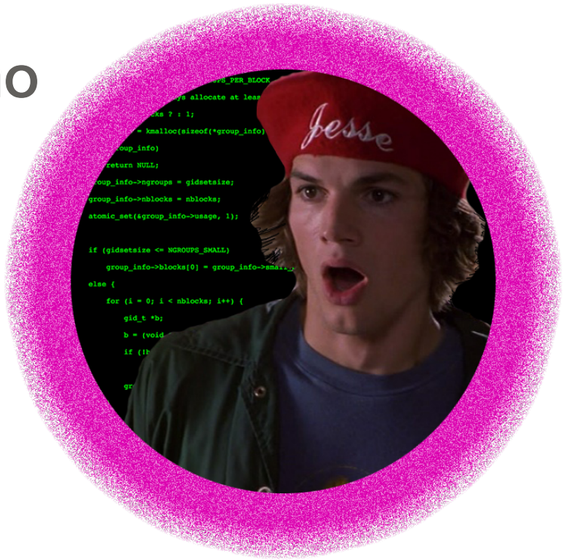
bit.ly/psudemo

Dude, Where's My Source Code?

CODE

https://github.com/futureimperfect/psu-pyobjc-demo

https://github.com/futureimperfect/PSUDemo



SLIDES

https://github.com/futureimperfect/slides

Agenda

1. What are system frameworks, and why should you care?
2. Brief overview of the frameworks, classes, and APIs that will be demonstrated.
3. Demo 1: PyObjC
4. Demo 2: Objective-C
5. Wrap up and questions.

What's a System Framework?

...and why should you care?

(OS X) system frameworks provide interfaces you need to write software for the Mac.

Many of these are useful for Mac admins creating:

- scripts
- GUI applications
- command-line tools

Learning about system frameworks will teach you more about OS X, which will probably make you a better admin.

```
/*  Foundation.h
    Copyright (c) 1994-2014, Apple Inc. All rights reserved.
*/


#include <CoreFoundation/CoreFoundation.h>

#import <Foundation/NSObjCRuntime.h>

#import <Foundation/NSArray.h>
#import <Foundation/NSAutoreleasePool.h>
#import <Foundation/NSBundle.h>
#import <Foundation/NSByteOrder.h>
#import <Foundation/NSCalendar.h>
#import <Foundation/NSCharacterSet.h>
#import <Foundation/NSCoder.h>
#import <Foundation/NSData.h>
#import <Foundation/NSDate.h>
#import <Foundation/NSDateFormatter.h>
#import <Foundation/NSDateIntervalFormatter.h>
#import <Foundation/NSMassFormatter.h>
#import <Foundation/NSLengthFormatter.h>
#import <Foundation/NSEnergyFormatter.h>
#import <Foundation/NSDecimal.h>
#import <Foundation/NSDecimalNumber.h>
#import <Foundation/NSDictionary.h>
#import <Foundation/NSEnumerator.h>
#import <Foundation/NSError.h>
#import <Foundation/NSException.h>
#import <Foundation/NSFileHandle.h>
#import <Foundation/NSFileManager.h>
#import <Foundation/NSFormatter.h>
#import <Foundation/NSHashTable.h>
#import <Foundation/NSHTTPCookie.h>
#import <Foundation/NSHTTPCookieStorage.h>
#import <Foundation/NSIndexPath.h>
#import <Foundation/NSIndexSet.h>
#import <Foundation/NSInvocation.h>
#import <Foundation/NSJSONSerialization.h>
#import <Foundation/NSKeyValueCoding.h>
#import <Foundation/NSKeyValueObserving.h>
#import <Foundation/NSKeyedArchiver.h>
#import <Foundation/NSLocale.h>
#import <Foundation/NSLock.h>
#import <Foundation/NSMapTable.h>
#import <Foundation/NSMethodSignature.h>
#import <Foundation/NSNotification.h>
#import <Foundation/NSNotificationQueue.h>
#import <Foundation/NSNull.h>
#import <Foundation/NSNumberFormatter.h>
#import <Foundation/NSObject.h>
#import <Foundation/NSOperation.h>
#import <Foundation/NSOrderedSet.h>
#import <Foundation/NSOrthography.h>
#import <Foundation/NSPathUtilities.h>
#import <Foundation/NSPointerArray.h>
#import <Foundation/NSPointerFunctions.h>
/System/Library/Frameworks/Foundation.framework/Versions/C/Headers/Foundatio
"/System/Library/Frameworks/Foundation.framework/Versions/C/Headers/Foundati
```

Frameworks, Classes, and APIs oh my!

Cocoa

- Foundation
  - NSFileManager
  - NSTask
  - NSURLSession
  - NSUserDefaults
- AppKit
  - NSApplication

CoreFoundation

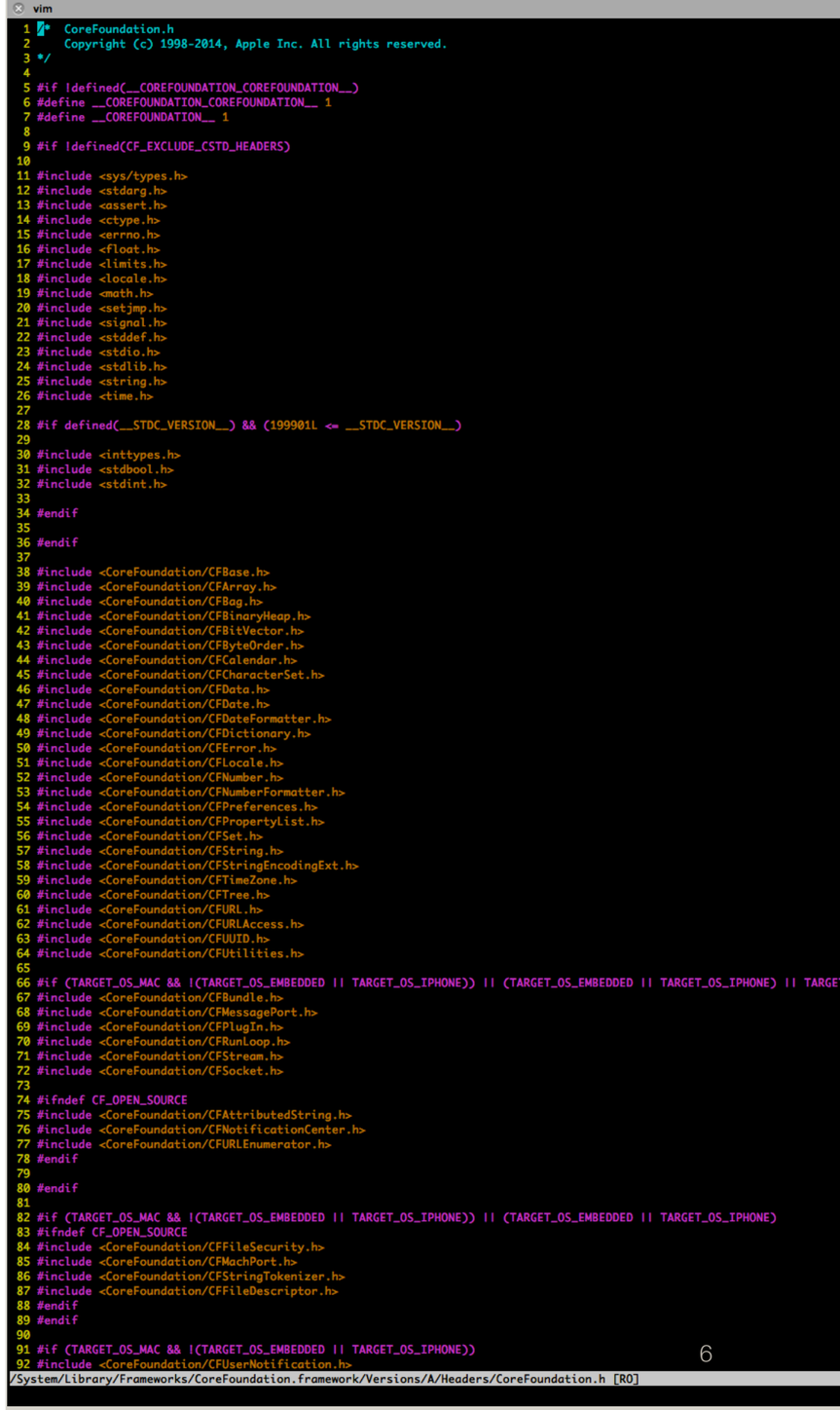
- CFPreferences

CoreGraphics

- Quartz

Quartz is a graphics layer that is often used synonymously with CoreGraphics. We won't be going into depth on this, just showing how to get the current system display state in Python.

This is the obligatory slide showing all the things we'll be talking about and using during this presentation.



PyObjC

colons -> underscores

Pros

- Allows Python programmers to take advantage of OS X system toolkits.
- Python is powerful and easy to pick up.

Cons

- Probably fewer answers on Stack Overflow.
- Can be awkward to use.
  - All colons following selectors are replaced with underscores, which can make the intent more ambiguous.
    - `[myCar refuelWith:gas replaceCap:@YES];`
    - `myCar.refuelWith_replaceCap_(gas, True)`

Demo 1

System Frameworks in Python

Quartz

```
def is_computer_locked():
    d = Quartz.CGSessionCopyCurrentDictionary()
    if d.get('kCGSSessionOnConsoleKey') is False or d.get(
            'CGSSessionScreenIsLocked') is True:
        return True
    return False
```

NSWorkspace and NSApplication

```
def block_user():
    ws = NSWorkspace.sharedWorkspace()
    ws.hideOtherApplications()
    # OR'ing the desired options yields the correct result.
    options = NSApplicationPresentationHideDock | \
        NSApplicationPresentationDisableProcessSwitching | \
        NSApplicationPresentationDisableForceQuit | \
        NSApplicationPresentationDisableSessionTermination
    NSApp().setPresentationOptions_(options)


def unblock_user():
    options = NSApplicationPresentationDefault
    NSApp().setPresentationOptions_(options)
```

CFPreferences(Copy|Set)AppValue

```
def get_pref_val(key, domain):
    val = CFPreferencesCopyAppValue(key, domain)
    return val
```

```
def set_pref_val(key, data, domain):
    CFPreferencesSetAppValue(key, data, domain)
    CFPreferencesAppSynchronize(domain)
```

CFPreferencesAppValueIsForced

```
# Check if the 'askForPassword' preference is managed.
ask_for_password_managed = CFPreferencesAppValueIsForced(
    'askForPassword',
    'com.apple.screensaver'
)
print_color(
    'Managing the screensaver askForPassword preference:'
)
print('\t%s' % ask_for_password_managed)
```

NSWorkspace

```
def get_running_apps():
    ws = NSWorkspace.sharedWorkspace()
    running_apps = ws.runningApplications()
    apps = []
    for app in running_apps:
        apps.append(app.localizedName())
    apps = set(apps)

    return apps
```

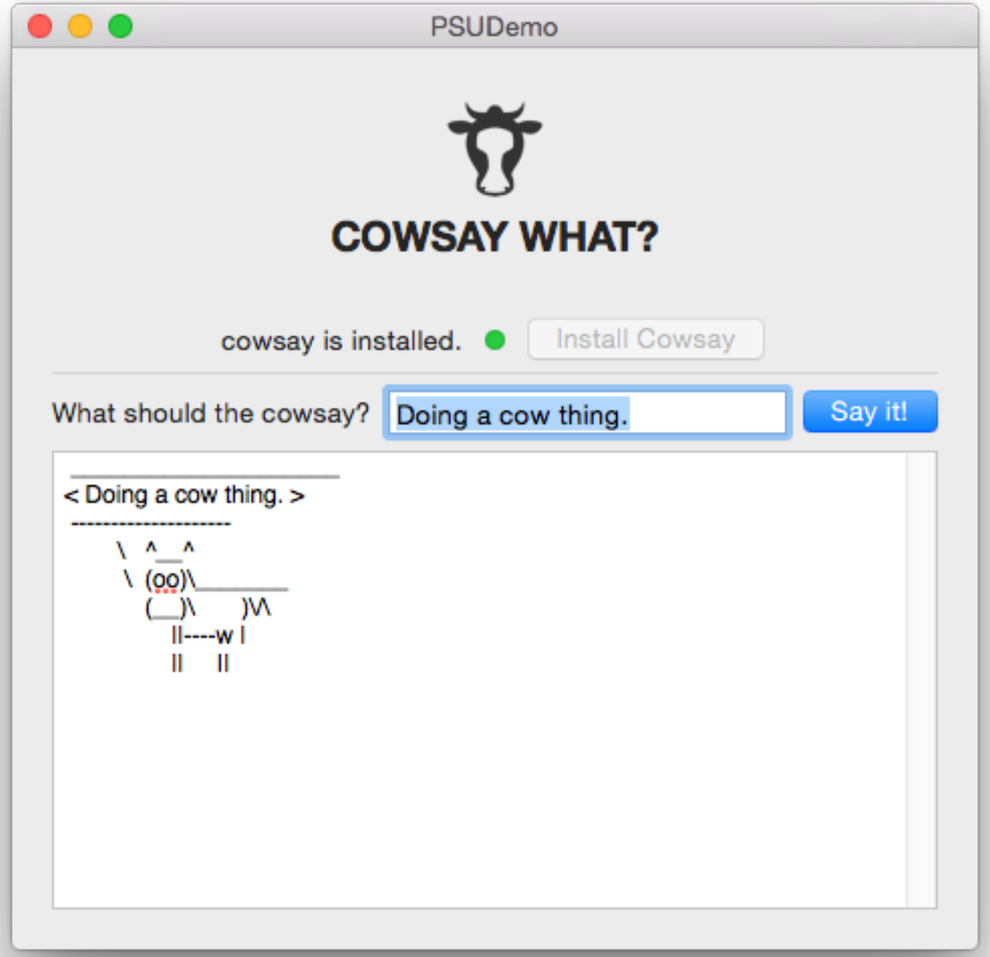
You're going to love this if you have a thing for square brackets and ridiculously long method names, (e.g., dateWithTimeIntervalSince1970).

Demo 2

System Frameworks in Objective-C

What we'll be building.

!usefulForSystemAdministrators



The session description says we'll be learning about "useful system frameworks for system administrators. We'll do this by demonstrating how to create the most _useless_ app ever.

What We Covered

Basic usage of CFPreferences, AppKit, and Quartz in Python.

- How to retrieve and set preference values.
- How to determine if an application preference value is forced.
- How to access the kiosk mode APIs and set presentation options.
- How to determine if a Mac’s screen is locked or if a console user is logged in.

Basic usage of NSTask, NSURLSession, NSUserDefaults, etc.

- How to make a cow app.

References

- https://developer.apple.com/library/mac/documentation/CoreFoundation/Conceptual/CFPreferences/Concepts/BestPractices.html
- https://developer.apple.com/library/mac/documentation/CoreFoundation/Reference/CFPreferencesUtils/
- https://developer.apple.com/library/mac/technotes/KioskMode/Introduction/Introduction.html
- http://www.gnu.org/software/gnustep/resources/documentation/Developer/Gui/Reference/NSApplication.html
- https://developer.apple.com/library/mac//documentation/Cocoa/Reference/ApplicationKit/Classes/NSApplication_Class/index.html
- http://www.cocoawithlove.com/2009/01/demystifying-nsapplication-by.html
- https://developer.apple.com/library/prerelease/ios/documentation/Cocoa/Reference/Foundation/Classes/NSFileManager_Class/index.html
- http://nshipster.com/nsfilemanager/
- https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSTask_Class/
- https://developer.apple.com/library/ios/documentation/Foundation/Reference/NSURLSession_class/
- https://developer.apple.com/library/prerelease/ios/documentation/Cocoa/Reference/Foundation/Classes/NSUserDefaults_Class/index.html
- http://pythonhosted.org/pyobjc/
- http://en.wikipedia.org/wiki/Core_Foundation
- http://ridiculousfish.com/blog/posts/bridge.html
- https://developer.apple.com/library/ios/documentation/CoreFoundation/Conceptual/CFDesignConcepts/CFDesignConcepts.html
- https://developer.apple.com/library/prerelease/ios/documentation/CoreFoundation/Reference/CoreFoundation_Collection/index.html
- https://developer.apple.com/library/mac/documentation/MacOSX/Conceptual/OSX_Technology_Overview/SystemFrameworks/SystemFrameworks.html

Questions?

So Long, and Thanks for All the Cows