

NSUserDefaults in Swift

Knoxville CocoaHeads knoxcocoa.org

April 19th, 2016

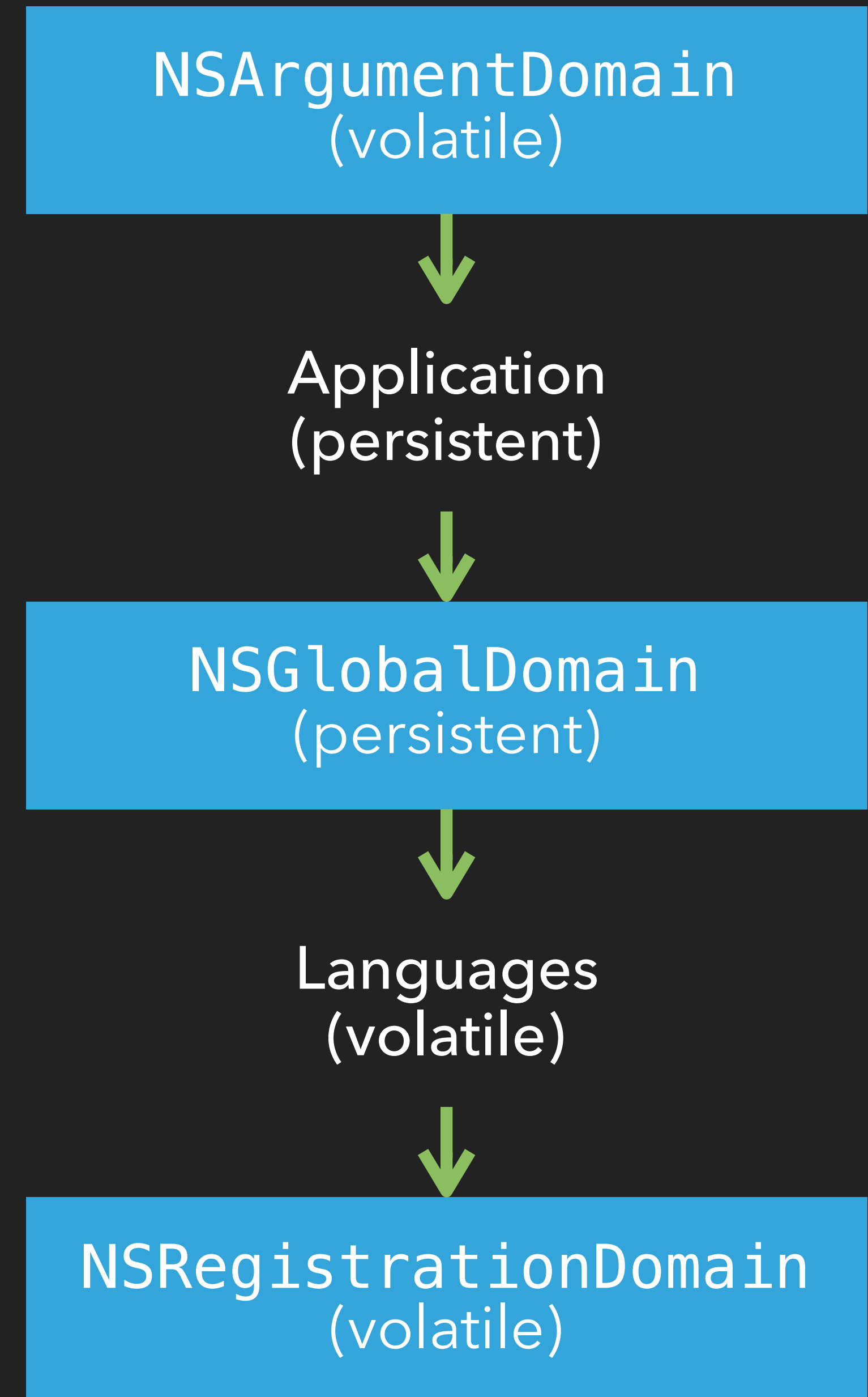
Gavin Wiggins

NSUserDefaults Class

- ▶ An interface to the user's defaults database (a fancy dictionary)
- ▶ Allows simple data types to persist between app launches or when a device restarts
- ▶ Best used for application settings and user preferences
- ▶ Stored object must be a property list consisting of **NSData**, **NSString**, **NSNumber**, **NSDate**, **NSArray**, or **NSDictionary**
- ▶ Sandboxed app cannot access or modify preferences of another app
- ▶ Levels of precedence for defaults known as domains

Domains

- ▶ User's defaults take precedence over factory defaults
- ▶ **NSArgumentDomain** write/read on the command line (see defaults tool)
- ▶ Application is the user's defaults database
- ▶ **NSGlobalDomain** set for entire system
- ▶ Language based on user's preferred language
- ▶ **NSRegistrationDomain** factory defaults registered at launch



Getting the Shared Instance of NSUserDefaults

```
class func standardUserDefaults() -> NSUserDefaults
```

- ▶ Returns the shared defaults object
- ▶ If object does not exist, it is created with a search through the domains

```
let userdefaults = NSUserDefaults.standardUserDefaults()
```

Registering Defaults

```
func registerDefaults(_ registrationDictionary: [String : AnyObject])
```

- ▶ Registers factory defaults for the application to the registration domain
- ▶ Contents of registration domain are not written to disk; you need to call this method each time your application starts

```
let dict:[String: AnyObject] = ["itemA": 1, "itemB": 2.89]
```

```
userDefaults.registerDefaults(dict)
```

Setting Default Values

- ▶ Set the value based on specified key

```
func setBool(_ value: Bool, forKey defaultName: String)
func setFloat(_ value: Float, forKey defaultName: String)
func setInteger(_ value: Int, forKey defaultName: String)
func setObject(_ value: AnyObject?, forKey defaultName: String)
func setDouble(_ value: Double, forKey defaultName: String)
func setURL(_ url: NSURL?, forKey defaultName: String)
```

Removing User Defaults

- ▶ Removes the contents of the specified persistent domain
- ▶ Application will return to using values from the factory defaults

```
userdefaults.removePersistentDomainForName(NSBundle.mainBundle().bundleIdentifier!)
```

Getting Default Values

- ▶ Return value (array, bool, integer, etc.) associated with the key
- ▶ User's previously stored default value or the factory value

```
func arrayForKey(_ defaultName: String) -> [AnyObject]?
func boolForKey(_ defaultName: String) -> Bool
func dataForKey(_ defaultName: String) -> NSData?
func dictionaryForKey(_ defaultName: String) -> [String : AnyObject]?
func floatForKey(_ defaultName: String) -> Float
func integerForKey(_ defaultName: String) -> Int
func objectForKey(_ defaultName: String) -> AnyObject?
func stringArrayForKey(_ defaultName: String) -> [String]?
func stringForKey(_ defaultName: String) -> String?
func doubleForKey(_ defaultName: String) -> Double
func URLForKey(_ defaultName: String) -> NSURL?
```


Defaults Command-Line Tool

- ▶ Read, write, delete user defaults from a command-line shell
- ▶ Do not edit defaults from command-line while application is running

```
$ defaults read com.apple.TextEdit
```

```
$ defaults read -app TextEdit
```

```
$ defaults read com.GavinWiggins.NSUserDefaults.plist
```

```
{
```

```
    checkboxState = 1;
```

```
    doubleNumber = "3.14568";
```

```
    floatNumber = "4.1";
```

```
    integerNumber = 12;
```

```
    stringText = "hello folks";
```

```
}
```

```
$ defaults write com.GavinWiggins.NSUserDefaults "checkboxState" 0
```

Resources

- ▶ Preferences and Settings Programming Guide
<https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/UserDefaults/Introduction/Introduction.html>
- ▶ UserDefaults Class Reference
https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSUserDefaults_Class/index.html
- ▶ UserDefaults article on Tinkker.io
<http://tinkker.io/swift/2016/02/27/nsuserdefaults.html>
- ▶ Handling Default Values with UserDefaults
<http://oleb.net/blog/2014/02/nsuserdefaults-handling-default-values/>