

Persistence

Why do we care?

Persistence menu

- File
- File (Data Base)
- File (Somewhere in net)

Dummy Application Life



State changing Data persistence

- Single Run
- Multiple Runs
- On single application installation
- Persist over delete/remove on app store
- Persistence binded to user



iOS File System Basics

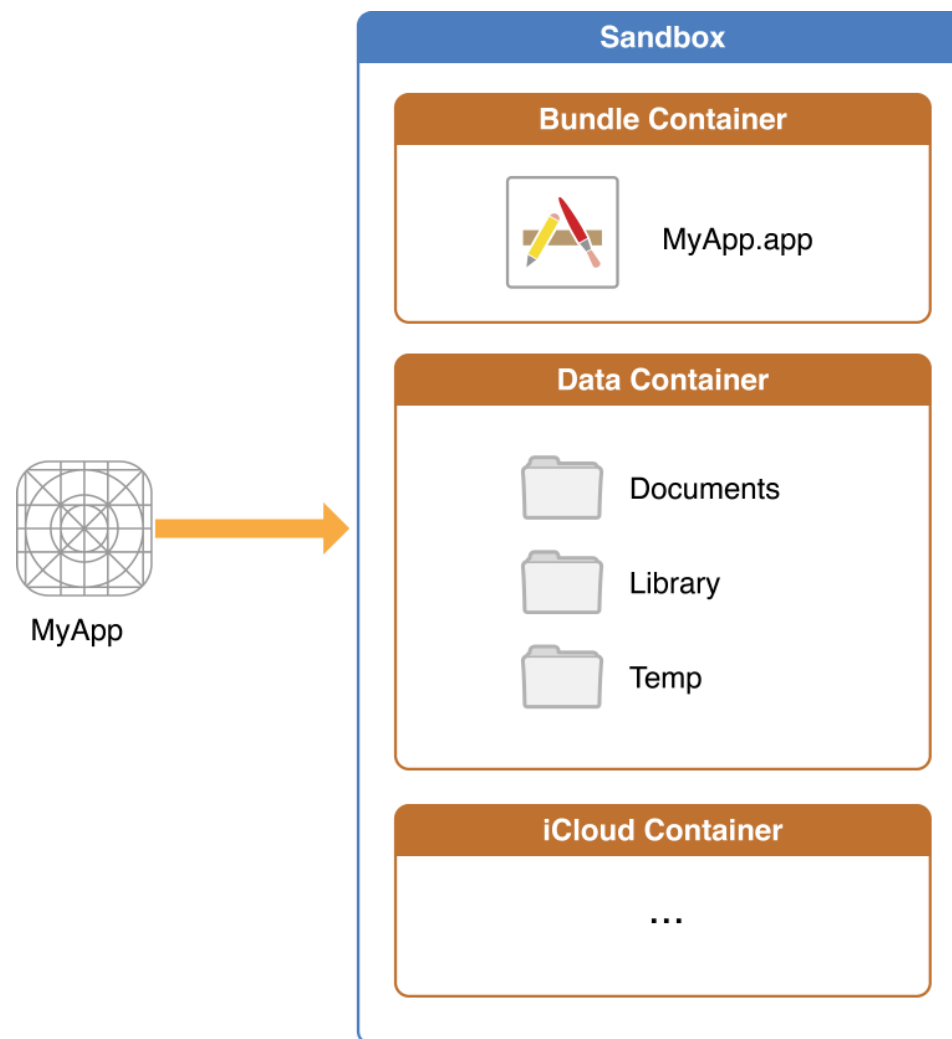
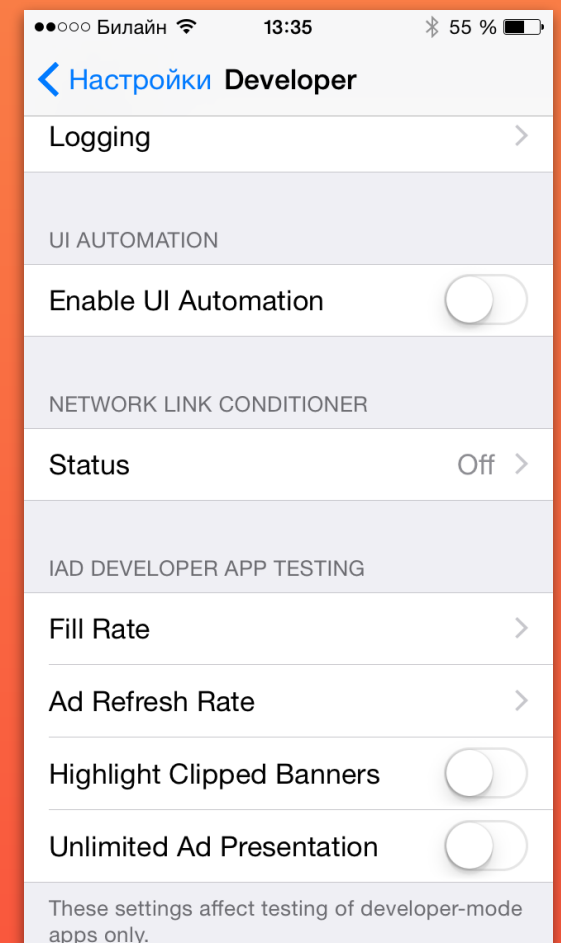


image source: developer.apple.com

UserDefaults

- Persist on single application installation
- UserDefaultsDidChangeNotification
- Key-Value storage
- User can edit some defaults in settings application



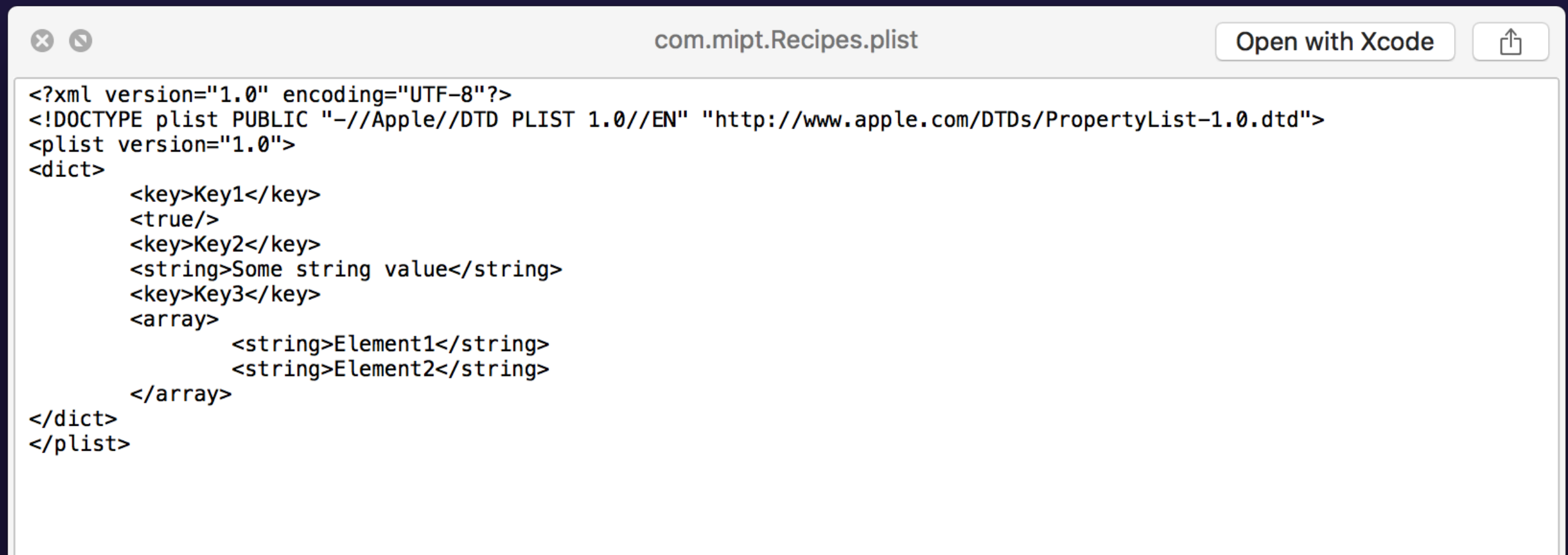
Access Defaults

```
// Access user defaults  
let userDefaults = UserDefaults.standard
```


Storing to Defaults

// Not every object can be saved

```
let defaults = UserDefaults.standard
defaults.set(true, forKey: "Key1")
defaults.set("Some string value", forKey: "Key2")
defaults.set(["Element1", "Element2"], forKey: "Key3")
```



The screenshot shows a text editor window titled "com.mipt.Recipes.plist". The window has standard macOS window controls (close, zoom, share) and a button that says "Open with Xcode". The text inside the editor is the XML representation of the defaults dictionary, which matches the Swift code shown above. The XML is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Key1</key>
  <true/>
  <key>Key2</key>
  <string>Some string value</string>
  <key>Key3</key>
  <array>
    <string>Element1</string>
    <string>Element2</string>
  </array>
</dict>
</plist>
```

Reading Defaults

```
let defaults = UserDefaults.standard
```

```
// returns Any?
```

```
defaults.object(forKey: "Key3")
```

```
// return String?
```

```
defaults.string(forKey: "Key2")
```

```
defaults.bool(forKey: "Key1")
```

File

- Can we do better?

NSCoding Protocol

```
public func encode(with aCoder: NSCoder)  
public init?(coder aDecoder: NSCoder)
```

Docs file path

```
// Access document directory
let documentDirectoryURL =
FileManager.default.urls(for: .documentDirectory,
in: .userDomainMask).first!
```