Multitasking in iOS 7

Tech Talk - January 2014

raywenderlich.com

Why use multitasking?

- **Users** can do multiple things at once (e.g. using an app while talking on the phone).
- **Devices** can do things in the background without bothering users.
- However, battery life is always a priority for Apple.

Multitasking in iOS 6

- Background Task Completion
- Background Audio
- Location Services
  - Region Monitoring
  - Significant Location Changes
  - Continuous Location Monitoring
- VoIP
- Newsstand
- Push Notifications

Multitasking Enhancements in iOS 7

- Fast App Switcher
- Background Fetch
- Remote Notifications
- Background Transfer Service

App Switcher in iOS 4-6

- Foreground app **shifts** upwards, reveals row of icons.
- Row of icons includes foreground app as well as all backgrounded apps.
- You can switch between apps by tapping on the icons.

Photos
Contacts
Settings
Game Center
Maps
Newsstand
Passbook
Safari
Passbook
Game Center
Contacts
Photos

App Switcher in iOS 7

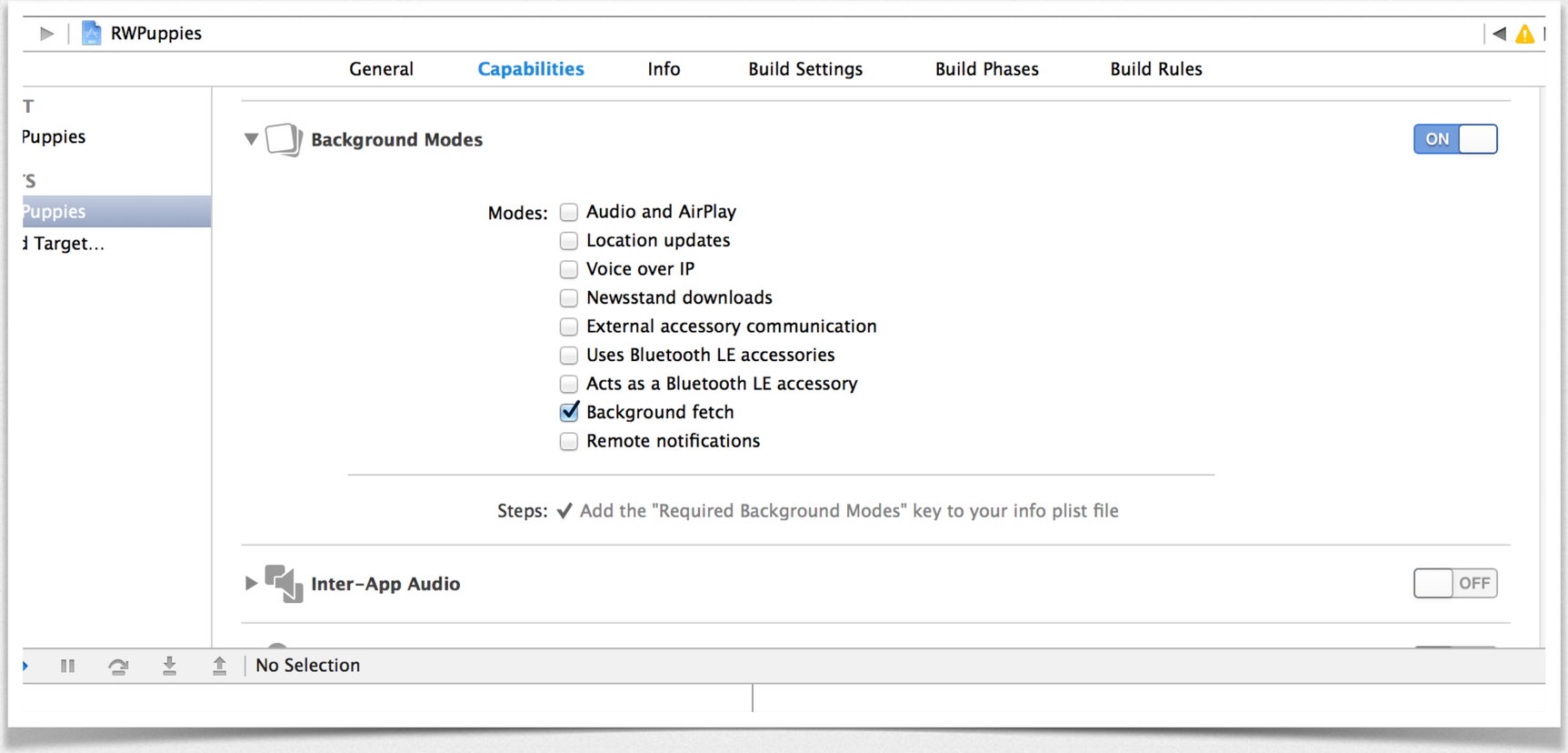
- Foreground app **zooms out** to reveal row of icons at the bottom and row of screenshots on top.
- You can switch between apps by tapping on the icons or the screenshots.
- Problem & opportunity: what should the screenshot be?
- New: flick screenshot upwards to terminate app, turn off multitasking.

Settings
General
Privacy
iCloud
Maps
Safari
Photos & Camera
Game Center
Twitter
Passbook
Photos
Settings
Passbook

Background Fetching (1)

- **Motivation:** The user should always see freshest content available. Also, app switcher screenshot should be as recent as possible.
- Possible applications: social networking feeds, news, entertainment, weather, finance, etc
- Background fetching is not guaranteed. Up to the OS.

Background Fetching (2)



Background Fetching (3)

```
- (void)application:(UIApplication *)application
  performFetchWithCompletionHandler:(void (^)(UIBackgroundFetchResult result))completionHandler;
```

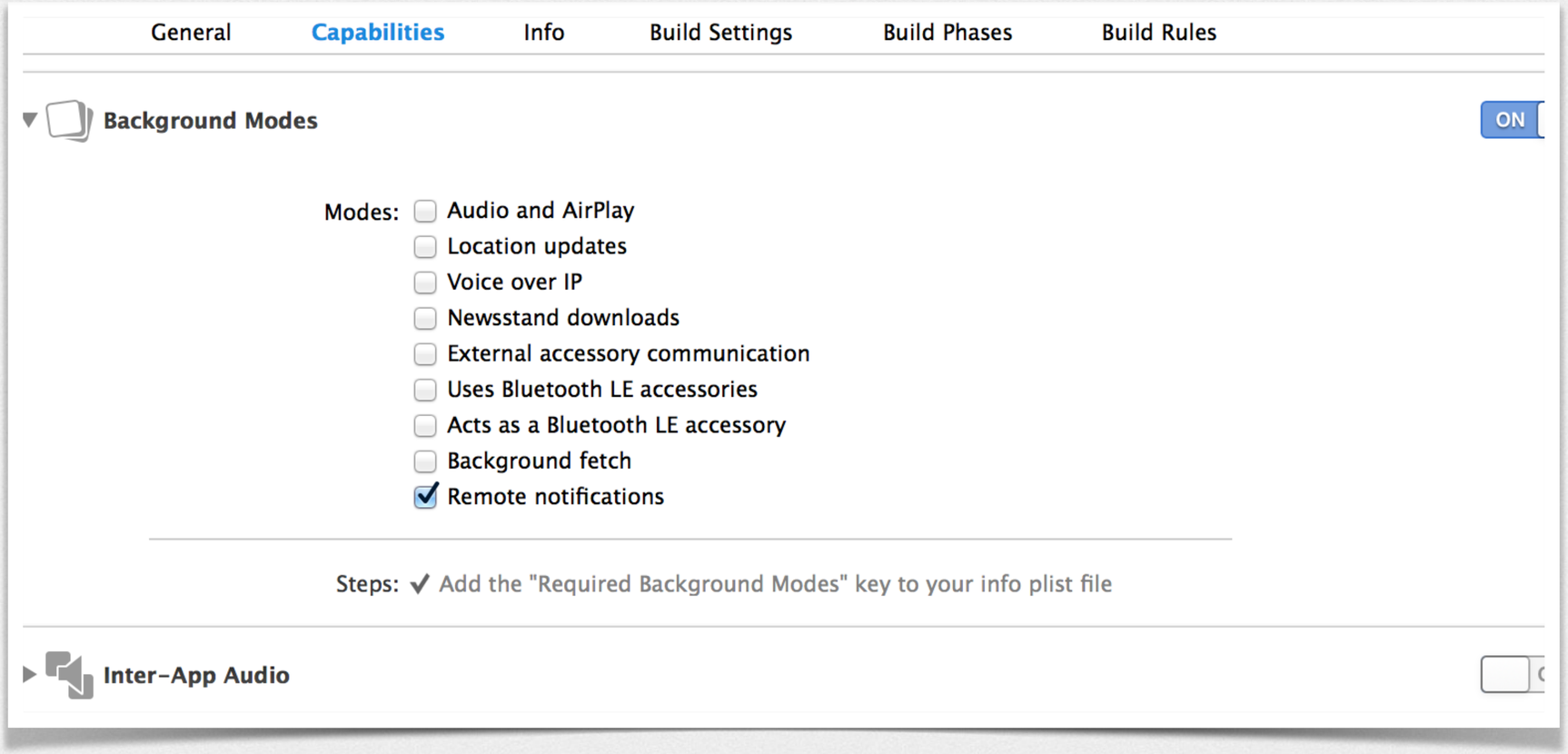
- Call completion handler when done.
- Report appropriate **UIBackgroundFetchResult**
  - UIBackgroundFetchResultNewData
  - UIBackgroundFetchResultNoData
  - UIBackgroundFetchResultFailed
- Set an appropriate minimum background fetch interval on UIApplication.

Background Fetching DEMO

Remote Notifications (1)

- Regular push notifications revamped in iOS 7.
- New in iOS 7: **Silent** push notifications.
- Possible applications for silent push notifications:
  - Syncing: file syncing, purchase syncing
  - Episodic content: new TV episodes, new podcast
- Silent push notifications are rate limited.

Remote Notifications (2)



Remote notifications (3)

Old push notification
iOS 6 and prior

```
aps {
  alert: {…}
}
```

```
- (void)application:
(UIApplication *)application
didReceiveRemoteNotification:
(NSDictionary *)userInfo;
```

New push notification
iOS 7

```
aps {
  content-available:1
  alert: {…}
}
```

```
- (void)application:
(UIApplication *)application
didReceiveRemoteNotification:
(NSDictionary *)userInfo
fetchCompletionHandler:(void (^)
(UIBackgroundFetchResult
result))handler;
```

Silent push notification
iOS 7

```
aps {
  content-available:1
}
```

```
- (void)application:
(UIApplication *)application
didReceiveRemoteNotification:
(NSDictionary *)userInfo
fetchCompletionHandler:(void (^)
(UIBackgroundFetchResult
result))handler;
```

Background Transfer Service (1)

- **Motivation:** Before iOS 7, large uploads or downloads often couldn't complete. Background task API didn't allow enough time.
- In iOS 7, create NSURLSession with a background configuration. Add download or upload tasks.
- Background uploads and downloads are up to the OS.

Background Transfer Service (2)

```
// Create background configuration
NSURLSessionConfiguration *configuration =
  [NSURLSessionConfiguration backgroundSessionConfiguration:@"foo"];

// Create NSURLSession
self.urlSession =
 [NSURLSession sessionWithConfiguration:configuration delegate:self delegateQueue:nil];

// Create off download task
NSURLRequest *request = [NSURLRequest requestWithURL:someURL];
self.downloadTask = [self.urlSession downloadTaskWithRequest:request];

// Resume to begin
[self.downloadTask resume];

/* Implement NSURLSessionDownloadTaskDelegate, NSURLSessionTaskDelegate, etc... */
```

Background Transfer Service (3)

```
#pragma mark - Background Transfer Service

- (void)application:(UIApplication *)application handleEventsForBackgroundURLSession:(NSString
*)identifier completionHandler:(void (^)())completionHandler {

    if ([identifier isEqualToString:@"foo"]) {

        // 1. Pass in completion handler as needed
        // 2. Refresh UI as needed
        // 3. Execute completion handler
    }
}
```

Potential Problems

- Data protection. Make sure sensitive data is handled appropriately when operating in the background.
- Privacy concerns. Be careful what you show in the app switcher.

One last thing…

- What does the future of multitasking look like in iOS?

Resources

- WWDC 2013 Session 204: What's New with Multitasking
- WWDC 2013 Session 705: What's New in Foundation Networking
- WWDC 2013 Session 714: Protecting your User's Privacy
- WWDC 2013 Session 709: Protecting Secrets with the Keychain
- WWDC 2013 Session 222: What's New in State Restoration

- iOS 7 By Tutorials: Ch 16, Ch 17, Ch 18
  http://www.raywenderlich.com/store/ios-7-by-tutorials