Optimizing Your App for Multitasking on iPad in iOS 9

Session 212

Brittany Paine iOS SpringBoard Engineer
Jon Drummond iOS SpringBoard Engineer
## Multitasking Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Multitasking Essentials for Media-Based Apps on iPad in iOS 9</td>
<td>Pacific Heights</td>
<td>Wednesday 2:30PM</td>
</tr>
<tr>
<td>Optimizing Your App for Multitasking in iOS</td>
<td>Presidio</td>
<td>Wednesday 3:30PM</td>
</tr>
</tbody>
</table>
## Multitasking Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Venue</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Multitasking Essentials for Media-Based Apps on iPad in iOS 9</td>
<td>Pacific Heights</td>
<td>Wednesday 2:30PM</td>
</tr>
<tr>
<td>Optimizing Your App for Multitasking in iOS</td>
<td>Presidio</td>
<td>Wednesday 3:30PM</td>
</tr>
</tbody>
</table>
CPU

1 frame
CPU

1 frame
CPU

1 frame
Memory

System

Primary App

Free
Memory

![Image of an iPad displaying the Memory app with options for System, Secondary App, and PiP.](image_url)
Memory
What Is SpringBoard?
What Is SpringBoard?
What Is SpringBoard?

UIApplication

UIApplication
SpringBoard
SpringBoard

The original multitasking app
SpringBoard

The original multitasking app

Faced the same multitasking challenges as you do now
SpringBoard

The original multitasking app
Faced the same multitasking challenges as you do now
Learned a lot of lessons along the way
Optimizing Your App
The easy stuff
Optimizing Your App

The easy stuff

Fix memory leaks
Optimizing Your App

The easy stuff

Fix memory leaks
Fix retain cycles and unbounded memory growth
Optimizing Your App

The easy stuff

Fix memory leaks
Fix retain cycles and unbounded memory growth
Fix inefficient algorithms
Great Performance Involves Tradeoffs

- Memory
- CPU
- Disk Space
- I/O
- GPU
Great Performance Involves Tradeoffs

- Memory
- CPU
- Disk Space
- I/O
- GPU
Great Performance Involves Tradeoffs

- Memory
- CPU
- Disk Space
- I/O
- GPU
Great Performance Involves Tradeoffs

- Memory
- CPU
- Disk Space
- I/O
- GPU
Digging In

- Memory
- CPU
- I/O
Digging In

CPU and disk I/O are minimal while scrolling
Digging In

CPU and disk I/O are minimal while scrolling
Digging In

CPU and disk I/O are minimal while scrolling

All icons kept in memory
Digging In

CPU and disk I/O are minimal while scrolling

All icons kept in memory
Digging In

CPU and disk I/O are minimal while scrolling

All icons kept in memory

System reclaims memory by terminating processes
Memory Management

Working set
Memory Management

Working set

The critical objects and resources you need right now
Memory Management

Working set

The critical objects and resources you need right now

• Keep it small
Memory Management

Working set

The critical objects and resources you need right now

• Keep it small
• It might change based on context
Memory Management

Working set

The critical objects and resources you need right now

- Keep it small
- It might change based on context
- Don’t let it grow unbounded
Working Set = Every Page
Working Set = Every Page
Working Set = Every Page
Digging In

- Memory
- CPU
- I/O
Digging In

CPU and disk I/O are high while scrolling
Digging In

CPU and disk I/O are high while scrolling
Digging In

CPU and disk I/O are high while scrolling

Very few icons in memory
Digging In

CPU and disk I/O are high while scrolling

Very few icons in memory
CPU Management

Main thread

Main thread’s top priority is responding to user events
Don’t do unnecessary work
## CPU Management

### Quality of service

<table>
<thead>
<tr>
<th></th>
<th>User Interactive*</th>
<th>User Initiated</th>
<th>Utility</th>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UI</strong></td>
<td>Is this work actively involved in updating the UI?</td>
<td>Is this work required to continue user interaction?</td>
<td>Is the user aware of the progress of this work?</td>
<td>Can this work be deferred to start at a better time?</td>
</tr>
<tr>
<td><strong>IN</strong></td>
<td><strong>UT</strong></td>
<td><strong>BG</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Main thread

---

**Building Responsive and Efficient Apps with GCD**  
Nob Hill  
Friday 10:00AM

**Power, Performance, and Diagnostics: What’s New in GCD and XPC**  
WWDC14
dispatch_queue_t iconQueue = dispatch_queue_create("IconGenerationQueue",
                                            DISPATCH_QUEUE_SERIAL);

for (IRIcon *icon in [iconPage icons]) {
    dispatch_async(iconQueue, ^{
        [self _iconQueue_generateImageForIcon:icon];
    });
}];
dispatch_queue_t iconQueue = dispatch_queue_create("IconGenerationQueue",
            DISPATCH_QUEUE_SERIAL);

for (IRIcon *icon in [iconPage icons]) {
    dispatch_async(iconQueue, ^{
        [self _iconQueue_generateImageForIcon:icon];
    });
}];
dispatch_queue_t iconQueue = dispatch_queue_create("IconGenerationQueue", DISPATCH_QUEUE_SERIAL);

for (IRIcon *icon in [iconPage icons]) {
    dispatch_async(iconQueue, ^{
        [self _iconQueue_generateImageForIcon:icon];
    });
}
CPU Management

QoS overrides

Lower priority queues can be temporarily boosted when a higher priority queue is waiting for it.

The system needs a hint that you want a QoS override to occur.
### QoS Overrides

#### Dispatch

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Overrides?</th>
</tr>
</thead>
<tbody>
<tr>
<td>dispatch_sync()</td>
<td>✔️</td>
</tr>
<tr>
<td>dispatch_block_wait()</td>
<td>✔️</td>
</tr>
<tr>
<td>dispatch_group_wait()</td>
<td>✗</td>
</tr>
<tr>
<td>dispatch_semaphore_wait()</td>
<td>✗</td>
</tr>
</tbody>
</table>

---

**Building Responsive and Efficient Apps with GCD**

Nob Hill  
Friday 10:00AM

**Power, Performance, and Diagnostics: What’s New in GCD and XPC**

WWDC14
dispatch_sync(iconQueue, ^{
    // ensure icon generation completes at caller QoS
});

UIImage *image = [icon image];
Working Set = 3 Pages
Working Set = 3 Pages
Working Set = 3 Pages
Working Set = 3 Pages

System
IconReel
Secondary App
Free
Managing Memory Growth

Memory warnings
Managing Memory Growth

Memory warnings

Happens when

• The system is under memory pressure
• Your process is approaching its memory limit
Managing Memory Growth

Memory warnings

Happens when

- The system is under memory pressure
- Your process is approaching its memory limit

What to do

- Remove anything not in the working set
  - Clear cached data
  - Release images
  - Release view controllers
Memory Warnings

APIs

In your UIApplicationDelegate and UIViewController
- `[UIApplicationDelegate applicationDidReceiveMemoryWarning:]`
- `[UIViewController didReceiveMemoryWarning]`

Anywhere in an application process

UIApplicationDidReceiveMemoryWarningNotification

In libraries and extensions

DISPATCH_SOURCE_TYPE_MEMORYPRESSURE
Managing Memory Growth

NSCache

NSDictionary-like
Ideal for objects that can be regenerated on demand
Trims under memory pressure
Trims for application lifecycle changes
Working Set = 3 Pages
Working Set = 3 Pages
Working Set = 3 Pages

System
IconReel
Secondary App
Free
Working Set = 3 Pages
Working Set = 3 Pages
Recap

Memory, CPU, I/O
Recap

- Memory
- CPU
- I/O

Working Set
Recap

Adaptive

NSCache

Working Set

Memory

CPU

I/O
Advanced Adaptive Memory
Leveraging the Virtual Memory System

Jon Drummond iOS SpringBoard Engineer
Primary App

System

Memory Warning

Secondary App
System

Secondary App

PiP
“The world outside your process should be regarded as hostile and bent upon your destruction.”

Me, almost everyday
Advanced Adaptive Memory
Advanced Adaptive Memory

Memory is the most constrained resource on iOS

• No swap
Advanced Adaptive Memory

Memory is the most constrained resource on iOS

- No swap

System can require memory faster than it can be released

- CPU contention during memory warnings can slow or prevent apps from releasing it
<table>
<thead>
<tr>
<th>Classification</th>
<th>Description</th>
<th>Reclaimable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirty Memory in active use</td>
<td>Otherwise-dirty memory designated as not in-use</td>
<td></td>
</tr>
<tr>
<td>Clean Memory</td>
<td>Read-only memory backed by files on disk</td>
<td></td>
</tr>
</tbody>
</table>
## Memory Classification

<table>
<thead>
<tr>
<th>Classification</th>
<th>Description</th>
<th>Reclaimable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirty</td>
<td>Memory in active use</td>
<td>✗</td>
</tr>
</tbody>
</table>
## Memory Classification

<table>
<thead>
<tr>
<th>Classification</th>
<th>Description</th>
<th>Reclaimable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirty</td>
<td>Memory in active use</td>
<td>✗</td>
</tr>
<tr>
<td>Purgeable</td>
<td>Otherwise-dirty memory designated as not in-use</td>
<td>✓</td>
</tr>
</tbody>
</table>
Memory Classification

<table>
<thead>
<tr>
<th>Classification</th>
<th>Description</th>
<th>Reclaimable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dirty</td>
<td>Memory in active use</td>
<td>✗</td>
</tr>
<tr>
<td>Purgeable</td>
<td>Otherwise-dirty memory designated as not in-use</td>
<td>✓</td>
</tr>
<tr>
<td>Clean</td>
<td>Read-only memory backed by files on disk</td>
<td>✓</td>
</tr>
</tbody>
</table>
Memory Classification

Clean or purgeable memory is reclaimable
Memory Classification

Leeway
Leveraging the Virtual Memory System

Goals

Minimize dirty memory usage
Maximize purgeable and clean memory usage
Minimizing Dirty Memory
Minimizing Dirty Memory

Use less of it!
Minimizing Dirty Memory

Use less of it!

Classify otherwise \textit{dirty} memory as \textit{purgeable}

- Can be automatically reclaimed when not in use
- Best for “nice to have” data that can be recomputed
IconReel

Adopting purgeable data

NSCache  Working Set  NSCache
IconReel

Adopting purgeable data

NSCache  Working Set  NSCache
IconReel

Adopting purgeable data
IconReel

Adopting purgeable data
IconReel

Adopting purgeable data
IconReel

Adopting purgeable data

NSCache

Working Set

NSCache

In Use/Dirty
IconReel

Adopting purgeable data

NSCache | Working Set | NSCache

Purgeable | In Use/Dirty | Purgeable
IconReel

Adopting purgeable data

NSCache

Working Set

NSCache

Purgeable

In Use/Dirty

Purgeable
IconReel

Adopting purgeable data
IconReel

Adopting purgeable data
IconReel

Adopting purgeable data
IconReel

Adopting purgeable data

NSCache

Working Set

NSCache

Purgeable

In Use/Dirty

Purgeable
System

Reclaimable

Primary App

Secondary App

PiP

!!
Memory Warning

System | Primary App | Secondary App | PiP

!!
Purgeable Data

API

NSPurgeableData

• `beginContentAccess` - memory now considered **dirty**
• `endContentAccess` - memory now considered **purgeable**
• `isContentDiscarded` - has the memory been reclaimed?
IconReel
System resources

Adaptive

NSCache
Working Set

Memory
CPU
I/O
IconReel
System resources

Adaptive

- Purgeable
- NSCache
- Working Set

Memory | CPU | I/O
IconReel

Data characteristics
IconReel

Data characteristics

✅ Absolutely essential
IconReel

Data characteristics

- Absolutely essential
- Expensive to generate
IconReel
Data characteristics

- Absolutely essential
- Expensive to generate
- Can be pre-computed
IconReel

Data characteristics

- Absolutely essential
- Expensive to generate
- Can be pre-computed
- Static once generated
IconReel

Data characteristics

- Absolutely essential
- Expensive to generate
- Can be pre-computed
- Static once generated
- Perfect candidate for caching to a file!
IconReel

System resources

- Working Set
- NSCache
- Purgeable

Memory | CPU | I/O | Disk Space
IconReel
System resources

- Memory
  - Working Set
  - NSCache
  - Purgeable

- CPU

- I/O

- Disk Space
Maximizing Clean Memory
Maximizing Clean Memory

Memory backed by a file on disk is considered **clean**

- Data in such a file can be “memory mapped”
Maximizing Clean Memory

Memory backed by a file on disk is considered **clean**

- Data in such a file can be “memory mapped”

Memory and file contents must match exactly

- Ideal for read-only data
Maximizing Clean Memory

Memory backed by a file on disk is considered clean

- Data in such a file can be “memory mapped”

Memory and file contents must match exactly

- Ideal for read-only data

Data in memory is evicted and reloaded on your behalf

- Allows for random access
Memory Mapped Data
Memory Mapped Data
Memory Mapped Data

VM

Memory Mapped Data

File on Disk
Memory Mapped Data
Memory Mapped Data
Memory Mapped Data
IconReel
IconReel
IconReel

Memory Mapped Image Data
IconReel
IconReel
IconReel
IconReel
System resources

NSCache/Purgeable
Working Set
Memory

CPU

I/O

Disk Space
typedef NS_OPTIONS(NSUInteger, NSDataReadingOptions) {
    NSDataReadingMappedIfSafe,
    NSDataReadingMappedAlways,
    ...
};

@interface NSData (NSDataCreation)
- (nullable instancetype)initWithContentsOfFile:(NSString *)path
    options:(NSDataReadingOptions)readOptionsMask
    error:(NSError **)errorPtr;
@end
Memory Mapped Data

API

typedef NS_OPTIONS(NSUInteger, NSDataReadingOptions) {
    NSDataReadingMappedIfSafe,
    NSDataReadingMappedAlways,
    ...}

@interface NSData (NSDataCreation)
-(nullable instancetype)initWithContentsOfFile:(NSString *)path
    options:(NSDataReadingOptions)readOptionsMask
    error:(NSError **)errorPtr;
@end
typedef NS_OPTIONS(NSUInteger, NSDataReadingOptions) {
    NSDataReadingMappedIfSafe,
    NSDataReadingMappedAlways,
    ...
};

@interface NSData (NSDataCreation)
- (nullable instancetype)initWithContentsOfFile:(NSString *)path
options:(NSDataReadingOptions)readOptionsMask
error:(NSError **)errorPtr;
@end
Leveraging the Virtual Memory System

Caveats
Leveraging the Virtual Memory System

Caveats

Not appropriate for small chunks of data

VM Addressable Space

0x00000000 0xFE000000
Leveraging the Virtual Memory System

Caveats

- Not appropriate for small chunks of data
- Virtual Memory misuse
  - Fragmentation
  - Exhaustion
Leveraging the Virtual Memory System

Caveats

Not appropriate for small chunks of data

Virtual Memory misuse

- Fragmentation
- Exhaustion

Abusing the VM system can result in process termination

VM Addressable Space

0x00000000 0xFFE00000
Summary

I just walked by your old cube and looked to see if you were in there. Not sure when I'll stop doing that. I just know that the guy who's sitting there now is probably starting to get annoyed with me. What can I say? We miss you. All of us. Especially me. Although I'm probably more productive now without you and your seemingly endless supply of office gossip.

Work is pretty much the same. I switched cubes and have gotten closer to an actual window. That's the good part. The bad part is that I'm directly across from Larry. You know, the guy with the terrible hair and the terrible taste in music. I actually caught myself singing along to the music coming from his office the other day. It definitely wasn't my proudest moment.

I've been traveling a bit more. Managed to parlay into a week of shows and ate nearly every one of the adventure. You'd have loved it.

I'm eager to hear what's going on. Advancement is worth not being a computer.

Sarah
Summary

I just walked by your old cube and looked to see if you were in there. Not sure when I’ll stop doing that. I just know that the guy who’s visiting there now is probably starting to get annoyed with me. What can I say? We miss you. All of us. Especially me, although I’m probably more productive now without you and your seemingly endless supply of office gossip.

Work is pretty much the same. I switched cubes and have gotten closer to an actual window. That’s the good part. The bad part is that I’m directly across from Larry. You know, the guy with the terrific hair and the terrible taste in music. I actually caught myself singing along to the music coming from his office the other day. It definitely wasn’t my proudest moment.

I’ve been traveling a bit more and managed to peek in at a few stores and was nearly every single time. You’d have to admit that the attention you’d have to get.

I’m eager to hear what’s going on. Advancement is worth it. Not bad.
Use Instruments to identify and fix bugs
Summary

Use Instruments to identify and fix bugs

Prioritize your work appropriately and don’t block the main thread
Summary

Use Instruments to identify and fix bugs
Prioritize your work appropriately and don’t block the main thread
Identify and manage your working set
Summary

Use Instruments to identify and fix bugs
Prioritize your work appropriately and don't block the main thread
Identify and manage your working set
Use caches and respond to memory warnings
Summary

Use Instruments to identify and fix bugs
Prioritize your work appropriately and don’t block the main thread
Identify and manage your working set
Use caches and respond to memory warnings
Leverage the virtual memory system to increase reclaimable memory
Summary

Use Instruments to identify and fix bugs

Prioritize your work appropriately and don’t block the main thread

Identify and manage your working set

Use caches and respond to memory warnings

Leverage the virtual memory system to increase reclaimable memory

Great performance requires tradeoffs
“It is not the strongest or the most intelligent who survive, but those who can best manage change.”

Charles Darwin (apocryphal)
More Information

Documentation
Adopting Multitasking Enhancements on iPad
Performance Overview
Instruments User Guide
http://developer.apple.com/iOS

Technical Support
Apple Developer Forums
http://developer.apple.com/forums

Developer Technical Support
http://developer.apple.com/support/technical

General Inquiries
Curt Rothert, App Frameworks Evangelist
rothert@apple.com
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Multitasking Essentials for Media-Based Apps on iPad in iOS 9</td>
<td>Pacific Heights</td>
<td>Wednesday 2:30PM</td>
</tr>
<tr>
<td>Performance on iOS and watchOS</td>
<td>Presidio</td>
<td>Friday 11:00AM</td>
</tr>
<tr>
<td>Building Responsive and Efficient Apps with GCD</td>
<td>Nob Hill</td>
<td>Friday 10:00AM</td>
</tr>
<tr>
<td>Labs</td>
<td>Frameworks Lab B</td>
<td>Wednesday 9:00AM</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>Cocoa Touch and Multitasking Lab</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Power and Performance Lab</strong></td>
<td>Frameworks Lab C</td>
<td>Friday 12:00PM</td>
</tr>
</tbody>
</table>