# iOS Printing System

## Dead Tree Output

Session 108

**Howard Miller**
Printing Engineering

**Andrew Platzer**
iOS Apps and Frameworks

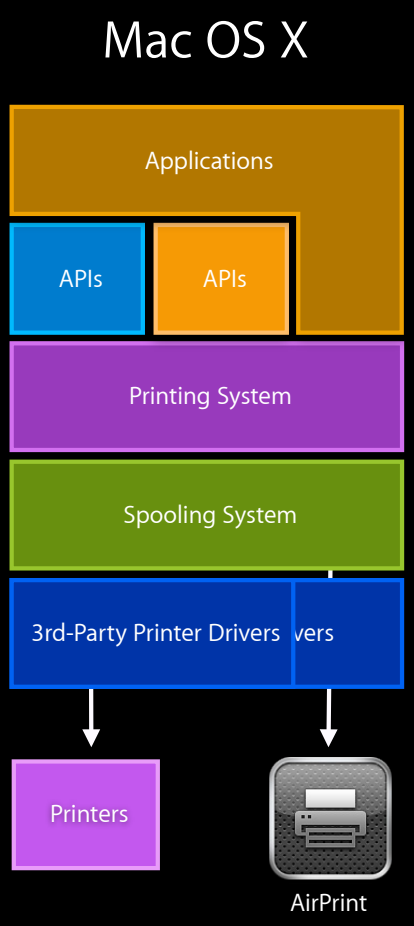
# Agenda

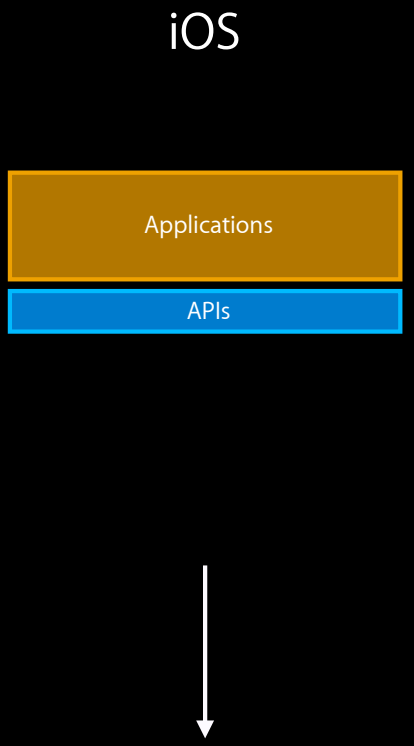
- Printing System overview
- iOS application printing paradigm
- Adopting iOS printing in your application
  - Basic printing
  - Use of Formatters and Renderers for more control while printing
  - Printing demonstration
  - Advanced printing features

# iOS Printing System

- Great user experience
  - Easy to use
  - Consistent high-quality output
  - No drivers or software to install, no configuration
- Great for application developers
  - Easy to add printing support to your application
  - Flexible and powerful printing system

# Apple Printing System Architecture



# AirPrint Printing Protocol

- Great user experience
  - No driver, no software to install
  - Full output quality
- Supported on all Apple platforms
  - iOS 4.2 and later
  - Mac OS X 10.7 Lion
- Standards based
  - Bonjour, IPP, PDF, JPEG, etc.
- Zero-cost license for printer manufacturers

# AirPrint 2010 Manufacturers

## Supported in popular late-2010 models

HP Envy e-All-in-One series (D410a)
HP Photosmart Plus e-AiO (B210a)
HP Photosmart Premium e-AiO (C310a)
HP Photosmart Premium Fax e-AiO (C410a)
HP Photosmart e-AiO (B110)
HP Photosmart e-AiO (D110)
HP Photosmart eStation (C510)
HP LaserJet Pro M1536dnf Multifunction Printer
HP LaserJet Pro CM1415fn Color Multifunction Printer
HP LaserJet Pro CM1415fnw Color Multifunction Printer
HP LaserJet Pro CP1525n Color Printer
HP LaserJet Pro CP1525nw Color Printer
HP Officejet 6500A e-AiO
HP Officejet 6500A Plus e-AiO
HP Officejet 7500A Wide Format e-AiO
HP Officejet Pro 8500A e-AiO
HP Officejet Pro 8500A Plus e-AiO
HP Officejet Pro 8500A Premium e-AiO

**Over 50% of the world's printers**

# AirPrint 2011 Manufacturers

## HP models since 2010

HP Color LaserJet Pro M1536dnf MFP
HP Color LaserJet Pro M1537dnf MFP
HP Color LaserJet Pro M1538dnf MFP
HP Color LaserJet Pro M1539dnf MFP
HP LaserJet P1102w
HP LaserJet P1606dn
HP LaserJet Pro CP
HP LaserJet Pro M1
HP LaserJet Pro M1
HP LaserJet Pro M1
HP LaserJet Pro M1
HP LaserJet Pro M1217nfw MFP

LEXMARK™

TOSHIBA

brother
At your side.

EPSON®

FUJI Xerox

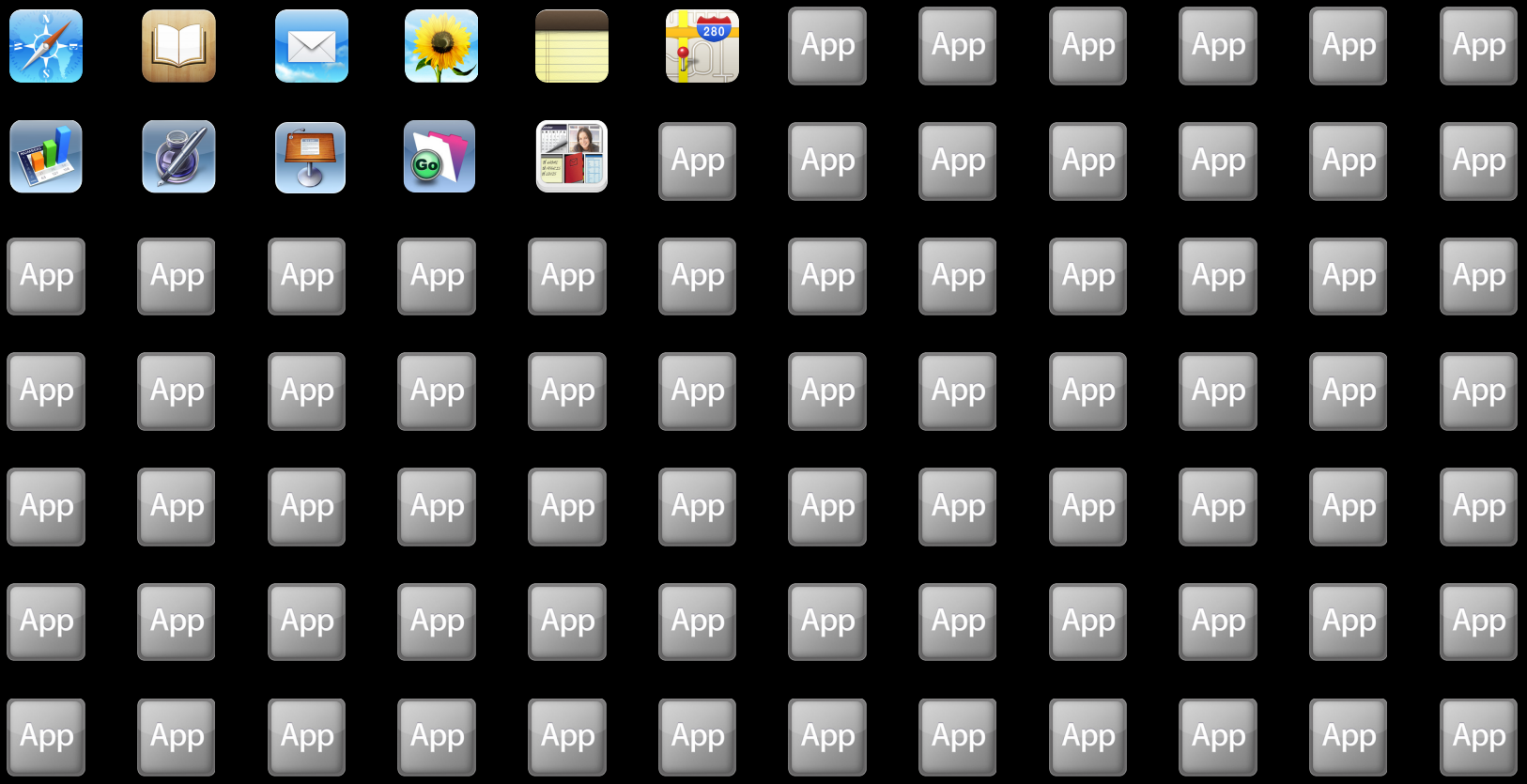
RICOH

SHARP®

SAMSUNG

# iOS Apps That Print

# iOS Apps That Print



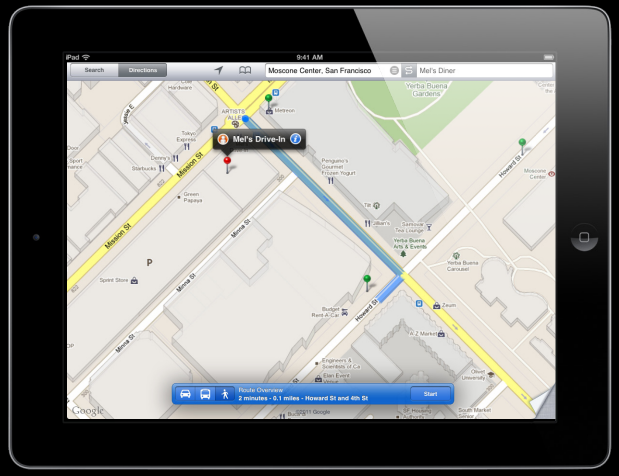

And 100s more

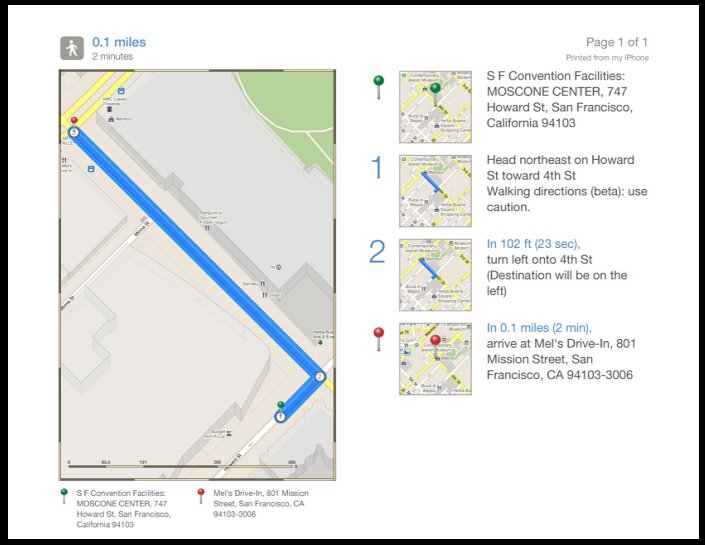
# iOS Printing Paradigm

- Dynamic
  - Printer picked at print time
  - Available paper sizes reported at print time
  - Layout when printing
    - Few document-centric applications
- Focused user options

# Route to a Great Printing Application

- Format for paper, not display
- Enhanced content
- High-quality drawing



# iOS Printing API

**Andrew Platzer**
iOS Apps and Frameworks

# iOS Printing API

- Simple API
- No Page Setup
- Printing is a modal task
- Not WYSIWYG

# What Can I Print?

- Items — PDF, JPEG, and other print-ready image types
- Formatters — plain and HTML text
- Renderers — for custom drawing

# Printing Classes

```
UIPrintInteractionController
   UIPrintInfo
   UIPrintPaper
UISimpleTextPrintFormatter
UIMarkupTextPrintFormatter
UIPrintPageRenderer
```

# Simple Printing

# Simple Printing

## Tiny example

- Get print controller
- Set item to print
- Present interface

# Simple Printing

## Tiny example

```
- (void)printData:(NSData *)data {

    UIPrintInteractionController *
           controller = [UIPrintInteractionController sharedPrintController];

    controller.printingItem = data;

    [controller presentAnimated:YES completionHandler:NULL];
}
```

# Simple Printing

## Full example

- Get print controller
- Set item to print
- Present interface

# Simple Printing

## Full example

- Get print controller
- Check data can be printed
- Set item to print
- Set output type
- Additional print settings
- Present interface

# Simple Printing

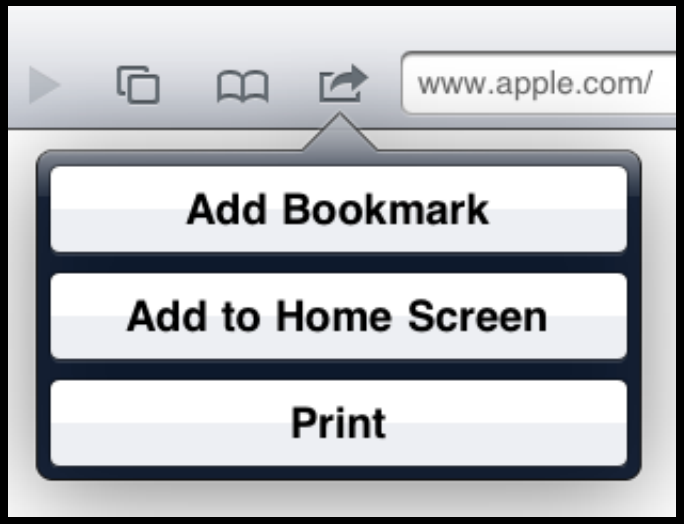
## Full example

```
- (void)printFile:(NSURL *)url {

    if ([UIPrintInteractionController canPrintURL:url]) {

        UIPrintInteractionController *
           controller = [UIPrintInteractionController sharedPrintController];

        controller.printingItem = url;

        UIPrintInfo *printInfo = [UIPrintInfo printInfo];
        printInfo.outputType = UIPrintInfoOutputGrayscale;
        printInfo.jobName    = [url lastPathComponent];
        controller.printInfo = printInfo;

        controller.showsPageRange = YES;

        [controller presentAnimated:YES completionHandler:NULL];
    }
}
```

# Simple Printing

## Showing the print action

- Check `+[UIPrintInteractionController isPrintingAvailable]`
- Hide Print button if NO



# Simple Printing

## Printable items

- Single item or array of items
  - PDF, JPEG, other image types (PNG, etc.)

    `NSURL, NSData`
  - Asset library

    `ALAsset, ALAssetURL`
  - Graphics objects

    `UIImage, CIImage`
- Each item is a separate job

# Simple Printing

## Checking the data

- Verify item

```
+[UIPrintInteractionController canPrintURL:]
+[UIPrintInteractionController canPrintData:]
```

- Item may not print if
  - Unknown format
  - Corrupt data
  - PDF is locked or printing disabled

# Simple Printing

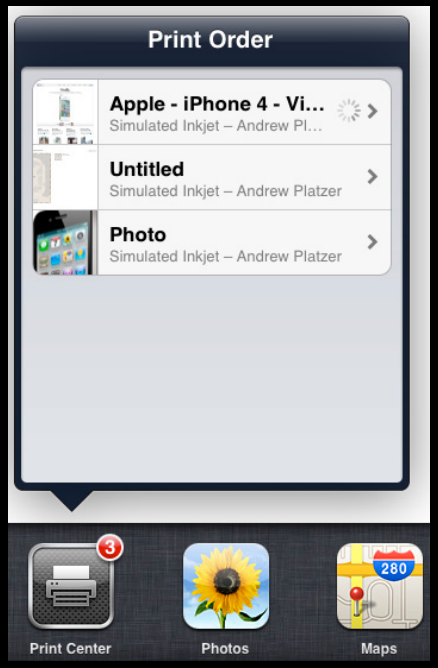
## Extra settings

`UIPrintInteractionController`

`showsPageRange`

`UIPrintInfo`

- Output type—photo or mixed
- Orientation—portrait or landscape
- Duplex—single or double-sided output
- Job name—displayed in Print Order



# Output Type

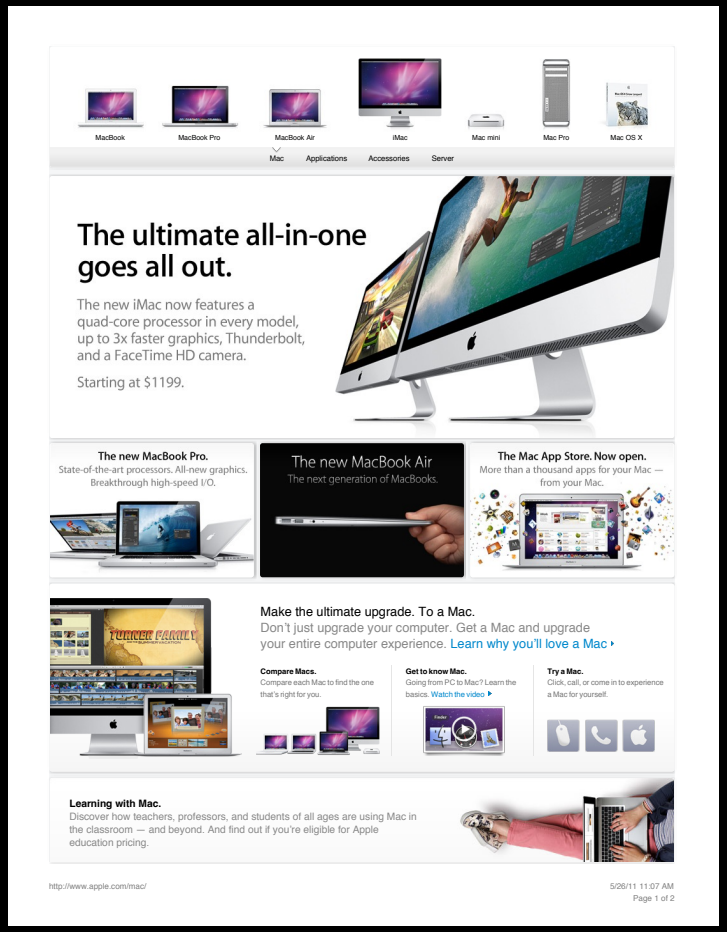
## UIPrintInfoOutputPhoto

- High quality
- Photo paper size
- No duplex mode
- No page range

# Output Type

## UIPrintInfoOutputGeneral

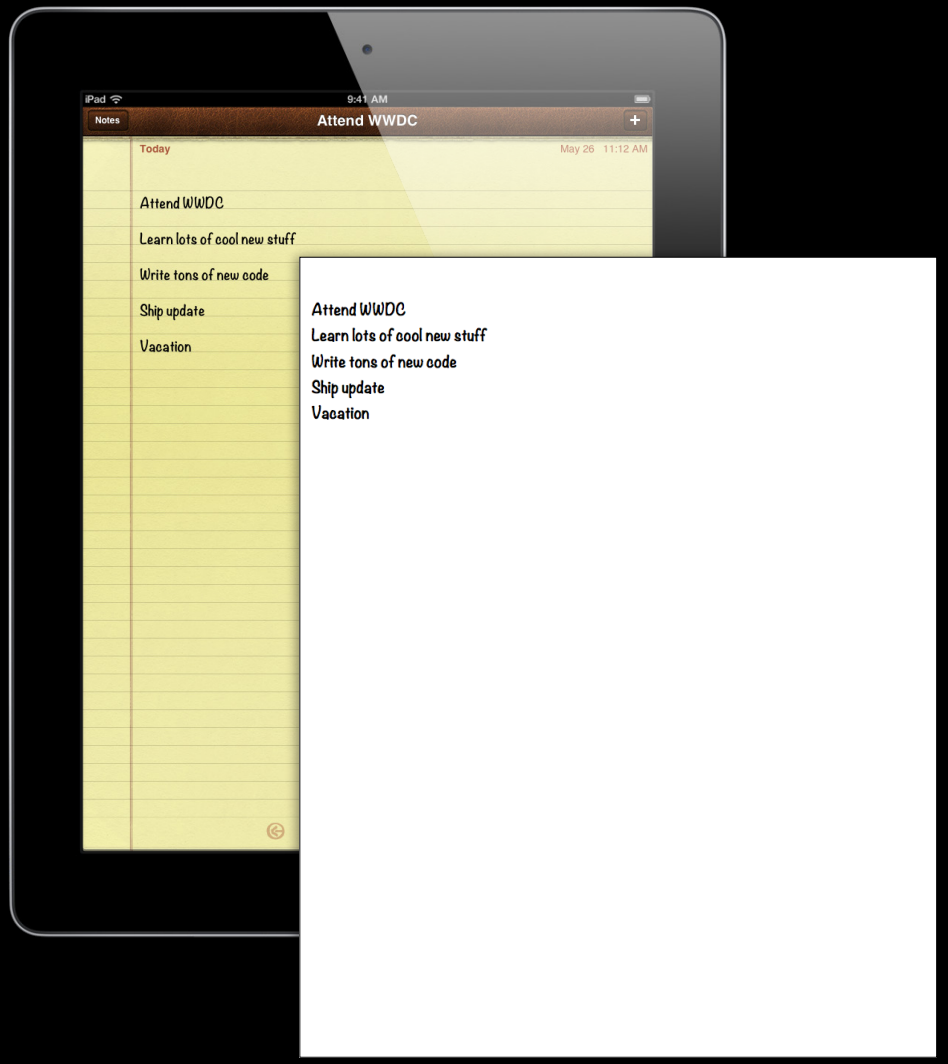
- Mixed text and graphics
- Normal quality
- Document paper size
- Duplex allowed
- Page range allowed



# Output Type

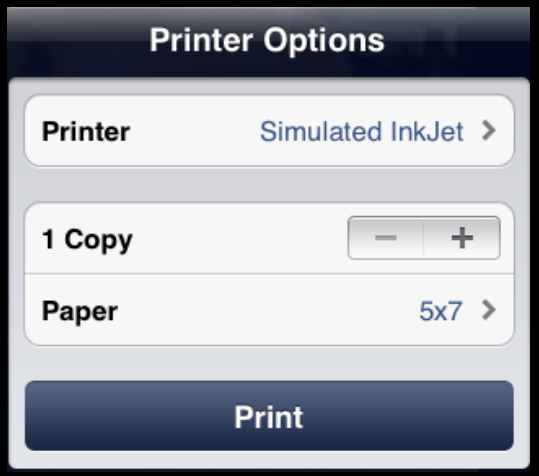
## UIPrintInfoOutputGrayscale

- Monochrome text and graphics
- Improved print speed
- Reduced ink usage
- Document paper size
- Duplex allowed
- Page range allowed



# Output Type

- Smart paper-size selection
- Appropriate print quality
- New paper-selection HI
- Improved user experience



# Simple Printing

## Summary

- PDF and image printing made easy
- Check printing is available
- Check your item is printable
- Set the output type

# Formatters

# Formatters

Use with `UIPrintInteractionController` or in a renderer

- Draw plain text and HTML text

`UISimpleTextFormatter` supports single

  - Font
  - Color
  - Alignment

`UIMarkupTextFormatter` supports HTML markup text

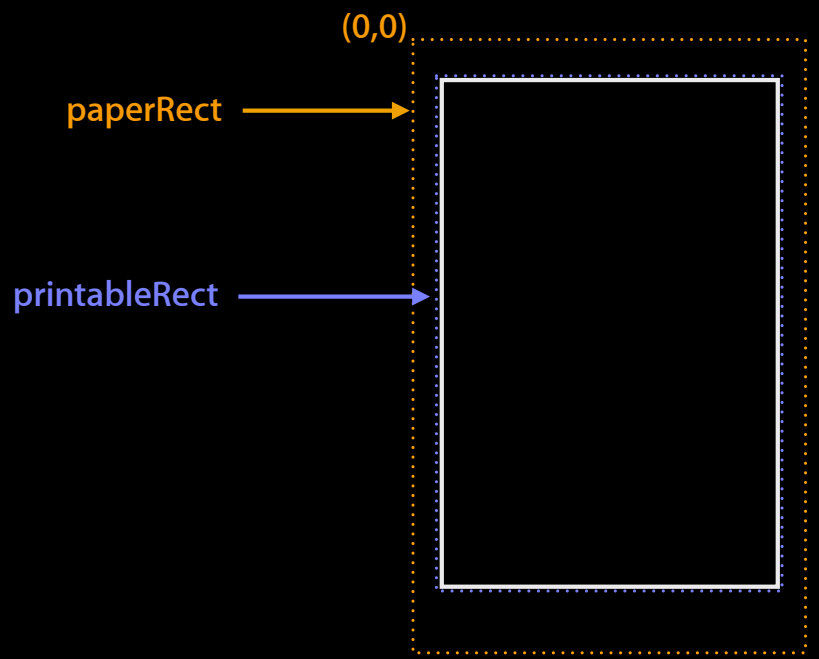
# Formatters

```
- (void)printHTMLText:(NSString *)text {

    UIPrintInteractionController *
           controller = [UIPrintInteractionController sharedPrintController];

    UIMarkupTextFormatter *formatter
                    = [[UIMarkupTextFormatter alloc] initWithText:text];

    controller.printFormatter = formatter;

    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo.outputType = UIPrintInfoOutputGrayscale;
    controller.printInfo = printInfo;

    [controller presentAnimated:YES completionHandler:NULL];
}
```

# Formatters

## Layout

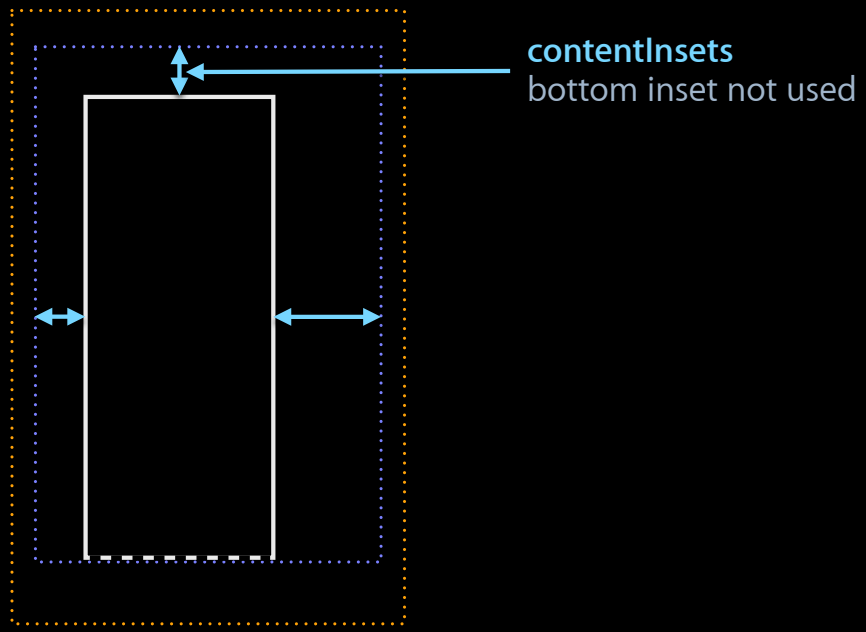
- Content drawn inside printable area



# Formatter Layout

## Layout

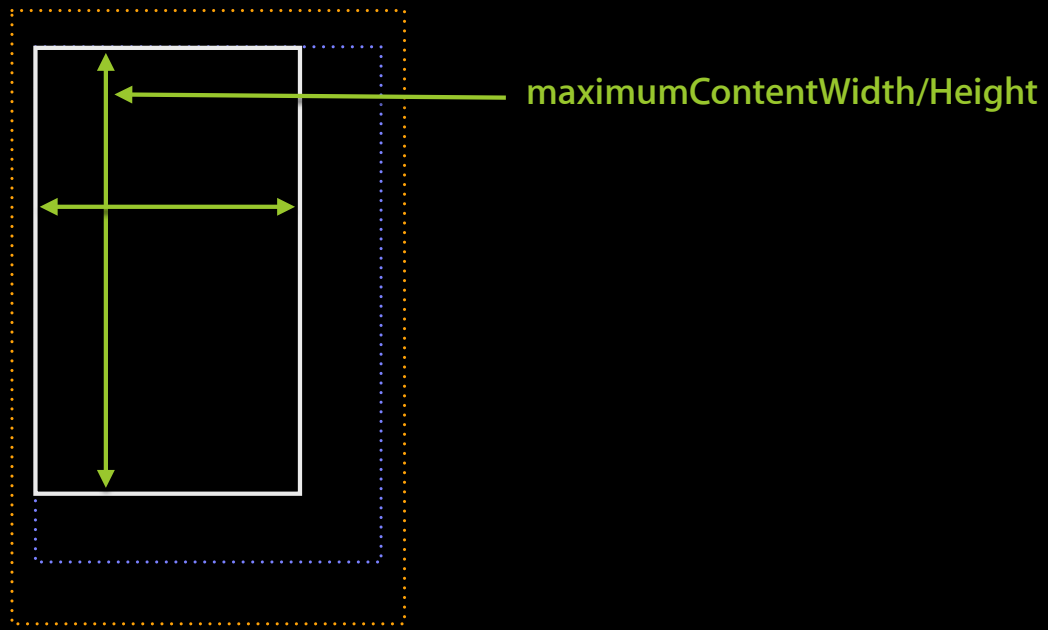
- Flexible insets



# Formatter Layout

## Layout

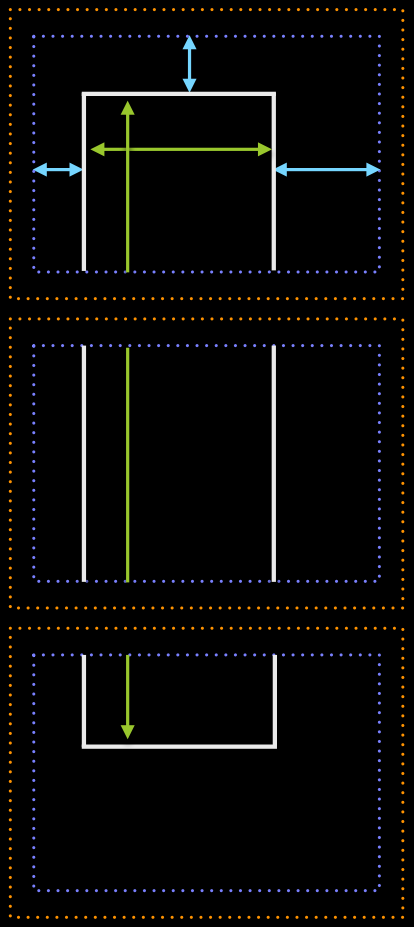
- Fixed width or height



# Formatters

## Layout

- Content can span multiple pages
- Breaks at printable rect boundaries
- Text formatters break between lines
- Mix insets and fixed width or height



# View Formatters

- Ask any view for its formatter

  `-[UIView viewPrintFormatter]`

  `UIWebView` and `UITextView` return text formatters

- Other UIViews return a generic view formatter

  `-drawRect:forViewPrintFormatter:`

  `-drawRect:`

# Renderers

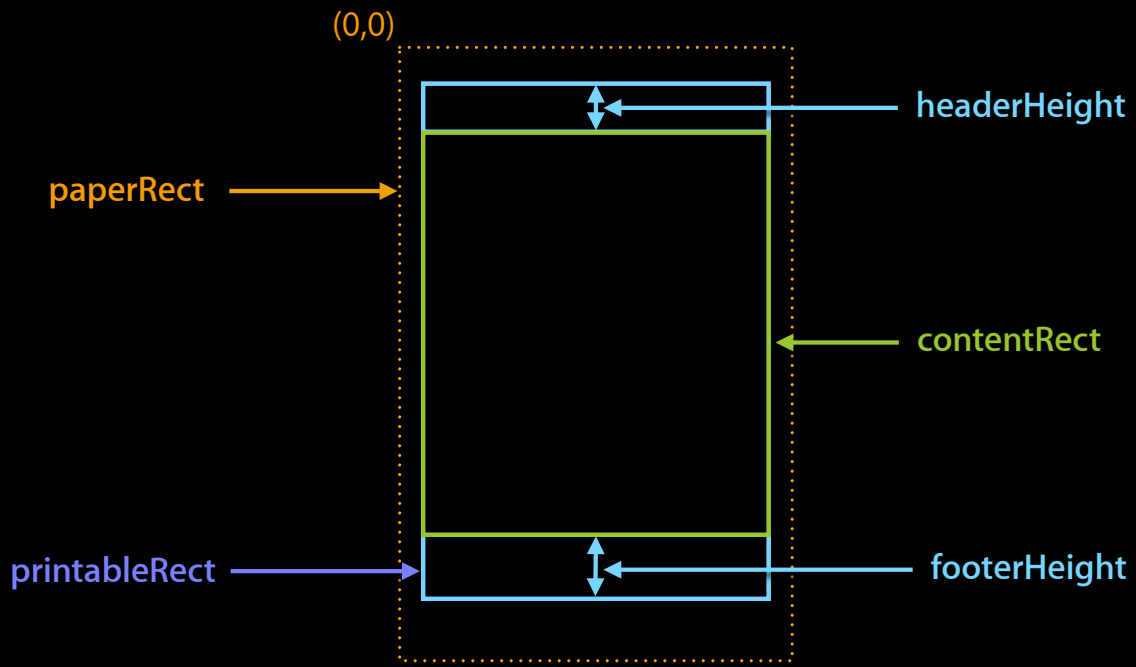
# Print Page Renderer

- Full drawing control
- Custom page-drawing object
  - Calculates page count
  - Draws page contents
- Add space for headers and footers
- Add formatters

# Basic Rendering

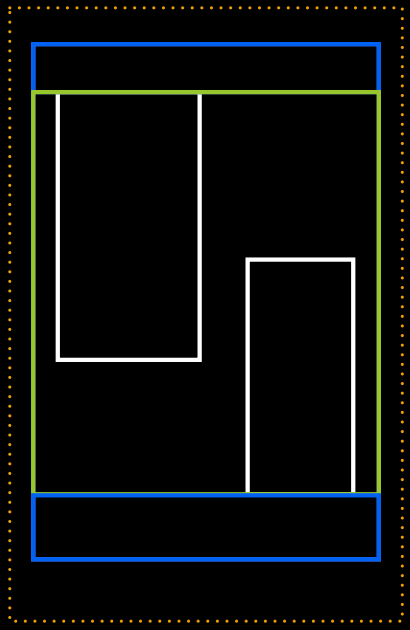
- Subclass `UPrintPageRenderer`
- Override
  - `-numberOfPages`
  - `-drawContentForPageAtIndex:inRect:`
- Set `UIPrintInteractionController.printPageRenderer`

# Renderer Layout



# Renderer Drawing Methods

```
-drawPageAtIndex:inRect:
    -drawHeaderForPageAtIndex:inRect:
    -drawContentForPageAtIndex:inRect:
    -drawPrintFormatter:forPageAtIndex:
    -drawFooterForPageAtIndex:inRect:
```



# Basic Rendering

```
@interface ItemRenderer : UIPrintPageRenderer { }

// New

- (void)drawItemAtIndex:(NSInteger)index
               atOffset:(CGPoint)offset;
@property NSInteger itemsPerPage;

// Overrides

- (NSInteger)numberOfPages;
- (void)drawContentForPageAtIndex:(NSInteger)pageIndex
                           inRect:(CGRect)contentRect
@end
```

# Basic Rendering

```
- (NSInteger)numberOfPages {

   self.itemsPerPage = floorf(self.printableRect.size.height / ITEM_HEIGHT);

   return ceilf(self.numberOfItems / self.itemsPerPage);
}

- (void)drawContentForPageAtIndex:(NSInteger)pageIndex
                         inRect:(CGRect)contentRect {

   UIRectClip(contentRect);

   for (int i = 0; i < self.itemsPerPage; i++) {
      [self drawItemAtIndex:pageIndex * self.itemsPerPage + i
                   atOffset:CGPointMake(contentRect.origin.x,
                                        contentRect.origin.y+i*ITEM_HEIGHT];
   }
}
```

# Rendering Formatters

- Add multiple formatters
- For each formatter
  - Set start page
  - Set offset on start page using top inset
- Automatically calculates page count

# Rendering Formatters

## Layout information

```
UIPrintFormatter.startPage
UIPrintFormatter.pageCount
-[UIPrintFormatter rectForPageAtIndex:]
```

# Rendering Formatters

```
@interface MarkupTextWithTitleRenderer : UIPrintPageRenderer { }

// New

- (id)initWithMarkupText:(NSString *)markupText
                   title:(NSString *)title;
@property(copy) NSString *title;

// Overrides

- (void)drawContentForPageAtIndex:(NSInteger)pageIndex
                           inRect:(CGRect)contentRect
@end
```

# Rendering Formatters

```
- (id)initWithMarkupText:(NSString *)markupText title:(NSString *)title {

   self = [super init];
   self.title = title;

   UIMarkupTextFormatter *formatter
                  = [[UIMarkupTextFormatter alloc] initWithText:markupText];
   formatter.contentInsets = UIEdgeInsetsMake(TITLE_HEIGHT, 0, 0, 0);

   [self addPrintFormatter:formatter startingAtPageAtIndex:0];

   return self;
}

- (void)drawContentForPageAtIndex:(NSInteger)pageIndex
                           inRect:(CGRect)contentRect {
   if (pageIndex == 0) {
      [self.title drawAtPoint:contentRect.origin withFont:TITLE_FONT];
   }
}
```

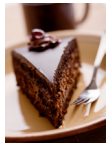
# Demo

## Adding printing to Recipes

**Dave DeLong**
iOS Apps and Frameworks

# Recipes


June 10, 2011 at 3:45 PM



## Chocolate Cake

Chocolate cake with chocolate frosting
Preparation Time: 1 hour

### Ingredients

- 1 cup Chocolate
- 1 cup Flour
- 2 Eggs
- 1 cup Sugar
- pinch Salt

### Instructions

Mix the ingredients, bake and voilà!

Page 1 of 1

# Printer Simulator

- Test without killing trees
- Simulated laser and inkjet printers
- Preview output
- Printer options

# Advanced Features

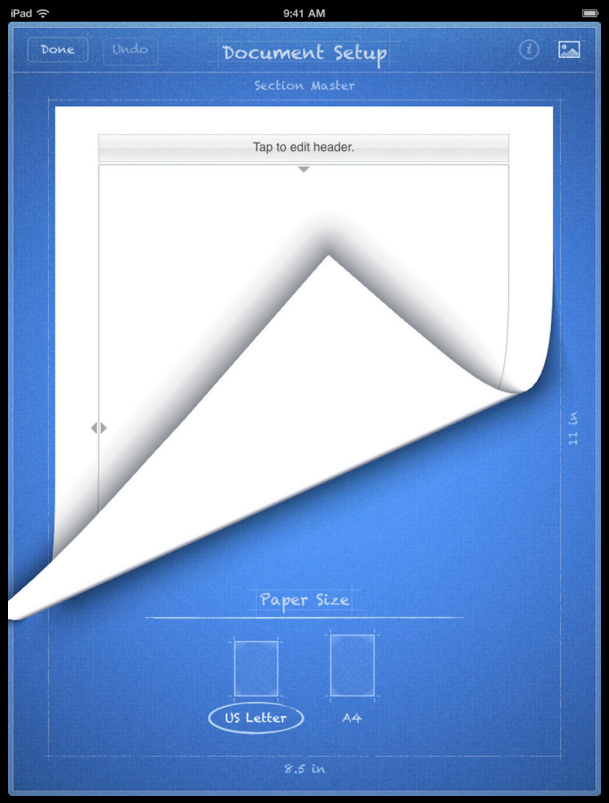
# Paper Size

- No Page Setup
- Be flexible about paper size
- Paper automatically chosen based on
  - Output type
  - Printer capabilities

# Paper Size

- If your document has a page view mode
  - Provide your own paper-selection interface
- Add delegate method

  `-printInteractionController:choosePaper:`
  - Called after user selects a printer
  - You provide the size
  - Match size to list of papers



# Paper Size

```
- (UIPrintPaper *)printInteractionController:(UIPrintInteractionController *)
                printInteractionController choosePaper:(NSArray *)paperList {

    CGSize paperSize = CGSizeMake(8.5 x 72.0, 11.0 x 72.0);

    return [UIPrintPaper bestPaperForPageSize:pageSize
                          withPapersFromArray:paperList];
}
```

# Embedded Printing HI

## Standard presentation

- iPhone

```
-presentAnimated:completionHandler:
```

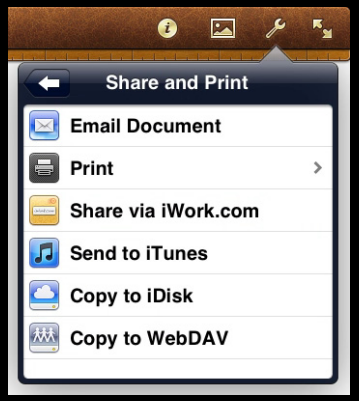
- iPad

```
-presentFromRect:inView:animated:completionHandler:
-presentFromBarButtonItem:animated:completionHandler:
```


# Embedded Printing HI

## Custom presentation

- Implement `-printInteractionControllerParentViewController:`
  - `UINavigationController` — push
  - `UIViewController` — modal
- Do not peek



# Summary

**Howard Miller**
Printing Engineering

# iOS Printing System

- Adding printing is simple
- Text layout made easy
- Custom drawing
- Enhances your application

# Developer Sample Code

**PrintWebView**
http://developer.apple.com/library/ios/#samplecode/PrintWebView

**PrintPhoto**
http://developer.apple.com/library/ios/#samplecode/PrintPhoto

# More Information

**Bill Dudney**
Application Frameworks Evangelist
dudney@apple.com

**iOS Printing Documentation**
http://developer.apple.com/search/index.php?q=printing

**Apple Developer Forums**
http://devforums.apple.com

# Related Sessions

| | |
|---|---|
| Practical Drawing for iOS Developers | Nob Hill<br>Thursday 4:30–5:30PM |

# Labs

| | |
|---|---|
| Printing on iOS and Mac OS X Lab | Application Frameworks Lab D<br>Tuesday 4:30–6:00PM |
| Cocoa Touch Lab | Application Frameworks Lab D<br>Wednesday 2:00–6:00PM |


WWDC2011