Adopting AirPrint

Session 718
Howard Miller
Printing Engineering
Agenda

AirPrint technologies update
iOS Printing
Printing without showing UI
Demo
AirPrint Technologies Update
What is AirPrint?
What is AirPrint?

Great user experience

• No driver, no software, no configuration
• Full output quality
What is AirPrint?

Great user experience
• No driver, no software, no configuration
• Full output quality

Protocol
• Standards based plus Apple technology
• Zero-cost license for printer and server manufacturers
What is AirPrint?
What is AirPrint?

Printers

• Ubiquitous—Supported by every major printer manufacturer
What is AirPrint?

Printers
• Ubiquitous—Supported by every major printer manufacturer

Printing System
• iOS 4.2 and later
What is AirPrint?

Printers
• Ubiquitous—Supported by every major printer manufacturer

Printing System
• iOS 4.2 and later

Your App
AirPrint Is Easy and Powerful

Simple user interaction
Efficient and focused APIs
High-quality output
AirPrint Advances

Expanded installed base
New classes of printers
Brother RuggedJet
AirPrint Advances
Continued

New APIs to support speciality printers and applications
Hints for Printing from Your App

Create your content for paper
• Use the space
• Provide higher quality artwork
• Consider readability

Work with the printing system
• Use just simplest APIs
• Do not introspect opaque objects
• Test with printer simulator
iOS Printing

Todd Ritland
AirPrint Engineer
iOS Printing Topics

Picking what to print
API overview
Printing UI options
iOS Printing Is Easy But Powerful
Providing Good Content for Printing

Useful, attractive, and high quality
What looks good on screen doesn’t always look good on paper
Make use of the dynamic printing system
• Paper size can be anything
• Printer hardware margins vary
• Best not to produce a fixed size PDF
Example

Math dictionary for kids
Example

Math dictionary for kids
**Laws of Arithmetic**

There are several laws that govern basic arithmetic operations.

- **Associative Law (Grouping Property)**
  Changing the way the addends or factors are grouped does not change the sum or product.

  **Examples:**
  - Addition: $a + (b + c) = (a + b) + c$
  - Multiplication: $a \times (b \times c) = (a \times b) \times c$

- **Commutative Law (Order Property)**
  Changing the order of the addends does not change the sum or product.

  **Examples:**
  - Addition: $a + b = b + a$
  - Multiplication: $a \times b = b \times a$

- **Distributive Law**
  A number that is multiplied by the sum of two or more numbers is the same as the sum of that number multiplied by each of the numbers separately.

  **Examples:**
  - $a \times (b + c) = (a \times b) + (a \times c)$
  - $2 \times (3 + 4) = (2 \times 3) + (2 \times 4)$

- **Zero Property of Addition**
  Any number plus zero equals that number.

  **Examples:**
  - $a + 0 = a$
  - $0 + a = a$

- **Zero Property of Multiplication**
  Any number multiplied by zero equals zero.

  **Examples:**
  - $a \times 0 = 0$
  - $0 \times a = 0$
API Overview
Basic Steps for Printing with UI

Your app will

• Get the print controller or activity controller
• Set up the attributes for the job
• Provide content to print
• Present the UI
Basic Steps for Printing with UI

Your app will
• Get the print controller or activity controller
• Set up the attributes for the job
• Provide content to print
• Present the UI

iOS will
• Communicate with AirPrint printer
• Daemon takes over and manages the job
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
   UISimpleTextPrintFormatter
   UIMarkupTextPrintFormatter
   UIViewPrintFormatter
UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
   UISimpleTextPrintFormatter
   UIMarkupTextPrintFormatter
   UIViewPrintFormatter
UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
UISimpleTextPrintFormatter
UIMarkupTextPrintFormatter
UIViewPrintFormatter
UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
iOS Printing Classes

UIPrintInfo
UIPrintPaper

UIPrintFormatter
  UISimpleTextPrintFormatter
  UIMarkupTextPrintFormatter
  UIViewPrintFormatter

UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
  UISimpleTextPrintFormatter
  UIMarkupTextPrintFormatter
  UIViewPrintFormatter
  UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
   UISimpleTextPrintFormatter
   UIMarkupTextPrintFormatter
   UIViewPrintFormatter
UIPrintPageRenderer
UIPrintInteractionController
UIActivityViewController
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
  UISimpleTextPrintFormatter
  UIMarkupTextPrintFormatter
  UIViewPrintFormatter
UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
UIPrintInfo
Setting the Job Name
Setting the Job Name

Now Printing...
WWDC-Apple Developer Guide
Output Type

Tell the printing system about the type of content to be printed
Allows the printing system to choose appropriate:
- Paper size
- Print quality mode
- Appropriate UI
Document

UIPrintInfoOutputGeneral

Mixed text and graphics
Normal quality
Document paper size
Duplex allowed
Page range allowed
Monochrome text and graphics
Improved print speed
Reduced ink usage
Document paper size
Duplex allowed
Page range allowed
Photo

UIPrintInfoOutputPhoto

High quality
Photo paper size
Borderless if available
No duplex mode
No page range
Photo High Quality Grayscale

UIPrintInfoOutputPhotoGrayscale

Photo paper size
Borderless if available
No duplex mode
No page range
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
UISimpleTextPrintFormatter
UIMarkupTextPrintFormatter
UIViewPrintFormatter
UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
   UISimpleTextPrintFormatter
   UIMarkupTextPrintFormatter
   UIViewPrintFormatter
UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
Providing Content
Printing items that already exist

Printing simple content

Fully custom drawn pages
Printing items that already exist

PDF files
Image files

Printing simple content

Fully custom drawn pages
Printing Items

Single item or array of items
- PDF, JPEG, other image types (PNG, etc.)
  - NSURL, NSData, UIImage, CIImage
- Asset library
  - ALAsset, ALAssetURL

Each item is a separate print job
Printing a PDF

- (void)printFile:(NSURL *)url {
    if ([UIPrintInteractionController canPrintURL:url]) {
        UIPrintInteractionController *controller = [UIPrintInteractionController sharedPrintController];
        controller.printingItem = url;
        UIPrintInfo *printInfo = [UIPrintInfo printInfo];
        printInfo.outputType = UIPrintInfoOutputGeneral;
        printInfo.jobName = [url lastPathComponent];
        controller.printInfo = printInfo;
        controller.showsPageRange = YES;
        [controller presentAnimated:YES completionHandler:NULL];
    }
}
Printing a PDF

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

  if ([UIPrintInteractionController canPrintURL:url]) {

    UIPrintInteractionController *controller = [UIPrintInteractionController sharedPrintController];

    controller.printingItem = url;

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

    controller.showsPageRange = YES;

    [controller presentAnimated:YES completionHandler:NULL];
  }
}
Printing a PDF

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

    if ([UIPrintInteractionController canPrintURL:url]) {

        UIPrintInteractionController *
        controller = [UIPrintInteractionController
            sharedPrintController];

        controller.printingItem = url;

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

        controller.showsPageRange = YES;

        [controller presentAnimated:YES completionHandler:NULL];
    }
}
Printing a PDF

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

    if ([UIPrintInteractionController canPrintURL:url]) {

        UIPrintInteractionController *
        controller = [UIPrintInteractionController
                      sharedPrintController];

        controller.printingItem = url;

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

        controller.showsPageRange = YES;

        [controller presentAnimated:YES completionHandler:NULL];
    }
}
Printing a PDF

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

    if ([UIPrintInteractionController canPrintURL:url]) {

        UIPrintInteractionController *
        controller = [UIPrintInteractionController
            sharedPrintController];

        controller.printingItem = url;

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

        controller.showsPageRange = YES;

        [controller presentAnimated:YES completionHandler:NULL];
    }
}
Printing a PDF

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

    if ([UIPrintInteractionController canPrintURL:url]) {

        UIPrintInteractionController *
        controller = [UIPrintInteractionController
                      sharedPrintController];

        controller.printingItem = url;

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

        controller.showsPageRange = YES;

        [controller presentAnimated:YES completionHandler:NULL];
    }
}
Printing a PDF

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

    if ([UIPrintInteractionController canPrintURL:url]) {

        UIPrintInteractionController *
            controller = [UIPrintInteractionController
                           sharedPrintController];

        controller.printingItem = url;

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

        controller.showsPageRange = YES;

        [controller presentAnimated:YES completionHandler:NULL];
    }
}
Printing a PDF

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

    if ([UIPrintInteractionController canPrintURL:url]) {

        UIPrintInteractionController *
        controller = [UIPrintInteractionController
                      sharedPrintController];

        controller.printingItem = url;

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

        controller.showsPageRange = YES;

        [controller presentAnimated:YES completionHandler:NULL];
    }
}
Printing items that already exist

- PDF files
- Image files

Printing simple content

Fully custom drawn pages
Printing items that already exist

---

Basic

Printing simple content

Text, HTML, etc.
Use one of the provided Formatter classes

---

Fully custom drawn pages
Formatters
What Is a Formatter?

101011100101010001010
11010010110110101101
101110010100101110110
What Is a Formatter?

Data

101011100101010001010
1101001011011010110
10110010100101110110

Rectangle
What Is a Formatter?

Data Rectangle

101011100101010001010
110100101101110101101
1011100101001011011010
What Is a Formatter?
What Is a Formatter?

Four score and seven years ago, our fathers brought forth on this continent…

String of text
What Is a Formatter?

Four score and seven years ago, our fathers brought forth on this continent…
What Is a Formatter?

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.
What Is a Formatter?

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate—we can not consecrate—we can not hallow—this ground. The brave men, living and dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task remaining before us—that from these honored dead we take increased devotion to that
Formatters

Use with `UIPrintInteractionController`, `UIActivityController` to format for the whole page

Use as a helper to format data in a full Renderer

For plain text use `UISimpleTextFormatter` and specify the
- Font
- Color
- Alignment

For HTML markup text use `UIMarkupTextFormatter`
Layout
Formatters

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

    UIPrintInteractionController *
    controller = [UIPrintInteractionController sharedPrintController];

    UIMarkupTextFormatter *formatter
        = [[UIMarkupTextFormatter alloc] initWithText:text];
    formatter.perPageContentInsets =
        UIEdgeInsetsMake(POINTS_PER_INCH * 0.75f, // 3/4 inch top margin
                         POINTS_PER_INCH * 0.75f, // 3/4 inch left margin
                         POINTS_PER_INCH * 0.75f, // 3/4 inch bottom margin
                         POINTS_PER_INCH * 0.75f); // 3/4 inch right margin
    controller.printFormatter = formatter;

    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo.outputType = UIPrintInfoOutputGeneral;
    printInfo.jobName = [urlField webPage];
    controller.printInfo = printInfo;

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

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

  UIPrintInteractionController *
  controller = [UIPrintInteractionController sharedPrintController];

  UIMarkupTextFormatter *formatter
    = [[UIMarkupTextFormatter alloc] initWithText:text];
  formatter.perPageContentInsets =
    UIEdgeInsetsMake(POINTS_PER_INCH * 0.75f, // 3/4 inch top margin
                     POINTS_PER_INCH * 0.75f, // 3/4 inch left margin
                     POINTS_PER_INCH * 0.75f, // 3/4 inch bottom margin
                     POINTS_PER_INCH * 0.75f); // 3/4 inch right margin
  controller.printFormatter = formatter;

  UIPrintInfo *printInfo = [UIPrintInfo printInfo];
  printInfo.outputType = UIPrintInfoOutputGeneral;
  printInfo.jobName    = [urlField webPage];
  controller.printInfo = printInfo;

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

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

    UIPrintInteractionController *
    controller = [UIPrintInteractionController sharedPrintController];

    UIMarkupTextFormatter *formatter
    ==[[UIMarkupTextFormatter alloc] initWithText:text];
    formatter.perPageContentInsets =
        UIEdgeInsetsMake(POINTS_PER_INCH * 0.75f, // 3/4 inch top margin
                          POINTS_PER_INCH * 0.75f, // 3/4 inch left margin
                          POINTS_PER_INCH * 0.75f, // 3/4 inch bottom margin
                          POINTS_PER_INCH * 0.75f); // 3/4 inch right margin

    controller.printFormatter = formatter;

    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo.outputType = UIPrintInfoOutputGeneral;
    printInfo.jobName    = [urlField webPage];
    controller.printInfo = printInfo;

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

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

    UIPrintInteractionController *
    controller = [UIPrintInteractionController sharedPrintController];

    UIMarkupTextFormatter *formatter
    = [[UIMarkupTextFormatter alloc] initWithText:text];
    formatter.perPageContentInsets =
        UIEdgeInsetsMake(POINTS_PER_INCH * 0.75f, // 3/4 inch top margin
                          POINTS_PER_INCH * 0.75f, // 3/4 inch left margin
                          POINTS_PER_INCH * 0.75f, // 3/4 inch bottom margin
                          POINTS_PER_INCH * 0.75f); // 3/4 inch right margin
    controller.printFormatter = formatter;

    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo.outputType = UIPrintInfoOutputGeneral;
    printInfo.jobName = [urlField webPage];
    controller.printInfo = printInfo;

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

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

    UIPrintInteractionController *
    controller = [UIPrintInteractionController sharedPrintController];

    UIMarkupTextFormatter *formatter
    = [[UIMarkupTextFormatter alloc] initWithText:text];
    formatter.perPageContentInsets =
    UIEdgeInsetsMake(POINTS_PER_INCH * 0.75f, // 3/4 inch top margin
                     POINTS_PER_INCH * 0.75f, // 3/4 inch left margin
                     POINTS_PER_INCH * 0.75f, // 3/4 inch bottom margin
                     POINTS_PER_INCH * 0.75f); // 3/4 inch right margin
    controller.printFormatter = formatter;

    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo.outputType = UIPrintInfoOutputGeneral;
    printInfo.jobName    = [urlField webPage];
    controller.printInfo = printInfo;

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

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

    UIPrintInteractionController *
    controller = [UIPrintInteractionController sharedPrintController];

    UIMarkupTextFormatter *formatter
        = [[UIMarkupTextFormatter alloc] initWithText:text];
    formatter/perPageContentInsets =
        UIEdgeInsetsMake(POINTS_PER_INCH * 0.75f, // 3/4 inch top margin
            POINTS_PER_INCH * 0.75f, // 3/4 inch left margin
            POINTS_PER_INCH * 0.75f, // 3/4 inch bottom margin
            POINTS_PER_INCH * 0.75f); // 3/4 inch right margin
    controller/printFormatter = formatter;

    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo/outputType = UIPrintInfoOutputGeneral;
    printInfo/jobName    = [urlField webPage];
    controller/printInfo = printInfo;

    [controller presentAnimated:YES completionHandler:NULL];
}
- (void)printMapView:(id) sender {

  UIPrintInteractionController *
  controller = [UIPrintInteractionController sharedPrintController];

  UIViewPrintFormatter *formatter
  = [self.myMapView viewPrintFormatter];

  controller.printFormatter = formatter;

  UIPrintInfo *printInfo = [UIPrintInfo printInfo];
  printInfo.outputType = UIPrintInfoOutputGeneral;
  printInfo.jobName = @“Map”;
  controller.printInfo = printInfo;

  [controller presentAnimated:YES completionHandler:NULL];
}
- (void)printMapView:(id)sender {
    UIPrintInteractionController *controller = [UIPrintInteractionController sharedPrintController];
    UIViewPrintFormatter *formatter = [self.myMapView viewPrintFormatter];
    controller.printFormatter = formatter;
    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo.outputType = UIPrintInfoOutputGeneral;
    printInfo.jobName = @"Map";
    controller.printInfo = printInfo;
    [controller presentAnimated:YES completionHandler:NULL];
}
View Formatters

- (void)printMapView:(id) sender {

    UIPrintInteractionController *
    controller = [UIPrintInteractionController sharedPrintController];

    UIViewPrintFormatter *formatter
    = [self.myMapView viewPrintFormatter];

    controller.printFormatter = formatter;

    UIPrintInfo *printInfo = [UIPrintInfo printInfo];
    printInfo.outputType = UIPrintInfoOutputGeneral;
    printInfo.jobName = @“Map”;
    controller.printInfo = printInfo;

    [controller presentAnimated:YES completionHandler:NULL];

}
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
   UISimpleTextPrintFormatter
   UIMarkupTextPrintFormatter
   UIViewPrintFormatter
UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
iOS Printing Classes

UIPrintInfo
UIPrintPaper
UIPrintFormatter
  UISimpleTextPrintFormatter
  UIMarkupTextPrintFormatter
  UIViewPrintFormatter
UIPrintPageRenderer

UIPrintInteractionController
UIActivityViewController
Printing items that already exist

Printing simple content

Text, HTML, etc.
Use one of the provided Formatter classes

Fully custom drawn pages
Printing items that already exist

Printing simple content

Advanced

Fully custom drawn pages

Render the pages from the content in your app
Use Formatter and Renderer classes
Renderers
What Is a Renderer?
What Is a Renderer?
What Is a Renderer?

Renderer

-numberOfPages
What Is a Renderer?

Renderer

- `--numberOfPages` → 2
What Is a Renderer?

**Renderer**
- `numberOfPages`: 2
- `drawContentForPageAtIndex:inRect:`

Draw First Page
What Is a Renderer?

Renderer

- `numberOfPages` → 2
- `drawContentForPageAtIndex:inRect:`
What Is a Renderer?

Renderer

- `number_of_pages` → 2

- `draw_content_for_pageAtIndex:inRect:`
What Is a Renderer?

Renderer

- numberOfPages → 2
- drawContentForPageAtIndex: inRect:
Basic Rendering

Subclass `UIPrintPageRenderer`

Override

- `numberOfPages`
- `drawContentForPageAtIndex:inRect:`

Set `UIPrintInteractionController.printPageRenderer`

or add to the array of activity items when creating the `UIActivityViewController` object

Can add a formatter object by calling `addPrintFormatter:startingAtPageIndex:`
Renderer with Formatter
Renderer with Formatter

Renderer

UISimpleTextFormatter

Draw Page
Four score and seven years ago, our fathers brought forth on this continent…
Four score and seven years ago, our fathers brought forth on this continent…
Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal. Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might
Showing the UI
Printing from the Share Sheet
Printing from the Share Sheet

NSArray *activityItems = @ [printInfo, myRenderer, [urlField text]];

UIActivityViewController *viewController = [[UIActivityViewController alloc]initWithActivityItems:activityItems applicationActivities:nil];

... present using standard view controller present methods
Printing from the Share Sheet

```objective-c
NSArray *activityItems = @ [ printInfo, myRenderer, [urlField text] ];

UIActivityViewController *viewController = [[UIActivityViewController alloc]
initWithActivityItems:activityItems applicationActivities:nil];

... present using standard view controller present methods
```
Printing from the Share Sheet

NSArray *activityItems = @ [ printInfo, myRenderer, [urlField text] ];

UIActivityViewController *viewController = [[UIActivityViewController alloc] initiWithActivityItems:activityItems applicationActivities:nil];

... present using standard view controller present methods
Printing from the Share Sheet

```objective-c
NSArray *activityItems = @ [ printInfo, myRenderer, [urlField text] ];

UIActivityViewController *viewController = [[UIActivityViewController alloc]
initWithActivityItems:activityItems applicationActivities:nil];

... present using standard view controller present methods
```
Printing Using a “Print” Button

Create and setup the `UIPrintInteractionController`

Standard presentation

- `presentAnimated:completionHandler:`

Popover presentation

- `presentFromRect:inView:animated:completionHandler:`
- `presentFromBarButtonItem:animated:completionHandler:`
Printing as a Menu Item

Set your class as the delegate for the shared `UIPrintInteractionController`

Implement `printInteractionControllerParentViewController:`

When the user taps “Print” call `presentAnimated:CompletionHandler:`

- UINavigationController — push
- UIViewController — modal
Printing as a Menu Item

Set your class as the delegate for the shared `UIPrintInteractionController`

Implement `printInteractionControllerParentViewController:`

When the user taps “Print” call `presentAnimated:completionHandler:`

- `UINavigationController`—push
- `UIViewController`—modal
Paper Size

Provide your own paper selection UI

Use delegate method

```-printInteractionController:choosePaper:`

- Called after user selects a printer
- You ask for a paper size that is a good match to the user-selected paper
- If a printer has paper sensors, the array will only be detected papers
Paper Size

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

  CGSize pageSize = CGSizeMake(POINTS_PER_INCH * 8.5f, POINTS_PER_INCH * 11.0f);

  return [UIPrintPaper bestPaperForPageSize:pageSize
      withPapersFromArray:paperList];
}
Paper Size

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

CGSize pageSize = CGSizeMake(POINTS_PER_INCH * 8.5f, POINTS_PER_INCH * 11.0f);

return [UIPrintPaper bestPaperForPageSize:pageSize
    withPapersFromArray:paperList];
}
Paper Size

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

  CGSize pageSize = CGSizeMake(POINTS_PER_INCH * 8.5f, POINTS_PER_INCH * 11.0f);

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

Roll Paper

Use delegate method

- `printInteractionController:cutLengthForPaper:`
  - Called after user selects a printer that has a roll loaded
  - `UIPrintPaper` width of the roll and the maximum height

By default, the cut length will be proportional to the default paper
Printing Without Showing UI
Submitting Jobs with No UI
Submitting Jobs with No UI

Opens up opportunity for new types of printing applications
Submitting Jobs with No UI

- Opens up opportunity for new types of printing applications
- Not for apps to provide custom print panel
Steps of Printing

Printer Options

Printer
Select Printer

1 Copy
[−] [+]
Steps of Printing

Printer Options

Printer
Select Printer

1 Copy
[+] [-]

Print
Steps of Printing

Printer Options

Printer
Select Printer

1 Copy

Print
Steps of Printing

Printer Options

- Printer: Select Printer
- 1 Copy: - +

Print
Steps of Printing

Printer Options

Printer
Select Printer

1 Copy

Print
Steps of Printing

Printer Options

Printer
Select Printer

1 Copy
- +

Print
Printer Picker

Save Original to Simulator
Todd Ritland's MacBook Pro

Simulated Color Laser
Todd Ritland's MacBook Pro

Simulated InkJet
Todd Ritland's MacBook Pro

Simulated Label Printer
Todd Ritland's MacBook Pro

Simulated Laser
Todd Ritland's MacBook Pro

Simulated 2-Sided InkJet
Todd Ritland's MacBook Pro

Simulated 36" Roll Printer
Todd Ritland's MacBook Pro
App Controlled Settings

Printer Picker

- Save Original to Simulator
  Todd Ritland's MacBook Pro
- Simulated Color Laser
  Todd Ritland's MacBook Pro
- Simulated InkJet
  Todd Ritland's MacBook Pro
- Simulated Label Printer
  Todd Ritland's MacBook Pro
- Simulated Laser
  Todd Ritland's MacBook Pro
- Simulated 2-Sided InkJet
  Todd Ritland's MacBook Pro
- Simulated 36" Roll Printer
  Todd Ritland's MacBook Pro

API
Basic Steps for Printing Without UI

Your app will include a setting for setting a printer
• Get the printer selector
• Present the UI
• Save the printer chosen by user

To print your app will
• Get the print controller
• Set up the attributes for the job
• Provide content to print
• Send the job with the new print controller method
Basic Steps for Printing Without UI

iOS will
- Communicate with AirPrint printer
- Daemon takes over and manages the job
iOS Background Printing Classes

UIPrinterPickerController
UIPrinter
UIPrintInteractionController
iOS Background Printing Classes

UIPrinterPickerController

UIPrinter

UIPrintInteractionController
iOS Background Printing Classes

UIPrinterPickerController

UIPrinter

UIPrintInteractionController
iOS Background Printing Classes

UIPrinterPickerController
UIPrinter

UIPrintInteractionController
UIPrinterPickerController

Your app must use this to choose the printer for printing
Same presentation options as `UIPrintInteractionController`
Present as a popover, sheet, or embedded in a navigation controller
Your app will be responsible for saving the URL
Allows filtering out printers
Create and setup the `UIPrinterPickerController`

- **Standard presentation**
  - `-presentAnimated:completionHandler:`

- **Popover presentation**
  - `-presentFromRect:inView:animated:completionHandler:`
  - `-presentFromBarButtonItem:animated:completionHandler:`

In completion handler, check if user selected a printer, then save printer selected
Printing Without UI

When ready to print, use the \texttt{UIPrintInteractionController} and call

\texttt{-printToPrinter:completionHandler:}

The \texttt{UIPrinter} passed in can be the object obtained by \texttt{UIPrinterPickerController}, or it can be a \texttt{UIPrinter} object instantiated with the URL
iOS Classes for Printing Without UI

UIPrinterPickerController
UIPrinter

UIPrintInteractionController
iOS Classes for Printing Without UI

UIPrinterPickerController
UIPrinter

UIPrintInteractionController
UIPrinter Methods and Properties

-contactPrinter:

-URL

-displayName
-displayLocation
-supportedJobTypes
-makeAndModel
-supportsColor
-supportsDuplex
UIPrinter Methods and Properties

- contactPrinter:
  - URL
  - displayName
  - displayLocation
  - supportedJobTypes
  - makeAndModel
  - supportsColor
  - supportsDuplex
UIPrinter Methods and Properties

- contactPrinter:
  - URL
- displayName
- displayLocation
- supportedJobTypes
- makeAndModel
- supportsColor
- supportsDuplex
UIPrinter Methods and Properties

- `contactPrinter`:

- `URL`

- `displayName`
- `displayLocation`
- `supportedJobTypes`
- `makeAndModel`
- `supportsColor`
- `supportsDuplex`

Recent Printers
- **Dad's Printer**
  - Upstairs Office
UIPrinter Methods and Properties

- contactPrinter:
- URL
- displayName
- displayLocation
- supportedJobTypes
- makeAndModel
- supportsColor
- supportsDuplex
UIPrinter Methods and Properties

- `contactPrinter`
- `URL`
- `displayName`
- `displayLocation`
- `supportedJobTypes`
- `makeAndModel`
- `supportsColor`
- `supportsDuplex`
UIPrinter Methods and Properties

- contactPrinter:
- URL
- displayName
- displayLocation
- supportedJobTypes
- makeAndModel
- supportsColor
- supportsDuplex
UIPrinter Methods and Properties

- contactPrinter:

- URL

- displayName
- displayLocation
- supportedJobTypes
- makeAndModel
- supportsColor
- supportsDuplex
Using UIPrinter for Your UI

NSURL *savedPrinterURL = [[NSUserDefaults standardUserDefaults] URLForKey:@"savedPrinter"];

if (savedPrinterURL) {
    self.savedPrinter = [UIPrinter printerWithURL:savedPrinterURL];

    [self.savedPrinter contactPrinter:^BOOL available) {
        if (available) {
            self.printerNameLabel.text = self.savedPrinter.displayName;
            self.printerLocationLabel.text = self.savedPrinter.displayLocation;
        } else {
            self.printerConnectionGoneIndicator.hidden = NO;
        }
    }];
}
Using UIPrinter for Your UI

```swift
NSURL *savedPrinterURL = [[NSUserDefaults standardUserDefaults]
    URLForKey:@"savedPrinter"];

if (savedPrinterURL) {
    self.savedPrinter = [UIPrinter printerWithURL:savedPrinterURL];

    [self.savedPrinter contactPrinter:^ (BOOL available) {
        if (available) {
            self.printerNameLabel.text = self.savedPrinter.displayName;
            self.printerLocationLabel.text = self.savedPrinter.displayLocation;
        }
        else {
            self.printerConnectionGoneIndicator.hidden = NO;
        }
    }];
}
```
Using UIPrinter for Your UI

```swift
NSURL *savedPrinterURL = [[NSUserDefaults standardUserDefaults]
                          URLForKey:@"savedPrinter"];

if (savedPrinterURL) {
    self.savedPrinter = [UIPrinter printerWithURL:savedPrinterURL];

    [self.savedPrinter contactPrinter:^BOOL available) {
        if (available) {
            self.printerNameLabel.text = self.savedPrinter.displayName;
            self.printerLocationLabel.text = self.savedPrinter.displayLocation;
        } else {
            self.printerConnectionGoneIndicator.hidden = NO;
        }
    }];
}
```
NSURL *savedPrinterURL = [[NSUserDefaults standardUserDefaults] URLForKey:@“savedPrinter”];

if (savedPrinterURL) {
    self.savedPrinter = [UIPrinter printerWithURL:savedPrinterURL];

    [self.savedPrinter contactPrinter:^BOOL available) {
        if (available) {
            self.printerNameLabel.text = self.savedPrinter.displayName;
            self.printerLocationLabel.text = self.savedPrinter.displayLocation;
        } else {
            self.printerConnectionGoneIndicator.hidden = NO;
        }
    }];
}
Using UIPrinter for Your UI

```objective-c
NSURL *savedPrinterURL = [[NSUserDefaults standardUserDefaults] URLForKey:@"savedPrinter"];

if (savedPrinterURL) {
    self.savedPrinter = [UIPrinter printerWithURL:savedPrinterURL];

    [self.savedPrinter contactPrinter:^ (BOOL available) {
        if (available) {
            self.printerNameLabel.text = self.savedPrinter.displayName;
            self.printerLocationLabel.text = self.savedPrinter.displayLocation;
        } else {
            self.printerConnectionGoneIndicator.hidden = NO;
        }
    }];
}
```
NSURL *savedPrinterURL = [[NSUserDefaults standardUserDefaults] 
    URLForKey:@“savedPrinter”];

if (savedPrinterURL) {
    self.savedPrinter = [UIPrinter printerWithURL:savedPrinterURL];

    [self.savedPrinter contactPrinter:^(_Bool available) {
        if (available) {
            self.printerNameLabel.text = self.savedPrinter.displayName;
            self.printerLocationLabel.text = self.savedPrinter.displayLocation;
        }
    }];
}
else {
    self.printerConnectionGoneIndicator.hidden = NO;
}
Using UIPrinter for Your UI

```objective-c
NSURL *savedPrinterURL = [[NSUserDefaults standardUserDefaults]
    URLForKey:@“savedPrinter”];

if (savedPrinterURL) {
    self.savedPrinter = [UIPrinter printerWithURL:savedPrinterURL];

    [self.savedPrinter contactPrinter:^(BOOL available) {
        if (available) {
            self.printerNameLabel.text = self.savedPrinter.displayName;
            self.printerLocationLabel.text = self.savedPrinter.displayLocation;
        } else {
            self.printerConnectionGoneIndicator.hidden = NO;
        }
    }];
}
```
For apps designed for specific printers
Apps can control which printers are shown
Any property of the printer can be used to filter.
When presenting `UIPrintPickerController`, use delegate method
  `shouldShowPrinter`:
Called for every printer that is discovered
New Class of Printing Applications
UI-less Printing Demo

Claudia Roberts
Printing System Engineer
Printing with No UI

Opens up door for new class of printing applications

- Endless possibilities
- Application can save multiple printers
Printing with No UI

Case Study—cash register App

• Unnecessary interaction with print dialog in current printing paradigm
• New API affords the developer the ability to enhance and simplify the user printing experience
• Easy to refactor code to adopt UI-less printing
Case Study—Cash Register App
Caffe Latte  $3.65
Blueberry Scone  $2.99
Vanilla Yogurt  $1.42
Subtotal  $8.06
Tax  $0.00
Total  $8.06
Demo

Charles Duyk
Printing System Engineer
Summary

Adding printing in your app is easy
Printing items such as PDFs or images is just 10 lines of code
Formatters and Renderers give you full control over printing
Apps can print without showing UI every time starting in iOS 8
More Information

Paul Danbold
Core OS Technologies Evangelist
danbold@apple.com

iOS Printing Documentation and Sample Code
http://developer.apple.com/airprint

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

AirPrint Basics
http://support.apple.com/kb/ht4356
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>What’s New in Cocoa Touch</td>
<td>Presidio</td>
<td>Tuesday 10:15 AM</td>
</tr>
<tr>
<td>Labs</td>
<td>Core OS Lab A</td>
<td>Friday 10:15AM</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>AirPrint Lab</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>