iOSドキュメントインタラクションプログラミングトピックス

2010-11-15

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
U.S.A.

アップルジャパン株式会社
〒163-1450 東京都新宿区西新宿
3 丁目20 番2 号
東京オペラシティタワー
http://www.apple.com/jp/

目次

リスト

ドキュメントインタラクションについて

iOSは、アプリケーションが直接的にはサポートしていないファイルのプレビューを表示するテクノロジーを提供しています。また、ファイルタイプの関連付けに関するシステムレベルのレジストリも提供しています。アプリケーションはこのレジストリを使って、インストールされているほかのアプリケーションのファイルを開く処理ができます。このような強力なテクノロジーとして、UIKitフレームワークの`UIDocumentInteractionController`クラス（『*UIDocumentInteractionController Class Reference*』を参照）とQuick Lookフレームワーク（『*Quick Look Framework Reference*』を参照）があります。

iOS 4.2から導入されたQuick Lookフレームワークは、さまざまな種類のドキュメントに対応した印刷サポートが組み込まれています。印刷とコピーの機能をカスタマイズして提供したい場合は、Document Interaction Controllerのデリゲートにメソッドを実装することによって、その機能を追加できます。

`UIDocumentInteractionController`クラスと`QLPreviewController`クラスの使用方法を示すコードについては、サンプルコードプロジェクト「*DocInteraction*」を参照してください。

ドキュメントのプレビューとオプションの表示

アプリケーションが直接的にはサポートしていないファイル（iWork文書、PDFファイル、画像ファイルなど）を扱うための選択肢をユーザに提供するには、Document Interaction Controller（`UIDocumentInteractionController`クラスのインスタンス）を使用します。Document Interaction Controllerは、ファイルのプレビューをモーダルモードで表示できます。また、ファイルを使用するためのさまざまな選択肢（別のアプリケーションでファイルを開いたり、AirPrintを使用してファイルを印刷するなど）を含むメニューを表示できます。

関連項目： 「ファイルのプレビューとオープン」（9 ページ）

ファイルタイプのサポートの登録

アプリケーションで特定の種類のファイルを開くことができる場合は、Xcodeプロジェクトの`Info.plist`ファイルにその情報を登録できます。他のアプリケーションがその種類のファイルを開こうとすると、ユーザに提示されるオプションメニューの中にデベロッパのアプリケーションが表示されます。

関連項目： 「アプリケーションでサポートするファイルタイプの登録」（13 ページ）

ほかのアプリケーションからのファイルのオープン

システムからアプリケーションに、特定のファイルを開いてユーザに表示するように要請する場合があります。これは通常、サポート対象としてデベロッパが登録した種類のファイルに、別のアプリケーションが遭遇した場合に発生します。この場合、システムはデベロッパのアプリケーションにファイルのURLを渡し、アプリケーションをフォアグラウンドに移します。

関連項目： 「サポート対象のファイルタイプのオープン」（15 ページ）

Quick Lookプレビューの表示と印刷

ファイルのプレビューをきめ細かく制御するために、Quick Lookフレームワークを直接使用できます。プレビューを表示するときに使用するアニメーションのスタイルを選ぶことができます。また、1つの項目だけでなく、複数の項目をプレビューすることもできます。Quick LookのPreview Controllerは、サポート対象のファイルタイプのAirPrint印刷機能も提供しています。

関連項目： 「Quick Lookフレームワークの使用」（17 ページ）

ファイルのプレビューとオープン

アプリケーション自身でプレビューやオープンができないファイルとやり取りする必要がある場合は、そうしたやり取りを管理するUIDocumentInteractionControllerオブジェクトを使用します。Document Interaction Controllerは、Quick Lookフレームワークと連携して、ファイルをその場でプレビューできるかどうかや、別のアプリケーションで開くことができるかどうかを判断します。次に、アプリケーションはDocument Interaction Controllerと連携して、利用可能な選択肢を適切なタイミングでユーザに表示します。

Document Interaction Controllerを使用するには、次の手順を実行します。

1. やり取りしたいファイルごとにUIDocumentInteractionControllerクラスのインスタンスを作成します。
2. アプリケーションのユーザインターフェイスに、そのファイルの表現を提供します（通常は、ファイル名または何らかのアイコンを表示します）。
3. ユーザがそのファイル表現を操作したときは（タップなど）、Document Interaction Controllerに次のいずれかのインターフェイスの表示を依頼します。
    - ファイルの内容を表示するファイルプレビュー
    - ファイルをプレビューしたり、別のアプリケーションを使用してファイルを開くための選択肢を含むメニュー。コピーや印刷の選択肢をこのメニューに追加することもできます。それには、Document Interaction Controllerのデリゲートに適切なメソッドを実装します。
    - 別のアプリケーションを使用してファイルを開くことだけをユーザに指示するメニュー

    Document Interaction Controllerは、これらのアクションを直接実装できる組み込みのGesture Recognizerを提供しています。

ファイルとやり取りをする任意のアプリケーションはすべて、Document Interaction Controllerを使用できます。これらの機能が必要になる可能性が最も高いのは、ネットワークからファイルをダウンロードするアプリケーションです。たとえば、電子メールアプリケーションでは、Document Interaction Controllerを使用して、電子メールに添付されたファイルをプレビューしたり開くことができます。

また、Document Interaction Controllerは、ファイルをダウンロードしないアプリケーションにとっても役立ちます。たとえば、アプリケーションでファイル共有をサポートする場合（「File-Sharing Support」 in *iOS Technology Overview*とサンプルコードプロジェクト「*DocInteraction*」を参照）、Document Interaction Controllerを使用して、アプリケーションのDocuments/Sharedディレクトリに割り当てられたファイルを扱うことができます。

Document Interaction Controllerの作成と設定

Document Interaction Controllerを新規作成するには、扱いたいファイルで`UIDocumentInteractionController`を初期化し、デリゲートオブジェクトを割り当てます。このデリゲートは、ビューを表示するために必要な情報をコントローラに提供します。また、必要であれば、これらのビューが表示されたときやユーザがビューを操作したときに、追加のアクションを実行するための情報も提供します。

次のコードでは、Document Interaction Controllerを新規に作成して、デリゲートを現在のオブジェクトに設定しています。このメソッドの呼び出し元は、返されたオブジェクトを保持する必要がある点に注意してください。

```
- (UIDocumentInteractionController *) setupControllerWithURL:(NSURL) fileURL
    usingDelegate:(id <UIDocumentInteractionControllerDelegate>)
interactionDelegate {

    UIDocumentInteractionController *interactionController =
        [UIDocumentInteractionController interactionControllerWithURL:fileURL];
    interactionController.delegate = interactionDelegate;

    return interactionController;
}
```

Document Interaction Controllerを作成したら、そのプロパティを使用して、対応するファイルについての情報（名前、種類、URLなど）を取得できます。Document Interaction Controllerは、ドキュメントのさまざまなサイズのアイコンを表す`UIImage`オブジェクトを含む`icons`プロパティも持っています。この情報はすべて、ユーザインターフェイス上にファイルを表示するときに利用できます。

別のアプリケーションでファイルを開くことができるようにする場合は、Document Interaction Controllerの`annotation`プロパティを使用して、オープン用のアプリケーションにカスタム情報を渡すことができます。オープン用のアプリケーションが認識できる形式で情報を提供するのは、デベロッパの責任です。たとえば、このプロパティを、同じアプリケーションスイート内のアプリケーションで使用できます。あるアプリケーションが、ファイルに関する追加情報を同じスイート内の別のアプリケーションに伝達したい場合、`annotation`プロパティを利用します。オープン用のアプリケーションは、この注釈データを、渡されたoptionsディクショナリの`UIApplicationLaunchOptionsAnnotationKey`キーに関連付けられた値として参照します。

Document Interaction Controllerの表示

Document Interaction Controllerを使用して、ファイルのプレビューを表示したり、ファイルに対するアクションをユーザに選ばせることができます。

- ファイルのプレビューをモーダルモードで表示するには、`presentPreviewAnimated:`メソッドを呼び出します。
- ユーザに一連の選択肢（別のアプリケーションでファイルを開くなど）を提示するには、`presentOptionsMenuFromRect:inView:animated:`メソッドまたは`presentOptionsMenuFromBarButtonItem:animated:`メソッドを呼び出します。
- 別のアプリケーションでファイルを開くオプションのみをユーザに選択させるには、`presentOpenInMenuFromRect:inView:animated:`メソッドまたは`presentOpenInMenuFromBarButtonItem:animated:`メソッドを呼び出します。

これらの各メソッドは、1つのビュー（ドキュメントのプレビューまたはメニューのいずれか）の表示を試みます。これらのメソッドを呼び出したときは、その戻り値を確認します。NOの戻り値は、要求したビューに何もコンテンツが含まれていないため、表示されないことを表します。たとえば、ファイルを開くことができるアプリケーションがインストールされていない場合、presentOpenInMenuFromRect:inView:animated:メソッドはNOを返します。

ファイルのプレビューを表示するメソッドを呼び出す場合は、デリゲートオブジェクトはdocumentInteractionControllerViewControllerForPreview:メソッドを実装している必要があります。プレビューはモーダルモードで表示されるため、このメソッドから返されるView Controllerはそのプレビューの親になります。このメソッドを実装していない場合や、メソッドからnilが返された場合、あるいは、返されたView Controllerが別のモーダルView Controllerを表示できない場合は、ドキュメントのプレビューは表示されません。

Document Interaction Controllerは、自身が表示したビューの破棄を自動的に行います。ただし、必要であれば、dismissMenuAnimated:メソッドまたはdismissPreviewAnimated:メソッドを呼び出して、プログラムでビューを破棄することもできます。

Gesture Recognizerを使用してDocument Interaction Controllerを表示する方法を示すサンプルコードについては、*DocInteraction*を参照してください。

アプリケーションでサポートするファイルタイプの登録

アプリケーションで特定のタイプのファイルを開ける場合は、そのことをシステムに登録するべきです。登録することにより、ほかのアプリケーションはiOSのDocument Interactionテクノロジーを利用して、そのファイルをデベロッパのアプリケーションに渡すという選択肢をユーザに提示できます。

ファイルタイプのサポートを宣言するには、アプリケーションの`Info.plist`プロパティリストファイルに`CFBundleDocumentTypes`キーを含めなければなりません（「Core Foundation Keys」を参照）。システムは、この情報をレジストリに追加します。ほかのアプリケーションは、Document Interaction Controllerを介してこの情報にアクセスできます。

`CFBundleDocumentTypes`キーには、辞書の配列が含まれており、それぞれの辞書は特定のドキュメントタイプについての情報を識別します。通常、1つのドキュメントタイプは、特定のファイルタイプと1対1に対応します。ただし、アプリケーションで複数のファイルタイプを同じように扱う場合は、これらのファイルタイプをグループにまとめて、1つのドキュメントタイプとして扱うこともできます。たとえば、アプリケーション固有のドキュメントタイプとして新旧2つのファイル形式がある場合は、両方を1つのドキュメントタイプエントリにまとめることができます。これによって、新旧の各ファイルは同じドキュメントタイプのように見えるため、同じように扱うことができます。

`CFBundleDocumentTypes`配列内のそれぞれの辞書には、次のキーを含めることができます。

- `CFBundleTypeName` ― ドキュメントタイプの名前を指定します。
- `CFBundleTypeIconFiles` ― ドキュメントのアイコンとして使用する画像リソース用のファイル名の配列です。
- `LSItemContentTypes` ― このグループ内でサポート可能なファイルタイプを表すUTI typesを含む文字列の配列が含まれます。
- `LSHandlerRank` ― このアプリケーションがドキュメントタイプを所有しているのか、それとも単にそのドキュメントタイプを開けるだけなのかを表します。

アプリケーション側から見ると、ドキュメントは、そのアプリケーションがサポートし、1つのエンティティとして扱う1つの（または複数の）ファイルタイプです。たとえば、画像処理アプリケーションでは、さまざまな画像ファイル形式を別々のドキュメントタイプとして扱うことによって、それぞれの形式に対応する動作をきめ細かく調節できます。逆に、ワードプロセッサアプリケーションでは、元の画像形式には関知せず、単純にすべての画像形式を1つのドキュメントタイプとして扱うことが考えられます。

リスト1に、カスタムファイルタイプを開くことができるアプリケーションの`Info.plist`のXMLの例を示します。`LSItemContentTypes`キーは、このファイル形式に関連付けられているUTIを示します。`CFBundleTypeIconFiles`キーは、それを表示するときに使用するアイコンリソースを指します。

リスト 1　　カスタムファイル形式用のドキュメントタイプ情報

```
<dict>
   <key>CFBundleTypeName</key>
```

```
    <string>My File Format</string>
    <key>CFBundleTypeIconFiles</key>
        <array>
            <string>MySmallIcon.png</string>
            <string>MyLargeIcon.png</string>
        </array>
    <key>LSItemContentTypes</key>
        <array>
            <string>com.example.myformat</string>
        </array>
    <key>LSHandlerRank</key>
    <string>Owner</string>
</dict>
```

CFBundleDocumentTypesキーの内容の詳細については、『*Information Property List Key Reference*』のこのキーの説明を参照してください。

サポート対象のファイルタイプを開く

システムがアプリケーションに、特定のファイルを開いてユーザに表示するように要請する場合があります。通常、このような状況は、別のアプリケーションがそのファイルを検出して、それを処理するためにDocument Interaction Controllerを使用した場合に発生します。開くべきファイルの情報は、アプリケーションデリゲートの`application:didFinishLaunchingWithOptions:`メソッドで受け取ります。アプリケーションがカスタムファイルタイプを扱う場合は、（`applicationDidFinishLaunching:`メソッドではなく）このデリゲートメソッドを実装して、それを使用してアプリケーションを初期化しなければなりません。

`application:didFinishLaunchingWithOptions:`メソッドに渡されたoptionsディクショナリには、開くべきファイルについての情報が含まれています。具体的には、次のキーをこのディクショナリ内で探す必要があります。

- `UIApplicationLaunchOptionsURLKey`―開くべきファイルを指定する`NSURL`オブジェクトが含まれています。
- `UIApplicationLaunchOptionsSourceApplicationKey`― オープン要求を出したアプリケーションのバンドル識別子を含む`NSString`が含まれています。
- `UIApplicationLaunchOptionsSourceApplicationKey`― ファイルを開くときにそのファイルに関連付けたいと要求元アプリケーションが希望していたプロパティリストオブジェクトが含まれています。

`UIApplicationLaunchOptionsURLKey`キーが存在する場合、アプリケーションはそのキーで参照されるファイルを開いて、その内容を直ちに表示しなければなりません。ディクショナリ内のその他のキーを使用して、ファイルを開くときの環境についての情報を収集できます。

Quick Lookフレームワークの使用

ファイルのプレビューをきめ細かく制御するために、Quick Lookフレームワークを直接使用できます。このフレームワークで一番重要なクラスは`QLPreviewController`です。このクラスは、プレビューアクションに応答するためにデリゲートを利用し、プレビュー項目を提供するためにデータソースを利用します。

iOS 4.2以降では、Quick LookのPreview Controllerによって提供される特殊なビューに、アクションボタンと「Print（プリント）」項目が付いています。Preview Controllerがファイルのプレビューを提供できる場合は、それを印刷することもできます。印刷用のコードを記述する必要はありません。

Quick LookのPreview Controllerを表示するには、次のいずれかの方法を使用できます。

- `UINavigationController`オブジェクトを使用してPreview Controllerをビューにプッシュします。
- 親クラス（`UIViewController`）の`presentModalViewController:animated:`メソッドを使用して、フルスクリーンのモーダルモードでPreview Controllerを表示します。
- Document Interaction Controllerを表示します（「ファイルのプレビューとオープン」（9 ページ）を参照）。ユーザは、Document Interaction Controllerのオプションメニューから「クイックルック」を選ぶことによって、Quick LookのPreview Controllerを呼び出すことができます。

Quick LookのPreview Controllerをデベロッパ自身で用意する場合は、アプリケーションの外観とナビゲーションのスタイルに最も適した表示オプションを選びます。アプリケーションでナビゲーションバーを使用しない場合は、フルスクリーンのモーダルモード表示が最適です。アプリケーションでiPhoneスタイルのナビゲーションを使用する場合は、プレビューをビューにプッシュする方法を選択できます。

表示されるプレビューには、項目のURLの最後のパスコンポーネントから取得したタイトルが入ります。これをオーバーライドするには、プレビュー項目の`previewItemTitle`アクセサを実装します。

Quick LookのPreview Controllerは、以下の項目のプレビューを表示できます。

- iWork文書
- Microsoft Office文書（Office 97以降）
- Rich Text Format(RTF)文書
- PDFファイル
- 画像
- `public.text`タイプに準拠したUTI(Uniform Type Identifier)を持つテキストファイル（『*Uniform Type Identifiers Reference*』を参照）。
- カンマ区切り(CSV)ファイル

Quick LookのPreview Controllerを使用するには、『*QLPreviewControllerDataSource Protocol Reference*』に記載されているメソッドを使用してデータソースオブジェクトを提供しなければなりません。このデータソースは、Preview Controllerにプレビュー項目を提供し、プレビュー用のナビゲーションリスト含める項目数を伝えます。このリストに複数の項目が存在する場合は、モーダルモード表示の（フルスクリーンの）コントローラがナビゲーションの矢印を表示します。これを利用して、ユーザは項目を切り替えることができます。Navigation Controllerを使用してプッシュされたQuick LookのPreview Controllerの場合は、プレビュー項目のリスト内を移動するために、ナビゲーションバーにボタンを提供することができます。

Quick Lookフレームワークの詳細については、『*Quick Look Framework Reference*』を参照してください。

書類の改訂履歴

この表は「*iOS*ドキュメントインタラクションプログラミングトピックス」の改訂履歴です。

日付	メモ
2010-11-15	プレビュー機能を提供する方法と、特定のファイルタイプを開くように登録する方法を解説した新規文書。