

# iOS Technology Usage Guidelines

## Routing

- > Deliver the functionality your app promises.
- > Streamline the UI for ease of use.
- > Focus on the route.
- > Provide information for every step of a route.
- > When users transition to your app from Maps, don't ask them to reenter information.
- > Display transit information both graphically and textually.
- > Enrich map views with additional information.
- > Give users different ways to sort multiple transit options.
- > Consider using push notifications to give people important information about their route.

## Passbook

- > As much as possible, avoid simply reproducing an existing physical pass.
- > Be selective about the information you put on the front of a pass.
- > In general, avoid using a plain white background.
- > Use the logo text field for your company name.
- > Use a white company logo.
- > Use a rectangular barcode when possible.
- > Optimize images for performance.
- > Enhance the utility of a pass by updating it when appropriate.

## Social Media

- > Give users a convenient way to compose a post without leaving your app.
- > When possible, avoid asking users to sign into a social media account.
- > Consider using an activity view controller to help users choose one of their social media accounts.

## iCloud

- > If appropriate, make it easy for users to enable iCloud for your app.
- > Respect the user's iCloud space.
- > Avoid asking users to choose which documents to store in iCloud.
- > Determine which types of information to store in iCloud.
- > Make sure that your app behaves reasonably when iCloud is unavailable.
- > Avoid giving users the option to create a "local" document.

- > When appropriate, update content automatically.
- > Warn users about the consequences of deleting a document.
- > Tell users about conflicts as soon as possible, but only when necessary.
- > Be sure to include the user's iCloud content in searches.

#### In-App Purchase

- > Elegantly integrate the store experience into your app.
- > Use simple, succinct titles and descriptions.
- > Don't alter the default confirmation alert.

#### Game Center

- > Don't create custom UI that prompts users to sign into Game Center.
- > In general, use the standard Game Center UI.
- > Give users the ability to turn off voice chat.

#### Multitasking

- > Be prepared for interruptions, and be ready to resume.
- > Make sure your UI can handle the double-high status bar.
- > Be ready to pause activities that require people's attention or active participation.
- > Ensure that your audio behaves appropriately.
- > Use local notifications sparingly.
- > When appropriate, finish user-initiated tasks in the background.

#### Notification Center

- > Keep badge contents up to date.
- > Don't send multiple notifications for the same event.
- > Provide a custom message that does not include your app name.
- > Optionally, provide a custom title for the action button in an alert.
- > Provide a sound that users can choose to hear when a notification arrives.
- > Optionally, provide a launch image.

#### Printing

- > Use the system-provided Share button.
- > Display the Print item when printing is a primary function in the current context.
- > If appropriate, provide additional printing options to users.
- > Don't display print-specific UI if users can't print.

## iAd Rich Media Ads

- > Place a standard banner view at or near the bottom of the screen.
- > Place a medium rectangle banner view where it doesn't interfere with the user's content.
- > Present a full screen banner modally when there are interludes in the user experience.
- > Present a full screen banner nonmodally when there are transitions between app views.
- > Ensure that all banners appear when and where it makes sense in your app.
- > Ensure that all banners appear when and where it makes sense in your app.
- > Avoid displaying banners on screens that users are likely to see only briefly.
- > As much as possible, display banner ads in both orientations.
- > Don't allow standard or medium rectangle banners to scroll off the screen.
- > While people view or interact with ads, pause activities that require their attention or interaction.
- > Don't stop an ad, except in rare circumstances.

## Location Services and Data Privacy

- > Make sure users understand why they're being asked to share their personal data.
- > Describe why your app needs the information, if it's not obvious.
- > Ask permission at app startup only if your app can't perform its primary function without the user's data.
- > Avoid making programmatic calls that trigger the alert before the user actually selects the feature that needs the data.
- > For location data, check the Location Services preference to avoid triggering the alert unnecessarily.

## Quick Look Document Preview

- > On iPad, display a document preview modally.
- > On iPhone, display a document preview in a dedicated view, preferably a navigation view.

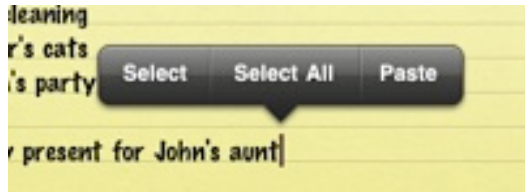
## Sound

### Define the Audio Behavior of Your App

- > If necessary, you can adjust relative, independent volume levels to produce the best mix in your app's audio output.

- > Ensure that your app can display the audio route picker, if appropriate.
- > If you need to display a volume slider, be sure to use the system-provided volume slider available when you use the `MPVolumeView` class.
- > If your app produces only UI sound effects that are incidental to its functionality, use System Sound Services.
- > If sound plays an important role in your app, use Audio Session Services.

## Edit Menu



- > Display commands that make sense in the current context.
- > Accommodate the menu display in your layout.
- > Support both gestures that people can use to invoke the menu.
- > Avoid creating a button in your UI that performs a command that's available in the edit menu.
- > Consider enabling the selection of static text if it's useful to the user.
- > Don't make button titles selectable.
- > Combine support for undo and redo with your support of copy and paste.

## Undo and Redo

- > Supply brief descriptive phrases that tell users precisely what they're undoing or redoing.
- > Avoid supplying text that is too long.
- > Avoid overloading the shake gesture.
- > Use the system-provided Undo and Redo buttons only if undo and redo are fundamental tasks in your app.
- > Clearly relate undo and redo capability to the user's immediate context, and not to an earlier context.

## Keyboards and Input Views