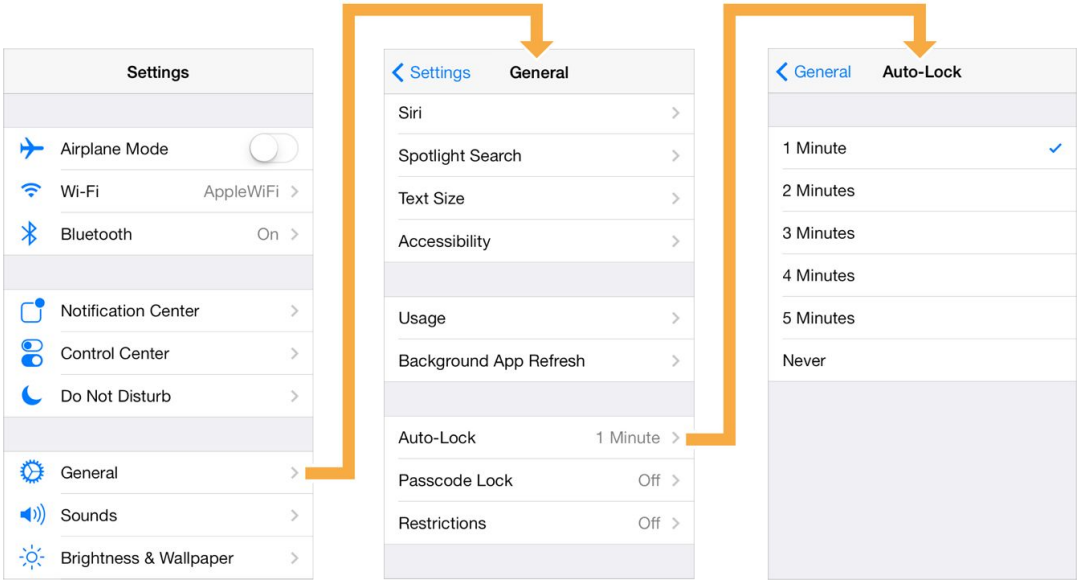
UINavigation Controller

Outline

1. Overview
2. Navigation Controller Views
3. Updating the Navigation Bar
4. Displaying a Toolbar
5. Adapting to Different Environments
6. Interface Behaviors
7. State Preservation

1. Overview

- A navigation controller is a container view controller that manages one or more child view controllers in a navigation interface. In this type of interface, only one child view controller is visible at a time.

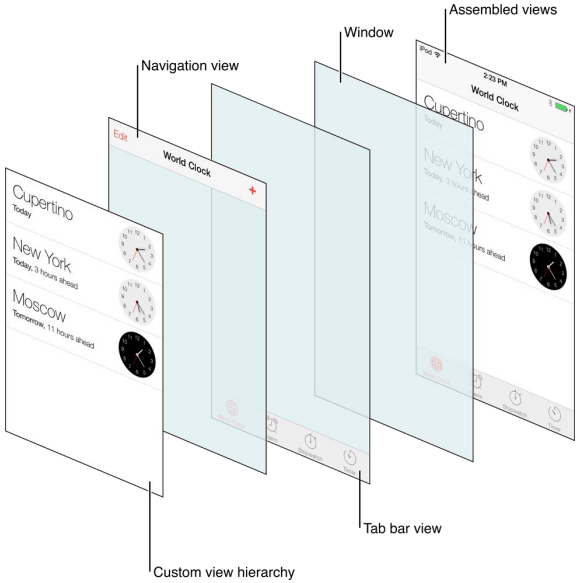


1. Overview (cont)

- A navigation controller object manages its child view controllers using an ordered array, known as the navigation stack.
  - ➔ The first view controller in the array is the root view controller and represents the bottom of the stack.
  - ➔ The last view controller in the array is the topmost item on the stack, and represents the view controller currently being displayed.
  - ➔ You add and remove view controllers from the stack using segues or using the methods of this class. The user can also remove the topmost view controller using the back button in the navigation bar or using a left-edge swipe gesture.

2. Navigation Controller Views

- A navigation controller is a container view controller—that is, it embeds the content of other view controllers inside of itself. You access a navigation controller's view from its view property. This view incorporates the navigation bar, an optional toolbar, and the content view corresponding to the topmost view controller.



3. Updating the Navigation Bar

1. General
2. The Left Item
3. The Middle Item
4. The Right Item

3.1 General

- Each time the top-level view controller changes, the navigation controller updates the navigation bar accordingly.

- Specifically, the navigation controller updates the bar button items displayed in each of the three navigation bar positions: left, middle, and right.

- Bar button items are instances of the `UIBarButtonItem` class. You can create items with custom content or create standard system items depending on your needs.

3.2 The Left Item

- For all but the root view controller on the navigation stack, the item on the left side of the navigation bar provides navigation back to the previous view controller.
- The contents of this left-most button are determined as follows:
  - If the new top-level view controller has a custom left bar button item, that item is displayed. To specify a custom left bar button item, set the leftBarButtonItem property of the view controller's navigation item.
  - If the top-level view controller does not have a custom left bar button item, but the navigation item of the previous view controller has an object in its backBarButtonItem property, the navigation bar displays that item.
  - If a custom bar button item is not specified by either of the view controllers, a default back button is used and its title is set to the value of the title property of the previous view controller—that is, the view controller one level down on the stack. (If there is only one view controller on the navigation stack, no back button is displayed.)

3.3 The Middle Item

- The navigation controller updates the middle of the navigation bar as follows:
  - If the new top-level view controller has a custom title view, the navigation bar displays that view in place of the default title view. To specify a custom title view, set the titleView property of the view controller's navigation item.
  - If no custom title view is set, the navigation bar displays a label containing the view controller's default title. The string for this label is usually obtained from the title property of the view controller itself. If you want to display a different title than the one associated with the view controller, set the title property of the view controller's navigation item instead.

3.4 The Right Item

- The navigation controller updates the right side of the navigation bar as follows:
  - If the new top-level view controller has a custom right bar button item, that item is displayed. To specify a custom right bar button item, set the rightBarButtonItem property of the view controller's navigation item.
  - If no custom right bar button item is specified, the navigation bar displays nothing on the right side of the bar.

4. Displaying a Toolbar

- A navigation controller object manages an optional toolbar in its view hierarchy.
- When displayed, this toolbar obtains its current set of items from the toolbarItems property of the active view controller.
- When the active view controller changes, the navigation controller updates the toolbar items to match the new view controller, animating the new items into position when appropriate.
- The navigation toolbar is hidden by default but you can show it for your navigation interface by calling the `setToolbarHidden(_:animated:)` method of your navigation controller object.

5. Adapting to Different Environments

- The navigation interface remains the same in both horizontally compact and horizontally regular environments. When toggling between the two environments, only the size of the navigation controller's view changes. The navigation controller does not change its view hierarchy or the layout of its views.

- When configuring segues between view controllers on a navigation stack, the standard **Show** and **Show Detail** segues behave as follows:

  - **Show segue**: The navigation controller pushes the specified view controller onto its navigation stack.
  - **Show Detail segue**: The navigation controller presents the specified view controller modally.

- The behaviors of other segue types are unchanged.

6. Interface Behaviors

- A navigation controller supports the following behaviors for its interface:
  - **Supported interface orientations**: A navigation controller object does not consult the view controllers on its navigation stack when determining the supported interface orientations. On iPhone, a navigation controller supports all orientations except portrait upside-down. On iPad, a navigation controller supports all orientations. If the navigation controller has a delegate object, the delegate can specify a different set of supported orientations using the `navigationControllerSupportedInterfaceOrientations(_:)` method.

  - **Presentation context**: A navigation controller defines the presentation context for modally presented view controllers. When the modal transition style is `UIModalPresentationStyle.currentContext` or `UIModalPresentationStyle.overCurrentContext`, modal presentations from the view controllers in the navigation stack cover the entire navigation interface.

7. State Preservation

- When you assign a value to a navigation controller's `restorationIdentifier` property, it attempts to preserve itself and the child view controllers on its navigation stack.
- The navigation controller starts at the bottom of the stack and moves upward, encoding each view controller that also has a valid restoration identifier string.
- During the next launch cycle, the navigation controller restores the preserved view controllers to the navigation stack in the same order that they were preserved.
- The child view controllers you push onto the navigation stack may use the same restoration identifiers. The navigation controller automatically stores additional information to ensure that each child's restoration path is unique.

Question & Answer?