

# Xcode

Giftbot

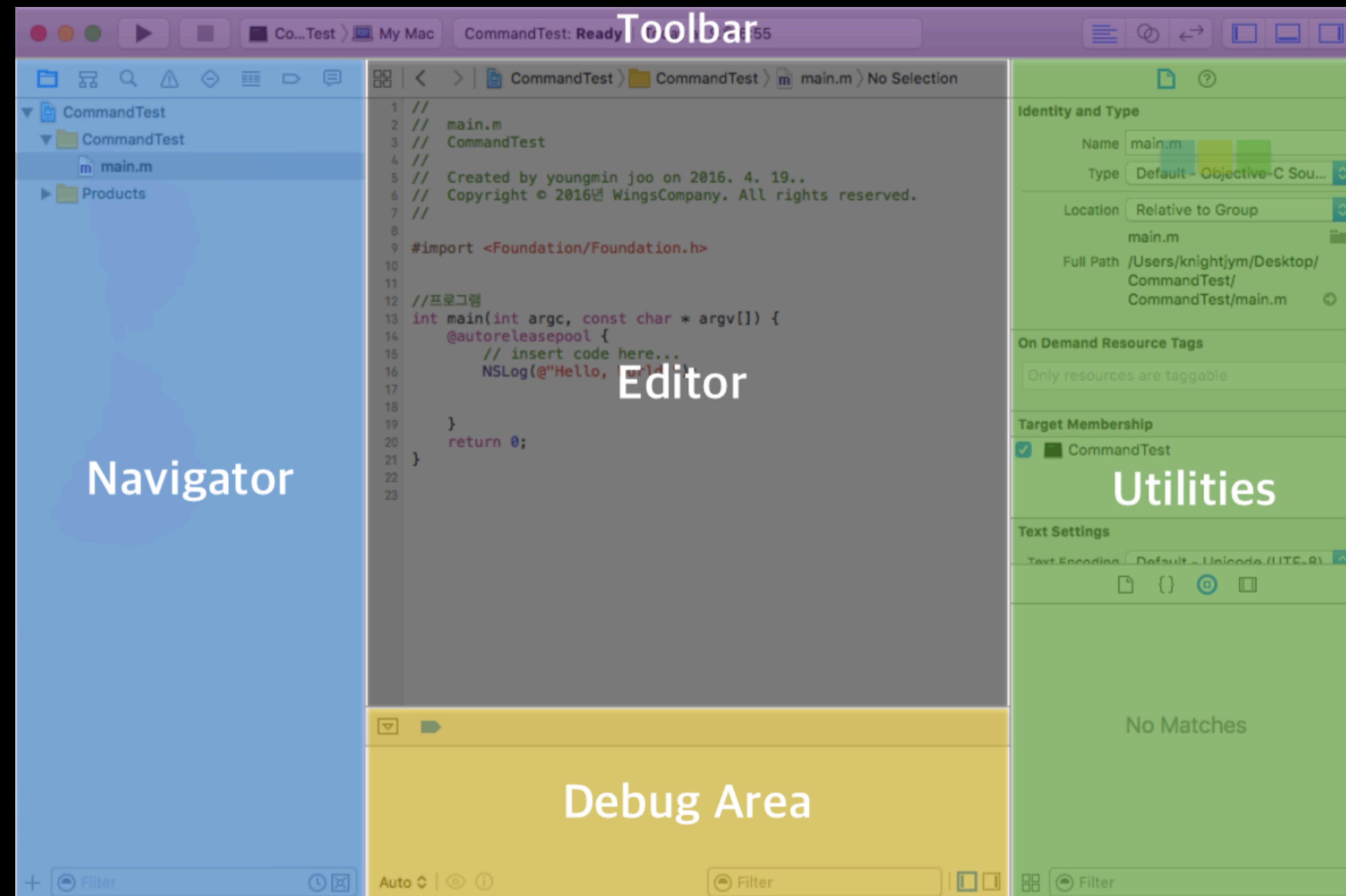
# Xcode

통합 개발 환경 (IDE, Integrated Development) = Editor + Compiler + Debugger + ...

e.g. Visual Studio, Eclipse, Delphi, Net Beans 등

macOS에서만 실행 가능

공식 지원 언어 : Swift, Objective-C, C, C++, Fotran, Ruby, Python, Java 등



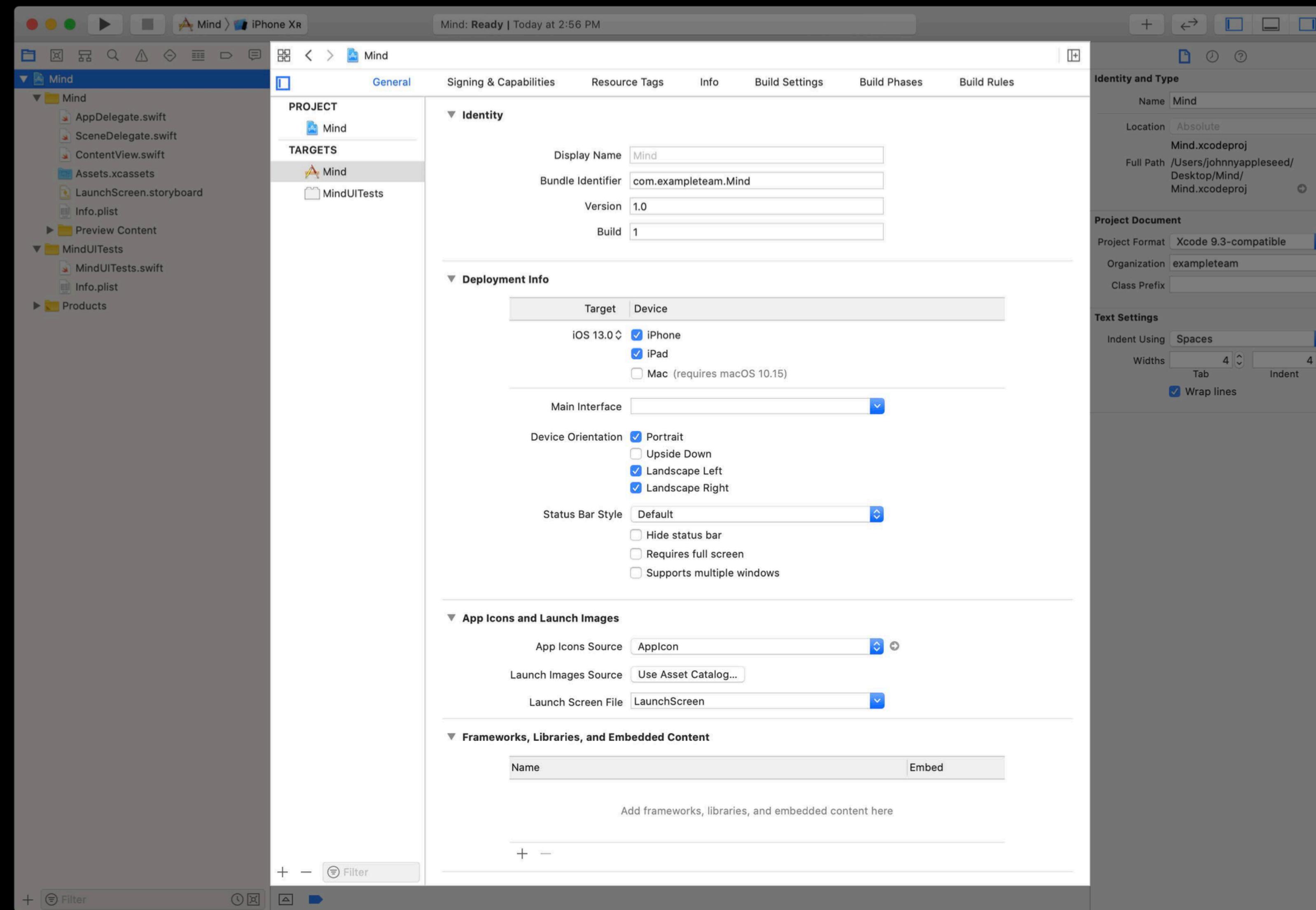
# Source Editor

The screenshot shows the Xcode Source Editor interface. The main area displays the `AppDelegate.swift` file from a project named "Mind". The code implements the `UIApplicationDelegate` protocol, handling application launch and termination, and defining a `UISceneSession Lifecycle` method.

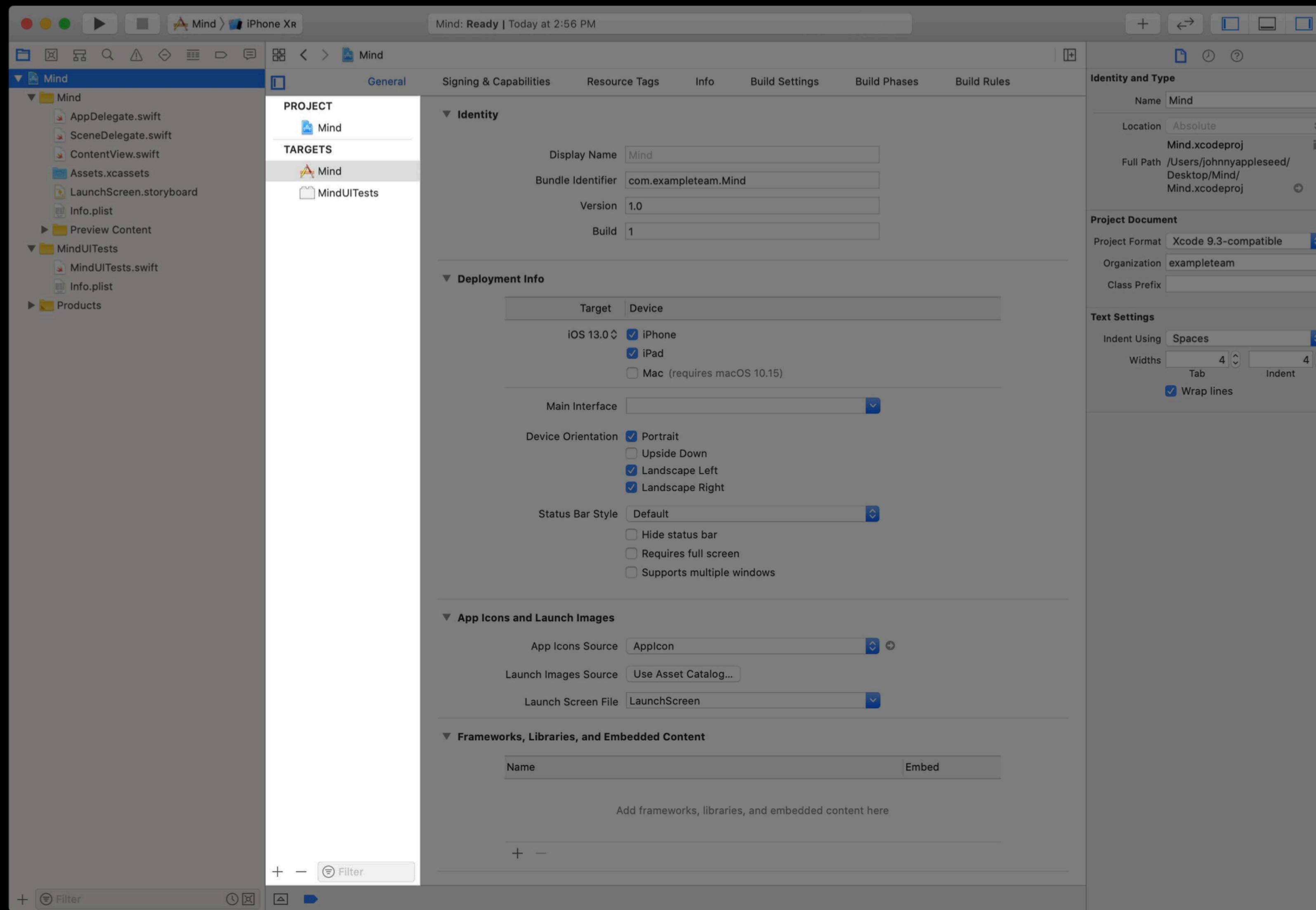
```
1 //  
2 //  AppDelegate.swift  
3 //  Mind  
4 //  
5  
6 import UIKit  
7  
8 @UIApplicationMain  
9 class AppDelegate: UIResponder, UIApplicationDelegate {  
10  
11     func application(_ application: UIApplication,  
12                      didFinishLaunchingWithOptions launchOptions:  
13                      [UIApplication.LaunchOptionsKey: Any]?) -> Bool {  
14         // Override point for customization after  
15         // application launch.  
16         return true  
17     }  
18  
19     func applicationWillTerminate(_ application:  
20                                   UIApplication) {  
21         // Called when the application is about to  
22         // terminate. Save data if appropriate. See also  
23         // applicationDidEnterBackground:.  
24     }  
25  
26     // MARK: UISceneSession Lifecycle  
27  
28     func application(_ application: UIApplication,  
29                     configurationForConnecting connectingSceneSession:  
30                     UISceneSession, options: UIScene.ConnectionOptions)  
31     -> UISceneConfiguration {  
32         // Called when a new scene session is being created.  
33         // Use this method to select a configuration to  
34         // create the new scene with.  
35         return UISceneConfiguration(name: "Default  
36                                     Configuration", sessionRole:
```

The Xcode interface includes a sidebar with file navigation, a central code editor, and a right-hand panel for "Identity and Type" settings, showing details like the file's name, type, location, and target membership.

# Project Editor



# Project Editor



# Navigator

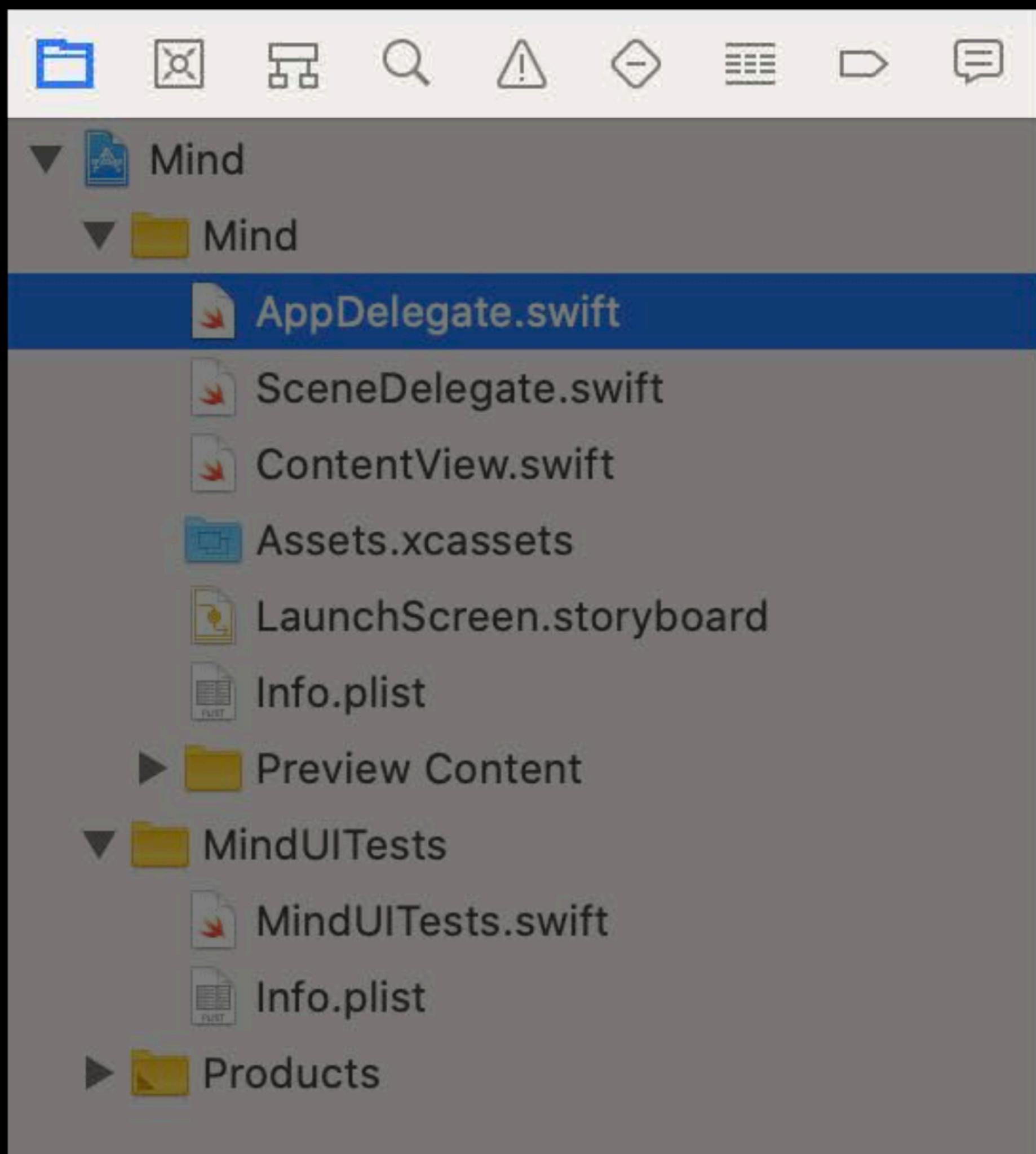
The screenshot shows the Xcode interface with the Navigator tab selected. The main editor area displays the `AppDelegate.swift` file, which is part of the `Mind` project. The code implements the `UIApplicationDelegate` protocol, handling application launch and termination, and defining a `UISceneSession Lifecycle` method.

```
// AppDelegate.swift
// Mind
// Mind
// AppDelegate.swift
// Mind
// 
// import UIKit
// 
// @UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    func application(_ application: UIApplication,
                     didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after
        // application launch.
        return true
    }
    func applicationWillTerminate(_ application:
UIApplication) {
        // Called when the application is about to
        // terminate. Save data if appropriate. See also
        // applicationDidEnterBackground:.
    }
    // MARK: UISceneSession Lifecycle
    func application(_ application: UIApplication,
                    configurationForConnecting connectingSceneSession:
UISceneSession, options: UIScene.ConnectionOptions)
-> UISceneConfiguration {
        // Called when a new scene session is being created.
        // Use this method to select a configuration to
        // create the new scene with.
        return UISceneConfiguration(name: "Default
        Configuration", sessionRole:
}
}

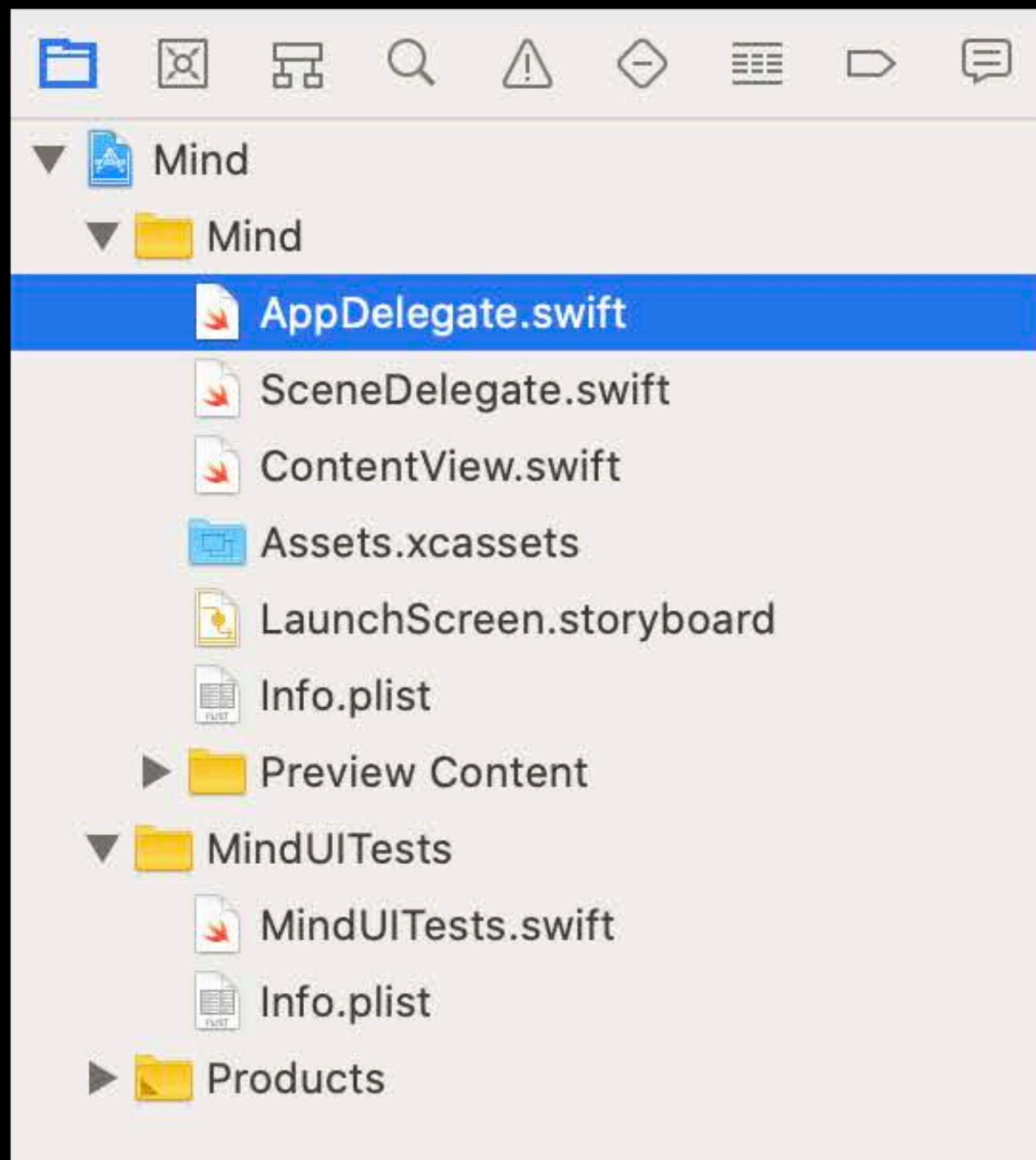
// SceneDelegate.swift
// ContentViewController.swift
// Assets.xcassets
// LaunchScreen.storyboard
// Info.plist
// Preview Content
// MindUITests
// MindUITests.swift
// Info.plist
// Products
```

The Navigator pane on the right side shows the `Identity and Type` settings for the `AppDelegate.swift` file, indicating it is a `Default - Swift Source` file located relative to the group. It also shows `Target Membership` for the `Mind` target and `Text Settings` for encoding, line endings, and indentation.

# Navigator



# Navigator



# Inspector

The screenshot shows the Xcode interface with the Inspector pane open on the right side. The file `AppDelegate.swift` is selected in the project navigator. The Inspector pane displays various settings for the file:

- Identity and Type**:
  - Name: `AppDelegate.swift`
  - Type: Default - Swift Source
  - Location: Relative to Group
  - Full Path: `/Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift`
- On Demand Resource Tags**: Only resources are taggable.
- Target Membership**:  Mind,  MindUITests
- Text Settings**:
  - Text Encoding: No Explicit Encoding
  - Line Endings: No Explicit Line Endings
  - Indent Using: Spaces
  - Widths: Tab 4, Indent 4
  - Wrap lines

The code editor shows the `AppDelegate.swift` file with the following content:

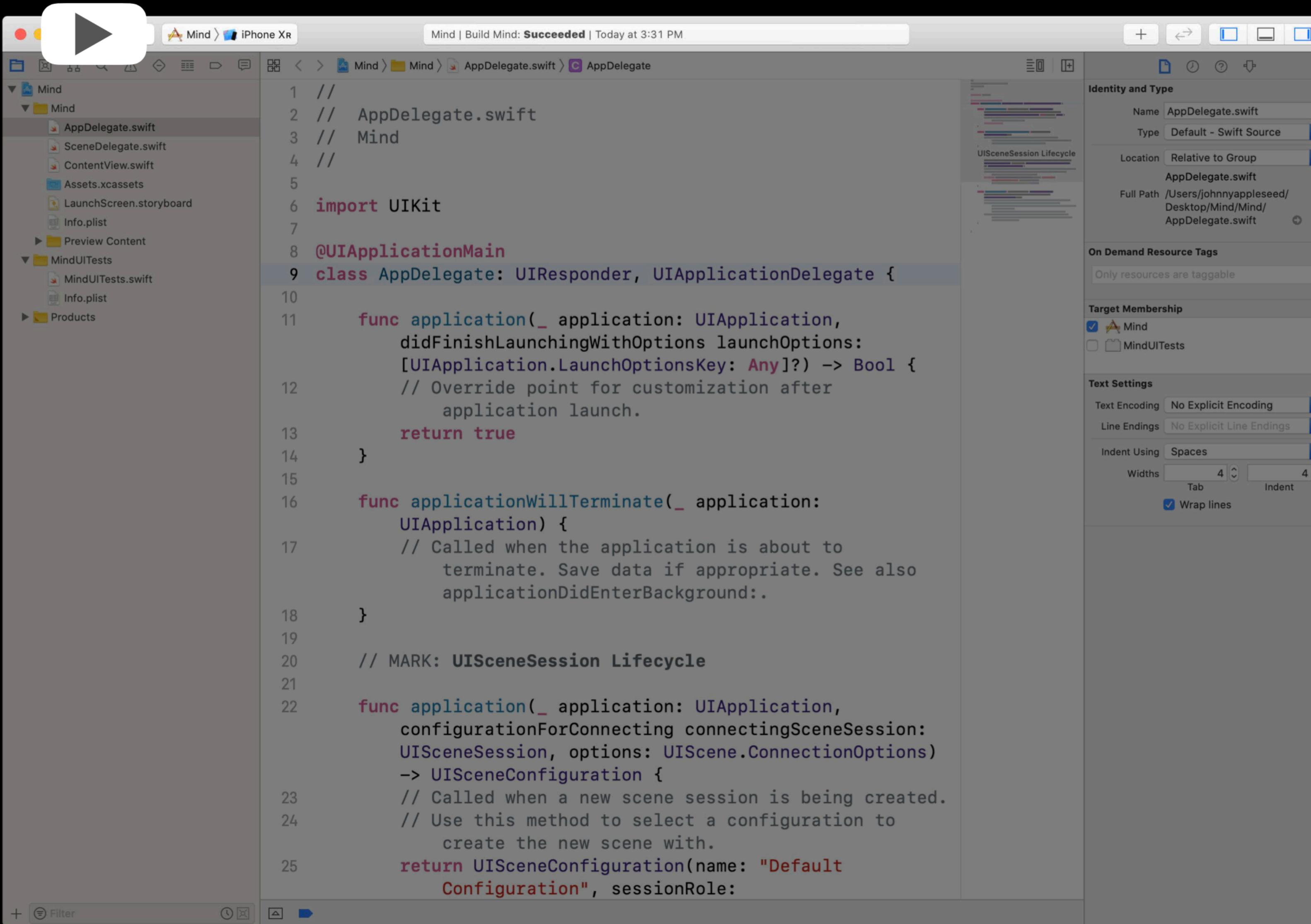
```
// AppDelegate.swift
// Mind
// import UIKit
// @UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    func application(_ application: UIApplication,
                     didFinishLaunchingWithOptions launchOptions:
                     [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after
        // application launch.
        return true
    }
    func applicationWillTerminate(_ application:
        UIApplication) {
        // Called when the application is about to
        // terminate. Save data if appropriate. See also
        // applicationDidEnterBackground:.
    }
    // MARK: UISceneSession Lifecycle
    func application(_ application: UIApplication,
                    configurationForConnecting connectingSceneSession:
                    UISceneSession, options: UIScene.ConnectionOptions)
    -> UISceneConfiguration {
        // Called when a new scene session is being created.
        // Use this method to select a configuration to
        // create the new scene with.
        return UISceneConfiguration(name: "Default
            Configuration", sessionRole:
            
```

# Toolbar

The screenshot shows the Xcode interface with the following details:

- Toolbar:** Standard Mac OS X-style toolbar with icons for file operations.
- Document Title:** Mind | Build Mind: **Succeeded** | Today at 3:31 PM
- Editor:** Shows the `AppDelegate.swift` file content. The code handles application lifecycle events like `application(_:didFinishLaunchingWithOptions:)` and `applicationWillTerminate(_)`, and implements the `UISceneSession Lifecycle` protocol.
- Utilities Panel:** Opened on the right side of the screen.
  - Identity and Type:** Shows the file's properties:
    - Name: `AppDelegate.swift`
    - Type: Default - Swift Source
    - Location: Relative to Group
    - Full Path: /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift
  - On Demand Resource Tags:** Only resources are taggable.
  - Target Membership:** Mind (checkbox checked) and MindUITests (checkbox unselected).
  - Text Settings:** Text Encoding: No Explicit Encoding, Line Endings: No Explicit Line Endings, Indent Using: Spaces, Widths: Tab 4, Indent 4, Wrap lines (checkbox checked).

# Toolbar

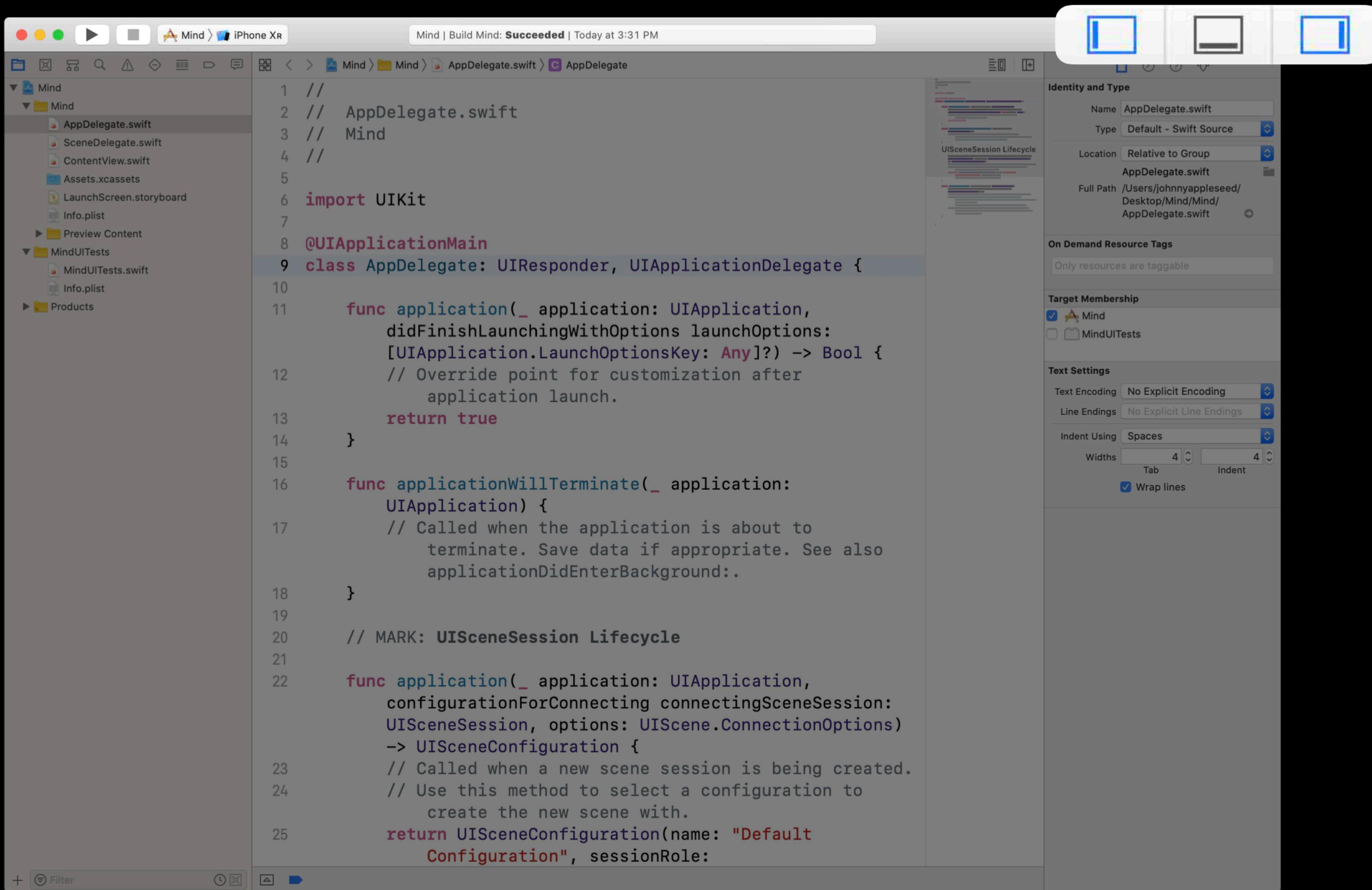


# Toolbar

The screenshot shows the Xcode interface with the following details:

- Title Bar:** Shows "Mind" and "iPhone XR" as the project and target respectively.
- Build Status:** "Mind | Build Mind: **Succeeded** | Today at 3:31 PM"
- File Navigator:** Displays the project structure under "Mind".
- Editor:** Shows the code for `AppDelegate.swift`. The code handles application launch and termination, and implements the `UISceneSession Lifecycle`.
- Utilities Panel:** Opened on the right side, showing the `Identity and Type` tab for `AppDelegate.swift`.
  - Name:** AppDelegate.swift
  - Type:** Default - Swift Source
  - Location:** Relative to Group
  - Full Path:** /Users/johnnyappleseed/Desktop/Mind/Mind/AppDelegate.swift
- Text Settings:** Includes options for Text Encoding (No Explicit Encoding), Line Endings (No Explicit Line Endings), Indent Using (Spaces), and Widths (Tab width 4, Indent width 4). The "Wrap lines" checkbox is checked.

# Toolbar



# Line Numbers

The screenshot shows the Xcode interface with the 'MeditationController.swift' file open. The code editor displays the following Swift code:

```
//  
//  MeditationController.swift  
//  Mind  
//  
//  
import Foundation  
import SwiftUI  
import Combine  
  
class MeditationController: BindableObject {  
    enum Duration: TimeInterval, CaseIterable, Hashable {  
        case fiveSeconds = 5  
        case fifteenSeconds = 15  
        case thirtySeconds = 30  
        case oneMinute = 60  
        case threeMinutes = 180  
        case fiveMinutes = 300  
        case tenMinutes = 600  
        case fifteenMinutes = 900  
  
        var formattedDuration: String {  
            return  
                MeditationController.timeFormatter  
                    .string(from: rawValue)  
        }  
    }  
  
    // MARK: - Initialization  
  
    let healthStore: HealthStore  
    var duration: Duration  
  
    init(healthStore: HealthStore =  
        HealthStoreFactory.makeHealthStore()) {  
        self.healthStore = healthStore  
        self.duration = .threeMinutes  
    }  
}
```

The code editor includes line numbers from 1 to 33. The right side of the Xcode window shows the 'Identity and Type' inspector, which details the file's properties: Name is 'MeditationController.swift', Type is 'Default - Swift Source', Location is 'Relative to Group', Full Path is '/Users/johnnyappleseed/Documents/WWDC19-IntroToXcode-App/Mind/Mind/MeditationController.swift'. The 'Text Settings' section shows Text Encoding as 'No Explicit Encoding', Line Endings as 'No Explicit Line Endings', Indent Using as 'Spaces', and Widths set to 4 for both Tab and Indent.

# Minimap

The screenshot shows the Xcode interface with the Minimap feature open. The main editor window displays the `MeditationController.swift` file. The Minimap panel on the right provides a visual summary of the code's structure, showing horizontal bars for each line of code across the entire file. The code itself is as follows:

```
//  
//  MeditationController.swift  
//  Mind  
//  
//  
//  
import Foundation  
import SwiftUI  
import Combine  
  
class MeditationController: BindableObject {  
    enum Duration: TimeInterval, CaseIterable, Hashable {  
        case fiveSeconds = 5  
        case fifteenSeconds = 15  
        case thirtySeconds = 30  
        case oneMinute = 60  
        case threeMinutes = 180  
        case fiveMinutes = 300  
        case tenMinutes = 600  
        case fifteenMinutes = 900  
    }  
  
    var formattedDuration: String {  
        return  
            MeditationController.timeFormatter  
                .string(from: rawValue)  
    }  
  
    // MARK: - Initialization  
  
    let healthStore: HealthStore  
    var duration: Duration  
  
    init(healthStore: HealthStore =  
        HealthStoreFactory.makeHealthStore()) {  
        self.healthStore = healthStore  
        self.duration = .threeMinutes  
    }  
}
```

The Xcode interface includes the following elements:

- Project Navigator:** Shows the project structure with files like `README.md`, `AppDelegate.swift`, and `LaunchScreen.storyboard`.
- Editor:** Displays the Swift code for `MeditationController.swift`.
- Minimap:** A vertical bar chart showing the structure and density of code.
- Identity and Type:** Shows the file's name, type, location, and full path.
- On Demand Resource Tags:** A section for managing resources.
- Target Membership:** Shows the target membership for the file.
- Text Settings:** Options for text encoding, line endings, and indentation.

# Jump Bar

The screenshot shows the Xcode IDE with the "MeditationController.swift" file open in the main editor. The "Jump Bar" is visible at the bottom of the window, featuring a series of small, semi-transparent preview cards for other files and sections within the same document. These cards include snippets from "HealthStore.swift", "ContentView.swift", "Icons.swift", and "MeditationSession.swift". The "Jump Bar" is a key feature in Xcode that allows quick navigation between different parts of the current file or related files.

```
//  
//  MeditationController.swift  
//  Mind  
//  
//  
//  
import Foundation  
import SwiftUI  
import Combine  
  
class MeditationController: BindableObject {  
    enum Duration: TimeInterval, CaseIterable, Hashable {  
        case fiveSeconds = 5  
        case fifteenSeconds = 15  
        case thirtySeconds = 30  
        case oneMinute = 60  
        case threeMinutes = 180  
        case fiveMinutes = 300  
        case tenMinutes = 600  
        case fifteenMinutes = 900  
    }  
  
    var formattedDuration: String {  
        return  
            MeditationController.timeFormatter  
                .string(from: rawValue)  
    }  
  
    // MARK: - Initialization  
  
    let healthStore: HealthStore  
    var duration: Duration  
  
    init(healthStore: HealthStore =  
        HealthStoreFactory.makeHealthStore()) {  
        self.healthStore = healthStore  
        self.duration = .threeMinutes  
    }  
}
```

# Jump Bar

The screenshot shows the Xcode interface with the "MeditationController.swift" file open in the main editor. A callout bubble highlights the "P" icon in the jump bar, which is used to quickly navigate to the declaration of the current symbol.

```
//  
//  MeditationController.swift  
//  Mind  
//  
//  
//  
import Foundation  
import SwiftUI  
import Combine  
  
class MeditationController: BindableObject {  
    enum Duration: TimeInterval, CaseIterable, Hashable {  
        case fiveSeconds = 5  
        case fifteenSeconds = 15  
        case thirtySeconds = 30  
        case oneMinute = 60  
        case threeMinutes = 180  
        case fiveMinutes = 300  
        case tenMinutes = 600  
        case fifteenMinutes = 900  
    }  
    var formattedDuration: String {  
        return  
            MeditationController.timeFormatter  
                .string(from: rawValue)  
    }  
}  
  
// MARK: - Initialization  
  
let healthStore: HealthStore  
var duration: Duration  
  
init(healthStore: HealthStore =  
    HealthStoreFactory.makeHealthStore()) {  
    self.healthStore = healthStore  
    self.duration = .threeMinutes
```

# Jump Bar

