

## Documentation

[Apple silicon](#) / Providing an edge-to-edge, full-screen experience in your iPad app running on a Mac

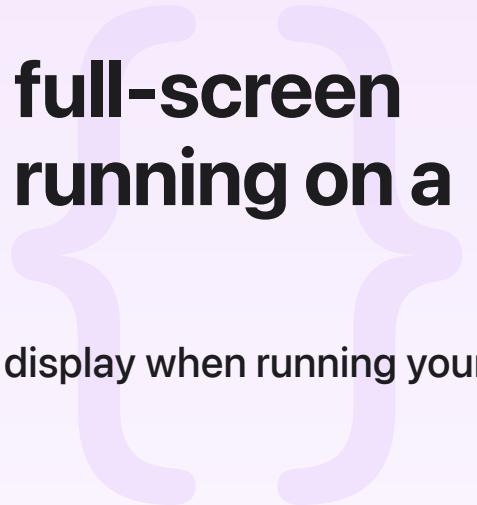
Sample Code

# Providing an edge-to-edge, full-screen experience in your iPad app running on a Mac

Take advantage of the true native resolution of a Mac display when running your iPad app in full-screen mode on a Mac.

[Download](#)

iOS 15.5+ | iPadOS 15.5+ | Xcode 13.4+



## Overview

Some iPad apps, such as games, require a full-screen experience. These apps don't participate in iPad multitasking, and they don't support resizable windows when run on a Mac. To ensure these behaviors, the apps include the [`UIRequiresFullScreen`](#) key with a value of `true` in their `Info.plist` file.

But when one of these iPad apps enters full-screen mode on a Mac, the app doesn't fill the entire screen. Instead, the app's window is a fixed size, leaving space on either side of it.

An iPad app that supports iPad multitasking, on the other hand, provides resizable windows when run on a Mac. A person can also view the app's content in full-screen mode, filling the entire Mac display.

This sample shows how to configure an iPad app that requires full-screen mode on iPad to take advantage of the native resolution of the Mac display for a pixel-perfect, edge-to-edge, full-screen experience.

## Configure the sample code project

To run the sample app:

1. Download the sample and open the project in Xcode.
2. Select a development team in the Signing & Capabilities tab of the target's settings.
3. Select My Mac (Designed for iPad) as the destination.
4. Build and run the app.

### Important

The My Mac (Designed for iPad) destination option is available only on a Mac with Apple silicon.

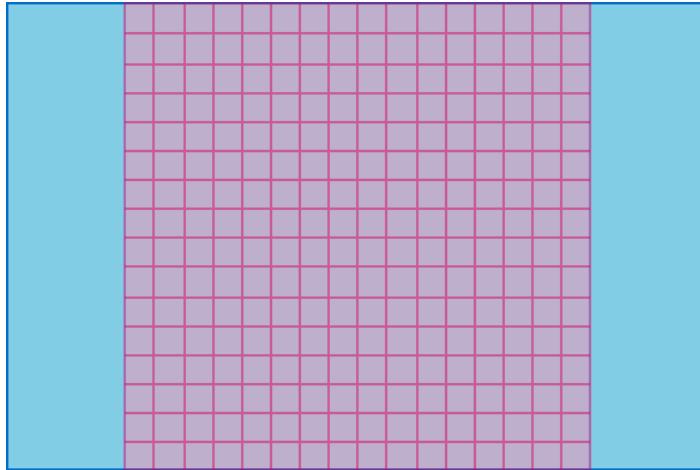
When you launch the sample app, it displays two grids across the screen. The first grid has white lines showing the pixel positions as defined in the backing store of the app's [UIWindowScene](#). Each line is 1 pixel wide, and the grid separates each line by 10 pixels. The second grid uses red lines to show the point layout of the [UIWindowScene](#) instance. Each line is 1 point wide, and the grid separates each line by 10 points.

The sample app can also show other visual elements, such as a border around the safe area insets. To change the display options, click anywhere within the app.

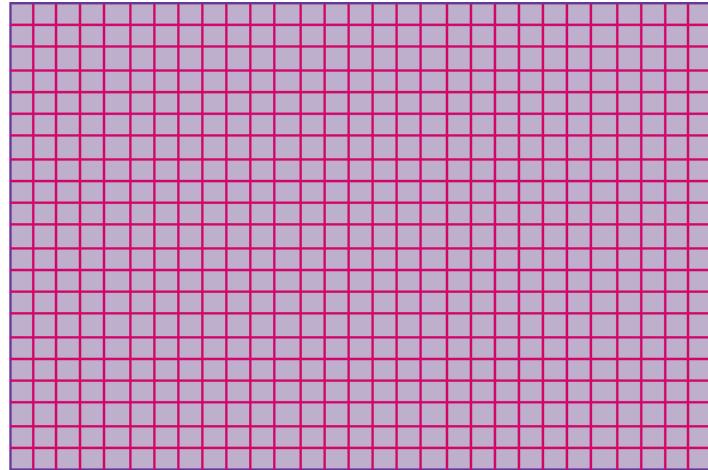
## Support the native resolution of a Mac display

An app that requires full screen on iPad — that is, an app that includes the [UIRequiresFullScreen](#) key with a value of `true` in its `Info.plist` file — doesn't take advantage of the entire display when run on a Mac in full-screen mode. The system determines the full-screen size of the app based on compatible iPad device sizes and displays the app using the aspect ratio of the chosen size. When the aspect ratio doesn't match the aspect ratio of the Mac display, space appears on either side of the app's content.

To eliminate the space and display content across the entire screen, the sample includes the [UISupportsTrueScreenSizeOnMac](#) key in its `Info.plist` file.



```
UIRequiresFullScreen = true
```



```
UIRequiresFullScreen = true  
UISupportsTrueScreenSizeOnMac = true
```

With the key `UISupportsTrueScreenSizeOnMac` set to `true`, the sample app shows the pixel and point grids across the entire display area of the Mac when running in full-screen mode. The full-screen size of the app's `UIWindowScene` matches the resolution of the display that was active at the time the app launched. This size doesn't change while the app is running, even when a person moves the window to a display that has a different resolution.

Setting the key to `true` also avoids content scaling in the app's window (when it's in full-screen mode on the display that was active at launch time) by mapping each rendered pixel in the window to one physical pixel on the screen.

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/Prop  
<plist version="1.0">  
<dict>  
    <key>UIRequiresFullScreen</key>  
    <true/>  
    <key>UISupportsTrueScreenSizeOnMac</key>  
    <true/>  
    ...  
</dict>  
</plist>
```

### Important

For more information about the responsibilities of an iPad app that supports arbitrary screen sizes and resolutions when running on Mac, see the `UISupportsTrueScreenSizeOnMac` documentation.

# Launch in full-screen mode by default

Many apps that require full-screen mode when running on iPad look equally as good on Mac when appearing in a window. For these apps, setting `UIRequiresFullScreen` to true provides a full-screen-only experience when the app runs in iPadOS and lets the app appear in a window by default when launched in macOS.

But for some apps like this sample app, the ideal experience is to run in full-screen mode on both iPad and Mac. Instead of appearing in a window by default when launched on a Mac, the sample app launches in full-screen mode.

To launch in full-screen mode by default, the sample includes the `UILaunchToFullScreenByDefaultOnMac` key with a value of `true` in its `Info.plist` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>UIRequiresFullScreen</key>
    <true/>
    <key>UISupportsTrueScreenSizeOnMac</key>
    <true/>
    <key>UILaunchToFullScreenByDefaultOnMac</key>
    <true/>
    ...
</dict>
</plist>
```

When the app launches for the first time, it displays in full-screen mode. However, state restoration can override this behavior if the person using the app exits full-screen mode before quitting the app.

## See Also

### iOS apps on Mac

-  [Running your iOS apps in macOS](#)  
Modernize the iOS apps you choose to run on a Mac with Apple silicon, or opt out of running on a Mac altogether.
-  [Adapting iOS code to run in the macOS environment](#)

Support modern iOS features that result in a better user experience when running on Apple silicon.

{ } Providing touch gesture equivalents using Touch Alternatives

Enable Touch Alternatives to provide keyboard, mouse, and trackpad equivalents to your iOS app when it runs on a Mac with Apple silicon.