Build an iOS app with .NET CLI

Article • 09/30/2024

In this tutorial, you'll learn how to create and run a .NET Multi-platform App UI (.NET MAUI) app on iOS using .NET Command Line Interface (CLI) on macOS:

1. To create .NET MAUI apps, you'll need to download and run the installer for the latest .NET runtime. You'll also need to download and install the version of Xcode that's required by the version of .NET MAUI that you're using. For information, see Release versions .

2. On your Mac, open **Terminal** and check that you have the latest .NET runtime installed:

   zsh
   ```
   dotnet --version
   ```

3. In **Terminal**, install the latest public build of .NET MAUI:

   zsh
   ```
   sudo dotnet workload install maui --source
   https://api.nuget.org/v3/index.json
   ```

   This command will install the latest released version of .NET MAUI, including the required platform SDKs.

4. In **Terminal**, create a new .NET MAUI app using .NET CLI:

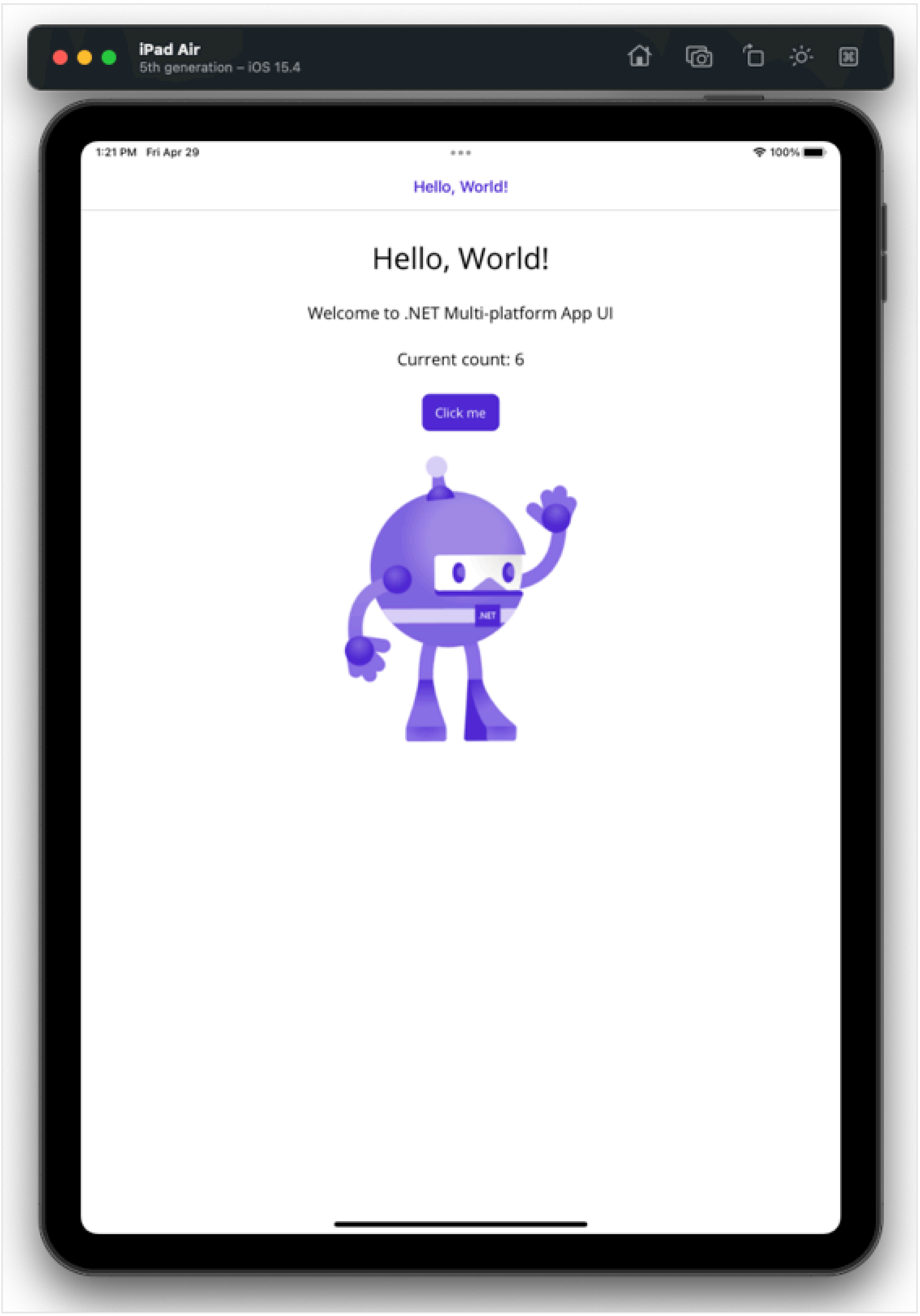
   zsh
   ```
   dotnet new maui -n "MyMauiApp"
   ```

5. In **Terminal**, change directory to *MyMauiApp*, and build and run the app:

   zsh
   ```
   cd MyMauiApp
   dotnet build -t:Run -f net8.0-ios
   ```

   The `dotnet build` command will restore the project the dependencies, build the app, and launch it in the default simulator.

6. In the default simulator, press the **Click me** button several times and observe that the count of the number of button clicks is incremented.



Build with a specific version of Xcode

If you have multiple versions of Xcode installed on your Mac, it's possible to specify which Xcode version should be used when building your app. There are a number of approaches that can be used to accomplish this, but the recommended approach is to set the `MD_APPLE_SDK_ROOT` environment variable to the path of the Xcode version.

> ⚠ **Warning**
>
> Using `xcode-select -s` to set the version of Xcode to use isn't recommended.

To set the `MD_APPLE_SDK_ROOT` environment variable for the duration of the current terminal session:

1. Open the **Terminal** application.
2. Type the following command, substituting in your version of Xcode, and press Enter:

   zsh

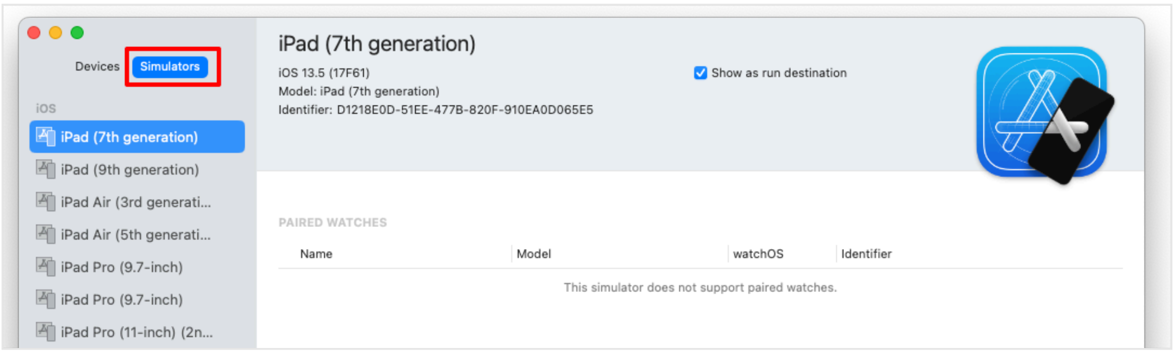
   ```zsh
   export MD_APPLE_SDK_ROOT=/Applications/Xcode_14.1.0.app
   ```

If you want to set this environment variable permanently, you can add the `export` command to your shell profile, such as *.zprofile*.

Launch the app on a specific simulator

A .NET MAUI iOS app can be launched on a specific iOS simulator from a Mac by providing its unique device id (UDID):

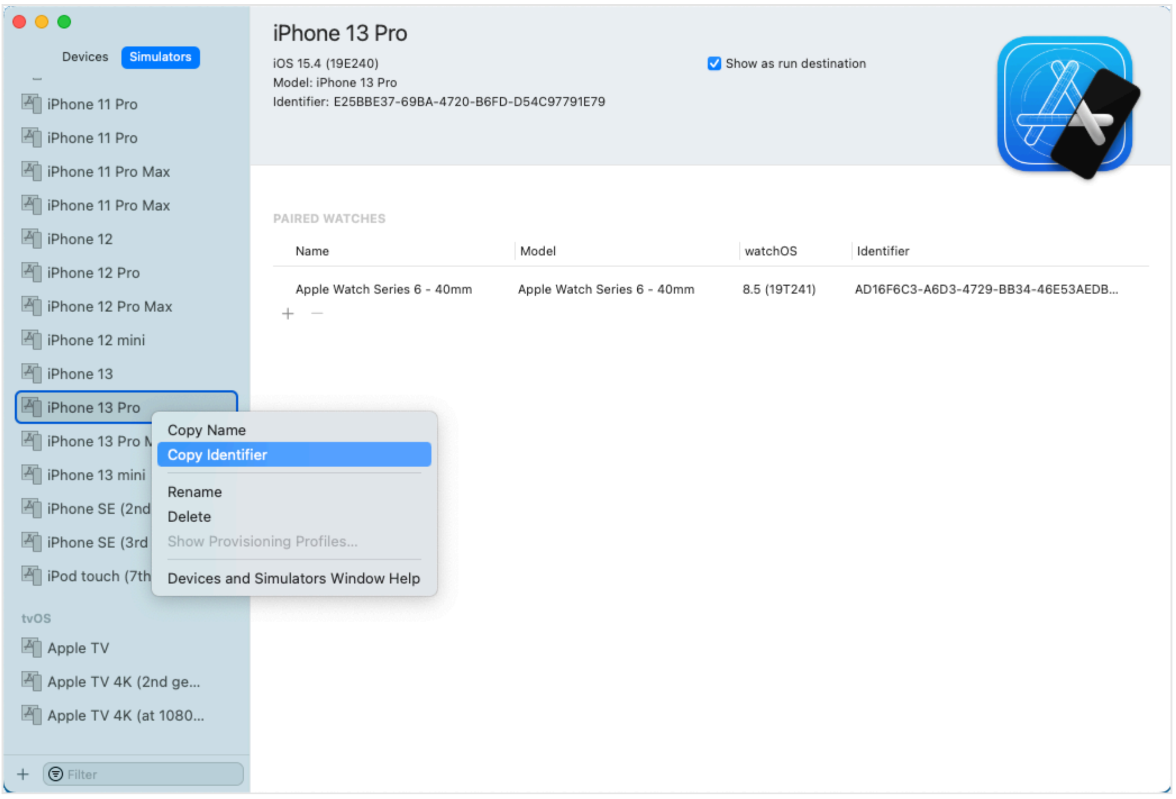
1. On your Mac, open **Xcode**, select the **Windows** > **Devices and Simulators** menu item, and then the **Simulators** tab.

   iPad (7th generation)
   iOS 13.5 (17F61)
   Model: iPad (7th generation)
   Identifier: D1218E0D-51EE-477B-820F-910EA0D065E5

   This simulator does not support paired watches.

2. Right-click on your chosen simulator, and select **Copy Identifier** to copy the UDID to the clipboard.



iPhone 13 Pro
iOS 15.4 (19E240)
Model: iPhone 13 Pro
Identifier: E25BBE37-69BA-4720-B6FD-D54C97791E79

PAIRED WATCHES

Name	Model	watchOS	Identifier
Apple Watch Series 6 - 40mm	Apple Watch Series 6 - 40mm	8.5 (19T241)	AD16F6C3-A6D3-4729-BB34-46E53AEDB...

Alternatively, you can retrieve a list of UDID values by executing the `simctl list` command:

```zsh
/Applications/Xcode.app/Contents/Developer/usr/bin/simctl list
```

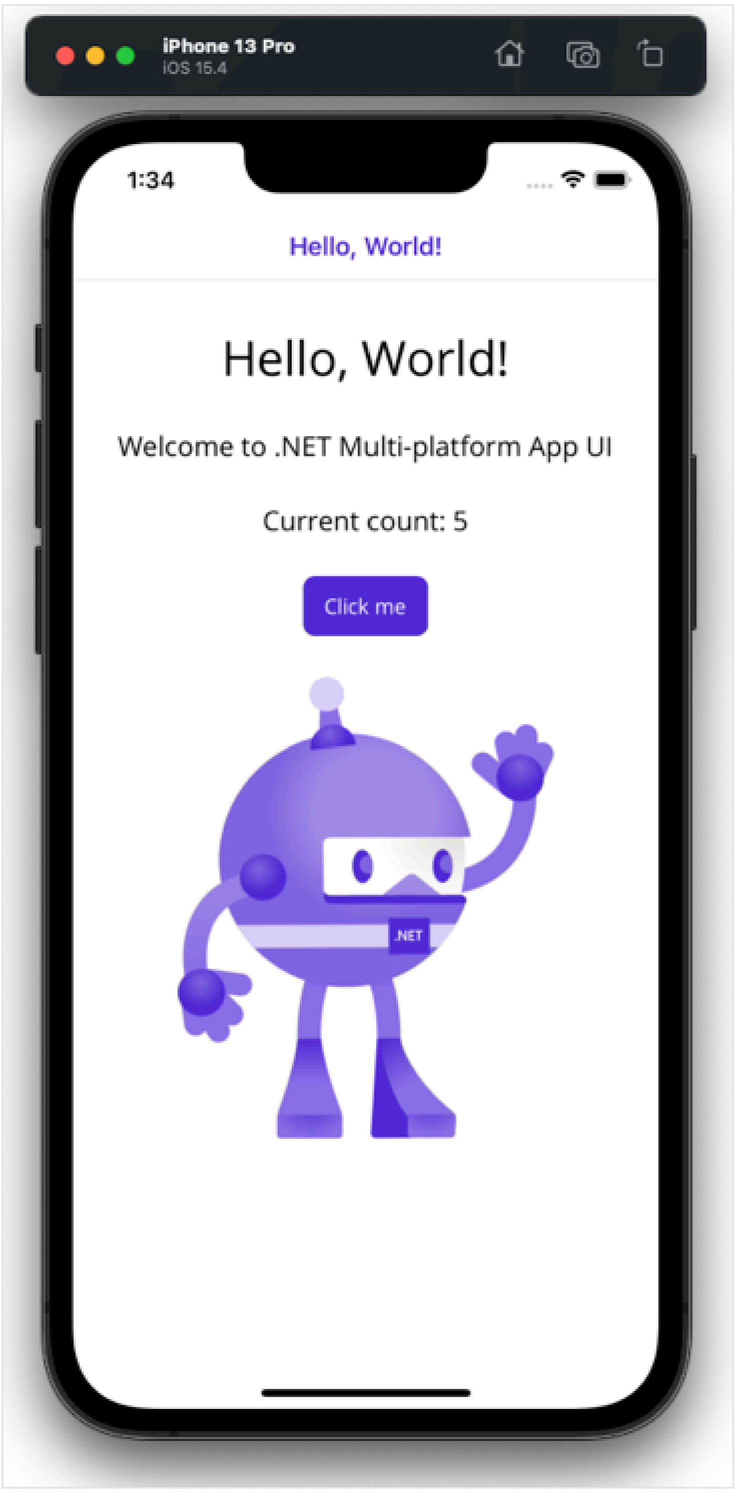
3. In **Terminal**, build the app and run it on your chosen simulator by specifying the `_DeviceName` MSBuild property using the `-p` MSBuild option:

```zsh
dotnet build -t:Run -f net8.0-ios -
p:_DeviceName=:v2:udid=MY_SPECIFIC_UDID
```

For example, use the following command to build the app and run it on the iPhone 13 Pro simulator:

```zsh
dotnet build -t:Run -f net8.0-ios -
p:_DeviceName=:v2:udid=E25BBE37-69BA-4720-B6FD-D54C97791E79
```

4. In your chosen simulator, press the **Click me** button several times and observe that the count of the number of button clicks is incremented.



Launch the app on a device

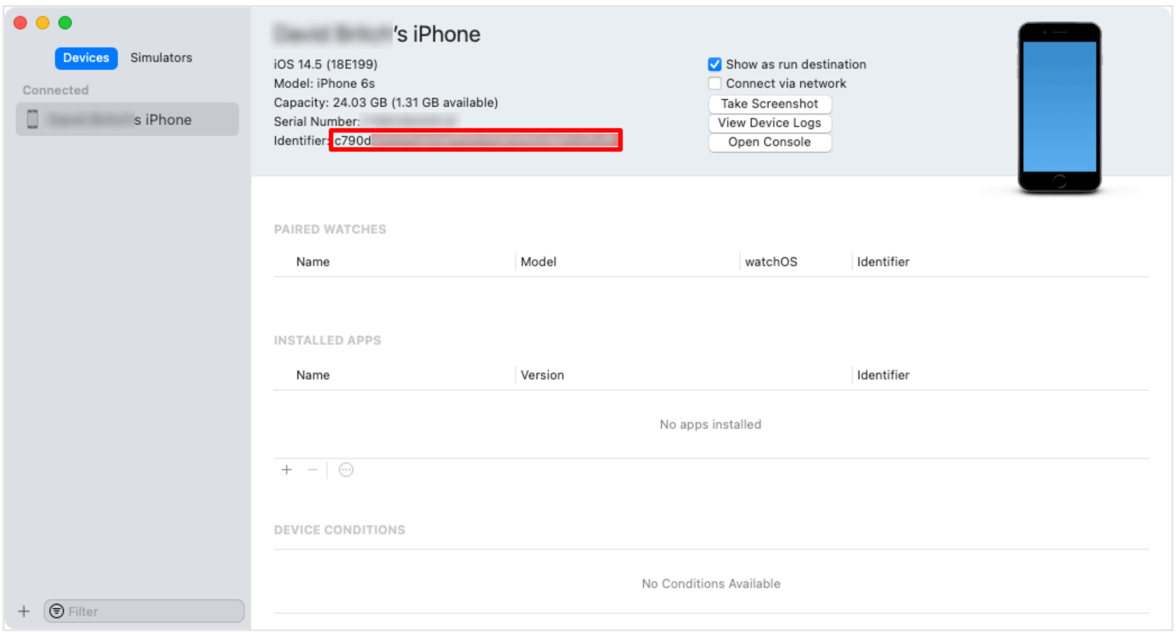
A device must be provisioned before you can deploy an iOS app to it. For more information, see Device provisioning for iOS. Once a device has been provisioned, a .NET MAUI iOS app can be launched on the device from a Mac by providing its unique device id (UDID):

1. Connect your device to your local Mac with a USB cable.

2. Open **Xcode**, and navigate to **Window** > **Devices and Simulators**.

3. In **Xcode**, select the **Devices** tab, and select the device from the list of connected devices.

4. In **Xcode**, copy the **Identifier** value to the clipboard:



[redacted]'s iPhone

iOS 14.5 (18E199)
Model: iPhone 6s
Capacity: 24.03 GB (1.31 GB available)
Serial Number: [redacted]
Identifier: c790d[redacted]

[✓] Show as run destination
[ ] Connect via network

PAIRED WATCHES

Name	Model	watchOS	Identifier

INSTALLED APPS

Name	Version	Identifier

No apps installed

DEVICE CONDITIONS

No Conditions Available

Alternatively, right-click on your device and select **Copy Identifier** to copy the UDID to the clipboard.

5. In **Terminal**, build the app and run it on your chosen device by specifying the `_DeviceName` MSBuild property using the `-p` MSBuild option:

zsh

```
dotnet build -t:Run -f net8.0-ios -p:RuntimeIdentifier=ios-arm64
-p:_DeviceName=MY_SPECIFIC_UDID
```

Replace "MY_SPECIFIC_UDID" with the device identifier you copied to the clipboard.