

Build a Mac Catalyst app with .NET CLI

Article • 03/19/2024

In this tutorial, you'll learn how to create and run a .NET Multi-platform App UI (.NET MAUI) app on Mac Catalyst 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-maccatalyst
```

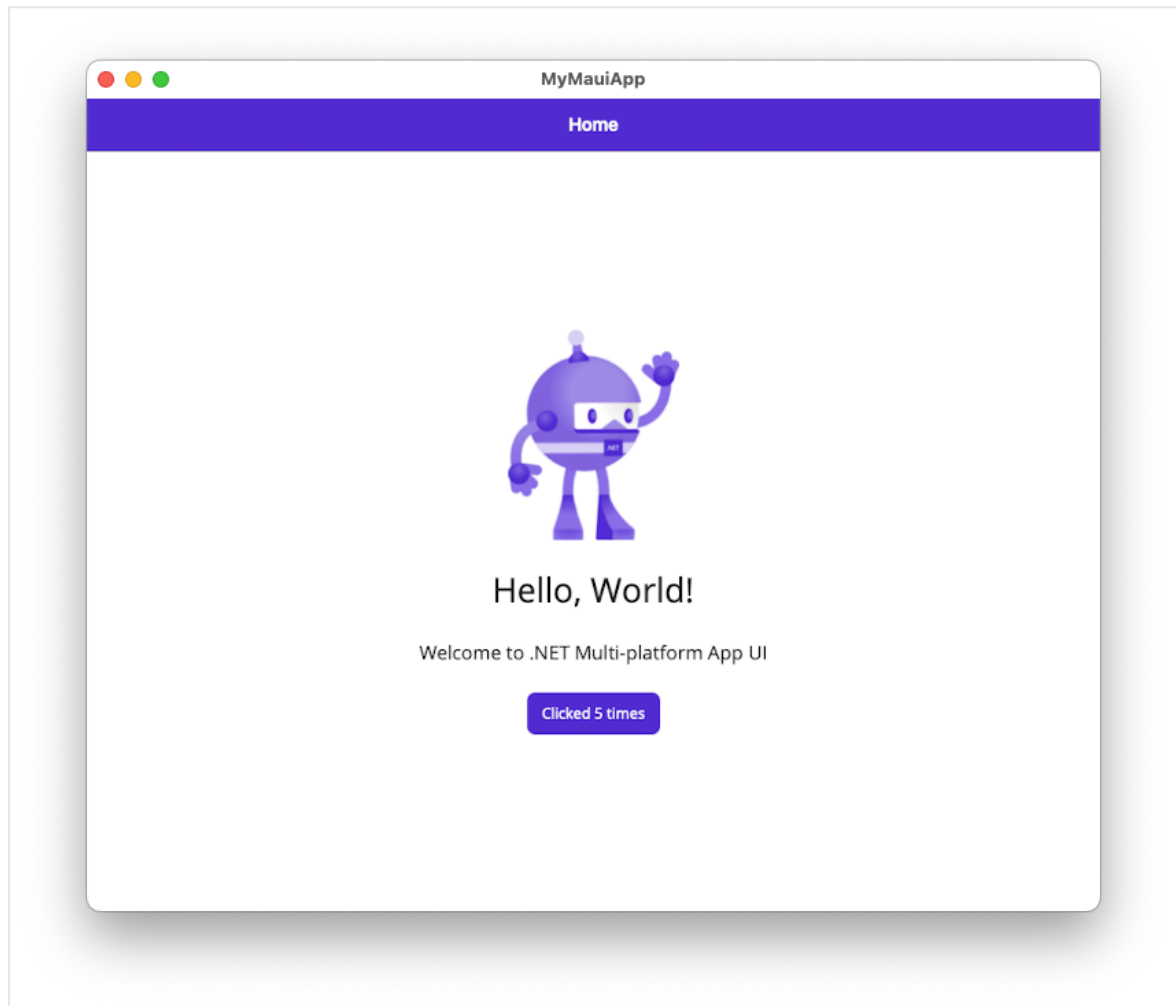
The `dotnet build` command will restore the project dependencies, build the app, and launch it.

If you see a build error and a warning that the Xcode app bundle could not be found, you may need to run the following command:

```
zsh
```

```
xcode-select --reset
```

6. In the running app, 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
```

```
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*.