

Mac Catalyst capabilities

Article • 08/30/2024

On Mac Catalyst .NET Multi-platform App UI (.NET MAUI) apps run in a sandbox that provides a set of rules that limit access between the app and system resources or user data. Apple provides *capabilities*, also known as *app services*, as a means of extending functionality and widening the scope of what Mac Catalyst apps can do. For more information about capabilities, see [Capabilities](#) on developer.apple.com.

Capabilities are added to your app's provisioning profile, and are used when code signing your app. The provisioning profile must contain an App ID, that matches your app's bundle identifier, with the required capabilities enabled. The provisioning profile should be created in your Apple Developer Account.

Capabilities are closely related to the concept of entitlements. They both request the expansion of the sandbox your app runs in, to give it additional capabilities. Entitlements are typically added when developing your app, while capabilities are typically added when code signing your app for distribution. For more information about entitlements, see [Entitlements](#).

Add capabilities in your Apple Developer Account

Capabilities can be added to your app's provisioning profile in your Apple Developer Account. This is a multi-step process that requires creating a signing certificate, App ID, and provisioning profile.

When adding a new capability to your provisioning profile, you should also add the same capability to your app's *Entitlements.plist* file and ensure that the app consumes this file. For more information, see [Entitlements](#). It may also be necessary to set privacy keys in *Info.plist*, for certain capabilities.

Create a signing certificate

Creating a provisioning profile requires you to have previously created a signing certificate. The type of signing certificate depends on the intended distribution channel for your app:

- If you intend to distribute your app via the Mac App Store, see [Create a distribution certificate](#) in [Publish a Mac Catalyst app for Mac App Store](#)

distribution.

- If you intend to distribute your app outside the Mac App Store, see [Create a developer ID application certificate](#) in [Publish a Mac Catalyst app for distribution outside the Mac App Store](#).
- If you intend to distribute your app to a limited number of users on registered devices, outside the Mac App Store, see [Create a development certificate](#) in [Publish a Mac Catalyst app for ad-hoc distribution](#).

Create an App ID with an app service

An App ID is similar to a reverse-DNS string, that uniquely identifies an app, and is required to identify the app that you are distributing. The App ID should be identical to the bundle identifier for your app.

Important

The bundle identifier for a .NET MAUI app is stored in the project file as the **Application ID** property. In Visual Studio, in **Solution Explorer** right-click on your .NET MAUI app project and select **Properties**. Then, navigate to the **MAUI Shared > General** tab. The **Application ID** field lists the bundle identifier.

When the value of the **Application ID** field is updated, the value of the **Bundle identifier** in the **Info.plist** will be automatically updated.

There are two types of App ID - explicit and wildcard. An explicit App ID is unique to a single app, and typically takes the form `com.domainname.myid`. An explicit App ID allows the installation of one app, with a matching bundle identifier, to a device. Explicit App IDs are required for apps that enable app-specific capabilities.

An explicit App ID can be created with the following steps:

1. In a web browser, go to the [Identifiers](#) section of your Apple Developer Account and click the + button.
2. In the **Register a new identifier** page, select **App IDs** and click the **Continue** button.
3. In the **Register a new identifier** page, select the **App** type and click the **Continue** button.
4. In the **Register an App ID** page, provide a **Description** and set the **Bundle ID** to **Explicit**. Then, enter an App ID in the format `com.domainname.myid`:

Register an App ID

BackContinue

Platform
iOS, macOS, tvOS, watchOS

App ID Prefix

(Team ID)

Description
MyMauiApp
You cannot use special characters such as @, &, *, ' , " , ~ , .

Bundle ID
☒ Explicit ☐ Wildcard
com.company.mymauiapp
We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).

- In the **Register an App ID** page, enable your required capabilities under the **Capabilities** and **App Services** tabs:

CapabilitiesApp Services

ENABLE	NAME	NOTES
<input checked="" type="checkbox"/>	Access WiFi Information ⓘ	
<input type="checkbox"/>	App Attest ⓘ	
<input type="checkbox"/>	App Groups ⓘ	
<input checked="" type="checkbox"/>	Apple Pay Payment Processing ⓘ	
<input type="checkbox"/>	Associated Domains ⓘ	
<input checked="" type="checkbox"/>	AutoFill Credential Provider ⓘ	
<input type="checkbox"/>	ClassKit ⓘ	

- In the **Register an App ID** page, click the **Continue** button.
- In the **Confirm your App ID** page, review the information and then click the **Register** button. Provided that your App ID successfully registers, you'll be returned to the Identifiers section of your Apple Developer Account.
- In the **Identifiers** page, click on the App ID you just created.
- In the **Edit your App ID Configuration** page, any of your enabled capabilities that require additional setup will have a **Configure** button:

Edit your App ID Configuration

[Remove](#)[Save](#)

Platform
iOS, macOS, tvOS, watchOS

App ID Prefix

(Team ID)

Description

MyMauiApp

You cannot use special characters such as @, &, *, ' , " , ~ , . .

Bundle ID
com.company.mymauiapp (explicit)

Capabilities

App Services

ENABLE	NAME	NOTES
<input checked="" type="checkbox"/>	Access WiFi Information ⓘ	
<input type="checkbox"/>	App Attest ⓘ	
<input type="checkbox"/>	App Groups ⓘ	<div>Configure</div>
<input checked="" type="checkbox"/>	Apple Pay Payment Processing ⓘ	<div>Configure</div> Enabled Merchant IDs (0)
<input type="checkbox"/>	Associated Domains ⓘ	
<input checked="" type="checkbox"/>	AutoFill Credential Provider ⓘ	
<input type="checkbox"/>	ClassKit ⓘ	

Click any **Configure** buttons to configure your enabled capabilities. For more information, see [Configure app capabilities](#) on developer.apple.com.

Create a provisioning profile

Once you've created an App ID you must create a provisioning profile for the App ID. The provisioning profile you create will also depend on the intended distribution channel for your app:

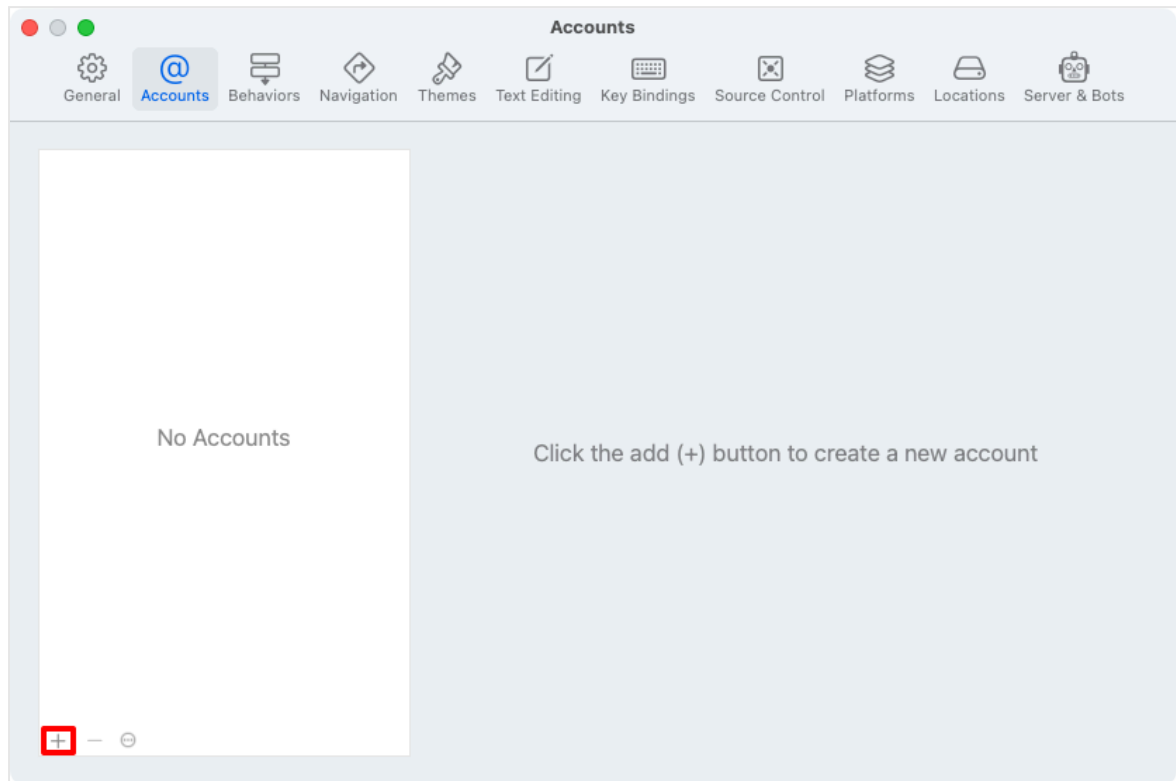
- If you intend to distribute your app via the Mac App Store, see [Create a provisioning profile](#) in [Publish a Mac Catalyst app for Mac App Store distribution](#).
- If you intend to distribute your app outside the Mac App Store, see [Create a provisioning profile](#) in [Publish a Mac Catalyst app for distribution outside the Mac App Store](#).
- If you intend to distribute your app to a limited number of users on registered devices, outside the Mac App Store, see [Create a provisioning profile](#) in [Publish a Mac Catalyst app for ad-hoc distribution](#).

Download your provisioning profile in Xcode

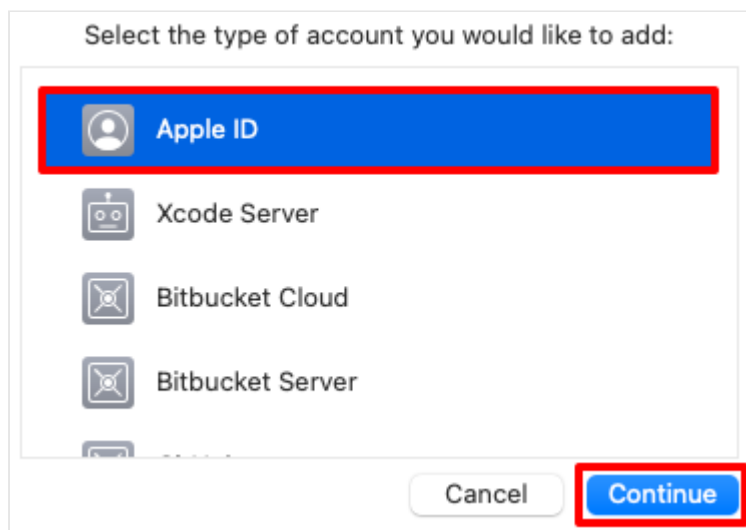
After creating a provisioning profile in your Apple Developer Account, Xcode can download it so that it's available for signing your app:

1. On your Mac, launch Xcode.
2. In Xcode, select the **Xcode > Preferences...** menu item.


3. In the **Preferences** dialog, select the **Accounts** tab.
4. In the **Accounts** tab, click the + button to add your Apple Developer Account to Xcode:



5. In the account type popup, select **Apple ID** and then click the **Continue** button:



6. In the sign in popup, enter your Apple ID and click the **Next** button.
7. In the sign in popup, enter your Apple ID password and click the **Next** button:



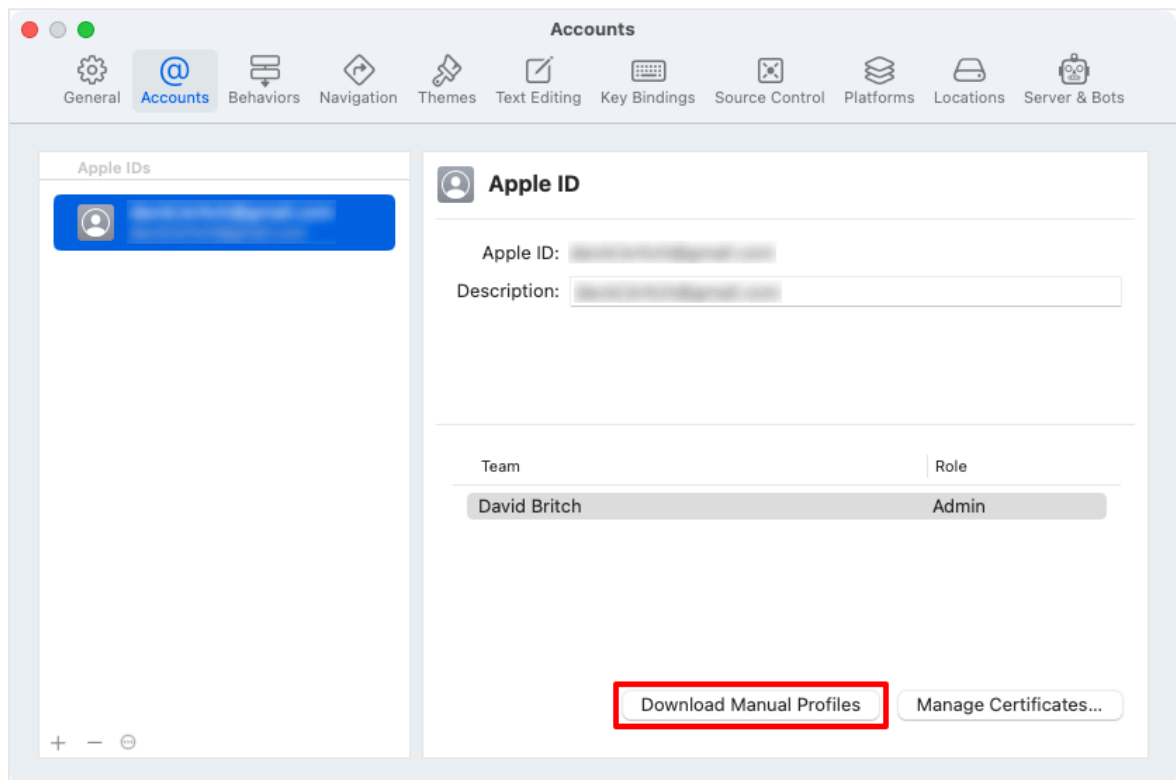
Sign in with your Apple ID.
 Don't have an Apple ID? You can create one for free.

Apple ID:

Password:

[Forgot Apple ID or Password?](#)

8. In the **Accounts** tab, click the **Manage Certificates...** button to ensure that your distribution certificate has been downloaded.
9. In the **Accounts** tab, click the **Download Manual Profiles** button to download your provisioning profiles:



10. Wait for the download to complete and then close Xcode.

Define build properties in your project file

The project file for your app should be updated to use the signing certificate, provisioning profile, and entitlements file. This can be achieved by adding the following build properties to a `<PropertyGroup>` in your project file:

Property	Value
<CodesignKey>	The name of the code signing key. Set to the name of your distribution certificate, as displayed in Keychain Access.
<CodesignEntitlements>	The path to the entitlements file that specifies the entitlements the app requires. Set to <code>Platforms\MacCatalyst\Entitlements.plist</code> .
<CodesignProvision>	The provisioning profile to use when signing the app bundle.
<EnableCodeSigning>	Set to <code>true</code> so that code signing is enabled.

The following example shows a typical property group for building and signing your Mac Catalyst app for Mac App Store distribution:

XML
<pre> <PropertyGroup Condition="'\$(Configuration) \$(TargetFramework) \$(Platform)'=='Release net8.0-maccatalyst AnyCPU'"> <EnableCodeSigning>True</EnableCodeSigning> <CodesignKey>Apple Distribution: John Smith (AY2GDE9QM7) </CodesignKey> <CodesignProvision>MyMauiApp</CodesignProvision> <CodesignEntitlements>Platforms\MacCatalyst\Entitlements.plist</CodesignEntitlements> </PropertyGroup> </pre>

Troubleshoot

The following list details the common issues that can cause issues when developing a .NET MAUI Mac Catalyst app that uses capabilities:

- Ensure that the correct App ID has been created and registered in the **Identifiers** section of your Apple Developer Account.
- Ensure that the capability has been added to the App ID and that the capability is configured using the correct values.
- Ensure that the provisioning profile has been installed on your development machine and that the app's *Info.plist* file is using a bundle identifier that's identical to your App ID.
- Ensure that the app's *Entitlements.plist* file has the correct entitlements enabled.
- Ensure that any required privacy keys are set in *Info.plist*.
- Ensure that the app consumes the *Entitlements.plist* file.