

# Troubleshooting

Troubleshooting is a necessary part of developing, testing, submitting, and releasing an app. The potential problems you may encounter are organized here in four general areas—certificates, provisioning, building, and debugging—with each problem followed by specific advice. Use this chapter as a reference to find solutions to problems you might encounter.

## Certificate Issues

Before you re-create a certificate that has become invalid or unusable, see whether it has one or more of the following common problems.

### No Code Signing Identities Found

Xcode detects when you're missing a signing identity. Typically, this happens when you move from one Mac to another. Follow the steps in [Creating the Team Provisioning Profile](#) to create your signing identity and add it to the team provisioning profile. You'll have the option of importing your signing identity from another Mac or resetting it. If you use a custom development provisioning profile that you manage yourself, it becomes invalid after revoking the development certificate. Read [Editing Provisioning Profiles in Your Developer Account](#) to regenerate it.

To avoid this problem, export your certificates as a developer profile file on the other Mac, and then import them on your new Mac, as described in [Exporting and Importing Certificates and Profiles](#).

### No Matching Provisioning Profiles Found

Xcode detects when you are missing a development provisioning profile and displays warning messages in both the General pane and the Capabilities pane. If a "No matching provisioning profiles found" message appears below the Team pop-up menu in the General pane, click the Fix Issue button below the message. If a "No Devices Registered" dialog appears, connect an eligible device to your Mac, as described in [Creating the Team Provisioning Profile](#), before clicking the Fix Issue button. If Xcode doesn't fix the issue, follow all the steps in [Configuring Identity and Team Settings](#) to verify your project configuration.

### The Private Key for a Developer ID Certificate Is Missing

Optionally, contact Apple at [product-security@apple.com](mailto:product-security@apple.com) if you need to revoke Developer ID certificates. Alternatively, you can continue to develop and distribute apps by creating additional Developer ID certificates, as described in [Creating Additional Developer ID Certificates](#).

### Your Certificates Are Invalid Because You're Missing an Intermediate Certificate

If your certificates are invalid, you could be missing the intermediate certificate used to authenticate your certificate. If you verify your certificate in Keychain Access, as described in [Verifying Using Keychain Access](#), and instead of a green circle with a checkmark, a red circle with a white X appears with the status "This certificate was signed by an unknown authority," you're missing the intermediate certificate. If you don't have a certificate called *Apple Worldwide Developer Relations Certification Authority* in your system keychain, read [Installing Missing Intermediate Certificate Authorities](#) to learn how to reinstall it. The intermediate certificate for Developer ID certificates is called the *Developer ID Certification Authority*.

## Your Certificates Have Trust Issues

If you view your certificate in Keychain Access, and a blue circle and white plus sign appear in the detail area instead of a green circle with a checkmark, your certificate has trust issues. To learn how to fix this problem, read [Xcode Doesn't Trust Your Certificate](#).

## Your Certificates Have Expired

You can't renew expired certificates. Read [Replacing Expired Certificates](#) for how to remove the expired certificates and create new ones.

**Mac Note:** If your Developer ID certificates expire, users can still download, install, and run versions of your Mac apps that were signed with these certificates. However, you'll need new Developer ID certificates to sign updates and create new apps.

## Your Request Has Timed Out

If a dialog appears stating that your request has timed out, the Certificates, Identifiers & Profiles section of your developer account is probably down for maintenance. Try again later.

## Your Provisioning Profile or Signing Identity Doesn't Appear in Xcode Menus

Occasionally, your provisioning profile or signing identity doesn't appear in a Provisioning Profile or a pop-up menu when distributing your app using the Devices window or when setting the Code Signing Identity build setting in the project editor. When this occurs, download provisioning profiles in Xcode, as described in [Downloading Provisioning Profiles in Xcode](#).

## Provisioning Issues

Common provisioning issues result from using the wrong provisioning profile with your app or using an invalid or expired provisioning profile.

### Provisioning Profiles Installed on Your Device Are Invalid

If the provisioning profile on your development device has expired or is invalid, remove it as described in [Verifying and Removing Provisioning Profiles on Devices](#).

### Provisioning Profiles Appear Invalid in Your Developer Account

A provisioning profile appears invalid if it contains a revoked or expired certificate. A provisioning profile also appears invalid if its App ID entitlements change or its App ID is deleted. A development or ad hoc provisioning profile is invalid if it contains a disabled device.

If a team provisioning profile appears invalid in your developer account, download provisioning profiles in Xcode, as described in [Downloading Provisioning Profiles in Xcode](#). For example, if you revoke your development certificate, use Xcode to regenerate the associated team provisioning profiles.

If a provisioning profile you manage appears invalid in your developer account, remove or edit it, as described in [Removing Provisioning Profiles from Your Developer Account](#) and [Editing Provisioning Profiles in Your Developer Account](#).

## No Such Provisioning Profile Was Found

If a “Your build settings specify a provisioning profile with [...] however, no such provisioning profile was found.” warning or similar message appears below the Team pop-up menu in the General pane, you may have incorrect Provisioning Profile and Code Signing Identity build settings from using an earlier version of Xcode.

When this occurs, click the Fix Issue button below the warning message. If necessary, click the Fix Issue button again. The Provisioning Profile build setting should be None and the Code Signing Identity build setting should be Automatic and iOS Developer for iOS, tvOS, and watchOS apps, and Automatic and Mac Developer for Mac apps. If you have an older project and want to use provisioning profiles managed by Xcode, read [Migrating to Xcode-Managed Provisioning Profiles](#).

If you’re using a custom development provisioning profile, read [Using Custom Provisioning Profiles](#) for how to configure your project.

## Build and Code Signing Issues

If you use the team provisioning profile that Xcode manages for you during development, as described in [Team Provisioning Profiles in Depth](#), Xcode fixes code signing and provisioning issues for you before you attempt to build your app. In this case, you shouldn’t set the Code Signing Identity build settings yourself. However, if you want to use a custom development provisioning profile and set these build settings, as described in [Using Custom Provisioning Profiles](#), you may encounter build issues described in this section. Common build errors tend to involve incorrect code signing identities.

### Xcode Can’t Find Your Provisioning Profile

You’ll receive the following error message after replacing a provisioning profile with a modified version, such as when a provisioning profile’s App ID changes:

```
Code Sign error: Provisioning Profile 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx' can't be found
```

To address this error, ensure that the correct provisioning profile and code signing identity are selected for the value of the Code Signing Identity build setting. This error may also occur if the project’s and target’s Code Signing Identity build settings are different.

### Xcode Doesn’t Trust Your Certificate

You get this error message when Xcode can’t verify the authenticity of your development or distribution certificate:

```
Code Sign error: CSSMERR_TP_NOT_TRUSTED
```

If the trust setting isn’t Use System Defaults, you’ll receive a `CSSMERR_TP_TRUSTED` error message from the `codesign(1)` command-line utility when you build and run your app. Don’t change the trust settings of your certificates from the default Use System Defaults. Follow these steps to repair your development, distribution, and intermediate certificates.

- 
- ▶ **To set the trust level of a certificate to the system defaults**
- 

## The Code Signing Identity Build Setting Doesn’t Match Any Certificates

You’ll receive the following error message when your certificate has expired or is otherwise invalid:

```
Code Signing Identity 'iPhone Developer' doesn't match any valid, non-expired,
certificate/private key pair in your keychain.
```

For multiple targets that use the same code signing identity, set this build setting at the project level, not the target. However, if it's set on both the project and the target, the target setting overrides the project setting. If the target's Code Signing Identity build setting is set, delete it by first selecting it and then choosing Edit > Delete. Finally, set the project's Code Signing Identity build setting to your certificate.

If your development certificate or team provisioning profile doesn't appear in the Code Signing Identity pop-up menu, try downloading the provisioning profiles, as described in [Downloading Provisioning Profiles in Xcode](#). If you are using an Xcode-managed provisioning profile, read [Migrating to Xcode-Managed Provisioning Profiles](#). If you are using a custom provisioning profile, read [Using Custom Provisioning Profiles](#).

## Your Keychain Contains Duplicate Code Signing Identities

You get one of these error messages when there are duplicate code signing identities in your keychain, such as two development identities or two distribution identities (your keychain must contain at most one code signing identity of each type):

```
Build error "iPhone Developer: <your_name> (XYZ123ABC): ambiguous (matches "iPhone
Developer: <your_name> (XYZ123ABC)" in /Library/Keychains/System.keychain and
"IPhone Developer: <your_name> (XYZ123ABC)" in
/Users/../../Library/Keychains/login.keychain)"
```

```
[BEROR]CodeSign error: Certificate identity 'iPhone Distribution: <your_name>'
appears more than once in the keychain. The codesign tool requires there only be
one.
```

To address these errors, try deleting the duplicate code signing identities from your keychain, as described in [Removing Signing Identities from Your Keychain](#).

## The App ID in Your Provisioning Profile Doesn't Match Your App's Bundle Identifier

When there's a conflict between the [App ID](#) in the provisioning profile selected in the Code Signing Identity build setting and your app's bundle identifier, you get error messages similar to the following:

```
Code Sign error: Provisioning profile 'MyApp Profile' specifies the Application
Identifier 'com.mycompany.MyApp.*' which doesn't match the current setting
'com.mycompany.MyApp'
```

To address these errors, ensure that your bundle identifier is set correctly in your Xcode project, that the certificate and provisioning profile specified in the Code Signing Identity build setting is correct, and that the provisioning profile uses the correct App ID.

## Your Device Isn't Listed or Is Ineligible in the Scheme Toolbar Menu

If you have a project or workspace open and your connected device isn't listed under Ineligible Devices in the Scheme toolbar menu, hover the mouse over the device to read the reason why it's ineligible. Verify that:

1. The app's targeted iOS version is equal to or greater than the iOS version installed on your device.

See [Setting the Target Devices \(iOS, watchOS\)](#) for details.

2. The version number of the iOS SDK your project uses is equal to or greater than the version number of the iOS version on your device.

For example, if Xcode shows iOS SDK 4.3 but your device has iOS 5.0 installed, you need to install on your Mac an Xcode version that includes iOS SDK 5.0.

## Debugging Information Issue

The most common potential problem with debugging is that Xcode hasn't yet collected the information from your device.

### Xcode Displays the Unknown iOS Detected Dialog When You Connect a Device

This message appears when Xcode hasn't seen a particular version of iOS before and needs to download (from the device) the debugging symbols for that version from the device. To successfully debug apps on the device, leave the device connected until Xcode finishes downloading the debugging symbols from the device.

## Archiving and Uploading Issue

If you select an archive in the Archives window and the Validate, Export, or "Upload to App Store" buttons are disabled, verify that the archive contains a single top-level app.

---

Copyright © 2016 Apple Inc. All Rights Reserved. Terms of Use | Privacy Policy | Updated: 2016-04-29