Distributing Apps Outside the Mac App Store

In some cases, you may want to distribute an app outside the Mac App Store. Because that app won't be distributed through the Mac App Store, use a Developer ID certificate to give your users assurance that you're an Apple-identified developer.

Mac users have the option of turning on Gatekeeper, a security feature that gives users the ability to install software only from the Mac App Store and identified developers. If your app isn't signed with a Developer ID certificate issued by Apple, it won't launch on a Mac that has Gatekeeper enabled. To avoid this situation, sign your apps and installer packages using a Developer ID certificate. Also, thoroughly test the end-user experience using a Gatekeeper-enabled Mac before distributing your app outside of the Mac App Store.

This chapter describes the Xcode steps to create and test Developer ID-signed apps for distribution outside the Mac App Store.

Creating Developer ID-Signed Apps or Installer Packages

Creating a Developer ID-signed app or installer package is a multistep process. First you tell Xcode that you intend to distribute your app outside the Mac App Store and then create Developer ID certificates. There are two types of Developer ID certificates: a Developer ID Application is used to sign apps, and a Developer ID Installer is used to sign installer packages. Using Xcode, you export and sign an archive of your app using the Developer ID Application certificate. You can also use command-line utilities to sign an installer package using the Developer ID Installer certificate.

> **Important:** Before you begin, enroll in the Apple Developer Program or the Apple Developer Enterprise Program, as described in Adding Your Apple ID Account in Xcode. Only team agents belonging to either the Apple Developer Program or the Apple Developer Enterprise Program are allowed to create Developer ID certificates and sign apps or installer packages using them.

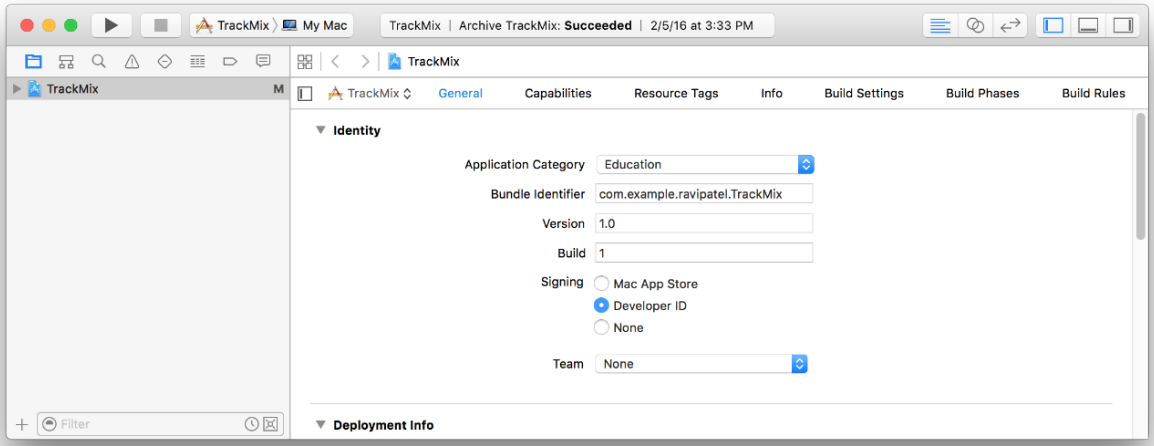
Setting the Signing Identity to Developer ID

First set the signing identity in the General pane to Developer ID.

To set the signing identity to Developer ID

1. In the project navigator, select the target to display the project editor.
2. Click General and, if necessary, click the disclosure triangle next to Identity to reveal the settings.
3. Verify that your bundle ID is unique.
4. If necessary, choose your team or "Add an Account" from the Team pop-up menu.

   The team must be a member of the Apple Developer Program.
5. Under Signing, select "Developer ID" as the signing identity.



6. If necessary, click Fix Issue under the Team pop-up menu.

You can't use some app services if you distribute outside the Mac App Store. If you enable a capability in the Capabilities pane, as described in Adding Capabilities, the Signing radio button reverts to Mac App Store. To add your Apple ID to Xcode and join the Apple Developer Program, read Managing Accounts.

Creating Developer ID Certificates

You use signing certificates that begin with the text "Developer ID" to distribute your app outside the Mac App Store. Use Accounts preferences to specifically create Developer ID certificates, described in Creating Signing Identities. In Accounts preferences, click Create next to the Developer ID Application or Developer ID Installer signing identity. If you have a Developer ID certificate but are missing the private key, create another Developer ID certificate, described in Creating Additional Developer ID Certificates. To use these certificates, you also need the Developer ID Certification Authority intermediate certificate that Xcode installs in your keychain for you to use these certificates. If you're missing this intermediate certificate, read Installing Missing Intermediate Certificate Authorities to restore it.

You should immediately back up your Developer ID signing identities after creating them, as described in Exporting Your Developer Profile.

> **Important:** Only a team agent can create Developer ID certificates. If you're an individual developer, you're the team agent and can create these certificates. Contact `product-security@apple.com` if you want to revoke Developer ID certificates.

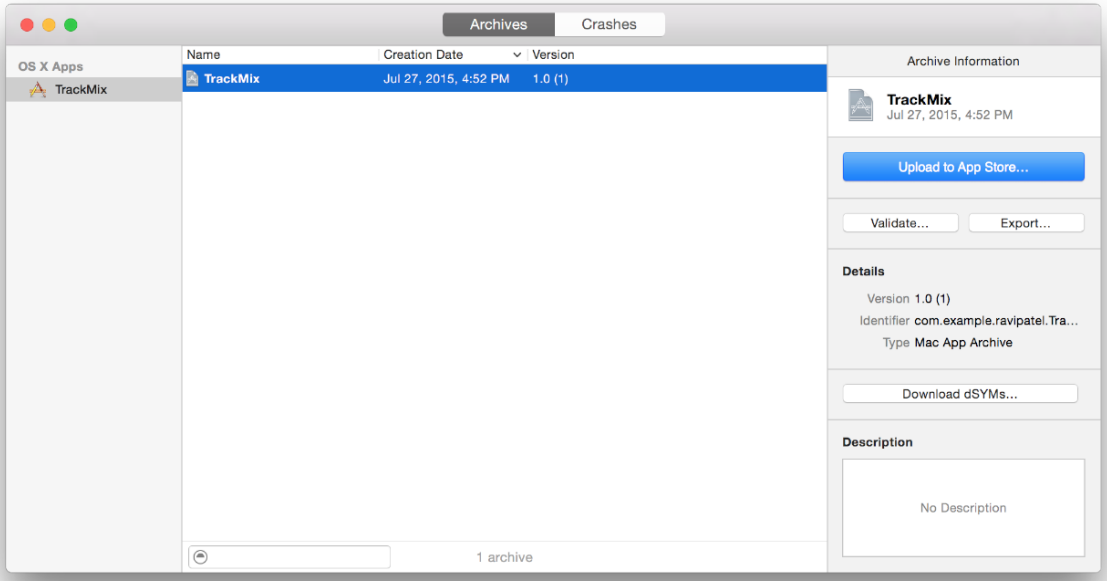
Creating an Archive

Before creating the archive, build and run your app one more time to ensure that it's the version you want to distribute.

To create an archive

1. In the Xcode project editor, select the project.
2. Choose Product > Archive.

   The Archives organizer appears and displays the new archive.



Validating a Developer ID-Signed App

Immediately after creating the archive, validate it and fix any validation errors before continuing.

To validate a Developer ID-signed archive

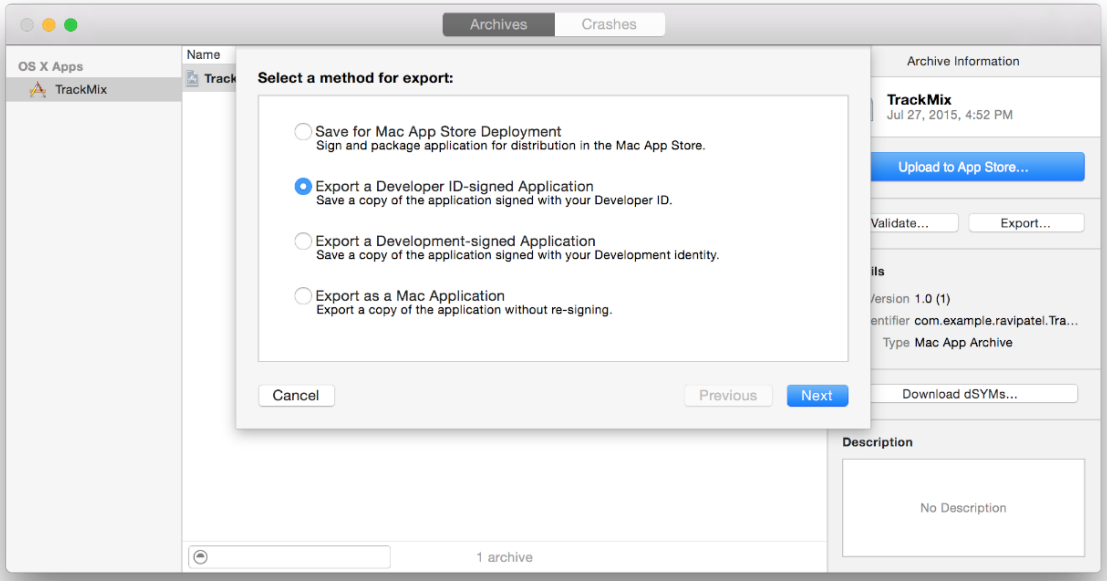
1. In the Archives organizer, select the archive and click the Validate button.
2. In the dialog that appears, select “Validate a Developer ID-signed Application” as the validation method and click Next.
3. In the dialog that appears, choose a team from the pop-up menu and click Choose.
4. Review the signing identity and entitlements, and click Validate.
5. Review validation issues found, if any, and click Done.

Exporting a Developer ID-Signed app

To export your app for distribution outside the Mac App Store, use the Archives organizer.

To create a Developer ID-signed app

1. In the Archives organizer, select the archive and click Export.
2. In the dialog that appears offering a choice of distribution methods, select “Export a Developer ID-signed Application,” and click Next.



3. In the dialog that appears, choose a team from the pop-up menu and click Choose.

   If you’re an individual developer, your name appears in the pop-up menu; otherwise, your organization name appears in the pop-up menu.

4. In the dialog that appears, review the signing identity and entitlements, and click Export.

   The Finder shows the exported file.

Signing an Installer Package

If you want to distribute your app outside the Mac App Store as part of an installer package, create the package as you normally do. Code sign the package with your Developer ID Installer certificate with the productsign(1) command-line utility. To test your installer package, use the spctl(8) command-line utility and replace `MyPackageName.pkg` with the filename of your package:

```
spctl -a -v --type install MyPackageName.pkg
```

> **Warning:** Make sure you sign the installer package using your Developer ID Installer certificate. The productsign(1) command-line utility allows you to sign an installer package using your Developer ID Application certificate. Although this approach may appear to work, the resulting installer archive will fail on the destination Mac.

If your development process includes code signing from the command line, read *Code Signing Guide*.

Verifying Your Steps

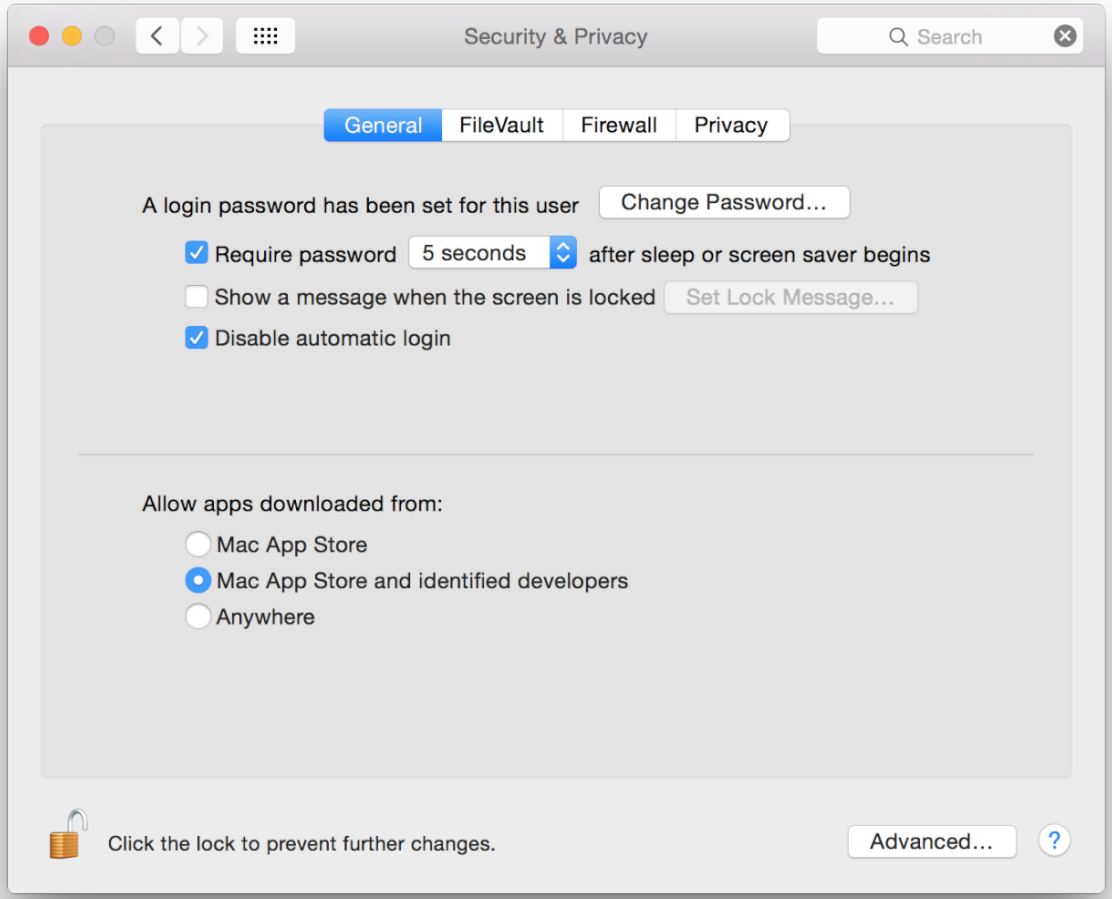
Before you distribute your app, test the end-user experience by launching your app with Gatekeeper enabled and disabled. You can enable and disable Gatekeeper using System Preferences. Use the spctl(8) command-line utility for verifying and testing Gatekeeper too. To simulate the end-user experience, you need to quarantine your app and test it again with Gatekeeper enabled.

Enabling and Disabling Gatekeeper

You turn on and off Gatekeeper by using the Security & Privacy preferences in System Preferences. You can turn off Gatekeeper and verify the status of Gatekeeper using the spctl(8) command-line utility.

To enable or disable Gatekeeper using the Security & Privacy preferences

1. In the Finder, launch System Preferences and select Security & Privacy.
2. Click the lock button if it appears locked, and enter the administrator password.
3. To enable Gatekeeper, select “Mac App Store and identified developers.”



4. To disable Gatekeeper, select Mac App Store or Anywhere and in the dialog that appears, confirm your selection.

To disable Gatekeeper using the spctl command

1. In Terminal, enter the following command:

```
$ sudo spctl --master-disable
```

2. Press Return.
3. When prompted, enter your administrator password.

To confirm that Gatekeeper is enabled using the spctl command

1. In Terminal, enter the following command:

```
$ spctl --status
```

2. Press Return.

   If Gatekeeper is enabled, the output of this command is:

```
assessments enabled
```

If Gatekeeper is disabled, the output of this command is:

```
assessments disabled
```

Testing Gatekeeper Behavior

After signing your app with a Developer ID certificate, you can test whether it was signed correctly and simulate the launch behavior of your app when Gatekeeper is enabled. On a Mac with Gatekeeper enabled, a quarantined copy of your app launches only if it's Developer ID signed. (Learn about quarantine in this Knowledge Base article.) You can also test the behavior of Gatekeeper for an app that isn't Developer ID signed.

Testing a Developer ID–Signed App

You can use the spctl(8) command-line utility to test whether your app is signed correctly using a Developer ID certificate.

To test your Developer ID–signed app

1. Enable Gatekeeper on your test Mac by selecting "Mac App Store and identified developers" in the Security & Privacy preferences in System Preferences.
2. Enter the following command in Terminal by replacing `TrackMix.app` with the path to your app.

   ```
   $ spctl -a -v TrackMix.app
   ```

3. Press Return.

   If the app is correctly signed, the output of this command is:

   ```
   ./TrackMix.app: accepted
   source=Developer ID
   ```

Testing the Launch Behavior

To thoroughly test your Developer ID–signed app, simulate launching the app on a Mac not used for development.

To prepare for testing Gatekeeper behavior

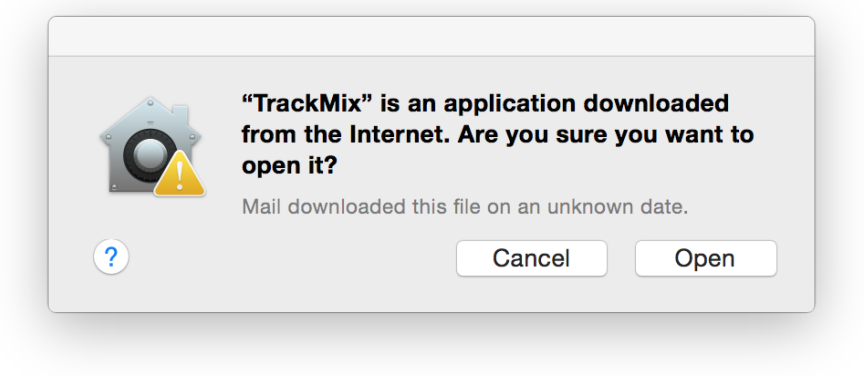
1. Enable Gatekeeper on your test Mac, as described in Enabling and Disabling Gatekeeper.
2. Quarantine a copy of your Developer ID–signed app. You can do this in either of the following ways:
   - Email your Developer ID–signed app to yourself and use the copy that Mail downloads.
   - Host your Developer ID–signed app on your own local or remote server and use the copy that Safari downloads.

You're ready to test Gatekeeper behavior.

To test Gatekeeper behavior for your Developer ID–signed app

1. In the Finder, locate the quarantined copy of your Developer ID–signed app and double-click its icon.

   The Mac displays an alert asking whether you're sure you want to open the app.



This alert, which allows you to open the quarantined app with Gatekeeper enabled, confirms that your Developer ID app is built correctly.

> **Important:** If an alert doesn't appear at this point, it's likely that you have opened a nonquarantined copy of your app.

To test Gatekeeper behavior for blocking apps that aren't Developer ID signed

1. Enable Gatekeeper on your test Mac, as described in Enabling and Disabling Gatekeeper.
2. Quarantine a copy of your app that isn't Developer ID signed.

   As before, you can invoke quarantine on this copy of your app in either of the following ways:

   - Email your app to yourself and use the copy that Mail.app downloads.
   - Host your Developer ID–signed app on your own local or remote server and use the copy that Safari downloads.

3. In the Finder, locate the quarantined copy of your non–Developer ID–signed app and double-click its icon.

   The Mac displays an alert that blocks you from opening the app. By way of this alert, Gatekeeper protects a Mac by preventing first-time opening of apps from unidentified developers. Apps previously opened by a user are no longer quarantined, and Gatekeeper doesn't prevent them from launching.

Recap

In this chapter, you learned how to distribute your Mac app outside the Mac App Store so that users won't block your app from launching.