

# Developer ID Tutorial

# Contents

## About Developer ID 3

At a Glance 3

Prerequisites 4

See Also 4

## Developer ID Tutorial 5

Obtaining Developer ID Certificates 5

Code Signing with a Developer ID Certificate 9

Code Signing an Application 9

Code Signing an Installer Package 12

Enabling and Testing Gatekeeper 12

Enabling and Disabling Gatekeeper 13

Testing Gatekeeper Behavior 14

## Document Revision History 17

# About Developer ID

**Important** This is a preliminary document for an API or technology in development. Although this document has been reviewed for technical accuracy, it is not final. Apple is supplying this information to help you plan for the adoption of the technologies and programming interfaces described herein. This information is subject to change, and software implemented according to this document should be tested with final operating system software and final documentation. Newer versions of this document may be provided with future seeds of the API or technology.

With Developer ID, you can distribute your Mac OS X applications outside of the Mac App Store in a way that provides users with greater assurance. Use Xcode 4.3 or newer to obtain your Developer ID certificates, code sign your application, and prepare it for distribution.

Developer ID is intended specifically for developers distributing their applications outside of the Mac App Store, providing greater flexibility in the options you can give users on where they obtain your software products.

- Use Mac App Store certificates for submission to the Mac App Store.
- Use Developer ID certificates for other distribution.

Each category of certificate can be used only for its intended purpose.

Mac OS X users will soon have the option of turning on Gatekeeper, a new Mac OS X security feature that complements Developer ID by giving users the ability to choose to install software only from identified developers. If you distribute your applications outside of the Mac App Store, adopt Developer ID now, as explained in this document.

## At a Glance

After you obtain Developer ID certificates and private keys using Xcode, you can code sign your applications for distribution outside of the Mac App Store. Xcode 4.3 or newer supports this entire workflow.

---

**Relevant sections** [“Obtaining Developer ID Certificates”](#) (page 5), [“Code Signing with a Developer ID Certificate”](#) (page 9)

---

To test Developer ID, start by enabling Gatekeeper by using the `spctl` command-line tool. Then provide a copy of your application that has been quarantined by way of downloading it. (Learn about quarantine in this [Knowledge Base article](#).)

---

**Relevant section** [“Enabling and Testing Gatekeeper”](#) (page 12)

---

## Prerequisites

To use the information in this document, it helps to be familiar with the basics of code signing as described in [“Code Signing Your App”](#) in *Tools Workflow Guide for Mac*.

## See Also

Read [Xcode 4 User Guide](#) to gain familiarity with Apple’s integrated development environment, Xcode.

If your development workflow includes code signing from the command line, read [Code Signing Guide](#). Read the [codesign](#), [productsign](#), and [packagemaker](#) man pages for detailed information about code signing and about creating a code signed installer package using the command line.

# Developer ID Tutorial

**Important** This is a preliminary document for an API or technology in development. Although this document has been reviewed for technical accuracy, it is not final. Apple is supplying this information to help you plan for the adoption of the technologies and programming interfaces described herein. This information is subject to change, and software implemented according to this document should be tested with final operating system software and final documentation. Newer versions of this document may be provided with future seeds of the API or technology.

If you choose to distribute an application outside of the Mac App Store, use Developer ID certificates to give your users assurance that you are identified by Apple.

As a member of the Mac Developer Program, you are immediately eligible to obtain and use Developer ID code signing certificates. For most developers, the entire Developer ID workflow takes place within Xcode. This document describes the Xcode workflow and provides links to more information for developers who use the command line for signing their applications or preparing them for distribution.

## Obtaining Developer ID Certificates

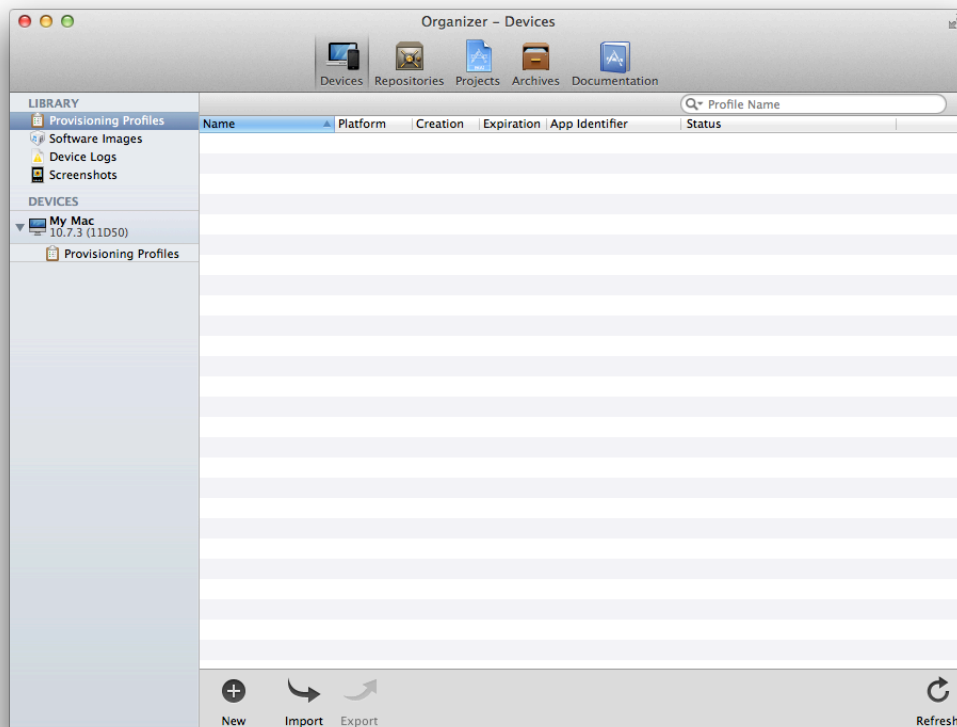
Use the Xcode Organizer window to obtain your Developer ID application and installer certificates, as well as the Developer ID Certificate Authority intermediate certificate.

Only a team agent can request Developer ID certificates. If you are an individual developer, you are your own team agent and can request the certificates directly.

### To obtain your Developer ID certificates and export them for safekeeping . . .

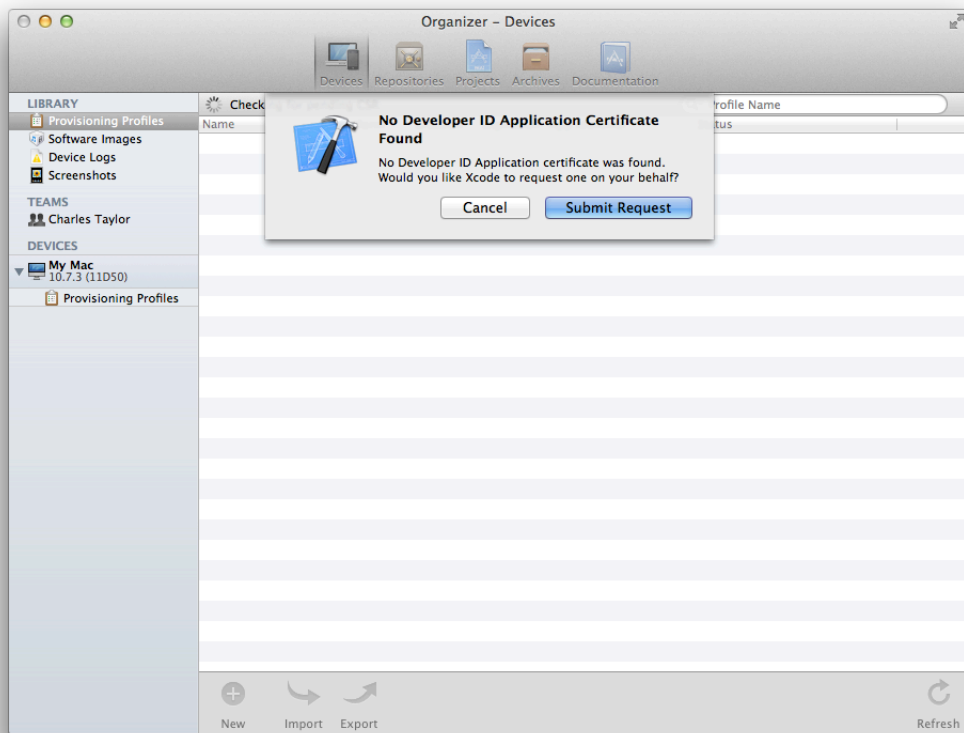
1. In Xcode, choose Window > Organizer.
2. Click Devices in the Organizer window's toolbar.
3. In the Library section in the sidebar, select Provisioning Profiles.

4. Click Refresh at the bottom-right of the window.



5. In the dialog that appears, enter your Mac Developer Program user name and password and click Log in.

After you log in, Xcode refreshes your certificates and displays another dialog asking if Xcode should request your Developer ID certificates.



(If you have just joined the Mac Developer Program, you see instead a dialog asking if Xcode should request your Mac development certificate.)

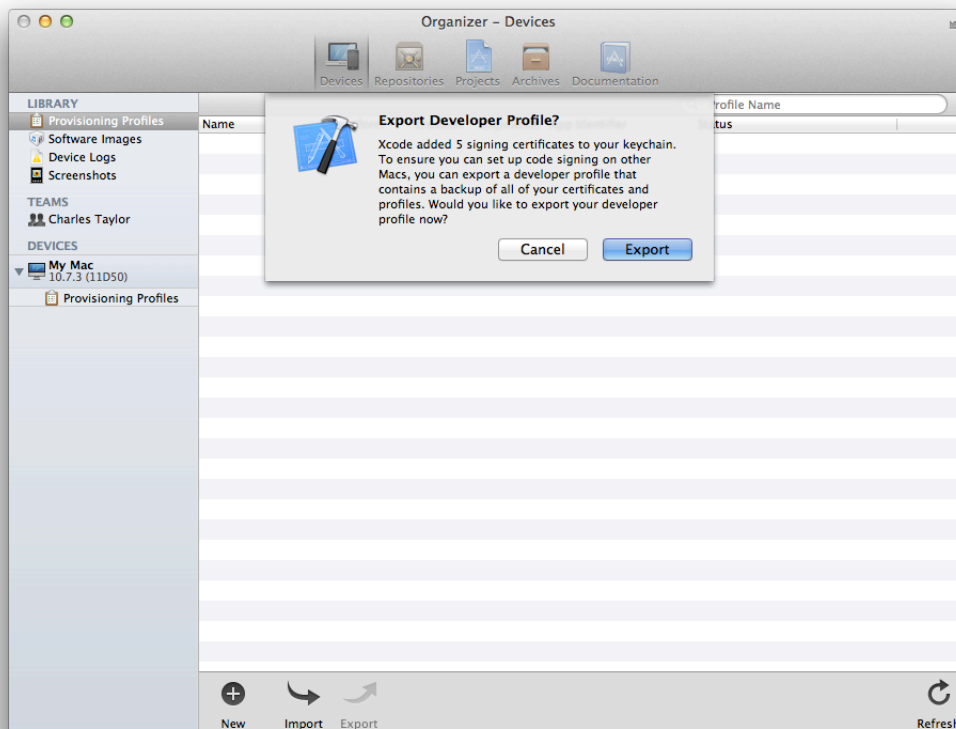
6. Click Submit Request.

The requested certificates and private keys are added to your default keychain.

Additional dialogs may appear asking if Xcode should request other certificates. Click Submit Request for every one of these dialogs.

After you submit the last certificate request, allow the refresh process to complete. When you receive your Developer ID certificates, the Developer ID Certificate Authority intermediate certificate is also added to your keychain.

7. A dialog offers to export your developer profile; click Export.



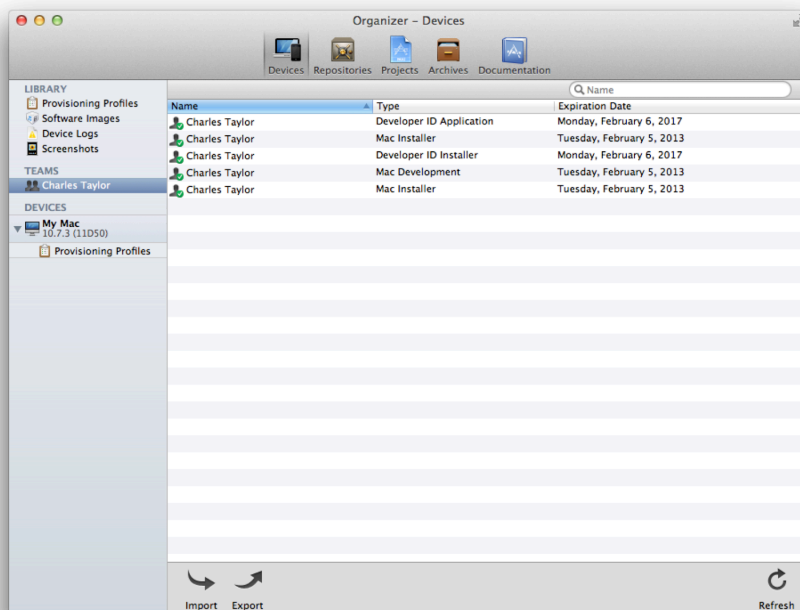
8. Enter a filename and a strong password, and click Save.

**Important** Your Developer ID private keys are valuable and you should back them up. Exporting your developer profile lets you create a password-protected backup. Save that backup as you would any essential backup; for example, save it to a different disk. Later, if you need to replace a private key, import it from your backup.

Your Developer ID Certificate Authority intermediate certificate, which is required for Developer ID code signing, is not exported. If you need to obtain another copy, retrieve it from Apple at <https://developer.apple.com/certificationauthority/DeveloperIDCA.cer>.



You can view your code signing certificates in the Devices pane of the Xcode organizer window.



## Code Signing with a Developer ID Certificate

Your Developer ID certificates are specifically for distributing your applications outside of the Mac App Store. If you are also submitting to the Mac App Store, read [Tools Workflow Guide for Mac](#).

### Code Signing an Application

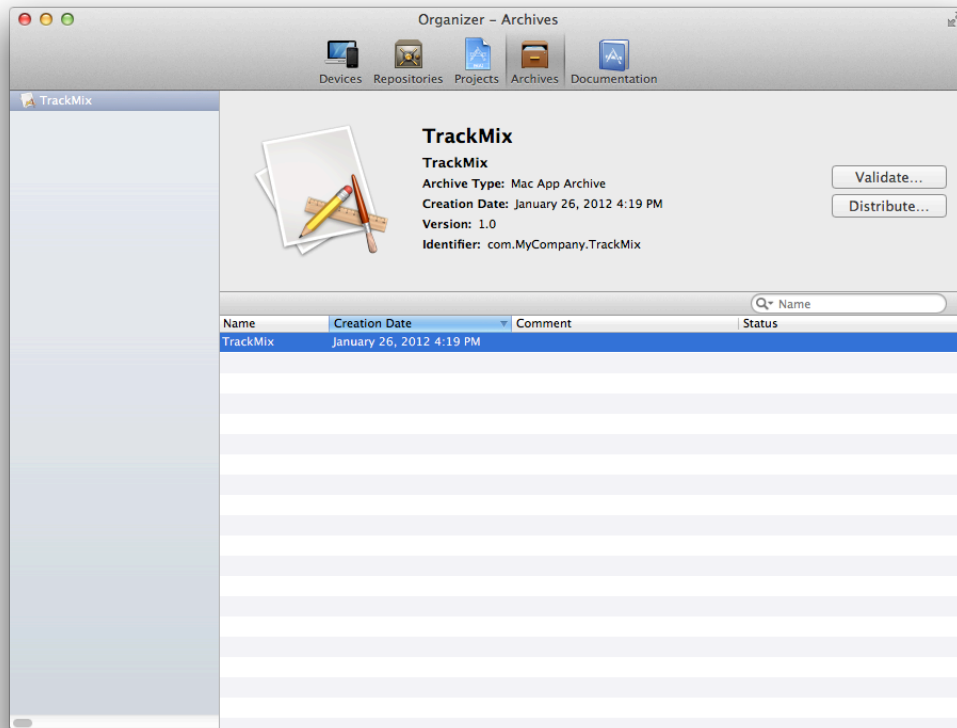
Use Xcode to code sign your application for distribution outside of the Mac App Store.

#### To code sign an application with your Developer ID application certificate . . .

1. In Xcode, select the project in the project navigator.  
Xcode displays the project editor.
2. In the left column of the project editor, ensure that the project is selected; then click Build Settings at the top of the window.
3. Type “code signing” into the project editor search field.  
The list of build settings now shows only the Code Signing setting.
4. From the Code Signing Identity pop-up menu, choose your Developer ID application certificate.

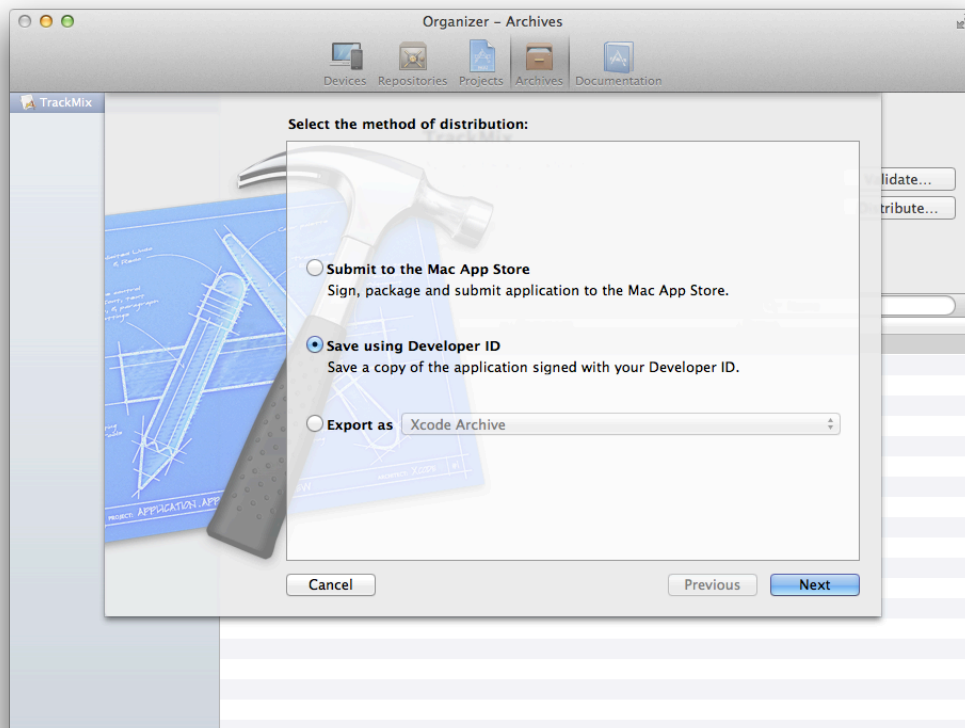
5. Choose Product > Archive.

Xcode constructs an archive containing your code signed application and opens the Organizer window, showing the archive.



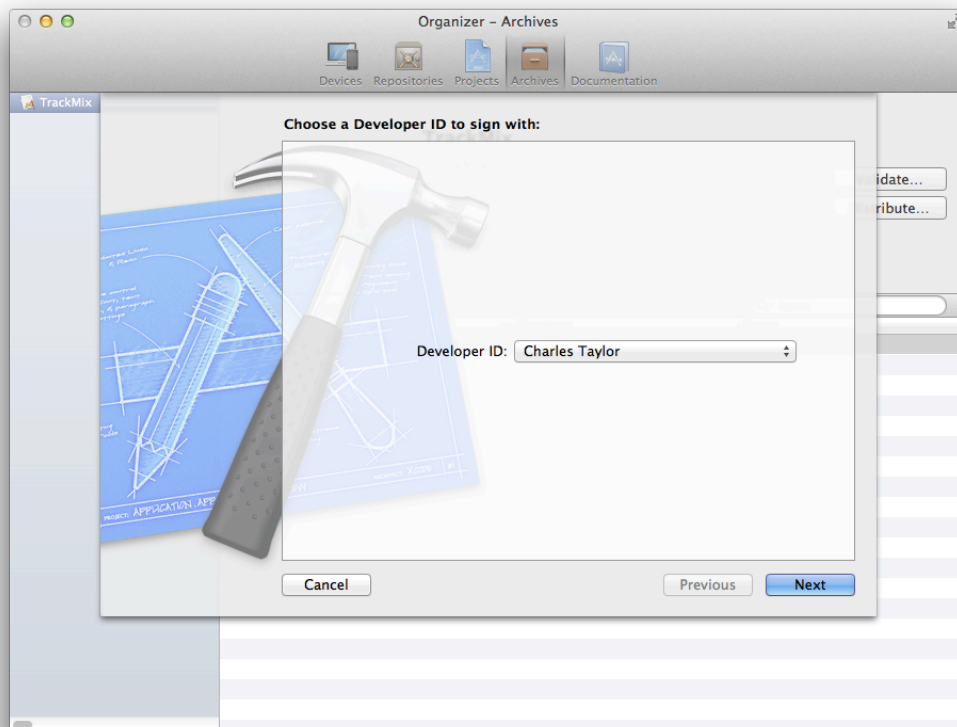
6. Select the newly-created archive in the Organizer window, then click Distribute.

A dialog appears, offering a choice of distribution methods.



7. Choose "Save using Developer ID" and then click Next.

8. Choose your Developer ID name from the Developer ID pop-up menu and click Next.



9. Enter a filename and location for the signed application and click Save.

## Code Signing an Installer Package

If you want to distribute your application outside of the Mac App Store as part of an installer package, create the package as you normally do, perhaps by using the [packagemaker](#) command. Code sign the resulting package with your Developer ID installer certificate by using the [productsign](#) command.

## Enabling and Testing Gatekeeper

Mac OS X users will soon have the option of turning on Gatekeeper, a new Mac OS X security feature. When a user does this, the system provides an additional measure of safety: it blocks that user from opening newly-downloaded applications that are not Developer ID–signed. In this scenario, the same user is easily able to launch downloaded applications that are Developer ID–signed.

By default, Gatekeeper is not enabled in Mac OS X v10.7.3. For testing purposes, you can turn it on by using the new Mac OS X *system policy control* command-line tool, `spctl(8)`.

This section first explains how to enable and disable Gatekeeper, and then provides the steps for testing Gatekeeper behavior for your application.

## Enabling and Disabling Gatekeeper

### To enable Gatekeeper . . .

1. In Terminal, enter the following command:

```
$ sudo spctl --enable
```

When prompted, enter your login password.

2. Confirm that Gatekeeper is successfully enabled by entering the following command:

```
$ spctl --status
```

With Gatekeeper enabled, the previous command prints the following text in Terminal:

```
assessments enabled
```

With Gatekeeper enabled, an application that is quarantined can run only if validly code signed with a Developer ID certificate. (Learn about quarantine in this [Knowledge Base article](#).) To return a Mac OS X v10.7.3 system to its default behavior (in which Gatekeeper is turned off), again invoke the `spctl` command.

### To disable Gatekeeper . . .

1. In Terminal, enter the following command:

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

When prompted, enter your login password.

2. Confirm that Gatekeeper is successfully disabled by entering the following command:

```
$ spctl --status
```

With Gatekeeper disabled, the previous command prints the following text in Terminal:

```
assessments disabled
```

## Testing Gatekeeper Behavior

When you have your Developer ID certificates you can test the launching behavior of your application. On a system with Gatekeeper turned on, a quarantined copy of your application can be launched only if it is Developer ID–signed.

### To prepare for testing Gatekeeper behavior . . .

1. Enable Gatekeeper on your test machine by entering the following command in Terminal:

```
$ sudo spctl --enable
```

When prompted, enter your login password.

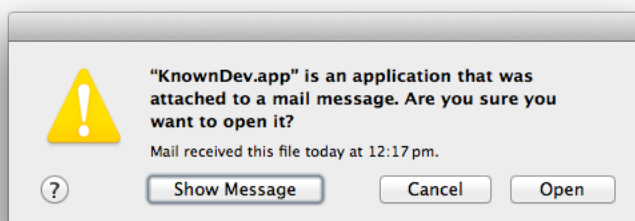
2. Quarantine a copy of your Developer ID–signed application. You can do this in either of the following ways:
  - Email your Developer ID–signed application to yourself and use the copy that Mail.app downloads;  
or
  - Host your Developer ID–signed application on your own local or remote server and use the copy that Safari.app downloads.

You are ready to test Gatekeeper behavior.

### To test Gatekeeper behavior for your Developer ID–signed application . . .

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

The system displays an alert asking if you are sure you want to open the application.



This alert, that allows you to open the quarantined application with Gatekeeper turned on, confirms that your Developer ID workflow is correct.



**Tip** If you do not see an alert at this point, it is likely that you have opened a nonquarantined copy of your application. Review the steps in [“To prepare for testing Gatekeeper behavior”](#) (page 14).

You can also test the behavior of Gatekeeper for an application that is not Developer ID–signed, as described next. The process is very similar but you use a non-Developer ID–signed copy of your application.

### To test Gatekeeper behavior for blocking applications that are not Developer ID–signed . . .

1. If Gatekeeper is not already enabled on your test machine, enable it now by entering the following command in Terminal:

```
$ sudo spctl --enable
```

When prompted, enter your login password.

2. Quarantine a copy of your application that is not Developer ID–signed. As before, you can invoke quarantine on this copy of your application in either of the following ways:
  - Email your application to yourself and use the copy that Mail.app downloads; or
  - Host your Developer ID–signed application on your own local or remote server and use the copy that Safari.app downloads.
3. In the Finder, locate the quarantined copy of your non-Developer ID–signed application and double-click its icon.

The system displays an alert that blocks you from opening the application. By way of this alert, the Gatekeeper feature protects a system by preventing first-time opening of applications from unidentified developers. Applications that have been previously opened by a user are no longer quarantined, and Gatekeeper does not prevent them from opening.



# Document Revision History

This table describes the changes to *Developer ID Tutorial*.

Date	Notes
2012-02-10	New document that explains the basics of using Developer ID for distributing code signed applications outside of the Mac App Store.



Apple Inc.

© 2012 Apple Inc.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Apple Inc.

1 Infinite Loop

Cupertino, CA 95014

408-996-1010

.Mac is a registered service mark of Apple Inc.

App Store is a service mark of Apple Inc.

Mac App Store is a service mark of Apple Inc.

Apple, the Apple logo, Finder, Keychain, Mac, Mac OS, Safari, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.