

---

# Safari Extensions Conversion Guide

User Experience



2010-08-03



Apple Inc.  
© 2010 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.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

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.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, and Safari are trademarks of Apple Inc., registered in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Simultaneously published in the United States and Canada.

**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.**

# Contents

**Introduction**      **About Safari Extensions** 5

---

[See Also](#) 5

**Chapter 1**      **Converting Chrome Extensions** 7

---

**Chapter 2**      **Converting Firefox Extensions** 9

---

**Chapter 3**      **Converting Greasemonkey Scripts** 11

---

**Document Revision History** 13

---



# About Safari Extensions

---

Safari extensions provide similar functionality to extensions for other browsers, but there are some important differences. This document points out some things you should be aware of and highlights some of the pieces you will use to build a Safari extension. If you have already written Google Chrome extensions, Firefox extensions, or Greasemonkey scripts, then this document is a great place to get started with Safari extensions.

## See Also

- *Safari Extensions Development Guide* provides comprehensive detail about how Safari Extensions work and how to build an extension.
- *Safari Extensions Reference* provides specific details about the JavaScript classes used to develop extensions.

## INTRODUCTION

### About Safari Extensions

# Converting Chrome Extensions

---

Extensions in Chrome and Safari have similar functionality, so you may find the API quite familiar. There are, of course, some key differences.

Use Extension Builder to create a new extension, to provide metadata such as your name, website, and a description of the extension, and to set up the user interface for your extension. You can also modify some parts of your UI at runtime—for example, you can disable toolbar items and add contextual menu items.

If your Chrome extension used browser or page actions, you should use toolbar items in Safari. If you used popups in Chrome, you may be able to use a toolbar item that opens a bar instead. If you used a background page, use the global page in Safari.

The event system is based on the DOM event system. To set up an event listener, call the `addEventListener` method on the object that should listen for the event. Any instance of `SafariEventTarget` or its subclasses register event listeners. For security reasons, you can dispatch messages only within your own extension.

The API is divided into two parts: the web content layer, which runs inside the web content area, and the application layer, which runs outside of it. To communicate between scripts running in different layers, use messages. In Safari, there is no difference between short-term message passing and long-term connections; you use the same methods for both.

You can submit your extension to the gallery from the Safari Dev Center. Safari extensions are always hosted by their developer, regardless of whether they are posted to the gallery.

Safari extensions must be digitally signed before they can be installed. To get your signing certificate, visit the Safari Dev Center.





# Converting Firefox Extensions

---

Both Firefox and Safari provide an extensions API, but the two APIs have substantially different functionality. As you bring your extension to Safari, you may find that some of the features you used in Firefox are in a different place in Safari, or are not available. You may still be able to bring your extension to Safari—consider the core functionality of your extension and determine how you can expose that functionality using the API that is provided. You should also examine the functionality provided in Safari that you didn't have in Firefox, to understand how you might use it to improve your extension.

To start developing Safari extensions, just enable extensions from the Develop menu; there is no special setup procedure. Safari extensions are written using HTML5, CSS3, and JavaScript; the information stored in RDF and XUL files for Firefox extensions is stored in the `Info.plist` file for Safari Extensions. This file, which you create and edit using Extension Builder, contains all the metadata about your extension and a description of its user interface.

Extension Builder, which is part of Safari, serves as a central tool during every stage of extension development. You use Extension Builder to:

- Create a new extension
- Provide metadata such as your name, website, and a description of the extension
- Set up the user interface for your extensions
- Install your extension so that you can test and debug it
- Sign and package your extension for distribution

Safari extensions must be digitally signed before they can be installed. To get your signing certificate, visit the Safari Dev Center.



# Converting Greasemonkey Scripts

---

Safari extensions can inject JavaScript and CSS, so bringing Greasemonkey scripts to Safari is generally quite straightforward. You need to add your injected content to your extension's bundle and add these files to your extension's list of injected scripts and style sheets using the Extension Builder. You can also inject scripts and style sheets at runtime using methods on the `SafariExtension` object. For more information, see "Injecting Scripts" and "Injecting Styles".

One important difference is how you access your extension's settings. The settings API isn't directly available to injected scripts because they are run as part of the web content layer. To access settings, dispatch a message from the injected script to the global page via the `SafariContentBrowserTabProxy` object. Because the global page is part of the application layer, scripts running in it can access your extension's settings. Then dispatch a message that contains the settings information from the global page back to the injected script.

Safari extensions must be digitally signed before they can be installed. To get your signing certificate, visit the Safari Dev Center.



# Document Revision History

---

This table describes the changes to *Safari Extensions Conversion Guide*.

Date	Notes
2010-08-03	Reorganized to improve readability.
2010-06-15	New document that describes extensions from a high level, pointing out similarities and differences between extensions in Safari and in other browsers.

**REVISION HISTORY**

Document Revision History