
Apple URL Scheme Reference

Networking & Internet



2009-06-17



Apple Inc.
© 2009 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

iTunes Music Store is a service mark of Apple Inc., registered in the U.S. and other countries.

Apple, the Apple logo, Cocoa, Cocoa Touch, iPhone, iTunes, Mac, Mac OS, Safari, and WebObjects are trademarks of Apple Inc., registered in the United States and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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 7

Organization of This Document 7

Mail Links 9

Phone Links 11

Text Links 13

Map Links 15

YouTube Links 17

iTunes Links 19

Document Revision History 21

Tables and Listings

Phone Links 11

Listing 1 Turning telephone number detection off 11

Map Links 15

Table 1 Supported Google Maps parameters 16

Introduction

Note: This document was previously titled *iPhone URL Scheme Reference*.

This document describes several URL schemes that are supported by system applications on iOS and Mac OS X. Native iOS applications and web applications running in Safari on any platform can use these schemes to integrate with system applications and provide a more seamless experience for the user. For example, if your iOS application displays telephone numbers, you could use an appropriate URL to launch the Phone application whenever someone taps one of those numbers. Similarly, clicking an iTunes link, launches the iTunes application and plays the song specified in the link. What happens when a user clicks a link depends on the platform and the installed system applications.

This document describes those schemes that require special attributes or special formatting in order to be understood by the associated system application. As a result, this document does not describe all URL schemes supported on different Apple platforms.

You should read this document if you want to launch a system application from your iOS application or from your web application running in Safari. This document contains both Cocoa Touch sample code—using the `openURL:` method of the shared `UIApplication` object to open URLs—and HTML samples. For more information on how to use the `openURL:` method, see *UIApplication Class Reference*.

Organization of This Document

Each article in this document describes the format of a particular type of URL:

- [“Mail Links”](#) (page 9) describes the format for sending email with the Mail application.
- [“Phone Links”](#) (page 11) describes the format for dialing phone numbers in the Phone application.
- [“Text Links”](#) (page 13) describes the format for launching the Text application.
- [“Map Links”](#) (page 15) describes the format for specifying locations in the Maps application.
- [“YouTube Links”](#) (page 17) describes the format for linking to YouTube videos.
- [“iTunes Links”](#) (page 19) describes the format for linking to items in the iTunes Music Store.

Mail Links

The `mailto` scheme is used to launch the Mail application and open the email compose sheet. When specifying a `mailto` URL, you must provide the target email address. The following examples show strings formatted for Safari and for native iPhone applications.

- HTML link:

```
<a href="mailto:frank@wwcdemo.example.com">John Frank</a>
```

- Native application URL string:

```
mailto:frank@wwcdemo.example.com
```

You can also include a subject field, a message, and multiple recipients in the To, Cc, and Bcc fields. (In iOS, the `from` attribute is ignored.) The following example shows a `mailto` URL that includes several different attributes:

```
mailto:foo@example.com?cc=bar@example.com&subject=Greetings%20from%20Cupertino!&body=Wish%20you%20were%20here!
```

For detailed information on the format of the `mailto` scheme, see [RFC 2368](#).

iOS Note: If the Mail application is not installed on the device, clicking a `mailto` URL displays an appropriate warning message to the user.

Phone Links

Note: Phone links are supported on iOS only.

The `tel` URL scheme is used to launch the Phone application and initiate dialing of the specified phone number. Tapping a telephone link on a webpage displays an alert asking if the user really wants to dial the phone number and initiates dialing if the user accepts. In a native application, opening a URL with the `tel` scheme initiates dialing without further prompting from the user.

You can specify phone links explicitly in both web and native iOS applications using the `tel` URL scheme. The following examples show the strings formatted for Safari and for a native application:

- HTML link:

```
<a href="tel:1-408-555-5555">1-408-555-5555</a>
```

- Native application URL string:

```
tel:1-408-555-5555
```

To prevent users from maliciously redirecting phone calls or changing the behavior of a phone or account, the Phone application supports most, but not all, of the special characters in the `tel` scheme. Specifically, if a URL contains the `*` or `#` characters, the Phone application does not attempt to dial the corresponding phone number. If your application receives URL strings from the user or an unknown source, you should also make sure that any special characters that might not be appropriate in a URL are escaped properly. For native applications, use the `stringByAddingPercentEscapesUsingEncoding:` method of `NSString` to escape characters, which returns a properly escaped version of your original string.

In Safari on iOS, telephone number detection is on by default. However, if your webpage contains numbers that can be interpreted as phone numbers, but are not phone numbers, you can turn off telephone number detection. You might also turn off telephone number detection to prevent the DOM document from being modified when parsed by the browser. To turn off telephone number detection in Safari on iOS, use the `format-detection` meta tag as follows:

```
<meta name = "format-detection" content = "telephone=no">
```

Listing 1 shows a simple webpage in which automatic telephone number detection is off. When displayed in Safari on iOS, the 408-555-5555 telephone number does not appear as a link. However, the 1-408-555-5555 number does appear as a link because it is in a phone link.

Listing 1 Turning telephone number detection off

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" >
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Telephone Number Detection</title>
```

Phone Links

```
<meta name = "viewport" content = "width=device-width">
  <!-- Turn off telephone number detection. -->
  <meta name = "format-detection" content = "telephone=no">
</head>
<body>
  <!-- Then use phone links to explicitly create a link. -->
  <p>A phone number: <a href="tel:1-408-555-5555">1-408-555-5555</a></p>
  <!-- Otherwise, numbers that look like phone numbers are not links. -->
  <p>Not a phone number: 1-408-555-5555</p>
</body>
</html>
```

iOS Note: If the Phone application is not installed on the device, opening a `tel` URL displays an appropriate warning message to the user.

For more information about the `tel` URL scheme, see [RFC 2806](#) and [RFC 2396](#).

Text Links

Note: Text links are supported on iOS only.

The `sms` scheme is used to launch the Text application. The format for URLs of this type is "`sms:<phone>`", where `<phone>` is an optional parameter that specifies the target phone number of the SMS message. This parameter can contain the digits 0 through 9 and the plus (+), hyphen (-), and period (.) characters. The URL string must not include any message text or other information.

The following examples show strings formatted for Safari and for native applications.

- HTML links:

```
<a href="sms:">Launch Text Application</a>  
<a href="sms:1-408-555-1212">New SMS Message</a>
```

- Native application URL strings:

```
sms:  
sms:1-408-555-1212
```

iOS Note: If the Text application is not installed on the device, clicking an `sms` URL displays an appropriate warning message to the user.

Map Links

The maps URL scheme is used to show geographical locations and to generate driving directions between two points. If your application includes address or location information, you can use map links to forward that information to the Maps application on iOS and the Google Maps website on other platforms.

Unlike some schemes, map URLs do not start with a “maps” scheme identifier. Instead, map links are specified as regular `http` links but are targeted at the Google Maps servers. The following examples show the strings you would use in Safari and in a native application to show a map of the city of Cupertino, California.

- HTML link:

```
<a href="http://maps.google.com/maps?q=cupertino">Cupertino</a>
```

- Native application URL string:

```
http://maps.google.com/maps?q=cupertino
```

The following examples show the strings you would use to provide driving directions between San Francisco and Cupertino:

- HTML link:

```
<a href="http://maps.google.com/maps?daddr=San+Francisco,+CA&saddr=cupertino">Directions</a>
```

- Native application string:

```
http://maps.google.com/maps?daddr=San+Francisco,+CA&saddr=cupertino
```

iOS Note: If the Maps application is not installed on the device, opening a map URL forwards the request to the Google Maps website.

Although the Maps application supports many of the Google Maps parameters and queries, it does not support them all. The rules for creating a valid map link are as follows:

- The domain must be `google.com` and the subdomain must be `maps` or `ditu`.
- The path must be `/`, `/maps`, `/local`, or `/m` if the query contains `site` as the key and `local` as the value.
- The path cannot be `/maps/*`.
- All parameters must be supported. See Table 1 for list of supported parameters.
- A parameter cannot be `q=*` if the value is a URL (so KML is not picked up).
- The parameters cannot include `view=text` or `dirflg=r`.

Table 1 lists the parameters supported by iOS along with a brief description of each. For a complete description of these parameters, see [Google Map Parameters](#).

Table 1 Supported Google Maps parameters

Parameter	Notes
q=	The query parameter. This parameter is treated as if it had been typed into the query box by the user on the maps.google.com page. q=* is not supported
near=	The location part of the query.
ll=	The latitude and longitude points (in decimal format, comma separated, and in that order) for the map center point.
sll=	The latitude and longitude points from which a business search should be performed.
spn=	The approximate latitude and longitude span.
sspn=	A custom latitude and longitude span format used by Google.
t=	The type of map to display.
z=	The zoom level.
saddr=	The source address, which is used when generating driving directions
daddr=	The destination address, which is used when generating driving directions.
latlng=	A custom ID format that Google uses for identifying businesses.
cid=	A custom ID format that Google uses for identifying businesses.

YouTube Links

The YouTube URL scheme is used to launch the YouTube application or connect to the YouTube website to play the specified video. If your application links to YouTube content, you can use this scheme to play videos from your application.

Unlike some schemes, YouTube URLs do not start with a “youtube” scheme identifier. Instead, they are specified as regular `http` links but are targeted at the YouTube server. The following examples show the basic strings you would use in Safari and in an application to show a YouTube video. In each example, you would need to replace the `VIDEO_IDENTIFIER` value with the identifier of the video you wanted to display:

- HTML links:

```
<a href="http://www.youtube.com/watch?v=VIDEO_IDENTIFIER">Play Video</a>  
<a href="http://www.youtube.com/v/VIDEO_IDENTIFIER">Play Video</a>
```

- Native application URL strings:

```
http://www.youtube.com/watch?v=VIDEO_IDENTIFIER  
http://www.youtube.com/v/VIDEO_IDENTIFIER
```

iOS Note: If the YouTube video cannot be viewed on the device, iOS displays an appropriate warning message to the user.

iTunes Links

The iTunes URL scheme is used to link to content on the iTunes Music Store. The iTunes URL format is complicated to construct, so you create it using an online tool called iTunes Link Maker. The tool allows you to select a country destination and media type, and then search by song, album, or artist. After you select the item you want to link to, it generates the corresponding URL.

The following examples show the strings you would use in Safari and in a native iOS application to link to a song on the iTunes Music Store. The HTML example includes the complete link returned by the iTunes Link Maker tool, which includes a link to any appropriate artwork for the target link.

- **HTML link:**

```
<a
  href="http://phobos.apple.com/WebObjects/MZStore.woa/wa/viewAlbum?i=156093464&id=156093462&s=143441">
  </img>
</a>
```

- **Native application URL string:**

```
http://phobos.apple.com/WebObjects/MZStore.woa/wa/viewAlbum?i=156093464&id=156093462&s=143441
```

For more information on creating iTunes links, see [iTunes Link Maker FAQ](#). That webpage contains a link to the iTunes Link Maker tool.

Document Revision History

This table describes the changes to *Apple URL Scheme Reference*.

Date	Notes
2009-06-17	Minor edits throughout.
2009-06-08	Updated the description of the mailto scheme to reflect the fact that iOS ignores the from attribute.
2009-01-06	Changed the title from "iPhone URL Scheme Reference" and applied other minor edits throughout.
2008-10-29	Added information about support for the sms URL scheme.
2008-10-15	First version of iPhone URL Scheme Reference.
	The information in this document was previously published in <i>iOS Programming Guide</i> .

