

[SiriKit](#) / Specifying Synonyms for Your App Name

## Article

# Specifying Synonyms for Your App Name

Provide alternative names for your app that are more familiar or easier for users to speak.

## Overview

When engaging your app's services through Siri, users must include the name of your app in any spoken requests. To simplify those requests, you can provide Siri with a list of synonyms for your app name.

### Important

Siri recognizes alternative app names only for apps that include an Intents app extension with support for one or more intents. Alternative names must be either based on your app's original name or be a legitimate name common to users of the app. You may not include the names of other apps on the system in an attempt to divert requests to your app.

To define synonyms for your app name, include the `INAlternativeAppNames` key in the `Info.plist` file of your iOS app or WatchKit extension. (watchOS apps don't inherit the synonyms of their iOS app, so you must declare them explicitly in both places). Place this key at the top level of your property list, configure it as an array of dictionaries, and add the following additional keys to each dictionary:

#### `INAlternativeAppName`

(Required) A string containing the alternative name by which users may refer to your app. Alternatively, specify a variable name that's present in your app's localized `InfoPlist.strings` files.

#### `INAlternativeAppNamePronunciationHint`

(Optional) A string containing a pronunciation hint for the alternate name, formatted in a "sounds like" format. Alternatively, specify a variable name that's present in your app's

localized `InfoPlist.strings` files.

### Important

If you specify synonyms for your watchOS app, you must build your app for deployment on watchOS 4 or later. An app that includes synonyms won't install on earlier versions of watchOS. In addition, the synonyms you specify must be identical to the ones that you specified for your iOS app.

To localize your app name and pronunciation hints, specify variable names (instead of actual values) for the `INAlternativeAppName` and `INAlternativeAppNamePronunciationHint` keys of each entry. In your app's `InfoPlist.strings` files, include the variable names as localization keys and set their values to appropriately localized strings. For example, you might set the value of the `INAlternativeAppName` key to `APP_NAME_SYNONYM_1` and include an entry in your `InfoPlist.strings` file such as the following:

```
APP_NAME_SYNONYM_1 = "Localized Name 1"
```

If you use a variable name for a key, you must provide an entry in your app's base localization at a minimum. If the system can't find a specific value for the key in the current localization, it uses the base localization as a fallback.

You may specify no more than three dictionaries in the array associated with the `INAlternativeAppNames` key. This limit is per localization, so each localized version of your `Info.plist` file may have its own set of alternatives.

---

## See Also

### Articles



[Adding User Interactivity with Siri Shortcuts and the Shortcuts App](#)

Add custom intents and parameters to help users interact more quickly and effectively with Siri and the Shortcuts app.



[Defining Relevant Shortcuts for the Siri Watch Face](#)

Inform Siri when your app's shortcuts may be useful to the user.



[Deleting Donated Shortcuts](#)

Remove your donations from Siri.



[Dispatching intents to handlers](#)

Provide SiriKit with an intent handler capable of handling a specific intent.

## Improving Siri Media Interactions and App Selection

Fine-tune voice controls and improve Siri Suggestions by sharing app capabilities, customized names, and listening habits with the system.

## Improving interactions between Siri and your messaging app

Donate app-specific content, use Siri's contact suggestions, and adopt the latest platform features to create a more consistent messaging experience.

## Registering Custom Vocabulary with SiriKit

Register your app's custom terminology, and provide sample phrases for how to use your app with Siri.

## Confirming the Details of an Intent

Perform final validation of the intent parameters and verify that your services are ready to fulfill the intent.

## Handling an Intent

Fulfill the intent and provide feedback to SiriKit about what you did.

## Resolving the Parameters of an Intent

Validate the parameters of an intent and make sure that you have the information you need to continue.

## Generating a List of Ride Options

Generate ride options for Maps to display to the user.

## Handling the Ride-Booking Intents

Support the different intent-handling sequences for booking rides with Shortcuts or Maps.

## Donating Reservations

Inform Siri of reservations made from your app.

## Intent Phrases

The keys that you include in your global vocabulary file to show how users engage your app from Siri.

## Localizing Your Vocabulary for Chinese Dialects

Apply emphasis markers to your pronunciation tips to assist Siri with Chinese dialects.