

## Article

# Parameter Vocabularies

The keys you include in your global vocabulary file to describe app-specific terms.

## Overview

The `ParameterVocabularies` key contains the custom terminology that is global to all users of your app. This key contains an array of dictionaries, each of which defines a set of terms and the intent properties to which those terms apply. The keys for the dictionary are as follows:

### `ParameterNames`

(Required) An array of strings, each of which contains a key path for a property name from an intent class. For example, to specify a custom ride option for a ride request intent, specify the key path `INRequestRideIntent.rideOptionName`.

### `ParameterVocabulary`

(Required) An array of dictionaries containing the custom terms to associate with the specified properties. Each dictionary contains the following keys.

### `VocabularyItemIdentifier`

(Required) The string that Siri assigns to the identifier property of the `INSpeakableString` object when it recognizes your custom vocabulary. This identifier applies to all synonyms.

### `VocabularyItemSynonyms`

(Required) An array of dictionaries that define the phrases for Siri to recognize. Use the following keys in the dictionary to specify the phrase, pronunciation hints, and examples:

### `VocabularyItemPhrase`

(Required) A string containing the custom phrase that you want to define. Spell the phrase the same way that your app uses it.

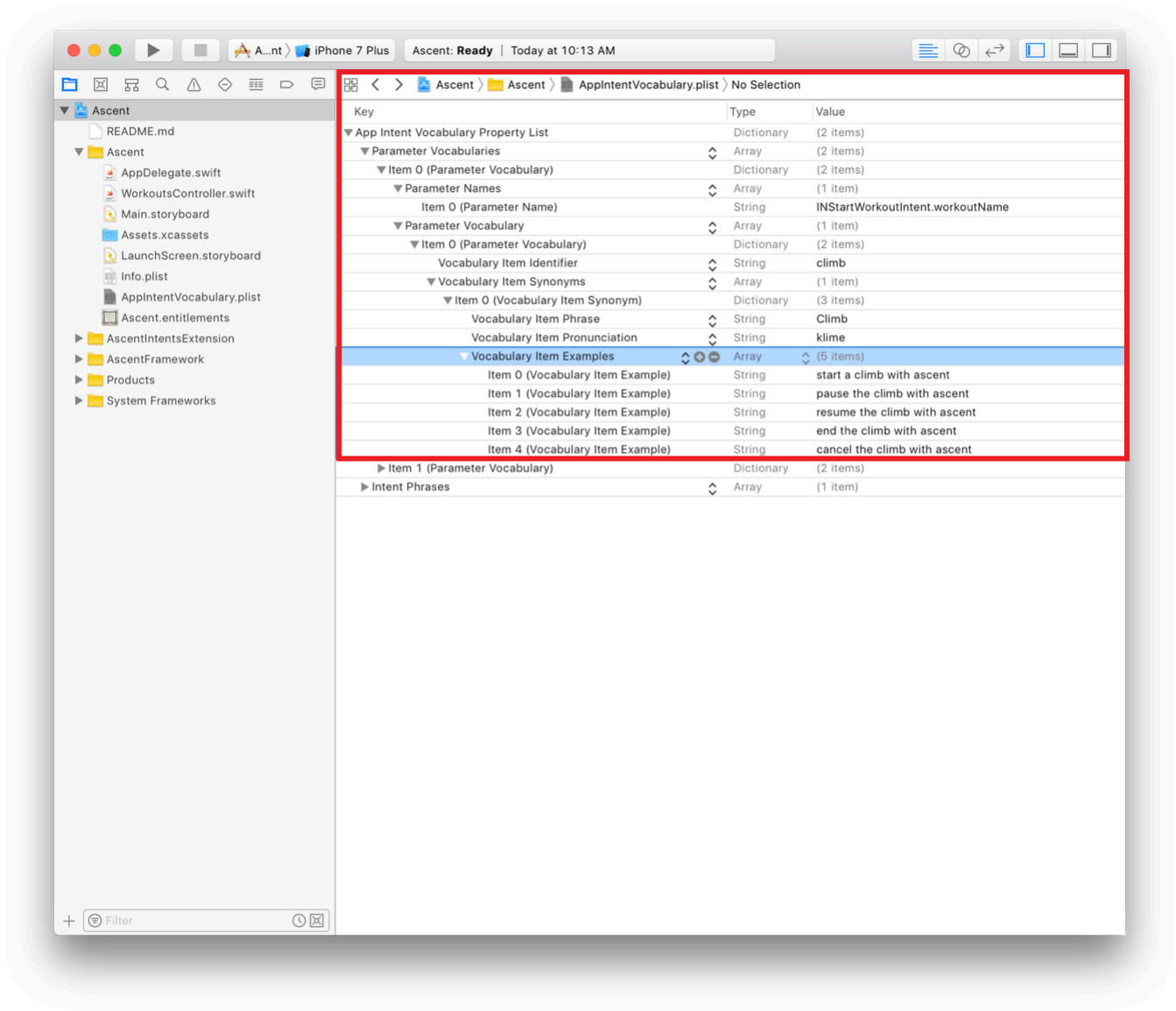
### `VocabularyItemPronunciation`

(Optional) A string containing pronunciation hints for the phrase, formatted in a “sounds like” format. For example, the pronunciation for the phrase “iTunes” might be “eye toons”.

## VocabularyItemExamples

(Optional) An array of strings, each of which contains examples of how to use the phrase.

Xcode helps you add terms to your vocabulary file. When you create a property list file with the name `AppIntentVocabulary.plist`, Xcode knows that the file should contain custom vocabulary, and it makes only the relevant keys available in the property list editor. The figure below shows an example of a global vocabulary file in Xcode that contains custom terms for a workout app.














## See Also

## Related Documentation

### Intent Phrases

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

# 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.



## Specifying Synonyms for Your App Name

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



## Intent Phrases

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