
NSSpeechSynthesizer Class Reference

User Experience: Speech Technologies



2009-08-20



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

Apple, the Apple logo, Cocoa, iPod, Mac, Mac OS, and QuickTime are trademarks of Apple Inc., registered in the United States 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

NSSpeechSynthesizer Class Reference 7

Overview	7
Speech Feedback Window	8
Tasks	8
Creating Speech Synthesizers	8
Configuring Speech Synthesizers	8
Getting Speech Synthesizer Information	9
Getting Speech State	9
Synthesizing Speech	9
Getting Phonemes	10
Class Methods	10
attributesForVoice:	10
availableVoices	10
defaultVoice	11
isAnyApplicationSpeaking	11
Instance Methods	11
addSpeechDictionary:	11
continueSpeaking	12
delegate	12
initWithVoice:	13
isSpeaking	13
objectForProperty:error:	14
pauseSpeakingAtBoundary:	14
phonemesFromText:	15
rate	15
setDelegate:	16
setObject:forProperty:error:	16
setRate:	16
setUsesFeedbackWindow:	17
setVoice:	17
setVolume:	18
startSpeakingString:	18
startSpeakingString:toURL:	19
stopSpeaking	20
stopSpeakingAtBoundary:	20
usesFeedbackWindow	21
voice	21
volume	21
Constants	22
Voice Attributes Keys	22
Voice Gender Keys	24

- Speech Synthesizer Property Keys 24
- Speaking Modes for NSSpeechInputModeProperty 29
- Speaking Modes for NSSpeechNumberModeProperty 29
- NSSpeechStatusProperty Dictionary Keys 29
- NSSpeechErrorProperty Dictionary Keys 30
- NSSpeechSynthesizerInfoProperty Dictionary Keys 31
- NSSpeechPhonemeSymbolsProperty Dictionary Keys 32
- Speech Command Delimiter Keys 33
- Speech Dictionary Properties Keys 33
- NSSpeechBoundary 34

Document Revision History 35

Index 37

Figures and Listings

NSSpeechSynthesizer Class Reference 7

Figure 1	Speech feedback window	8
Listing 1	Identifiers of the Mac OS X system voices	23

NSSpeechSynthesizer Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.3 and later.
Companion guide	Speech
Declared in	NSSpeechSynthesizer.h
Related sample code	SayIt

Overview

The `NSSpeechSynthesizer` class is the Cocoa interface to Speech Synthesis on Mac OS X. Instances of this class are called **speech synthesizers**.

Speech Synthesis, also called text-to-speech (TTS), parses text and converts it into audible speech. It offers a concurrent feedback mode that can be used in concert with or in place of traditional visual and aural notifications. For example, your application can use an `NSSpeechSynthesizer` object to “pronounce” the text of important alert dialogs. Synthesized speech has several advantages. It can provide urgent information to users without forcing them to shift attention from their current task. And because speech doesn’t rely on visual elements for meaning, it is a crucial technology for users with vision or attention disabilities.

In addition, synthesized speech can help save system resources. Because sound samples can take up large amounts of room on disk, using text in place of sampled sound is extremely efficient, and so a multimedia application might use an `NSSpeechSynthesizer` object to provide a narration of a QuickTime movie instead of including sampled-sound data on a movie track.

When you create an `NSSpeechSynthesizer` instance using the default initializer (`init`), the class uses the **default voice** selected in System Preferences > Speech. Alternatively, you can select a specific voice for an `NSSpeechSynthesizer` instance by initializing it with `initWithVoice:` (page 13). To begin synthesis, send either `startSpeakingString:` (page 18) or `startSpeakingString:toURL:` (page 19) to the instance. The former generates speech through the system’s default sound output device; the latter saves the generated speech to a file. If you wish to be notified when the current speech concludes, set a delegate object using `setDelegate:` (page 16) and implement the delegate method `speechSynthesizer:didFinishSpeaking:.`

Speech Synthesis is just one of the Mac OS X speech technologies. The Speech Recognizer technology allows applications to “listen to” text spoken in U.S. English; the `NSSpeechRecognizer` class is the Cocoa interface to this technology. Both technologies provide benefits for all users, and are particularly useful to those users who have difficulties seeing the screen or using the mouse and keyboard.

Speech Feedback Window

The speech feedback window (Figure 1) displays the text recognized from the user’s speech and the text from which an `NSSpeechSynthesizer` object synthesizes speech. Using the feedback window makes spoken exchange more natural and helps the user understand the synthesized speech.

Figure 1 Speech feedback window



For example, your application may use an `NSSpeechRecognizer` object to listen for the command “Play some music.” When it recognizes this command, your application might then respond by speaking “Which artist?” using a speech synthesizer.

When `usesFeedbackWindow` is YES, the speech synthesizer uses the feedback window if its visible, which the user specifies in System Preferences > Speech.

Tasks

Creating Speech Synthesizers

- `initWithVoice:` (page 13)
Initializes the receiver with a voice.

Configuring Speech Synthesizers

- `usesFeedbackWindow` (page 21)
Indicates whether the receiver uses the speech feedback window.
- `setUsesFeedbackWindow:` (page 17)
Specifies whether the receiver uses the speech feedback window.

- [voice](#) (page 21)
Returns the identifier of the receiver's current voice.
- [setVoice:](#) (page 17)
Sets the receiver's current voice.
- [rate](#) (page 15)
Provides the receiver's speaking rate.
- [setRate:](#) (page 16)
Specifies the receivers speaking rate.
- [volume](#) (page 21)
Provides the receiver's speaking volume.
- [setVolume:](#) (page 18)
Specifies the receiver's speaking volume.
- [addSpeechDictionary:](#) (page 11)
Registers the given speech dictionary with the receiver.
- [objectForProperty:error:](#) (page 14)
Provides the value of a receiver's property.
- [setObject:forProperty:error:](#) (page 16)
Specifies the value of a receiver's property.
- [delegate](#) (page 12)
Returns the receiver's delegate.
- [setDelegate:](#) (page 16)
Sets the receiver's delegate.

Getting Speech Synthesizer Information

- + [availableVoices](#) (page 10)
Provides the identifiers of the voices available on the system.
- + [attributesForVoice:](#) (page 10)
Provides the attribute dictionary of a voice.
- + [defaultVoice](#) (page 11)
Provides the identifier of the default voice.

Getting Speech State

- + [isAnyApplicationSpeaking](#) (page 11)
Indicates whether any application is currently speaking through the sound output device.

Synthesizing Speech

- [isSpeaking](#) (page 13)
Indicates whether the receiver is currently generating synthesized speech.
- [startSpeakingString:](#) (page 18)
Begins speaking synthesized text through the system's default sound output device.

- [startSpeakingString:toURL:](#) (page 19)
Begins synthesizing text into a sound (AIFF) file.
- [pauseSpeakingAtBoundary:](#) (page 14)
Pauses synthesis in progress at a given boundary.
- [continueSpeaking](#) (page 12)
Resumes synthesis.
- [stopSpeaking](#) (page 20)
Stops synthesis in progress.
- [stopSpeakingAtBoundary:](#) (page 20)
Stops synthesis in progress at a given boundary.

Getting Phonemes

- [phonemesFromText:](#) (page 15)
Provides the phoneme symbols generated by the given text.

Class Methods

attributesForVoice:

Provides the attribute dictionary of a voice.

```
+ (NSDictionary *)attributesForVoice:(NSString *)voiceIdentifier
```

Parameters

voiceIdentifier

Identifier of the voice whose attributes you want to obtain.

Return Value

Attribute dictionary of the voice identified by *voiceIdentifier*. The attributes keys and value types are listed in “[Voice Attributes](#)” (page 22)

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSSpeechSynthesizer.h

availableVoices

Provides the identifiers of the voices available on the system.

```
+ (NSArray *)availableVoices
```

Return Value

Array of strings representing the identifiers of each voice available on the system.

Availability

Available in Mac OS X v10.3 and later.

See Also

+ [attributesForVoice:](#) (page 10)

- [setVoice:](#) (page 17)

Declared In

NSSpeechSynthesizer.h

defaultVoice

Provides the identifier of the default voice.

+ (NSString *)defaultVoice

Return Value

Identifier of the default voice.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSSpeechSynthesizer.h

isAnyApplicationSpeaking

Indicates whether any application is currently speaking through the sound output device.

+ (BOOL)isAnyApplicationSpeaking

Return Value

YES when another application is producing speech through the sound output device, NO otherwise.

Discussion

You usually invoke this method to prevent your application from speaking over speech being generated by another application or system component.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSSpeechSynthesizer.h

Instance Methods

addSpeechDictionary:

Registers the given speech dictionary with the receiver.

- (void)addSpeechDictionary:(NSDictionary *)*speechDictionary*

Parameters

speechDictionary

Speech dictionary to add to the receiver's dictionaries. The key-value pairs are listed in "[Speech Dictionary Properties](#)" (page 33).

Discussion

See the discussion of `UseSpeechDictionary` in *Speech Synthesis Manager Reference* for more information.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSSpeechSynthesizer.h

continueSpeaking

Resumes synthesis.

- (void)continueSpeaking

Discussion

At any time after [pauseSpeakingAtBoundary:](#) (page 14) is called, [continueSpeaking](#) (page 12) can be called to continue speaking from the beginning of the word at which speech paused.

Sending [continueSpeaking](#) (page 12) to a receiver that is not currently in a paused state has no effect on the synthesizer or on future calls to the [pauseSpeakingAtBoundary:](#) (page 14) function. If you call [continueSpeaking](#) (page 12) on a synthesizer before a pause is effective, [continueSpeaking](#) (page 12) cancels the pause.

If the [pauseSpeakingAtBoundary:](#) (page 14) method stopped speech in the middle of a word, the synthesizer will start speaking that word from the beginning when you call [continueSpeaking](#) (page 12).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [pauseSpeakingAtBoundary:](#) (page 14)

Declared In

NSSpeechSynthesizer.h

delegate

Returns the receiver's delegate.

- (id < NSSpeechSynthesizerDelegate >)delegate

Return Value

The receiver's delegate.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setDelegate:](#) (page 16)

Declared In

NSSpeechSynthesizer.h

initWithVoice:

Initializes the receiver with a voice.

```
- (id)initWithVoice:(NSString *)voiceIdentifier
```

Parameters

voiceIdentifier

Identifier of the voice to set as the current voice. When `nil`, the default voice is used. Passing in a specific voice means the initial speaking rate is determined by the synthesizer's default speaking rate; passing `nil` means the speaking rate is automatically set to the rate the user specifies in Speech preferences.

Return Value

Initialized speech synthesizer or `nil` when the voice identified by *voiceIdentifier* is not available or when there's an allocation error.

Availability

Available in Mac OS X v10.3 and later.

See Also

+ [availableVoices](#) (page 10)

Declared In

NSSpeechSynthesizer.h

isSpeaking

Indicates whether the receiver is currently generating synthesized speech.

```
- (BOOL)isSpeaking
```

Return Value

YES when the receiver is generating synthesized speech, NO otherwise.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [startSpeakingString:](#) (page 18)
 - [startSpeakingString:toURL:](#) (page 19)
 - [stopSpeaking](#) (page 20)

Declared In

NSSpeechSynthesizer.h

objectForProperty:error:

Provides the value of a receiver's property.

```
- (id)objectForProperty:(NSString *)speechProperty error:(NSError **)out_error
```

Parameters

speechProperty

Property to get.

out_error

On output, error that occurred while obtaining the value of *speechProperty*.

Return Value

The value of *speechProperty*.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setObject:forProperty:error:](#) (page 16)

Declared In

NSSpeechSynthesizer.h

pauseSpeakingAtBoundary:

Pauses synthesis in progress at a given boundary.

```
- (void)pauseSpeakingAtBoundary:(NSSpeechBoundary)boundary
```

Parameters

boundary

Boundary at which to pause speech. The supported bound types are listed in [“NSSpeechBoundary”](#) (page 34).

Discussion

Pass the constant [NSSpeechImmediateBoundary](#) (page 34) to pause immediately, even in the middle of a word. Pass [NSSpeechWordBoundary](#) (page 34) or [NSSpeechSentenceBoundary](#) (page 34) to pause speech at the end of the current word or sentence, respectively.

You can determine whether your application has paused a synthesizer's speech output by obtaining the [NSSpeechStatusProperty](#) (page 25) property through the [objectForProperty:error:](#) (page 14) method. While a synthesizer is paused, the speech status information indicates that [NSSpeechStatusOutputBusy](#) (page 30) and [NSSpeechStatusOutputPaused](#) (page 30) are both YES.

If the end of the string being spoken is reached before the specified pause point, speech output pauses at the end of the string.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [continueSpeaking](#) (page 12)

Declared In

NSSpeechSynthesizer.h

phonemesFromText:

Provides the phoneme symbols generated by the given text.

```
- (NSString *)phonemesFromText:(NSString *)text
```

Parameters*text*

Text from which to generate phonemes.

Return Value

Phonemes generated from text.

Discussion

Converting textual data into phonetic data is particularly useful during application development, when you might wish to adjust phrases that your application generates to produce smoother speech. By first converting the target phrase into phonemes, you can see what the synthesizer will try to speak. Then you need correct only the parts that would not have been spoken the way you want

The string returned by `phonemesFromText:` corresponds precisely to the phonemes that would be spoken had the input text been sent to `startSpeakingString:` (page 18) instead. All current property settings for the synthesizer are applied to the converted speech.

No delegate methods are called while `phonemesFromText:` (page 15) method is generating its output.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSSpeechSynthesizer.h

rate

Provides the receiver's speaking rate.

```
- (float)rate
```

Return Value

Speaking rate (words per minute).

Discussion

The range of supported rates is not predefined by the Speech Synthesis framework; but the synthesizer may only respond to a limited range of speech rates. Average human speech occurs at a rate of 180 to 220 words per minute.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setRate:` (page 16)

Declared In

NSSpeechSynthesizer.h

setDelegate:

Sets the receiver's delegate.

```
- (void)setDelegate:(id < NSSpeechSynthesizerDelegate >)delegate
```

Parameters*delegate*

Object to be the receiver's delegate.

Availability

Available in Mac OS X v10.3 and later.

See Also- [delegate](#) (page 12)**Declared In**

NSSpeechSynthesizer.h

setObject:forProperty:error:

Specifies the value of a receiver's property.

```
- (BOOL)setObject:(id)object forProperty:(NSString *)property error:(NSError
***)outError
```

Parameters*speechProperty*Property to set. The supported properties are listed in "[NSSpeechStatusProperty Dictionary Keys](#)" (page 29).*out_error*On output, error that occurred while setting *speechProperty*.**Return Value**YES when the *speechProperty* was set. NO when there was an error, specified in *out_error*.**Availability**

Available in Mac OS X v10.5 and later.

See Also- [objectForProperty:error:](#) (page 14)**Declared In**

NSSpeechSynthesizer.h

setRate:

Specifies the receivers speaking rate.

- (void)setRate:(float)rate

Parameters

rate

Words to speak per minute.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [rate](#) (page 15)

Declared In

NSSpeechSynthesizer.h

setUsesFeedbackWindow:

Specifies whether the receiver uses the speech feedback window.

- (void)setUsesFeedbackWindow:(BOOL)useFeedbackWindow

Parameters

useFeedbackWindow

YES to make the receiver use the speech feedback window if it's visible when the user begins speaking.

NO not to use the feedback window.

Discussion

See the class description for details on the `UsesFeedbackWindow` attribute.

Important: The delegate only receives the `speechSynthesizer:didFinishSpeaking:` message when speaking occurs through the feedback window.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [usesFeedbackWindow](#) (page 21)

Declared In

NSSpeechSynthesizer.h

setVoice:

Sets the receiver's current voice.

- (BOOL)setVoice:(NSString *)voiceIdentifier

Parameters

voiceIdentifier

Identifier of the voice to set at the receiver's current voice. When `nil`, the receiver sets the default voice as its current voice.

Return Value

YES when the receiver is not currently synthesizing speech and the current voice is set successfully, NO otherwise.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [voice](#) (page 21)
- + [defaultVoice](#) (page 11)

Declared In

NSSpeechSynthesizer.h

setVolume:

Specifies the receiver's speaking volume.

```
- (void)setVolume:(float)volume
```

Parameters

volume

Sound level to use for speech.

Discussion

Volumes are expressed in floating-point units ranging from 0.0 through 1.0. A value of 0.0 corresponds to silence, and a value of 1.0 corresponds to the maximum possible volume. Volume units lie on a scale that is linear with amplitude or voltage. A doubling of perceived loudness corresponds to a doubling of the volume.

Setting a value outside this range is undefined.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [volume](#) (page 21)

Declared In

NSSpeechSynthesizer.h

startSpeakingString:

Begins speaking synthesized text through the system's default sound output device.

```
- (BOOL)startSpeakingString:(NSString *)text
```

Parameters

text

Text to speak. When nil or empty, no synthesis occurs.

Return Value

YES when speaking starts successfully, NO otherwise.

Discussion

If the receiver is currently speaking synthesized speech when `startSpeakingString:` is called, that process is stopped before *text* is spoken.

When synthesis of *text* finishes normally or is stopped, the message `speechSynthesizer:didFinishSpeaking:` is sent to the delegate.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [isSpeaking](#) (page 13)
- [startSpeakingString:toURL:](#) (page 19)
- [stopSpeaking](#) (page 20)

Related Sample Code

SayIt

Declared In

NSSpeechSynthesizer.h

startSpeakingString:toURL:

Begins synthesizing text into a sound (AIFF) file.

```
(BOOL)startSpeakingString:(NSString *)text toURL:(NSURL *)url
```

Parameters

text

Text to speak. When `nil` or empty, no synthesis is started.

url

Filesystem location of the output sound file.

Return Value

YES when synthesis starts successfully, NO otherwise.

Discussion

When synthesis of *text* finishes normally or is stopped, the message `speechSynthesizer:didFinishSpeaking:` is sent to the delegate.

One example of how you might use this method is in an email program that automatically converts new messages into sound files that can be stored on an iPod for later listening.

Note: In Mac OS X V 10.4 and earlier, the delegate does not receive `speechSynthesizer:willSpeakWord:ofString:` and `speechSynthesizer:willSpeakPhoneme:` messages when text is being synthesized to a file.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [isSpeaking](#) (page 13)

- [startSpeakingString:](#) (page 18)
- [stopSpeaking](#) (page 20)

Declared In

NSSpeechSynthesizer.h

stopSpeaking

Stops synthesis in progress.

- (void)stopSpeaking

Discussion

If the receiver is currently generating speech, synthesis is halted, and the message `speechSynthesizer:didFinishSpeaking:` is sent to the delegate.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [isSpeaking](#) (page 13)
- [startSpeakingString:](#) (page 18)
- [startSpeakingString:toURL:](#) (page 19)

Related Sample Code

SayIt

Declared In

NSSpeechSynthesizer.h

stopSpeakingAtBoundary:

Stops synthesis in progress at a given boundary.

- (void)stopSpeakingAtBoundary:(NSSpeechBoundary) *boundary*

Parameters*boundary*

Boundary at which to stop speech. The supported bound types are listed in “[NSSpeechBoundary](#)” (page 34).

Discussion

Pass the constant `NSSpeechImmediateBoundary` (page 34) to stop immediately, even in the middle of a word. Pass `NSSpeechWordBoundary` (page 34) or `NSSpeechSentenceBoundary` (page 34) to stop speech at the end of the current word or sentence, respectively.

If the end of the string being spoken is reached before the specified stopping point, the synthesizer stops at the end of the string without generating an error.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSSpeechSynthesizer.h

usesFeedbackWindow

Indicates whether the receiver uses the speech feedback window.

- (BOOL)usesFeedbackWindow

Return Value

YES when the receiver uses the speech feedback window, NO otherwise.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setUsesFeedbackWindow:](#) (page 17)

Declared In

NSSpeechSynthesizer.h

voice

Returns the identifier of the receiver's current voice.

- (NSString *)voice

Return Value

Identifier of the receiver's current voice.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setVoice:](#) (page 17)

Declared In

NSSpeechSynthesizer.h

volume

Provides the receiver's speaking volume.

- (float)volume

Return Value

Speaking volume: From 0.0 (minimum) to 1.0 (maximum).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setVolume:](#) (page 18)

Declared In

NSSpeechSynthesizer.h

Constants

Voice Attributes Keys

The following constants are keys for the dictionary returned by [attributesForVoice:](#) (page 10).

```
NSString *const NSVoiceIdentifier;
NSString *const NSVoiceName;
NSString *const NSVoiceAge;
NSString *const NSVoiceGender;
NSString *const NSVoiceDemoText;
NSString *const NSVoiceLanguage;
NSString *const NSVoiceLocaleIdentifier;
NSString *const NSVoiceSupportedCharacters;
NSString *const NSVoiceIndividuallySpokenCharacters;
```

Constants

NSVoiceIdentifier

A unique string identifying the voice. The identifiers of the system voices are listed in [Listing 1](#) (page 23).

Available in Mac OS X v10.3 and later.

Declared in NSSpeechSynthesizer.h.

NSVoiceName

The name of the voice suitable for display. An NSString.

Available in Mac OS X v10.3 and later.

Declared in NSSpeechSynthesizer.h.

NSVoiceAge

The perceived age (in years) of the voice. An NSString

Available in Mac OS X v10.3 and later.

Declared in NSSpeechSynthesizer.h.

NSVoiceGender

The perceived gender of the voice. The supported values are listed in ["Voice Genders"](#) (page 24). An NSString

Available in Mac OS X v10.3 and later.

Declared in NSSpeechSynthesizer.h.

NSVoiceDemoText

A demonstration string to speak. An NSString

Available in Mac OS X v10.3 and later.

Declared in NSSpeechSynthesizer.h.

`NSVoiceLanguage`

The language of the voice (currently US English only). An `NSString`

Deprecated: Use `NSVoiceLocaleIdentifier` (page 23) instead.

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.5.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceLocaleIdentifier`

The language of the voice. An `NSString`

The canonical locale identifier string describing the voice's locale. A locale is generally composed of three pieces of ordered information: a language code, a region code, and a variant code. Refer to documentation about the `NSLocale` class or *Locales Programming Guide* for more information.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceSupportedCharacters`

A list of unicode character id ranges that define the unicode characters supported by this voice. a dictionary containing two keys: "UnicodeCharBegin", an integer value containing the beginning unicode id of this range; and "UnicodeCharEnd", an integer value containing the ending unicode id of this range. The synthesizer will convert or ignore any characters not contained in the range of supported characters.

Some voices may not provide this attribute.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceIndividuallySpokenCharacters`

A list of unicode character id ranges that define the unicode characters that can be spoken in character-by-character mode by this voice. Each list entry is a dictionary containing two keys: "UnicodeCharBegin", an integer value containing the beginning unicode id of this range; and "UnicodeCharEnd", an integer value containing the ending unicode id of this range.

These ranges can be used by your application to determine if the voice can speak the name of an individual character when spoken in character-by-character mode.

Some voices may not provide this attribute.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

Discussion

Listing 1 lists the identifiers of the system voices (defined in `/System/Library/Speech/Voices`):

Listing 1 Identifiers of the Mac OS X system voices

```
com.apple.speech.synthesis.voice.Agnes
com.apple.speech.synthesis.voice.Albert
com.apple.speech.synthesis.voice.Alex
com.apple.speech.synthesis.voice.BadNews
com.apple.speech.synthesis.voice.Bahh
com.apple.speech.synthesis.voice.Bells
com.apple.speech.synthesis.voice.Boing
com.apple.speech.synthesis.voice.Bruce
com.apple.speech.synthesis.voice.Bubbles
com.apple.speech.synthesis.voice.Cellos
com.apple.speech.synthesis.voice.Deranged
```

```

com.apple.speech.synthesis.voice.Fred
com.apple.speech.synthesis.voice.GoodNews
com.apple.speech.synthesis.voice.Hysterical
com.apple.speech.synthesis.voice.Junior
com.apple.speech.synthesis.voice.Kathy
com.apple.speech.synthesis.voice.Organ
com.apple.speech.synthesis.voice.Princess
com.apple.speech.synthesis.voice.Ralph
com.apple.speech.synthesis.voice.Trinoids
com.apple.speech.synthesis.voice.Vicki
com.apple.speech.synthesis.voice.Victoria
com.apple.speech.synthesis.voice.Whisper
com.apple.speech.synthesis.voice.Zarvox

```

Voice Gender Keys

The following constants define voice gender attributes, which are the allowable values of the [NSVoiceGender](#) (page 22) key returned by [attributesForVoice:](#) (page 10).

```

NSString *const NSVoiceGenderNeuter;
NSString *const NSVoiceGenderMale;
NSString *const NSVoiceGenderFemale;

```

Constants

`NSVoiceGenderNeuter`

A neutral voice (or a novelty voice with a humorous or whimsical quality).

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceGenderMale`

A male voice

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSVoiceGenderFemale`

A female voice

Available in Mac OS X v10.3 and later.

Declared in `NSSpeechSynthesizer.h`.

Speech Synthesizer Property Keys

These constants are used with [setObject:forProperty:error:](#) (page 16) and [objectForProperty:error:](#) (page 14) to get or set the characteristics of a synthesizer.

```

NSString *const NSSpeechStatusProperty;
NSString *const NSSpeechErrorsProperty;
NSString *const NSSpeechInputModeProperty;
NSString *const NSSpeechCharacterModeProperty;
NSString *const NSSpeechNumberModeProperty;
NSString *const NSSpeechRateProperty;
NSString *const NSSpeechPitchBaseProperty;
NSString *const NSSpeechPitchModProperty;
NSString *const NSSpeechVolumeProperty;
NSString *const NSSpeechSynthesizerInfoProperty;
NSString *const NSSpeechRecentSyncProperty;
NSString *const NSSpeechPhonemeSymbolsProperty;
NSString *const NSSpeechCurrentVoiceProperty;
NSString *const NSSpeechCommandDelimiterProperty;
NSString *const NSSpeechResetProperty;
NSString *const NSSpeechOutputToFileURLProperty;

```

Constants

NSSpeechStatusProperty

Get speech-status information for the synthesizer. An `NSDictionary` that contains speech-status information for the synthesizer. See “[NSSpeechStatusProperty Dictionary Keys](#)” (page 29) for a description of the keys present in the dictionary.

This property is used with `setObject:forProperty:error:` (page 16).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechErrorsProperty

Get speech-error information for the synthesizer. An `NSDictionary` object that contains speech-error information. See “[NSSpeechErrorProperty Dictionary Keys](#)” (page 30) for a description of the keys present in the dictionary.

This property lets you get information about various run-time errors that occur during speaking, such as the detection of badly formed embedded commands. Errors returned directly by the Speech Synthesis Manager are not reported here.

If your application implements the `speechSynthesizer:didEncounterErrorAtIndex:ofString:message:delegate` message, the delegate message can use this property to get error information.

This property is used with `setObject:forProperty:error:` (page 16).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechInputModeProperty

Get or set the synthesizer’s current text-processing mode. An `NSString` object that specifies whether the channel is currently in text input mode or phoneme input mode. The supported values are listed in “[Speaking Modes for NSSpeechInputModeProperty](#)” (page 29).

When in phoneme-processing mode, a text string is interpreted to be a series of characters representing various phonemes and prosodic controls. Some synthesizers might support additional input-processing modes and define constants for these modes.

This property is used with `setObject:forProperty:error:` (page 16) and `objectForProperty:error:` (page 14).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechCharacterModeProperty

Get or set the synthesizer's current text-processing mode. An `NSString` object that specifies whether the channel is currently in text input mode or phoneme input mode. The supported values are listed in ["Speaking Modes for NSSpeechInputModeProperty"](#) (page 29).

When the character-processing mode is `NSSpeechModeNormal` (page 29), input characters are spoken as you would expect to hear them. When the mode is `NSSpeechModeLiteral` (page 29), each character is spoken literally, so that the word "cat" is spoken "C-A-T".

This property is used with [setObject:forProperty:error:](#) (page 16) and [objectForProperty:error:](#) (page 14).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechNumberModeProperty

Get or set the synthesizer's current number-processing mode. An `NSString` object that specifies whether the synthesizer is currently in normal or literal number-processing mode. The constants `NSSpeechModeNormal` (page 29) and `NSSpeechModeLiteral` (page 29) are the possible values of this string.

When the number-processing mode is `NSSpeechModeNormal` (page 29), the synthesizer assembles digits into numbers (so that "12" is spoken as "twelve"). When the mode is `NSSpeechModeLiteral` (page 29), each digit is spoken literally (so that "12" is spoken as "one, two").

This property is used with [setObject:forProperty:error:](#) (page 16) and [objectForProperty:error:](#) (page 14).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechRateProperty

Get or set the synthesizer's baseline speech pitch. An `NSNumber` object that specifies the synthesizer's baseline speech pitch.

Typical voice frequencies range from around 90 hertz for a low-pitched male voice to perhaps 300 hertz for a high-pitched child's voice. These frequencies correspond to approximate pitch values in the ranges of 30.000 to 40.000 and 55.000 to 65.000, respectively.

This property is used with [setObject:forProperty:error:](#) (page 16) and [objectForProperty:error:](#) (page 14).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechPitchBaseProperty

Get or set a synthesizer's baseline speech pitch. An `NSNumber` object that specifies the baseline speech pitch.

Typical voice frequencies range from around 90 hertz for a low-pitched male voice to perhaps 300 hertz for a high-pitched child's voice. These frequencies correspond to approximate pitch values in the ranges of 30.000 to 40.000 and 55.000 to 65.000, respectively.

This property is used with [setObject:forProperty:error:](#) (page 16) and [objectForProperty:error:](#) (page 14).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechPitchModProperty

Get or set a synthesizer's pitch modulation. An `NSNumber` object that specifies the synthesizer's pitch modulation.

Pitch modulation is also expressed as a floating-point value in the range of 0.000 to 127.000. These values correspond to MIDI note values, where 60.000 is equal to middle C on a piano scale. The most useful speech pitches fall in the range of 40.000 to 55.000. A pitch modulation value of 0.000 corresponds to a monotone in which all speech is generated at the frequency corresponding to the speech pitch. Given a speech pitch value of 46.000, a pitch modulation of 2.000 would mean that the widest possible range of pitches corresponding to the actual frequency of generated text would be 44.000 to 48.000.

This property is used with [setObject:forProperty:error:](#) (page 16) and [objectForProperty:error:](#) (page 14).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechVolumeProperty

Get or set the speech volume for a synthesizer. An `NSNumber` that specifies the synthesizer's speech volume.

Volumes are expressed in floating-point values ranging from 0.0 through 1.0. A value of 0.0 corresponds to silence, and a value of 1.0 corresponds to the maximum possible volume. Volume units lie on a scale that is linear with amplitude or voltage. A doubling of perceived loudness corresponds to a doubling of the volume.

This property is used with [setObject:forProperty:error:](#) (page 16) and [objectForProperty:error:](#) (page 14).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechSynthesizerInfoProperty

Get information about the speech synthesizer being used on the specified synthesizer. An `NSDictionary` object that contains information about the speech synthesizer being used on the specified synthesizer. See ["Speech Synthesizer Property Keys"](#) (page 24) for a description of the keys present in the dictionary.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechRecentSyncProperty

Get the message code for the most recently encountered synchronization command. An `NSNumber` object that specifies the most recently encountered synchronization command.

This property works with [setObject:forProperty:error:](#) (page 16).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechPhonemeSymbolsProperty

Get a list of phoneme symbols and example words defined for the synthesizer. An `NSDictionary` object that contains the phoneme symbols and example words defined for the current synthesizer.

Your application might use this information to show the user what symbols to use when entering phonemic text directly. See “[NSSpeechPhonemeSymbolsProperty Dictionary Keys](#)” (page 32) for a description of the keys present in the dictionary.

This property works with `setObject:forProperty:error:` (page 16).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechCurrentVoiceProperty

Set the current voice on the synthesizer to the specified voice. An `NSDictionary` object that contains the phoneme symbols and example words defined for the current synthesizer.

Your application might use this information to show the user what symbols to use when entering phonemic text directly. See “[NSSpeechPhonemeSymbolsProperty Dictionary Keys](#)” (page 32) for the keys you can use to specify values in this dictionary.

This property works with `setObject:forProperty:error:` (page 16).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechCommandDelimiterProperty

Set the embedded speech command delimiter characters to be used for the synthesizer. An `NSDictionary` object that contains the delimiter information. See “[Command Delimiter Keys](#)” for the keys you can use to specify values in this dictionary.

By default, the opening delimiter is “[” and the closing delimiter is “]”. Your application might need to change these delimiters temporarily if those character sequences occur naturally in a text buffer that is to be spoken. Your application can also disable embedded command processing by passing empty delimiters (as empty strings). See “[Speech Command Delimiter](#)” (page 33) for the keys you can use to specify values in this dictionary.

This property works with `setObject:forProperty:error:` (page 16).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechResetProperty

Set a synthesizer back to its default state. There is no value associated with this property; to reset the channel to its default state, set the key to `NULL`.

You can use this function to, for example, set speech pitch and speech rate to default values.

This property works with `setObject:forProperty:error:` (page 16).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechOutputToFileURLProperty

Set the speech output destination to a file or to the computer’s speakers. An `NSURL` object. To write the speech output to a file, use the file’s `NSURL`; to generate the sound through the computer’s speakers, use `NULL`.

This property works with `setObject:forProperty:error:` (page 16).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

Speaking Modes for NSSpeechInputModeProperty

These constants identify input modes are used with [NSSpeechInputModeProperty](#) (page 25).

```
NSString *const NSSpeechModeText;
NSString *const NSSpeechModePhoneme;
```

Constants

`NSSpeechModeText`

Indicates that the synthesizer is in text-processing mode.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechModePhoneme`

Indicates that the synthesizer is in phoneme-processing mode. When in phoneme-processing mode, a text buffer is interpreted to be a series of characters representing various phonemes and prosodic controls.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

Speaking Modes for NSSpeechNumberModeProperty

These constants define the available text-processing and number-processing modes for a synthesizer. These keys are used with [NSSpeechInputModeProperty](#) (page 25) and [NSSpeechNumberModeProperty](#) (page 26))

```
NSString *const NSSpeechModeNormal;
NSString *const NSSpeechModeLiteral;
```

Constants

`NSSpeechModeNormal`

Indicates that the synthesizer assembles digits into numbers (so that 12 is spoken as "twelve") and text into words.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechModeLiteral`

Indicates that each digit or character is spoken literally (so that 12 is spoken as "one, two", or the word "cat" is spoken as "C A T").

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechStatusProperty Dictionary Keys

These constants identify speech status keys used with [NSSpeechStatusProperty](#) (page 25).

```
NSString *const NSSpeechStatusOutputBusy;
NSString *const NSSpeechStatusOutputPaused;
NSString *const NSSpeechStatusNumberOfCharactersLeft;
NSString *const NSSpeechStatusPhonemeCode;
```

Constants

`NSSpeechStatusOutputBusy`

Indicates whether the synthesizer is currently producing speech.

A synthesizer is considered to be producing speech even at some times when no audio data is being produced through the computer's speaker. This occurs, for example, when the synthesizer is processing input, but has not yet initiated speech or when speech output is paused.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechStatusOutputPaused`

Indicates whether speech output in the synthesizer has been paused by sending the message [pauseSpeakingAtBoundary:](#) (page 14).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechStatusNumberOfCharactersLeft`

The number of characters left in the input string of text.

When the value of this key is zero, you can destroy the input string.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechStatusPhonemeCode`

Indicates that the synthesizer is in phoneme-processing mode. When in phoneme-processing mode, a text buffer is interpreted to be a series of characters representing various phonemes and prosodic controls.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechErrorProperty Dictionary Keys

These key constants identify errors that may occur during speech synthesis. They are used with [NSSpeechErrorsProperty](#) (page 25).

```
NSString *const NSSpeechErrorCount;
NSString *const NSSpeechErrorOldestCode;
NSString *const NSSpeechErrorOldestCharacterOffset;
NSString *const NSSpeechErrorNewestCode;
NSString *const NSSpeechErrorNewestCharacterOffset;
```

Constants

`NSSpeechErrorCount`

The number of errors that have occurred in processing the current text string, since the last call to [objectForProperty:error:](#) (page 14) with the [NSSpeechErrorsProperty](#) (page 25) property. An NSInteger

Using the [NSSpeechErrorOldestCode](#) (page 31) keys and the [NSSpeechErrorNewestCode](#) (page 31) keys, you can get information about the oldest and most recent errors that occurred since the last call to [objectForProperty:error:](#) (page 14), but you cannot get information about any intervening errors.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechErrorOldestCode`

The error code of the first error that occurred since the last call to [objectForProperty:error:](#) (page 14) with the [NSSpeechErrorsProperty](#) (page 25) property. An NSInteger

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechErrorOldestCharacterOffset`

The position in the text string of the first error that occurred since the last call to [objectForProperty:error:](#) (page 14) with the [NSSpeechErrorsProperty](#) (page 25) property. An NSInteger

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechErrorNewestCode`

The error code of the most recent error that occurred since the last call to [objectForProperty:error:](#) (page 14) with the [NSSpeechErrorsProperty](#) (page 25) property. An NSInteger

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechErrorNewestCharacterOffset`

The position in the text string of the most recent error that occurred since the last call to [objectForProperty:error:](#) (page 14) with the [NSSpeechErrorsProperty](#) (page 25) property. An NSInteger.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechSynthesizerInfoProperty Dictionary Keys

These constants are keys used in the `NSSpeechSynthesizerInfoProperty` dictionary.

```
NSString *const NSSpeechSynthesizerInfoIdentifier;
NSString *const NSSpeechSynthesizerInfoVersion;
```

Constants

```
NSSpeechSynthesizerInfoIdentifier
```

The identifier of the speech synthesizer.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

```
NSSpeechSynthesizerInfoVersion
```

The version of the speech synthesizer.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

NSSpeechPhonemeSymbolsProperty Dictionary Keys

These constants are keys used in the NSSpeechPhonemeSymbolsProperty dictionary.

```
NSString *const NSSpeechPhonemeInfoOpcode
NSString *const NSSpeechPhonemeInfoSymbol;
NSString *const NSSpeechPhonemeInfoExample;
NSString *const NSSpeechPhonemeInfoHiliteStart;
NSString *const NSSpeechPhonemeInfoHiliteEnd;
```

Constants

```
NSSpeechPhonemeInfoOpcode
```

NSNumber

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

```
NSSpeechPhonemeInfoSymbol
```

The symbol used to represent the phoneme.

The symbol does not necessarily have a phonetic connection to the phoneme, but might simply be an abstract textual representation of it.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

```
NSSpeechPhonemeInfoExample
```

An example word that illustrates the use of the phoneme.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

```
NSSpeechPhonemeInfoHiliteStart
```

The character offset into the example word that identifies the location of the beginning of the phoneme.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

```
NSSpeechPhonemeInfoHiliteEnd
```

The character offset into the example word that identifies the location of the end of the phoneme.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

Speech Command Delimiter Keys

These constants speech-command delimiters keys used in [NSSpeechCommandDelimiterProperty](#) (page 28).

```
NSString *const NSSpeechCommandPrefix;
NSString *const NSSpeechCommandSuffix;
```

Constants

`NSSpeechCommandPrefix`

The command delimiter string that prefixes a command, by default, this is `[]`.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechCommandSuffix`

The command delimiter string that suffixes a command, by default, this is `][]`.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

Speech Dictionary Properties Keys

These constants identify key-value pairs used to add vocabulary to the dictionary using [addSpeechDictionary:](#) (page 11).

```
NSString *const NSSpeechDictionaryLocaleIdentifier;
NSString *const NSSpeechDictionaryModificationDate;
NSString *const NSSpeechDictionaryPronunciations;
NSString *const NSSpeechDictionaryAbbreviations;
NSString *const NSSpeechDictionaryEntrySpelling;
NSString *const NSSpeechDictionaryEntryPhonemes;
```

Constants

`NSSpeechDictionaryLocaleIdentifier`

The canonical locale identifier string describing the dictionary's locale. A locale is generally composed of three pieces of ordered information: a language code, a region code, and a variant code. Refer to documentation about `NSLocale` or *Locales Programming Guide* for more information

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechDictionaryModificationDate`

A string representation of the dictionary's last modification date in the international format (YYYY-MM-DD HH:MM:SS ±HHMM). If the same word appears across multiple dictionaries, the one from the dictionary with the most recent date will be used.

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

`NSSpeechDictionaryPronunciations`

An array of dictionary objects containing the keys [NSSpeechDictionaryEntrySpelling](#) (page 34) and [NSSpeechDictionaryEntryPhonemes](#) (page 34).

Available in Mac OS X v10.5 and later.

Declared in `NSSpeechSynthesizer.h`.

NSSpeechDictionaryAbbreviations

An array of dictionary objects containing the keys [NSSpeechDictionaryEntrySpelling](#) (page 34) and [NSSpeechDictionaryEntryPhonemes](#) (page 34).

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

NSSpeechDictionaryEntrySpelling

The spelling of an entry. An NSString.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

NSSpeechDictionaryEntryPhonemes

The phonemic representation of an entry. An NSString.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

NSSpeechBoundary

These constants are used to indicate where speech should be stopped and paused. See [pauseSpeakingAtBoundary:](#) (page 14) and [stopSpeakingAtBoundary:](#) (page 20).

```
enum {
    NSSpeechImmediateBoundary = 0,
    NSSpeechWordBoundary,
    NSSpeechSentenceBoundary
};
typedef NSUInteger NSSpeechBoundary;
```

Constants

NSSpeechImmediateBoundary

Speech should be paused or stopped immediately.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

NSSpeechWordBoundary

Speech should be paused or stopped at the end of the word.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

NSSpeechSentenceBoundary

Speech should be paused or stopped at the end of the sentence.

Available in Mac OS X v10.5 and later.

Declared in NSSpeechSynthesizer.h.

Document Revision History

This table describes the changes to *NSSpeechSynthesizer Class Reference*.

Date	Notes
2009-08-20	Updated for Mac OS X v10.6. Delegate methods moved to NSSpeechSynthesizerDelegate Protocol Reference. Documented constants.
2009-01-06	Clarified how the speech rate is set when using initWithVoice:.
2007-07-23	Updated for Mac OS X v10.5
2006-07-24	Clarified role and usage of the NSVoiceIdentifier constant. Added identifiers for the built-in voices.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

`addSpeechDictionary`: instance method 11
`attributesForVoice`: class method 10
`availableVoices` class method 10

C

`continueSpeaking` instance method 12

D

`defaultVoice` class method 11
`delegate` instance method 12

I

`initWithVoice`: instance method 13
`isAnyApplicationSpeaking` class method 11
`isSpeaking` instance method 13

N

`NSSpeechBoundary` 34
`NSSpeechCharacterModeProperty` constant 26
`NSSpeechCommandDelimiterProperty` constant 28
`NSSpeechCommandPrefix` constant 33
`NSSpeechCommandSuffix` constant 33
`NSSpeechCurrentVoiceProperty` constant 28
`NSSpeechDictionaryAbbreviations` constant 34
`NSSpeechDictionaryEntryPhonemes` constant 34
`NSSpeechDictionaryEntrySpelling` constant 34
`NSSpeechDictionaryLocaleIdentifier` constant 33
`NSSpeechDictionaryModificationDate` constant 33
`NSSpeechDictionaryPronunciations` constant 33

`NSSpeechErrorCount` constant 31
`NSSpeechErrorNewestCharacterOffset` constant 31
`NSSpeechErrorNewestCode` constant 31
`NSSpeechErrorOldestCharacterOffset` constant 31
`NSSpeechErrorOldestCode` constant 31
`NSSpeechErrorProperty Dictionary Keys` 30
`NSSpeechErrorsProperty` constant 25
`NSSpeechImmediateBoundary` constant 34
`NSSpeechInputModeProperty` constant 25
`NSSpeechModeLiteral` constant 29
`NSSpeechModeNormal` constant 29
`NSSpeechModePhoneme` constant 29
`NSSpeechModeText` constant 29
`NSSpeechNumberModeProperty` constant 26
`NSSpeechOutputToFileURLProperty` constant 28
`NSSpeechPhonemeInfoExample` constant 32
`NSSpeechPhonemeInfoHiliteEnd` constant 32
`NSSpeechPhonemeInfoHiliteStart` constant 32
`NSSpeechPhonemeInfoOpcode` constant 32
`NSSpeechPhonemeInfoSymbol` constant 32
`NSSpeechPhonemeSymbolsProperty` constant 28
`NSSpeechPhonemeSymbolsProperty Dictionary Keys` 32
`NSSpeechPitchBaseProperty` constant 26
`NSSpeechPitchModProperty` constant 27
`NSSpeechRateProperty` constant 26
`NSSpeechRecentSyncProperty` constant 27
`NSSpeechResetProperty` constant 28
`NSSpeechSentenceBoundary` constant 34
`NSSpeechStatusNumberOfCharactersLeft` constant 30
`NSSpeechStatusOutputBusy` constant 30
`NSSpeechStatusOutputPaused` constant 30
`NSSpeechStatusPhonemeCode` constant 30
`NSSpeechStatusProperty` constant 25
`NSSpeechStatusProperty Dictionary Keys` 29
`NSSpeechSynthesizerInfoIdentifier` constant 32
`NSSpeechSynthesizerInfoProperty` constant 27
`NSSpeechSynthesizerInfoProperty Dictionary Keys` 31
`NSSpeechSynthesizerInfoVersion` constant 32
`NSSpeechVolumeProperty` constant 27
`NSSpeechWordBoundary` constant 34
`NSVoiceAge` constant 22

NSVoiceDemoText **constant** [22](#)
 NSVoiceGender **constant** [22](#)
 NSVoiceGenderFemale **constant** [24](#)
 NSVoiceGenderMale **constant** [24](#)
 NSVoiceGenderNeuter **constant** [24](#)
 NSVoiceIdentifier **constant** [22](#)
 NSVoiceIndividuallySpokenCharacters **constant** [23](#)
 NSVoiceLanguage **constant** (**Deprecated in Mac OS X v10.5**) [23](#)
 NSVoiceLocaleIdentifier **constant** [23](#)
 NSVoiceName **constant** [22](#)
 NSVoiceSupportedCharacters **constant** [23](#)

O

objectForProperty:error: **instance method** [14](#)

P

pauseSpeakingAtBoundary: **instance method** [14](#)
 phonemesFromText: **instance method** [15](#)

R

rate **instance method** [15](#)

S

setDelegate: **instance method** [16](#)
 setObject:forProperty:error: **instance method** [16](#)
 setRate: **instance method** [16](#)
 setUsesFeedbackWindow: **instance method** [17](#)
 setVoice: **instance method** [17](#)
 setVolume: **instance method** [18](#)
 Speaking Modes for NSSpeechInputModeProperty [29](#)
 Speaking Modes for NSSpeechNumberModeProperty [29](#)
 Speech Command Delimiter Keys [33](#)
 Speech Dictionary Properties Keys [33](#)
 Speech Synthesizer Property Keys [24](#)
 startSpeakingString: **instance method** [18](#)
 startSpeakingString:toURL: **instance method** [19](#)
 stopSpeaking **instance method** [20](#)
 stopSpeakingAtBoundary: **instance method** [20](#)

U

usesFeedbackWindow **instance method** [21](#)

V

Voice Attributes Keys [22](#)
 Voice Gender Keys [24](#)
 voice **instance method** [21](#)
 volume **instance method** [21](#)