Speech Recognition Manager Reference

User Experience > Speech Technologies

2003-02-01

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

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

Speech Recognition Manager Reference

Framework:	Carbon/Carbon.h
Declared in	SpeechRecognition.h

Overview

The Speech Recognition Manager provides speech recognition support in applications.

Functions by Task

Opening and Closing Recognition Systems

`SRCloseRecognitionSystem` (page 14)
: Closes a recognition system when your application is finished using it (for example, just before your application quits).

`SROpenRecognitionSystem` (page 27)
: Opens a recognition system.

Creating and Manipulating Recognizers

`SRCancelRecognition` (page 12)
: Cancels the attempt to recognize the current utterance.

`SRContinueRecognition` (page 14)
: Causes a recognizer to continue recognizing speech.

`SRGetLanguageModel` (page 18)
: Gets a recognizer's active language model.

`SRIdle` (page 21)
: Grants processing time to the Speech Recognition Manager if your application does not call `WaitNextEvent` frequently.

`SRNewRecognizer` (page 25)
: Creates a new recognizer.

`SRSetLanguageModel` (page 33)
: Sets a recognizer's active language model.

`SRStartListening` (page 36)
: Starts a recognizer listening and reporting results to your application.

`SRStopListening` (page 37)
Stops a recognizer listening and reporting results to your application.

Managing Speech Objects

`SRGetProperty` (page 19)
Gets the current value of a property of a speech object.

`SRGetReference` (page 20)
Obtains an extra reference to a speech object.

`SRReleaseObject` (page 30)
Releases a speech object.

`SRSetProperty` (page 33)
Sets the value of a property of a speech object.

Traversing Speech Objects

`SRCountItems` (page 15)
Determines the number of subitems in a container object.

`SRGetIndexedItem` (page 17)
Gets a subitem in a container object.

`SRRemoveIndexedItem` (page 31)
Removes a subitem from a container object.

`SRSetIndexedItem` (page 32)
Replaces a subitem in a container object with some other object.

Creating Language Objects

`SRNewLanguageModel` (page 21)
Creates a new language model.

`SRNewPath` (page 24)
Creates a new path.

`SRNewPhrase` (page 25)
Creates a new phrase.

`SRNewWord` (page 26)
Creates a new word.

Manipulating Language Objects

`SRAddLanguageObject` (page 11)
Adds a language object to some other language object.

`SRAddText` (page 12)
Adds text to the contents of a language object.

`SRChangeLanguageObject` (page 13)
Changes the contents of a language object.

`SREmptyLanguageObject` (page 17)
Empties the contents of a language object.

`SRNewLanguageObjectFromHandle` (page 23)
Creates a language object from the handle previously created by the `SRPutLanguageObjectIntoHandle` function.

`SRNewLanguageObjectFromDataFile` (page 22)
Reads a language object from a data file.

`SRPutLanguageObjectIntoHandle` (page 30)
Puts a language object (and any embedded languages objects it contains) into a handle.

`SRPutLanguageObjectIntoDataFile` (page 29)
Puts a language object (and any embedded language objects it contains) into a data file.

`SRRemoveLanguageObject` (page 31)
Removes a language object from another language object that contains it.

Using the System Feedback Window

`SRDrawRecognizedText` (page 15)
Draws recognized text in the feedback window.

`SRDrawText` (page 16)
Draws output text in the feedback window.

`SRProcessBegin` (page 27)
Indicates that a recognition result is being processed.

`SRProcessEnd` (page 28)
Indicates that a recognition result is done being processed.

`SRSpeakAndDrawText` (page 34)
Draws output text in the feedback window and causes the feedback character in the feedback window to speak that text.

`SRSpeakText` (page 35)
Causes the feedback character in the feedback window to speak a text string.

`SRSpeechBusy` (page 36)
Determines if the feedback character in a feedback window is currently speaking.

`SRStopSpeech` (page 37)
Terminates speech by the feedback character in a feedback window.

Creating, Invoking and Disposing UPPs

`NewSRCallBackUPP` (page 10)
Creates a new universal procedure pointer (UPP) to a speech recognition callback function.

`DisposeSRCallBackUPP` (page 10)
Disposes of a universal procedure pointer (UPP) to a speech recognition callback function.

`InvokeSRCallBackUPP` (page 10)
Invokes your speech recognition callback function.

Functions

DisposeSRCallBackUPP

Disposes of a universal procedure pointer (UPP) to a speech recognition callback function.

```
void DisposeSRCallBackUPP (
   SRCallBackUPP userUPP
);
```

Parameters

userUPP

The UPP to dispose of.

Availability

Available in CarbonLib 1.0.2 and later.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

InvokeSRCallBackUPP

Invokes your speech recognition callback function.

```
void InvokeSRCallBackUPP (
   SRCallBackStruct *param,
   SRCallBackUPP userUPP
);
```

Discussion

You should not have to call the `InvokeSRCallbackUPP` function, as the system calls your speech recognition callback function for you.

Availability

Available in CarbonLib 1.0.2 and later.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

NewSRCallBackUPP

Creates a new universal procedure pointer (UPP) to a speech recognition callback function.

```
SRCallBackUPP NewSRCallBackUPP (
   SRCallBackProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your speech recognition callback function.

Return Value

A UPP to the speech recognition callback function. See the description of the SRCallBackUPP data type.

Availability

Available in CarbonLib 1.0.2 and later.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRAddLanguageObject

Adds a language object to some other language object.

```
OSErr SRAddLanguageObject (
   SRLanguageObject base,
   SRLanguageObject addon
);
```

Parameters

base

The language object to which to add the language object specified by the addon parameter.

addon

The language object to add on to the language object specified in the base parameter. For example, if addon specifies a word and base specifies a phrase, then SRAddLanguageObject appends that word to the end of that phrase.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRAddLanguageObject function is useful for adding language objects to phrases, paths, and language models. For a phrase or a path, SRAddLanguageObject appends the specified object to the end of the phrase or path. For a language model, SRAddLanguageObject adds the specified object to the list of alternative recognizable utterances.

The language object to which you add an object acquires a new reference to that object. Accordingly, any changes you subsequently make to the added object are reflected in any object to which you added it. The base object releases its reference to the added object when the base object is disposed of.

SRAddLanguageObject does not alter the value of the reference constant property of the language object specified by the base parameter.

See SRAddText (page 12) for a useful shortcut function.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRAddText

Adds text to the contents of a language object.

```
OSErr SRAddText (
   SRLanguageObject base,
   const void *text,
   SInt32 textLength,
   SRefCon refCon
);
```

Parameters

base

A language object to which to add the text.

text

A pointer to a buffer that contains the words or phrase to add to the contents of the specified language object.

textLength

The size, in bytes, of the specified text.

refCon

An application-defined reference constant. The value of the reference constant property of the new word or phrase representing the specified text is set to this value.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRAddText function is useful for phrases, paths, and language models. If the base parameter specifies a path or language model, SRAddText is equivalent to calling SRNewPhrase, SRAddLanguageObject, and SRReleaseObject for the phrase specified by the text parameter and calling SRSetProperty to reset the value of the reference constant property of the new phrase.

If the base parameter specifies a phrase, SRAddText is equivalent to calling SRNewPhrase, SRAddLanguageObject, and SRReleaseObject for each distinguishable word in the text parameter and calling SRSetProperty to set the value of the reference constant property of the new words.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRCancelRecognition

Cancels the attempt to recognize the current utterance.

```
OSErr SRCancelRecognition (
   SRRecognizer recognizer
);
```

Parameters

recognizer

A recognizer.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRCancelRecognition function instructs the recognizer specified by the recognizer parameter to stop recognizing speech. You need to call either SRContinueRecognition or SRCancelRecognition each time your application is notified that the user has started speaking (using Apple events or through an application-defined callback routine).

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRChangeLanguageObject

Changes the contents of a language object.

```
OSErr SRChangeLanguageObject (
   SRLanguageObject languageObject,
   const void *text,
   SInt32 textLength
);
```

Parameters

languageObject

A language object.

text

A pointer to a buffer that contains the words or phrase to which the contents of the specified language object are to be changed.

textLength

The size, in bytes, of the specified text.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

SRChangeLanguageObject is a convenient shortcut for calling SREmptyLanguageObject and then SRAddText.

SRChangeLanguageObject does not alter the value of the reference constant property of the language object specified by the languageObject parameter.

If there are no other references to the language object specified by the languageObject parameter, calling SRChangeLanguageObject causes that object to be released.

If you want to swap rapidly among several language models, you should use the SRSetLanguageObject function instead of SRChangeLanguageObject. Or, you could use the kSREnabled property to rapidly enable and disable parts of the current language model to reflect the current context.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
`SpeechRecognition.h`

SRCloseRecognitionSystem

Closes a recognition system when your application is finished using it (for example, just before your application quits).

```
OSErr SRCloseRecognitionSystem (
   SRRecognitionSystem system
);
```

Parameters

`system`

A recognition system.

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
The `SRCloseRecognitionSystem` function closes the recognition system specified by the `system` parameter. If any speech objects are still attached to that recognition system, they are disposed of and any references you have to those objects are thereby rendered invalid.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
`SpeechRecognition.h`

SRContinueRecognition

Causes a recognizer to continue recognizing speech.

```
OSErr SRContinueRecognition (
   SRRecognizer recognizer
);
```

Parameters

`recognizer`

A recognizer.

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
You need to call either `SRContinueRecognition` or `SRCancelRecognition` each time your application is notified that the user has started speaking (using Apple events or through an application-defined callback routine).

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
`SpeechRecognition.h`

SRCountItems

Determines the number of subitems in a container object.

```
OSErr SRCountItems (
   SRSpeechObject container,
   long *count
);
```

Parameters

container

A speech object.

count

On return, a pointer to a long containing the number of subitems in the specified speech object.

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
This function is useful only for speech objects that have distinguishable subitems, such as phrases (which contain words), paths (which contain words, phrases, and language models), and language models (which contain words, phrases, paths, and possibly other language models).

Version Notes
In Speech Recognition Manager version 1.5, this function is useful only for operating on language objects (of type `SRLanguageObject`), although it is defined for all speech objects.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
`SpeechRecognition.h`

SRDrawRecognizedText

Draws recognized text in the feedback window.

```
OSErr SRDrawRecognizedText (
   SRRecognizer recognizer,
   const void *dispText,
   SInt32 dispLength
);
```

Parameters

recognizer

A recognizer.

dispText

A pointer to a buffer that contains the text to be drawn.

dispLength

The size, in bytes, of the specified text.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRDrawRecognizedText function draws the text specified by the dispText and dispLength parameters in the transcript portion of the feedback window associated with the recognizer specified by the recognizer parameter. The text is drawn in the style characteristic of all recognized text. You might want to use this function to display a recognized phrase using a different spelling than the one used in the language model.

If the value of the kSRWantsResultTextDrawn property of the specified recognizer is TRUE (which is the default value), a transcript of the text of a recognition result is automatically sent directly to the feedback window. As a result, you should call SRDrawRecognizedText only when the value of the recognizer's kSRWantsResultTextDrawn property is FALSE.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRDrawText

Draws output text in the feedback window.

```
OSErr SRDrawText (
   SRRecognizer recognizer,
   const void *dispText,
   SInt32 dispLength
);
```

Parameters

recognizer

A recognizer.

dispText

A pointer to a buffer that contains the text to be drawn.

dispLength

The size, in bytes, of the specified text.

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
The `SRDrawText` function draws the text specified by the `dispText` and `dispLength` parameters in the transcript portion of the feedback window associated with the recognizer specified by the `recognizer` parameter. The text is drawn in the style characteristic of all output text.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
`SpeechRecognition.h`

SREmptyLanguageObject

Empties the contents of a language object.

```
OSErr SREmptyLanguageObject (
   SRLanguageObject languageObject
);
```

Parameters

`languageObject`

A language object.

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
The `SREmptyLanguageObject` function empties the contents of the language object specified by the `languageObject` parameter. (For example, if `languageObject` specifies a phrase containing two words, calling `SREmptyLanguageObject` would result in a phrase that contains no words.) Any properties of that object that are not related to its contents are unchanged. In particular, `SREmptyLanguageObject` does not alter the value of the reference constant property of that language object.

If there are no other references to the words, phrases, and paths that were contained in the language object, calling `SREmptyLanguageObject` causes them to be disposed of.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
`SpeechRecognition.h`

SRGetIndexedItem

Gets a subitem in a container object.

```
OSErr SRGetIndexedItem (
   SRSpeechObject container,
   SRSpeechObject *item,
   long index
);
```

Parameters

container

A speech object.

item

On return, a reference to the subitem in the specified speech object that has the specified index.

index

An integer ranging from 0 to one less than the number of subitems in the specified speech object. (You can call the `SRCountItems` function to determine the number of subitems contained in a speech object.) If the index you specify is not in this range, `SRGetIndexedItem` returns the result code `kSRParamOutOfRange`.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

This function is useful for iterating through all subitems in a container object.

`SRGetIndexedItem` increases the reference count of the specified speech object. You should call the `SRReleaseObject` function to release the object reference returned by `SRGetIndexedItem` when you are done using it. For example, you can get a reference to the third word in a phrase by executing this code:

```
myErr = SRGetIndexedItem(myPhrase, &myWord, 2)
```

Then, when you are finished using the word, you should execute this code:

```
myErr = SRReleaseObject(myWord);
```

Version Notes

In Speech Recognition Manager version 1.5, this function is useful only for operating on language objects (of type `SRLanguageObject`), although it is defined for all speech objects.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRGetLanguageModel

Gets a recognizer's active language model.

```
OSErr SRGetLanguageModel (
   SRRecognizer recognizer,
   SRLanguageModel *languageModel
);
```

Parameters

recognizer

A recognizer.

languageModel

On return, a reference to the language model currently active for the specified recognizer.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

`SRGetLanguageModel` increases the reference count of the specified language model. You should call the `SRReleaseObject` function to release the language model reference returned by `SRGetLanguageModel` when you are done using it.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRGetProperty

Gets the current value of a property of a speech object.

```
OSErr SRGetProperty (
   SRSpeechObject srObject,
   OSType selector,
   void *property,
   Size *propertyLen
);
```

Parameters

srObject

A speech object.

selector

A property selector. See "Recognizer Properties" (page 54), "Recognizer Listen Key Properties" (page 53), "Language Object Properties" (page 47), and "Recognition System Properties" (page 52) for lists of the available property selectors.

property

A pointer to a buffer into which the value of the specified property is to be copied.

propertyLen

On entry, a pointer to the length, in bytes, of the specified buffer. If the value is of a fixed size, then `propertyLen` should point to a variable of type `Size` that specifies that size. If the size of the value can vary (for example, if the value is a string), then `propertyLen` should point to a variable of type `Size` that specifies the number of bytes in the buffer pointed to by the `property` parameter.

On return, if the buffer is large enough to hold the returned property value and no error occurs, `SRGetProperty` sets `propertyLen` to the total number of bytes in the value of the specified property. If the buffer is not large enough to hold the returned property value, `SRGetProperty` sets `propertyLen` to the number of bytes required to store the requested property and returns the `kSRBufferTooSmall` error code.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

Not all selectors are valid for all types of speech objects. If the selector you specify does not specify a property of the specified speech object, `SRGetProperty` returns the result code `kSRCantGetProperty`.

If `SRGetProperty` returns an object reference, you must make sure to release that object reference (by calling `SRReleaseObject`) when you are finished using it. Most selectors do not cause `SRGetProperty` to return object references. For example, passing the selector `kSRSpelling` causes `SRGetProperty` to return a buffer of text, not an object reference.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRGetReference

Obtains an extra reference to a speech object.

```
OSErr SRGetReference (
   SRSpeechObject srObject,
   SRSpeechObject *newObjectRef
);
```

Parameters

srObject

A speech object.

newObjectRef

On return, a new reference to the specified speech object.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The original object reference (contained in `srObject`) and the new reference (returned in `newObjectRef`) may have different values. Accordingly, you cannot simply compare two object references to determine whether they are references to the same speech object.

SRGetReference increases the reference count of the specified speech object. You should call the SRReleaseObject function to release the object reference returned by SRGetReference when you are done using it.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.
Available in Mac OS X 10.0 and later.

Declared In
SpeechRecognition.h

SRIdle

Grants processing time to the Speech Recognition Manager if your application does not call WaitNextEvent frequently.

```
OSErr SRIdle (
   void
);
```

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
The SRIdle function grants processing time to the Speech Recognition Manager, thereby allowing it to process incoming sound and send recognition results.

Most applications do not need to call the SRIdle function. You need to call it only if your application does a significant amount of processing without periodically calling WaitNextEvent. If you do use the SRIdle function, you should call it often enough that the Speech Recognition Manager can perform its work.

Note, however, that if you call SRIdle and not WaitNextEvent, you give time to the recognizer but not to the feedback window. You must call WaitNextEvent periodically to have the feedback animations work correctly if your recognizer is using the standard feedback window.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.
Available in Mac OS X 10.0 and later.

Declared In
SpeechRecognition.h

SRNewLanguageModel

Creates a new language model.

```
OSErr SRNewLanguageModel (
   SRRecognitionSystem system,
   SRLanguageModel *model,
   const void *name,
   SInt32 nameLength
);
```

Parameters

system

A recognition system.

model

On return, a reference to a new empty language model associated with the specified recognition system.

name

A pointer to a buffer that contains the name of the language model. The name of the language model should be unique among all the language models your application creates, and it should be comprehensible to users. (For example, a language model that defined a list of names might be called "«Names»").

The convention that language model names begin with the character "«" and end with the character "»" is adopted to support future utilities that display the names of language models to the user (perhaps as part of showing the user what he or she can say).

nameLength

The size, in bytes, of the specified name.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

You can add language objects (that is, words, phrases, paths, and other language models) to a language model by calling the `SRAddText` and `SRAddLanguageObject` functions.

`SRNewLanguageModel` sets the reference count of the specified language model to 1. You should call the `SRReleaseObject` function to release the language model reference returned by `SRNewLanguageModel` when you are done using it.

You can get or set the name of an existing language model by calling the `SRGetProperty` or `SRSetProperty` functions with the `kSRSpelling` property selector.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRNewLanguageObjectFromDataFile

Reads a language object from a data file.

```
OSErr SRNewLanguageObjectFromDataFile (
   SRRecognitionSystem system,
   SRLanguageObject *languageObject,
   short fRefNum
);
```

Parameters

system

A recognition system.

languageObject

On return, a reference to a new language object whose description is stored in the open data file that has the file reference number specified by the fRefNum parameter.

fRefNum

A file reference number of an open data file.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

SRNewLanguageObjectFromDataFile reads data beginning at the current file mark.

If the language object is successfully created and initialized, the file mark is left at the byte immediately following the language object description. Otherwise, if the language object data is not appropriately formatted, SRNewLanguageObjectFromDataFile returns the result code kSRCantReadLanguageObject as its function result and the file mark is not moved.

You should call the SRReleaseObject function to release the language object reference returned by SRNewLanguageObjectFromDataFile when you are done using it.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRNewLanguageObjectFromHandle

Creates a language object from the handle previously created by the SRPutLanguageObjectIntoHandle function.

```
OSErr SRNewLanguageObjectFromHandle (
   SRRecognitionSystem system,
   SRLanguageObject *languageObject,
   Handle lObjHandle
);
```

Parameters

system

A recognition system.

languageObject

On return, a reference to a new language object created and initialized using the private data to which the lobjHandle parameter is a handle.

lObjHandle

A handle to a language object. The data specified by `lobjHandle` should have been created by a previous call to the `SRPutLanguageObjectIntoHandle` function; if that data is not appropriately formatted, `SRNewLanguageObjectFromHandle` returns the result code `kSRCantReadLanguageObject` as its function result.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

You can use this function to load language objects from resources (for example, by using the Resource Manager function `GetResource`).

You should call the `SRReleaseObject` function to release the language object reference returned by `SRNewLanguageObjectFromHandle` when you are done using it.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRNewPath

Creates a new path.

```
OSErr SRNewPath (
   SRRecognitionSystem system,
   SRPath *path
);
```

Parameters

system

A recognition system.

path

On return, a reference to a new empty path associated with the specified recognition system.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

You can then add objects to a path by calling the `SRAddText` or `SRAddLanguageObject` functions.

You should call the `SRReleaseObject` function to release the path reference returned by `SRNewPath` when you are done using it.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRNewPhrase

Creates a new phrase.

```
OSErr SRNewPhrase (
   SRRecognitionSystem system,
   SRPhrase *phrase,
   const void *text,
   SInt32 textLength
);
```

Parameters

system

A recognition system.

phrase

On return, a reference to a new phrase associated with the specified recognition system.

text

A pointer to a buffer that contains the words that comprise the phrase.

textLength

The size, in bytes, of the specified text.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The phrase's contents (that is, the words that comprise the phrase) is specified by the `text` and `textLength` parameters. You can, if you wish, create a new empty phrase and then add words to it by calling the `SRAddText` or `SRAddLanguageObject` functions.

You should call the `SRReleaseObject` function to release the phrase reference returned by `SRNewPhrase` when you are done using it.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRNewRecognizer

Creates a new recognizer.

```
OSErr SRNewRecognizer (
   SRRecognitionSystem system,
   SRRecognizer *recognizer,
   OSType sourceID
);
```

Parameters

system

A recognition system.

recognizer

On return, a reference to a new recognizer associated with the specified recognition system and using the specified speech source.

sourceID

A speech source ID. See "Speech Source Constants" (page 57).

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

`SRNewRecognizer` may need to load substantial amounts of data from disk into memory. As a result, you might want to change the cursor to the watch cursor before you call `SRNewRecognizer`.

You should call the `SRReleaseObject` function to release the object reference returned by `SRNewRecognizer` when you are done using it.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRNewWord

Creates a new word.

```
OSErr SRNewWord (
   SRRecognitionSystem system,
   SRWord *word,
   const void *text,
   SInt32 textLength
);
```

Parameters

system

A recognition system.

word

On return, a reference to a new word associated with the specified recognition system.

text

A pointer to a buffer that contains the characters that comprise the word.

textLength

The size, in bytes, of the specified text.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

You should call the `SRReleaseObject` function to release the word reference returned by `SRNewWord` when you are done using it.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
`SpeechRecognition.h`

SROpenRecognitionSystem

Opens a recognition system.

```
OSErr SROpenRecognitionSystem (
   SRRecognitionSystem *system,
   OSType systemID
);
```

Parameters

`system`

On return, a reference to the recognition system having the specified system ID.

`systemID`

A recognition system ID.

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
Generally, you should open a single recognition system when your application starts up and close it (by calling the function `SRCloseRecognitionSystem`) before your application exits.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
`SpeechRecognition.h`

SRProcessBegin

Indicates that a recognition result is being processed.

```
OSErr SRProcessBegin (
   SRRecognizer recognizer,
   Boolean failed
);
```

Parameters

`recognizer`

A recognizer.

`failed`

A Boolean value that determines how the feedback gestures are to be altered and whether the response sound is to be played (`FALSE`) or not (`TRUE`).

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRProcessBegin function causes the Speech Recognition Manager to provide the relevant feedback (in the feedback window associated with the recognizer specified by the recognizer parameter) indicating that the application is in the process of responding to a spoken command. Currently, the gestures of the feedback character are changed to indicate that processing is occurring.

If you set the value of the recognizer's kSRWantsAutoFBGestures property to FALSE, you should call SRProcessBegin at the beginning of your response to a recognition result and SRProcessEnd at the end of your response. During the interval separating the two calls, the feedback character displays an appropriate set of gestures showing the user that the task is being processed. If you pass the value TRUE in the failed parameter (indicating that the recognition result cannot successfully be processed), the feedback character displays frowns, shrugs, or other appropriate gestures. In addition, when failed is TRUE, you do not need to call SRProcessEnd to end the processing. If you pass the value FALSE in the failed parameter but determine subsequently that the recognition result cannot successfully be processed, you should call SRProcessEnd with the failed parameter set to TRUE.

If the value of the kSRWantsAutoFBGestures property of the specified recognizer is TRUE, the Speech Recognition Manager calls SRProcessBegin internally before notifying your application of a recognition result, and it calls SRProcessEnd internally after your application is notified. As a result, you should call SRProcessBegin or SRProcessEnd only when the value of the recognizer's kSRWantsAutoFBGestures property is FALSE.

Because the default value of the kSRWantsAutoFBGestures property is TRUE, most applications do not need to call SRProcessBegin. Calling SRProcessBegin is useful, however, when you know the resulting action might take a significant amount of time.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRProcessEnd

Indicates that a recognition result is done being processed.

```
OSErr SRProcessEnd (
   SRRecognizer recognizer,
   Boolean failed
);
```

Parameters

recognizer

A recognizer.

failed

A Boolean value that determines how the feedback gestures are to be altered (FALSE) or not (TRUE).

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
The SRProcessEnd function causes the Speech Recognition Manager to provide the relevant feedback (in the feedback window associated with the recognizer specified by the recognizer parameter) indicating that a recognition result is done being processed. Currently, the gestures of the feedback character are changed and a response sound is played.

If the value of the kSRWantsAutoFBGestures property of the specified recognizer is TRUE, the Speech Recognition Manager calls SRProcessBegin internally before notifying your application of a recognition result, and it calls SRProcessEnd internally after your application is notified. As a result, you should call SRProcessBegin or SRProcessEnd only when the value of the recognizer's kSRWantsAutoFBGestures property is FALSE.

Because the default value of the kSRWantsAutoFBGestures property is TRUE, most applications do not need to call SRProcessBegin. Calling SRProcessBegin is useful, however, when you know the resulting action might take a significant amount of time.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.
Available in Mac OS X 10.0 and later.

Declared In
SpeechRecognition.h

SRPutLanguageObjectIntoDataFile

Puts a language object (and any embedded language objects it contains) into a data file.

```
OSErr SRPutLanguageObjectIntoDataFile (
   SRLanguageObject languageObject,
   short fRefNum
);
```

Parameters

languageObject
A language object.

fRefNum
A file reference number of an open data file into which the data describing the specified language object is to be put.

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
The SRPutLanguageObjectIntoDataFile function puts a description of the language object specified by the languageObject parameter into the data file specified by the fRefNum parameter. Data are written starting at the current file mark, and the file mark is moved to the end of the written data.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.
Available in Mac OS X 10.0 and later.

Declared In
SpeechRecognition.h

SRPutLanguageObjectIntoHandle

Puts a language object (and any embedded languages objects it contains) into a handle.

```
OSErr SRPutLanguageObjectIntoHandle (
   SRLanguageObject languageObject,
   Handle lobjHandle
);
```

Parameters

languageObject

A language object.

lobjHandle

A handle to a block of memory into which the data describing the specified language object is to be put. On entry, this handle can have a length of 0.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRPutLanguageObjectIntoHandle function puts a description of the language object specified by the languageObject parameter into the block of memory specified by the lobjHandle parameter. This replaces the data in the handle and resizes the handle if necessary.

You can use Resource Manager functions (such as AddResource) to store language objects into resources.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRReleaseObject

Releases a speech object.

```
OSErr SRReleaseObject (
   SRSpeechObject srObject
);
```

Parameters

srObject

A speech object.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

If there are no other remaining references to the object, SRReleaseObject disposes of the memory occupied by the object.

Your application should balance every function call that returns an object reference with a call to SRReleaseObject. This means that every call to a function whose name begins with SRNew or SRGet that successfully returns an object reference must be balanced with a call to SRReleaseObject.

In addition, you should call SRReleaseObject to release references to SRSearchResult objects that are passed to your application (via an Apple event handler or a callback routine).

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
SpeechRecognition.h

SRRemoveIndexedItem

Removes a subitem from a container object.

```
OSErr SRRemoveIndexedItem (
   SRSpeechObject container,
   long index
);
```

Parameters

container

A speech object.

index

An integer ranging from 0 to one less than the number of subitems in the specified speech object. (You can call the SRCountItems function to determine the number of subitems contained in a speech object.) If the index you specify is not in this range, SRRemoveIndexedItem returns the result code kSRParamOutOfRange.

Return Value
A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion
The SRRemoveIndexedItem function removes from the speech object specified by the container parameter the subitem located at the position specified by the index parameter. If SRRemoveIndexedItem completes successfully, the number of subitems in the container object is reduced by 1, and the index of each subitem that follows the removed item is reduced by 1.

Version Notes
In Speech Recognition Manager version 1.5, this function is useful only for operating on language objects (of type SRLanguageObject), although it is defined for all speech objects.

Availability
Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In
SpeechRecognition.h

SRRemoveLanguageObject

Removes a language object from another language object that contains it.

```
OSErr SRRemoveLanguageObject (
   SRLanguageObject base,
   SRLanguageObject toRemove
);
```

Parameters

base

The language object containing the language object to remove.

toRemove

The language object to remove.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The object specified by the base parameter should be a container one of whose subitems is the object specified by the toRemove parameter.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRSetIndexedItem

Replaces a subitem in a container object with some other object.

```
OSErr SRSetIndexedItem (
   SRSpeechObject container,
   SRSpeechObject item,
   long index
);
```

Parameters

container

A speech object.

item

A speech object.

index

An integer ranging from 0 to one less than the number of subitems in the specified speech object.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRSetIndexedItem function replaces the subitem having the index specified by the index parameter in the container object specified by the container parameter with the speech object specified by the item parameter. A reference to the replacement item is maintained separately by the container; as a result, you can release any reference to that item if you no longer need it. The reference to the replaced item is removed from the container; if that reference was the last remaining reference to the object, the object is released.

Version Notes

In Speech Recognition Manager version 1.5, this function is useful only for operating on language objects (of type `SRLanguageObject`), although it is defined for all speech objects.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRSetLanguageModel

Sets a recognizer's active language model.

```
OSErr SRSetLanguageModel (
   SRRecognizer recognizer,
   SRLanguageModel languageModel
);
```

Parameters

recognizer

A recognizer.

languageModel

The language model you wish to become the active model for the specified recognizer.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

If no other references exist to the language model currently in use by the specified recognizer, calling `SRSetLanguageModel` with a different language model causes the current one to be released.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRSetProperty

Sets the value of a property of a speech object.

```
OSErr SRSetProperty (
   SRSpeechObject srObject,
   OSType selector,
   const void *property,
   Size propertyLen
);
```

Parameters

srObject

A speech object.

selector

A property selector. See "Recognizer Properties" (page 54), "Recognizer Listen Key Properties" (page 53), "Language Object Properties" (page 47), and "Recognition System Properties" (page 52) for lists of the available property selectors.

property

A pointer to a buffer containing the value to which the specified property is to be set.

propertyLen

The length, in bytes, of the specified buffer.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The `SRSetProperty` function sets the value of the property of the speech object specified by the `srObject` parameter to the value specified through the `property` parameter. The `selector` parameter specifies which property is to be set and the `propertyLen` parameter specifies it size, in bytes.

Not all properties can be set. If you attempt to set a property that cannot be set, `SRSetProperty` returns the result code `kSRCantSetProperty` or `kSRBadSelector` as its function result.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRSpeakAndDrawText

Draws output text in the feedback window and causes the feedback character in the feedback window to speak that text.

```
OSErr SRSpeakAndDrawText (
   SRRecognizer recognizer,
   const void *text,
   SInt32 textLength
);
```

Parameters

recognizer

A recognizer.

text

A pointer to a buffer that contains the text to be drawn and spoken.

textLength

The size, in bytes, of the specified text.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRSpeakText

Causes the feedback character in the feedback window to speak a text string.

```
OSErr SRSpeakText (
   SRRecognizer recognizer,
   const void *speakText,
   SInt32 speakLength
);
```

Parameters

recognizer

A recognizer.

speakText

A pointer to a buffer that contains the text to be spoken. The text pointed to by the `speakText` parameter can contain embedded speech commands to enhance the prosody of the spoken string.

speakLength

The size, in bytes, of the specified text.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

While speaking, the feedback character lip-syncs the spoken string using the Speech Synthesis Manager's phoneme callback routines. `SRSpeakText` uses the default voice and rate selected in the Speech control panel. (The Speech Synthesis Manager was formerly called the Speech Manager. Its name has been changed to distinguish it from the Speech Recognition Manager and to describe its operation more clearly.)

You can use the `SRSpeechBusy` function to determine whether the feedback character is already speaking. If it is, you can call the `SRStopSpeech` function to stop that speaking immediately.

The `SRSpeakText` function speaks the specified text but does not display it. Use the `SRSpeakAndDrawText` function if you want to speak and display the text.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRSpeechBusy

Determines if the feedback character in a feedback window is currently speaking.

```
Boolean SRSpeechBusy (
   SRRecognizer recognizer
);
```

Parameters

recognizer

A recognizer.

Return Value

On return, `true` if the feedback character in the feedback window associated with the recognizer specified by the `recognizer` parameter is currently speaking; otherwise `false`.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRStartListening

Starts a recognizer listening and reporting results to your application.

```
OSErr SRStartListening (
   SRRecognizer recognizer
);
```

Parameters

recognizer

A recognizer.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The `SRStartListening` function instructs the recognizer specified by the `recognizer` parameter to begin processing sound from its speech source and reporting its results to your application (either using Apple events or through a speech recognition callback routine).

You must already have built a language model and attached it to the recognizer (by calling the `SRSetLanguageModel` function) before you call `SRStartListening`.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

`SpeechRecognition.h`

SRStopListening

Stops a recognizer listening and reporting results to your application.

```
OSErr SRStopListening (
   SRRecognizer recognizer
);
```

Parameters

recognizer

A recognizer.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRStopListening function instructs the recognizer specified by the recognizer parameter to stop processing sound from its speech source and reporting its results to your application.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

SRStopSpeech

Terminates speech by the feedback character in a feedback window.

```
OSErr SRStopSpeech (
   SRRecognizer recognizer
);
```

Parameters

recognizer

A recognizer.

Return Value

A result code. See "Speech Recognition Manager Result Codes" (page 58).

Discussion

The SRStopSpeech function immediately terminates any speaking by the feedback character in the feedback window associated with the recognizer specified by the recognizer parameter.

Availability

Available in CarbonLib 1.0 and later when Speech Recognition 1.0 or later is present.

Available in Mac OS X 10.0 and later.

Declared In

SpeechRecognition.h

Callbacks

SRCallBackProcPtr

Defines a pointer to a speech recognition callback function which is called whenever the recognizer encounters one of the events specified in its kSRNotificationParam property.

```
typedef void (*SRCallBackProcPtr) (
    SRCallBackStruct * param
);
```

If you name your function MySRCallBackProc, you would declare it like this:

```
void MySRCallBackProc (
    SRCallBackStruct * param
);
```

Parameters

param

A pointer to a speech recognition callback structure. See SRCallBackStruct (page 39) for a description of this structure.

Discussion

You can receive notification of recognizer events either by installing an Apple event handler or by installing a speech recognition callback function. In general, you should use an Apple event handler to process recognition notifications. You should use callbacks only for executable code that cannot easily receive Apple events.

You can determine what event caused your function to be called by inspecting the what field of the speech recognition callback structure specified by the param parameter.

Because the Speech Recognition Manager is not fully reentrant, you should not call any of its functions other than SRContinueRecognition or SRCancelRecognition from within your speech recognition callback. Accordingly, your callback should simply queue the notification for later processing by your software (for instance, when it receives background processing time).

If the event is of type kSRNotifyRecognitionBeginning (which occurs only if you request speech-begun notifications), you must call either SRContinueRecognition or SRCancelRecognition before speech recognition can continue. A recognizer that has issued a recognition notification suspends activity until you call one of these two functions.

In general, when your speech recognition callback receives the kSRNotifyRecognitionBeginning notification, it should queue an indication for your main code both to adjust the current language model (if necessary) and to call the SRContinueRecognition function. When your callback receives the kSRNotifyRecognitionDone notification, it should queue an indication for your main code to handle the recognition result passed in the message field of the speech recognition callback structure specified by the param parameter. You should make sure, however, that the message field contains a valid reference to a recognition result by inspecting the status field of that structure; if status contains any value other than noErr, the contents of the message field are undefined.

When your callback is executed, your application is not the current process. As a result, some restrictions apply; for example, the current resource chain might not be that of your application.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

Data Types

SRCallBackParam

Defines a speech recognition callback parameter structure.

```
struct SRCallBackParam {
    SRCallBackUPP callBack;
    long refCon;
};
typedef struct SRCallBackParam SRCallBackParam;
```

Fields

`callBack`
A UPP for a speech recognition callback function. You can use the function `NewSRCallBackUPP` to create this UPP.

`refCon`
An application-defined reference constant. This value is passed to your callback routine in the `refcon` field of a speech recognition callback structure. You can pass any 4-byte value you wish.

Discussion
If you want to receive recognition notifications using a speech recognition callback routine instead of an Apple event handler, you must change the value of the `kSRCallBackParam` property of the current recognizer. The value of the `kSRCallBackParam` property is the address of a callback function parameter structure, defined by the `SRCallBackParam` data type.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRCallBackStruct

Defines a structure sent to your speech recognition callback function.

```
struct SRCallBackStruct {
    long what;
    long message;
    SRRecognizer instance;
    OSErr status;
    short flags;
    long refCon;
};
typedef struct SRCallBackStruct SRCallBackStruct;
```

Fields

`what`

A notification flag that indicates the kind of event that caused this notification to be issued. This field contains either `kSRNotifyRecognitionBeginning` or `kSRNotifyRecognitionDone`. See "Notification Flags" (page 50) for complete details on the available notification flags.

`message`

If the value of the `status` field is `noErr` and the value of the `what` field is `kSRNotifyRecognitionDone`, this field contains a reference to a recognition result. Your callback routine can inspect the properties of this recognition result to determine what the user said.

Note that your callback routine must release this reference (by calling `SRReleaseObject`) when it is finished using it. If the value of the `status` field is not `noErr`, the value of this field is undefined.

`instance`

A reference to the recognizer that issued this notification. You should not call `SRReleaseObject` on this recognizer reference in response to a recognition notification.

`status`

An error code indicating the status of the recognition. If the value of this field is `noErr`, the `message` field contains a reference to a recognition result. If the value of this field is `kSRRecognitionDone` and the value of the `what` field is `kSRNotifyRecognitionDone`, the recognizer finished without error but nothing belonging to that recognizer was recognized; in this case, the `message` field does not contain a reference to a recognition result. If the value of this field is any other value, some other error occurred.

`flags`

Reserved for use by Apple Computer, Inc.

`refCon`

An application-defined reference constant. The value in this field is the value you passed in the `refcon` field of a callback function parameter structure (of type `SRCallBackParam`).

Discussion

When you receive a notification of recognition results through an application-defined callback function (instead of using an Apple event handler), the Speech Recognition Manager sends your callback function a pointer to a speech recognition callback structure, defined by the `SRCallBackStruct` data type.

For information on writing a speech recognition callback function, see `SRCallBackProcPtr` (page 38).

Availability

Available in Mac OS X v10.0 and later.

Declared In

`SpeechRecognition.h`

SRCallBackUPP

Defines a universal procedure pointer (UPP) to a speech recognition callback function.

```
typedef SRCallBackProcPtr SRCallBackUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRLanguageModel

Represents a language model.

```
typedef SRLanguageObject SRLanguageModel;
```

Discussion
A language model is a list of zero or more words, phrases, or paths.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRLanguageObject

Represents a language object.

```
typedef SRSpeechObject SRLanguageObject;
```

Discussion
A language model is built using four kinds of objects, collectively called language objects: words, phrases, paths, and language models.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRPath

Represents a language object identifying a path.

```
typedef SRLanguageObject SRPath;
```

Discussion
A path is a sequence of zero or more words, phrases, or language models.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRPhrase

Represents a language object identifying a phrase.

```
typedef SRLanguageObject SRPhrase;
```

Discussion
A phrase is a sequence of zero or more words.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRRecognitionResult

Represents a recognition result which contains information about a recognized utterance.

```
typedef SRSpeechSource SRRecognitionResult;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRRecognitionSystem

Represents a speech object identifying a recognition system.

```
typedef SRSpeechObject SRRecognitionSystem;
```

Discussion
A speech object is an instance of a speech class, which defines a set of properties for objects in the class. The behavior of a speech object is determined by the set of properties associated with the object's class. Recognition systems have the properties associated with the `SRRecognitionSystem` class of speech objects.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRRecognizer

Represents a speech object identifying a speech recognizer.

```
typedef SRSpeechObject SRRecognizer;
```

Discussion
A speech object is an instance of a speech class, which defines a set of properties for objects in the class. The behavior of a speech object is determined by the set of properties associated with the object's class. Speech recognizers have the properties associated with the `SRRecognizer` class of speech objects.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRRejectionLevel

```
typedef  SRRejectionLevel;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRSpeechObject

Defines a reference to a speech object.

```
typedef struct OpaqueSRSpeechObject * SRSpeechObject;
```

Discussion
The Speech Recognition Manager is object oriented in the sense that many of its capabilities are accessed by creating and manipulating speech objects. A speech object is an instance of a speech class, which defines a set of properties for objects in the class. The behavior of a speech object is determined by the set of properties associated with the object's class.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRSpeechSource

Represents a speech object identifying a speech source.

```
typedef SRSpeechObject SRSpeechSource;
```

Discussion
A speech object is an instance of a speech class, which defines a set of properties for objects in the class. The behavior of a speech object is determined by the set of properties associated with the object's class. Speech sources have the properties associated with the `SRSpeechSource` class of speech objects.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRSpeedSetting

```
typedef  SRSpeedSetting;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

SRWord

Represents a language object identifying a word.

```
typedef SRLanguageObject SRWord;
```

Discussion
A word represents a single speakable word.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`SpeechRecognition.h`

Constants

Feedback and Listening Modes

Identify the feedback and listening modes of the recognition system.

```
enum {
    kSRNoFeedbackNoListenModes = 0,
    kSRHasFeedbackHasListenModes = 1,
    kSRNoFeedbackHasListenModes = 2
};
```

Constants

`kSRNoFeedbackNoListenModes`

If the feedback and listening modes value of a recognition system is set to `kSRNoFeedbackNoListenModes`, the next created recognizer has no feedback window and does not use the listening modes selected by the user in the Speech control panel. (For example, push-to-talk is a listening mode.)

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRHasFeedbackHasListenModes`

If the feedback and listening modes value of a recognition system is set to `kSRHasFeedbackHasListenModes`, the next created recognizer opens a feedback window that uses the listening modes selected by the user in the Speech control panel.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRNoFeedbackHasListenModes`

If the feedback and listening modes value of a recognition system is set to `kSRNoFeedbackHasListenModes`, the next created recognizer has no feedback window but does use the listening modes selected by the user in the Speech control panel.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Apple Event Selectors

Define selectors that you can use to handle recognition notifications in your Apple event handler.

```
enum {
    kAESpeechDone = 'srsd',
    kAESpeechDetected = 'srbd'
};
enum {
    kAESpeechSuite = 'sprc'
};
enum {
    keySRRecognizer = 'krec',
    keySRSpeechResult = 'kspr',
    keySRSpeechStatus = 'ksst'
};
enum {
    typeSRRecognizer = 'trec',
    typeSRSpeechResult = 'tspr'
};
```

Constants

`kAESpeechDone`

The message ID for a speech-done event.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kAESpeechDetected`

The message ID for a speech-detected event.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kAESpeechSuite`

The Apple event suite for speech recognition events.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`keySRRecognizer`

The ID for the recognizer parameter.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`keySRSpeechResult`

The ID for the recognition result parameter.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`keySRSpeechStatus`

The ID for the speech status parameter, which is of type `typeShortInteger`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`typeSRRecognizer`

The type for the recognizer parameter.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`typeSRSpeechResult`

The type for the recognition result parameter.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

Default Rejection Level

Represents a default rejection level.

```
enum {
    kSRDefaultRejectionLevel = 50
};
```

Language Object Properties

Define property selectors for language objects.

```
enum {
    kSRSpelling = 'spel',
    kSRLMObjType = 'lmtp',
    kSRRefCon = 'refc',
    kSROptional = 'optl',
    kSREnabled = 'enbl',
    kSRRepeatable = 'rptb',
    kSRRejectable = 'rjbl',
    kSRRejectionLevel = 'rjct'
};
```

Constants

`kSRSpelling`

The spelling of a language object. The value of this property is a variable-length string of characters. For an object of type `SRWord`, the value is the spelled word. For an object of type `SRPhrase`, the value is the concatenation of the spellings of each word in the phrase, separated by a language-dependent separation character (for example, by a space character). For an object of type `SRPath`, the value is the concatenation of the spellings of each word and language model name in the path. For an object of type `SRLanguageModel`, the value is the name of the language model. For any object, the string value does not include either a length byte (as in Pascal strings) or a null terminating character (as in C strings).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRLMObjType`

The type of a language object. The value of this property is a four-character constant of type `OSType`; see the section "Language Object Types" (page 49) for the values that are defined for this property. You cannot set a property of this type.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRRefCon`

The reference constant. The value of this property is a 4-byte value specified by your application. By default, the value of a reference constant property is zero (0).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSROptional`

The optional flag. The value of this property is a Boolean value that indicates whether speaking the words, phrases, paths, and language models represented by the object is optional (`TRUE`) or required (`FALSE`). A user is not required to utter optional words, phrases, or language models. By default, the value of an object's optional flag is `FALSE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSREnabled`

The enabled flag. The value of this property is a Boolean value that indicates whether the object is enabled (`TRUE`) or disabled (`FALSE`). Disabled objects are ignored during speech recognition. By default, the value of an object's enabled flag is `TRUE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRRepeatable`

The repeatable flag. The value of this property is a Boolean value that indicates whether the object is repeatable (`TRUE`) or not (`FALSE`). A user can utter a repeatable object more than once. By default, the value of an object's repeatable flag is `FALSE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRRejectable`

The rejectable flag. The value of this property is a Boolean value that indicates whether the object is rejectable (`TRUE`) or not (`FALSE`). An object is rejectable if a recognition system can return the rejected word instead of that object. (The rejected word is the value of the `kSRRejectedWord` property of the recognition system.) By default, the value of an object's rejectable flag is `FALSE`. However, if an entire utterance is rejected, you can still get the rejected word. See "Recognition Result Properties" (page 51).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRRejectionLevel`

The rejection level. The value of this property is a 2-byte unsigned integer of type `SRRejectionLevel` between 0 and 100, inclusive, that determines how likely a recognizer is to reject a language object whose `kSRRejectable` property is `TRUE`. If an object's rejection level is close to 0, the recognizer is less likely to reject utterances (and hence more likely to return a result with phrases from the current language model, whether or not the user actually said something in that language model) if an object's rejection level is close to 100, the recognizer is more likely to reject utterances. You can set an object's rejection flag to `TRUE` and its rejection level to some appropriate value to reduce the likelihood that a recognizer will mistakenly recognize a random user utterance as part of the active language model. By default, the value of an object's rejection level is `50`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion
Every language object (that is, any instance of a subclass of the `SRLanguageObject` class) has a set of properties that you can inspect and change by calling the `SRGetProperty` and `SRSetProperty` functions. You specify a property by passing a property selector to those functions.

Language Object Types

Identify the four subclasses of the `SRLanguageObject` class.

```
enum {
    kSRLanguageModelType = 'lmob',
    kSRPathType = 'path',
    kSRPhraseType = 'phra',
    kSRWordType = 'word'
};
```

Constants

`kSRLanguageModelType`

A language model (that is, an object of type `SRLanguageModel`).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRPathType`

A path (that is, an object of type `SRPath`).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRPhraseType`

A phrase (that is, an object of type `SRPhrase`).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRWordType`

A word (that is, an object of type `SRWord`).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion
You can use these constants, for instance, to help interpret the value of a language object's `kSRLMObjType` property.

Listen Key Modes

Identify listen key modes.

```
enum {
    kSRUseToggleListen = 0,
    kSRUsePushToTalk = 1
};
```

Constants

`kSRUseToggleListen`

The recognizer interprets presses on the listen key as a toggle to turn listening on or off.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRUsePushToTalk`

The recognizer listens only when the listen key is held down.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

You can get (but not set) a recognizer's listen key mode by accessing its property of type `kSRListenKeyMode`. That property's value is a 2-byte unsigned integer that determines whether the listen key operates in push-to-talk or toggle-listening mode.

Notification Flags

Identify the recognizer events that may be sent to an application.

```
enum {
    kSRNotifyRecognitionBeginning = 1L << 0,
    kSRNotifyRecognitionDone = 1L << 1
};
```

Constants

`kSRNotifyRecognitionBeginning`

If this bit is set, your application will be notified when the user starts speaking and recognition is ready to begin. When your application gets this notification, it must call either `SRContinueRecognition` or `SRCancelRecognition` in order for recognition either to continue or be canceled. If you do not call one of these functions, the recognizer will simply wait until you do (and hence appear to have quit working). Note that calling `SRCancelRecognition` cancels a recognition only for the application that requested it, not for all applications.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRNotifyRecognitionDone`

If this bit is set, your application will be notified when recognition is finished and the result (if any) of that recognition is available.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

You can indicate which recognizer events you want your application be to notified of by setting the recognizer's notification property, which is a property of type `kSRNotificationParam`. That property's value is a 4-byte unsigned integer. The Speech Recognition Manager defines these masks for bits in that value.

Recognition Result Properties

Identify property selectors for recognition results.

```
enum {
    kSRTEXTFormat = 'TEXT',
    kSRPhraseFormat = 'lmph',
    kSRPathFormat = 'lmpt',
    kSRLanguageModelFormat = 'lmfm'
};
```

Constants

`kSRTEXTFormat`

The text format. The value of this property is a variable-length string of characters that is the text of the recognized utterance. If the utterance was rejected, this text is the spelling of the rejected word. The string value does not include either a length byte (as in Pascal strings) or a null terminating character (as in C strings).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRPhraseFormat`

The phrase format. The value of this property is a phrase that contains one word (of type `SRWord`) for each word in the recognized utterance. If the utterance was rejected, this path or phrase contains one object, the rejected word. The reference constant value of the phrase is always 0, but each word in the phrase retains its own reference constant property value.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRPathFormat`

The path format. The value of this property is a path that contains a sequence of words (of type `SRWord`) and phrases (of type `SRPhrase`) representing the text of the recognized utterance. If the utterance was rejected, this path or phrase contains one object, the rejected word. The reference constant value of the path is always 0, but each word or phrase in the path retains its own reference constant property value.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRLanguageModelFormat`

The language model format. The value of this property is a language model that contains a copy of each word, phrase, path, and language model used in the recognized utterance. If the utterance was rejected, the value of this property is the rejected word (that is, the `kSRRejectedWord` property of the recognition system). The name and reference constant of this language model are the same as the name and reference constant of the active language model, and each subitem in the language model retains its own reference constant property value.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

Every recognition result object has a set of properties that you can inspect by calling the `SRGetProperty` function. You specify a property by passing a property selector to those functions.

`SRGetProperty` returns an object reference as the value of a recognition result's `kSRPhraseFormat`, `kSRPathFormat`, or `kSRLanguageModelFormat` property. You must make sure to release that object reference (by calling `SRReleaseObject`) when you are finished using it.

Recognition System IDs

Defines a recognition system ID.

```
enum {
    kSRDefaultRecognitionSystemID = 0
};
```

Constants

`kSRDefaultRecognitionSystemID`

The default speech recognition system.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

When you call `SROpenRecognitionSystem` to open a recognition system, you indicate the system to open by passing a recognition system ID.

Recognition System Properties

Define property selectors for recognition systems.

```
enum {
    kSRFeedbackAndListeningModes = 'fbwn',
    kSRRejectedWord = 'rejq',
    kSRCleanupOnClientExit = 'clup'
};
```

Constants

`kSRFeedbackAndListeningModes`

The feedback and listening modes of the recognition system. The value of this property is an integer that determines some of the features of a recognizer subsequently created by your application. See "Feedback and Listening Modes" (page 44) for a description of the values possible here.

The default value for version 1.5 is `kSRNoFeedbackNoListenModes`, but most applications should set this to `kSRHasFeedbackHasListenModes`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRRejectedWord`

The rejected word of the recognition system. The value of this property is a value of type `SRWord` that will be returned in a recognition result object when a recognizer encounters an unrecognizable utterance. For example, if an utterance is rejected, the `kSRLMObjType` property of the rejection result is the rejected word. By default, a recognition system's rejected word is spelled "???" and has a reference constant of 0.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRCleanupOnClientExit`

The cleanup mode of the recognition system. Applications should never set this property. If, however, you do not have a process ID (as issued by the Process Manager), you should set this property to `FALSE` so that speech objects you allocate will not be associated with any other process. By default, the value of a recognition system's cleanup mode is `TRUE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

A recognition system (that is, an instance of the `SRRecognitionSystem` class) has a set of properties that you can inspect and change by calling the `SRGetProperty` and `SRSetProperty` functions. You specify a property by passing a property selector to those functions.

Recognizer Listen Key Properties

Define listen key property selectors for recognizers.

```
enum {
    kSRListenKeyMode = 'lkmd',
    kSRListenKeyCombo = 'lkey',
    kSRListenKeyName = 'lnam',
    kSRKeyWord = 'kwrd',
    kSRKeyExpected = 'kexp'
};
```

Constants

`kSRListenKeyMode`

The listen key mode. The value of this property is a 2-byte unsigned integer that indicates whether the listen key operates in push-to-talk or toggle-listening mode. See "Listen Key Modes" (page 49) for a description of the available listen key modes. The value of a recognizer's listen key mode is whatever the user has selected in the Speech control panel. This property is read-only you cannot set a property of this type.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRListenKeyCombo`

The listen key combination property. The value of this property is a 2-byte unsigned integer that specifies the key combination the user must press for the listen key. The high-order byte of this value has the same format as the high-order byte of the `modifiers` field of an event record. The low-order byte of this value has the same format as the key code contained in the `message` field of an event record. The value of a recognizer's listen key combination property is whatever the user has selected in the Speech control panel. This property is read-only you cannot set a property of this type.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRListenKeyName`

The listen key name property. The value of this property is a string (of type `Str63`) that represents the listen key combination specified by the `kSRListenKeyCombo` property. The value of a recognizer's listen key name property is whatever the user has selected in the Speech control panel. This property is read-only you cannot set a property of this type.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

kSRKeyWord

The key word property. The value of this property is a string (of type Str255) that represents the key word that must precede utterances when the recognizer is in toggle-listen mode. The value of a recognizer's key word property is whatever the user has selected in the Speech control panel. This property is read-only you cannot set a property of this type.

Available in Mac OS X v10.0 and later.

Declared in SpeechRecognition.h.

kSRKeyExpected

The key expected flag. The value of this property is a Boolean value that indicates whether the recognizer expects the user to hold down a key or to utter the key word in order to have the recognizer begin listening (TRUE) or not (FALSE). The value of a recognizer's key expected flag is a function of the user's Speech control panel selections. This property is TRUE whenever text is visible below the feedback character in the lower-left corner of the feedback window. This property is read-only.

Available in Mac OS X v10.0 and later.

Declared in SpeechRecognition.h.

Discussion

Every recognizer has a set of properties that you can inspect and change by calling the SRGetProperty and SRSetProperty functions. You specify a property by passing a property selector to those functions.

The listen key properties are provided for use by applications that want to provide their own visual feedback. If your application uses the default feedback mechanisms, you do not need to access those properties.

Recognizer Properties

Define property selectors for recognizers.

```
enum {
    kSRNotificationParam = 'noti',
    kSRCallBackParam = 'call',
    kSRSearchStatusParam = 'stat',
    kSRAutoFinishingParam = 'afin',
    kSRForegroundOnly = 'fgon',
    kSRBlockBackground = 'blbg',
    kSRBlockModally = 'blmd',
    kSRWantsResultTextDrawn = 'txfb',
    kSRWantsAutoFBGestures = 'dfbr',
    kSRSoundInVolume = 'volu',
    kSRReadAudioFSSpec = 'aurd',
    kSRCancelOnSoundOut = 'caso',
    kSRSpeedVsAccuracyParam = 'sped'
};
```

Constants

kSRNotificationParam

The notification property. The value of this property is a 4-byte unsigned integer whose bits encode the kinds of events of which the recognizer will notify your application. See the section "Notification Flags" (page 50) for the bit masks that are defined for this property. By default, the value of a recognizer's notification property is kSRNotifyRecognitionDone.

Available in Mac OS X v10.0 and later.

Declared in SpeechRecognition.h.

`kSRCallBackParam`

The callback property. The value of this property is of type `SRCallBackParam` that determines whether recognition notifications are sent to your application via Apple events or via an application-defined callback routine. To specify a callback routine, set the value of this property to the address of a callback routine parameter structure. By default, the value of a recognizer's callback property is `NULL`, indicating that Apple events are to be used to report recognizer events.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRSearchStatusParam`

The search status. The value of this property is a 4-byte unsigned integer whose bits indicate the current state of the recognizer. See the section "Search Status Flags" (page 57) for the bit masks that are defined for this property. This property is read-only; you cannot set a property of this type.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRAutoFinishingParam`

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRForegroundOnly`

The foreground-only flag. The value of this property is a Boolean value that indicates whether the recognizer is enabled only when your application is the foreground application (`TRUE`) or not (`FALSE`). By default, the value of a recognizer's foreground-only flag is `TRUE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRBlockBackground`

The background-blocking flag. The value of this property is a Boolean value that indicates whether all recognizers owned by other applications are automatically disabled whenever your application is the foreground application (`TRUE`) or are not automatically disabled (`FALSE`). By default, the value of a recognizer's background-blocking flag is `FALSE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRBlockModally`

The modal-blocking flag. The value of this property is a Boolean value that indicates whether the language model associated with this recognizer is the only active language model (`TRUE`) or not (`FALSE`). When this flag is `TRUE`, your application's recognizer blocks those of other applications even when it is not the foreground application in addition, the feedback window is hidden if you are not using it. Setting this property to `TRUE` prevents speech recognition from working for other applications, so you want to use this property only if your application is taking over the computer (like some games) or briefly attempting to constrain the language model severely. By default, the value of a recognizer's modal-blocking flag is `FALSE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRWantsResultTextDrawn`

The text feedback flag. The value of this property is a Boolean value that indicates whether the results of a search are to be automatically displayed as text in the feedback window (`TRUE`) or not (`FALSE`). If you set the value of this property to `FALSE`, you should call `SRDrawRecognizedText` with a string representing what the user said. By default, the value of a recognizer's text feedback flag is `TRUE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRWantsAutoFBGestures`

The automatic feedback gestures flag. The value of this property is a Boolean value that determines whether the feedback gestures are automatically drawn (`TRUE`) or not (`FALSE`). If you want more control over feedback behavior, you should set this property to `FALSE`; then call `SRProcessBegin` when you want to begin responding to a spoken request and `SRProcessEnd` when you are finished. During that time, the feedback character displays appropriate animated gestures to indicate that it is busy performing the task. By default, the value of a recognizer's automatic feedback gestures flag is `TRUE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRSoundInVolume`

The sound input volume. The value of this property is a 2-byte unsigned integer between 0 and 100, inclusive, that indicates the current sound input volume. This property is read-only; you cannot set a property of this type.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRReadAudioFSSpec`

The audio file property. You can use this property to perform speech recognition from an audio file. The value of this property is a pointer to a file system specification (a structure of type `FSSpec`). The file system specification indicates an AIFF file that contains raw audio data (16-bit audio data sampled at 22.050 kHz). After you create a new recognizer using the speech source ID `kSRCanned22kHzSpeechSource`, you must set this recognizer property to perform recognition from an audio file. Setting the audio source to a file also allows the Speech Recognition Manager to process sound data at system background time rather than at interrupt time or deferred task time.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRCancelOnSoundOut`

The cancel during sound output flag. The value of this property is a Boolean value that indicates whether speech recognition is canceled whenever any sound is output by the computer during an utterance (`TRUE`) or whether speech recognition continues (`FALSE`). By default, the value of a recognizer's cancel during sound output flag is `TRUE`.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRSpeedVsAccuracyParam`

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

Every recognizer has a set of properties that you can inspect and change by calling the `SRGetProperty` and `SRSetProperty` functions. You specify a property by passing a property selector to those functions.

Search Status Flags

Indicate the status of a recognizer search.

```
enum {
    kSRIdleRecognizer = 1L << 0,
    kSRSearchInProgress = 1L << 1,
    kSRSearchWaitForAllClients = 1L << 2,
    kSRMustCancelSearch = 1L << 3,
    kSRPendingSearch = 1L << 4
};
```

Constants

`kSRIdleRecognizer`

If this bit is set, the search engine is not active and the user is able to make a new utterance.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRSearchInProgress`

If this bit is set, a search is currently in progress.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRSearchWaitForAllClients`

If this bit is set, a search is not currently in progress, but will begin as soon as every recognizer using the speech source used by this recognizer has called `SRContinueRecognition` to indicate that the search should begin.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRMustCancelSearch`

If this bit is set, a search is about to be canceled (for example, because the recognizer determined a sound to be non-speech).

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRPendingSearch`

If this bit is set, a search is about to begin.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

You can determine the current status of a recognizer search by getting the recognizer's search status, which is a property of type `kSRSearchStatusParam`. That property's value is a 4-byte unsigned integer. The Speech Recognition Manager defines these masks for bits in that value.

Speech Source Constants

Identify Speech Recognition Manager-supported speech sources.

```
enum {
    kSRDefaultSpeechSource = 0,
    kSRLiveDesktopSpeechSource = 'dklv',
    kSRCanned22kHzSpeechSource = 'ca22'
};
```

Constants

`kSRDefaultSpeechSource`

The default speech source.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRLiveDesktopSpeechSource`

Live desktop sound input.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

`kSRCanned22kHzSpeechSource`

AIFF file based 16 bit, 22.050 KHz sound input.

Available in Mac OS X v10.0 and later.

Declared in `SpeechRecognition.h`.

Discussion

The Speech Recognition Manager supports several speech sources, which you can specify using these constants. In version 1.5, the default speech source is `kSRLiveDesktopSpeechSource`.

Result Codes

The most common result codes returned by Speech Recognition Manager are listed below.

Result Code	Value	Description
`kSRNotAvailable`	-5100	Requested service not available or applicable Available in Mac OS X v10.0 and later.
`kSRInternalError`	-5101	Internal system or hardware error condition Available in Mac OS X v10.0 and later.
`kSRComponentNotFound`	-5102	Required component cannot be located Available in Mac OS X v10.0 and later.
`kSROutOfMemory`	-5103	Not enough memory available Available in Mac OS X v10.0 and later.
`kSRNotASpeechObject`	-5104	Object is not valid Available in Mac OS X v10.0 and later.
`kSRBadParameter`	-5105	Invalid parameter specified Available in Mac OS X v10.0 and later.

Result Code	Value	Description
kSRParamOutOfRange	-5106	Parameter is out of valid range Available in Mac OS X v10.0 and later.
kSRBadSelector	-5107	Unrecognized selector specified Available in Mac OS X v10.0 and later.
kSRBufferTooSmall	-5108	Buffer is too small Available in Mac OS X v10.0 and later.
kSRNotARecSystem	-5109	Specified object is not a recognition system Available in Mac OS X v10.0 and later.
kSRFeedbackNotAvail	-5110	No feedback window associated with recognizer Available in Mac OS X v10.0 and later.
kSRCantSetProperty	-5111	Cannot set the specified property Available in Mac OS X v10.0 and later.
kSRCantGetProperty	-5112	Cannot get the specified property Available in Mac OS X v10.0 and later.
kSRCantSetDuringRecognition	-5113	Cannot set property during recognition Available in Mac OS X v10.0 and later.
kSRAlreadyListening	-5114	System is already listening Available in Mac OS X v10.0 and later.
kSRNotListeningState	-5115	System is not listening Available in Mac OS X v10.0 and later.
kSRModelMismatch	-5116	No acoustical models available to match request Available in Mac OS X v10.0 and later.
kSRNoClientLanguageModel	-5117	Cannot access specified language model Available in Mac OS X v10.0 and later.
kSRNoPendingUtterances	-5118	No utterances to search Available in Mac OS X v10.0 and later.
kSRRecognitionCanceled	-5119	Search was canceled Available in Mac OS X v10.0 and later.
kSRRecognitionDone	-5120	Search has finished, but nothing was recognized Available in Mac OS X v10.0 and later.

Result Code	Value	Description
kSROtherRecAlreadyModal	-5121	Another recognizer is already operating modally Available in Mac OS X v10.0 and later.
kSRHasNoSubItems	-5122	Specified object has no subitems Available in Mac OS X v10.0 and later.
kSRSubItemNotFound	-5123	Specified subitem cannot be located Available in Mac OS X v10.0 and later.
kSRLanguageModelTooBig	-5124	Language model too big to be built Available in Mac OS X v10.0 and later.
kSRAlreadyReleased	-5125	Specified object has already been released Available in Mac OS X v10.0 and later.
kSRAlreadyFinished	-5126	Specified language model has already been finished Available in Mac OS X v10.0 and later.
kSRWordNotFound	-5127	Spelling could not be found Available in Mac OS X v10.0 and later.
kSRNotFinishedWithRejection	-5128	Language model not finished with rejection Available in Mac OS X v10.0 and later.
kSRExpansionTooDeep	-5129	Language model is left recursive or is embedded too many levels Available in Mac OS X v10.0 and later.
kSRTooManyElements	-5130	Too many elements added to phrase, path, or other language object Available in Mac OS X v10.0 and later.
kSRCantAdd	-5131	Can't add specified type of object to the base language object Available in Mac OS X v10.0 and later.
kSRSndInSourceDisconnected	-5132	Sound input source is disconnected Available in Mac OS X v10.0 and later.
kSRCantReadLanguageObject	-5133	Cannot create language object from file or pointer Available in Mac OS X v10.0 and later.
kSRNotImplementedYet	-5199	Feature is not yet implemented Available in Mac OS X v10.0 and later.

Gestalt Constants

You can check for version and feature availability information by using the Speech Recognition Manager selectors defined in the Gestalt Manager. For more information see *Inside Mac OS X: Gestalt Manager Reference*.

Document Revision History

This table describes the changes to *Speech Recognition Manager Reference*.

Date	Notes
2003-02-06	Fixed erroneous cross references in Constants section.
	Updated formatting.
	Fixed typographical errors.

Index