

# AVAudioRecorder Class Reference

사운드 콘텐츠 응용

Dept. of Multimedia Science, Sookmyung Women's University, prof. JongWoo Lee

## Index

- Overview
- Tasks
  - Initializing an AVAudioRecorder Object
  - Configuring and Controlling Recording
  - Managing Information About a Recording
  - Using Audio Level Metering

# Overview

---

## AVAudioRecorder

– providing audio recording capability

### Using an audio Recorder you can:

- Record until the user stops the recording
- Record for a specified duration
- Pause and resume a recording
- Obtain input audio-level data that you can use to provide level metering

## Initializing an AVAudioRecorder Object

# Initializing a AVAudioRecorder Object

## initWithURL:settings:error:

Initializing and returns an audio recorder

```
- (id) initWithURL:(NSURL *)url  
  settings:(NSDictionary *)settings  
  error:(NSError **)outError
```

- **Parameters**

url	<ul style="list-style-type: none"><li>• The file system location to record to.</li><li>• The file type to record to is inferred from the file extension included in this parameter's value.</li></ul>
settings	<ul style="list-style-type: none"><li>• Settings for the recording session.</li><li>• For information on the settings available for an audio recorder.</li></ul>
outError	<ul style="list-style-type: none"><li>• Pass in the address of a nil-initialized NSError object.</li><li>• If an error occurs, upon return the NSError object describes the error.</li><li>• NULL : If you do not want error information</li></ul>

- **Return Value**

On success : initializing AVAudioRecorder object

nil : the outError parameter contains a code that describes the problem

- **Availability**

Available in iOS 3.0 and later.

## Configuring and Controlling Recording

# Configuring and Controlling Recording

## prepareToRecord

Creates an audio file and prepares the system for recording.

– (BOOL)prepareToRecording

- Discussion

- Creates an audio file at the location specified by the `url` parameter in the `initWithURL:settings:error:` method. If a file already exists at that location, this method overwrites it.
- The preparation invoked by this method takes place automatically when you call `record`.
- Use `prepareToRecord` to start as quickly as possible upon calling `record`

- Return Value

YES : success

NO : failure.

- Availability

Available in iOS 3.0 and later.

# Configuring and Controlling Recording

## record

Starts or resumes recording.

– (BOOL)record

- Return Value

YES : success

NO : failure

- Discussion

- Calling this method implicitly calls `prepareToPlay`
- creates an audio file and prepares the system for recording

- Availability

Available in iOS 3.0 and later.

# Configuring and Controlling Recording

## recordForDuration:

Records for a specified duration of time.

```
- (BOOL)recordForDuration:(NSTimeInterval)duration
```

- **Parameters**

duration	• The maximum duration, in seconds, for the recording.
----------	--

- **Discussion**

- The recorder stops when the duration of recorded audio reaches the value in the *duration* parameter.
- Calling this method implicitly calls `prepareToRecord`.

- **Return Value**

YES : success

NO : failure.

- **Availability**

Available in iOS 3.0 and later.

# Configuring and Controlling Recording

## pause

pauses a recording.

```
- (void)pause
```

- **Discussion**

Calling `record` to resume recording.

- **Availability**

Available in iOS 3.0 and later.

# Configuring and Controlling Recording

## stop

Stops recording and closes the audio file.

```
- (void)stop
```

- **Availability**

Available in iOS 3.0 and later.

# Configuring and Controlling Recording

## delegate

The delegate object for the audio recorder.

```
@property(assign) id<AVAudioRecorderDelegate> delegate
```

- **Availability**

Available in iOS 3.0 and later.

# Configuring and Controlling Recording

## deleteRecording

Deletes a recorded audio file.

```
- (void)deleteRecording
```

- **Discussion**

The audio recorder must be stopped before you call this method.

- **Return Value**

YES : success

NO : failure.

- **Availability**

Available in iOS 3.0 and later.

## Managing Information About a Recording

# Managing Information About a Recording

## recording

A Boolean value that indicates whether the audio recorder is recording(YES), or not(NO).

```
@property (readonly, getter=isRecording) BOOL recording;
```

- **Availability**

Available in iOS 3.0 and later.

# Managing Information About a Recording

## url (read-only)

The URL for the audio associated with the audio player.

```
@property(readonly) NSURL *url
```

- **Availability**

Available in iOS 3.0 and later.



# Managing Information About a Recording

## currentTime

The time, in seconds, since the beginning of the recording.

```
@property (readonly) NSTimeInterval currentTime;
```

- **Discussion**
  - When the audio recorder is stopped, calling this method returns a value of 0.
- **Availability**

Available in iOS 3.0 and later.

# Managing Information About a Recording

## settings

The audio settings for the audio recorder.

```
@property (readonly) NSDictionary *settings;
```

- **Discussion**
  - Audio recorder settings are in effect only after you explicitly call the [prepareToRecord](#) method, or after you call it implicitly by starting recording.
  - The audio settings keys are described in AV Foundation Audio Settings Constants.
- **Availability**

Available in iOS 3.0 and later.

# Using Audio Level Metering

## Using Audio Level Metering

### meteringEnabled

A Boolean value that indicates whether audio-level metering is enabled(YES), or not(NO).

```
@property(getter=isMeteringEnabled) BOOL meteringEnabled
```

- **Discussion**

- By default, audio level metering is off for an audio recorder because metering uses computing resources, turn it on only if you intend to use it.

- **Availability**

Available in iOS 3.0 and later.

# Using Audio Level Metering

## - updateMeters

Refreshes the average and peak power values for all channels of an audio recorder.

```
- (void)updateMeters
```

- **Discussion**

To obtain current audio power values, you must call this method before calling `averagePowerForChannel:` or `peakPowerForChannel:`.

- **Availability**

Available in iOS 3.0 and later.

# Using Audio Level Metering

## - peakPowerForChannel:

Returns the peak power for a given channel, in decibels, for the sound being recorded.

```
- (float)peakPowerForChannel:(NSUInteger)channelNumber
```

- **Parameters**

channelNumber	• The number of the channel that you want the peak power value for.
---------------	---

- **Return Value**

- 0 dB : full scale, or maximum power
- -160 dB : minimum power (near silence).

- **Discussion**

To obtain a current peak power value, you must call the `updateMeters` method before calling this method.

- **Availability**

Available in iOS 3.0 and later.

# Using Audio Level Metering

## - averagePowerForChannel:

Returns the average power for a given channel, in decibels, for the sound being recorded.

```
- (float)averagePowerForChannel:(NSUInteger)channelNumber
```

- **Parameters**

channelNumber	• The number of the channel that you want the average power value for.
---------------	--

- **Return Value**

- 0 dB : full scale, or maximum power;
- -160 dB : minimum power (near silence).

- **Discussion**

To obtain a current average power value, you must call the [updateMeters](#) method before calling this method.

- **Availability**

Available in iOS 3.0 and later.