| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AudioFormat.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sound/sampled/AudioFileFormat.Type.html)   [**NEXT CLASS**](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sound/sampled/AudioFormat.html)    [**NO FRAMES**](http://docs.google.com/AudioFormat.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

## **javax.sound.sampled**

Class AudioFormat

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.sound.sampled.AudioFormat**

public class **AudioFormat**extends [Object](http://docs.google.com/java/lang/Object.html)

AudioFormat is the class that specifies a particular arrangement of data in a sound stream. By examing the information stored in the audio format, you can discover how to interpret the bits in the binary sound data.

Every data line has an audio format associated with its data stream. The audio format of a source (playback) data line indicates what kind of data the data line expects to receive for output. For a target (capture) data line, the audio format specifies the kind of the data that can be read from the line. Sound files also have audio formats, of course. The [AudioFileFormat](http://docs.google.com/javax/sound/sampled/AudioFileFormat.html) class encapsulates an AudioFormat in addition to other, file-specific information. Similarly, an [AudioInputStream](http://docs.google.com/javax/sound/sampled/AudioInputStream.html) has an AudioFormat.

The AudioFormat class accommodates a number of common sound-file encoding techniques, including pulse-code modulation (PCM), mu-law encoding, and a-law encoding. These encoding techniques are predefined, but service providers can create new encoding types. The encoding that a specific format uses is named by its encoding field.

In addition to the encoding, the audio format includes other properties that further specify the exact arrangement of the data. These include the number of channels, sample rate, sample size, byte order, frame rate, and frame size. Sounds may have different numbers of audio channels: one for mono, two for stereo. The sample rate measures how many "snapshots" (samples) of the sound pressure are taken per second, per channel. (If the sound is stereo rather than mono, two samples are actually measured at each instant of time: one for the left channel, and another for the right channel; however, the sample rate still measures the number per channel, so the rate is the same regardless of the number of channels. This is the standard use of the term.) The sample size indicates how many bits are used to store each snapshot; 8 and 16 are typical values. For 16-bit samples (or any other sample size larger than a byte), byte order is important; the bytes in each sample are arranged in either the "little-endian" or "big-endian" style. For encodings like PCM, a frame consists of the set of samples for all channels at a given point in time, and so the size of a frame (in bytes) is always equal to the size of a sample (in bytes) times the number of channels. However, with some other sorts of encodings a frame can contain a bundle of compressed data for a whole series of samples, as well as additional, non-sample data. For such encodings, the sample rate and sample size refer to the data after it is decoded into PCM, and so they are completely different from the frame rate and frame size.

An AudioFormat object can include a set of properties. A property is a pair of key and value: the key is of type String, the associated property value is an arbitrary object. Properties specify additional format specifications, like the bit rate for compressed formats. Properties are mainly used as a means to transport additional information of the audio format to and from the service providers. Therefore, properties are ignored in the [matches(AudioFormat)](http://docs.google.com/javax/sound/sampled/AudioFormat.html#matches(javax.sound.sampled.AudioFormat)) method. However, methods which rely on the installed service providers, like [(AudioFormat, AudioFormat) isConversionSupported](http://docs.google.com/javax/sound/sampled/AudioSystem.html#isConversionSupported(javax.sound.sampled.AudioFormat.Encoding,%20javax.sound.sampled.AudioFormat)) may consider properties, depending on the respective service provider implementation.

The following table lists some common properties which service providers should use, if applicable:

| Property key | Value type | Description |
| --- | --- | --- |
| "bitrate" | [Integer](http://docs.google.com/java/lang/Integer.html) | average bit rate in bits per second |
| "vbr" | [Boolean](http://docs.google.com/java/lang/Boolean.html) | true, if the file is encoded in variable bit rate (VBR) |
| "quality" | [Integer](http://docs.google.com/java/lang/Integer.html) | encoding/conversion quality, 1..100 |

Vendors of service providers (plugins) are encouraged to seek information about other already established properties in third party plugins, and follow the same conventions.

**Since:** 1.3 **See Also:**[DataLine.getFormat()](http://docs.google.com/javax/sound/sampled/DataLine.html#getFormat()), [AudioInputStream.getFormat()](http://docs.google.com/javax/sound/sampled/AudioInputStream.html#getFormat()), [AudioFileFormat](http://docs.google.com/javax/sound/sampled/AudioFileFormat.html), [FormatConversionProvider](http://docs.google.com/javax/sound/sampled/spi/FormatConversionProvider.html)

| **Nested Class Summary** | |
| --- | --- |
| static class | [**AudioFormat.Encoding**](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html)            The Encoding class names the specific type of data representation used for an audio stream. |

| **Field Summary** | |
| --- | --- |
| protected  boolean | [**bigEndian**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#bigEndian)            Indicates whether the audio data is stored in big-endian or little-endian order. |
| protected  int | [**channels**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#channels)            The number of audio channels in this format (1 for mono, 2 for stereo). |
| protected  [AudioFormat.Encoding](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) | [**encoding**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#encoding)            The audio encoding technique used by this format. |
| protected  float | [**frameRate**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#frameRate)            The number of frames played or recorded per second, for sounds that have this format. |
| protected  int | [**frameSize**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#frameSize)            The number of bytes in each frame of a sound that has this format. |
| protected  float | [**sampleRate**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#sampleRate)            The number of samples played or recorded per second, for sounds that have this format. |
| protected  int | [**sampleSizeInBits**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#sampleSizeInBits)            The number of bits in each sample of a sound that has this format. |

| **Constructor Summary** | |
| --- | --- |
| [**AudioFormat**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#AudioFormat(javax.sound.sampled.AudioFormat.Encoding,%20float,%20int,%20int,%20int,%20float,%20boolean))([AudioFormat.Encoding](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) encoding, float sampleRate, int sampleSizeInBits, int channels, int frameSize, float frameRate, boolean bigEndian)            Constructs an AudioFormat with the given parameters. |
| [**AudioFormat**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#AudioFormat(javax.sound.sampled.AudioFormat.Encoding,%20float,%20int,%20int,%20int,%20float,%20boolean,%20java.util.Map))([AudioFormat.Encoding](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) encoding, float sampleRate, int sampleSizeInBits, int channels, int frameSize, float frameRate, boolean bigEndian, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Object](http://docs.google.com/java/lang/Object.html)> properties)            Constructs an AudioFormat with the given parameters. |
| [**AudioFormat**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#AudioFormat(float,%20int,%20int,%20boolean,%20boolean))(float sampleRate, int sampleSizeInBits, int channels, boolean signed, boolean bigEndian)            Constructs an AudioFormat with a linear PCM encoding and the given parameters. |

| **Method Summary** | |
| --- | --- |
| int | [**getChannels**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getChannels())()            Obtains the number of channels. |
| [AudioFormat.Encoding](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) | [**getEncoding**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getEncoding())()            Obtains the type of encoding for sounds in this format. |
| float | [**getFrameRate**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getFrameRate())()            Obtains the frame rate in frames per second. |
| int | [**getFrameSize**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getFrameSize())()            Obtains the frame size in bytes. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getProperty(java.lang.String))([String](http://docs.google.com/java/lang/String.html) key)            Obtain the property value specified by the key. |
| float | [**getSampleRate**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getSampleRate())()            Obtains the sample rate. |
| int | [**getSampleSizeInBits**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getSampleSizeInBits())()            Obtains the size of a sample. |
| boolean | [**isBigEndian**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#isBigEndian())()            Indicates whether the audio data is stored in big-endian or little-endian byte order. |
| boolean | [**matches**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#matches(javax.sound.sampled.AudioFormat))([AudioFormat](http://docs.google.com/javax/sound/sampled/AudioFormat.html) format)            Indicates whether this format matches the one specified. |
| [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Object](http://docs.google.com/java/lang/Object.html)> | [**properties**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#properties())()            Obtain an unmodifiable map of properties. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/sound/sampled/AudioFormat.html#toString())()            Returns a string that describes the format, such as: "PCM SIGNED 22050 Hz 16 bit mono big-endian". |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### encoding

protected [AudioFormat.Encoding](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) **encoding**

The audio encoding technique used by this format.

### sampleRate

protected float **sampleRate**

The number of samples played or recorded per second, for sounds that have this format.

### sampleSizeInBits

protected int **sampleSizeInBits**

The number of bits in each sample of a sound that has this format.

### channels

protected int **channels**

The number of audio channels in this format (1 for mono, 2 for stereo).

### frameSize

protected int **frameSize**

The number of bytes in each frame of a sound that has this format.

### frameRate

protected float **frameRate**

The number of frames played or recorded per second, for sounds that have this format.

### bigEndian

protected boolean **bigEndian**

Indicates whether the audio data is stored in big-endian or little-endian order.

| **Constructor Detail** |
| --- |

### AudioFormat

public **AudioFormat**([AudioFormat.Encoding](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) encoding,  
 float sampleRate,  
 int sampleSizeInBits,  
 int channels,  
 int frameSize,  
 float frameRate,  
 boolean bigEndian)

Constructs an AudioFormat with the given parameters. The encoding specifies the convention used to represent the data. The other parameters are further explained in the [class description](http://docs.google.com/javax/sound/sampled/AudioFormat.html).

**Parameters:**encoding - the audio encoding techniquesampleRate - the number of samples per secondsampleSizeInBits - the number of bits in each samplechannels - the number of channels (1 for mono, 2 for stereo, and so on)frameSize - the number of bytes in each frameframeRate - the number of frames per secondbigEndian - indicates whether the data for a single sample is stored in big-endian byte order (false means little-endian)

### AudioFormat

public **AudioFormat**([AudioFormat.Encoding](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) encoding,  
 float sampleRate,  
 int sampleSizeInBits,  
 int channels,  
 int frameSize,  
 float frameRate,  
 boolean bigEndian,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Object](http://docs.google.com/java/lang/Object.html)> properties)

Constructs an AudioFormat with the given parameters. The encoding specifies the convention used to represent the data. The other parameters are further explained in the [class description](http://docs.google.com/javax/sound/sampled/AudioFormat.html).

**Parameters:**encoding - the audio encoding techniquesampleRate - the number of samples per secondsampleSizeInBits - the number of bits in each samplechannels - the number of channels (1 for mono, 2 for stereo, and so on)frameSize - the number of bytes in each frameframeRate - the number of frames per secondbigEndian - indicates whether the data for a single sample is stored in big-endian byte order (false means little-endian)properties - a Map<String,Object> object containing format properties**Since:** 1.5

### AudioFormat

public **AudioFormat**(float sampleRate,  
 int sampleSizeInBits,  
 int channels,  
 boolean signed,  
 boolean bigEndian)

Constructs an AudioFormat with a linear PCM encoding and the given parameters. The frame size is set to the number of bytes required to contain one sample from each channel, and the frame rate is set to the sample rate.

**Parameters:**sampleRate - the number of samples per secondsampleSizeInBits - the number of bits in each samplechannels - the number of channels (1 for mono, 2 for stereo, and so on)signed - indicates whether the data is signed or unsignedbigEndian - indicates whether the data for a single sample is stored in big-endian byte order (false means little-endian)

| **Method Detail** |
| --- |

### getEncoding

public [AudioFormat.Encoding](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) **getEncoding**()

Obtains the type of encoding for sounds in this format.

**Returns:**the encoding type**See Also:**[AudioFormat.Encoding.PCM\_SIGNED](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html#PCM_SIGNED), [AudioFormat.Encoding.PCM\_UNSIGNED](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html#PCM_UNSIGNED), [AudioFormat.Encoding.ULAW](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html#ULAW), [AudioFormat.Encoding.ALAW](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html#ALAW)

### getSampleRate

public float **getSampleRate**()

Obtains the sample rate. For compressed formats, the return value is the sample rate of the uncompressed audio data. When this AudioFormat is used for queries (e.g. [AudioSystem.isConversionSupported](http://docs.google.com/javax/sound/sampled/AudioSystem.html#isConversionSupported(javax.sound.sampled.AudioFormat,%20javax.sound.sampled.AudioFormat))) or capabilities (e.g. [DataLine.Info.getFormats](http://docs.google.com/javax/sound/sampled/DataLine.Info.html#getFormats())), a sample rate of AudioSystem.NOT\_SPECIFIED means that any sample rate is acceptable. AudioSystem.NOT\_SPECIFIED is also returned when the sample rate is not defined for this audio format.

**Returns:**the number of samples per second, or AudioSystem.NOT\_SPECIFIED**See Also:**[getFrameRate()](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getFrameRate()), [AudioSystem.NOT\_SPECIFIED](http://docs.google.com/javax/sound/sampled/AudioSystem.html#NOT_SPECIFIED)

### getSampleSizeInBits

public int **getSampleSizeInBits**()

Obtains the size of a sample. For compressed formats, the return value is the sample size of the uncompressed audio data. When this AudioFormat is used for queries (e.g. [AudioSystem.isConversionSupported](http://docs.google.com/javax/sound/sampled/AudioSystem.html#isConversionSupported(javax.sound.sampled.AudioFormat,%20javax.sound.sampled.AudioFormat))) or capabilities (e.g. [DataLine.Info.getFormats](http://docs.google.com/javax/sound/sampled/DataLine.Info.html#getFormats())), a sample size of AudioSystem.NOT\_SPECIFIED means that any sample size is acceptable. AudioSystem.NOT\_SPECIFIED is also returned when the sample size is not defined for this audio format.

**Returns:**the number of bits in each sample, or AudioSystem.NOT\_SPECIFIED**See Also:**[getFrameSize()](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getFrameSize()), [AudioSystem.NOT\_SPECIFIED](http://docs.google.com/javax/sound/sampled/AudioSystem.html#NOT_SPECIFIED)

### getChannels

public int **getChannels**()

Obtains the number of channels. When this AudioFormat is used for queries (e.g. [AudioSystem.isConversionSupported](http://docs.google.com/javax/sound/sampled/AudioSystem.html#isConversionSupported(javax.sound.sampled.AudioFormat,%20javax.sound.sampled.AudioFormat))) or capabilities (e.g. [DataLine.Info.getFormats](http://docs.google.com/javax/sound/sampled/DataLine.Info.html#getFormats())), a return value of AudioSystem.NOT\_SPECIFIED means that any (positive) number of channels is acceptable.

**Returns:**The number of channels (1 for mono, 2 for stereo, etc.), or AudioSystem.NOT\_SPECIFIED**See Also:**[AudioSystem.NOT\_SPECIFIED](http://docs.google.com/javax/sound/sampled/AudioSystem.html#NOT_SPECIFIED)

### getFrameSize

public int **getFrameSize**()

Obtains the frame size in bytes. When this AudioFormat is used for queries (e.g. [AudioSystem.isConversionSupported](http://docs.google.com/javax/sound/sampled/AudioSystem.html#isConversionSupported(javax.sound.sampled.AudioFormat,%20javax.sound.sampled.AudioFormat))) or capabilities (e.g. [DataLine.Info.getFormats](http://docs.google.com/javax/sound/sampled/DataLine.Info.html#getFormats())), a frame size of AudioSystem.NOT\_SPECIFIED means that any frame size is acceptable. AudioSystem.NOT\_SPECIFIED is also returned when the frame size is not defined for this audio format.

**Returns:**the number of bytes per frame, or AudioSystem.NOT\_SPECIFIED**See Also:**[getSampleSizeInBits()](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getSampleSizeInBits()), [AudioSystem.NOT\_SPECIFIED](http://docs.google.com/javax/sound/sampled/AudioSystem.html#NOT_SPECIFIED)

### getFrameRate

public float **getFrameRate**()

Obtains the frame rate in frames per second. When this AudioFormat is used for queries (e.g. [AudioSystem.isConversionSupported](http://docs.google.com/javax/sound/sampled/AudioSystem.html#isConversionSupported(javax.sound.sampled.AudioFormat,%20javax.sound.sampled.AudioFormat))) or capabilities (e.g. [DataLine.Info.getFormats](http://docs.google.com/javax/sound/sampled/DataLine.Info.html#getFormats())), a frame rate of AudioSystem.NOT\_SPECIFIED means that any frame rate is acceptable. AudioSystem.NOT\_SPECIFIED is also returned when the frame rate is not defined for this audio format.

**Returns:**the number of frames per second, or AudioSystem.NOT\_SPECIFIED**See Also:**[getSampleRate()](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getSampleRate()), [AudioSystem.NOT\_SPECIFIED](http://docs.google.com/javax/sound/sampled/AudioSystem.html#NOT_SPECIFIED)

### isBigEndian

public boolean **isBigEndian**()

Indicates whether the audio data is stored in big-endian or little-endian byte order. If the sample size is not more than one byte, the return value is irrelevant.

**Returns:**true if the data is stored in big-endian byte order, false if little-endian

### properties

public [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[Object](http://docs.google.com/java/lang/Object.html)> **properties**()

Obtain an unmodifiable map of properties. The concept of properties is further explained in the [class description](http://docs.google.com/javax/sound/sampled/AudioFileFormat.html).

**Returns:**a Map<String,Object> object containing all properties. If no properties are recognized, an empty map is returned.**Since:** 1.5 **See Also:**[getProperty(String)](http://docs.google.com/javax/sound/sampled/AudioFormat.html#getProperty(java.lang.String))

### getProperty

public [Object](http://docs.google.com/java/lang/Object.html) **getProperty**([String](http://docs.google.com/java/lang/String.html) key)

Obtain the property value specified by the key. The concept of properties is further explained in the [class description](http://docs.google.com/javax/sound/sampled/AudioFileFormat.html).

If the specified property is not defined for a particular file format, this method returns null.

**Parameters:**key - the key of the desired property **Returns:**the value of the property with the specified key, or null if the property does not exist.**Since:** 1.5 **See Also:**[properties](http://docs.google.com/javax/sound/sampled/AudioFormat.html#properties)

### matches

public boolean **matches**([AudioFormat](http://docs.google.com/javax/sound/sampled/AudioFormat.html) format)

Indicates whether this format matches the one specified. To match, two formats must have the same encoding, the same number of channels, and the same number of bits per sample and bytes per frame. The two formats must also have the same sample rate, unless the specified format has the sample rate value AudioSystem.NOT\_SPECIFIED, which any sample rate will match. The frame rates must similarly be equal, unless the specified format has the frame rate value AudioSystem.NOT\_SPECIFIED. The byte order (big-endian or little-endian) must match if the sample size is greater than one byte.

**Parameters:**format - format to test for match **Returns:**true if this format matches the one specified, false otherwise.

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string that describes the format, such as: "PCM SIGNED 22050 Hz 16 bit mono big-endian". The contents of the string may vary between implementations of Java Sound.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string that describes the format parameters

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AudioFormat.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sound/sampled/AudioFileFormat.Type.html)   [**NEXT CLASS**](http://docs.google.com/javax/sound/sampled/AudioFormat.Encoding.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sound/sampled/AudioFormat.html)    [**NO FRAMES**](http://docs.google.com/AudioFormat.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).