| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Mixer.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/LineUnavailableException.html)   [**NEXT CLASS**](http://docs.google.com/javax/sound/sampled/Mixer.Info.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sound/sampled/Mixer.html)    [**NO FRAMES**](http://docs.google.com/Mixer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | CONSTR | [METHOD](#2et92p0) | DETAIL: FIELD | CONSTR | [METHOD](#3dy6vkm) |

## **javax.sound.sampled**

Interface Mixer

**All Superinterfaces:** [Line](http://docs.google.com/javax/sound/sampled/Line.html)

public interface **Mixer**extends [Line](http://docs.google.com/javax/sound/sampled/Line.html)

A mixer is an audio device with one or more lines. It need not be designed for mixing audio signals. A mixer that actually mixes audio has multiple input (source) lines and at least one output (target) line. The former are often instances of classes that implement [SourceDataLine](http://docs.google.com/javax/sound/sampled/SourceDataLine.html), and the latter, [TargetDataLine](http://docs.google.com/javax/sound/sampled/TargetDataLine.html). [Port](http://docs.google.com/javax/sound/sampled/Port.html) objects, too, are either source lines or target lines. A mixer can accept prerecorded, loopable sound as input, by having some of its source lines be instances of objects that implement the [Clip](http://docs.google.com/javax/sound/sampled/Clip.html) interface.

Through methods of the Line interface, which Mixer extends, a mixer might provide a set of controls that are global to the mixer. For example, the mixer can have a master gain control. These global controls are distinct from the controls belonging to each of the mixer's individual lines.

Some mixers, especially those with internal digital mixing capabilities, may provide additional capabilities by implementing the DataLine interface.

A mixer can support synchronization of its lines. When one line in a synchronized group is started or stopped, the other lines in the group automatically start or stop simultaneously with the explicitly affected one.

**Since:** 1.3

| **Nested Class Summary** | |
| --- | --- |
| static class | [**Mixer.Info**](http://docs.google.com/javax/sound/sampled/Mixer.Info.html)            The Mixer.Info class represents information about an audio mixer, including the product's name, version, and vendor, along with a textual description. |

| **Method Summary** | |
| --- | --- |
| [Line](http://docs.google.com/javax/sound/sampled/Line.html) | [**getLine**](http://docs.google.com/javax/sound/sampled/Mixer.html#getLine(javax.sound.sampled.Line.Info))([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)            Obtains a line that is available for use and that matches the description in the specified Line.Info object. |
| int | [**getMaxLines**](http://docs.google.com/javax/sound/sampled/Mixer.html#getMaxLines(javax.sound.sampled.Line.Info))([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)            Obtains the approximate maximum number of lines of the requested type that can be open simultaneously on the mixer. |
| [Mixer.Info](http://docs.google.com/javax/sound/sampled/Mixer.Info.html) | [**getMixerInfo**](http://docs.google.com/javax/sound/sampled/Mixer.html#getMixerInfo())()            Obtains information about this mixer, including the product's name, version, vendor, etc. |
| [Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html)[] | [**getSourceLineInfo**](http://docs.google.com/javax/sound/sampled/Mixer.html#getSourceLineInfo())()            Obtains information about the set of source lines supported by this mixer. |
| [Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html)[] | [**getSourceLineInfo**](http://docs.google.com/javax/sound/sampled/Mixer.html#getSourceLineInfo(javax.sound.sampled.Line.Info))([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)            Obtains information about source lines of a particular type supported by the mixer. |
| [Line](http://docs.google.com/javax/sound/sampled/Line.html)[] | [**getSourceLines**](http://docs.google.com/javax/sound/sampled/Mixer.html#getSourceLines())()            Obtains the set of all source lines currently open to this mixer. |
| [Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html)[] | [**getTargetLineInfo**](http://docs.google.com/javax/sound/sampled/Mixer.html#getTargetLineInfo())()            Obtains information about the set of target lines supported by this mixer. |
| [Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html)[] | [**getTargetLineInfo**](http://docs.google.com/javax/sound/sampled/Mixer.html#getTargetLineInfo(javax.sound.sampled.Line.Info))([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)            Obtains information about target lines of a particular type supported by the mixer. |
| [Line](http://docs.google.com/javax/sound/sampled/Line.html)[] | [**getTargetLines**](http://docs.google.com/javax/sound/sampled/Mixer.html#getTargetLines())()            Obtains the set of all target lines currently open from this mixer. |
| boolean | [**isLineSupported**](http://docs.google.com/javax/sound/sampled/Mixer.html#isLineSupported(javax.sound.sampled.Line.Info))([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)            Indicates whether the mixer supports a line (or lines) that match the specified Line.Info object. |
| boolean | [**isSynchronizationSupported**](http://docs.google.com/javax/sound/sampled/Mixer.html#isSynchronizationSupported(javax.sound.sampled.Line%5B%5D,%20boolean))([Line](http://docs.google.com/javax/sound/sampled/Line.html)[] lines, boolean maintainSync)            Reports whether this mixer supports synchronization of the specified set of lines. |
| void | [**synchronize**](http://docs.google.com/javax/sound/sampled/Mixer.html#synchronize(javax.sound.sampled.Line%5B%5D,%20boolean))([Line](http://docs.google.com/javax/sound/sampled/Line.html)[] lines, boolean maintainSync)            Synchronizes two or more lines. |
| void | [**unsynchronize**](http://docs.google.com/javax/sound/sampled/Mixer.html#unsynchronize(javax.sound.sampled.Line%5B%5D))([Line](http://docs.google.com/javax/sound/sampled/Line.html)[] lines)            Releases synchronization for the specified lines. |

| **Methods inherited from interface javax.sound.sampled.**[**Line**](http://docs.google.com/javax/sound/sampled/Line.html) |
| --- |
| [addLineListener](http://docs.google.com/javax/sound/sampled/Line.html#addLineListener(javax.sound.sampled.LineListener)), [close](http://docs.google.com/javax/sound/sampled/Line.html#close()), [getControl](http://docs.google.com/javax/sound/sampled/Line.html#getControl(javax.sound.sampled.Control.Type)), [getControls](http://docs.google.com/javax/sound/sampled/Line.html#getControls()), [getLineInfo](http://docs.google.com/javax/sound/sampled/Line.html#getLineInfo()), [isControlSupported](http://docs.google.com/javax/sound/sampled/Line.html#isControlSupported(javax.sound.sampled.Control.Type)), [isOpen](http://docs.google.com/javax/sound/sampled/Line.html#isOpen()), [open](http://docs.google.com/javax/sound/sampled/Line.html#open()), [removeLineListener](http://docs.google.com/javax/sound/sampled/Line.html#removeLineListener(javax.sound.sampled.LineListener)) |

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

### getMixerInfo

[Mixer.Info](http://docs.google.com/javax/sound/sampled/Mixer.Info.html) **getMixerInfo**()

Obtains information about this mixer, including the product's name, version, vendor, etc.

**Returns:**a mixer info object that describes this mixer**See Also:**[Mixer.Info](http://docs.google.com/javax/sound/sampled/Mixer.Info.html)

### getSourceLineInfo

[Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html)[] **getSourceLineInfo**()

Obtains information about the set of source lines supported by this mixer. Some source lines may only be available when this mixer is open.

**Returns:**array of Line.Info objects representing source lines for this mixer. If no source lines are supported, an array of length 0 is returned.

### getTargetLineInfo

[Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html)[] **getTargetLineInfo**()

Obtains information about the set of target lines supported by this mixer. Some target lines may only be available when this mixer is open.

**Returns:**array of Line.Info objects representing target lines for this mixer. If no target lines are supported, an array of length 0 is returned.

### getSourceLineInfo

[Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html)[] **getSourceLineInfo**([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)

Obtains information about source lines of a particular type supported by the mixer. Some source lines may only be available when this mixer is open.

**Parameters:**info - a Line.Info object describing lines about which information is queried **Returns:**an array of Line.Info objects describing source lines matching the type requested. If no matching source lines are supported, an array of length 0 is returned.

### getTargetLineInfo

[Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html)[] **getTargetLineInfo**([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)

Obtains information about target lines of a particular type supported by the mixer. Some target lines may only be available when this mixer is open.

**Parameters:**info - a Line.Info object describing lines about which information is queried **Returns:**an array of Line.Info objects describing target lines matching the type requested. If no matching target lines are supported, an array of length 0 is returned.

### isLineSupported

boolean **isLineSupported**([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)

Indicates whether the mixer supports a line (or lines) that match the specified Line.Info object. Some lines may only be supported when this mixer is open.

**Parameters:**info - describes the line for which support is queried **Returns:**true if at least one matching line is supported, false otherwise

### getLine

[Line](http://docs.google.com/javax/sound/sampled/Line.html) **getLine**([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)  
 throws [LineUnavailableException](http://docs.google.com/javax/sound/sampled/LineUnavailableException.html)

Obtains a line that is available for use and that matches the description in the specified Line.Info object.

If a DataLine is requested, and info is an instance of DataLine.Info specifying at least one fully qualified audio format, the last one will be used as the default format of the returned DataLine.

**Parameters:**info - describes the desired line **Throws:** [LineUnavailableException](http://docs.google.com/javax/sound/sampled/LineUnavailableException.html) - if a matching line is not available due to resource restrictions [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if this mixer does not support any lines matching the description [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a matching line is not available due to security restrictions

### getMaxLines

int **getMaxLines**([Line.Info](http://docs.google.com/javax/sound/sampled/Line.Info.html) info)

Obtains the approximate maximum number of lines of the requested type that can be open simultaneously on the mixer. Certain types of mixers do not have a hard bound and may allow opening more lines. Since certain lines are a shared resource, a mixer may not be able to open the maximum number of lines if another process has opened lines of this mixer. The requested type is any line that matches the description in the provided Line.Info object. For example, if the info object represents a speaker port, and the mixer supports exactly one speaker port, this method should return 1. If the info object represents a source data line and the mixer supports the use of 32 source data lines simultaneously, the return value should be 32. If there is no limit, this function returns AudioSystem.NOT\_SPECIFIED.

**Parameters:**info - a Line.Info that describes the line for which the number of supported instances is queried **Returns:**the maximum number of matching lines supported, or AudioSystem.NOT\_SPECIFIED

### getSourceLines

[Line](http://docs.google.com/javax/sound/sampled/Line.html)[] **getSourceLines**()

Obtains the set of all source lines currently open to this mixer.

**Returns:**the source lines currently open to the mixer. If no source lines are currently open to this mixer, an array of length 0 is returned. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the matching lines are not available due to security restrictions

### getTargetLines

[Line](http://docs.google.com/javax/sound/sampled/Line.html)[] **getTargetLines**()

Obtains the set of all target lines currently open from this mixer.

**Returns:**target lines currently open from the mixer. If no target lines are currently open from this mixer, an array of length 0 is returned. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the matching lines are not available due to security restrictions

### synchronize

void **synchronize**([Line](http://docs.google.com/javax/sound/sampled/Line.html)[] lines,  
 boolean maintainSync)

Synchronizes two or more lines. Any subsequent command that starts or stops audio playback or capture for one of these lines will exert the same effect on the other lines in the group, so that they start or stop playing or capturing data simultaneously.

**Parameters:**lines - the lines that should be synchronizedmaintainSync - true if the synchronization must be precisely maintained (i.e., the synchronization must be sample-accurate) at all times during operation of the lines , or false if precise synchronization is required only during start and stop operations **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the lines cannot be synchronized. This may occur if the lines are of different types or have different formats for which this mixer does not support synchronization, or if all lines specified do not belong to this mixer.

### unsynchronize

void **unsynchronize**([Line](http://docs.google.com/javax/sound/sampled/Line.html)[] lines)

Releases synchronization for the specified lines. The array must be identical to one for which synchronization has already been established; otherwise an exception may be thrown. However, null may be specified, in which case all currently synchronized lines that belong to this mixer are unsynchronized.

**Parameters:**lines - the synchronized lines for which synchronization should be released, or null for all this mixer's synchronized lines **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the lines cannot be unsynchronized. This may occur if the argument specified does not exactly match a set of lines for which synchronization has already been established.

### isSynchronizationSupported

boolean **isSynchronizationSupported**([Line](http://docs.google.com/javax/sound/sampled/Line.html)[] lines,  
 boolean maintainSync)

Reports whether this mixer supports synchronization of the specified set of lines.

**Parameters:**lines - the set of lines for which synchronization support is queriedmaintainSync - true if the synchronization must be precisely maintained (i.e., the synchronization must be sample-accurate) at all times during operation of the lines , or false if precise synchronization is required only during start and stop operations **Returns:**true if the lines can be synchronized, false otherwise

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Mixer.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/LineUnavailableException.html)   [**NEXT CLASS**](http://docs.google.com/javax/sound/sampled/Mixer.Info.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sound/sampled/Mixer.html)    [**NO FRAMES**](http://docs.google.com/Mixer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | CONSTR | [METHOD](#2et92p0) | DETAIL: FIELD | CONSTR | [METHOD](#3dy6vkm) |

[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).