| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TargetDataLine.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/SourceDataLine.html)   [**NEXT CLASS**](http://docs.google.com/javax/sound/sampled/UnsupportedAudioFileException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sound/sampled/TargetDataLine.html)    [**NO FRAMES**](http://docs.google.com/TargetDataLine.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

## **javax.sound.sampled**

Interface TargetDataLine

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

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

A target data line is a type of [DataLine](http://docs.google.com/javax/sound/sampled/DataLine.html) from which audio data can be read. The most common example is a data line that gets its data from an audio capture device. (The device is implemented as a mixer that writes to the target data line.)

Note that the naming convention for this interface reflects the relationship between the line and its mixer. From the perspective of an application, a target data line may act as a source for audio data.

The target data line can be obtained from a mixer by invoking the [getLine](http://docs.google.com/javax/sound/sampled/Mixer.html#getLine(javax.sound.sampled.Line.Info)) method of Mixer with an appropriate [DataLine.Info](http://docs.google.com/javax/sound/sampled/DataLine.Info.html) object.

The TargetDataLine interface provides a method for reading the captured data from the target data line's buffer.Applications that record audio should read data from the target data line quickly enough to keep the buffer from overflowing, which could cause discontinuities in the captured data that are perceived as clicks. Applications can use the [available](http://docs.google.com/javax/sound/sampled/DataLine.html#available()) method defined in the DataLine interface to determine the amount of data currently queued in the data line's buffer. If the buffer does overflow, the oldest queued data is discarded and replaced by new data.

**Since:** 1.3 **See Also:**[Mixer](http://docs.google.com/javax/sound/sampled/Mixer.html), [DataLine](http://docs.google.com/javax/sound/sampled/DataLine.html), [SourceDataLine](http://docs.google.com/javax/sound/sampled/SourceDataLine.html)

| **Nested Class Summary** | |
| --- | --- |

| **Nested classes/interfaces inherited from interface javax.sound.sampled.**[**DataLine**](http://docs.google.com/javax/sound/sampled/DataLine.html) |
| --- |
| [DataLine.Info](http://docs.google.com/javax/sound/sampled/DataLine.Info.html) |

| **Method Summary** | |
| --- | --- |
| void | [**open**](http://docs.google.com/javax/sound/sampled/TargetDataLine.html#open(javax.sound.sampled.AudioFormat))([AudioFormat](http://docs.google.com/javax/sound/sampled/AudioFormat.html) format)            Opens the line with the specified format, causing the line to acquire any required system resources and become operational. |
| void | [**open**](http://docs.google.com/javax/sound/sampled/TargetDataLine.html#open(javax.sound.sampled.AudioFormat,%20int))([AudioFormat](http://docs.google.com/javax/sound/sampled/AudioFormat.html) format, int bufferSize)            Opens the line with the specified format and requested buffer size, causing the line to acquire any required system resources and become operational. |
| int | [**read**](http://docs.google.com/javax/sound/sampled/TargetDataLine.html#read(byte%5B%5D,%20int,%20int))(byte[] b, int off, int len)            Reads audio data from the data line's input buffer. |

| **Methods inherited from interface javax.sound.sampled.**[**DataLine**](http://docs.google.com/javax/sound/sampled/DataLine.html) |
| --- |
| [available](http://docs.google.com/javax/sound/sampled/DataLine.html#available()), [drain](http://docs.google.com/javax/sound/sampled/DataLine.html#drain()), [flush](http://docs.google.com/javax/sound/sampled/DataLine.html#flush()), [getBufferSize](http://docs.google.com/javax/sound/sampled/DataLine.html#getBufferSize()), [getFormat](http://docs.google.com/javax/sound/sampled/DataLine.html#getFormat()), [getFramePosition](http://docs.google.com/javax/sound/sampled/DataLine.html#getFramePosition()), [getLevel](http://docs.google.com/javax/sound/sampled/DataLine.html#getLevel()), [getLongFramePosition](http://docs.google.com/javax/sound/sampled/DataLine.html#getLongFramePosition()), [getMicrosecondPosition](http://docs.google.com/javax/sound/sampled/DataLine.html#getMicrosecondPosition()), [isActive](http://docs.google.com/javax/sound/sampled/DataLine.html#isActive()), [isRunning](http://docs.google.com/javax/sound/sampled/DataLine.html#isRunning()), [start](http://docs.google.com/javax/sound/sampled/DataLine.html#start()), [stop](http://docs.google.com/javax/sound/sampled/DataLine.html#stop()) |

| **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** |
| --- |

### open

void **open**([AudioFormat](http://docs.google.com/javax/sound/sampled/AudioFormat.html) format,  
 int bufferSize)  
 throws [LineUnavailableException](http://docs.google.com/javax/sound/sampled/LineUnavailableException.html)

Opens the line with the specified format and requested buffer size, causing the line to acquire any required system resources and become operational.

The buffer size is specified in bytes, but must represent an integral number of sample frames. Invoking this method with a requested buffer size that does not meet this requirement may result in an IllegalArgumentException. The actual buffer size for the open line may differ from the requested buffer size. The value actually set may be queried by subsequently calling [DataLine.getBufferSize()](http://docs.google.com/javax/sound/sampled/DataLine.html#getBufferSize())

If this operation succeeds, the line is marked as open, and an [OPEN](http://docs.google.com/javax/sound/sampled/LineEvent.Type.html#OPEN) event is dispatched to the line's listeners.

Invoking this method on a line that is already open is illegal and may result in an IllegalStateException.

Some lines, once closed, cannot be reopened. Attempts to reopen such a line will always result in a LineUnavailableException.

**Parameters:**format - the desired audio formatbufferSize - the desired buffer size, in bytes. **Throws:** [LineUnavailableException](http://docs.google.com/javax/sound/sampled/LineUnavailableException.html) - if the line cannot be opened due to resource restrictions [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the buffer size does not represent an integral number of sample frames, or if format is not fully specified or invalid [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the line is already open [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the line cannot be opened due to security restrictions**See Also:**[open(AudioFormat)](http://docs.google.com/javax/sound/sampled/TargetDataLine.html#open(javax.sound.sampled.AudioFormat)), [Line.open()](http://docs.google.com/javax/sound/sampled/Line.html#open()), [Line.close()](http://docs.google.com/javax/sound/sampled/Line.html#close()), [Line.isOpen()](http://docs.google.com/javax/sound/sampled/Line.html#isOpen()), [LineEvent](http://docs.google.com/javax/sound/sampled/LineEvent.html)

### open

void **open**([AudioFormat](http://docs.google.com/javax/sound/sampled/AudioFormat.html) format)  
 throws [LineUnavailableException](http://docs.google.com/javax/sound/sampled/LineUnavailableException.html)

Opens the line with the specified format, causing the line to acquire any required system resources and become operational.

The implementation chooses a buffer size, which is measured in bytes but which encompasses an integral number of sample frames. The buffer size that the system has chosen may be queried by subsequently calling [DataLine.getBufferSize()](http://docs.google.com/javax/sound/sampled/DataLine.html#getBufferSize())

If this operation succeeds, the line is marked as open, and an [OPEN](http://docs.google.com/javax/sound/sampled/LineEvent.Type.html#OPEN) event is dispatched to the line's listeners.

Invoking this method on a line that is already open is illegal and may result in an IllegalStateException.

Some lines, once closed, cannot be reopened. Attempts to reopen such a line will always result in a LineUnavailableException.

**Parameters:**format - the desired audio format **Throws:** [LineUnavailableException](http://docs.google.com/javax/sound/sampled/LineUnavailableException.html) - if the line cannot be opened due to resource restrictions [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if format is not fully specified or invalid [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the line is already open [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the line cannot be opened due to security restrictions**See Also:**[open(AudioFormat, int)](http://docs.google.com/javax/sound/sampled/TargetDataLine.html#open(javax.sound.sampled.AudioFormat,%20int)), [Line.open()](http://docs.google.com/javax/sound/sampled/Line.html#open()), [Line.close()](http://docs.google.com/javax/sound/sampled/Line.html#close()), [Line.isOpen()](http://docs.google.com/javax/sound/sampled/Line.html#isOpen()), [LineEvent](http://docs.google.com/javax/sound/sampled/LineEvent.html)

### read

int **read**(byte[] b,  
 int off,  
 int len)

Reads audio data from the data line's input buffer. The requested number of bytes is read into the specified array, starting at the specified offset into the array in bytes. This method blocks until the requested amount of data has been read. However, if the data line is closed, stopped, drained, or flushed before the requested amount has been read, the method no longer blocks, but returns the number of bytes read thus far.

The number of bytes that can be read without blocking can be ascertained using the [available](http://docs.google.com/javax/sound/sampled/DataLine.html#available()) method of the DataLine interface. (While it is guaranteed that this number of bytes can be read without blocking, there is no guarantee that attempts to read additional data will block.)

The number of bytes to be read must represent an integral number of sample frames, such that:

[ bytes read ] % [frame size in bytes ] == 0

The return value will always meet this requirement. A request to read a number of bytes representing a non-integral number of sample frames cannot be fulfilled and may result in an IllegalArgumentException.

**Parameters:**b - a byte array that will contain the requested input data when this method returnsoff - the offset from the beginning of the array, in byteslen - the requested number of bytes to read **Returns:**the number of bytes actually read **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the requested number of bytes does not represent an integral number of sample frames. or if len is negative. [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if off is negative, or off+len is greater than the length of the array b.**See Also:**[SourceDataLine.write(byte[], int, int)](http://docs.google.com/javax/sound/sampled/SourceDataLine.html#write(byte%5B%5D,%20int,%20int)), [DataLine.available()](http://docs.google.com/javax/sound/sampled/DataLine.html#available())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TargetDataLine.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/SourceDataLine.html)   [**NEXT CLASS**](http://docs.google.com/javax/sound/sampled/UnsupportedAudioFileException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sound/sampled/TargetDataLine.html)    [**NO FRAMES**](http://docs.google.com/TargetDataLine.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#4d34og8) |

[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).