| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Synthesizer.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/midi/SoundbankResource.html)   [**NEXT CLASS**](http://docs.google.com/javax/sound/midi/SysexMessage.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sound/midi/Synthesizer.html)    [**NO FRAMES**](http://docs.google.com/Synthesizer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

## **javax.sound.midi**

Interface Synthesizer

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

public interface **Synthesizer**extends [MidiDevice](http://docs.google.com/javax/sound/midi/MidiDevice.html)

A Synthesizer generates sound. This usually happens when one of the Synthesizer's [MidiChannel](http://docs.google.com/javax/sound/midi/MidiChannel.html) objects receives a [noteOn](http://docs.google.com/javax/sound/midi/MidiChannel.html#noteOn(int,%20int)) message, either directly or via the Synthesizer object. Many Synthesizers support Receivers, through which MIDI events can be delivered to the Synthesizer. In such cases, the Synthesizer typically responds by sending a corresponding message to the appropriate MidiChannel, or by processing the event itself if the event isn't one of the MIDI channel messages.

The Synthesizer interface includes methods for loading and unloading instruments from soundbanks. An instrument is a specification for synthesizing a certain type of sound, whether that sound emulates a traditional instrument or is some kind of sound effect or other imaginary sound. A soundbank is a collection of instruments, organized by bank and program number (via the instrument's Patch object). Different Synthesizer classes might implement different sound-synthesis techniques, meaning that some instruments and not others might be compatible with a given synthesizer. Also, synthesizers may have a limited amount of memory for instruments, meaning that not every soundbank and instrument can be used by every synthesizer, even if the synthesis technique is compatible. To see whether the instruments from a certain soundbank can be played by a given synthesizer, invoke the [isSoundbankSupported](http://docs.google.com/javax/sound/midi/Synthesizer.html#isSoundbankSupported(javax.sound.midi.Soundbank)) method of Synthesizer.

"Loading" an instrument means that that instrument becomes available for synthesizing notes. The instrument is loaded into the bank and program location specified by its Patch object. Loading does not necessarily mean that subsequently played notes will immediately have the sound of this newly loaded instrument. For the instrument to play notes, one of the synthesizer's MidiChannel objects must receive (or have received) a program-change message that causes that particular instrument's bank and program number to be selected.

**See Also:**[MidiSystem.getSynthesizer()](http://docs.google.com/javax/sound/midi/MidiSystem.html#getSynthesizer()), [Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html), [Instrument](http://docs.google.com/javax/sound/midi/Instrument.html), [MidiChannel.programChange(int, int)](http://docs.google.com/javax/sound/midi/MidiChannel.html#programChange(int,%20int)), [Receiver](http://docs.google.com/javax/sound/midi/Receiver.html), [Transmitter](http://docs.google.com/javax/sound/midi/Transmitter.html), [MidiDevice](http://docs.google.com/javax/sound/midi/MidiDevice.html)

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

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

| **Method Summary** | |
| --- | --- |
| [Instrument](http://docs.google.com/javax/sound/midi/Instrument.html)[] | [**getAvailableInstruments**](http://docs.google.com/javax/sound/midi/Synthesizer.html#getAvailableInstruments())()            Obtains a list of instruments that come with the synthesizer. |
| [MidiChannel](http://docs.google.com/javax/sound/midi/MidiChannel.html)[] | [**getChannels**](http://docs.google.com/javax/sound/midi/Synthesizer.html#getChannels())()            Obtains the set of MIDI channels controlled by this synthesizer. |
| [Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) | [**getDefaultSoundbank**](http://docs.google.com/javax/sound/midi/Synthesizer.html#getDefaultSoundbank())()            Obtains the default soundbank for the synthesizer, if one exists. |
| long | [**getLatency**](http://docs.google.com/javax/sound/midi/Synthesizer.html#getLatency())()            Obtains the processing latency incurred by this synthesizer, expressed in microseconds. |
| [Instrument](http://docs.google.com/javax/sound/midi/Instrument.html)[] | [**getLoadedInstruments**](http://docs.google.com/javax/sound/midi/Synthesizer.html#getLoadedInstruments())()            Obtains a list of the instruments that are currently loaded onto this Synthesizer. |
| int | [**getMaxPolyphony**](http://docs.google.com/javax/sound/midi/Synthesizer.html#getMaxPolyphony())()            Obtains the maximum number of notes that this synthesizer can sound simultaneously. |
| [VoiceStatus](http://docs.google.com/javax/sound/midi/VoiceStatus.html)[] | [**getVoiceStatus**](http://docs.google.com/javax/sound/midi/Synthesizer.html#getVoiceStatus())()            Obtains the current status of the voices produced by this synthesizer. |
| boolean | [**isSoundbankSupported**](http://docs.google.com/javax/sound/midi/Synthesizer.html#isSoundbankSupported(javax.sound.midi.Soundbank))([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank)            Informs the caller whether this synthesizer is capable of loading instruments from the specified soundbank. |
| boolean | [**loadAllInstruments**](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadAllInstruments(javax.sound.midi.Soundbank))([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank)            Loads onto the Synthesizer all instruments contained in the specified Soundbank. |
| boolean | [**loadInstrument**](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstrument(javax.sound.midi.Instrument))([Instrument](http://docs.google.com/javax/sound/midi/Instrument.html) instrument)            Makes a particular instrument available for synthesis. |
| boolean | [**loadInstruments**](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D))([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank, [Patch](http://docs.google.com/javax/sound/midi/Patch.html)[] patchList)            Loads the instruments referenced by the specified patches, from the specified Soundbank. |
| boolean | [**remapInstrument**](http://docs.google.com/javax/sound/midi/Synthesizer.html#remapInstrument(javax.sound.midi.Instrument,%20javax.sound.midi.Instrument))([Instrument](http://docs.google.com/javax/sound/midi/Instrument.html) from, [Instrument](http://docs.google.com/javax/sound/midi/Instrument.html) to)            Remaps an instrument. |
| void | [**unloadAllInstruments**](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadAllInstruments(javax.sound.midi.Soundbank))([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank)            Unloads all instruments contained in the specified Soundbank. |
| void | [**unloadInstrument**](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadInstrument(javax.sound.midi.Instrument))([Instrument](http://docs.google.com/javax/sound/midi/Instrument.html) instrument)            Unloads a particular instrument. |
| void | [**unloadInstruments**](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D))([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank, [Patch](http://docs.google.com/javax/sound/midi/Patch.html)[] patchList)            Unloads the instruments referenced by the specified patches, from the MIDI sound bank specified. |

| **Methods inherited from interface javax.sound.midi.**[**MidiDevice**](http://docs.google.com/javax/sound/midi/MidiDevice.html) |
| --- |
| [close](http://docs.google.com/javax/sound/midi/MidiDevice.html#close()), [getDeviceInfo](http://docs.google.com/javax/sound/midi/MidiDevice.html#getDeviceInfo()), [getMaxReceivers](http://docs.google.com/javax/sound/midi/MidiDevice.html#getMaxReceivers()), [getMaxTransmitters](http://docs.google.com/javax/sound/midi/MidiDevice.html#getMaxTransmitters()), [getMicrosecondPosition](http://docs.google.com/javax/sound/midi/MidiDevice.html#getMicrosecondPosition()), [getReceiver](http://docs.google.com/javax/sound/midi/MidiDevice.html#getReceiver()), [getReceivers](http://docs.google.com/javax/sound/midi/MidiDevice.html#getReceivers()), [getTransmitter](http://docs.google.com/javax/sound/midi/MidiDevice.html#getTransmitter()), [getTransmitters](http://docs.google.com/javax/sound/midi/MidiDevice.html#getTransmitters()), [isOpen](http://docs.google.com/javax/sound/midi/MidiDevice.html#isOpen()), [open](http://docs.google.com/javax/sound/midi/MidiDevice.html#open()) |

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

### getMaxPolyphony

int **getMaxPolyphony**()

Obtains the maximum number of notes that this synthesizer can sound simultaneously.

**Returns:**the maximum number of simultaneous notes**See Also:**[getVoiceStatus()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getVoiceStatus())

### getLatency

long **getLatency**()

Obtains the processing latency incurred by this synthesizer, expressed in microseconds. This latency measures the worst-case delay between the time a MIDI message is delivered to the synthesizer and the time that the synthesizer actually produces the corresponding result.

Although the latency is expressed in microseconds, a synthesizer's actual measured delay may vary over a wider range than this resolution suggests. For example, a synthesizer might have a worst-case delay of a few milliseconds or more.

**Returns:**the worst-case delay, in microseconds

### getChannels

[MidiChannel](http://docs.google.com/javax/sound/midi/MidiChannel.html)[] **getChannels**()

Obtains the set of MIDI channels controlled by this synthesizer. Each non-null element in the returned array is a MidiChannel that receives the MIDI messages sent on that channel number.

The MIDI 1.0 specification provides for 16 channels, so this method returns an array of at least 16 elements. However, if this synthesizer doesn't make use of all 16 channels, some of the elements of the array might be null, so you should check each element before using it.

**Returns:**an array of the MidiChannel objects managed by this Synthesizer. Some of the array elements may be null.

### getVoiceStatus

[VoiceStatus](http://docs.google.com/javax/sound/midi/VoiceStatus.html)[] **getVoiceStatus**()

Obtains the current status of the voices produced by this synthesizer. If this class of Synthesizer does not provide voice information, the returned array will always be of length 0. Otherwise, its length is always equal to the total number of voices, as returned by getMaxPolyphony(). (See the VoiceStatus class description for an explanation of synthesizer voices.)

**Returns:**an array of VoiceStatus objects that supply information about the corresponding synthesizer voices**See Also:**[getMaxPolyphony()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getMaxPolyphony()), [VoiceStatus](http://docs.google.com/javax/sound/midi/VoiceStatus.html)

### isSoundbankSupported

boolean **isSoundbankSupported**([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank)

Informs the caller whether this synthesizer is capable of loading instruments from the specified soundbank. If the soundbank is unsupported, any attempts to load instruments from it will result in an IllegalArgumentException.

**Parameters:**soundbank - soundbank for which support is queried **Returns:**true if the soundbank is supported, otherwise false**See Also:**[loadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D)), [loadAllInstruments(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadAllInstruments(javax.sound.midi.Soundbank)), [unloadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D)), [unloadAllInstruments(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadAllInstruments(javax.sound.midi.Soundbank)), [getDefaultSoundbank()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getDefaultSoundbank())

### loadInstrument

boolean **loadInstrument**([Instrument](http://docs.google.com/javax/sound/midi/Instrument.html) instrument)

Makes a particular instrument available for synthesis. This instrument is loaded into the patch location specified by its Patch object, so that if a program-change message is received (or has been received) that causes that patch to be selected, subsequent notes will be played using the sound of instrument. If the specified instrument is already loaded, this method does nothing and returns true.

The instrument must be part of a soundbank that this Synthesizer supports. (To make sure, you can use the getSoundbank method of Instrument and the isSoundbankSupported method of Synthesizer.)

**Parameters:**instrument - instrument to load **Returns:**true if the instrument is successfully loaded (or already had been), false if the instrument could not be loaded (for example, if the synthesizer has insufficient memory to load it) **Throws:** IllegalArgumentException - if this Synthesizer doesn't support the specified instrument's soundbank**See Also:**[unloadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadInstrument(javax.sound.midi.Instrument)), [loadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D)), [loadAllInstruments(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadAllInstruments(javax.sound.midi.Soundbank)), [remapInstrument(javax.sound.midi.Instrument, javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#remapInstrument(javax.sound.midi.Instrument,%20javax.sound.midi.Instrument)), [SoundbankResource.getSoundbank()](http://docs.google.com/javax/sound/midi/SoundbankResource.html#getSoundbank()), [MidiChannel.programChange(int, int)](http://docs.google.com/javax/sound/midi/MidiChannel.html#programChange(int,%20int))

### unloadInstrument

void **unloadInstrument**([Instrument](http://docs.google.com/javax/sound/midi/Instrument.html) instrument)

Unloads a particular instrument.

**Parameters:**instrument - instrument to unload **Throws:** IllegalArgumentException - if this Synthesizer doesn't support the specified instrument's soundbank**See Also:**[loadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstrument(javax.sound.midi.Instrument)), [unloadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D)), [unloadAllInstruments(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadAllInstruments(javax.sound.midi.Soundbank)), [getLoadedInstruments()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getLoadedInstruments()), [remapInstrument(javax.sound.midi.Instrument, javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#remapInstrument(javax.sound.midi.Instrument,%20javax.sound.midi.Instrument))

### remapInstrument

boolean **remapInstrument**([Instrument](http://docs.google.com/javax/sound/midi/Instrument.html) from,  
 [Instrument](http://docs.google.com/javax/sound/midi/Instrument.html) to)

Remaps an instrument. Instrument to takes the place of instrument from.

For example, if from was located at bank number 2, program number 11, remapping causes that bank and program location to be occupied instead by to.

If the function succeeds, instrument from is unloaded.

To cancel the remapping reload instrument from by invoking one of [loadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstrument(javax.sound.midi.Instrument)), [loadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D)) or [loadAllInstruments(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadAllInstruments(javax.sound.midi.Soundbank)).

**Parameters:**from - the Instrument object to be replacedto - the Instrument object to be used in place of the old instrument, it should be loaded into the synthesizer **Returns:**true if the instrument succeessfully remapped, false if feature is not implemented by synthesizer **Throws:** IllegalArgumentException - if instrument from or instrument to aren't supported by synthesizer or if instrument to is not loaded NullPointerException - if from or to parameters have null value**See Also:**[loadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstrument(javax.sound.midi.Instrument)), [loadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D)), [loadAllInstruments(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadAllInstruments(javax.sound.midi.Soundbank))

### getDefaultSoundbank

[Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) **getDefaultSoundbank**()

Obtains the default soundbank for the synthesizer, if one exists. (Some synthesizers provide a default or built-in soundbank.) If a synthesizer doesn't have a default soundbank, instruments must be loaded explicitly from an external soundbank.

**Returns:**default soundbank, or null if one does not exist.**See Also:**[isSoundbankSupported(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#isSoundbankSupported(javax.sound.midi.Soundbank))

### getAvailableInstruments

[Instrument](http://docs.google.com/javax/sound/midi/Instrument.html)[] **getAvailableInstruments**()

Obtains a list of instruments that come with the synthesizer. These instruments might be built into the synthesizer, or they might be part of a default soundbank provided with the synthesizer, etc.

Note that you don't use this method to find out which instruments are currently loaded onto the synthesizer; for that purpose, you use getLoadedInstruments(). Nor does the method indicate all the instruments that can be loaded onto the synthesizer; it only indicates the subset that come with the synthesizer. To learn whether another instrument can be loaded, you can invoke isSoundbankSupported(), and if the instrument's Soundbank is supported, you can try loading the instrument.

**Returns:**list of available instruments.**See Also:**[getLoadedInstruments()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getLoadedInstruments()), [isSoundbankSupported(Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#isSoundbankSupported(javax.sound.midi.Soundbank)), [loadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstrument(javax.sound.midi.Instrument))

### getLoadedInstruments

[Instrument](http://docs.google.com/javax/sound/midi/Instrument.html)[] **getLoadedInstruments**()

Obtains a list of the instruments that are currently loaded onto this Synthesizer.

**Returns:**a list of currently loaded instruments**See Also:**[loadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstrument(javax.sound.midi.Instrument)), [getAvailableInstruments()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getAvailableInstruments()), [Soundbank.getInstruments()](http://docs.google.com/javax/sound/midi/Soundbank.html#getInstruments())

### loadAllInstruments

boolean **loadAllInstruments**([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank)

Loads onto the Synthesizer all instruments contained in the specified Soundbank.

**Parameters:**soundbank - the Soundbank whose are instruments are to be loaded **Returns:**true if the instruments are all successfully loaded (or already had been), false if any instrument could not be loaded (for example, if the Synthesizer had insufficient memory) **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the requested soundbank is incompatible with this synthesizer.**See Also:**[isSoundbankSupported(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#isSoundbankSupported(javax.sound.midi.Soundbank)), [loadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstrument(javax.sound.midi.Instrument)), [loadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D))

### unloadAllInstruments

void **unloadAllInstruments**([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank)

Unloads all instruments contained in the specified Soundbank.

**Parameters:**soundbank - soundbank containing instruments to unload **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - thrown if the soundbank is not supported.**See Also:**[isSoundbankSupported(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#isSoundbankSupported(javax.sound.midi.Soundbank)), [unloadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadInstrument(javax.sound.midi.Instrument)), [unloadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D))

### loadInstruments

boolean **loadInstruments**([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank,  
 [Patch](http://docs.google.com/javax/sound/midi/Patch.html)[] patchList)

Loads the instruments referenced by the specified patches, from the specified Soundbank. Each of the Patch objects indicates a bank and program number; the Instrument that has the matching Patch is loaded into that bank and program location.

**Parameters:**soundbank - the Soundbank containing the instruments to loadpatchList - list of patches for which instruments should be loaded **Returns:**true if the instruments are all successfully loaded (or already had been), false if any instrument could not be loaded (for example, if the Synthesizer had insufficient memory) **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - thrown if the soundbank is not supported.**See Also:**[isSoundbankSupported(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#isSoundbankSupported(javax.sound.midi.Soundbank)), [Instrument.getPatch()](http://docs.google.com/javax/sound/midi/Instrument.html#getPatch()), [loadAllInstruments(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadAllInstruments(javax.sound.midi.Soundbank)), [loadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstrument(javax.sound.midi.Instrument)), [Soundbank.getInstrument(Patch)](http://docs.google.com/javax/sound/midi/Soundbank.html#getInstrument(javax.sound.midi.Patch)), [Sequence.getPatchList()](http://docs.google.com/javax/sound/midi/Sequence.html#getPatchList())

### unloadInstruments

void **unloadInstruments**([Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html) soundbank,  
 [Patch](http://docs.google.com/javax/sound/midi/Patch.html)[] patchList)

Unloads the instruments referenced by the specified patches, from the MIDI sound bank specified.

**Parameters:**soundbank - soundbank containing instruments to unloadpatchList - list of patches for which instruments should be unloaded **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - thrown if the soundbank is not supported.**See Also:**[unloadInstrument(javax.sound.midi.Instrument)](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadInstrument(javax.sound.midi.Instrument)), [unloadAllInstruments(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#unloadAllInstruments(javax.sound.midi.Soundbank)), [isSoundbankSupported(javax.sound.midi.Soundbank)](http://docs.google.com/javax/sound/midi/Synthesizer.html#isSoundbankSupported(javax.sound.midi.Soundbank)), [Instrument.getPatch()](http://docs.google.com/javax/sound/midi/Instrument.html#getPatch()), [loadInstruments(javax.sound.midi.Soundbank, javax.sound.midi.Patch[])](http://docs.google.com/javax/sound/midi/Synthesizer.html#loadInstruments(javax.sound.midi.Soundbank,%20javax.sound.midi.Patch%5B%5D))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Synthesizer.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/midi/SoundbankResource.html)   [**NEXT CLASS**](http://docs.google.com/javax/sound/midi/SysexMessage.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sound/midi/Synthesizer.html)    [**NO FRAMES**](http://docs.google.com/Synthesizer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

[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).