| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/VoiceStatus.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/Transmitter.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/sound/midi/VoiceStatus.html)    [**NO FRAMES**](http://docs.google.com/VoiceStatus.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#1ksv4uv) | METHOD |

## **javax.sound.midi**

Class VoiceStatus

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

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

A VoiceStatus object contains information about the current status of one of the voices produced by a [Synthesizer](http://docs.google.com/javax/sound/midi/Synthesizer.html).

MIDI synthesizers are generally capable of producing some maximum number of simultaneous notes, also referred to as voices. A voice is a stream of successive single notes, and the process of assigning incoming MIDI notes to specific voices is known as voice allocation. However, the voice-allocation algorithm and the contents of each voice are normally internal to a MIDI synthesizer and hidden from outside view. One can, of course, learn from MIDI messages which notes the synthesizer is playing, and one might be able deduce something about the assignment of notes to voices. But MIDI itself does not provide a means to report which notes a synthesizer has assigned to which voice, nor even to report how many voices the synthesizer is capable of synthesizing.

In Java Sound, however, a Synthesizer class can expose the contents of its voices through its [getVoiceStatus()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getVoiceStatus()) method. This behavior is recommended but optional; synthesizers that don't expose their voice allocation simply return a zero-length array. A Synthesizer that does report its voice status should maintain this information at all times for all of its voices, whether they are currently sounding or not. In other words, a given type of Synthesizer always has a fixed number of voices, equal to the maximum number of simultaneous notes it is capable of sounding.

If the voice is not currently processing a MIDI note, it is considered inactive. A voice is inactive when it has been given no note-on commands, or when every note-on command received has been terminated by a corresponding note-off (or by an "all notes off" message). For example, this happens when a synthesizer capable of playing 16 simultaneous notes is told to play a four-note chord; only four voices are active in this case (assuming no earlier notes are still playing). Usually, a voice whose status is reported as active is producing audible sound, but this is not always true; it depends on the details of the instrument (that is, the synthesis algorithm) and how long the note has been going on. For example, a voice may be synthesizing the sound of a single hand-clap. Because this sound dies away so quickly, it may become inaudible before a note-off message is received. In such a situation, the voice is still considered active even though no sound is currently being produced.

Besides its active or inactive status, the VoiceStatus class provides fields that reveal the voice's current MIDI channel, bank and program number, MIDI note number, and MIDI volume. All of these can change during the course of a voice. While the voice is inactive, each of these fields has an unspecified value, so you should check the active field first.

**See Also:**[Synthesizer.getMaxPolyphony()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getMaxPolyphony()), [Synthesizer.getVoiceStatus()](http://docs.google.com/javax/sound/midi/Synthesizer.html#getVoiceStatus())

| **Field Summary** | |
| --- | --- |
| boolean | [**active**](http://docs.google.com/javax/sound/midi/VoiceStatus.html#active)            Indicates whether the voice is currently processing a MIDI note. |
| int | [**bank**](http://docs.google.com/javax/sound/midi/VoiceStatus.html#bank)            The bank number of the instrument that this voice is currently using. |
| int | [**channel**](http://docs.google.com/javax/sound/midi/VoiceStatus.html#channel)            The MIDI channel on which this voice is playing. |
| int | [**note**](http://docs.google.com/javax/sound/midi/VoiceStatus.html#note)            The MIDI note that this voice is playing. |
| int | [**program**](http://docs.google.com/javax/sound/midi/VoiceStatus.html#program)            The program number of the instrument that this voice is currently using. |
| int | [**volume**](http://docs.google.com/javax/sound/midi/VoiceStatus.html#volume)            The current MIDI volume level for the voice. |

| **Constructor Summary** | |
| --- | --- |
| [**VoiceStatus**](http://docs.google.com/javax/sound/midi/VoiceStatus.html#VoiceStatus())() |

| **Method Summary** | |
| --- | --- |

| **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()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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** |
| --- |

### active

public boolean **active**

Indicates whether the voice is currently processing a MIDI note. See the explanation of [active and inactive voices](#3znysh7).

### channel

public int **channel**

The MIDI channel on which this voice is playing. The value is a zero-based channel number if the voice is active, or unspecified if the voice is inactive.

**See Also:**[MidiChannel](http://docs.google.com/javax/sound/midi/MidiChannel.html), [active](http://docs.google.com/javax/sound/midi/VoiceStatus.html#active)

### bank

public int **bank**

The bank number of the instrument that this voice is currently using. This is a number dictated by the MIDI bank-select message; it does not refer to a SoundBank object. The value ranges from 0 to 16383 if the voice is active, and is unspecified if the voice is inactive.

**See Also:**[Patch](http://docs.google.com/javax/sound/midi/Patch.html), [Soundbank](http://docs.google.com/javax/sound/midi/Soundbank.html), [active](http://docs.google.com/javax/sound/midi/VoiceStatus.html#active), [MidiChannel.programChange(int, int)](http://docs.google.com/javax/sound/midi/MidiChannel.html#programChange(int,%20int))

### program

public int **program**

The program number of the instrument that this voice is currently using. The value ranges from 0 to 127 if the voice is active, and is unspecified if the voice is inactive.

**See Also:**[MidiChannel.getProgram()](http://docs.google.com/javax/sound/midi/MidiChannel.html#getProgram()), [Patch](http://docs.google.com/javax/sound/midi/Patch.html), [active](http://docs.google.com/javax/sound/midi/VoiceStatus.html#active)

### note

public int **note**

The MIDI note that this voice is playing. The range for an active voice is from 0 to 127 in semitones, with 60 referring to Middle C. The value is unspecified if the voice is inactive.

**See Also:**[MidiChannel.noteOn(int, int)](http://docs.google.com/javax/sound/midi/MidiChannel.html#noteOn(int,%20int)), [active](http://docs.google.com/javax/sound/midi/VoiceStatus.html#active)

### volume

public int **volume**

The current MIDI volume level for the voice. The value ranges from 0 to 127 if the voice is active, and is unspecified if the voice is inactive.

Note that this value does not necessarily reflect the instantaneous level of the sound produced by this voice; that level is the result of many contributing factors, including the current instrument and the shape of the amplitude envelope it produces.

**See Also:**[active](http://docs.google.com/javax/sound/midi/VoiceStatus.html#active)

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

### VoiceStatus

public **VoiceStatus**()

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/VoiceStatus.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/Transmitter.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/sound/midi/VoiceStatus.html)    [**NO FRAMES**](http://docs.google.com/VoiceStatus.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#1ksv4uv) | METHOD |

[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).