Sibelius

Sibelius 5
Sound Set Editor User Guide

Edition 1.1.0 beta
October 2007

Sound Set Editor written by Peter Marks.

Sound Set Editor User Guide written by Daniel Spreadbury.

Contents

Introduction

Sibelius Sound Set Editor allows you to create your own sound set files for use with Sibelius 5.

What is a sound set?

A sound set is an XML document that describes the capabilities of a particular playback device, whether it is a physical MIDI device, or a virtual instrument.

Technical help

We cannot offer detailed technical help on the creation of sound set files, but you will probably be able to get answers to your questions on the Sibelius chat page in the online Help Center, at www.sibelius.com/helpcenter.

Installation and getting started

Adobe Integrated Runtime (AIR)

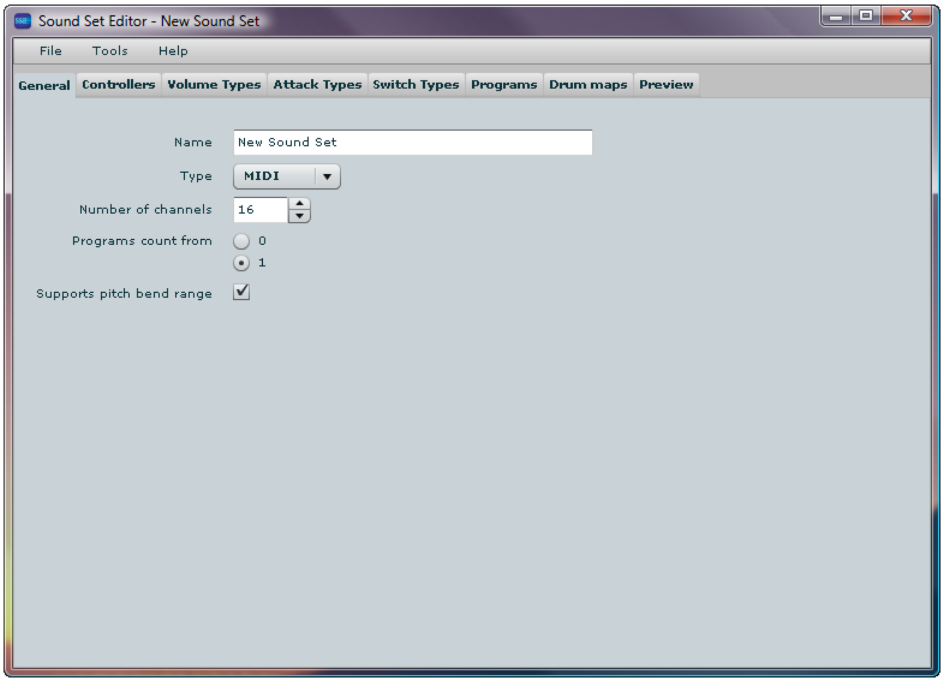
Sibelius Sound Set Editor requires the Adobe Integrated Runtime (AIR) in order to run. You can download the latest version of AIR from http://labs.adobe.com/downloads/air.html.

Installation

- Double-click the downloaded sse.air file to start the installation
- This application is not currently signed, so you will be warned about the publisher of the application not being verified; click Install
- You are prompted to choose an installation location and whether you want a shortcut on your desktop; adjust these settings as you like, then click Continue.
- You are told that installation has completed; click Finish.
- If you chose to run the Sound Set Editor immediately after installing it, it will now launch.

Getting started

When you run the Sound Set Editor, you will see a pretty splash screen, then a window like this:



Name: New Sound Set
Type: MIDI
Number of channels: 16
Programs count from: 0 [✓ 1]
Supports pitch bend range: ✓

You can move between the different pages of the editor by clicking the tabs at the top of the window. Each of the pages is discussed below in detail, but in summary:

- General allows you to set up the name and type of the sound set file you are creating.
- Controllers is for defining the various MIDI controllers that your device supports.
- Volume Types describes the different means of changing volume that the sounds on your device supports; these refer to the MIDI controllers defined on the Controllers page.
- Attack Types allows you to set up different controllers for adjusting the attack of individual notes, as a complement to the options on the Volume Types page. These also refer to the MIDI controllers defined on the Controllers page.
- Switch Types describes the means of obtaining different sounds within a single patch, e.g. using keyswitches or MIDI controllers.
- Programs is where you provide information about each sound on your device. Things you define here refer to data entered on the Volume Types, Attack Types and Switch Types pages.
- Drum Maps is where you list the sound IDs for each sound in a drum patch on your device.

- Preview allows you to see the XML file that will be output when you save your sound set.

You should basically move through the pages from left to right, from General to Drum Maps, defining the different elements of the sound set.

The menus

The Sound Set Editor has very simple menus:

- From the File menu you can start a New sound set, Open an existing sound set, Save the current sound set, Save As under a different name, Revert to the last saved version of the sound set, and Exit the program.
- From the Tools menu you can Validate your existing sound set (we'll talk more about validation below).
- From the Help menu you can see details About Sound Set Editor.

Save your work often!

The Sound Set Editor does not have an auto-save feature, and nor can you use the standard shortcut Ctrl+S *or* ⌘S to save your work, so don't forget to use File ▸ Save or File ▸ Save As often.

Validating your sound set

Before you test your sound set in Sibelius, choose Tools ▸ Validate to validate your sound set. This will check for any duplicate sound IDs, and also any unrecognized sound IDs.

Testing your sound set

In order to test your sound set, you will need to set up a playback configuration in Sibelius 5 to use it.

First, take your new sound set and put it in a location where Sibelius 5 can find it:

- *Windows XP:* C:\Documents and Settings\your username\Application Data\Sibelius Software\Sibelius 5\Sounds
- *Windows Vista:* C:\Users\your username\AppData\Roaming\Sibelius Software\Sibelius 5\Sounds
- *Mac OS X:* /Users/*your username*/Library/Application Support/Sibelius Software/Sibelius 5/Sounds

(You may have to create the Sounds folder yourself.)

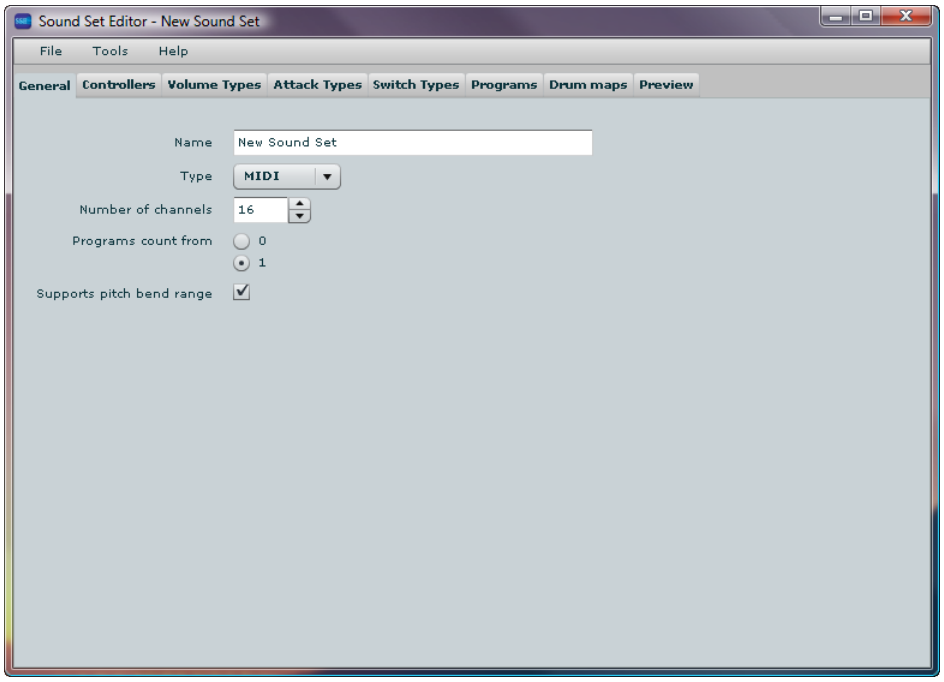
Next, restart Sibelius: your sound set will now appear in the list of sound sets in the Active Devices list in Play ▸ Playback Devices. For help with setting up a playback configuration, 🕮 **4.11 Playback Devices** in Help ▸ Sibelius Reference.

Sharing your sound set

If you have produced a sound set that you would like to share with the Sibelius user community, please send it to Sibelius Product Manager Daniel Spreadbury at dspreadbury@sibelius.com.

General page

The General page has the following options:



Name: New Sound Set
Type: MIDI
Number of channels: 16
Programs count from: ○ 0 [✓ ● 1]
Supports pitch bend range: ✓

Name

This sets the name of the sound set, as it will appear in Play ▸ Playback Devices in Sibelius. This is not the file name of the sound set, which you specify when you use File ▸ Save.

Type

There are three types of sound set:

- MIDI: for devices that respond to program and bank change messages.
- Kontakt: for sample libraries that load into Kontakt Player 2 or the full Kontakt sampler, where you want Sibelius to load the samples automatically for you.
- Fixed: for devices where the sounds provided by each channel are fixed, i.e. it does not respond to program changes.

External MIDI devices, such as the Roland JV-1080, typically use MIDI type sound sets, while virtual instruments, such as Vienna Symphonic Library, typically use Fixed sound sets. Only libraries that can be specifically loaded into Kontakt Player 2 or Kontakt can use Kontakt type sound sets.

Note that it's sometimes appropriate to use a MIDI type sound set for a VST or Audio Unit virtual instrument, and conversely it's sometimes appropriate to use a Fixed type sound set for a MIDI device. For example, Native Instruments Bandstand is a virtual instrument that behaves like a MIDI sound module and responds to program and bank change messages, so it would need a MIDI type sound set (in fact you could simply choose Sibelius's built-in General MIDI sound set); similarly, you may have a MIDI device that uses a fixed setup (with your favorite flute sound on channel 1, your favourite oboe sound on channel 2, etc.) for which it would be most appropriate to use a Fixed type sound set, so that Sibelius doesn't send program change messages.

Finally, it's also sometimes appropriate to use a Fixed type sound set for Kontakt Player 2 or Kontakt 2 libraries. If you want to load sounds yourself, but still have Sibelius able to use keyswitches and controllers automatically, then you need a Fixed type sound set.

Number of channels

This is the number of channels provided by the device. For MIDI devices, this is normally 16 (the device will present multiple inputs if it supports more than 16 channels). Virtual instruments may provide any number of channels up to 16: for example, VSL Vienna Instruments each provide a single channel; the Native Instruments Kompakt Player used by EastWest Quantum Leap Symphonic Orchestra provides 8 channels; each instance of Kontakt Player 2 provides 16 channels (though it can provide 64 channels, only the first 16 are directly addressable via a host application such as Sibelius), and so on.

Programs count from

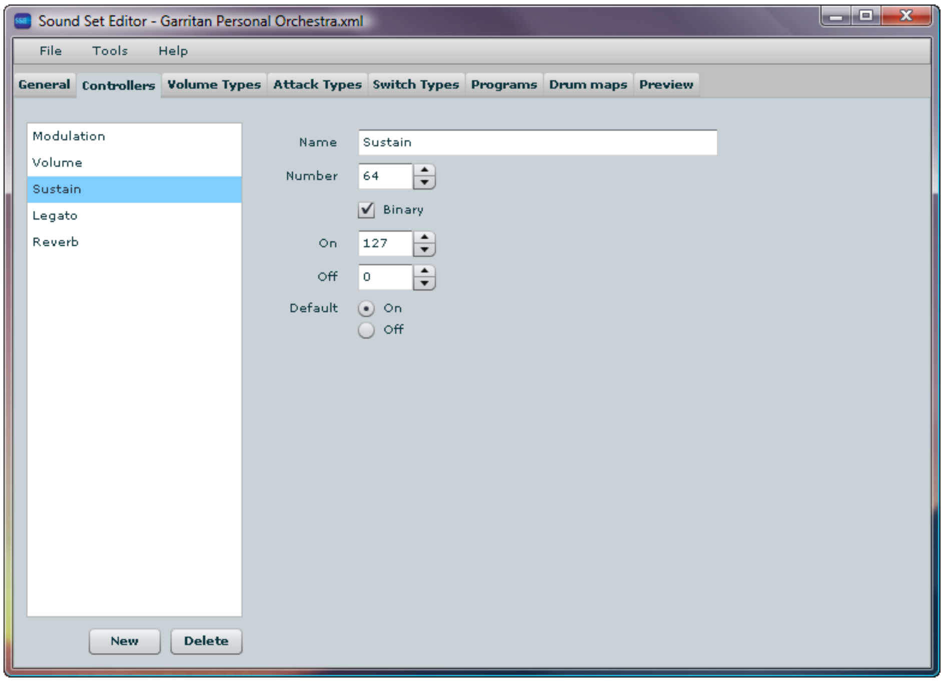
This option only appears for MIDI type sound sets. If your device's printed documentation or on-screen display count program numbers from 0, set Programs count from to 0; otherwise, leave it set to 1.

Supports pitch bend range

This option does not appear for Kontakt type sound sets (because neither Kontakt Player 2 nor Kontakt support the standard MIDI message for setting the range for pitch bend MIDI messages, and so is automatically switched off), but defaults to being switched on for MIDI and Fixed sound sets. If you switch this off, you can then optionally determine the pitch bend range for each individual sound in your sound set on the Programs page (see **Programs page** on page 14).

Controllers page

The Controllers page looks like this:



Name: Sustain
Number: 64
✓ Binary
On: 127
Off: 0
Default: ● On ○ Off

The list box on the left-hand side lists the defined controller types. When you select one of the items in that list, the controls on the right-hand side of the window update to allow you to edit the values associated with that controller.

To add a new controller type, click New. To delete an existing controller type, click Delete.

The controls on the right-hand side of the window are as follows:

Name

This is the name of the controller, as will appear on the other pages of the Sound Set Editor. Use a descriptive name, e.g. you might call controller 68 Legato, or controller 91 Reverb.

Number

This is the actual MIDI controller number. The exact meaning of the MIDI controller will depend on the device or sample library for which you're creating a sound set, but the purpose of each of the standard MIDI controllers can be found here:

http://www.indiana.edu/~emusic/cntrlnumb.html

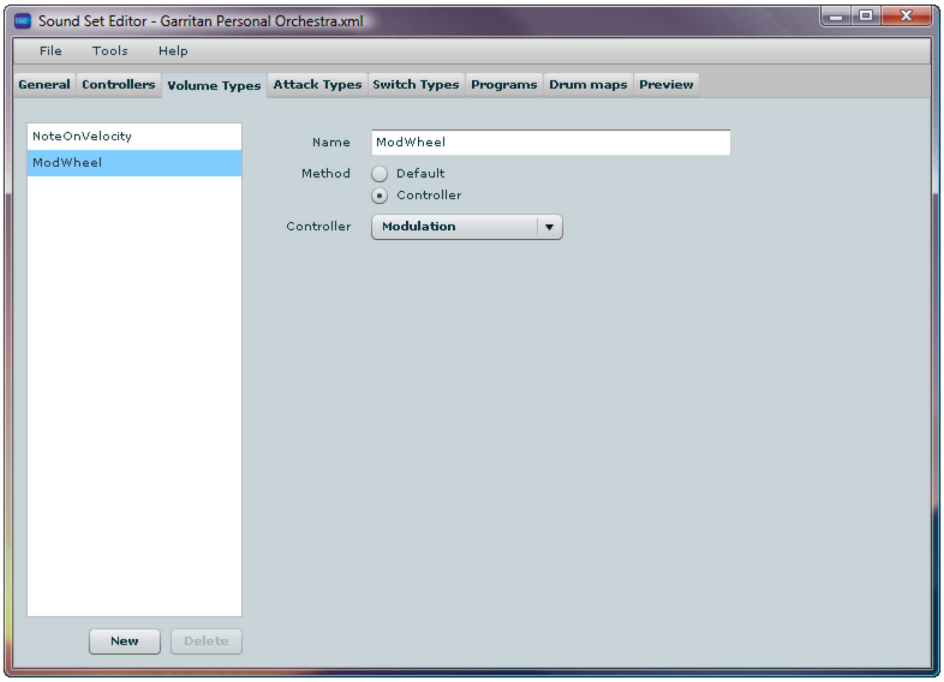
Binary

Binary denotes whether the controller is a simple on-off switch, or whether it can have any value between 0 and 127. For example, controller 64 (sustain pedal) would typically have Binary switched on, because most devices only switch on the sustain effect when this controller is set to 127; while controller 1 (modulation wheel) would typically have Binary switched off, because modulation is normally used as a continuous controller.

When Binary is switched on, you can define the value to be used for the On state (normally 127) and the Off state (normally 0). You can also specify whether the controller should Default On or Off.

Volume Types page

The Volume Types page looks like this:



Name: ModWheel

Method: Default [✓ Controller]

Controller: Modulation

The list box on the left-hand side lists the defined volume types. When you select one of the items in that list, the controls on the right-hand side of the window update to allow you to edit the values associated with that volume type.

To add a new volume type, click New. To delete an existing volume type, click Delete.

The controls on the right-hand side of the window are as follows:

Name

This is the name of the volume type, as will appear on the other pages of the Sound Set Editor.

Method

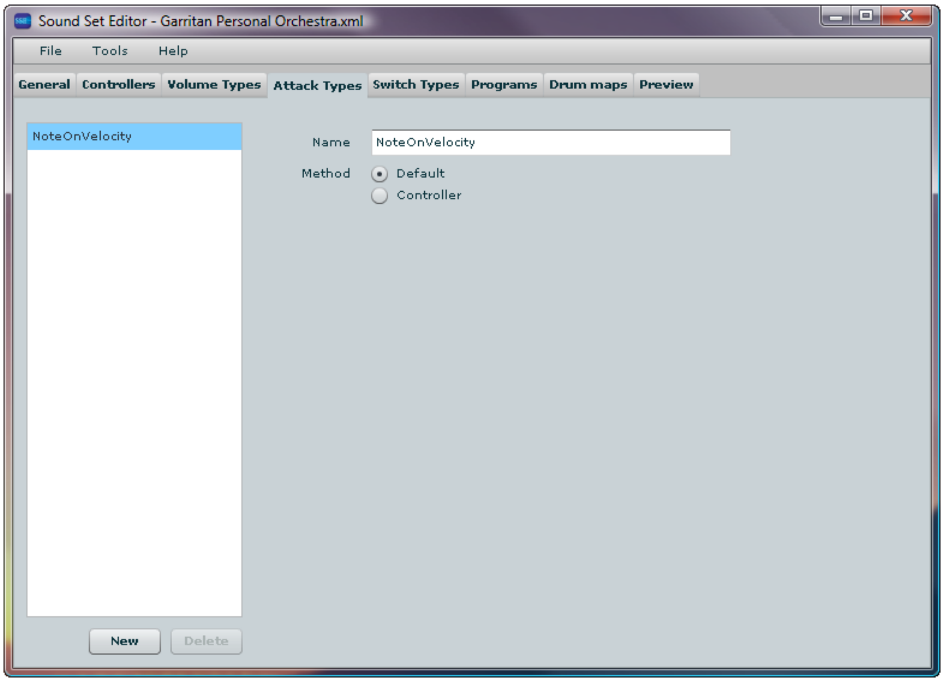
Allows you to choose between the Default volume type (which is note-on velocity) and a specific Controller.

Controller

This only appears if Method is set to Controller. You can choose which of the controller types you have defined on the Controllers page should be used for this volume type. Only continuous controllers (i.e. those with Binary switched off) are suitable for use by volume types.

Attack Types page

The Attack Types page looks like this:



NoteOnVelocity

Name: NoteOnVelocity

Method: [✓ Default] Controller

The list box on the left-hand side lists the defined attack types. When you select one of the items in that list, the controls on the right-hand side of the window update to allow you to edit the values associated with that attack type.

To add a new attack type, click New. To delete an existing attack type, click Delete.

The controls on the right-hand side of the window are as follows:

Name

This is the name of the attack type, as will appear on the other pages of the Sound Set Editor.

Method

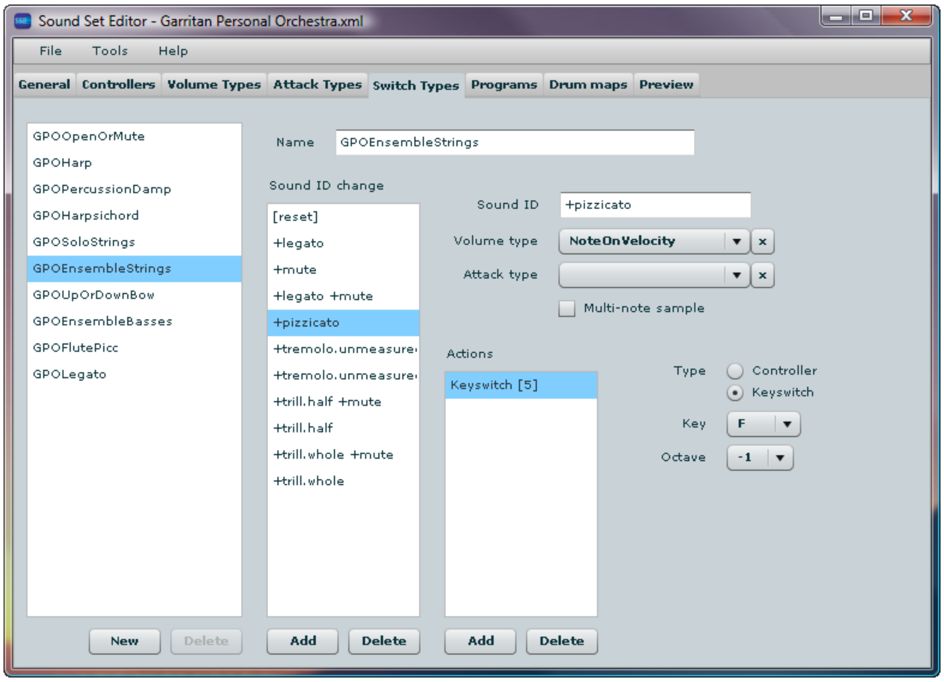
Allows you to choose between the Default attack type (which is note-on velocity) and a specific Controller.

Controller

This only appears if Method is set to Controller. You can choose which of the controller types you have defined on the Controllers page should be used for this attack type.

Switch Types page

The Switch Types page looks like this:



Name: GPOEnsembleStrings

Sound ID: +pizzicato

Volume type: NoteOnVelocity

Attack type:

Multi-note sample: [ ]

Type: Controller [✓ Keyswitch]

Key: F

Octave: -1

The list box on the left-hand side lists the defined switch types. When you select one of the items in that list, the controls on the right-hand side of the window update to allow you to edit the values associated with that switch type.

Each entry in a switch type is a switch, which provides a specific sound ID change, by way of either a keyswitch or a controller change. Each switch can also optionally change the default volume and attack types for the patch.

First define a new switch type by clicking New. Type the switch type's name into the Name control at the top of the dialog. Add a switch by clicking Add at the bottom of the Sound ID change list. Set the switch's Sound ID, and define a new Volume type and/or Attack type if you wish. In the Actions list, click Add to add the mechanism that drives the switch: either a Keyswitch, or a Controller. You can add multiple keyswitches and controllers if required.

In more detail:

Name

This is the name of the switch type, as will appear on the Programs page of the Sound Set Editor.

Sound ID change

Each switch in a switch type provides a sound ID change. Click Add to add a new switch.

Sound ID

This is the relative sound ID change provided by the switch. Relative sound ID changes are one or more element of a sound ID, and always start with + or -, e.g. +pizzicato, +mute, -mute.harmon, or the special reset value, [reset].

You can define multiple relative sound ID changes in the same switch, e.g. +pizzicato +mute, which means that the switch adds *both* relative sound ID changes to the current sound ID. This is normally preferable to specifying +pizzicato.mute because Sibelius will then look for that exact sequence of elements (i.e. pizzicato followed immediately by mute). However, if you want to switch to very specific relative sound IDs, e.g. +mute.harmon, +mute.straight, +trill.half, +trill.whole, then you should specify them as a single, multi-element sound ID change.

Each switch type should include a [reset] switch, which should emit a default keyswitch (if keyswitches are used by other switches in the switch type) and also emit the default values for any controllers used by other switches. So if you set (say) controller 64 to 127 for a +sustain switch, you should set it to 0 in your [reset] switch.

Volume type

If this switch requires a different volume type than that used by the patch or patches for which this switch type is designed, choose one of the volume types defined on the Volume Types page here.

For example, GPO's *arco* violin sounds use modulation wheel for volume, but the *pizzicato* violin sounds use pizzicato, so the +pizzicato switch requires a change of volume type to note-on velocity.

Attack type

If this switch requires a different attack type than that used by the patch or patches for which this switch type is designed, choose one of the attack types defined on the Attack Types page here.

Multi-note sample

If the sample that results from applying the switch is a sampled roll, trill, tremolo or similar sample, switch this on.

Actions

The list of Actions defines the mechanism for the switch. To add a new action, click Add. To delete an existing action, click Delete.

Type

To the right of the Actions list you can choose whether the selected action is a Controller or a Keyswitch.

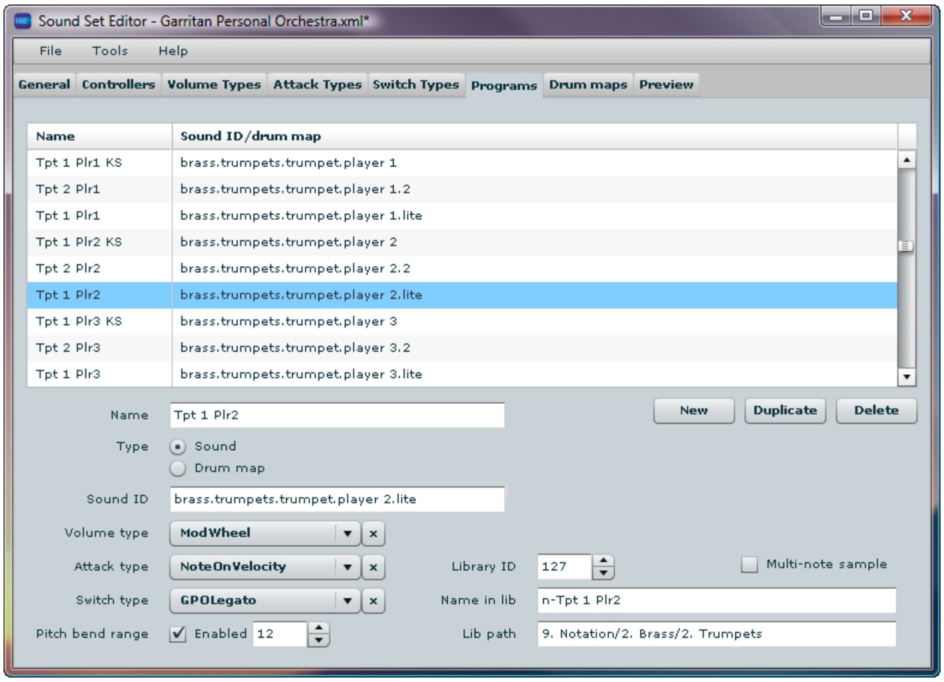
If you choose Controller, you can choose one of the controllers defined on the Controller Types page. If that controller is a Binary controller, you can then choose whether this switch has the controller On or Off.

If you choose Keyswitch, you can then specify the keyswitch pitch to play, by choosing the Key name and the Octave number. Middle C is C4.

Note that you must ensure that any controller or keyswitch that you set for one of your actions must also be specified in their "off" state for the [reset] switch, to ensure that Sibelius is able to reset everything properly.

Programs page

The **Programs** page looks like this:



Name	Sound ID/drum map
Tpt 1 Plr1 KS	brass.trumpets.trumpet.player 1
Tpt 2 Plr1	brass.trumpets.trumpet.player 1.2
Tpt 1 Plr1	brass.trumpets.trumpet.player 1.lite
Tpt 1 Plr2 KS	brass.trumpets.trumpet.player 2
Tpt 2 Plr2	brass.trumpets.trumpet.player 2.2
Tpt 1 Plr2	brass.trumpets.trumpet.player 2.lite
Tpt 1 Plr3 KS	brass.trumpets.trumpet.player 3
Tpt 2 Plr3	brass.trumpets.trumpet.player 3.2
Tpt 1 Plr3	brass.trumpets.trumpet.player 3.lite

Name	Tpt 1 Plr2
Type	[✓ Sound] Drum map
Sound ID	brass.trumpets.trumpet.player 2.lite
Volume type	ModWheel
Attack type	NoteOnVelocity
Switch type	GPOLegato
Pitch bend range	✓ Enabled 12
Library ID	127
Multi-note sample	☐
Name in lib	n-Tpt 1 Plr2
Lib path	9. Notation/2. Brass/2. Trumpets

The box at the top of the page lists all the defined programs in the device. To add a new program, click **New**. To delete a selected program, click **Delete**.

The controls on the page are as follows:

Name

This name is for your own reference; it does not currently appear in Sibelius. Normally you would choose the same name here as appears in the documentation for your device or sound library.

Type

Choose whether the program is a pitched **Sound** or an unpitched **Drum map**. If you choose **Sound**, the control below allows you to type the **Sound ID** that represents the program. If you choose **Drum map**, the control below allows you to choose the drum map defined on the **Drum Maps** page that matches this program.

Sound ID

Type the best possible match for the current program into the **Sound ID** control. The Sound Set Editor will automatically complete the sound ID for you, to help you choose an appropriate sound ID. If the Sound Set Editor does not offer a sound ID that absolutely matches the program's sound, you can create your own. Try to create your own sound IDs by basing them on the closest existing sound ID, e.g. by adding one or more elements to the end. This ensures that Sibelius's SoundWorld system will be able to properly substitute your chosen sound ID for the closest available one when your scores are played back on other devices.

For more information about sound IDs, 📖 **4.14 SoundWorld™** in **Help ▸ Sibelius Reference**.

Drum map

Choose the drum map that matches this program, as defined on the **Drum Maps** page of the Sound Set Editor.

Volume type

This is the default volume type for this sound, as defined on the **Volume Types** page of the Sound Set Editor.

Attack type

This is the default attack type for this sound, as defined on the **Attack Types** page of the Sound Set Editor.

Switch type

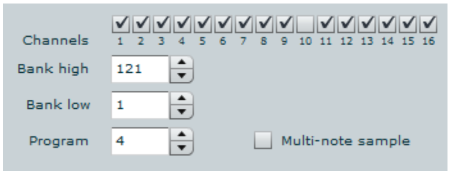
This is the switch type for this program, as defined on the Switch Types page of the Sound Set Editor.

Pitch bend range

If you are editing a Kontakt type sound set, or if Supports pitch bend range is switched off on the General page, you can switch on Enabled in order to specify the fixed pitch bend range of the program you're editing. The pitch bend range is specified in half-steps (semitones).

MIDI type sound sets

If you are creating a MIDI type sound set, the controls on the right-hand side of the window are as follows:

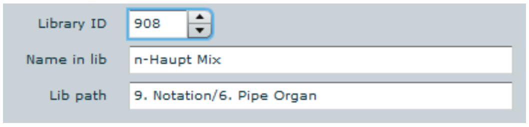


Channels ✓1 ✓2 ✓3 ✓4 ✓5 ✓6 ✓7 ✓8 ✓9 10 ✓11 ✓12 ✓13 ✓14 ✓15 ✓16
Bank high: 121
Bank low: 1
Program: 4
Multi-note sample

- Channels allows you to specify which channels this patch can use. For pitched Sound patches, the Sound Set Editor defaults to all channels other than channel 10; for unpitched Drum map patches, the Sound Set Editor defaults to channel 10 only.
- Bank high is the MSB required for the bank change.
- Bank low is the LSB required for the bank change.
- Program is the program number, counting from 0 or 1 as specified on the General page.
- Multi-note sample should be switched on if this program's sound is a sampled roll, trill, tremolo or similar sample.

Kontakt type sound sets

If you are creating a Kontakt type sound set, the controls on the right-hand side of the window are as follows:



Library ID: 908
Name in lib: n-Haupt Mix
Lib path: 9. Notation/6. Pipe Organ

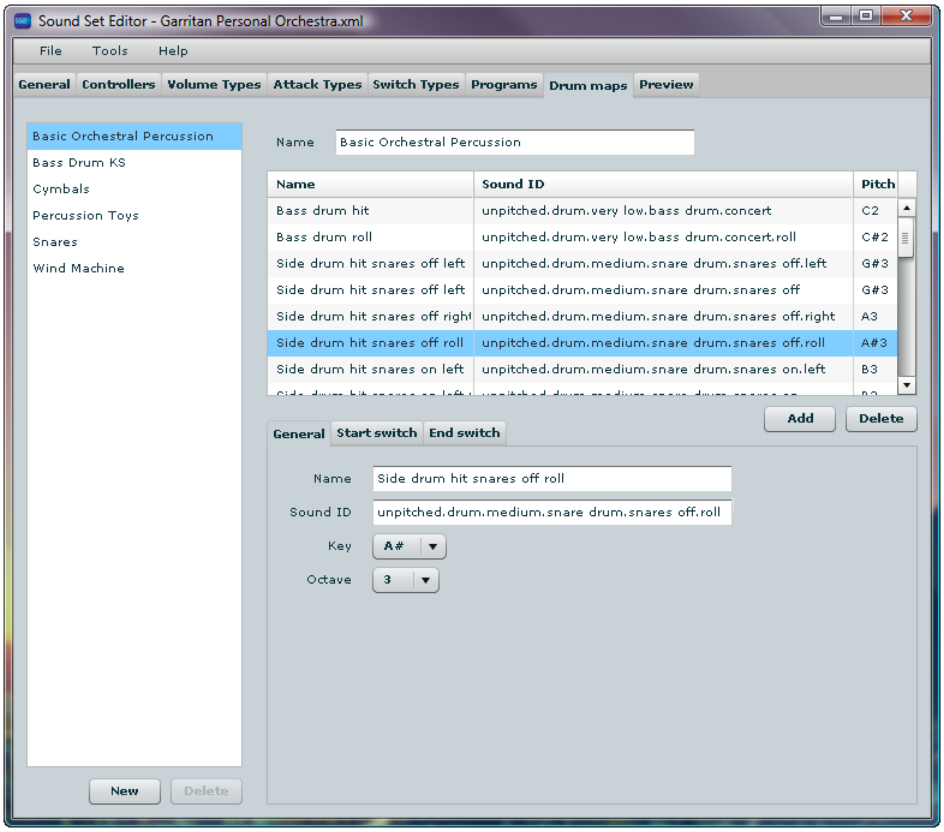
- Library ID is the sample library's unique ID.
- Name in lib is the name of the NKI file to be loaded for this patch, without the .nki file extension.
- Lib path is the path to the NKI file to be loaded for this patch, relative to the root of the library as displayed in the library browser panel of the Kontakt Player 2 window.
- Multi-note sample should be switched on if this program's sound is a sampled roll, trill, tremolo or similar sample.

Fixed type sound sets

If you are creating a Fixed type sound set, the only control on the right-hand side of the window is Multi-note sample, which should be switched on if this program's sound is a sampled roll, trill, tremolo or similar sample.

Drum Maps page

On the **Drum Maps** page you define a drum map, which is a list of each pitch provided by an unpitched drum set patch and its corresponding sound ID. The **Drum Maps** page looks like this:



Name: Basic Orchestral Percussion

Name	Sound ID	Pitch
Bass drum hit	unpitched.drum.very low.bass drum.concert	C2
Bass drum roll	unpitched.drum.very low.bass drum.concert.roll	C#2
Side drum hit snares off left	unpitched.drum.medium.snare drum.snares off.left	G#3
Side drum hit snares off left	unpitched.drum.medium.snare drum.snares off	G#3
Side drum hit snares off righ	unpitched.drum.medium.snare drum.snares off.right	A3
Side drum hit snares off roll	unpitched.drum.medium.snare drum.snares off.roll	A#3
Side drum hit snares on left	unpitched.drum.medium.snare drum.snares on.left	B3

Name: Side drum hit snares off roll
Sound ID: unpitched.drum.medium.snare drum.snares off.roll
Key: A#
Octave: 3

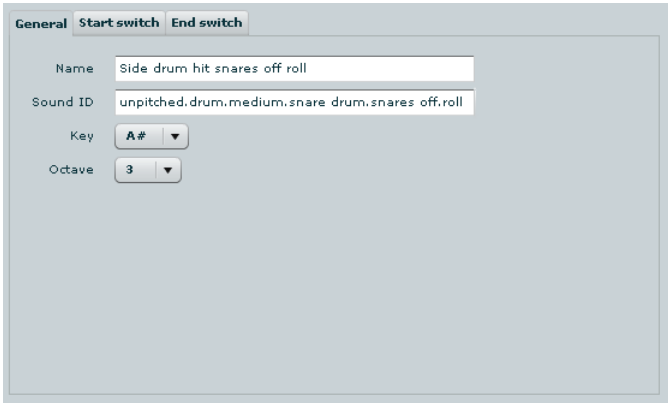
This list box on the left of the page lists all the defined drum maps. To add a new drum map, click **New**. To delete an existing drum map, click **Delete**.

The large box on the right-hand side of the page lists each sound in the drum map. To add a sound to the drum map, click **Add**. To delete an existing sound from the drum map, click **Delete**.

When you select a sound from the list, the controls below appear, split into three groups as follows:

General tab

The options on the **General** tab must be set for every sound in the drum map:



Name: Side drum hit snares off roll
Sound ID: unpitched.drum.medium.snare drum.snares off.roll
Key: A#
Octave: 3

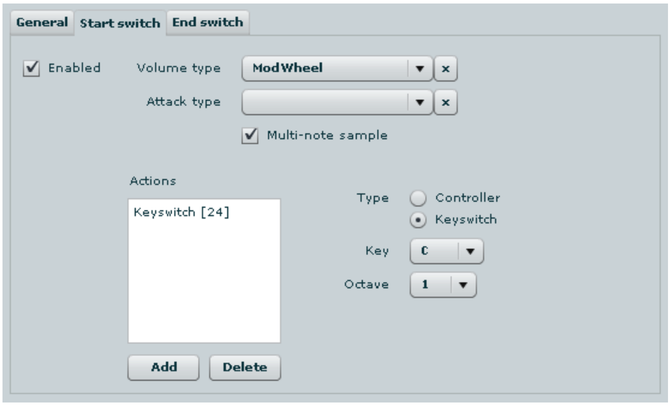
- **Name**: this name is for your own reference; it only appears in Sibelius if you have **Display** set to **Program names** on the **Play-back** page of **Preferences**. Normally you would choose the same name here as appears in the documentation for your device or sound library.

- Sound ID: Type the best possible match for the current program into the Sound ID control. The Sound Set Editor will automatically complete the sound ID for you, to help you choose an appropriate sound ID. If the Sound Set Editor does not offer a sound ID that absolutely matches the program's sound, you can create your own. Try to create your own sound IDs by basing them on the closest existing sound ID, e.g. by adding one or more elements to the end. This ensures that Sibelius's SoundWorld system will be able to properly substitute your chosen sound ID for the closest available one when your scores are played back on other devices.

  For more information about sound IDs, 📖 **4.14 SoundWorld™** in Help ▸ Sibelius Reference.
- Key: This is the pitch name of the note that triggers this sound.
- Octave: This is the number of the octave in which the note is found. Middle C = C4.

Start switch tab

The options on the Start switch tab appear if you switch on the Enabled checkbox. You will need to use these options if the drum sound you are defining is a sample representing multiple notes, or if you need to set a specific keyswitch or MIDI controller in order to obtain the sound. The options are as follows:



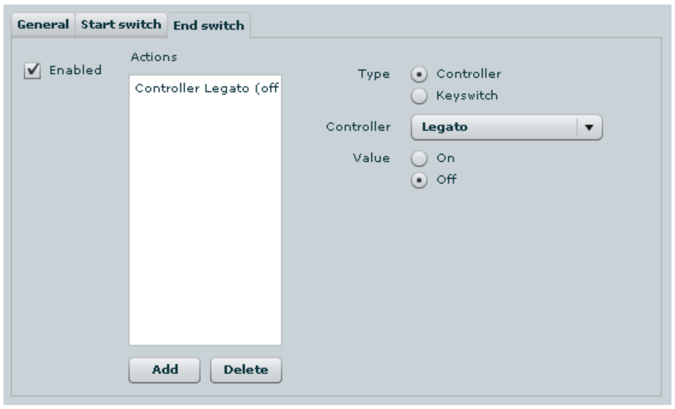
✓ Enabled
Volume type: ModWheel
Attack type:
✓ Multi-note sample
Actions: Keyswitch [24]
Type: Controller [✓ Keyswitch]
Key: C
Octave: 1

- Volume type: If this drum sound requires a different volume type than that used by the other sounds in the drum map, choose one of the volume types defined on the Volume Types page here.

  For example, GPO's snare drum rolls sounds use modulation wheel for volume, whereas most drum hits use note-on velocity for volume.
- Attack type: If this drum sound requires a different attack type than that used by the other sounds in the drum map, choose one of the attack types defined on the Attack Types page here.
- Multiple-note sample: If the sample that results from applying the switch is a sampled roll, trill, tremolo or similar sample, switch this on.
- You can also define Actions to be performed at the start of the note, e.g. keyswitches or controllers. The controls for defining these actions are the same as those on the Switch Types page (see **Switch Types page** on page 12).

End switch tab

The options on the End switch tab appear if you switch on the Enabled checkbox. You will need to use these options if the drum sound you are defining requires a specific MIDI controller or keyswitch performed at the *end* of the note (e.g. to reset a keyswitch or controller used in the Start switch settings for the drum sound).



[✓] Enabled

Actions: Controller Legato (off

Type: [● Controller] Keyswitch

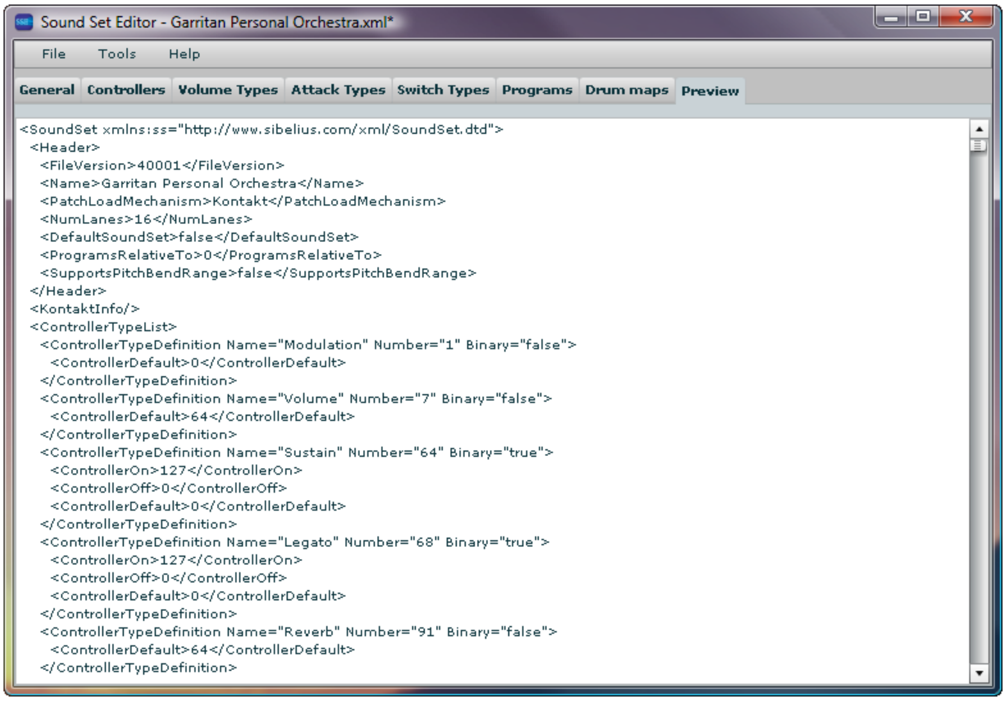
Controller: Legato

Value: On [● Off]

To define Actions to be performed at the end of the note, e.g. keyswitches or controllers, set the controls appropriately; the controls for defining these actions are the same as those on the Switch Types page (see **Switch Types page** on page 12).

Preview page

The **Preview** page looks like this:


Sound Set Editor - Garritan Personal Orchestra.xml*
File Tools Help
General Controllers Volume Types Attack Types Switch Types Programs Drum maps Preview

```
<SoundSet xmlns:ss="http://www.sibelius.com/xml/SoundSet.dtd">
 <Header>
  <FileVersion>40001</FileVersion>
  <Name>Garritan Personal Orchestra</Name>
  <PatchLoadMechanism>Kontakt</PatchLoadMechanism>
  <NumLanes>16</NumLanes>
  <DefaultSoundSet>false</DefaultSoundSet>
  <ProgramsRelativeTo>0</ProgramsRelativeTo>
  <SupportsPitchBendRange>false</SupportsPitchBendRange>
 </Header>
 <KontaktInfo/>
 <ControllerTypeList>
  <ControllerTypeDefinition Name="Modulation" Number="1" Binary="false">
   <ControllerDefault>0</ControllerDefault>
  </ControllerTypeDefinition>
  <ControllerTypeDefinition Name="Volume" Number="7" Binary="false">
   <ControllerDefault>64</ControllerDefault>
  </ControllerTypeDefinition>
  <ControllerTypeDefinition Name="Sustain" Number="64" Binary="true">
   <ControllerOn>127</ControllerOn>
   <ControllerOff>0</ControllerOff>
   <ControllerDefault>0</ControllerDefault>
  </ControllerTypeDefinition>
  <ControllerTypeDefinition Name="Legato" Number="68" Binary="true">
   <ControllerOn>127</ControllerOn>
   <ControllerOff>0</ControllerOff>
   <ControllerDefault>0</ControllerDefault>
  </ControllerTypeDefinition>
  <ControllerTypeDefinition Name="Reverb" Number="91" Binary="false">
   <ControllerDefault>64</ControllerDefault>
  </ControllerTypeDefinition>
```

On this page you can see a live preview of the XML document that will be saved when you choose **File ▸ Save**. You cannot edit the XML directly in this window.

Sound set reference

Sound set syntax

Sound sets in Sibelius 5 are now XML-based documents, consisting of six distinct sections:

- Header and root element
- Definition of controller types (`ControllerTypeList`)
- Definition of volume types (`VolumeTypeList`)
- Definition of attack types (`AttackTypeList`)
- Definition of switch types (`SwitchTypeList`)
- Definition of available patches (`PatchList`)
- Definition of available drum sounds (`DrumMapList`)

Root element

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SoundSet SYSTEM "SoundSet.dtd"[]>
<SoundSet xmlns:ss="http://www.sibelius.com/xml/SoundSet.dtd">
```

The first two lines of the file are standard XML stuff, defining the XML version, text file encoding, and the document type (in this case, `SoundSet`) and the document type definition (DTD) for this document type.

The `SoundSet` element is the "root element" of the XML document. Its attributes are as follows:

- `xmlns`: the namespace of the XML document; this is only required for web services, but its value should be set to `"http://www.sibelius.com/SoundSet.dtd"`

Header element

```
<Header>
   <FileVersion>40001</FileVersion>
   <Name>GPO</Name>
   <PatchLoadMechanism>Kontakt</PatchLoadMechanism>
   <NumLanes>16</NumLanes>
   <DefaultSoundSet>false</DefaultSoundSet>
   <ProgramsRelativeTo>0</ProgramsRelativeTo>
   <SupportsPitchBendRange>false</SupportsPitchBendRange>
</Header>
```

The `Header` element defines basic information about the sound set:

- `FileVersion`: the version of the sound set syntax itself, rather than this specific sound set; at the moment, this is `40001`.
- `Name`: the name that appears in the program itself, e.g. in the Play ▸ Playback Devices dialog.
- `PatchLoadMechnism`: tells Sibelius how to load the patches; there are three valid values: MIDI, Kontakt and Fixed.
- `NumLanes`: this is the channel mask for the device; for most MIDI device it's `16`, but it can vary (usually if `PatchLoadMechnism` is set to `Fixed` or `Kontakt`).
- `DefaultSoundSet`: if true, tells Sibelius that this should be used as the default sound set; this should only be set to true for the General MIDI sound set.
- `ProgramsRelativeTo`: determines whether the scheme for numbering patches (in the `PatchList` element) is 0- or 1-based; valid values are `0` and `1`. This element is optional, if missing defaults to 0.
- `SupportsPitchBendRange`: defaults to true; specify it as false if the device does not support the standard method of setting pitch bend range.

ControllerTypeList element

```
<ControllerTypeList>
    <ControllerTypeDefinition Name="Sustain" Number="64" Binary="true">
        <ControllerOn>127</ControllerOn>
        <ControllerOff>0</ControllerOff>
        <ControllerDefault>Off</ControllerDefault>
    </ ControllerTypeDefinition>
    <ControllerTypeDefinition Name="Reverb" Number="91" Binary="false">
        <ControllerDefault>64</ControllerDefault>
    </ControllerTypeDefinition>
    <ControllerTypeDefinition Name="Volume" Number="7" Binary="false">
        <ControllerDefault>64</ControllerDefault>
    </ControllerTypeDefinition>
    <ControllerTypeDefinition Name="Modulation" Number="68" Binary="false">
        <ControllerDefault>64</ControllerDefault>
    </ControllerTypeDefinition>
    <!-- etc... -->
</ControllerTypeList>
```

`ControllerTypeList` defines one or more **`ControllerTypes`**, which can then be referred to by name by **`VolumeType`** and **`SwitchType`** elements.

If there are no controller types defined for the soundset the **`ControllerTypeList`** may be omitted.

The **`ControllerTypeDefinition`** element has the following attributes:

- **`Name`**: the name of the **`ControllerTypeDefinition`**, and is what is specified when referring to this **`ControllerTypeDefinition`** elsewhere in the sound set via a **`ControllerType`** element.
- **`Number`**: the MIDI controller number.
- **`Binary`**: tells Sibelius whether this controller behaves like a binary switch (i.e. on or off), or whether it can express a range of values; valid values are **`true`** and **`false`**. This attribute is optional, value defaults to **`false`**.

Each **`ControllerTypeDefinition`** element may also have one or more of the following nested elements:

- **`ControllerOn`**: if **`Binary`** is **`true`**, this defines the MIDI controller value that sets the controller "on."
- **`ControllerOff`**: if **`Binary`** is **`false`**, this defines the MIDI controller value that sets the controller "off."
- **`ControllerDefault`**: the default value for this controller, if no other value is specified.

VolumeTypeList element

```
<VolumeTypeList>
    <VolumeTypeDefinition Name="ModWheel" Method="Controller">
        <ControllerType>Modulation</ControllerType>
    </VolumeTypeDefinition>
    <VolumeTypeDefinition Name="NoteOnVelocity" Method="Default" />
    <!-- etc... -->
</VolumeTypeList>
```

`VolumeTypeList` defines one or more **`VolumeTypeDefinitions`**, which determine the means of controlling volume for a **`Patch`** or a **`SwitchType`**. Each **`VolumeTypeDefinition`** refers to a **`ControllerType`** (unless it specifies using the Sibelius default mechanism).

If there are no volume types defined for the soundset (i.e. if all instruments use the Sibelius default mechanism of note-on velocity) the **`VolumeTypeList`** may be omitted.

The **`VolumeTypeDefinition`** element has two attributes:

- **`Name`**: defines the name that is used when referring to this **`VolumeTypeDefinition`** elsewhere in the sound set.
- **`Method`**: defines the method by which the **`VolumeTypeDefinition`** works; valid values are **`Controller`**, which means that volume is achieved by a specific MIDI controller (and requiring a **`ControllerType`** element), and **`Default`**, which means that Sibelius should use its default means of determining volume changes on a staff, namely MIDI note-on velocity. This attribute is optional, and the default value is **`Default`**.

If the **Method** attribute for **VolumeTypeDefinition** is set to **Controller**, the **VolumeTypeDefinition** element requires a nested element, **ControllerType**, the value of which is the name of a **ControllerTypeDefinition** defined elsewhere in the sound set.

AttackTypeList element

```
<AttackTypeList>
    <AttackTypeDefinition Name="Velocity" Method="Default" />
    <AttackTypeDefinition Name="Controller102" Method="Controller" />
       <ControllerType>Controller102</ControllerType>
    </AttackTypeDefinition>
    <!-- etc... -->
</AttackTypeList>
```

AttackTypeList defines one or more **AttackTypeDefinitionss**, which determine the means of controlling attack for a **Patch** or a **SwitchType**. Each **AttackTypeDefinition** normally refers to a **ControllerType**.

If there are no attack types defined for the sound set, the **AttackTypeList** may be omitted.

The **AttackTypeDefinition** element has two attributes:

- **Name**: defines the name that is used when referring to this **AttackType** elsewhere in the sound set.
- **Method**: defines the method by which the **AttackType** works; valid values are **Controller**, which means that volume is achieved by a specific MIDI controller (and requiring a **ControllerType** element), and **Default**, which means that Sibelius should use it's default behaviour for controlling attack. This attribute is optional, default value is **Default**.

If the **Method** attribute for **AttackTypeDefinition** is set to **Controller**, the **AttackTypeDefinition** element requires a nested element, **ControllerType**, the value of which is the name of a **ControllerTypeDefinition** defined elsewhere in the sound set.

SwitchTypeList element

```
<SwitchTypeList>
   <SwitchTypeDefinition Name="GPOEnsembleStrings">
        <Switch SoundIDChange="[reset]">
            <KeySwitch Key="C0"/>
            <VolumeType>ModWheel</VolumeType>
        <Switch SoundIDChange="+.pizzicato">
            <KeySwitch Key="F#0"/>
            <VolumeType>NoteOnVelocity</VolumeType>
        </Switch>
        <Switch SoundIDChange="+.legato" >
            <ControllerSwitch Name="Sustain" Value="on" />
        </Switch>
        <!-- etc. -->
    </SwitchTypeDefinition>
    <!-- etc. -->
</SwitchTypeList>
```

SwitchTypeList defines one or more **SwitchTypeDefinitions**, which provide a list of the sound IDs that are available for a given **Patch**, both by means of switching within a program, e.g. via a combination of a keyswitch, MIDI controller, and so on.

If there are no switch types defined for the sound set, the **SwitchTypeList** may be omitted.

The **SwitchTypeDefinition** element has a single attribute, **Name**, which defines the name that is used when referring to this **SwitchTypeDefinition** elsewhere in the sound set, e.g. in multiple **Patch** elements. The **SwitchTypeDefinition** element contains one or more nested **Switch** elements.

Switch defines the available sound IDs, and has the following attributes:

- **SoundIDChange**, representing the sound ID change (in the form e.g. **+pizzicato** or **-mute**, or the special value **[reset]**, which removes all relative sound ID changes) achieved by activating this switch. Mandatory.

- **`IsMultipleNoteSample`**, value is **`true`** if the sample that results from applying the switch is a sampled roll, trill, tremolo or similar sample (this will prevent Sibelius from playing back using its own implementation of these). This attribute is optional, and defaults to **`false`**.

Each **`Switch`** element may have following nested elements:

- **`KeySwitch`**: the MIDI note that needs to be emitted to activate the sound necessary to produce the **`SoundIDChange`**. The note is given in the **`Key`** attribute of the element. The note can be specified either as *name+octave* (e.g. **`C#0`**, **`F2`**, **`F#3`**), using the Sibelius convention of middle C = C4, or using a MIDI pitch number (e.g. **`60`**, which is middle C).
- **`ControllerSwitch`**: referring to one of the **`ControllerTypeDefinitions`** defined in **`ControllerTypeList`** elsewhere in the sound set. This element has two attributes:
  - **`Name`**: the name of the **`ControllerTypeDefinition`**;
  - **`Value`**: the value to be passed to the controller. Valid values are **`on`**/**`off`** (for **`ControllerTypeDefinitions`** where **`Binary`** is **`true`**), or integers in the range 0–127.
- **`VolumeType`**: the name of one of the **`VolumeTypeDefinitions`** defined in **`VolumeTypeList`** elsewhere in the sound set. This is so that a particular sound ID change can override the standard **`VolumeTypeDefinition`** for the **`Patch`** in use. This element is optional. One common example of this is for the *pizzicato* samples in GPO's keyswitch string patches. All of the *arco* sounds are sustaining and therefore use modulation for volume, but the *pizzicato* sounds have a fast decay, and so need to use regular MIDI note on velocity to determine volume. So the **`Patch`** entry for (say) Vlns 1 KS sets **`VolumeType`** to **`ModWheel`**, and the **`Switch`** entry for a Sound ID change of **`+pizzicato`** sets **`VolumeType`** to **`NoteOnVelocity`**.
- **`AttackType`**: the name of one of the **`AttackTypeDefinitions`** defined in **`AttackTypeList`** elsewhere in the sound set. This is so that a particular sound ID change can override the standard **`AttackTypeDefinition`** for the **`Patch`** in use.

At least one of **`KeySwitch`** or **`ControllerSwitch`** must be present: multiple instances of each are possible, though this makes more sense for controllers than keyswitches. **`VolumeType`** and **`AttackType`** are optional.

PatchList element

```
<PatchList>
   <Patch Name="Vlns 1 KS">
      <KontaktPatch Instrument="Vlns 1 KS" LibraryID="111" Path="GPO KP2" />
      <SoundID>strings.violin.ensemble</SoundID>
      <VolumeType>ModWheel</VolumeType>
      <SwitchType>GPOEnsembleStrings</SwitchType>
   </Patch>
   <Patch Name="Vln 1 Solo KS">
      <KontaktPatch Instrument="Vln 1 Solo KS" LibraryID="111" Path="GPO KP2" />
      <SoundID>strings.violin.ensemble</SoundID>
      <VolumeType>ModWheel</VolumeType>
      <AttackType>Velocity</AttackType>
      <SwitchType>GPOEnsembleStrings</SwitchType>
   </Patch>
   <Patch Name="Drums and percussion">
      <DrumMap>General MIDI</DrumMap>
      <VolumeType>NoteOnVelocity</VolumeType>
      <MIDIPatch Program="124" ChannelMask="0xFDFF" />
   </Patch>
   <!-- etc... -->
</PatchList>
```

`PatchList` defines a list of the playback device's available **`Patch`**es.

The **`Patch`** element has the following attributes:

- **`Name`**, which is the name of the patch (e.g. as it will appear in Sibelius's user interface). Mandatory.
- **`IsMultipleNoteSample`**, value is **`true`** if the sample that results from applying the switch is a sampled roll, trill, tremolo or similar sample (this will prevent Sibelius from playing back using its own implementation of these). This attribute is optional, and defaults to **`false`**.

Each **`Patch`** element may also have one or more of the following nested elements:

- **`KontaktPatch`**: this element provides Sibelius with the information it needs to load the specified patch, if **`PatchLoad-Mechanism`** is set to **`Kontakt`**. It has three attributes:
  - **`Instrument`**: the name of the Kontakt Player instrument that needs to be loaded.
  - **`LibraryID`**: the numeric ID of the Kontakt Player library from which the Kontakt Player instrument is taken.
  - **`Path`**: the path corresponding to the LibraryID specified, i.e. the folder name in which the .nki et al files may be found.
- MIDIPatch: this element provides Sibelius with the information it needs to load the specified patch, if PatchLoadMechanism is set to MIDI. It has four attributes:
  - **`Program`**: the program number of the specified patch;
  - **`ChannelMask`**: a hexadecimal representation of a bitfield corresponding to which of the 16 available channels can play this patch; common values are **`0xFDFF`** (all channels except channel 10, for pitched instruments in General MIDI) and **`0x0200`** (channel 10 only, for unpitched instruments in General MIDI).
  - **`BankHigh`**, **`BankLow`**: decimal integers in the range 0-127 representing the bank switch numbers. Optional.
- **`SoundID`**: the single, complete sound ID that best describes the base sound of this patch.
- **`DrumMap`**: the name of the **`DrumMap`** to use for this patch, as defined in **`DrumMapList`** elsewhere in the sound set. If **`Drum-Map`** is set for a **`Patch`**, **`SoundID`** cannot be set; these two elements are mutually exclusive.
- **`VolumeType`**: the name of the **`VolumeType`** to use for this patch, as defined in **`VolumeTypeList`** elsewhere in the sound set. This element is optional; if not specified, the Sibelius default mechanism will be used.
- **`AttackType`**: the name of the **`AttackType`** to use for this patch, as defined in **`AttackTypeList`** elsewhere in the sound set. This element is optional; if not specified, the Sibelius default mechanism will be used.
- **`SwitchType`**: the name of the **`SwitchType`** to use for this patch, as defined in **`SwitchTypeList`** elsewhere in the sound set. This element is optional; if not given, then no switches are available for this sound.

DrumMapList element

```
<DrumMapList>
   <DrumMap Name="General MIDI">
      <DrumSound Pitch="27" SoundID="unpitched.exotic.high q" Name="High Q" />
      <DrumSound Pitch="28" SoundID="unpitched.exotic.slap" Name="Slap" />
      <DrumSound Pitch="29" SoundID="unpitched.exotic.scratch" Name="Scratch" />
      <!-- etc... -->
   </DrumMap>
   <DrumMap Name="Marching Snares (Manual)">
      <DrumSound Pitch="36" SoundID="unpitched.metal.cymbal.crash" Name="Crash" />
      <DrumSound Pitch="38" SoundID="unpitched.metal.cymbal.ride" Name="Ride" />
      <!-- etc... -->
   </DrumMapList>
```

`DrumMapList` defines one or more **`DrumMap`**s, which provide a mapping between MIDI note and sound ID for unpitched percussion patches.

If there are no drum maps defined for the soundset, the **`DrumMapList`** may be omitted.

The **`DrumMap`** element has a single attribute, **`Name`**, which defines the name that is used when referring to this **`DrumMap`** elsewhere in the sound set, e.g. in multiple **`Patch`** elements.

Each **`DrumMap`** element contains one or more nested **`DrumSound`** elements. The **`DrumSound`** element has four attributes, as follows:

- **`Pitch`**: the MIDI note number or Sibelius pitch name (where middle C is C4) of the specific sound. This attribute is mandatory;
- **`SoundID`**: the complete sound ID that best describes this sound. Mandatory;
- **`Name`**: a more "human-readable" name (e.g. from the device manufacturer's documentation), which can be used in Sibelius's

The **`DrumSound`** element may optionally contain a **`StartSwitch`** and/or an **`EndSwitch`** element to define switches sent to the virtual instrument to trigger and/or cancel the correct sound for the sound ID, and (in the case of the **`StartSwitch`**) to hold other information describing the sound.

The **`StartSwitch`** element is identical to the **`Switch`** element, with the exception of not having a **`SoundIDChange`** attribute:

- **`StartSwitch`** has a single attribute **`IsMultipleNoteSample`**, value is **`true`** if the sample that results from applying the switch is a sampled roll, trill, tremolo or similar sample (this will prevent Sibelius from playing back using its own implementation of these). This attribute is optional, and defaults to **`false`**.
- Otherwise, the **`StartSwitch`** element contains all the nested elements of **`Switch`**, namely **`KeySwitch`**, **`ControllerSwitch`**, **`VolumeType`**, and **`AttackType`**.

The **`EndSwitch`** element contains only **`KeySwitch`** and **`ControllerSwitch`** elements.

If a **`DrumSound`** has no associated switches but needs non-default settings for any of multi-note sample, volume and attack type, these are defined in a **`StartSwitch`** element (which consequently needn't contain any switches!)

Close root element

```
</SoundSet>
```

The last line in a valid XML file closes the root element.

License Agreement

By installing or using any component of the Software, or by registering the Product, you (an individual or legal entity) agree with the Licensor to be bound by the terms of this License which will govern your use of the Product.

The Product is copyright © 2007 Sibelius Software, a division of Avid Technology, Inc., and its licensors.

1. DEFINITIONS

In this License the following words and expressions have the following meanings:

"Documentation": the Sound Set Editor User Guide and any other documentation relating to the Software supplied to you in any form by the Licensor or with the Software.

"License": this agreement between you and the Licensor and, if permitted by the context, the conditional license granted to you in this agreement.

"Licensor": Avid Technology, Inc., of Avid Technology Park, One Park West, Tewksbury, MA 01876 USA, through its division Sibelius Software, of 20-23 City North, Fonthill Road, London N4 3HF, UK

"Product": the Software and the Documentation.

"Single Copy": a Product provided for use on a single computer terminal.

"Software": Sibelius Sound Set Editor.

2. License

2.1 (1) The Licensor grants to you a non-exclusive non-transferable license to use the Software in accordance with the Documentation.

(2) You may install and use the Software on any number of computers.

(3) Title to the Product is not transferred to you. Ownership of the Product remains vested in the Licensor and its licensors, subject to the rights granted to you under this License. All other rights are reserved.

2.2 You may make one printout for your own use of any part of the Documentation provided in electronic form. You shall not make or permit any third party to make any further copies of any part of the Product whether in eye or machine-readable form.

2.3 You shall not, and shall not cause or permit any third party to, translate, enhance, modify, alter, adapt or create derivative works based on the Product or any part of it for any purpose (including without limitation for the purpose of error correction), or cause the whole or any part of the Product to be combined with or incorporated into any other program, file or product for any purpose, except as expressly permitted by the Documentation.

2.4 You shall not, and shall not cause or permit any third party to, decompile, decode, disassemble or reverse engineer the Software in whole or in part for any purpose.

3. Copyright

3.1 You acknowledge that copyright in the Product as a whole and in the components of the Product as between you and the Licensor belongs to the Licensor or its licensors and is protected by copyright laws, national and international, and all other applicable laws. Further details of the ownership of all copyright in the components of the Product are set out in the Product.

4. Liability of the Licensor

4.1 The Licensor shall not be liable for any claim arising from:

(1) any failure or malfunction resulting wholly or to any material extent from your negligence, operator error, use other than in accordance with the Documentation or any other misuse or abuse of the Product;

(2) any loss of or corruption to any data, however caused, where such loss or corruption could have been avoided or corrected or substantially reduced if you had taken and retained in a secure place appropriate backup copies;

(3) the decompilation or modification of the Software or its merger with any other program or any maintenance repair adjustment alteration or enhancement of the Software by any person other than the Licensor or its authorized agent;

(4) the failure by you to implement recommendations previously advised by the Licensor in respect of, or solutions for faults in, the Product;

(5) any loss or damage whatsoever resulting from any omissions or inaccuracies in any information or data contained in the Product.

(6) Except as otherwise expressly provided in Section 4.1, all conditions, warranties, terms representations and undertakings express or implied, statutory or otherwise in respect of the Product are hereby expressly excluded.

(7) Except as expressly provided in Section 4.1, the Licensor shall have no liability to you for loss of profits, revenue or goodwill or any type of special, indirect or consequential loss (including loss or damages suffered by you as a result of an action brought by a third party) whether such loss is caused by the Licensor's breach of its contractual obligations hereunder or any negligence or other tortious act or omission.

(8) The Licensor's entire liability for breach of its covenants and warranties in this License and for any defect or errors in the Product shall (except as expressly provided in Section 4.1) be limited to the price paid by you for the Product.

5. Termination

5.1 This License shall terminate automatically upon your destruction of the Product. In addition, the Licensor may elect to terminate this License in the event of a material breach by you of any condition of this License or of any of your representations, warranties, covenants or obligations hereunder. Upon notification of such termination by the Licensor, you will immediately delete all copies of the Software from your computer(s), destroy any other copies of the Product or any part thereof, and return the Product to the Licensor.

6. Miscellaneous

6.1 No failure to exercise and no delay in exercising on the part of the Licensor of any right, power or privilege arising hereunder shall operate as a waiver thereof, nor shall any single or partial exercise of any right, power or privilege preclude any other or further exercise thereof or the exercise of any other right, power or privilege. The rights and remedies of the Licensor in connection herewith are not exclusive of any rights or remedies provided by law.

6.2 You may not distribute, loan, sub-license, rent, lease (including without limitation renting or leasing a computer on which the Product is installed) or otherwise transfer the Product to any third party without the Licensor's prior written consent, which the Licensor may grant, condition or withhold in the Licensor's sole discretion.

6.3 You agree to provide accurate personal data when registering the Product and to the use of this data in accordance with the Licensor's privacy policy (available from the Licensor or on www.sibelius.com) which may change from time to time.

6.4 This License is intended by the parties hereto to be a final expression of their agreement with respect to the subject matter hereof and a complete and exclusive statement of the terms of such agreement. This License supersedes any and all prior understandings, whether written or oral, between you and the Licensor relating to the subject matter hereof.

6.5 (This section only applies if you are resident in the European Union:) This License shall be construed and governed by the laws of England, and both parties agree to submit to the exclusive jurisdiction of the English courts.

6.6 (This section only applies if you are not resident in the European Union:) This License shall be construed and enforced in accordance with and governed by the laws of the State of California. Any suit, action or proceeding arising out of or in any way related or connected to this License shall be brought and maintained only in the United States District Court for the Northern District of California, sitting in the City of San Francisco. Each party irrevocably submits to the jurisdiction of such federal court over any such suit, action or proceeding. Each party knowingly, voluntarily and irrevocably waives trial by jury in any suit, action or proceeding (including any counterclaim), whether at law or in equity, arising out of or in any way related or connected to this License or the subject matter hereof.

(Sibelius Sound Set Editor: License v1.0 Beta, 2 October 2007)