

# **Synful Orchestra Users Guide**



# Contents

|  |            |
|--|------------|
| <b>Preface: Welcome.....</b>   | <b>vii</b> |
| <b>Chapter 1: Getting Started.....</b>   | <b>9</b>   |
| Before you Install.....  | 10         |
| Mac OSX System Requirements.....   | 10         |
| Windows System Requirements.....   | 10         |
| Free Crossgrade for Windows and Mac.....   | 10         |
| Wagner, Beethoven, Stravinsky, and Horn Examples.....  | 10         |
| Synful Orchestra Needs a Host Sequencer Program.....   | 10         |
| Sequencers We Test With.....   | 10         |
| Activating or Authorizing Multiple Machines and Upgrading Machines.....                        | 10         |
| Mac OSX Installation.....  | 11         |
| Windows Installation.....  | 11         |
| Purchasing and Authorizing Synful Orchestra.....   | 12         |
| Authorizing Multiple Machines.....   | 13         |
| <b>Chapter 2: Using Synful Orchestra.....</b>  | <b>15</b>  |
| Basic Concepts.....  | 16         |
| Channel.....   | 16         |
| Instrument.....  | 16         |
| Programs and Playing Modes and Articulations.....  | 16         |
| Voices.....  | 16         |
| Synthesizer.....   | 17         |
| Bank.....  | 17         |
| Project.....   | 17         |
| Preset.....  | 17         |
| Opening the Control Panel.....   | 17         |
| Logic.....   | 17         |
| Digital Performer.....   | 17         |
| Sonar.....   | 18         |
| Cubase.....  | 18         |
| The Synful Orchestra Control Panel.....  | 18         |
| Assigning Programs to Channels and Selecting Playing Modes.....                                | 20         |
| Using the Synful Orchestra Control Panel.....  | 20         |
| Blocking Midi Program Changes.....   | 20         |
| Using the MIDI Track Settings of The Host Program.....   | 21         |
| Using Keyswitching to Select Playing Modes.....  | 21         |
| Sending MIDI Program Changes and Bank Changes to the Channel.....                              | 21         |
| Determining MIDI Program and Bank Numbers for Synful Orchestra Programs and Playing Modes..... | 21         |
| Easier Switching By Changing Bank Only.....  | 23         |
| Changing Playing Mode Using Symbols in Finale and Sibelius.....                                | 23         |
| MIDI Program Numbering: 1-128 or 0-127.....  | 23         |
| Control Parameters in a Synthesizer.....   | 23         |
| Control Parameters in Programs and Playing Modes.....  | 24         |
| Playing Mode Control Parameters.....   | 24         |
| Program Control Parameters.....  | 26         |

|  |    |
|--|----|
| Localization and Section Controls.....   | 27 |
| Room Size Controls.....  | 28 |
| Listener Location Controls.....  | 29 |
| Player Location Controls.....  | 29 |
| How Does a Real Listener Determine Where a Player Is Located.....  | 30 |
| Localization Mode Options.....   | 30 |
| Absorption Control.....  | 31 |
| Section Controls.....  | 31 |
| CPU Power for Sections.....  | 32 |
| What Happens When the CPU Overloads?.....  | 32 |
| But I Need to Use a BIG ORCHESTRA!.....  | 32 |
| Performance is Better with Delay for Expression ON.....  | 33 |
| Natural Sounding Divisi.....   | 33 |
| Why are Some Controls Disabled.....  | 33 |
| Why Do Some Controls Change When I Move Other Controls?.....   | 33 |
| How Does Synful's Localization Control Work With Reverb Plugins.....                                       | 33 |
| I Set a Player's Location Behind the Listener But it Didn't Sound Like it Was Coming From the<br>Back..... | 33 |
| MIDI Control and Automation of Synful Orchestra Parameters.....  | 34 |
| Renaming, Creating and Deleting Programs.....  | 34 |
| Load Store Default and Local Channel Programs.....   | 35 |
| Saving and Loading Programs Banks and Presets.....   | 35 |
| Logic.....   | 35 |
| Digital Performer.....   | 35 |
| Cubase.....  | 36 |
| Sonar.....   | 37 |
| Delay for Expression.....  | 37 |
| What is Delay for Expression?.....   | 37 |
| When Should Delay for Expression Be Used?.....   | 37 |
| Using Delay For Expression While Overdubbing, or With Other Synthesizers, or With Time Code...             | 37 |
| Questions and Answers About Delay for Expression (DFE).....  | 38 |
| Synful Pitch Wheel.....  | 39 |
| Portamento in Real String and Wind Playing.....  | 39 |
| The Problem with MIDI Pitch-Wheel and MIDI Portamento.....   | 39 |
| The Synful Pitch Wheel Solution.....   | 40 |
| Tips for Good Phrasing.....  | 40 |
| RPM Reconstructive Phrase Modeling.....  | 41 |
| RPM Phrase Database.....   | 42 |
| Searching and Splicing.....  | 42 |
| Coarticulation in RPM - the Key to Natural Phrasing.....   | 43 |
| Synful RPM Additive Synthesis.....   | 43 |
| Windows Tips on Audio Drivers and Low Latency.....   | 43 |
| What is an Audio Device Driver?.....   | 43 |
| What are Some Types of Audio Device Drivers for Windows Machines?.....                                     | 44 |
| What Happens When a Driver Doesn't Work?.....  | 44 |
| How Do I Adjust for the Lowest Latency Without Getting Clicks and Pops?.....                               | 44 |
| Using the New French Horn.....   | 44 |
| No Delay for Expression.....   | 44 |
| Old and New Horns.....   | 45 |
| Expression Brassiness Control.....   | 45 |
| Gliss and Rips.....  | 45 |
| Trills.....  | 45 |
| Mute and Stopped Sounds and the Wah Control.....   | 46 |
| Horn Solo and Horn Sections.....   | 46 |
| The Horndemo Files.....  | 46 |
| Video Tutorials.....   | 46 |

Apple Logic Multi-Channel Choices..... 46



---

# Preface

---

## Welcome

---

Congratulations and welcome to the Synful users community! Synful makes innovative synthesizers with uncommon expressivity, and realism. While Synful synthesizers can be simple to use, there are a number of new concepts which should be studied to take full advantage of the Synful synthesizer capabilities. This manual describes basic capabilities as well as the new concepts in detail. Where appropriate Flash demo movie tutorials are provided to make learning efficient and easy.

Some of the many Synful Orchestra advantages are:

- Woodwinds, brass, and strings sound completely realistic.
- Realistic instrument sections with flexible control, natural division, and more.
- Unprecedented expressive control and natural phrasing.
- Easy to use instruments controlled by ordinary MIDI.
- Patented Synful RPM additive synthesis technology renders samplers obsolete.



---

# Chapter 1

---

## Getting Started

---

### Topics:

- *[Before you Install](#)*
  - *[Mac OSX Installation](#)*
  - *[Windows Installation](#)*
  - *[Purchasing and Authorizing Synful Orchestra](#)*
-

## Before you Install

---

### Mac OS X System Requirements

- Mac OS X 10.6.0 or higher (including Leopard).
- Universal Binary supports Intel Mac or G4 or G5 Power PC Processor
- Synful Orchestra will run on any Intel Mac. The number of real-time voices generated depends on the CPU speed and the number of CPU cores. A 1 GHz or faster Intel CPU is recommended. Synful Orchestra takes advantage of the SSE3 SIMD Extensions.
- 1G memory required (2G recommended)
- 200 Mbytes of hard disk space required.

### Windows System Requirements

- Windows Vista32/64, Windows XP/XP64, Windows 2000, Windows 7, Windows 8, or higher required.
- Synful Orchestra will run on any Pentium class processor. The number of real-time voices generated depends on the CPU speed and number of CPU cores. A 1 GHz or faster CPU is recommended. Synful Orchestra takes advantage of the SSE3 SIMD Extensions.
- 1G memory required (2G recommended)
- 200 Mbytes of hard disk space required.

### Free Crossgrade for Windows and Mac

If you already own Synful Orchestra or if you just purchased a new license you can use it on both your Mac and Windows machines. Free Crossgrade for Mac and Windows.

### Wagner, Beethoven, Stravinsky, and Horn Examples

One of the best ways to learn Synful Orchestra is to examine the examples included with the installation. There are examples from several composers including a Beethoven String Quartet, Stravinsky Rite of Spring, and Wagner Tristan und Isolde, and an extended Horn Demo. The examples are provided as project files for Logic, Protools, Cubase, Sonar. There are also ordinary midi files.

Mac OSX. Go to /Applications/Synful/Synful Orchestra/Examples.

Windows. Go to /Program Files/Synful/Synful Orchestra/Examples

### Synful Orchestra Needs a Host Sequencer Program

Synful Orchestra is a plugin program that cannot run by itself. It needs to run with a sequencer host program. It supports four plugin formats AU, VST, AAX, and RTAS. AU is Mac only and the rest run on both Mac and Windows. There are many AU, VST, AAX, and RTAS host programs including some free ones. See below for the host sequencers we test with.

### Sequencers We Test With

We currently test Synful Orchestra with Apple Logic, Digital Performer, Cubase SX, Sonar, and Digidesign Protools. There may be some problems with other sequencers that we do not currently test with -- see [Known Problems With Synful Orchestra](#). Please let us know by sending an email to [support@synful.com](mailto:support@synful.com) and we will try to address the problem. Generally if Synful Orchestra works with one product from a company it will work with others. For example Nuendo, Cubase SL, and V-stack all work well since Cubase SX works well.

### Activating or Authorizing Multiple Machines and Upgrading Machines

If you have purchased a Synful Orchestra license and received an Activation Code then you can authorize Synful Orchestra on more than one personal machine. For example, a desktop and a laptop or a PC and a MAC. The Synful Orchestra license is considered to be for one person. However, we will support as many machines as you have within

reason. 2-3 is normal, 10 is way too many! Do not share your Activation Code with friends or colleagues. It identifies you as the only person who can receive authorizations for your personal Synful Orchestra License.

Certain kinds of hardware upgrades and sometimes reformatting of the hard disk will require a re-authorization. Simple follow the steps described in the previous section to re-authorize.

## Mac OSX Installation

---

Mac OS X 10.6.0 or higher is required.

This section will guide you through the installation of the Synful Orchestra software on a Mac running OSX. The installation process installs all the files you need automatically. It installs AU, VST, RTAS and AAX versions of Synful Orchestra. If you are installing for a Free Trial then NO AUTHORIZATION IS REQUIRED -- just install and run for 15 days.

1. If you have not already done so go to [Mac Updates](#) and download the latest Synful Orchestra for MAC release. This will download Synful Orchestra XXX.zip where XXX is the version number. When the zip file arrives on your system it should automatically be converted to the file Synful Orchestra XXX.mpkg. If you have already downloaded the free-trial and installed it and are ready to purchase a Synful Orchestra license or have already purchased a license and received a Activation Code from Synful, you do not need to download and install again. Just go to [Purchasing and Authorizing Synful Orchestra](#) on page 12
2. Double click on the SynfulOrchestra.mpkg to begin the installation. Follow the onscreen installation instruction. Synful Orchestra must be installed on the main system hard disk. This is usually called Macintosh HD.
3. Now you can launch your sequencer program and begin using Synful Orchestra. To see how to begin using Synful Orchestra open the Synful Orchestra Help. To open Synful Orchestra Help browse to /Applications/Synful/Synful Orchestra/ in the Finder and double click on Help.htm. Select one of the Tutorials -- XXXX Setup of Synful Orchestra -- where XXXX stands for your sequencer. If there isn't a tutorial for your sequencer then watch one of the tutorials anyway since much of the material is independent of the sequencer. You will need an internet connection to view the video tutorials.
4. If you have purchased a Synful Orchestra license then don't forget to Authorize or the program will stop running -- see [Purchasing and Authorizing Synful Orchestra](#) on page 12 You can authorize by pressing the Authorize button at the upper left of the Synful Orchestra Control Panel.
5. If you have installed the trial version of Synful Orchestra it will run for 15 days. At any time during the 15 days you can see the remaining time available at the top of the Synful Orchestra Control Panel.

## Windows Installation

---

This section will guide you through the installation of the Synful Orchestra software on a Windows 32 bit or 64 bit machine. The installation process installs all the files you need automatically. You can install both DXi and/ or VST versions of Synful Orchestra. The installation is the same for the free trial and purchased versions of the program. If you are using Cakewalk Sonar you should install the native DXi version and not use the VST adaptor available in Sonar. If you have not already done so go to [Download Synful Orchestra](#) and download the install file. Make sure to download the 32 bit version if you are running a 32 bit version of Windows or the 64 bit version if you are running a 64 bit version of Windows. The download file synful\_orchestra\_XXX\_VV\_setup.exe can be placed anywhere on your hard disk. "XXX" stands for the Synful Orchestra version number and VV stands for either 32 or 64 corresponding to the 32 or 64 bit version. The setup .exe is the full version of the program. If you have already downloaded the free-trial and installed it and have now purchased a license and received an activation code you do not need to download again --see [Purchasing and Authorizing Synful Orchestra](#) on page 12

1. You can install Synful Orchestra directly over any previous release. That release will be deleted and replaced with the new update. However, if you have previously moved the file Synful Orchestra VST.dll by hand into a VSTPlugins folder then you should remove it by hand now.

2. Browse to the location of the `synful_orchestra_XXX_VV_setup.exe` on your hard disk and double click on it to begin the installation. A Microsoft Windows security dialog may appear. If it does, click on Run to proceed with the installation. The Welcome to Synful Orchestra Installation Wizard dialog appears. Click Next.
3. Read the End User License Agreement and click the "I accept the license agreement" radio button. Then click Next.
4. The Synful Orchestra Readme Dialog appears. This will inform you of what is new in the latest release. You can read this now or later -- click Next.
5. The Select Features dialog appears. There is only one feature -- VST. Here you can browse your VST plugins folder where you want the VST dll file to be placed. For example, if you run Cubase this might be `C:\Program Files\Steinberg\Cubase 4\VSTPlugins`. If you have previously moved the file `Synful Orchestra VST.dll` by hand into a VSTPlugins folder then you should remove that old file by hand now, so that after installation you will be certain to load the new Synful Orchestra VST.dll. Click Next.
6. The Update System dialog appears. Wait until the installation is finished. This may take a few minutes. When the Synful Orchestra Has Been Successfully Installed dialog appears click Finish.
7. You will be prompted to reboot your computer. You must reboot before you can successfully use Synful Orchestra.
8. When running Synful Orchestra for the first time on a machine it will be in Free Trial Mode. If you have purchased a license and received an activation code then you can authorize Synful Orchestra by clicking on the Authorize button at the top of the Synful Orchestra Control Panel. See [Purchasing and Authorizing Synful Orchestra](#) on page 12 for details.
9. You can immediately begin using Synful Orchestra. To see how to begin using Synful Orchestra open the Synful Orchestra Help and then view one of the setup video tutorials. You will need an internet connection to view these online video tutorials. To open Synful Orchestra Help click on your Windows start menu and then the programs pop-up list. There will be a new Synful entry in this list. Click on it and on the Synful Orchestra submenu. Then select Help.

## Purchasing and Authorizing Synful Orchestra

---

You must authorize the software or it will stop running after 15 days.

If you have installed the free trial version of Synful Orchestra and have not purchased a license it will run for 15 days. You do not need to authorize -- just install and run. If you are ready to purchase and authorize then follow the instructions below.

1. If you have not already installed Synful Orchestra on your machine then see [Mac OSX Installation](#) on page 11 or [Windows Installation](#) on page 11.
2. If you have not already done so, you can purchase a Synful Orchestra license and receive an Activation Code by going to [Buy a Synful Orchestra License](#). Follow the directions on the site to finalize your purchase. After a short time -- usually a few minutes but never more than a few hours -- you will receive a confirmation email including a Synful Orchestra Activation Code.
3. Start up your host sequencer and open the Synful Orchestra Control Panel -- see [Opening the Control Panel](#) on page 17 for details.
4. If your trial period has already expired then the Control Panel will consist of a screen with a single Authorize button. If your trial period has not expired then at the upper left of the Synful Orchestra Control Panel is the Click to Authorize button. Click it and an Authorization panel comes up with instructions. Those instructions are repeated here in greater detail. Click the new Authorize button at the bottom of this panel.
5. The Challenge/Response dialog box appears. Click on the Copy Challenge button. This copies the challenge code in the Challenge Box to the clipboard.
6. Prepare an email message to `sales@synful.com` with the subject Authorize Synful Orchestra.
7. In the text of this message paste the challenge code you copied in step 5 and the Activation Code you received in your original order confirmation email. Send the message.
8. Within 24 hours you will receive a response email from Synful. This response email will contain a response code similar to the challenge code you sent earlier (it is longer however). Again open the Synful Orchestra control

panel and click Authorize to get to the Challenge/Response panel. Copy the response code from the Synful email message and paste it into the response box in the Challenge/Response panel.

9. Click the Next button. You should see a message saying Authorized this system with a valid response. You have successfully authorized Synful Orchestra. Click the Finish button. You are DONE! You should now be able to run Synful Orchestra permanently on your machine.
10. If you do not see the success message or if you run Synful Orchestra and are told it is not authorized then go back and review this procedure from the beginning. If you continue to have problems, contact [support@synful.com](mailto:support@synful.com).
11. If you are authorizing Synful Orchestra on a machine that does not have your email program, then after step 5 you can paste the challenge code into a simple text file and copy that file to your email machine. In the other direction copy the response code to a text file on your email machine and copy that file to your install machine. Open the text file and copy the response code into the Challenge/Response dialog box.

## Authorizing Multiple Machines

You can authorize Synful Orchestra on more than one personal machine. For example, a desktop and a laptop or a PC and a MAC. The Synful Orchestra license is considered to be for one person. However, we will support as many machines as you have within reason. 2-3 is normal, 10 is way too many! Do not share your Activation Code with friends or colleagues. It identifies you as the only person who can receive authorizations for your personal Synful Orchestra License. Certain kinds of hardware upgrades and sometimes reformatting of the hard disk will require a re-authorization. Simple follow the steps described in the previous section to re-authorize.



---

# Chapter

# 2

---

## Using Synful Orchestra

---

### Topics:

- *Basic Concepts*
- *Opening the Control Panel*
- *The Synful Orchestra Control Panel*
- *Assigning Programs to Channels and Selecting Playing Modes*
- *Control Parameters in a Synthesizer*
- *Control Parameters in Programs and Playing Modes*
- *Localization and Section Controls*
- *MIDI Control and Automation of Synful Orchestra Parameters*
- *Renaming, Creating and Deleting Programs*
- *Load Store Default and Local Channel Programs*
- *Saving and Loading Programs Banks and Presets*
- *Delay for Expression*
- *Synful Pitch Wheel*
- *Tips for Good Phrasing*
- *RPM Reconstructive Phrase Modeling*
- *Windows Tips on Audio Drivers and\_Low Latency*
- *Using the New French Horn*
- *Video Tutorials*

Topic paragraph

## Basic Concepts

---

In this section we define the basic concepts Synthesizer, Program, Playing Mode, Instrument, Channel, Voice, and Bank. In later sections we will describe how to manipulate these.

### Channel

This refers to a MIDI Channel. There are sixteen MIDI Channels in one instance of the Synful Orchestra synthesizer. More than one instance of the Synful Orchestra synthesizer plugin can be inserted into a project providing a virtually unlimited number of MIDI channels. There are two ways to view the MIDI channels in a synthesizer: multi-channel view and single channel view -- see [The Synful Orchestra Control Panel](#) on page 18.

### Instrument

Synful Orchestra is shipped with a fixed collection of Instruments -- Flute, Violin, Trumpet, etc. You do not deal directly with Instruments. You deal with Programs and Playing Modes. You can create many different Programs that use the basic set of Instruments.

### Programs and Playing Modes and Articulations

A Program includes an Instrument and a collection of Playing Modes. For example, for a Violin Program the Playing Modes include arco, pizzicato, tremolo, harmonics, and others. Within one Playing Mode are generally many different Articulations. For example for the Violin arco Playing Mode there are slurred, legato, detach (connected separately bowed notes), spiccato (short bounced bow notes) and other Articulations, all with loud and soft variations. These different articulations within one Playing Mode are automatically selected by Synful Orchestra based on the standard MIDI input. No special controls are needed. Different Playing Modes, however, are selected by the user. This selection can be done from the Synful Orchestra Control Panel, or by using either Keyswitching or MIDI Program Changes. A control parameter is a value that you can modify to alter the sound of the Instrument. There are control parameters associated with the Program as a whole. For example, physical player position on the virtual stage is determined by a number of Program control parameters. The settings of these Program control parameters apply to the entire Program regardless of what Playing Mode is currently selected. Each Playing Mode within a Program has its individual control parameters. For example, Harmonic Tilt is a Playing Mode control parameter. When you select the pizzicato violin Playing Mode you can adjust Harmonic Tilt to something that sounds good to you. When you switch to the arco violin Playing Mode the Harmonic Tilt parameter will switch to the last value set in the arco Playing Mode. You can adjust the arco Harmonic Tilt to whatever you like. When you switch back to pizzicato Playing Mode the Harmonic Tilt parameter you last selected in the pizzicato mode will be restored (See [Control Parameters in Programs and Playing Modes](#) on page 24 for details). At any time one and only one Program can be assigned to a MIDI Channel. There are several ways to change the Program assigned to a particular MIDI Channel (see [Assigning Programs to Channels and Selecting Playing Modes](#) on page 20). When you change the Program assigned to a channel you can then select from the different Playing Modes in that Program. If you click on a Program name you can rename the program. Renaming the Program creates a new copy of the entire Program (see [Renaming, Creating and Deleting Programs](#) on page 34). The new copy includes all the Playing Modes for the Program. You can then modify the control parameters of the new Program or any of its Playing Modes without affecting the parameter settings of the old Program. Many Programs can refer to the same Instrument. For example, there can be three Programs that all refer to the Violin Instrument but have different control parameters. See [Control Parameters in Programs and Playing Modes](#) on page 24 and [Renaming, Creating and Deleting Programs](#) on page 34.

### Voices

When a user plays a note on a MIDI controller or from a sequence a Channel receives a note-on message. If that channel has a Synful Orchestra Program assigned to it then in response to the note-on message a Voice is allocated for that note on that Channel. If you play many notes simultaneously, all on the same Channel, then many Voices are allocated for that Channel. When a Voice is allocated to a Channel it plays a note according to the Program and currently selected Playing Mode that is assigned to that Channel. Since only one Program can be assigned to a Channel and only one Playing Mode selected at any given time, this means that all Voices assigned to a Channel refer to the same Program, Playing Mode, and Instrument. For example, if you play the chord C-E-G on MIDI Channel

2, and if a Cello Program and arco Playing Mode has been assigned to Channel 2, then 3 voices will be allocated to Channel 2, and you will here C-E-G played by the three arco cello Voices. The number of simultaneous Synful Orchestra Voices is limited by the computer power of your machine. There is actually an exception to the rule of one Program and Playing Mode per channel. If you play a note, for example a cello arco note, and hold it, and then switch the Program and/or Playing mode and play a second note, for example switch to violin pizzicato while holding the first cello arco note, then you will hear both the cello arco note and the new violin pizzicato note on the same channel at the same time. In other words there is note overlap when you switch Programs and/or Playing Modes on a channel.

## Synthesizer

A single instance of the Synful Orchestra Synthesizer plugin has sixteen MIDI Channels. Each MIDI Channel can have a different Program assigned to it with different selections of Playing Modes. More than one Synthesizer can be inserted in a project. In addition to the control parameters associated with each Program and its Playing Modes, there are a number of control parameters associated with each Synthesizer (see [Control Parameters in a Synthesizer](#) on page 23).

## Bank

The collection of currently active Synthesizer plugins shares a Bank of Programs. This Bank is the collection of Programs that can be assigned to Channels of the Synthesizers. When the first Synthesizer is inserted in a project it has the default Bank of Programs shipped with Synful Orchestra. You can then copy and modify these Programs to create new Programs in the Bank (see [Renaming, Creating and Deleting Programs](#) on page 34, and [Saving and Loading Programs Banks and Presets](#) on page 35). When you create new Programs they immediately become available to all the active instances of the Synthesizer.

## Project

The host program (for example Logic, Digital Performer, Sonar or Cubase) operates on Project files. When a Project is saved or loaded the entire state of each Synthesizer instance and all the Programs in the Project are saved or loaded in addition to all MIDI sequence information. This includes Program assignments to Channels, Synthesizer parameters, Program parameters, etc.

## Preset

This means different things depending on the host Program. In Sonar, for example, saving or loading a Preset saves the state of a single Synthesizer instance -- including all Programs in the Synthesizer Banks, Program assignments to Channels, Synthesizer parameters, etc. (see [Saving and Loading Programs Banks and Presets](#) on page 35).

## Opening the Control Panel

---

If you are using a different sequencer from one listed here - Live, Fruity Loops, Garage Band, etc - then you should look at one of the paragraphs below anyway because the process is probably similar on your sequencer.

### Logic

You must first create an instance of the Synful Orchestra plugin in Logic see [Video: Setting up Synful Orchestra in Logic](#). In Logic's Arrange view double click on the small Audio Instrument icon in the track inspector of the desired track. This brings up the Mixer view. In the Mixer view double click on the Synful Orchestra insert in the desired audio instrument channel strip. This brings up the Synful Orchestra Control Panel.

### Digital Performer

You must first create an instance of the Synful Orchestra plugin in Digital Performer -- see [Video: Setting up Synful Orchestra in Digital Performer](#). From the Project menu in Digital Performer select Effects. This brings up the Effects mini-menu. You may see the Synful Orchestra Control Panel right away depending on what selections had already been made. If you don't see the Control Panel then on the Effects mini-menu, on the left-most of the three selection

boxes select the Synful Orchestra instrument track. If this doesn't bring up the Control Panel then on the middle of the three selection boxes select the Instrument insert. This will bring up the Control Panel.

## Sonar

You must first create an instance of the Synful Orchestra plugin in Sonar [Video: Setting up Synful Orchestra in Sonar](#). Then see [Video: Open Control Panel in Sonar](#).

## Cubase

You must first create an instance of the Synful Orchestra plugin in Sonar [Video: Setting up Synful Orchestra in Cubase](#). Then see [Video: Open Control Panel in Cubase](#).

## The Synful Orchestra Control Panel

The Synful Orchestra Control Panel is the same for all versions of the Synful Orchestra (AU,,VST,RTAS,AAX etc). The Control Panel represents a single instance of the Synful Orchestra synthesizer plugin. There can be many instances of Synful Orchestra synthesizers inserted in a Project, each with its own Control Panel.

**Figure 1: Synful Orchestra Control Panel**



## Chanel Grid

At the top left of the Control Panel is the Channel Grid. Here you can assign Programs to any of the sixteen MIDI Channels of the synthesizer. Each MIDI Channel is represented by a rectangle of the Channel Grid. Clicking in one

of the channel boxes causes it to change color and become the currently selected MIDI channel. A MIDI Channel can have no Program assigned to it in which case it is blank. When a MIDI has a Program assigned to it then two names appear in the channel box one above the other. The top name is the Program name and the bottom name is the Playing mode name. In the Synful Orchestra Control Panel screenshot above MIDI Channel 1 has the Program violin1 selected with the arco playing mode. Clicking on the downward triangle at the right of the channel box brings up a pop-up list from which you can select any Program and Playing Mode to make it become the active Program for the currently selected MIDI channel. When a MIDI channel received a midi message from a sequencer, or controller a little red light lights up at the bottom right of the channel box.

### Block Midi Program Changes

Under the Synful Orchestra logo at the upper of the Control panel are three buttons. The "block midi program changes" button. When this is active its color becomes brighter. and any MIDI program changes arriving from a MIDI controller, from a MIDI sequence, or from the track inspector of the host sequencer will be ignored. This is useful if you want to assign Programs to channels using the Synful Orchestra Channel Grid and don't want them to change accidentally due to a button press on a keyboard, etc. If you want to change Programs or Playing Modes dynamically from a sequence or other source then you need to deactivate the "block midi program changes button" by clicking on it again - see [Assigning Programs to Channels and Selecting Playing Modes](#) on page 20 for more details.

### All Notes Off

To the right of the "block midi program changes" button is the "all notes off button". This is for panic situations where there may be hanging notes that do not turn off for various reasons. This button will turn off all notes for all Synful Orchestra synthesizer instances.

### Help

To the right of the "all notes off button" button is help button that opens the Synful Orchestra help documentation.

### Control Parameters for Currently Selected Channel

Below the Channel Grid are a group of horizontal sliders. These represent the real time control parameters for the Program and Playing mode of the currently selected MIDI Channel in the Channel Grid. The names of the parameters will change depending on what Program and Playing mode is currently selected. Not all the horizontal sliders are used for all programs. Unused sliders are grayed out and have no parameter name. Below the horizontal sliders is a collection of rotary dials and buttons divided into different sections. The Room Controls and Listener Location controls apply to the entire synthesizer instance -- i.e. to all channels of the synthesizer. All the other dials and buttons apply to the current channel only. See [Control Parameters in Programs and Playing Modes](#) on page 24 for details.

### Control Parameters for a Synthesizer

As mentioned above the Room Controls and Listener locations controls apply to the entire synthesizer. In addition the Tune Orchestra horizontal slider applies to the entire synthesizer also. See [Control Parameters in a Synthesizer](#) on page 23 for details.

### Keyswitch Controls

Below the rotary dials and buttons and just above the MIDI keyboard display is a horizontal strip of buttons. The left most button is actually a drop down menu control -- the Keyswitch control. This is used to setup keyswitching for dynamically changing playing modes from a MIDI keyboard or other MIDI controller. For more details see [Assigning Programs to Channels and Selecting Playing Modes](#) on page 20.

### Delay for Expression

To the right of the Keyswitch control is the "delay for expression" button. For more details see [Control Parameters in a Synthesizer](#) on page 23.

## Synful Pitch Wheel

To the right of the "delay for expression" button is the "synful pitch wheel" button. This button applies only to the currently selected channel. It sets up a special behavior for the MIDI pitch wheel controller. For more details see [Synful Pitch Wheel](#) on page 39.

## Load, Save, Delete

To the right of the "synful pitch wheel" button are three buttons: load default program; save default program; and delete program. See [Renaming, Creating and Deleting Programs](#) on page 34 for details.

## Midi Keyboard

At the bottom of the Control Panel is the MIDI Keyboard control. Pressing a key on this keyboard with the mouse causes a note on message to be sent to the currently selected channel. If there is a Program assigned to the channel then the channel output can be listened to. To the left and right of the keyboard are little gray triangles that allow the keyboard range to be selected. The C4 label on the keyboard represents middle C.

## Assigning Programs to Channels and Selecting Playing Modes

---

For a brief description of Programs and Playing Modes see [Basic Concepts](#) on page 16.

Programs can be assigned to Channels, and Playing Modes selected using four methods:

1. The Synful Orchestra Control Panel
2. The MIDI track settings of the host program
3. Sending MIDI program changes to the Channel from a keyboard or MIDI sequence
4. Using Keyswitches for Playing Mode Selection

## Using the Synful Orchestra Control Panel

In the Synful Orchestra Control Panel the Channel Grid is found at the upper left. The sixteen numbered boxes in the Channel Grid represent the sixteen MIDI channels of the Synthesizer instance. Clicking in one of the boxes causes it to turn a brighter color and become the currently selected MIDI channel. The top name in each box corresponds to the Program assigned to that Channel. The bottom name in each box corresponds to the Playing Mode. Clicking on the downward triangle in any Channel Grid box pops up the Program and Playing Mode selection list. Any currently available Program and Playing Mode can be selected. The names will then appear in the box and the selected Program and Playing Mode will be assigned to the MIDI channel corresponding to that box. It is possible to select "none" as the Program, in which case the Channel Grid box will be empty and no Program or Playing Mode will be assigned to that MIDI channel. A single Program and Playing Mode can be assigned to multiple Channels. Synthesizer, Program, Channel and Playing Mode selections and settings are saved and loaded when the Project is saved or loaded. Depending on the host sequencer program an individual Program can be saved to a file, or a set of programs called a Bank can be saved.

## Blocking Midi Program Changes

Beneath the Synful Orchestra logo at the upper right of the Control Panel is the "block midi program changes" button. When it is clicked and enabled it turns a brighter color and MIDI program changes coming from a source external to Synful Orchestra will be ignored. Any program changes arriving from a MIDI controller, from a MIDI sequence, or from the track inspector of the host sequencer will be ignored. Keyswitches are an exception. They continue to function even if the "block midi program changes" button is enabled. This is useful if you want to assign Programs to channels in the Synful Orchestra Channel Grid and do not want to change them dynamically while a score is playing. If you want to change Programs dynamically from a sequence or other source then you need to disable the "block midi program changes button" by clicking it again. If you forget enable the "block midi program changes" button then that program selection will generally be overridden by the host sequencer. When this occurs you will see the program names in the Control Panel change as midi program changes are received. The most common case where this occurs is when the program selection in the host sequencer track inspector overrides the selection in the Synful

Orchestra Control Panel. External midi program changes typically come from three sources: The patch or program settings in the sequencer track inspector if there is one. Program changes embedded in the midi sequence itself. Program changes coming from a midi keyboard or other controller. If you find that your Control Panel settings are automatically changing when you start playing a sequence or when you rewind the sequencer be sure to check these issues. You may want to enable the "block midi program changes" button.

## Using the MIDI Track Settings of The Host Program

In most host sequencer programs there is a way to assign an initial Program to the MIDI Channel associated with each MIDI Track. See the [Video Tutorials](#) on page 46 for setting up Synful Orchestra with a number of sequencer programs or consult the documentation for the host program. Note that this only sets the initial Program. The initial program is assigned to the channel when you hit the start or play button on the host program transport control. Any MIDI program changes embedded in a MIDI sequence or arriving from a MIDI controller will override this initial setting. If the "block midi program changes" button is enabled in the Control Panel the initial Program setting in the host program will be ignored. For example, if the "block midi program changes" button is disabled, and if the Control Panel setting is "violin" and the track setting is "flute" then as soon as you hit the host's start/play button the Control Panel setting will be updated to "flute" and the sound for that channel will be flute.

## Using Keyswitching to Select Playing Modes

When you use keyswitching to select Playing Modes you reserve the bottom several keys of your keyboard controller as special selection switches. They no longer play notes. Just above the MIDI keyboard control on the Control Panel at the extreme left is the keyswitch start pop-up control. This is used to select the starting note for keyswitching. You usually want to set this to the lowest note on your keyboard. The keyswitch display shows the lowest or starting keyswitch note. The note is displayed in two ways: note-name-octave format — C3, E2, etc; and midi note number between 1 and 128. Clicking on the downward triangle to the right of the note name and number pops up a list of popular keyboard sizes: 88 keys, 76 keys, 49 keys, etc. By selecting one of these you set the keyswitch start note to the bottom key of the corresponding keyboard. When you select "keyswitch off" from the keyswitch start pop-up menu then keyswitching is disabled. The smaller keyboards have octave transposition switches. The keyswitch start note refers to the bottom note of the keyboard when the keyboard has no octave transposition. You can also type in the keyswitch start note, either using note-name-octave format or midi note number. The keyswitch start note corresponds to the top entry in the Playing Mode list. One note above the start note corresponds to the second entry in the Playing Mode list, two notes above to the third entry, etc. For example, if you have a 49 key keyboard and set it to no octave transposition, and select 49 keys from the keyswitch start menu then the keyswitch start note is set to C3 (36) which is the bottom note on the 49 key keyboard. If you select Violin as the Program with then the first Playing Mode is "arco". If you play low C3 on the keyboard this will select "arco" Playing Mode. If you play low C#3 on the keyboard this will select the next playing mode "pizzicato", D3 selects "bartok-pizz", etc. Immediately to the left keyswitch start display is a small button marked "ALL". . When it is enabled your next keyswitch start selection applies to all channels of all synthesizer instances in the current project. If "ALL" is disabled then your next keyswitch start selection applies only to the currently selected MIDI channel.

## Sending MIDI Program Changes and Bank Changes to the Channel

A MIDI program change message can be sent from most MIDI controllers. Keyboards often have buttons dedicated to sending program changes. If a MIDI sequence is recorded live from a MIDI controller any program changes coming from the controller will be recorded with the sequence. When the sequence is played back those program changes will be sent to Synful Orchestra. Most host programs provide a means to insert a program change in an already recorded MIDI sequence using editing commands. Consult the host program documentation for this operation.

## Determining MIDI Program and Bank Numbers for Synful Orchestra Programs and Playing Modes

In order to send a MIDI program and bank change to Synful Orchestra from a host sequencer or from a MIDI controller you need to know the MIDI program number and bank number corresponding to the Synful Orchestra Program and Playing Mode.

The MIDI program and bank numbers associated with a Synful Orchestra Program and Playing Mode can be determined from the Playing Mode pop-up list in the Synful Orchestra Control Panel. To see the Playing Mode pop-

up list, first click on the downward triangle in any channel box of the Channel Grid in multi-channel view, or on the downward triangle in the Program Select in the single-channel view, to select the desired Program, and then click on the downward triangle in Playing Mode display. In the Playing Mode pop-up list you see the name of the playing mode followed by the currently selected keyswitch start note, then the patch number (the same as the MIDI program number) and finally the MIDI bank. The MIDI bank is 2 numbers separated by a comma in parenthesis. The bank number on the left is the "most significant byte" of the bank number (msb). The bank number on the right is the "least significant byte" of the bank number (lsb).

For example, if you select Violin and then look at the Playing Mode pop-up list the first two entries are:

- arco key=OFF patch=(40) bank=(0,0)
- pizzicato key=OFF patch=(40) bank=(2,0)

This means, for example, that the pizzicato Playing Mode of Program "Violin" corresponds to MIDI program (or patch) 40 of bank (2,0) where 2 is the most significant byte of the bank number and 0 is the least significant byte.

In the MIDI world the most significant byte of the bank number corresponds to MIDI continuous controller 0 (cc0) and the least significant byte corresponds to MIDI continuous control 32 (cc32). If you have a MIDI keyboard controller with programmable sliders, and set one of the sliders to control cc0 then you will be changing the most significant byte of the bank when you move that slider. Similar, if you set another of the sliders to control cc32 then you will be changing the least significant byte of the bank when you move that slider.

So, if you want to change to the Violin pizzicato Playing Mode on MIDI channel 1 using a MIDI bank and program change then first make sure that MIDI channel 1 is unlocked on the Synful Orchestra Control Panel channel grid, then set cc0 to 2, cc32 to 0 on channel 1 and then send a MIDI program change message with value 40 to channel 1. If you have channel 1 selected on the Synful Orchestra Control Panel then you will see it switch to Violin pizzicato.

Synful Orchestra always has the least significant bank byte (cc32) set to zero, so you can set that to zero and never change it.

All of the Playing Modes for a given Program in Synful Orchestra use the same Program Number. For example, all of the Violin Playing Modes ( arco, pizzicato, sul-pont, etc) use program (or patch) number 40. The different Playing Modes for the Program are then selected by changing the most significant bank number (cc0). For families of instruments with similar Playing Modes such as the strings, the bank numbers for different Playing Modes are always the same, so pizzicato Playing Mode is always selected by bank (2,0) for all the strings.

When you click on the downward triangle in any channel box of the Channel Grid in multi-channel view, or on the downward triangle in the Program Select in the single-channel view, to display the Program List you will see the program (patch) number and bank numbers associated with the first playing mode of the Program. So when you see Violin on the Program pop-up list you see Violin patch=(40) bank=(0,0) where the most significant byte bank number (0 in this case) corresponds to the arco Playing Mode.

In some host sequencers and controllers the bank number is shown as a single large number. This number is equal to 128 times the most significant bank byte plus the least significant bank byte:  $\text{single\_bank\_number} = 128 * \text{most\_significant\_byte} + \text{least\_significant\_byte}$ .

So, if a host sequencer uses this representation for bank numbers then the Violin pizzicato bank will correspond to bank 256:  $\text{Violin pizzicato bank} = 256 = 2 * 128$ .

Some sequencers, notation editors, and controllers allow you to select how a bank and program change is sent to a plugin. The most general way a program change is sent, and the way Synful Orchestra expects the program change to be sent, is to first send the bank number followed by the actual program change command with the program number. The bank number is sent as two continuous controller MIDI messages. The first message sends the most significant byte of the bank number to MIDI continuous controller 0 (cc0). The second message sends the least significant byte of the bank number to MIDI continuous controller 32 (cc32). However, you may need to select the bank sending method in the sequencer. If this selection is available then select cc0, cc32 for the bank select method.

When you create a new Program in Synful Orchestra by renaming an existing program (see Renaming, Creating, Copying, and Deleting Programs), a new program (patch) number and bank number are automatically assigned to this Program. These numbers will appear on the pop-up list along side the name of the new Program. In the rare event that you create hundreds of your own programs Synful Orchestra may finally resort to using the least significant bank byte rather than leaving it set to zero.

## Easier Switching By Changing Bank Only

Often you may want to change between different Playing Modes within a program. For example, you may want to change from violin arco to violin pizz. We discussed how to do this with keyswitching. You can also do this by changing the bank number only without sending a MIDI program change message. If you change the bank by changing cc0 then the Playing Mode will change when you play the next note.

## Changing Playing Mode Using Symbols in Finale and Sibelius

Program number 127 is special. If you change bank by changing cc0 and then send a MIDI Program Change message with program number 127, then the bank will change but the previously selected program will be maintained. This allows you to change Playing Modes within a Program. This is useful in Finale or Sibelius when you want to create a *pizz* symbol that sends a bank change without having to know what the current program is. If you create a *pizz* symbol that sets bank most significant byte (cc0) to 2 (bank = 256) then sends a Program Change of 127, then regardless of the currently selected program -- violin, viola, cello, or bass -- the Playing Mode will change to *pizz*.

## MIDI Program Numbering: 1-128 or 0-127

There is disagreement among manufacturers of controllers and sequencers. Some number MIDI program (or patches) from 0-127 and some number them from 1-128. Synful Orchestra (along with most manufacturers) uses the 0-127 method. In this case the Synful Orchestra violin corresponds to MIDI program 40. If you encounter a sequencer or controller that uses the 1-128 method you may need to add 1 to the patch (or program) number. In this case selecting Violin might require a program number value of 41 from the controller.

## Control Parameters in a Synthesizer

---

The Control Parameters affect an entire Synful Orchestra synthesizer instance.

- **delay for expression** — this is an on-off check box. Delay for expression is a special feature of Synful Orchestra that allows for added expressivity for recorded midi sequences — see [Delay for Expression](#) on page 37 for complete details. At the lower right of the delay for expression checkbox is a smaller all chans/synths checkbox. When this is selected your delay for expression selection applies to all channels on all active synthesizers in the current project. When it is unselected (unchecked) the delay for expression selection applies to the current channel only. A common method of working is to lay down successive tracks live from a MIDI controller. In this case you may want to have delay for expression on for all tracks/channels except for the current track/channel you are playing into live. To do this select the channel you are currently recording on. Set all/chans synths on and then set delay for expression on -- this sets delay for expression on for all synths/channels/tracks. Then switch all chans/synths off and set delay for expression off. This turns delay for expression off for the current track. There is no Midi Control available for delay for expression.
- **tune orchestra** — this slider is found just below the Channel Grid in the Synful Orchestra Control Panel. The value of the slider is displayed in hertz (Hz) or vibrations per second and gives the frequency of A above middle C. Standard tuning is A=440 Hz so this is the Synful Orchestra default. Many professional orchestras tune slightly higher than A 440 Hz -- for example, A 445 Hz -- to achieve a more "brilliant" sound. You can adjust the tune orchestra slider to achieve desired brilliance or to match other live instruments or recordings. The tune orchestra slider controls all channels of all active synthesizers in the current project. It can be automated using MIDI continuous controller 110 (cc110) — see [MIDI Control and Automation of Synful Orchestra Parameters](#) on page 34.
- **room size W** — see [Localization and Section Controls](#) on page 27.
- **room size L** — see [Localization and Section Controls](#) on page 27.
- **room size H** — see [Localization and Section Controls](#) on page 27.
- **golden ration** — see [Localization and Section Controls](#) on page 27.
- **listener loc X** — see [Localization and Section Controls](#) on page 27.
- **listener loc Y** — see [Localization and Section Controls](#) on page 27.

## Control Parameters in Programs and Playing Modes

---

The Program and Playing Mode control parameters are adjusted by sliders and switches on the Synful Orchestra Control Panel. The positions of the sliders and switches reflect the values of the currently selected program and playing mode. Many of the control parameters can be controlled by standard MIDI continuous controllers. If this is the case then there is a MIDI cc assignment display to the right of the control slider see.

### Playing Mode Control Parameters

Each Program may have several different Playing Modes. For example, the Violin Program has Playing Modes that include arco, pizzicato, tremolo, harmonics, and others. Each Playing Mode within a Program has its individual control parameters. For example, vibrato depth and attack noise are Playing Mode control parameters for the Violin arco Playing Mode. Each Playing Mode within a Program can have these controls set differently. The Playing Mode control parameters are adjusted by sliders and number boxes on the Synful Orchestra Control Panel. The positions of the sliders and switches reflect the values of the currently selected Playing Mode within the current Program see [The Synful Orchestra Control Panel](#) on page 18. The available control parameters change depending on the currently selected Playing Mode. For example the tremolo rate control is available only in the tremolo and sul-pont-tremolo Playing Modes. The parameter name under the slider will often change when the Playing Mode. When a slider has no function in a Playing Mode then the number box to the left of the slider for that control displays N/A (not available) and the slider cannot be moved. All of the Playing Mode control parameters can be automated using MIDI continuous controllers - see [MIDI Control and Automation of Synful Orchestra Parameters](#) on page 34.

- expression** — Available in all Playing Modes. The value is 0 to 127 in MIDI controller units with a value of 90 corresponding roughly to a typical *forte* level. The expression control determines the intensity or loudness of the sound as well as affecting the brightness of the timbre. By default this is controlled by MIDI Expression (cc11) and typically is controlled by a keyboard expression pedal or wind/breath control see [MIDI Control and Automation of Synful Orchestra Parameters](#) on page 34. The default output of some keyboard expression pedals is MIDI Volume cc7 and in some keyboards this cannot be changed (usually it can so check your controller documentation). In certain host sequencers (for example Apple Logic) the MIDI Volume (cc7) control is not available to a plugin synthesizer because it is used directly by the Logic Mixer. In this case a different MIDI controller must be used (e.g. cc11) to control Synful Orchestra expression. If you happen to be using a combination of Apple Logic together with a keyboard controller that only maps its expression pedal to cc7 then you have no way to control Synful Orchestra expression using the pedal. You may be able to setup a MIDI Filter inside Logic to map cc7 to cc11 or some other MIDI control that will be sent to the plugin. The Synful Orchestra expression control not only affects the loudness of the instrument but also its timbre -- for example, the Trumpet not only gets louder but also gets brighter in timbre as the expression control is increased. The expression control is somewhat special because it does not change when the Playing Mode is changed. This is so that you can maintain a continuous expression contour when you switch from arco to pizzicato for example. The last setting of expression is always saved with the project.
- gain** — Available in all Playing Modes. The value is in dB, with 0 db being a "normal" value that balances the Playing Mode in a natural way relative to other Programs and Playing Modes. By default there is no MIDI controller mapped to gain, although you can set one. In many situations you may want to set gain to be controlled by MIDI Volume (cc7) and use it as a mixer gain. This is in contrast to expression which is typically a more rapidly changing control used to contour individual notes and phrases, somewhat analogous to the breath of a wind player. expression may also affect the timbre or brightness of the sound which gain does not do. gain is typically more slowly changing and is used to balance the Program with other Programs and Playing Modes like a mixing console slider. In Apple Logic cc7 is used by the host sequencer directly as a mixing control and is not sent to the synthesizer plugin. In this case you can still use gain and map it to some other MIDI controller but you may prefer to simply leave it set to 0 db.
- vibrato depth** — Available only in arco strings and appropriate winds. The value is in arbitrary continuous units from 0 to 1 with 1 meaning the vibrato intensity of the original recording. By default this is controlled by the MIDI Mod Wheel (cc1). This controls the intensity or depth of vibrato of the sound. Synful Orchestra uses the vibrato of the original recorded notes. However, due to the additive synthesis sound representation of Synful's RPM the depth of the original vibrato can be controlled by the user. Vibrato depth has a somewhat special behavior. If you control it from the Control Panel slider then it behaves just like a normal Playing Mode parameter: it changes when you change to a new playing mode and remembers its position in each Playing Mode.

If you control it from an external MIDI controller (usually the Mod wheel) however, then when you stop or start playback, switch playing modes, etc. it returns to the last position you set from the Control Panel slider. This is because we expect that during expressive phrasing you may be moving the mod-wheel allot and the last value you set from the Mod wheel really has no importance for when you start playing again.

- **harmonic tilt** — Available in all Playing Modes. The value is in db with 0 db meaning no tilt, just the original natural sound. By default there is no MIDI controller mapped to harmonic tilt, although you can set one. Part of Synful's RPM technology is the additive synthesis of harmonics - see [RPM Reconstructive Phrase Modeling](#) on page 41. The amplitude or loudness of the individual harmonics of a note is constantly changing with time in a complex manner dependent on Instrument, expression, MIDI velocity and other factors. The harmonic tilt slider allows you to adjust the harmonics so that higher frequency or lower frequency harmonics are emphasized. The value displayed is in dB units and determines the slope that will be applied to the harmonics across frequency. With harmonic tilt at zero no slope is applied so you hear the harmonics of the original Synful Orchestra Instrument. With the value adjusted to a positive value -- for example, 20 dB -- you hear the highest frequency harmonics boosted 20 dB relative to the lowest with a linear ramp determining the level of intermediate harmonics. So a mid-frequency harmonic would be boosted 10 dB relative to the lowest. With the value adjusted to a negative value -- for example, -20 dB -- you hear the lowest frequency harmonics boosted 20 dB relative to the highest frequency. harmonic tilt functions as a kind of EQ or treble-bass control for a Playing Mode.
- **harmonic parity** — Available in all Playing Modes. The value is in db with 0 db meaning no parity setting, just the original natural sound. By default there is no MIDI controller mapped to harmonic parity, although you can set one. The harmonic parity slider allows you to adjust the relative balance of odd and even harmonics. Positive values of harmonic parity boost the odd harmonics, creating a more clarinet like sound. Negative values of harmonic parity boost the even harmonics creating a more "even harmonic" sound. It is fun to play with harmonic parity on a whole range of sounds, string pizz, tremolo, etc are particular favorites.
- **release trim** — Available in winds and arco strings. The value is in db relative to the original sound, so 0 db means the release characteristics of the original recording. By default there is no MIDI controller mapped to release trim, although you can set one. This determines the amount that the high harmonics of the sounds are attenuated as a note is released and decays. The control also affects the perceived release time as more attenuation is perceived as a faster release. The release of the original Synful Orchestra Instrument is complicated and varies for different notes, different harmonics, different Instruments, etc. The release trim slider allows you to influence this release behavior.
- **release time** — Available in all non-arco strings. The value is in seconds with values of .4 to 1 second being typical for a natural effect and larger values being quite valuable for special effects. By default there is no MIDI controller mapped to release time, although you can set one. release time determines how long the sound rings after a note-off. The timbre also changes during the release so that, for example, a bowed string release sounds like the bow has been lifted from the string and the string is ringing. Pizzicato, bartok-pizz and col-legno sounds are particularly transformed by release time. If you select long release times you may want to set release noise level to a low level so as not to interfere with the ringing release sound.
- **release noise level**— Available in pizzicato strings only. The value is in db relative to the original sound, so 0 db means the release noise level of the original recording. By default there is no MIDI controller mapped to release noise level, although you can set one. For certain instruments and Playing Modes there is a particular sound at the end when a note is released. This is particularly true for pizzicato string sounds when the right-hand plucking finger returns to the string to mute it, creating a short somewhat noisy buzz. For single long ringing tones the player may not mute the sound but for any kind of reasonably fast pizzicato passage or even for simple staccato short note playing this buzz invariable occurs and is important to the naturalness of the playing. Synful Orchestra intelligently adjusts the release noise level in a complex manner as a function of intensity and speed of playing but the release noise level allows you to affect this level. If you select a long release time you may want to reduce release noise levelso as not to interfere with the ringing release sound.
- **attack noise** — Available in flute and arco strings. The value is in db relative to the original sound, so 0 db means the attack noise level of the original recording. By default there is no MIDI controller mapped to attack noise , although you can set one. For certain instruments there is a particular noise-like sound that occurs at the attack of a note — a flute chiff or a bow scratch. Synful Orchestra intelligently adjusts the attack noise level in a complex manner as a function of intensity, note separation and other factors but the attack noise control allows you to affect this level.
- **bow noise** — Available in arco, harmonics, tremolo, sul-pont, sul-pont tremolo, and mute strings. The value is in db relative to the original sound, so 0 db means the original bow noise of the recording. By default there

is no MIDI controller mapped to bow noise , although you can set one. There is a sustained whispering noise associated with a bow running across a string. The bow noise control allows you to continually adjust the level of this whisper for arco, sul-pont, harmonic, and mute playing. For tremolo strings the bow noise is combined with the repeated attack noises as the bow constantly changes direction. The bow noise control allows you to transform a tremolo continuously from an edgy gritty tremolo to a smooth shimmer.

- **breath noise** — Available in flute only. The value is in db relative to the original sounds, so 0 db means the breath noise level of the original recording. By default there is no MIDI controller mapped to breath noise , although you can set one. Certain instruments, especially flute, have an audible continuous breath noise that is a characteristic part of the sound. Synful Orchestra intelligently adjusts the breath noise level in a complex manner as a function of intensity, speed of playing, and register. The breath noise control allows you to affect this level.
- **sforzando strength** — Available in tremolo strings only. The value is in arbitrary units between 0 and 1 with 0 being no sforzando attack and 1 being a big sforzando attack. By default there is no MIDI controller mapped to sforzando strength, although you can set one. A sforzando attack at the beginning of a note, especially tremolo notes, is a common instrumental gesture. For tremolo attacks this has two affects on the sound: it makes the beginning of the sound louder; and it makes the first stroke(s) of the tremolo considerably longer than the rest. Controlling the amount of sforzando, especially in tremolo playing, is a key factor in expressive phrasing. Synful Orchestra combines the sforzando strength setting and the Midi velocity value at note-on to determine the actual sforzando strength. With the sforzando strength set to zero there is no sforzando and no affect of velocity on sforzando. As the sforzando strength is increased both the sforzando strength increases and the affect of velocity on the sforzando strength increases. For non tremolo sounds the effects of sforzando can be simulated with suitable use of the expression control. This is not the case for tremolo strings where the length of the first bow stroke(s) must also be changed. Stretching the length of bow strokes in a continuous way is something that is impossible to do with standard sampler technology and is a distinctive feature of Synful's RPM technology.
- **tremolo rate** — Available in tremolo strings only. The value is in arbitrary units from -1 to 1 with 0 being the "normal" tremolo rate of the original recording. By default there is no MIDI controller mapped to tremolo rate, although you can set one. Consider using Mod Wheel (cc1) as a possible choice. tremolo rate controls the speed of the tremolo stretching or shrinking every tremolo stroke in a natural way without affecting other aspects of the sound. Stretching and shrinking the length of individual bow strokes in a continuous way is something that is impossible to do with standard sampler technology and is a distinctive feature of Synful's RPM technology.
- **tremolo rate spread** — Available in tremolo strings only. The value is in arbitrary units from 0 to 1 with a value of 0.2 being typical. By default there is no MIDI controller mapped to tremolo rate spread, although you can set one. When a real string section plays tremolo they are not all playing at exactly the same speed. The tremolo rate spread allows you to control how varied the tremolo speed is across the section. Note that, even with tremolo rate spread set to zero the tremolos for different players in a Synful Orchestra string section are taken from different sample recordings and are still not mechanically in sync.
- **gliss rip** — Available in the new Synful Orchestra French Horn only. Set all the way to the left the control is OFF. If you play a slur (two overlapping notes) with a rising pitch, when the gliss rip control is on, then a gliss rip sound is generated. The position of the control determines the speed of the gliss rip. It is most natural to map this control to a keyboard slider, the Mod Wheel, or a pedal.
- **trill** — Available in the new Synful Orchestra French Horn only. The position of the control corresponds to the speed of the trill. To generate a trill, play a first note and hold it and then play a 2nd note. If the Trill control is on when you play the 2 nd note then a trill will occur between the two notes as long as you hold both notes. The trill can occur between any two pitches with a speed determined by the Trill control position. It is most natural to map this control to a keyboard slider, the Mod Wheel, or a pedal. Example: Play C one octave above middle C. Move Trill control to about .8. Continue holding the C and play D. You should hear the trill.
- **wah** — Available in the new Synful Orchestra French Horn only. When you select a mute or stopped playing mode the Wah control becomes active. The Wah control has the effect of gradually removing the mute or hand from the bell of the horn. When the Wah control is all the way to the left the mute/hand is fully in the bell. As the Wah control is moved to the right the mute/hand is gradually removed from the bell. When the Wah control is all the way to the right the sound is the same as the normal playing mode.

## Program Control Parameters

- **Synful pitch wheel** — Above the keyswitch start display is the small Synful Pitch Wheel switch. There is no MIDI mapping available for this control. This is an on-off check box. Synful pitch wheel determines whether the

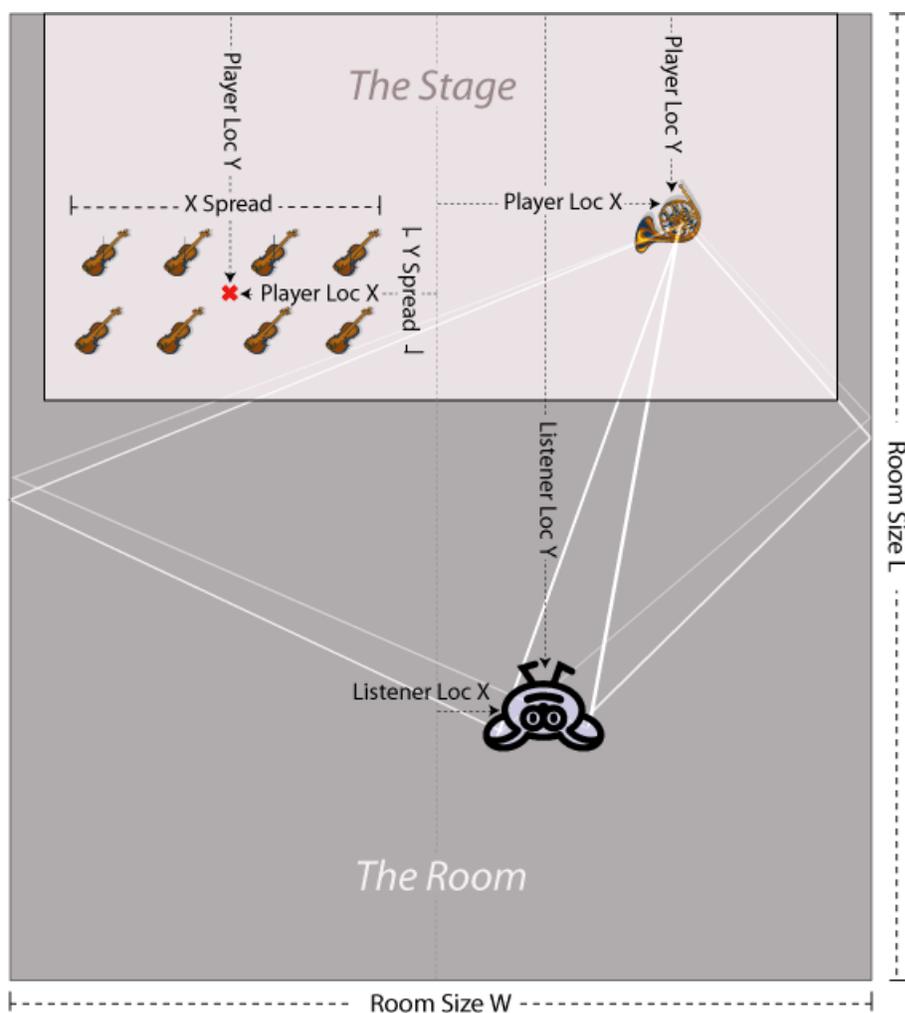
currently selected Program is in normal Pitch Wheel mode or the special Synful Pitch Wheel mode. Synful Pitch Wheel mode is useful for creating realistic pitch slides and portamento for string and wind instruments - see [Synful Pitch Wheel](#) on page 39 for details.

- **pitch wheel range** — This determines how far the pitch wheel bends the pitch when the wheel is all the way up or down. This applies to both normal Pitch Wheel mode and Synful Pitch Wheel mode. The value displayed is in units of half-steps. By default there is no MIDI controller mapped to pitch wheel range, although you can set one. The default value is 4 half-steps. With this setting the pitch is bent up one 4 half steps when the Pitch Wheel is all the way up and down 4 half steps when the Pitch Wheel is all the way down. The center detent position of the Pitch Wheel is always zero pitch bend regardless of pitch wheel range.
- **tune instrument** — The tune instrument slider tunes the individual Program relative to the orchestra tuning applied to the whole synthesizer(s). The displayed value is in cents (1/100 of half step) relative to the orchestra tuning frequency. By default there is no MIDI controller mapped to tune instrument, although you can set one.
- **Player Loc X** — see [Localization and Section Controls](#) on page 27.
- **Player Loc Y** — see [Localization and Section Controls](#) on page 27.
- **midi pan** — see [Localization and Section Controls](#) on page 27.
- **direct** — see [Localization and Section Controls](#) on page 27.
- **2 walls** — see [Localization and Section Controls](#) on page 27.
- **4 walls** — see [Localization and Section Controls](#) on page 27.
- **num players** — see [Localization and Section Controls](#) on page 27.
- **X spread** — see [Localization and Section Controls](#) on page 27.
- **Y spread** — see [Localization and Section Controls](#) on page 27.
- **time spread** — see [Localization and Section Controls](#) on page 27.
- **detune** — see [Localization and Section Controls](#) on page 27.
- **vel spread** — see [Localization and Section Controls](#) on page 27.
- **HT spread** — see [Localization and Section Controls](#) on page 27.
- **mod spread** — see [Localization and Section Controls](#) on page 27.
- **Absorb** — see [Localization and Section Controls](#) on page 27.

## Localization and Section Controls

---

The following drawing will help us to describe the Player Localization and Section Controls available on the Synful Orchestra Control Panel.



**Figure 2: Localization of Sound**

## Room Size Controls

In Synful Orchestra instrument Players and the orchestra Listener can be positioned in a rectangular room whose size can be adjusted.

None of these controls can be adjusted using MIDI controllers. Synful Orchestra only supports MIDI controllers when a parameter can be adjusted in real-time without introducing artifacts in the sound. This is not the case for the Room Size Controls. If you want to change Room Size in real time without introducing glitches in the sound then setup a number of different Programs with different room sizes and switch between them.

**Room Size W** — This control describes the width of the room in meters. The room width can be adjusted from 1 to 100 meters. A typical value for W is 20-40 meters. Room Size W is a Synthesizer Control that affects all the instances of Synful Orchestra Synthesizer in a project. If you change the control in one Synthesizer instance it will also be changed in all the other instances in the project. If the room-size is set to large then you may hear discreet "slap-back" echoes in the 2 and 4 walls modes.

**Room Size L** — This control describes the length of the room in meters. The room length can be adjusted from 1 to 100 meters. A typical value for L is  $1.6 * W$ . Room Size L is a Synthesizer Control that affects all the instances of Synful Orchestra Synthesizer in a project. If the room-size is set to large then you may hear discreet "slap-back" echoes in the 2 and 4 walls modes.

**Room Size H** — This control describes the height of the ceiling in the room in meters. This control is not shown in the drawing above. A typical value for H is  $W/1.6$ . The room height can be adjusted from 1 to 100 meters. Room Size

H is a Synthesizer Control that affects all the instances of Synful Orchestra Synthesizer in a project. If the room-size is set to large then you may hear discreet "slap-back" echoes in the 2 and 4 walls modes.

**golden ratio** — If this box is checked then the Room Size dimensions W, L, and H are forced to the "Golden Ratio" where L is  $1.62*W$  and W is  $1.62*H$ . If you adjust one of the dimensions all of them will change so that these relationships are maintained. The "Golden Ratio" dates back to the Ancient Greeks or before, and is known to have good acoustic properties. golden ratio is a Synthesizer Control that affects all the instances of Synful Orchestra Synthesizer in a project.

**constrain ratio** — If this box is checked then the Room Size dimensions W, L, and H are forced to maintain the current ratios between W, L and H. If you adjust one of the dimensions all of them will change so that the current relationships are maintained. constrain ratio is a Synthesizer Control that affects all the instances of Synful Orchestra Synthesizer in a project. If golden ratio is checked then it doesn't matter if constrain ratio is checked on not.

## Listener Location Controls

When using Room Size and Listener Loc, and Player Loc controls Synful Orchestra produces a stereo output whose left and right audio signals correspond to the signals that arrive at the left and ear right ears of a virtual Listener sitting somewhere in the room.

None of these controls can be adjusted using MIDI controllers. If you want to change Listener Location in real time without introducing glitches in the sound then setup a number of different Programs with different Listener Location settings and switch between them.

**Listener Loc X** — This control determines the distance in meters of the virtual Listener from the center of the room. If the value is negative then the Listener is to the left of center. If the value is positive then the Listener is to the right of center. In the drawing above Listener Loc X is positive since the Listener is positioned to the right of center. Generally it is best to leave Listener Loc X set to zero so that the listener is in the middle of the room. Listener Loc X is a Synthesizer Control that affects all the instances of Synful Orchestra Synthesizer in a project.

**Listener Loc Y** — This control determines the distance in meters of the virtual Listener from the front wall behind the stage. The distance is always positive and a typical value might be halfway back — e.g. 40 meters in a room with  $L=80$ . Listener Loc Y is a Synthesizer Control that affects all the instances of Synful Orchestra Synthesizer in a project.

## Player Location Controls

In Synful Orchestra sound originates from a number of Players who are positioned in the room — generally, but not necessarily, on the virtual Stage near the front of the room.

None of these controls can be adjusted using MIDI controllers. If you want to change Player Location in real time without introducing glitches in the sound then setup a number of different Programs with different Player Location settings and switch between them. If you select midi pan as the Localization Mode then you can use the Midi Pan controller (cc10) to perform typical stereo panning with changes in real-time.

**Player Loc X** — This control determines the distance in meters of the a Player from the center of the room. If the value is negative then the Player is to the left of center. If the value is positive then the Player is to the right of center. In the drawing above the French Horn Player Loc X is positive since the French Horn Player is positioned to the right of center. Unlike the traditional MIDI Pan, which is a MIDI controller associated with a Channel, Player Loc X is associated with a Program and affects only the currently selected Program. Reasons for this are discussed below. We can assign a Program to multiple MIDI channels. For example, we may want to assign a French Horn Program to MIDI channel one and two. However if the Program is the same then the two French Horn Players will have the same location. This is generally a bad idea, so it is better to make two copies of the French Horn Program so that the locations can be separately adjusted. NOTE: it is possible to have multiple Players in one Program corresponding to a Section. See Section Controls for more details.

**Player Loc Y** — This control determines the distance in meters of the a Player from the front wall behind the stage. The distance is always positive and a typical value is generally small reflecting the fact that the stage is generally near the front of the room. Player Loc Y is a Program control with the same considerations described for Player Loc X.

## How Does a Real Listener Determine Where a Player Is Located

A real listener in a room perceives complex acoustic signals arriving at the left and right ears. In the drawing above if the French Horn plays alone then the listener first receives acoustic signals coming directly in straight lines from the player. In the drawing above the sound from the French Horn reaches the listeners right ear slightly before the left ear since the distance to the right ear is shorter. This difference in time is called Interaural Time Difference (ITD). In addition, the signal to the right ear is slightly louder than the left ear largely because the sound is increasingly attenuated over distance. This difference in level is called Interaural Level Difference (ILD).

If the listener was positioned near the front and close to the left wall then the sound to the left ear coming from the French Horn would be considerably softer than to the right ear because the listeners head would block some of the energy — this is called "head shadow" and cause ILD to be large. Shortly after the direct signals arrive at the ears the listener receives acoustic signals that have been reflected off the right wall of the room. These signals (and the signals reflected from the other walls, ceiling, etc.) are referred to as Early Reflections. The early reflections from the right wall arrive at the right ear before they arrive at the left ear. Also the ILD is large due to head shadow. The reflections from the left wall arrive after the reflections from the right wall since the paths to the right wall and back to the listener are longer, and again ILD is large due to head shadow. The listener's perception of the location of the player in the room is largely due to the ITD and ILD associated with the direct signals and early reflections.

When there are multiple players in a room then there are separate ITD and ILD for the direct and early reflections associated with each player. The result is that the individual players have separate perceived locations which not only provides a sense of space but also helps the listener to mentally separate the various parts in a complex musical piece.

## Localization Mode Options

The localization mode options are Program controls. They affect all of the Playing Modes of the currently selected Program and the Player (or Players in the case of a Section) associated with the Program. The localization options other than midi pan generate synthetic direct and early reflection signals with corresponding IAD and ITD for each player. When there are many Players in an orchestra then this can result in considerable computation. The fewer signals that are generated the lighter the computation. midi pan and direct have the same amount of computation. 2 walls requires more, and four walls requires the most computation. While selecting 4 walls provides the best direct and early reflection signals it also decreases the number of available voices because of the added computation. If the room-size is set to large then you may hear discreet "slap-back" echoes in the 2 and 4 walls modes. These options are radio buttons so only one option can be selected for a Program.

The selection of Localization Mode cannot be adjusted using MIDI controllers. If you select midi pan then you can use the Midi Pan controller (cc10) to perform typical stereo panning with changes in real-time. If you want to change Localization Mode in real time without introducing glitches in the sound then setup a number of different Programs with different Localization Modes and switch between them.

**midi pan** — When this box is checked then traditional MIDI pan is used. Traditional panning generates IAD but not ITD so it is a weak and ambiguous way to create a sense of localization for recorded sound. With midi pan selected the Room Size, Listener Loc and Player Loc controls are grayed out since they have no effect. The MIDI pan control (cc10) determines the IAD for the MIDI channel. One advantageous to midi pan is that it can be swept in real-time. The Room Size, Listener Loc, and Player Loc controls cannot be swept in real-time due to high computation requirement.

**direct** — When this box is checked then Synful Orchestra generates the only left and right ear direct acoustic signals corresponding to each Player. This provides effective localization but lacks the richness of 2 walls or 4 walls.

**2 walls** — When this box is checked then Synful Orchestra generates the direct signals and the early reflections coming from the left and right walls corresponding to each Player. This provides greater complexity and better localization than the direct option but requires somewhat more computation.

**4 walls** — When this box is checked then Synful Orchestra generates the direct signals and the early reflections coming from the left, right, front, back walls and ceiling corresponding to each Player. This provides the best complexity and localization at the cost of greater computation.

**IMPORTANT NOTE FOR LOGIC USERS:** If you use Apple Logic and want to use the Synful Orchestra direct, 2 walls, or 4 walls localization options then you must set the MIDI pan in Logic to the center or to off. This is because Logic performs panning itself which will interfere with the more powerful Synful localization modes.

## Absorption Control

When sound is reflected from a wall part of the energy is absorbed. The amount depends on the material the wall is made from — velvet wall paper absorbs much more sound than bathroom tile.

**absorb** — This control determines the amount of attenuation in decibels for early reflection signals due to different wall materials. Typical values are -2 to -6 dB. This effects the currently selected Program only. absorb cannot be adjusted using MIDI controllers. However, you can safely change it in real-time from the Control Panel without introducing glitches in the sound.

## Section Controls

Synful Orchestra synthesizes sections. For example a violin section with ten players is created from ten separate synthesized violins rather than from a recording of an actual violin section. This allows great flexibility in determining how the violin section sounds and behaves. Of course, the players in a real violin section don't all sound exactly the same — this is what gives the section its richness. Synful generates this richness in a highly controllable way using the section controls. All the section controls apply to the currently selected Program only.

Regardless of the settings of section controls on the Control Panel, Synful Orchestra guarantees certain kinds of performance variation across the section. As described in *RPM Reconstructive Phrase Modeling* on page 41, Synful Orchestra picks phrase fragments — note transition, sustains, etc — from a database containing many phrases. When synthesizing sections Synful Orchestra guarantees that at any given moment each Player in the section plays a different phrase fragment from the database. If there are ten players in a violin section and they all play a slur from C to D then this transition is created by ten different slur transition selected from the database. This is a key feature of Synful's section synthesis. During a note sustain each Player plays a different sustain segment from the database. This results in a different, unsynchronized vibrato pattern for each player. This is another essential feature for synthesis of realistic sounding sections.

**num players** — This control determines the number of Players in a section. For example, to create a Ten Violin Section Program make a copy of the basic Violin Program, rename it to something like Ten\_Violins and then set num\_players to 10. If you play a chord that has more notes than the number of players then Synful Orchestra will temporarily add the necessary number of players to fill the entire chord. In the special case where number of players is set to 1, Synful Orchestra will interpret chords on a string instrument as if they are multiple-stops. However, Synful Orchestra will not automatically arpeggiate the multi-stop. You have to do that in your playing or sequence. num\_players cannot be adjusted using MIDI controllers. If you want to change num\_players in real time without introducing glitches in the sound then setup a number of different Programs with different num\_players and switch between them.

**X spread** — The Players in a section are spread out in space over the stage. In the drawing above there is an Eight Violin Section with the Players spread out in the X and Y directions. The X spread control determines the spread in meters in the X direction of the Players in the section. The players are spread out around a center point. In the drawing above the red x mark shows the center point for the violin section. The center point is determined by the Program's Player Loc X and Player Loc Y controls. When the localization mode for the Program is Direct, 2 Walls, or 4 Walls then Synful Orchestra generates the IAD and ITD signals for the left and right ear of the listen for each Player in the Section. This is key to generating the sense of spatial distribution for the players. X spread can also be used with midi pan mode in which case the panning is adjusted differently for each Player to simulate something approaching a spatial distribution although the effect is not as effective as with the other localization modes. In the case of midi pan mode the control is in arbitrary units going from the min to max values 0 to 100 of the X spread control. X spread cannot be adjusted using MIDI controllers. If you want to change X spread in real time without introducing glitches in the sound then setup a number of different Programs with different X spread and switch between them.

**Y spread** — The Y spread control determines the spread of Players in meters in the Y direction around the center location. The center location is specified by the Program's Player Loc X and Player Loc Y controls. The control has the same considerations as Y spread. However, Y spread is disabled in midi pan mode. Y spread cannot be adjusted using MIDI controllers. If you want to change Y spread in real time without introducing glitches in the sound then setup a number of different Programs with different Y spread and switch between them.

**time spread** — Players in a section do not play rhythmically in perfect sync. Notes begin and end at slightly different times. The time spread control determines the amount of random variation in rhythmic sync across the section. Value is in milliseconds however the value random variation are generally somewhat large for long notes

then for short notes. Typical values are 60 - 100 milliseconds. time spread cannot be adjusted using MIDI controllers. However, you can safely change it in real-time from the Control Panel without introducing glitches in the sound.

**detune** — Players in a section due not play perfectly in tune. The perceived detuning generally increases with pitch so a violin section playing in the high range of the instrument sound more out of tune then when playing in the low range. The detune control determines the amount of random detuning across the section. The value is in cents — 1/100 of a half-step — referenced to the mid-range of the instrument. Note that the unsynchronized vibratos discussed above will contribute to a sense of detuning even when the detune control is set to zero. detune cannot be adjusted using MIDI controllers. However, you can safely change it in real-time from the Control Panel without introducing glitches in the sound.

**vel spread** — This control spreads out MIDI Velocities across the section. If the note-on MIDI velocity is 70 and the vel spread is 10 then the various Players in the section will receive MIDI velocities randomly distributed in the range 65 to 75. vel spread cannot be adjusted using MIDI controllers. However, you can safely change it in real-time from the Control Panel without introducing glitches in the sound.

**HT spread** — This control spreads out Harmonic Tilt across the section in order to accentuate timbral variation. If the Program Harmonic Tilt control is set to -5 and HT spread is set to 20 the various Players in the section will have Harmonic Tilts randomly distributed in the range -15 to 5. HT spread cannot be adjusted using MIDI controllers. However, you can safely change it in real-time from the Control Panel without introducing glitches in the sound.

**mod spread** — This control spreads out MIDI Modulation Control (cc1) across the section. If MIDI mod is currently 75 and the mod spread is 20 then the various Players in the section will receive MIDI mod controls randomly distributed in the range 65 to 85. mod spread cannot be adjusted using MIDI controllers. However, you can safely change it in real-time from the Control Panel without introducing glitches in the sound.

## CPU Power for Sections

Synful models each player in a section as a separate Synful Orchestra RPM voice. This is different from a sampler where, for example, an entire violin section is recorded and saved as a single sound file that then represents a single sample. This approach provides maximum flexibility for the user. The user does need to set up the individual Synful Orchestra voices — that is done automatically. However, all the voices require CPU power. A 1.5 GHz laptop — PC Pentium IV or MAC G4 — is able to generate a reasonable size orchestra completely in real-time. For example the included Wagner Tristan und Isolde example, with 18 violins, 8 violas, 8 celli, 7 bass, and assorted winds is able to run in real-time on these laptops. However, increasing the number of strings to 32 violins, 16 violas, 16 celli, 16 bass may cause the synthesis to stall on these laptops due to insufficient CPU power. A more powerful desktop or server computer, especially with multiple CPUs or CPU cores, willsupport larger real-time orchestras.

## What Happens When the CPU Overloads?

This depends on the host sequencer. In Apple Logic and Digital Performer running the AU Synful Orchestra plugin, a polite message appears saying the system has been overloaded. Nothing bad happens, you're work is not lost, however the audio stops and you need to hit play again. In Cubase the audio may mysteriously stop. Generally what has happened is that the Synful Orchestra VST plugin has automatically gone into Bypass mode. To get out of this first wait a few seconds — the condition may fix itself. If that doesn't work then go to the Synful Orchestra Control Panel. In the upper left are two buttons that look like this . The left button is the plugin enable button. The right button is the plugin bypass button. If the plugin has gone into bypass first click on the bypass button to remove the plugin from the bypass state. Then click on the enable button twice -- first click turns off the plugin, second turns it back on. In very severe overload situations the system may hang up completely and you will need to power off and restart the computer. In Sonar an overload may crash Sonar and you will have to reopen the program.

## But I Need to Use a BIG ORCHESTRA!

There are several things you can do to get around the CPU overload problem:

- Use a more powerful computer.
- Freeze Tracks — all of the modern sequencers have the ability to "freeze" tracks. This is basically a very convenient way to turn the track temporarily into a recording rather than a live synthesized track. Check the documentation for each sequencer for how to do this. You normally freeze tracks that you are not currently

working on since you cannot edit the midi or other parameters on a track while it is frozen. Of course, you should freeze tracks with many players if possible.

- **Temporarily Reduce the Number of Players** — While you are working on tracks you can reduce the number of players in some sections temporarily. Then when you're ready to produce the final output, increase the number of players and export to an audio file. The export operation will not overload the CPU.
- Use a combination of the techniques just mentioned.

## Performance is Better with Delay for Expression ON

The danger of CPU overload is greater when Delay for Expression is OFF for a number of tracks. Generally we suggesting having DFE on for most tracks except actively recording on a track. This not only improves performance but also makes the music sound better! In version 2.3 of Synful Orchestra we did considerable work to improve CPU usage so that sections work well on computers with moderate CPU power. We will no doubt make further improvements in performance in the future — and of course the CPUs will also get faster.

## Natural Sounding Divisi

Synful Orchestra sections simulate realistic and natural divisi. For example, suppose MIDI channel one has its Program set to Ten\_Violin, a ten violin section Program you have created. If you play a monophonic line on channel one then all players will play that line with appropriate variations as described above. If you suddenly play a ten note chord on channel one then each Player will play one note in the chord. This creates a much more natural effect than a sampled violin section in which a ten note chord may sound like it's being played by 100 players. The Synful Orchestra section automatically adapts to the changing polyphony of the part. If you play 3 note chords or polyphony on channel one then the ten Players distribute themselves automatically 4-3-3 across the notes. If you return to monophonic playing the all ten players return to the same notes. If you have selected Ten\_Violin and you play an 11 note chord then Synful Orchestra will add the necessary number of players, temporarily, to fill the entire chord.

## Why are Some Controls Disabled

If Localization Mode is set to midi pan then the Room Size, Listener Loc, and Player Loc controls serve no purpose so they are disabled and grayed out. Also when num players is set to 1 then the various spread controls — X spread, Y spread, etc. — serve no purpose and are grayed out. Other examples of this kind of disabling will occur in appropriate circumstances.

## Why Do Some Controls Change When I Move Other Controls?

If Room Size Y is set to 80 meters then Listener Loc Y, Player Loc Y must be less than 80 meters. Adjusting the Room Size Y control will force these and other control to stay within the room. If Room Size W is 20 and Player Loc X is 7 then X Spread is forced to be less 6 because otherwise Players would be beyond the right wall of the Room. These and other similar constraints are enforced as you alter the localization and section controls.

## How Does Synful's Localization Control Work With Reverb Plugins

Synful Orchestra's Localization controls are not a substitute for a good Reverberator. A high quality reverb generates early reflections followed by the reverb tank made up of very many closely spaced decaying reflections. Synful Orchestra generates the direct signals and first few early reflections. The important contribution of Synful Orchestra's early reflection generator is that it generates a different set of reflections for each player. This greatly enhances the sense of spatial separation and imaging associated with the orchestra. You still need to feed the stereo output of Synful into a quality reverb plugin to create the impression of a complete listening space.

## I Set a Player's Location Behind the Listener But it Didn't Sound Like it Was Coming From the Back

Synful Orchestra's direct signals and early reflections are very effective in positioning a Player to the left or right of the Listener. They are much less effective at positioning the Player in front or in back of the Listener, or above or below the Listener. These positioning effects require a greater level of signal processing and computation that involves modelling the shape of the Listener's ears, the shape of the head, etc. This is of limited value for

general orchestral listening and so is not implemented in Synful Orchestra. Specialized spatialization plugins are commercially available to create these kinds of effects. In a real auditorium the sense of distance from the player is largely determined by the relative levels of the direct and early reflections compared to the level of the reverb tank. By adjusting the wet/dry level of your reverb plugin you can easily change this balance and better control the Listener's distance from the orchestra.

## MIDI Control and Automation of Synful Orchestra Parameters

---

Many of the control parameters in the Synful Orchestra Control Panel can be controlled by MIDI continuous controllers. When this is the case then there is a midi cc display to the right of the parameter slider. The midi cc display shows the name of the continuous MIDI controller that is currently set to control the Synful Orchestra parameter. When the name displayed is OFF then midi control of the parameter is disabled.

Continuous Midi Controllers are numbered from 0 to 127. Some of the control numbers have been historically used for a specific purposes and so they also have names. For example cc11 is commonly used for MIDI Expression, cc7 for MIDI Volume, etc. When you click on the downward triangle in the midi cc display a pop-up list shows commonly named Midi Controllers next to their MIDI controller numbers. Selecting a name/number from the list assigns it to control the Synful Orchestra parameter. There is no law that says you need to use MIDI expression cc11 to control Synful Orchestra expression but it is common practice and may make it easier to map control pedals on a keyboard controller to Synful Orchestra expression. If you use a wind controller then you would probably map Breath cc2 to Synful Orchestra expression.

In addition to the named controllers in the pop-up list you can type in any number between 0-127 to set the MIDI controller. You can also type in the abbreviated name if you desire.

By default Synful Orchestra maps MIDI Expression cc11 to Synful Orchestra expression and Mod Wheel cc1 to Synful Orchestra vibrato depth when it applies.. Other than that Synful Orchestra leaves the midi cc mapping OFF and lets you select what you want.

Some of the names on the midi cc pop-up list are not standard names of MIDI controllers but are names that were used in previous revisions of Synful Orchestra. This is true for all the controller names having numbers from 102-111.

There are some reserved midi controller numbers that are not available for mapping to parameters. These are MIDI Pan cc10 which is reserved for use as traditional midi pan when the Localization Mode is set to midi pan. cc0 and cc32 are also reserved for use as bank numbers.

## Renaming, Creating and Deleting Programs

---

You can rename a Program by double clicking on its name in the Channel Grid and typing a new name. When you rename a program you actually make a copy of it and give that copy the new name. So you can select Violin and type the name WeirdVln and you have just created a new program with that name. The old Violin Program still exists. The new Program becomes the currently selected Program for the channel. The new Program also becomes available on the pop-up Program list and can be assigned to any channel on any synthesizer instance in the current project. When the Program is copied it is copied with all its Playing Modes and with all Program and Playing Mode parameter settings. If you change any of these settings in the new Program the changes do not affect the settings of the original Program.

Clicking on the "delete program" button just above the MIDI keyboard in the Control Panel deletes the Program on the currently selected channel and deletes it everywhere on any channel in any synthesizer in the current project. The program no longer exists. Anywhere where the program was assigned to a channel a new default program is assigned to that channel. You cannot delete a "factory" program, only ones created by you.

The purpose of creating a new Program is to adjust control parameters differently from the original program. For example, you may create a new violin Program and edit the attack noise level of the new program so that the bow scratch on attack is louder than the original. Then in a sequence you might insert Program changes to change from the original violin to the new violin for certain phrases.

## Load Store Default and Local Channel Programs

---

Often you may want to set several different channels to the same program and then adjust the parameters of these programs differently. For example you may have three solo violins assigned one each to MIDI channel 1,2, and 3. Then you may want to adjust player loc, or other parameters differently on different Program/channels.

When you assign a Program to a channel in Synful Orchestra (see [Assigning Programs to Channels and Selecting Playing Modes](#) on page 20) then, in fact, a new copy of that program is made and assigned to the channel as a local copy. If you assign the same program to another channel then that channel gets its own local copy of the Program. You can change parameters on one of the channels without affecting the settings on the other channel. If you want to see the other channel settings changed to the same values (we generally think this is less likely) then you can click on the "save default" button near the upper right of the Control Panel while the desired Program/channel is selected, then switch to the other channel and click "load default".

After clicking "save default" any time you load that Program into any other channels on any synthesizer instance in the project you will get the settings that were in place when the "save default" was clicked. Clicking on "save default" for factory program has no affect. So if you want to save a default Violin different from the one that came from Synful then rename it to something like MyViolin, make your changes, and save that as the default.

## Saving and Loading Programs Banks and Presets

---

When you save a host synthesizer project all of the settings for all of the Synful Orchestra synthesizers, channels, Programs, and Playing Modes are saved. In addition, depending on the sequencer you are using, you can save Synful Orchestra synthesizers and or Programs in separate files so that you can load your favorite setups into new sequencer projects. Here is how it works for some sequencers. Even if you are not using one of these sequencers you can still benefit from following one of these links.

### Logic

#### Saving and Loading a Setting

In Logic saving or loading a Setting saves the state of all synthesizer instances -- including all Programs and Playing Modes, Channel Program assignments, control parameters, etc.. To save to a new Setting in Logic click on the downward triangle in the upper left just above the Control Panel and beneath the Bypass button. In the drop down menu select Save Setting As. In the Save Setting As dialog type the name of the new setting and browse to the location of the setting or preset file. Then click Save.

To load a previously saved preset click on the downward triangle and select Load Setting. In the Open dialog browse to the location of the desired setting or preset file and click Open. When a new setting is loaded the state of the Synthesizer is overwritten by the contents of the preset. This includes all Programs in the Synthesizer Bank, Program assignments to Channels, Synthesizer parameters, etc.

#### Saving and Loading a Project

When a Project is saved or loaded in Logic the entire state of each Synthesizer in the Project is saved or loaded in addition to all MIDI sequence information. This includes all Programs in the Synthesizer Banks, Program assignments to Channels, Synthesizer parameters, etc.

### Digital Performer

#### Saving and Loading a Preset

In Digital Performer saving or loading a Preset saves the state of a Synthesizer -- including all Programs in the Synthesizer Bank, Program assignments to Channels, Synthesizer parameters, etc. To save the Synthesizer state to a new preset in Digital Performer click on the small menu icon in the upper left just above the Control Panel. In the drop down menu select save settings. In the save settings dialog type the name of the new preset and click OK.

To load a previously saved preset click on the small menu icon in the upper left at the top of the Control Panel and select User Presets. Then select the desired preset. When a new preset is selected the state of the Synthesizer is overwritten by the contents of the preset. This includes all Programs in the Synthesizer Bank, Program assignments to Channels, Synthesizer parameters, etc.

### **Saving and Loading a Project**

When a Project is saved or loaded in Digital Performer the entire state of each Synthesizer in the Project is saved or loaded in addition to all MIDI sequence information. This includes all Programs in the Synthesizer Banks, Program assignments to Channels, Synthesizer parameters, etc.

## **Cubase**

### **Saving and Loading a Bank**

In Cubase saving or loading a Bank saves the state of a Synthesizer -- including all Programs in the Synthesizer Bank, Program assignments to Channels, Synthesizer parameters, etc. Saving a Bank in Cubase creates a special .fxb file. This file can then be loaded into any Synful Orchestra Synthesizer.

To save a Cubase Bank, open the Synful Orchestra Control Panel. Left click on the small floppy disk icon to the right of the Presets text box. This text box is found at the top of the Control Panel and will say "Synful Orchestra" if no Presets have been saved or loaded. A pop-up list of save-load options appears. Select "Save Bank". A dialog box appears prompting for the name of an .fxb file. Follow the instructions in the dialog box.

To load a Cubase Bank, left click on the floppy disk icon and select Load Bank from the pop-up list. A file select dialog box opens. Browse to the desired .fxb file and select it.

### **Saving and Loading a Instrument**

In Cubase the term "Instrument" means "Program" as we define it in Basic Concepts. In Cubase you can save or load an individual Synful Orchestra Program (or Cubase "Instrument") in a separate .fxp file. This saves only the currently selected Program.

To save a Synful Orchestra Program (or Cubase "Instrument"), open the Synful Orchestra Control Panel. Left click on the small floppy disk icon to the right of the Presets text box. This text box is found at the top of the Control Panel and will say "Synful Orchestra" if no Presets have been saved or loaded. A pop-up list of save-load options appears. Select "Save Instrument". A dialog box appears prompting for the name of an .fxp file. Follow the instructions in the dialog box.

To load a Synful Orchestra Program (or Cubase "Instrument"), left click on the floppy disk icon and select Load Instrument from the pop-up list. A file select dialog box opens. Browse to the desired .fxp and select it. Loading a Synful Orchestra Program overwrites the currently selected Program with the contents of the .fxp file. Only the currently selected Program is overwritten. No other parts of the Synthesizer are changed. The name of the currently selected program is also overwritten with the Program name stored in the .fxp file (this is not the file name but the name given to the Program when it was created).

### **Saving and Loading a Preset**

In Cubase, saving or loading a Preset is the same as saving or loading an individual Synful Orchestra Program (or Cubase "Instrument"). However, the Program is not saved or loaded to/from a separate .fxp file. The Program Name is added to the presets list.

To save a Preset open the Synful Orchestra Control Panel. Left click on the small floppy disk icon to the right of the Presets text box. This text box is found at the top of the Control Panel and will say "Synful Orchestra" if no Presets have been saved or loaded. A pop-up list of save-load options appears. Select "Store Preset". A dialog box appears prompting for the name of a preset. Type in the preset name -- it may be new or not. Click OK. If the preset name already exists you will be prompted to overwrite the existing preset.

To load a Preset, left click on the floppy disk icon. At the bottom of the pop-up Preset list is a list of previously stored Presets. Select one of them. The currently selected Program is overwritten by the Preset. No other parts of the Synthesizer are changed. The name of the currently selected program is also overwritten with the Program name stored in the Preset (this is not the preset name but the name given to the Program when it was created).

A Cubase Preset can be deleted by first selecting it and then clicking on the floppy disk icon and selecting "Delete Preset".

### **Saving and Loading a Project**

When a Project is saved or loaded in Cubase the entire state of each Synthesizer in the Project is saved or loaded in addition to all MIDI sequence information. This includes all Programs in the Synthesizer Banks, Program assignments to Channels, Synthesizer parameters, etc.

## **Sonar**

### **Saving and Loading a Preset**

In Sonar saving or loading a Preset saves the state of a Synthesizer -- including all Programs in the Synthesizer Bank, Program assignments to Channels, Synthesizer parameters, etc. To save the Synthesizer state to a new preset in Sonar left click in the text box next to the word "Presets:" at the top of the Synful Orchestra Control Panel. Type the name of the new preset. Then left click the floppy disk icon to the right of the text box.

To select a previously saved preset, left click on the downward triangle to the right of the Presets text box. A list of previously saved presets appears. Select the desired preset. When a new preset is selected the state of the Synthesizer is overwritten by the contents of the preset. This include all Programs in the Synthesizer Bank, Program assignments to Channels, Synthesizer parameters, etc.

To save the current Synthesizer state to an existing preset you must first select the existing preset as described above, then modify the state of the Synthesizer, then left click the floppy disk icon.

There are no separate preset files directly accessible to the user in Sonar. All access to individual presets is through the pop-list described above.

### **Saving and Loading a Project**

When a Project is saved or loaded in Sonar the entire state of each Synthesizer in the Project is saved or loaded in addition to all MIDI sequence information. This includes all Programs in the Synthesizer Banks, Program assignments to Channels, Synthesizer parameters, etc.

## **Delay for Expression**

---

Just below the Channel Grid in the Synful Orchestra Control Panel is the delay for expression check-box. At the lower right of the delay for expression checkbox is a smaller all chans/synths checkbox. When all chans/synth is checked your delay for expression selection applies to all channels on all active synthesizes in the current project. When it is unchecked your delay for expression selection applies to the current channel only.

### **What is Delay for Expression?**

When delay for expression is on, a one second delay is inserted at the MIDI inputs of the Synful Orchestra Synthesizer. This delay allows Synful Orchestra to examine the incoming MIDI data and make more sophisticated and expressive synthesis choices.

### **When Should Delay for Expression Be Used?**

delay for expression should be used when playing MIDI sequences with Synful Orchestra. If you are playing live then turn delay for expression off.

### **Using Delay For Expression While Overdubbing, or With Other Synthesizers, or With Time Code**

A common method of working is to lay down successive tracks live from a MIDI controller. In this case you may want to have delay for expression on for all tracks/channels except for the current track/channel you are playing into live. To do this select the channel your are currently recording on. Set all/chans synths on and then set delay for

expression on — this sets delay for expression on for all synths/channels/tracks. Then switch all chans/synths off and set delay for expression off. This turns delay for expression off for the current track. There is no Midi Control available for delay for expression.

If delay for expression is on and a one second delay applied to the tracks that use the Synful Orchestra Synthesizer, synchronization with other non-Synful Orchestra tracks or other Synful Orchestra Synthesizers that have delay for expression off will be lost: The Synful Orchestra Synthesizer tracks will be delayed one second relative to the other tracks. The best way to compensate for this is to mix the delayed (Synful Orchestra) and non-delayed (non-Synful Orchestra) tracks to different audio busses. Then add a one second audio-delay effect to the output of the non-delayed tracks (see host program documentation for how to insert audio delay effects).

Of course, all of this is rather cumbersome. You should first check to see if there is any advantage to using delay for expression for the particular material being synthesized. Just listen to the tracks with delay for expression on and off and see if there's a noticeable difference. Often the result with delay for expression off is excellent. In terms of work flow, a good way to proceed is to leave delay for expression off until the final output mix. All of the process of composing and overdubbing tracks can be done with delay for expression off so that the time delay is not an issue.

Hope for the Future! Currently the VST, DXi, AU, and RTAS plugin standards do not provide a way for plugin synthesizers to directly interrogate the future content of MIDI sequences while they are being played. Until Synful Orchestra there has never been a need for this kind of sophistication. In the future this feature will undoubtedly be added to sequencers and then the delay for expression switch will no longer be needed and the benefits will be completely transparent to you.

NOTE: In general one cannot compensate for the delay for expression delay by sliding tracks in a project window because while the track is offset in time the tempo track which applies to the entire project is not offset in time, therefore the offset tracks will have the wrong tempo for one second whenever there is a tempo change.

## Questions and Answers About Delay for Expression (DFE)

Questions have been asked by many people about DFE. "Will DFE be unnecessary when more computer power is available? ", "Will DFE go away with better algorithms from Synful?" "Why does Synful need DFE when no other synthesizer/sampler has this?" "With DFE turned off does Synful sound bad or lose all its expressivity?"

- **Why does Synful have DFE?**

When a wind or string player plays two connected notes — e.g. a legato transition from C to D — he/she generally changes the end of the first note in anticipation of the transition to the second note. This occurs partly because of the physical human interface with the instrument — e.g. the need to slow down the bow before beginning to move it in the opposite direction — and partly as a deliberate part of expressive phrasing. This anticipation may occur 0.030 to .250 seconds or more before the next note begins. With DFE this is exactly what Synful does: it changes the end of one note in anticipation of the transition to a second note.

- **Why is there a difference between playing from a sequence and playing live?**

There is a problem when playing live from a keyboard: when holding one note — say C — there is generally no indication from the MIDI keyboard of when a second note will occur — say D — until the new note-on is received. At this point it is too late to do anything about the ending of the C — except on a very small time scale e.g. < .030 seconds. When playing from a sequence however it is possible to look into the future and know in advance when a new note will occur and how long it will be. This allows Synful to change the end of the first note. Synful will also adjust the beginning of the second note based on various information such as the pitch-interval of the note transition, the length and intensity of the second note, etc.

- **Why Doesn't Synful just look at the sequence if it is available and if there is no sequence (i.e. playing live) then just do the best it can? Why is the annoying one second delay required?**

Currently there is no way for a plugin (VST, DXi, AU, etc) to interrogate a host sequencer about the future of a sequence. However, this is a small technical detail and can be resolved when the host sequence program provides this ability. Which popular sequencer will be the first to provide this ability (Logic? Cubase? Digital Performer? Sonar?). A special Award for Synfulness will be presented to the first sequencer company that provides this ability! When this ability is provided the DFE button will go away. However, this will not change the fact that Synful will still do a better job of expressive phrasing when playing from a sequence than when playing live. This is a fact of life that results from the fundamental lack of information coming from a keyboard.

- **Why don't other synthesizers/samplers require a DFE button or something similar?**

No other synthesizer/sampler, to our knowledge, attempts to model note transitions as realistically as Synful. In particular, no other synthesizer/sampler attempts to model the ending of a first note in anticipation of a second connected note — or at least not in the fairly long 0.1 to 0.250 seconds time range. Any synthesizer/sampler that attempts to model this kind of expressivity will need to deal with the same issues that Synful has already confronted.

- **Does Synful Orchestra sound bad or become just another average sampler when playing live and DFE is turned off?**

No. Synful continues to use considerable intelligence in modeling note transitions even when DFE is turned off. The intensity of attack, the amount of transient noise, the shape of the transition over a short time range — e.g. @ .030 seconds — the dependence of the shape of the attack based on how much separation there is between the preceding note and the following note, the duration of the preceding note, the pitch interval, all these factors are taken into consideration when Synful models note transitions, even when playing live and DFE is off. This is certainly more than any other current synthesizer attempts to do. So the answer is — when playing from a sequence with DFE on, Synful note transitions sound very much more expressive than any other synthesizer/sampler. With DFE off playing live, Synful note transitions sound only much more expressive than any other synthesizer/sampler.

- **How are the two modes DFE on and DFE off used in practice?**

When playing a live concert performance DFE is turned off. When recording DFE is turned off while laying down live tracks. DFE is turned on when listening back to the tracks and when making adjustments to velocities, volume and modulation controls, and note durations to improve phrasing. When using Synful together with other plugin synthesizers/samplers and live recorded tracks, one second delay effects are added to the output of these tracks. These delays are turned on and off together with DFE when recording and listening back. When the sequencer companies provide the ability to interrogate the future of a sequence and the DFE button goes away, these delays will no longer be necessary.

## Synful Pitch Wheel

---

### Portamento in Real String and Wind Playing

Standard MIDI synthesizers provide two controls that support pitch-slides: MIDI Pitch-Wheel and Midi Portamento Control. Suppose the performer wants to generate a slide-step-slide portamento transition as described above using the standard pitch-wheel. The performer can effectively perform the initial pitch-slide that occurs at the end of the first note. However, when the second note is struck the pitch jumps to the keyboard pitch plus the current value of the pitch wheel. If the performer quickly releases the pitch-wheel after striking the second note an undesirable quick slide back to the target pitch occurs. Alternatively, if the performer releases the pitch-wheel just prior to striking the second note there is an equally undesirable bend back to the first pitch before jumping abruptly to the second pitch. No matter how quickly and precisely the pitch-wheel is controlled the desired slide-step-slide curve is impossible to achieve.

The MIDI portamento control also does not support the generation of slide-step-slide pitch curves. The pitch shapes are deficient in two ways: first, the pitch slide does not begin until the second note is played on the keyboard; second, the pitch slides continuously from the first note pitch to the second note pitch — there is no step. Both MIDI portamento control and standard MIDI pitch-wheel curves are unacceptable.

### The Problem with MIDI Pitch-Wheel and MIDI Portamento

Standard MIDI synthesizers provide two controls that support pitch-slides: MIDI Pitch-Wheel and Midi Portamento Control. Suppose the performer wants to generate a slide-step-slide portamento transition as described above using the standard pitch-wheel. The performer can effectively perform the initial pitch-slide that occurs at the end of the first note. However, when the second note is struck the pitch jumps to the keyboard pitch plus the current value of the pitch wheel. If the performer quickly releases the pitch-wheel after striking the second note an undesirable quick slide back to the target pitch occurs. Alternatively, if the performer releases the pitch-wheel just prior to striking the second note there is an equally undesirable bend back to the first pitch before jumping abruptly to the second pitch. No matter how quickly and precisely the pitch-wheel is controlled the desired slide-step-slide curve is impossible to achieve.

The MIDI portamento control also does not support the generation of slide-step-slide pitch curves. The pitch shapes are deficient in two ways: first, the pitch slide does not begin until the second note is played on the keyboard; second, the pitch slides continuously from the first note pitch to the second note pitch – there is no step. Both MIDI portamento control and standard MIDI pitch-wheel curves are unacceptable.

## The Synful Pitch Wheel Solution

Synful Orchestra introduces a special new Synful Pitch Wheel mode to solve the problems just described. The Synful Pitch Wheel mode can be turned on or off separately for each Program. Use the Synful Pitch Wheel switch found to the right of the Program Select box in the Synful Orchestra Control Panel. When the Synful Pitch Wheel switch is off (gray) the pitch wheel behaves like a standard MIDI pitch-wheel. When the Synful Pitch Wheel switch is on (red) Synful Pitch Wheel mode is entered for the program. The description of the special Synful Pitch Wheel mode sounds complex when described in words, but the actual behavior of the Synful Pitch Wheel is intuitive and natural. You may choose to turn on Synful Pitch Wheel mode and start playing with it from a keyboard before reading the following description.

The special Synful Pitch Wheel mode is "context sensitive". When a first note is struck and then the pitch-wheel is displaced the pitch of the first note is "bent" in the standard manner corresponding to the position of the pitch-wheel. However, when a second note is struck, if there is little or no intervening time between the release of the first note and the strike of the second note then the response to the pitch-wheel during the second note is disabled. As a result, if the pitch-wheel is smoothly displaced in the direction of the pitch of the second note but prior to the strike of the second note, then the pitch slides in the direction of the second note. Then when the second note is struck the pitch steps in the direction of the second note.

If the pitch-wheel is released just prior to the strike of the second note then the pitch steps to exactly the pitch of the second note when the second note is struck. In this case a slide-step curve is generated. If the pitch-wheel is not released prior to the strike of the second note then the pitch steps part way to the pitch of the second note and then begins a slide up to the final second note pitch after the second note is struck. This is the slide-step-slide curve discussed above.

If you bend the pitch-wheel while no keys are being held, and then play a key, the note will be immediately bent as in standard pitch-wheel mode. You can then gradually release the pitch wheel to create a normal attack with slide pitch bend effect.

Synful Pitch Wheel behaves naturally and intuitively whether playing single lines or chords and polyphony. However, there is one special case you should be aware of when playing polyphony. If you play a chord and then bends the pitch all notes of the chord will be bent. Then if you change one note of the chord while not releasing the pitch-wheel, the other notes of the chord will be cut off. For this kind of polyphonic playing it may be better to turn Synful Pitch Wheel mode off.

## Tips for Good Phrasing

---

The Synful RPM technology used in Synful Orchestra (see [RPM Reconstructive Phrase Modeling](#) on page 41) is very sensitive to MIDI controls -- Pitch, Expression or Volume, Velocity, Modulation, Pitch-Wheel as well as separation between notes, and duration of notes. Different phrase fragments are selected from the RPM database depending on these controls. In addition to different phrase fragments the timbre, nature of vibrato, loudness of noise elements, and other factors are also dependent on these controls.

If you are unsatisfied with the results of a synthesized phrase it is generally possible to change it by changing these controls. The most natural way to achieve good phrasing is by playing live in real-time. However, it is also quite possible to achieve good phrasing by editing controls manually.

Here are some Synful Orchestra phrasing pointers:

- Overlapped notes result in slurred phrasing unless the velocity is very high.
- If notes are overlapped too much it is assumed they are a chord not a slurred monophonic phrase.
- The mod wheel generally controls the amount of vibrato but other factors including the nature of the selected phrase fragment also affect vibrato so the mod wheel is not completely deterministic.

- Highly articulated attacks require detached playing -- space between notes. High velocity with no space between notes will result in tongued or bowed notes.
- Synful Orchestra will automatically create very quick portamento and crescendo/diminuendo effects especially if Delay for Expression is on. However, larger portamento effects require explicit use of the Synful Pitch Wheel mode (see [Synful Pitch Wheel](#) on page 39) and larger crescendo/diminuendo require use of the Volume pedal.
- Many synthesizers have moved towards a piano model with little or no use of the Expression or Volume control. These controls are crucial for expressive use of Synful Orchestra. See the expression select section in Control Parameters in a Synthesizer for details about correctly selecting and setting up the expression options in Synful Orchestra.
- Use of Delay for Expression in a last final mix is recommended (see [Delay for Expression](#) on page 37).
- If undesirable attacks occur (e.g. too noisy, not noisy enough, etc) trimming Velocity and or Expression/Volume will generally affect these qualities.

## RPM Reconstructive Phrase Modeling

---

Real instruments have many ways of moving from note to note: slurs with varying amounts of portamento, lightly tongued or bowed note transitions, fast runs, detached hard attacks, etc. These note transitions are the connective tissue of musical expression. Synful Orchestra models these transitions. Synful Orchestra models the way notes are sustained, the graceful onset of vibrato, breath sounds, the sound of the bow on the string, the change of tone color as a note becomes louder or softer.

Synful's innovative Reconstructive Phrase Modeling (RPM) technology is the core technology used in Synful Orchestra. It is the result of many years of research, several patents, and a lifetime of experimentation, design, and frustration with older synthesis technologies. You do not need to understand Synful RPM in order to use Synful Orchestra. Synful Orchestra is natural and easy to use. Just select the program and play. All the realism and expressive power of Synful Orchestra is available to you through normal MIDI operation. The complexity and sophistication of RPM is mainly on the inside. This section is provided for those more technically inclined users who are curious to understand what lies at the heart of Synful's technology.

The following picture gives an overview of Synful RPM. We discuss this picture in more detail below.

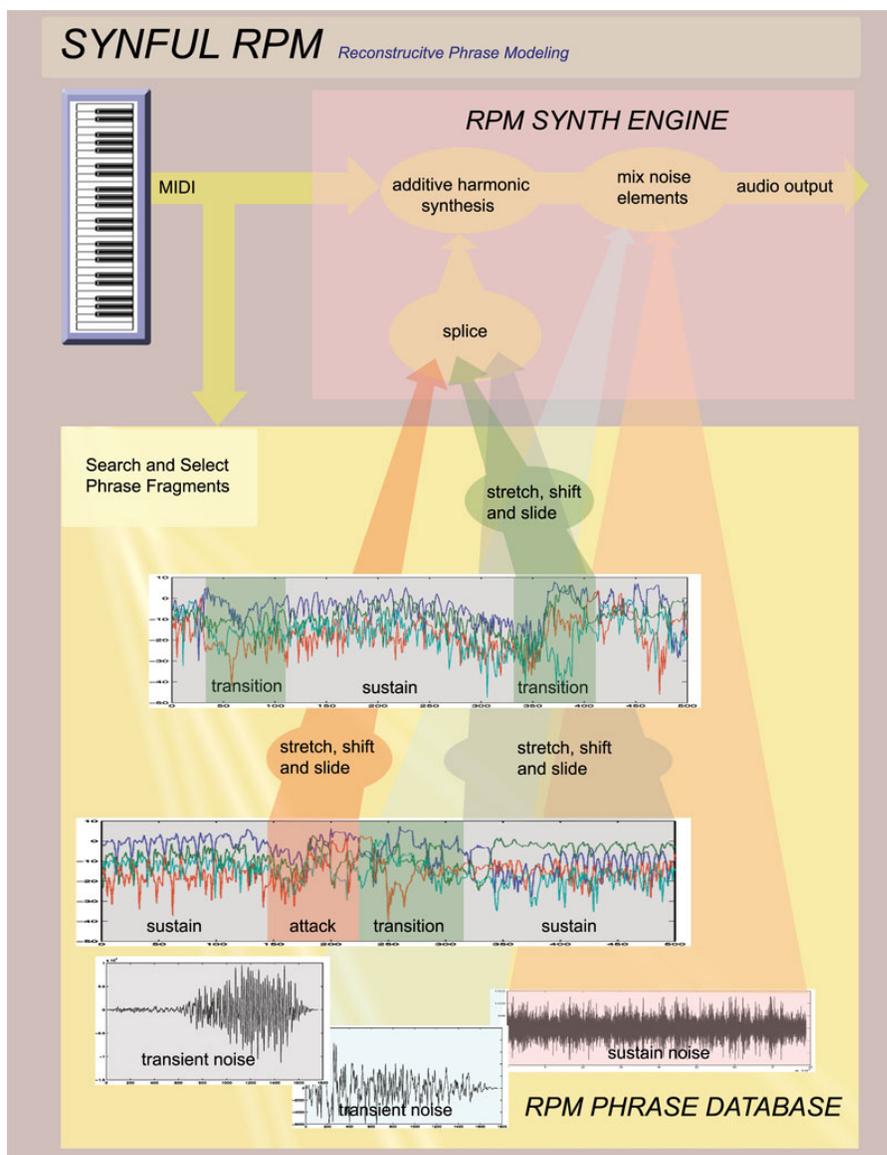


Figure 3: Synful Reconstructive Phrase Modeling (RPM)

## RPM Phrase Database

The RPM Phrase Database contains musical phrases for each instrument. These are not recordings of isolated notes but complete musical passages that represent all kinds of articulation and phrasing - detached, slurred, portamento, sharp attacks, soft attacks, etc. These phrases represent the many ways each instrument can move from one note to the next. You do not have direct access to the RPM Phrase Database.

## Searching and Splicing

When a Synful RPM Synthesizer such as Synful Orchestra receives MIDI input, either from a MIDI controller or a MIDI sequencer, it looks at the pitch, velocity, amount of separation or overlap between notes, note duration, volume, pitch, and mod wheels to determine what kind of phrase is being played. Synful Orchestra searches the RPM Phrase Database for fragments that can be spliced together to form this phrase. These fragments represent transitions between notes, slurs with portamento, lightly tongued transitions, aggressive fast bowing, rapid runs, long sustain regions with graceful vibrato, and noise elements like flute chiffs and bow scratches. Synful Orchestra stretches and shifts these fragments in time and pitch so that they combine to form the final output.

For example, suppose the RPM Synthesizer receives MIDI input consisting of a loud low G of 3/4 second duration followed by a loud low A of 1/8 second duration with no separation between the notes and silence before the first note and after the second note. The RPM Synthesizer interprets this as a loud slur from G to A with an accentuated short second note. The RPM Synthesizer searches the RPM Phrase Database for examples of loud low slurs. Suppose the RPM Synthesizer finds a low G# to B slur in the RPM Phrase Database where the G# is 1 second long and the B 1/3 second long. The RPM Synthesizer adjusts the note lengths of the database example and shifts the pitches to fit the desired G to A slurred output.

## Coarticulation in RPM - the Key to Natural Phrasing

In the example described above the G# to B slur found in the RPM Phrase Database is much more than an example of two notes one following the other. It is an example of a single "musical gesture". The timbre, envelope, and pitch contour of the end of the G# is affected by the fact that it is leading up to a new note. The articulation, pitch contour, and timbre of the B note is affected by the fact that it is connected to the end of another note. For example, the end of the first G# exhibits a crescendo and a slight rising portamento that indicates it is leading to a new higher pitched note. The beginning of the B note has a slight transient articulation that is perfectly matched to the end of the G#. This note connectedness, where adjacent notes affect each other is called "coarticulation". This is a term taken from speech synthesis where it is well known that the pronunciation of a syllable is affected by surrounding syllables.

Coarticulation effects are important in expressive instruments such as strings and winds. Percussion instruments including piano are less likely to exhibit coarticulation effects simply because there is not much one can do to the sound once a note is struck. Modelling coarticulation effects is essential for natural sounding expressive synthesis.

## Synful RPM Additive Synthesis

A single note in Synful Orchestra may be built from three or more rapidly spliced phrase fragments. Splicing ordinary PCM sampled sounds in this way would create unacceptable warbles and clicks. Synful Orchestra uses a patented form of additive synthesis in which sounds are generated from combinations of pure sine waves and noise elements. This gives Synful Orchestra the ability to rapidly stretch, shift, and splice phrase fragments while preserving perfect phrase continuity.

## Windows Tips on Audio Drivers and Low Latency

---

If you are not performing live with Synful Orchestra and are not sensitive to latency — the time from hitting a key on a MIDI controller till when the sound comes out — then just about any audio hardware and audio driver will function correctly for you (although some will have better sound quality than others). By low latency we mean that the sound comes out in less than 20-30 milliseconds from the time the key is pressed.

If you do intend to perform live then you need to PAY ATTENTION to your audio hardware and drivers. Interfacing to low latency audio drivers and hardware is primarily the responsibility of the host software program — for example, Sonar or Cubase. Please consult the documentation for these programs. This section provides some tips that we at Synful have learned that may help you have better success.

## What is an Audio Device Driver?

The audio device driver is a piece of software written by the audio hardware manufacturer that allows other programs like Cubase and Sonar to talk to the audio hardware. There can be more than one driver installed on a computer. For example, I am currently typing on an IBM Thinkpad R40. There are two device drivers installed on this laptop: an Analog Devices SoundMax audio driver that came with the Thinkpad when it was purchased; and a Tascam US-122 driver that came on a CD-ROM with the Tascam US-122 USB digital audio interface I purchased and installed several months ago.

Often there will be a choice of several types of device driver for a single piece of audio hardware. These correspond to different device driver standards that have come into being for Windows machines over the last many years.

## What are Some Types of Audio Device Drivers for Windows Machines?

- MME or Multimedia Driver — this is the old style Windows audio driver. You will not be able to achieve low latency with this type of driver. Latency may be 1/2 second or more.
- DirectX Driver — you may be able to achieve low latency with this type of driver but it will probably not be reliable. That means you may hear random clicks and pops as the sound plays. The problem will be worse the more programs that simultaneously run on your machine.
- ASIO Driver — This is a driver standard developed by Steinberg (manufacturers of Cubase) that supports reliable low latency. Many hardware manufacturers support this standard.
- WDM (Windows Driver Model) — This is a more recent driver standard from Microsoft intended to remedy the problems of MME and DirectX and provide reliable low latency. Most manufacturers now support WDM but you should check your hardware manufacturer to make sure.

If you intend to play live or real-time low latency music you should use an audio device that supports ASIO or WDM. Sonar makes good use of WDM but may not support ASIO. Cubase makes good use of ASIO but may not support WDM.

## What Happens When a Driver Doesn't Work?

If the driver is too slow you will generally hear clicks and pops in the audio output. These may occur very frequently like a kind of buzz, or randomly, or regularly — for example every two seconds. The lower the latency the more likely you will here clicks and pops.

## How Do I Adjust for the Lowest Latency Without Getting Clicks and Pops?

This can be done in the host program — for example:

### Sonar

Under Options select Audio. The Audio dialog box appears. Select the General Tab. If you have not already done so click the Wave Profiler button. This will test your hardware. There are two controls that affect latency: Buffer Size; and Buffers in Playback Queue. A very low latency setting is 2 Buffers in Playback Queue and Buffer Size around 11 msec or less. With the WDM driver I have gotten good results with this setting on my TASCAM US-122. ADI SoundMax has sometimes required 20 msec buffers. Although a synthesizer should work without any input driver (which is used for recording sound) I have found that when I disable the input driver nothing works. See Sonar Documentation for complete details.

### Cubase

Under Devices select Device Setup .... The Device Setup dialog box appears. Select VST Multitrack. In the ASIO Driver pull down box you will see a selection of drivers. ASIO Multimedia is the slow MME driver — not good for real-time live playing. ASIO DirectX Full Duplex is the DirectX driver. This can work for live playing but may not be reliable. For my Tascam US-122 hardware I have two drivers: an ASIO US-122 16 bit and an ASIO US-122 24 bit. Either of these work very well for low latency. Clicking on Control Panel brings up a Tascam dialog box that lets me fine tune the buffer size: a buffer size of 256 or 512 samples corresponds to about 5.5 or 11 milliseconds. These both work well with my hardware under Cubase. After changing latency in this dialog and hitting OK the sound is garbled until I click Apply. A different hardware device will undoubtedly have a different Control Panel dialog. Clicking Expert in the device setup dialog box brings up another set of controls that should affect latency and reliability but these controls have not had much affect in my system.

## Using the New French Horn

---

The new Synful French Horn appears in version 2.5 and later of Synful Orchestra.

### No Delay for Expression

The new French Horn does not use Delay for Expression (DFE). The DFE button can still be clicked and the sound will be delayed one second, but this is only for synchronizing with other Synful instruments. It has no affect on

the sound. We believe we have been able to achieve a level of natural phrasing as good or better than our DFE instruments without the need for the delay.

## Old and New Horns

The new Horn programs are called "horn", "horns4" and "horns8". Of course you can change the number of players from 1 to 16 in any of these programs. The old Horn has been renamed "old\_horn". When you open an old project with horn programs those programs are renamed with a prefix old\_ and they use the old horn sound. You must explicitly select one of the new horn programs if you want to use the new horn sound. Of course you can modify, rename, and copy the new programs for your own customizations.

## Expression Brassiness Control

The Synful Orchestra Expression Brassiness control is used to set the overall brassiness and loudness of the Horn sound. You can change this continuously from a very dark sound to an extremely brassy sound. For many phrases the Expression Brassiness control can be set once and does not need to change during the phrase. This makes the new horn much easier to use than many older Synful Orchestra instruments. You can also use the control to generate dramatic crescendi over the course of a single note or a phrase. The Gain also changes the loudness of the instrument but does not affect the timbre.

### Velocity

Velocity changes the shape of a note. A soft velocity produces a gentle attack with a fairly steady sustain. A hard velocity produces a sharper attack with a more pronounced decay. High velocity notes are also louder than low velocity notes, but to go from a pianissimo sound to a fortissimo sound requires that you change the Expression Brassiness control or Gain control.

### Examples Settings

- **Brassy and Heroic Sound.** Set expression to the max 127 and use high velocity.
- **Full Loud Sustained Sound.** Set expression to about 90-100 and use low to med velocities.
- **Dark Soft Sustained Sound.** Set expression to around 60-70. Use a variety of velocities.

## Gliss and Rips

The new horn introduces a continuous Gliss Rip control. Set all the way to the left the control is OFF. If you play a slur (two overlapping notes) with a rising pitch, when the Gliss Rip control is on, then a gliss rip sound is generated. The position of the control determines the speed of the gliss rip. It is most natural to map this control to a keyboard slider, the Mod Wheel, or a pedal.

### Example Settings

- **Classic Horn Rip.** Set gliss rip control to around .8 or .9. Set Expression Brassiness to a high value. Play a rising interval slur of about one octave or more. Set num\_players to something like 8.
- **Bluesy Bends.** Set gliss rip to a low value .1 or .2. Play a rising interval slur of a third or fourth.

## Trills

The new horn introduces a Trill control. The position of the control corresponds to the speed of the trill. To generate a trill, play a first note and hold it and then play a 2nd note. If the Trill control is on when you play the 2nd note then a trill will occur between the two notes as long as you hold both notes. The trill can occur between any two pitches with a speed determined by the Trill control position. It is most natural to map this control to a keyboard slider, the Mod Wheel, or a pedal.

Example: Play C one octave above middle C. Move Trill control to about .8. Continue holding the C and play D. You should hear the trill.

## Mute and Stopped Sounds and the Wah Control

The new French Horn has four playing modes: normal, mute1, mute2, and stopped. Like all playing modes these can be changed from the playing mode menu, using keyswitches, or with MIDI program changes. When you select a mute or stopped playing mode the Wah control becomes active. The Wah control has the effect of gradually removing the mute or hand from the bell of the horn. When the Wah control is all the way to the left the mute/hand is fully in the bell. As the Wah control is moved to the right the mute/hand is gradually removed from the bell. When the Wah control is all the way to the right the sound is the same as the normal playing mode.

## Horn Solo and Horn Sections

Three horn programs are provided : "horn", "horns4", and "horns8". These are for convenience. They are all based on the same horn instrument. If you turn the num\_players control in the "horn" program from 1 to 8 it is identical to the "horns8" program. You can have a solo horn or sections from 2 to 16 players. All the usual player positioning controls apply.

## The Horndemo Files

Included in the version 2.5 release are horndemo project files and a horn demo midi file. The horndemo consists of a number of different examples on the single track. If you open these to view the single MIDI track in an editor you will notice that the first two examples “Heldenleben” and “Mozart” use very little Expression Brassiness control. It is set at the beginning of each example and does not move during the phrase. The third example “Hannibal Rips” makes more extensive use of this control. The horndemo project files are in Logic, Digital Performer, Protools, Cubase, and Sonar Formats.

If you do not use one of these sequencers you can open the horndemo midi file in a project. Then do the following:

1. Create an instance of Synful Orchestra.
2. Unlock channel 1 on the Synful Orchestra Control Panel channel grid. Set the MIDI CC control of gliss rip to Mod Wheel.
3. Switch to the horns8 program on channel 1.
4. Switch temporarily to the stopped Playing Mode of horns8.
5. Set the MIDI CC control of wah to 73.
6. Set Key Switch to 88 Keys
7. Add a reverb
8. Play the horndemo from the beginning.

The horndemo midi file automatically changes to the correct programs and settings.

## Video Tutorials

---

An Internet connection is required to view the tutorial videos. If you don't have one on this machine then you can go to <http://www.synful.com/VideoTutorials.htm> on any machine. These videos show an older version of the Synful Orchestra but the overall operation is the same.

- *Video: Setting up Synful Orchestra in Logic*
- *Video: Setting up Synful Orchestra in Digital Performer*
- *Video: Setting up Synful Orchestra in Cubase*
- *Video: Setting up Synful Orchestra in Sonar*
- *Video: Open Control Panel in Cubase*
- *Video: Open Control Panel in Sonar*

## Apple Logic Multi-Channel Choices

In most host sequencers (Cubase, Protools, Sonar, etc) you create a synthesizer instrument track that holds an instance of the Synful Orchestra plugin. Then you create separate MIDI tracks that send their MIDI outputs from

separate MIDI channels to the single Synful Orchestra instrument track. The single instance can accommodate 16 MIDI channels. These MIDI channels would generally receive the outputs of 16 separate MIDI tracks in the host sequencer. If you need more than 16 channels you create more than one synthesizer instrument track.

This is a simple enough structure that is quite functional. Unfortunately, in Apple Logic this structure is not really available because you don't have a convenient way to send the output of a MIDI track to a separate instrument track that might contain a multi-channel (or what's generally called a multi-timbral) instrument. Instead in Logic you have two available options:

- Create separate instances of Synful Orchestra on separate instrument tracks where each track also holds its own single channel MIDI sequence. The disadvantage of this approach is that when you open the Synful Orchestra user interface for a particular track you only see the settings for a single MIDI channel (generally MIDI channel 1), and you generally must open the UI for a separate Synful Orchestra instance, to see the settings for a different MIDI channel. The advantage of this approach is that when you switch to mixer view in Logic you see a separate track with a separate gain control for each MIDI channel. This approach does not use more memory or compute resources compared to creating one Synful Orchestra instance. For those who have used versions of Synful Orchestra prior to 2.6.0 this is the way the Logic example projects were setup.
- Create a Logic multi-timbral instrument. This is done when first creating a new software instrument track in Logic. During this process you can check the multi-timbral option and select a number of MIDI channels for your multi-timbral instrument. Then Logic creates a number of software instrument tracks, but when you insert an instrument plugin like Synful Orchestra into any of those tracks a single instance will be shared by all the multi-timbral instrument tracks with a different MIDI channel assigned to each track -- see the Logic documentation for more details. This approach has the advantage that you can open up a single Synful Orchestra instance UI and set up its separate MIDI channels and they will be driven by the different software instrument tracks. However, this approach has the disadvantage that when you switch to mixer view in Logic you see only one gain control for the entire multi-channel Synful Orchestra instance. So, you have to adjust the individual gains for the different MIDI channels by adjusting the gain for that channel in the Synful Orchestra UI or mapping the gain to some MIDI controller (but not cc7 which is reserved by Logic). This is the approach used in the Synful Orchestra Logic examples from version 2.6.0 and later.

These Logic annoyances are not specific to Synful Orchestra but apply to all multi MIDI channel plugins.

