

# Advanced Multiport Environment

© 2012 by Norbert Drees

## 1 Introduction

This document was written to complement the chapter „Using the multiport layer“ from the Orchestral template tutorial (Version 2.0) by Tobias Escher.

In addition, it describes the inside of the port-macro of the multi-port layer example of the Vienna Symphonic Library company.

Finally this document presents a solution for the usage of more than 16 multi timbral instruments in Logic Pro and how to send its midi data to VEP. All taking into account the information from the Orchestral template tutorial.

Currently I am using 37 of the 48 possible VEP midi ports in my template. Everything is running on a Mac Pro (early 2008) with 20 GB ram.

### 1.1 Pre-Conditions

For the understanding of this document it is necessary to have a deep understanding of the following facts.

- The Orchestral template tutorial by Tobias Escher
- Knowledge about the Logic Pro environment and its transformer objects
- Midi and its data<sup>1</sup>

### 1.2 The idea

A large part of my template includes East West instruments. Certainly, other manufacturers are represented, but I still give the East West instruments a large portion of my songs. However, these instruments are not really performant as others. Especially when it comes to memory consumption. The PLAY manual recommends to always load several patches of an instrument in one and the same PLAY instance. This reduces memory consumption, because the different instrument patches use more or less the same samples.

Because I always use the Master-Keyswitch patch for an instrument, but also one or more instrument patches, I load them all into one PLAY instance. In general, the Master-Keyswitch patch is on midi channel 2 and all other patches are on the following midi channels of the respective instrument.

In addition, there are also instruments - e.g. Harp - for which there is no Master-Keyswitch patch.

In this case it is advisable to also use a multi timbral instrument to reflect the different articulations of such instruments in one PLAY instance.

In other words, each PLAY instance represents one instrument with one or more different

---

1

NRPN: <http://en.wikipedia.org/wiki/NRPN>

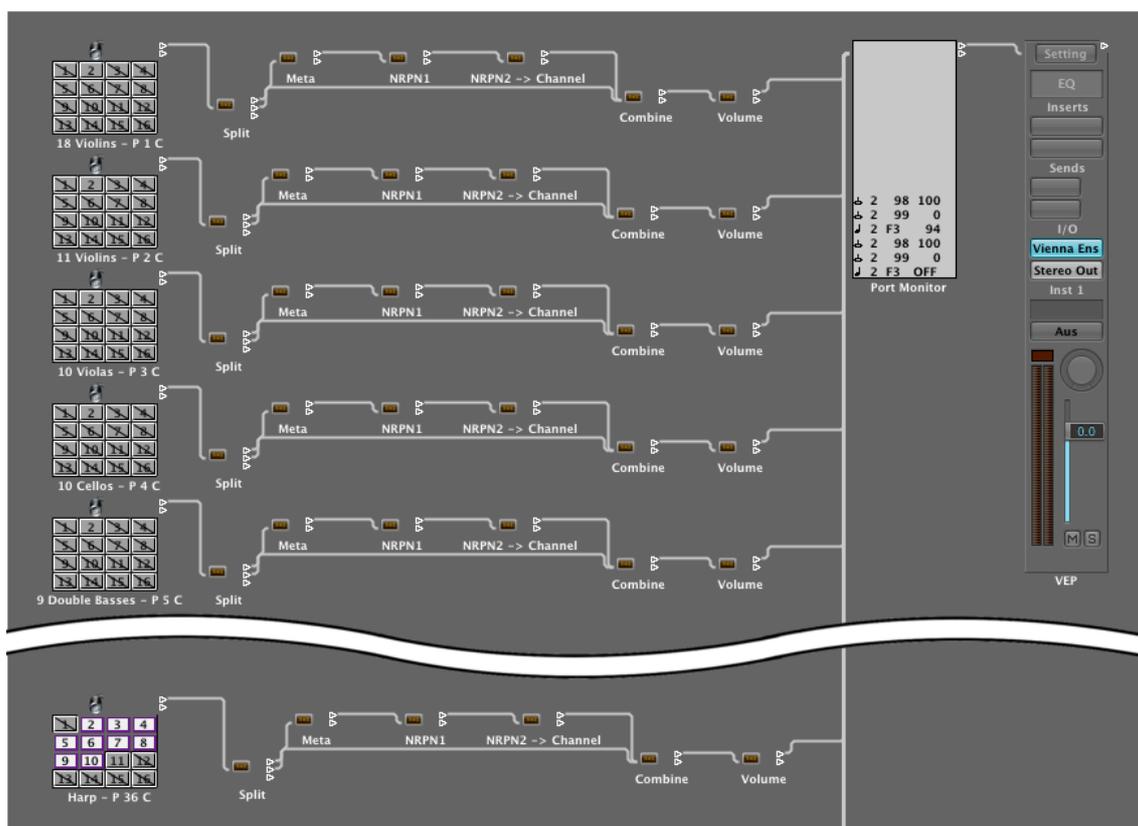
Midi : <http://www.rolandmusik.de/links/midiguide> (german language)

patches and/or articulations that is controlled via one multi timbral instrument.

When composing, I prefer a simple and practical arrangement view in Logic Pro. Moreover I do not like bouncing or freezing while composing. To this end, I just setup one VEP instance containing all instruments and route the stereo output of VEP into Logic Pro. This approach has the positive effect that the processor load is reduced significantly and I do not have to worry about technical issues while composing.

In order that the stereo output is not clipping, it was necessary to control the fader of each instrument in VEP with the corresponding fader of the multi timbral instrument in Logic Pro.

Under these conditions and considering the number of all instruments of an orchestra, I quickly realized that the number of the available 16 multi timbral instruments and their port macros from the example of the Vienna Symphonic Library company does not suffice.



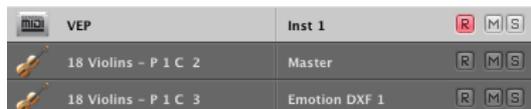
*Current orchestral template overview - simplified*

## 2 Naming Scheme for Multi Timbral Instruments

Many tutorials describe the use of clear patch names as well as introducing a color scheme. I wholeheartedly agree and I go one step further when it comes to the instrument label in the arrangement view. In addition to the patch name, I would like to see the MIDI-Port and MIDI-Channel in the label of each instrument.

To achieve this goal, I use a property of Logic Pro to my advantage. Logic Pro shows the channel number for each external midi track in the arrangement view. Therefore I developed a naming scheme where the Instrument name, the port number as well as the channel number are included.

The scheme is: <Instrument name> - P <Port number> C



*Channel and Port for a multi timbral instrument in arrangement view*

Where P stands for Port and C for Channel (Note that Logic will add the channel number automatically to the string if you are in the arrangement view).

I use the **Channel Strip Name** for the Instrument with channel and port information and the **Track Name** for the specific patch name.

### 3 Inside the port-macro

This chapter describes the inside of the port-macro of the multi port layer example of the Vienna Symphonic Library company. Who wants to try the functioning of the port macros first, can follow the chapter „Using the multi-port layer“ from the Orchestral template tutorial.

After importing there are 16 midi ports available. If you resize PORT1 you see all of its transformers. The PORT1 macro includes a Split, Meta, NRPN1, NRPN2 and a Combine transformer. These are described in the next section.

Unfortunately, we can not see the wiring of these transformers. The wiring of these transformers can only be guessed by their names.

Therefore I recommend to create and test the first port manually. To create more ports, simply copy the transformers and adapt them to your needs.

For the sake of simplicity I recommend to not „pack“ the transformers into a macro-object, because necessary adjustments on a macro-object can only be carried out if it was „unpacked“.

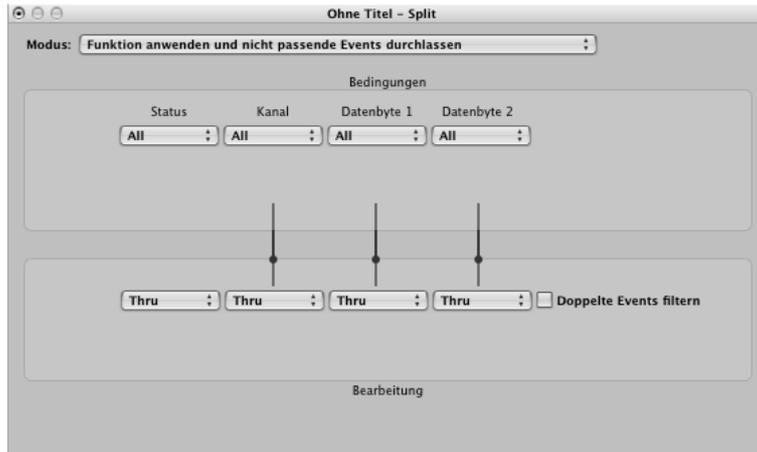
In addition, it should be noted that VEP supports 48 midi ports. I recommend to configure the maximum number of midi ports, because I have not noticed any runtime issues.

#### 3.1 How To

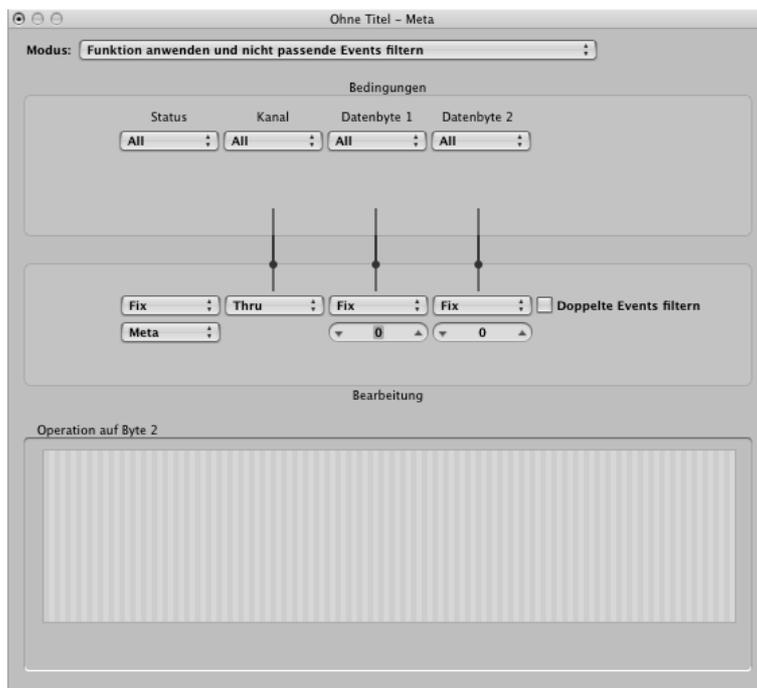
As a first step create a multi instrument object. Ignore midi channel 1, just select the number of midi channels you want to use for the specific instrument - e.g. channel 2 for a violin master patch and channel 3 for a legato patch and so on.

Next, the five transformers are generated and configured. The transformer NRPN2 contains the VEP midi port in data byte 2. Note that the data byte value for the VEP midi port is index based. This means that a 0 corresponds to the VEP midi port 1, a 1 corresponds to the VEP midi port 2 and so on.

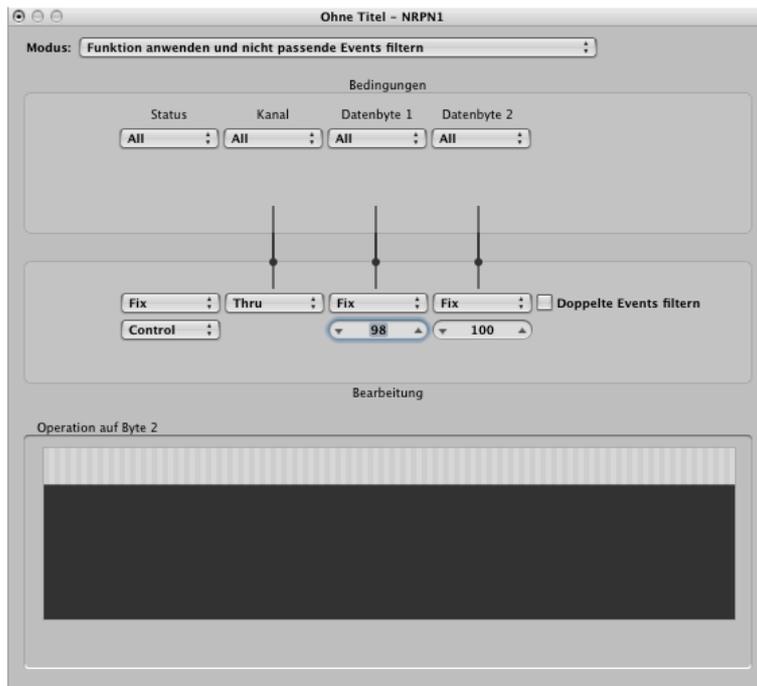
The pictures below show each of the five transformer. Create and configure them one by one.



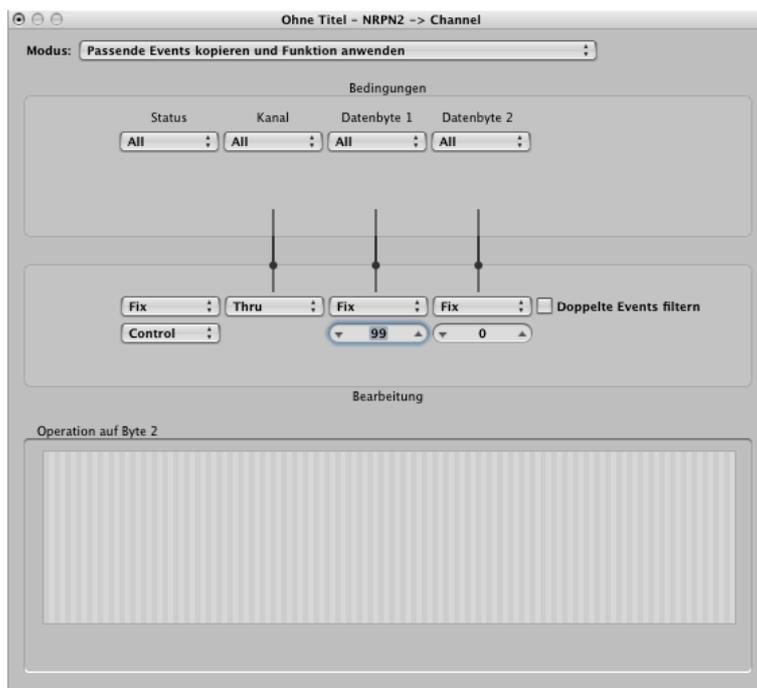
*Split Transformer - Apply Operation and Let Non-Matching Events Pass Thru*



*Meta Transformer - Apply Operation and Filter Non-Matching Events*



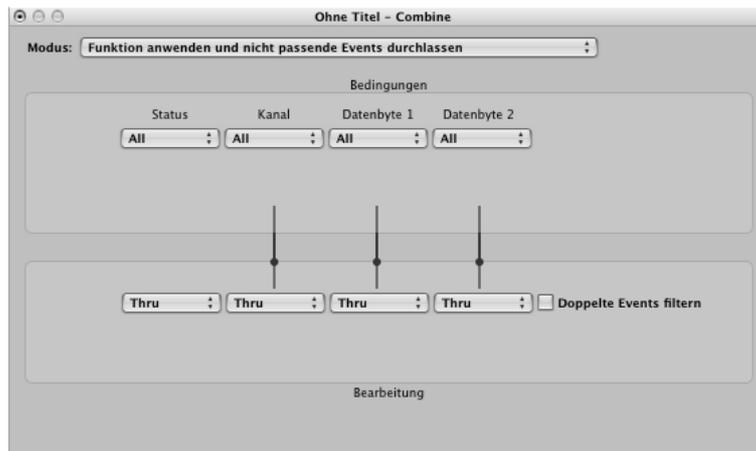
*NRPN1 Transformer - Apply Operation and Filter Non-Matching Events*



*NRPN2 Transformer - Copy Matching Events and Apply Operation*

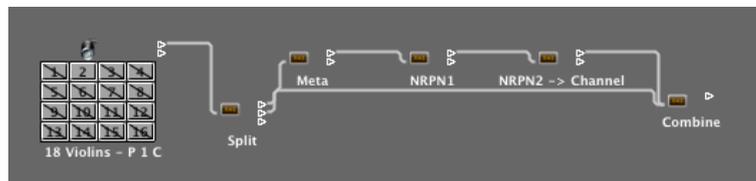
Again, note that the port number is index based! The picture shows the NRPN2 transformer configuration for VEP midi port 1.

If you want to create a port to access another VEP midi port, this is the transformer which must be adjusted. Simply adapt data byte 2 to your needs.



*Combine Transformer - Apply Operation and Let Non-Matching Events Pass Thru*

As one of the last steps wire the transformers as shown in the picture below. Moreover deselect the icon in the object parameter box for each of the transformers.



*Wired Transformers*

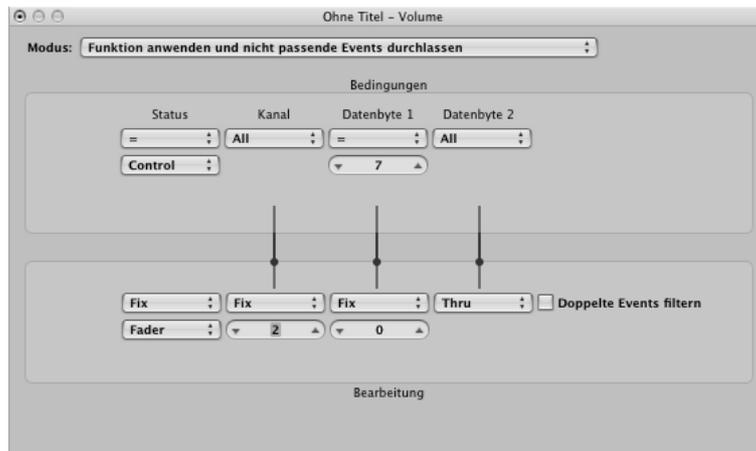
Finally wire the **Combine Transformer** to the VEP instrument.

Now it should be possible to create an external instrument in the arrangement view and assign its port to your created VEP midi port.

## 4 Control the VEP channel fader

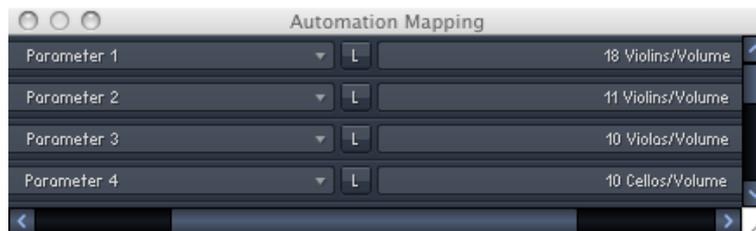
This chapter can be understood as a complement to the chapter „Making your instances Logic-friendly“ from the *Orchestral* template tutorial. It was added to this document to show just another way of controlling the volume of each instrument channel in VEP without changing the overall volume of an VEP PlugIn instance in Logic Pro.

The idea is to map control change 7 midi messages (CC7/Volume) from each external instrument on a specific midi channel to a VEP Automation Mapping Parameter. To achieve this, add a volume transformer between each port setup and the VEP instrument. Configure the transformer as shown in the picture below.



*Volume Transformer - Apply Operation and Let Non-Matching Events Pass Thru*

Note that the data byte 1 contains the midi port. Moreover note that the data byte value for the VEP midi port is index based. This means that a 0 corresponds to the VEP midi port 1, a 1 corresponds to the VEP midi port 2 and so on.



*VEP Automation Mapping*

Finally create Automation Mappings in VEP. For example map parameter 1 to the volume of channel 1.

Now it should be possible to control the VEP channel volume fader via the Logic Pro Fader. Moreover it is possible to write a CC7 message into your midi regions to control the VEP channel volume.

## 5 Copyright

The document is © 2012 by Norbert Drees - Norbert.Drees@GMX.de.

You are...

- welcome to distribute this document in its unaltered form.
- not allowed to use any extracts of this document without proper acknowledgement of its author.
- not allowed to make money out of this document.

All trademarks in this document belong to their respective owners.

I take no responsibility for any problems that arise in applying the information contained in this document.

In other words... if you follow the information in this document, you do this at your own risk.