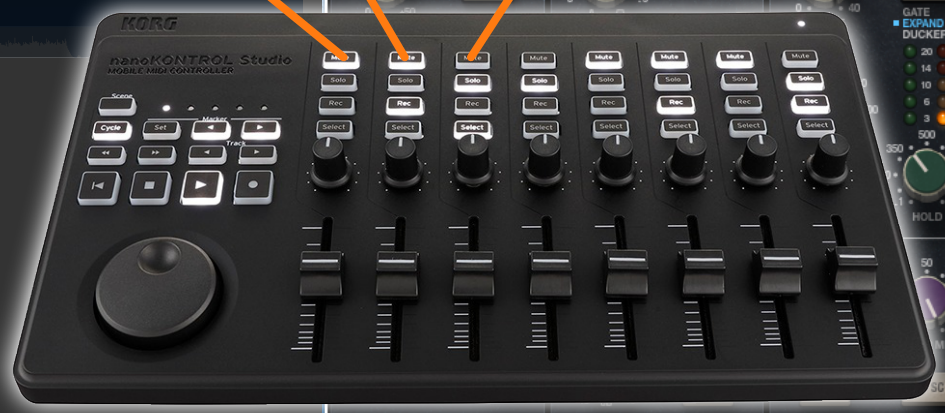
Practical projects using Logic's Controller Assignments

Volume 1

by **Simon Croft**



How to turn almost any MIDI device into a powerful hardware controller for all your Logic Pro X plug-ins

This book is dedicated to my wife, Hazel, who has listened to me ramble on about Controller Assignments for years now.

Supporting tutorial videos to accompany this book with be made available on the You Tube channel Open Your DAW to Hardware and More: https://www.youtube.com/channel/UC1c4Pbq_Y7BvSqGSudGluZA

These videos take time to make, so please be patient if there is not as much content on the channel as you hoped in the first instance.

You are welcome to download, copy and distribute this book. However, you may not charge anyone money for this book, and it must be distributed as a complete work.

For further help with your own builds, please use the forums at https://www.logicprohelp.com, and be sure to start a new thread if you have a question.

Who is this Controller Assignments book for?

I started to think that either Controller Assignments in Logic didn't work, or I wasn't clever enough. But I got there, and so can you…

If you know your way round Logic Pro X (LPX), then you're undoubtedly smart enough to learn how to create the control surfaces you desire using the integral Controller Assignment functions.

In fact, there's a fair chance you've already written a few, but maybe didn't get the results you were hoping for. Having been round that loop a few times before I started to get real results, I can assure you that you are exactly the person this book is designed to help.

What exactly are these 'Controller Assignments'?

They are a way of getting physical controls – rotaries, buttons and faders – to interact with a considerable range of LPX functions, from transport functions such as Play and Stop, through to channel levels, EQ settings, the on-screen controls in a vast range of plug-ins... In fact, if there is a way to get something to happen on LPX using a key-stroke, or a pull-down menu, the chances are you can assign that same function to a physical control on a MIDI device.

Fancy getting Drum Pad 1 to open the Arrange window, Drum Pad 2 to open the Mixer window, all the way to Drum Pad 8 opening Project Audio? You can do that in mere minutes, if that's how you'd like to work. □

Note – In the course of this book, I'll be using the terms 'Controller Assignments', 'CAs 'and 'assignments' interchangeably. Similarly, when I refer to 'LPX', I mean Logic Pro X.

About the author

I was lucky. Very lucky. Having started in pro audio sales in 1979, I wrote a users' manual for a mixing desk in 1983. A chance remark to a staffer on the *Melody Maker* music weekly led to a succession of commissions to write technical stuff for the section at the back.

Next thing I knew, I was editing *Sound Engineer & Producer* magazine... Since then, I've written thousands of articles – and a number of books – about broadcast technology, pro audio, musical instruments and media production.

I can't remember exactly when I jumped ship from Steinberg's Cubase to Emagic's Logic, but I was already using Apple Macs for magazine production, as well as making music. So when Apple bought the company on 2002, it didn't spoil my day too much.

What did spoil my day later on was discovering my low-budget MIDI controllers were no longer supported by the manufacturers, meaning plug 'n' play-time was over. I'd have to write the Controller Assignments for myself if I wanted to use the latest macOS and LPX.

Hence, I had to find out how to use LPX' Controller Assignments section. I'm not a programmer, so I made a lot of false starts. Eventually, a kindly gentleman by the name of Des66 answered my plea on LogicProHelp.com, and put me straight on some basics.

This little book is my way of saying thank you, and sharing what I've learned so far.

Disclaimers Simon Croft is not affiliated with, or sponsored by, Apple Inc, Korg Inc, Akai Professional, M-Audio, Waves, Cherry Audio, Arturia, Relab Developments, the Rose & Crown, or Happy Garden Chinese takeaway, Worthing. Any mention of trademarks belonging to these, or other, organisations is for informational purposes only.
This book is offered 'as is', and comes with no guarantees of performance. © Simon Croft 2023

Contents

1. Introduction p3
2. Transport control made easy (just to get you started) p8

Things you need to know (reference)

3. Exploring the Controller Assignment window p12
4. Zones, Modes, Controls and Parameters – what the heck? p14
5. Where Key Commands and Smart Controls fit into this p23
6. Preparation and planning (the most important chapter!) p25

Build Projects (step-by-step examples)

7. 24 channel mixer with graphic EQ p35
8. Live performance system with backing tracks and mixer p41
9. Some example plug-in Controller Assignments p47
10. CAs for classic Cherry Audio keys p58
10. Arturia V Collection the easy way p62
11. A brief wrap-up before we meet again p64

Introduction

Controller Assignments – why should I bother?

In my case, there was one very serious motivation: after years of work at a computer, during which I've laid out literally thousands of newspaper pages and written untold thousands of words, my right wrist is showing signs of RSI (Repetitive Strain Injury). If I spend endless hours trying to tweak on-screen controls using a mouse, I'm going to cause more damage, and it's permanent.

But there's also a creative benefit that applies to all of who use LPX to make music: you soon find that the ability to interact via physical controls encourages to do so much more frequently. For instance, synth parts become a lot more expressive if you can ride filters, oscillators and the like in real-time. That modelled analogue classic, which looks and sounds so much like the real thing, starts to *feel and behave* so much more like the real thing.

It's the same with mixes. With real faders, solo buttons, control over effects, and so on, the experience comes a lot closer to working on a big console. It's more ergonomic, it's faster and it's a lot more fun.

Best of all, if you already have a MIDI controller it will cost you nothing! If you don't, you can pick up a suitable controller for about the cost of a used, entry-level electric guitar.

How can this book help me?

Chapters 3-6 explain what Controller Assignments are, how to use them, and some invaluable advice on how you can design out all the irritating things that can derail your CA build project. Although it's a good idea to read through these chapters at least once before you start any large-scale assignment schemes, the information is broken down, to make it easy for you to go back and refresh your memory. It's designed as a reference work, as well as a straight-ahead read.

The rest of this book in broken down into practical projects that can be achieved by anyone with a bit of patience, LPX and a MIDI controller. In addition to step-by-step explanations of the process, there are extensive screen grabs that make it clear what the successful assignments should look like.

While you're very welcome to copy any of the assignment schemes in this book – in fact, that is the intention – it's highly likely that your set-up and needs are different from those used in the example Build Projects.

For instance, you probably have different

Continued over...

Will we be using Mackie's HUI and MCU protocols?

In a word: No. Mackie developed HUI (Human User Interface), and then MCU (Mackie Control Universal) are protocols that allow a DAW and a hardware controller to communicate with a minimum of setting up. Originally developed by Mackie in conjunction with Emagic specifically for Logic, MCU is now supported by most DAWs. Similarly, a number of manufacturers now offer control surfaces that use MCU.

These protocols operate at a different level to Controller Assignments within LPX, and it's proprietary technology that isn't available to us mere mortals. To implement HUI and MCU protocol requires considerable software development, not to mention the payment of a licence fee to use it and gain access to the source code.

However, the actual control commands sent out under MCU are still MIDI, so we can get pretty close to that level of functionality if we're prepared to put in the hours with the Controller Assignments dialogue window.

From previous page...

plug-ins, your hardware controller isn't one of the ones used in the examples, and you'd like to create a set-up that matches your style of working. Don't worry, these chapters are intended to give you the building blocks you need to do just that. There are multiple examples of how these building blocks can be used, and there is a check-list at the start of each chapter, so you can see at-a-glance what's covered.

How does MIDI fit into all this?

Apologies if you already have a good understanding of MIDI, and know most of what's in this small section. But even if you have a deep understanding of MIDI, it's worth reminding ourselves what it can do for us when it comes to Controller Assignments.

'MIDI' in this context means the current implementation of the MIDI 1.0 specification. While a number of manufacturers have shown prototypes of gear that supports the newer MIDI 2.0 specification, it is not widely implemented at the time of writing and has no bearing on what we can do with LPX Controller Assignments today.

If you have a thirst for a thorough understanding of MIDI, you probably won't be surprised to learn that www.midi.org is a good place to go. Specifically, if you follow this link https://www.midi.org/midi-articles/what-is-midi-guide-by-paul-lehrman, you can download an excellent introductory 22-page PDF by Paul D Lehrman, PhD Director, Music Engineering Tufts University.

A pocket history of MIDI

The MIDI standard was first proposed in 1981, and it started a revolution by allowing synthesisers to 'talk to one another'. We've all seen pictures of rock-star keyboard players from the 1970s playing a bank of keyboard instruments, or with their left hand on one instrument and their right hand stretched over to a different keyboard in another rack.

Iconic though it was at the time, MIDI made all that unnecessary. Suddenly, a keyboard made by, say, Roland could transmit all the notes from a chord, along with information about velocity, panning and more to an instrument made by Sequential Circuits, Yamaha, or any other manufacturer who supported the MIDI standard in their products.

Because MIDI also supported a whole range of other information, including Continuous Controllers – such as modulation – as well as patch changes, tempo and much more, it soon became obvious just how much MIDI made possible. For instance, if the remote synth was being being played from another keyboard, there was no real need for it to have a keyboard of its own. The 'MIDI module' that sat in a rack was born.

Not only that, with MIDI Show Control, MIDI Machine Control and MIDI Time Code included in the protocol, MIDI devices could be integrated in all sorts of ways. For instance, it led to the birth of MIDI sequencing, where all the data needed to replay an arrangement was stored on a disk, but the audio was not. That meant entire songs – even whole collections of songs – could be stored on a cheap, removable computer disk.

(Whereas audio files typically run to many MB of data each (which you can multiply for every track in an arrangement), a MIDI file might only be 16KB for an entire song. Ideal for when processor speeds were slow, RAM was expensive and the main storage option was a 'diskette' that stored less than a megabyte, even by the early 1980s.)

The fact that it is possible to record to a sequencer and play back to a sound module tells us that MIDI is bi-directional. This can be important when it comes to a MIDI controller, because it means, for instance, devices that have a 'skirt' of LEDs round the rotaries can reflect the position of the assigned on-screen controls, and vice versa.

Continued over...

N.B. I have no Korg software installed for the nanoKontrol. According to the Korg manual, Controller Assignments can still be made with with the nanoKontroller in Assignable Mode. Apparently, this is achieved by holding down the Scene and Cycle buttons, then switching the unit on. I've not tried this myself.

From previous page...

What MIDI does

The full MIDI 1.0 specification runs to just over 85 pages. A PDF can be downloaded free at the MIDI Association web site – midi.org – by anyone who signs up. Obviously, we can only scratch the surface here, and in many ways that's probably as it should be.

Some people drive cars perfectly safely, with only the vaguest notion of how the engine works, the function of the gearbox, what the distributor does and so on. Other people can set the timing of the distributor, or even swap out an engine. Perhaps it's the same with people using MIDI.

To those of you who are in the 'don't know how it works' group, let me assure you that you can successfully create Controller Assignments with a minimum of MIDI knowledge, and a lot of that can be picked up as we go along.

To those of you who are more technical, I apologise if what I'm about to explain is so shallow, it leaves you feeling vaguely insulted! As we know, MIDI was devised primarily as a way of communicating between synths. So how does the ability to do that adapt to the world of Controller Assignments, where we're communicating with LPX?

When you play a note on a MIDI keyboard, the most important message transmitted is Note-On. Suffice to say, that message has to contain data as to the unique number of the key(s) being pressed, which tells whatever is generating the sound (keyboard/module/plug-in...) which notes to play. Also, the receiving sound generator needs to know when to stop!

This is achieved either by sending a Note-Off message, or a Note-On message with a Velocity value of zero. (Velocity is one of the messages MIDI can convey but doesn't really apply to Controller Assignments, so we needn't go into it here.)

What else is MIDI transmitting? It's also telling the receiving device which of 16 channels it's transmitting on. This is how each part in an arrangement plays using the appropriate sound. As anyone who has accidentally routed a drum part to a piano knows, any confusion in this department can be quite unpleasant!

MIDI Channel numbers

Where MIDI Channel numbers can be relevant to Controller Assignments is, for instance, if you are sending them from more than one physical device. If LPX understands the difference between messages from my keyboard and from my controller, things are a lot more orderly.

So what sends Note-On/Note-Off messages on a MIDI controller? No prizes for guessing it's all the push switches on the panel, each of which sends a unique MIDI message. The fact that LPX is using the incoming message to do something entirely different from sounding a musical note is down to what you've set up in the Controller Assignments window.

Continuous Control

When you waggle the Mod wheel on a keyboard, it transmits a whole series of Control messages, which are sometimes referred to as 'CCs', or 'Continuous Control' messages. Because there are 128 possible values (ranged from 0-127), the human ear perceives the result as a continuous change, rather than a series of steps.

And those are exactly the MIDI messages sent out by all the rotary controls and sliders/faders on a MIDI controller. Again, the graduations between one Control message and the next are fine enough to make level, filter and other changes to the sound seem completely smooth.

If you've already had a peek into the Controller Assignments window (assuming it's in Advanced mode), or you have a utility such as MIDI Monitor (free from www.snoize.com), you'll already know that MIDI messages are written in hexadecimal. That is to say, it uses numbers 0-9 and letters A-F to represent a total of 16 values.

While some people are clever enough to write those messages manually, you may be relieved to know it's not compulsory. Providing you can point at what you want on screen and operate the physical control you want assigned

Continued over...

From previous page...

to it, Logic will do the rest.

Well, that's how it should work. If you don't start out right, LPX can get very confused regarding your intentions, and can start to appear as if it's acting at random, or being downright bloody minded!

There will be a full examination in *Chapter 6 – Preparation and Planning*, of the things you can do to make CA consistent and reliable. I recommend you to read it before you get too hands-on, because it could save you a lot of frustration and heartache.

Choosing a suitable MIDI controller for your set-up

Maybe you don't have a MIDI controller yet, or perhaps the one you already have doesn't have enough physical controls to execute the scheme you have in mind. (You might want to read the rest of this book before deciding just how big or small your new controller needs to be though...)

No one can tell you which is the 'best controller', because the best one for you depends on exactly what you are hoping to achieve, and how much you are prepared to spend. Please don't imagine the message I'm sending out is: "You're going to need a lot of money." Actually, it's the opposite. All the CAs created in this book were done using a Korg nanoKontrol, mostly for the performance system and an Akai Professional APC40 Ableton Controller for the other projects. I don't think I paid more than £90 for either of them.

In addition, I'm using an M-Audio Axiom 49. It was actually this keyboard that got me into the whole business of Controller Assignments, because I discovered that the transport controls stopped working after I'd updated macOS and LPX. It turned out that new software for the Axiom 49 wasn't being released, so if I wanted anything much beyond the keys, pitch and mod wheels on the keyboard to work, I was going to have to make that happen myself.

Making a scene

The capabilities of the controller are obviously key to your purchasing decision. For instance, I decided against a number of highly rated models, because they were 'all buttons and no knobs', if you see what I mean. There's not much point in being able to address 40+ plug-ins, if all you can do when you have a plug-in loaded is adjust four parameters.

Although you can get some idea of the capabilities of a MIDI controller by just looking at it, there are ways of looking a bit deeper. In the case of the APC40, I found a PDF, grandly titled: Communications Protocol For Akai APC40. I can't say it was a fun read, but it told me what I wanted to know.

For instance, not all of the buttons transmit MIDI. There's a line of nine marked 'Track Selection', and they cause the lower eight rotaries and eight buttons to transmit different MIDI messages, depending which button is pushed. As well see, that's brilliant when creating assignments for software synths, because it allows you to address multiple sections from the same small group of controls.

In this way, the two banks of eight rotaries offer much more than 16 potential assignments per plugin. By working though the track selection buttons, the lower section provides a further 64 rotary assignment and the same

Continued over...

Only if you're Able?

You might be thinking: "Why did you pick an Ableton Controller when you're running LPX?" The answer is they sell quite cheaply on ebay, especially for such a well specified and nicely made unit. My guess is that a lot of potential users assume they're only any use if you run Ableton, so buy something else.

The only thing I would warn you is some of the APC40s out there have been heavily used by DJs in performance situations. If you see signs of heavy wear on the front panel, the chances are, some of the faders and rotaries are on their last legs.

I know because I ended up negotiating the first APC40 I bought down to less than £25, after I discovered three faders weren't working and some of the rotaries were sending Control messages when no one was touching them. You can probably imagine what chaotic CAs you create if you hit Learn and that starts happening!

However, I'd only buy a knackered example of you're handy with a soldering iron.

From previous page...

number of On/Off assignments. That means it's possible to write full sets of assignments for even very complicated-looking analogue synth emulations. In the case of classic compressors, you can choose from eight or more just using the Track Selection to address each one in turn.

The compact Korg nanoKontrol does something similar with a single Scene button that scrolls through five different sets of MIDI messages transmitted by the 32 buttons at the top of the unit. It doesn't take a maths graduate to work out that's 160 possible MIDI messages from a controller that takes up less space on a desk than most iPads.

The takeaway here is it's best to find out exactly what a controller can do in MIDI Message terms, rather than just having a button-count before you buy. It may be that the controller is more versatile than you first thought.

Seeing is believing

A limitation of the nanoKontrol is that it provides very little visual feedback on the control surface. For instance, the only way you can find out if physical position of the rotary controls and sliders match those of the settings in LPX is to move them around until the on-screen equivalents 'pick-up' and start moving.

In contrast, the LED skirts round the main 16 rotaries on the Akai unit reflect the positions of the controls on the selected plug-in, and the control surface and on-screen settings are always in sync. (Alternatively, if you move the on-screen controls with a mouse, the LEDs on the Akai dance round to match, which is fun to watch. Oh, how the winter evenings fly past in the Croft household…)

There are a number of illuminated buttons on the Akai unit that remain useful when working with LPX, notably: Record Arm; Solo/Cue; Activator and Track Selection. The 40 illuminated Clip Launch buttons are probably a life-saver when you're working with Ableton in a badly lit club. In my set-up, they're how I address specific plug-ins. (As we'll discover, they can be used for far more than 40 plug-ins too.)

Armed with that information, you might want to ask yourself if you'd be better off paying a bit more and getting a controller that is designed to work with LPX straight out of the box. If you come to that conclusion, and you're thinking of buying a used unit, I'd urge you to check on the manufacturer's web site to see the last version of macOS they support.

I've seen too many posts on-line from people who have bought controllers that were state-of-the-art maybe 15 years ago, only to discover they won't work with any modern Mac. A lot of manufacturers stop updating software drivers and the like after maybe less than five years, which is exactly how I came to get involved with the whole Controller Assignment business.

Fortunately, all the MIDI devices I own will operate without driver software from the manufacturer. That makes them useful to me, even without manufacturer support. Some controllers need specific driver software before the Mac can 'see' the controller at all. At the risk of stating the obvious, that means the device is next to useless to you if the only available drivers are not compatible with the oldest OS you need to run the version of LPX you have.

MIDI muddle managed

There's another question worth asking. Can you edit the MIDI settings on the controller?

Specifically, is there a way to set the MIDI Channel the device is transmitting on? This is important when you have more than one MIDI device, such as a keyboard and a dedicated controller, because things will start to get messy if they are both sending the same MIDI messages on the same channel.

Sometimes, the ability to edit MIDI settings is managed on-board, usually with the aid on an LCD window that shows what parameter is being adjusted, along with its current value.

Other times, there is editor software supplied. Again, there may come a time when editor software for your device is no longer available. This is something worth factoring in when buying a used unit. Even if the supporting software is available for download today, if the unit is already a few years old, it may be that future versions of macOS will not be supported.

This is especially the case if the unit is now discontinued. Software development is expensive, so it's hard for a manufacturer to justify updating products for which there is no revenue stream. Good luck with your hunt, and I hope you find a bargain. □

Transport control made easy (just to get you started)

What you'll learn in this chapter:

When I reference 'transport controls', I'm talking about functions like Play, Stop and Record, etc. In this chapter you'll learn how to create CAs for them, and also how to change the function of these CA, simply by picking a different Command from the list.

I've put this chapter up front – and ahead of the other hands-on chapters in the Build Project section – for two reasons. Firstly, I know that some of you will have MIDI devices for which there is no manufacturer-supplied software. Maybe you updated macOS, realised those transport buttons had stopped working on your MIDI controller. Then you went to the manufacturer's web site and discovered that the most recent software doesn't support the OS you've just installed. Don't worry, I'll show you how to fix it.

Secondly, I know some of you will want to get hands-on right now, to prove to yourself that this Controller Assignment business is for you. After all, why should you want to read through pages of my yada-yada, when you're not even sure you can make this work? Trust me. You can.

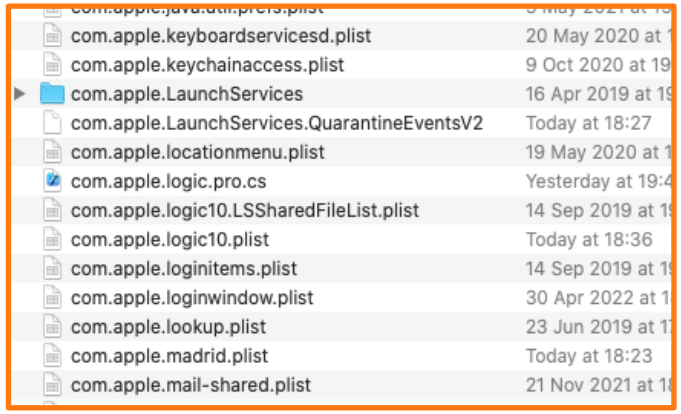
A little preparation

Before we start writing CAs, we need to put in a little preparation to make sure your copy of LPX is in the same state as mine. (There will be a lot more detail on this in Chapter 6.) Otherwise, what you see might not be the same as the screen-shots in this book, and you might not get the same results. (It doesn't matter what *version* of LPX you have: it will work the same.)

The first step takes place before you launch LPX. You need to navigate to here: *~/Library/Preferences/com.apple.logic.pro.cs* file.

If you're not sure how to find *Preferences*, here goes. Click on Finder, or the Desktop. Select the Go menu from the bar at the top of the screen. Hold down the alt key, so that Library appears, then select Library. Open the Preferences folder. Scroll down to the file 'com.apple.logic.pro.cs'. Now, move it to the Bin/Trash. Alternatively, as this file contains all your current controller assignments and all Control Surfaces preferences, you may wish to save it somewhere so that you have the option of restoring LPX to its present state if you decide to do that down the line.

The next step is to launch LPX and go to *Logic Pro/Preferences/Advanced Tools*. You will need to click on the Show Advanced Tools checkbox and the Control Surfaces checkbox. Without doing this, you will not be able to use the advanced functions described throughout this book.



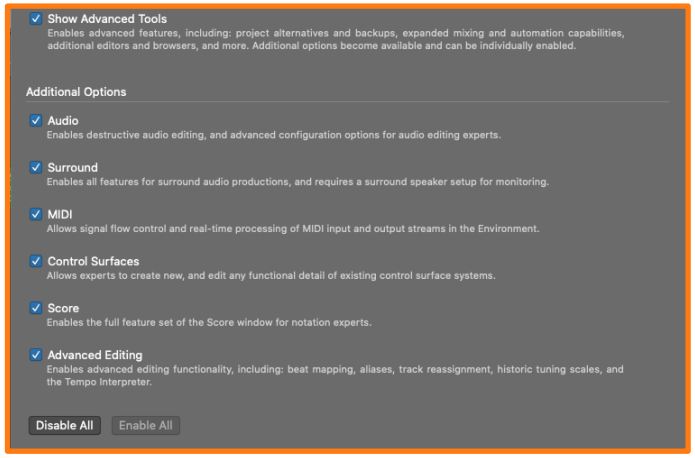
Tip: If you open the apple.logic.pro.cs file just once in Apple's Xcode, it will forever more display the blue Xcode icon, making it very easy to spot in a long list.

For the rest of this chapter, and throughout this book, we will be making use of the Learn Assignment function (Cmd-L). It can also be accessed through *Logic Pro/Control Surfaces/Learn Assignment*, or activated using the blue Learn button in a number of dialogue windows.

The other important control tool we will be using is the Controller Assignments window (Alt/Shift-K). It can also be accessed through *Logic Pro/Control Surfaces/Controller Assignments*... This window provides the environment where you create, edit and organise CAs.

Because we'll be using the Learn function and the Controller Assignment window so

Continued over...



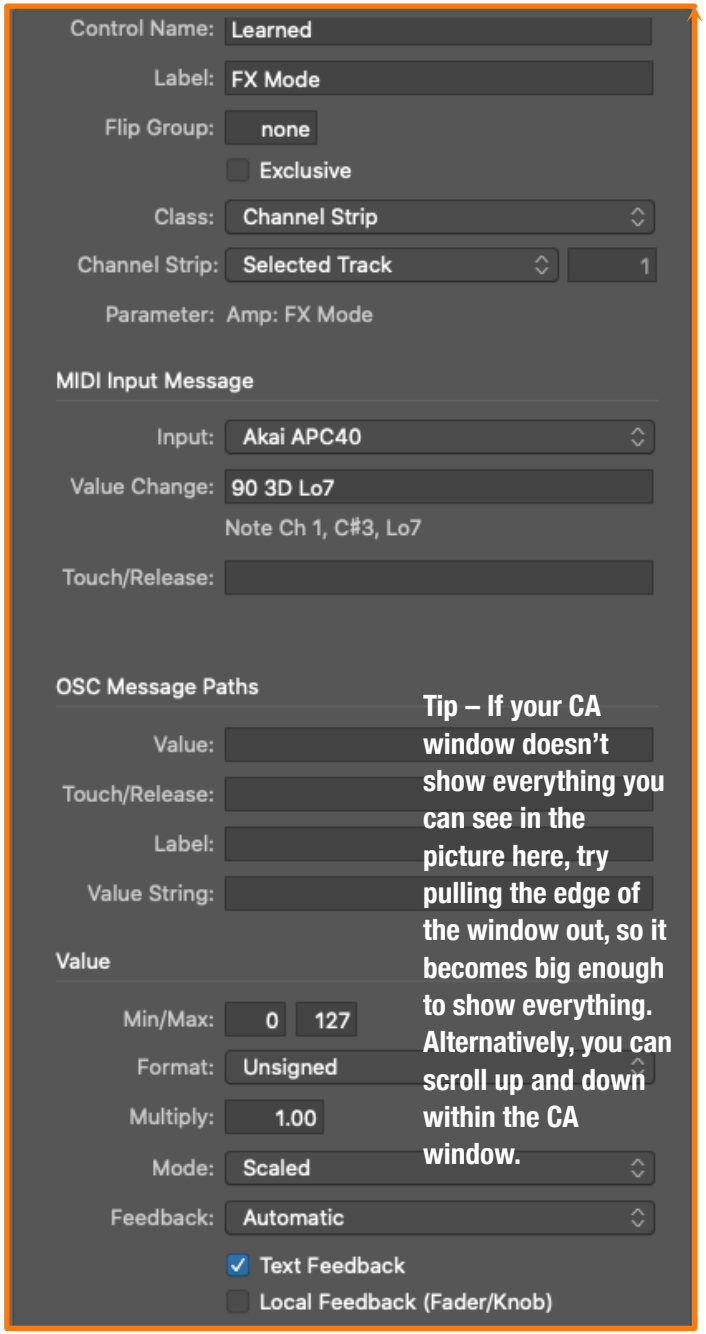
Advanced tools: Logic Pro/Preferences/Advanced Mode, while you're there, you might as well check the lot!

From previous page..

much, I can't recommend highly enough that you get used to using the Cmd-L and Alt/Shift-K key commands.

Our first three Assignments

The Learn function allows you to assign whatever you mouse-clicked on last to a physical control. This is done by keying Cmd-L, then touching the control you want to assign to that function. For instance, if the last thing you did was put LPX into Play, if you key Cmd-L, whatever button you press next on you MIDI controller becomes the Play button. You can see how this looks in the Controller Assignments window (next page). (But don't forget to key Cmd-L a second time to turn off Learn first!)



Tip – If your CA window doesn't show everything you can see in the picture here, try pulling the edge of the window out, so it becomes big enough to show everything. Alternatively, you can scroll up and down within the CA window.

Note - Global Commands need to be accessible to you at all times, regardless of what else your MIDI controller is assigned to at the time. For this reason, they belong in (No Zone). If they've ended up anywhere else, I suggest you use Cmd-X and Cmd-V to Cut & Paste them to where they should be. Chapter 4 - *Zones, Modes, Controls and Parameters - what the heck?* Examines the relationship between these concepts, along with the hierarchical organisation they bring to your CAs.

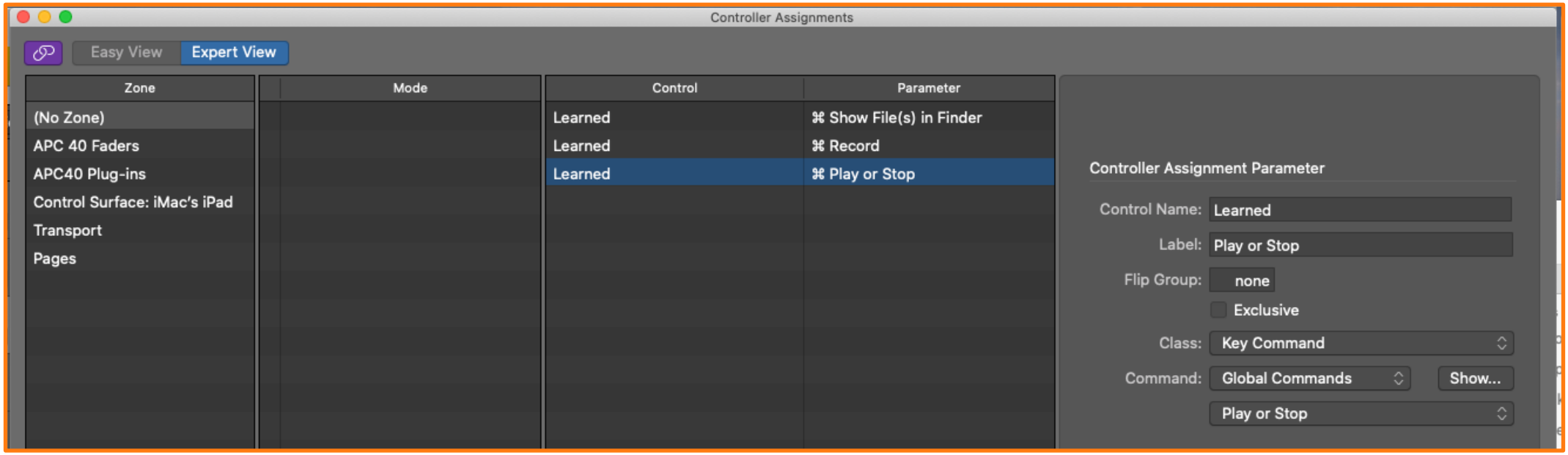
1. Assign Play button

The following instructions will work for almost any MIDI buttons, so if you want to use three drum pads for the fun of it, go right ahead! Once you have decided which three physical buttons are going to be your transport controls, press Play on screen, then key Cmd-L and press the physical control that will be Play. Key Cmd-L again to stop the Learn function. Now let's find out what you've just created.

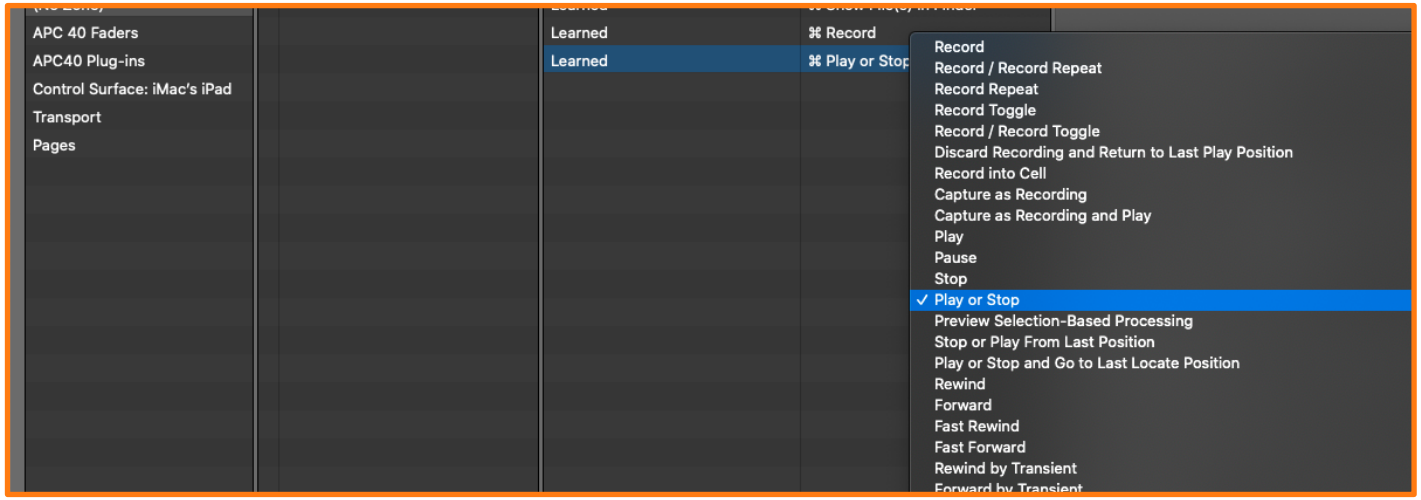
Hit your assigned Play button, and it probably doesn't do quite what you think. If LPX is stopped, the Play button does put it into Play. But if you press it again, it acts as a Stop button. Huh? Let's go Alt/Shift-K to find out what's really going on.

In the Controller Assignments window, we

Continued over...



Play or Stop. After I'd created my Play and Stop assignments, I thought about how pointless it is to use two buttons, when the Space bar toggles between the two. So I changed Stop to Play or Stop from the pull-down menu, and changed the Play button to Show File(s) in Finder. This is a brilliant timesaver that opens the folder where the file for any audio region you click on the in Arrange window is located. Of course, if all your related files are in the same folder, you can simply Drag and Drop any additional files in your Project onto the Arrange window.



From previous page..

can see that the command we thought was going to be Play is actually called 'Play or Stop'. That's exactly what it does: toggle between Play and Stop. Really quite useful if you only want one button to handle both functions.

But what if you really want that button to be Play, and nothing else. No sweat. Where it says Play or Stop in the CA window, click on it to reveal the full pull-down menu of options. Three items up the list is Play and if you select that, you'll have just made yourself a Play button.

Congratulations on your first successful assignment! Believe me, there will be many more.

Continued over...

From previous page..

2. Assign Stop Button

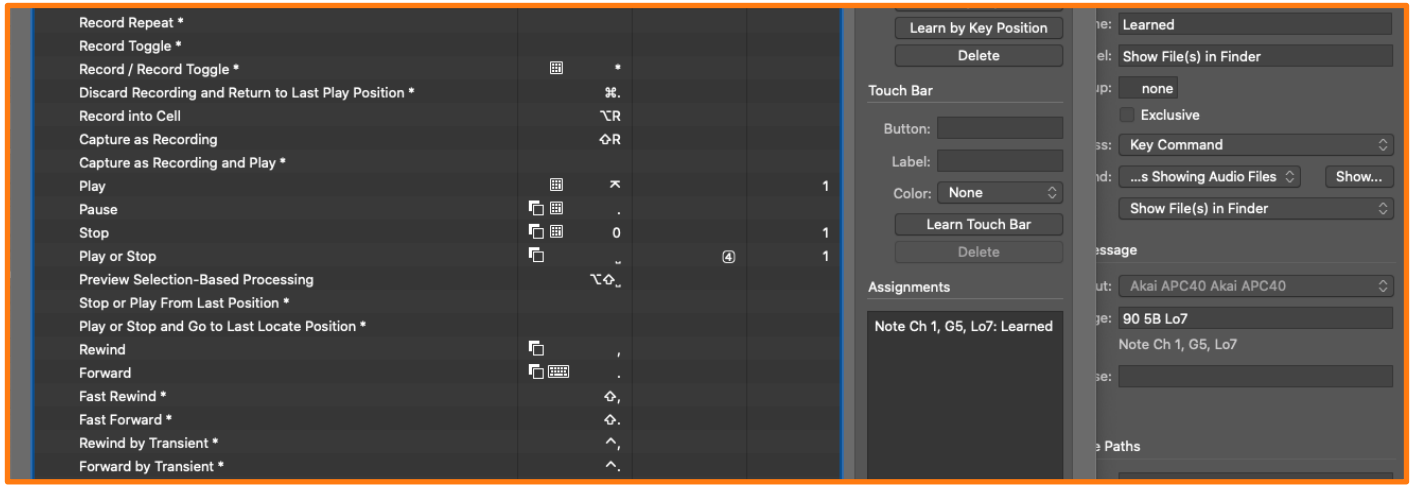
Believe it or not, you can create a Stop button by following the steps above for assigning a Play button. Only this time, press the button you want to be Stop. The chances are, the command you'll get is 'Play or Stop', but you can change it to Stop using the pull-down menu again.

This ability to modify Commands after the assignment has been created is very important. In addition to using it to correct mistakes, you can scroll through the whole long list and see what Commands are useful to you by trying them out.

3. Assign Record Button – and so much more!

I'm sure by now you've grasped that you can create a Record assignment using exactly the same method as the Play and Stop buttons, simply by choosing Record from the top of the pull-down menu.

More generally, you can choose *any* command from the list, meaning you can create CAs that take you to specific locations in your arrangement, for instance. In real life, if you created a CA for every possible Global Command, you would probably find that you only used a small number on a regular basis. Unless you are prepared to physically label your controller (or create an overlay you can wite on) you are likely to forget what some of the lesser-used buttons actually do.



Key Commands can be accessed from within the Controller Assignments window, by using the Show… button

It's also worth considering that a lot of Global Commands already have Qwerty keyboard short-cuts, or Key Commands. Unless you are planning on a largely Qwerty-free working environment, you'll probably use at least some of these on a regular basis. For instance using the Space bar for Play/Stop and the R key to enter record. No matter how many MIDI controllers I have, or how many CAs I write for them, about the only time I use a physical controller to go into Play is in a performance situation, where I don't use a Qwerty keyboard.

One way of viewing the list of Key Commands is from within the Controller Assignment window. If you look to the right of the Command pull-down menu, you'll see a button marked Show, which brings up the Key Commands menu. We'll look at the relationship between CAs and KCs in more detail in Chapter 5 – *Where Key Commands and Smart Controls fit into this*. This Transport Controller chapter is essentially just a Quick Start Guide to get you up and running.

For now, if you've got yourself three or more working CAs, well done! □

Note – This isn't the only way to go about creating the transport assignments. Remember, I said "The Learn function allows you to assign **whatever you mouse-clicked on last** to a physical control." If you click on Record, then go into Learn mode, you'll get a Record button when you hit a button on your MIDI device.

Exploring the Controller Assignment window

What you'll learn in this chapter:

You'll gain overview of how to access the Controller Assignments window, and how it is structured. It is suggested you read Chapters 4 and 5 directly after this to get a better idea of the 'big picture'. These three chapters are, by design, short. On one hand, there's a lot to take in. On the other, when you want to refer back to something, it'll be easier to find if the topic is broken into short chapters.

There are at least two ways to open the Controller Assignments window. One is to key Alt/Shift-K. Another is to go to Logic Pro at the top of the window, then navigate to *Control Surfaces/Controller Assignments*, which is definitely the long way round, but always there if you forget the Key Command.

You'll be accessing the CA window a lot, and the sooner you get Alt/Shift-K into 'muscle memory', the faster you'll be able to work.

The CA window is best read from left-to-right. This is because Modes sit inside Zones. Similarly, Controls and their Parameters generally sit within Modes. Don't worry if this all sounds like the instructions to a game you've never even seen, let alone, played! Things will become a lot clearer when we get to Chapter 4. For now, I just want to talk you through the user interface, so you know where everything is.

At the top of the screen you'll see a blue button marked Expert View. (You can see the CA window on the next page, or by launching LPX and using Alt/Shift-K). This is the view we will be using exclusively throughout this book. In fact, if you have already set up Advanced Tools, you'll probably find Easy View is greyed out.

If you haven't already set up Advanced tools, now is a good time to go to *Logic Pro/ Preferences/Advanced Tools*. You will need to click on the Show Advanced Tools checkbox and the Control Surfaces checkbox. Without doing this, you will not be able to use the advanced functions described throughout this book.

At the bottom of the CA window, you'll see three buttons with '+' symbols. These are for adding new Zones, Modes and Controls. When you click on a '+', a blank version of the Zone, Mode or Control is created. It already by highlighted, so you can immediately name it.

To the right of the '+' buttons is a Learn button. When Learn is active you can create Parameters for your Control. In other words, you can define what that Control does – usually by clicking on an on-screen knob, button, or fader, then moving a physical equivalent on your MIDI controller(s). Of course, you can also switch the Learn function off/on by using Command-L. Even when you do this, the Learn button is useful, because it gives visual confirmation of whether Learn is on or off.

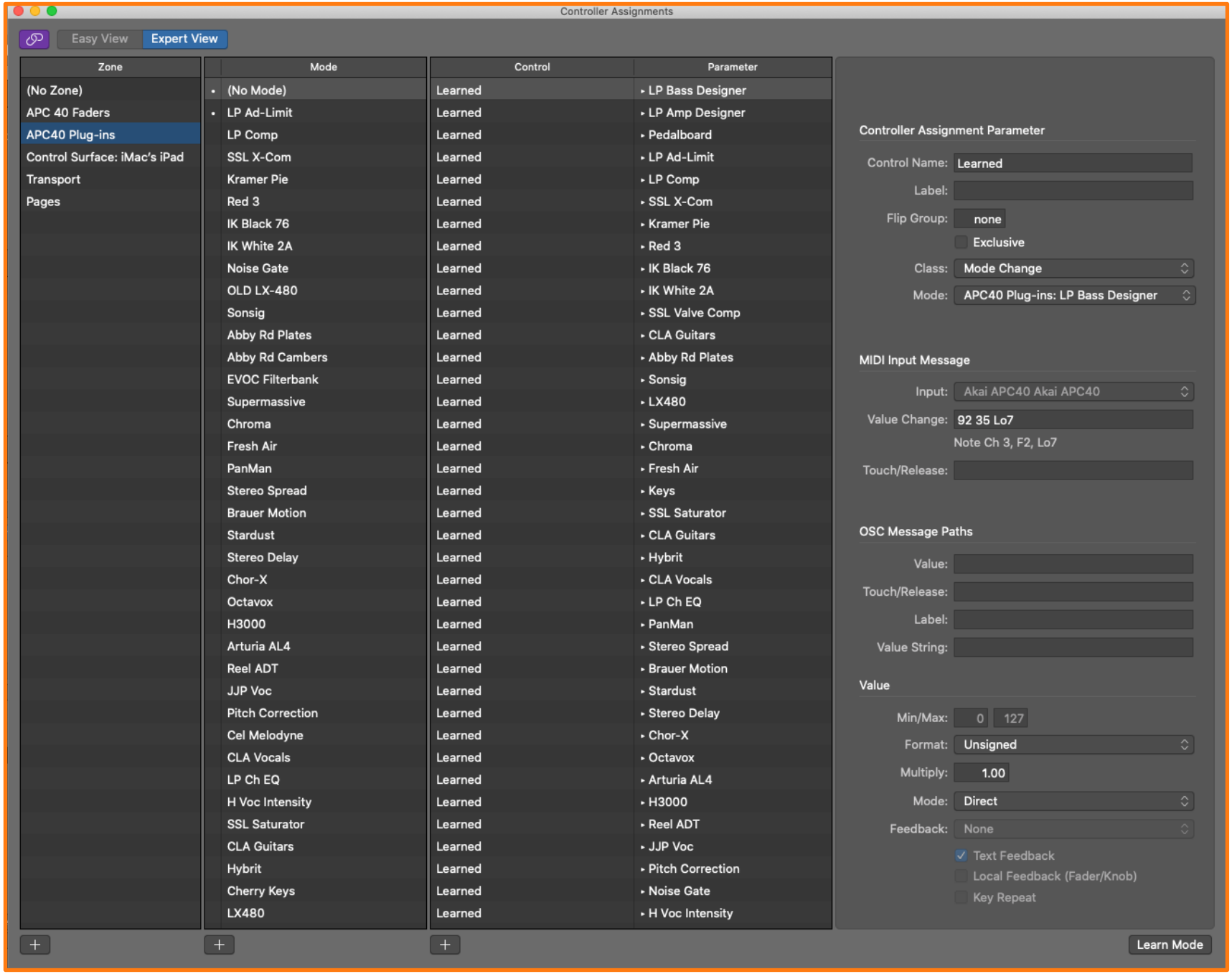
When you do that, a Parameter is automatically given a name that describes what it actually does. When you have multiple Parameters with the same name, you can manually name the Control to remind you what it does.

The column to the far right of the CA window essentially has two functions. First, it shows you in detail what a given Parameter does. Just as importantly, its second role is to allow you to alter an existing Parameter using the pulldown menus, or manually writing in the MIDI Value Change etc.

Important note: When creating or manually altering Zones, Modes, Controls and Parameters via the CA window, Learn Mode should be off. Learn Mode is a bit like Automation. You only use it when you want LPX to copy something you've just done, like a fader move. In Learn Mode, LPX is waiting for an incoming MIDI signal, so if you activate it at an inappropriate time, you are likely to get something pretty random!

If you take a closer look at the right-hand column, you'll see it's divided into four areas. These cover: Controller Assignment Parameter; Input Message; OSC Message Paths, and Value.

In the next chapter, we'll look at what Zones, Modes, Controls and Parameters really mean. □



The CA window

If you don't see the full range of parameters in the right hand column when you click on a Mode, it's probably because you don't have the CA window open wide enough. Although it's possible to scroll up and down the right hand column, it's probably easier to open the window fully.

We'll look at this in more depth in the next chapter, but now let's just say that Zones contain Modes. Modes contain Controls. This hierarchy is a bit like files inside folders, which are themselves nested inside folders.

Or, if a Zone is a cook book, Modes are like chapters and Controls are individual recipes. Sort of!

Zones, Modes, Controls and Parameters – what the heck?

What you'll learn in this chapter:

You'll start to get a feel for Zones, Modes and Parameters are all about. Just as importantly, you'll get an insight into how to structure and organise your CAs, so that they do what you want them to do.

In this way, you can use a MIDI controller to address vastly more plug-ins than there are physical controls on the device.

Once you get a grasp of how you can structure your CAs, you'll find it easier and more reliable when you tell LPX to Learn something. You'll find it easier to see what's going on when you look at the CA window, and you'll know what you did when you come back to your CAs months later, when you want to add to them.

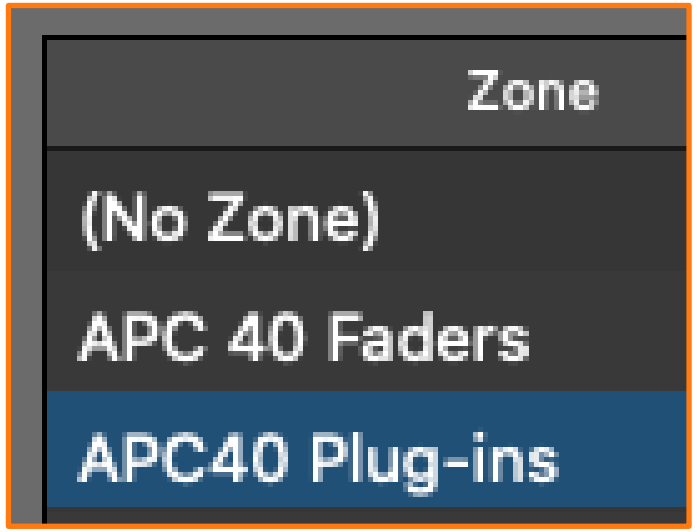
If you find this chapter difficult to absorb, please don't worry, or start to think this is all above your pay grade. That's pretty much the place we all started from! As you gain more hands-on experience, and start to create more ambitious assignments, it will all start to make more sense. When that happens, you can always revisit this section of the book for reference.

As a bare minimum, I'd suggest you at least become familiar with Zones, Modes and Controls, even if you leave Parameters for another day.

What is a Zone?

A Zone acts as a container for a group of physical controls that typically share a purpose. Those controls aren't necessarily on the same MIDI device. For instance, if you have a MIDI controller that has eight faders, plus a MIDI keyboard that also has eight faders, you might create a Zone called 'Faders 1-16' that comprised all 16 physical faders. The CAs for those faders would reside within the 'Faders 1-16' Zone.

In one sense, Zones serve an organisational purpose. You wouldn't put all the letters a company sent or received in one filing cabinet



folder marked 'Letters' because it would be impossible to find a specific letter in a hurry. If the company was a law firm, you'd probably have a folder for every individual client, so if David Morris phoned to ask about his property purchase, you could go straight to all his documents – and only his documents.

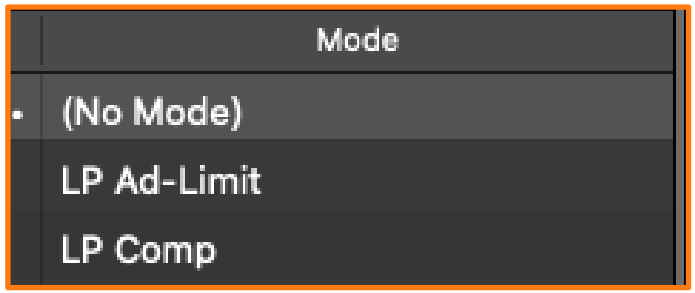
So, you could say Zones are a bit like drawers in a filing cabinet, because they keep things tidy and easy to find again.

You may be wondering at this point: "So how do I define what physical controls are in my Zone?" The short answer is: you don't. When you have a Zone selected and you create a CA, whatever physical control on your MIDI controller you assigned is now part of that Zone.

Although that might sound a bit complicated, it soon becomes intuitive once you get into the swing of creating your own CAs.

Continued over...

From previous page..



What is a Mode?

As suggested above, if you think of a Mode as being a folder inside a filing cabinet drawer called a Zone, you are understanding the underlying structure of CAs.

A Zone can contain one or more Modes, but only one Mode is active at any one time. Modes contain Controls, or CAs, as I tend to think of them. It's by switching between Modes, that the physical controls can take on completely new CAs.

To give a simple example, let's say button 1 on your MIDI controller is assigned to a Mode called EQ and button 2 is assigned to a Mode called Compressor. It may well the that the first knob on your MIDI controller is assigned to equaliser LF cut/boost when you press button 1 and compressor Input Gain when you press button 2.

What is a Control?

If a Zone is like a drawer and a Mode is like a folder, then Controls are like a document that contains a list of instructions.

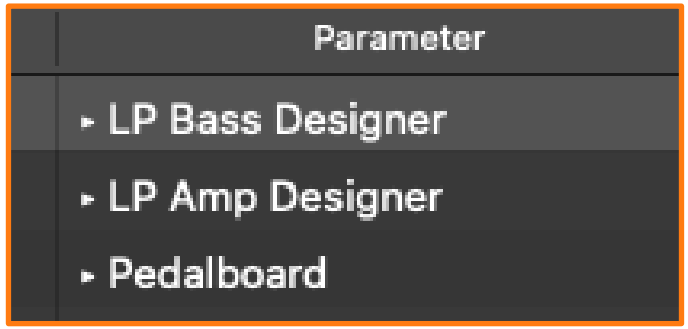
Imagine you've taken your car into a garage for a service. The mechanic might open a drawer in the filing cabinet that contains all the service manuals, then go to the folder for your make and model, then look for the service document for your car. In reality, there would probably be several documents relating to servicing your car, depending on how many miles it had done etc. Well, it's much the same for Controls.

Let's say you've created a Mode dedicated to the LPX Linear EQ. If you include the High Cut and Low Cut, the plug-in has eight bands,

all with three variable parameters. In order to assign that little lot, you will need 24 Controls, each with its own Parameter. (Actually, there is at least one way to use less physical controls, but let's gloss over that for the time being.)

What is a parameter?

Strictly speaking, Apple's use of the term is a bit misleading. Generally, we would take 'parameter' to mean a single numerical value that varies from case to case, although it can



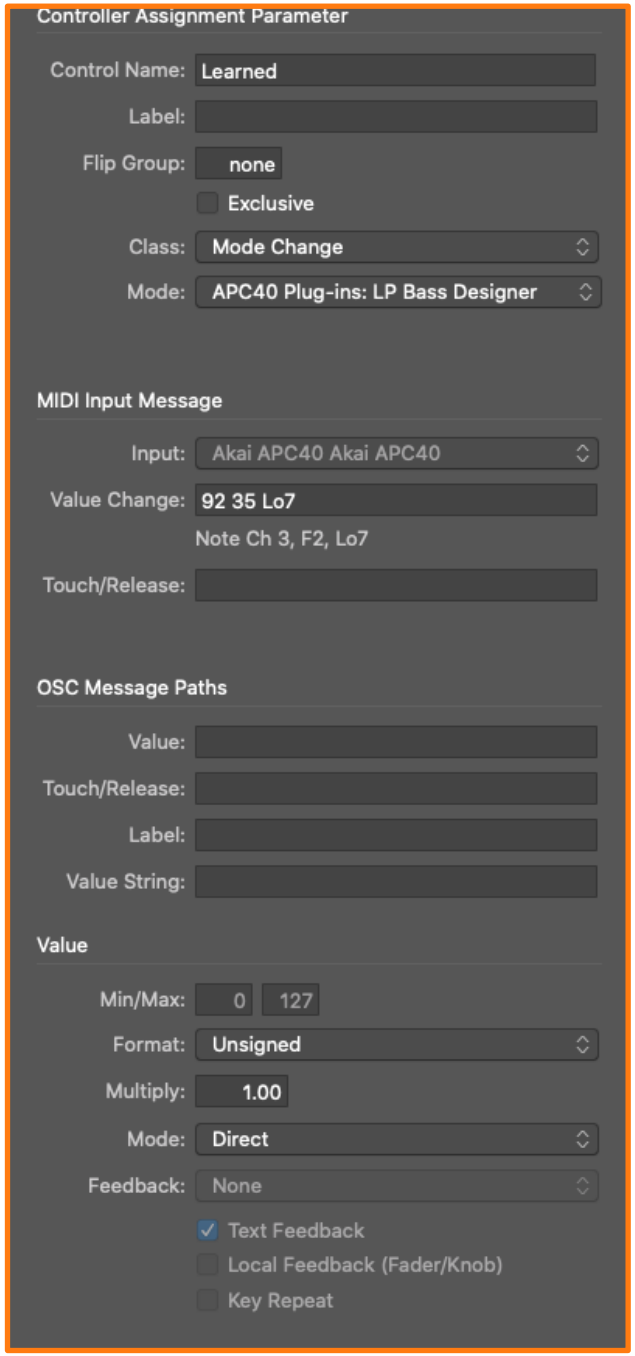
also be used to describe a characteristic that is used in defining the nature of a system.

However, when Apple uses the term Parameter in the Controller Assignment window, it really means a whole bunch of characteristics. Prepare to have your mind well and truly blown if you don't already know how extensive that list is! Basically, if you can do it with a keystroke, mouse-click or mouse-drag in LPX, the chances are you can create a CA that does it using a MIDI controller.

While that makes CAs incredibly powerful, it also means the documentation needed to cover every possibility would be huge. The summary below really only scratches the surface, but I would encourage you to explore the pulldown menus in the Controller Assignment window. The scope of the Class and Channel Strip menus is vast!

Here is a breakdown of all the fields and pull-down menus shown in the right column in Expert View. If you don't want to dive in this deep today, you can jump ahead to Chapter 5 – *Where Key Commands and Smart Controls fit into this.*

Continued over...



Parameter fields: the deep dive

From previous page..

Control Name: If you are using a supported MIDI controller – ie one that is listed in Logic Pro/ Control Surfaces/Setup – the name of the controller will show in this field. If you are using an unsupported controller, it will simply read 'Learned'.

If you want to, you can rename this field. It doesn't do anything other than display a name. As we'll see when discussing the next section, the name of the controller you're using for the assignment is shown there, so there's sometimes no real point in writing it in yourself. But you could use the Control Name field to store a brief description of what the CA is for.

Label: In many instances, this field will remain blank. However, for *supported* devices that have a display, what's written in the Label field determines what's displayed on the controller.

On the most basic level, that means if you write 'Gain' into the label field, you'll see 'Gain' on the MIDI controller's display.

But it goes deeper than that, because it is possible to create placeholders for dynamically generated text. In other words, the display message can change because of a variable, such as which channel number is the selected one.

If the field simply contains fixed text, it can contain any number of characters (although that's not to say the controller will be capable of showing them all). When used as a placeholder for dynamically created text, however, the field must three characters that represent the label. These go after whatever text you use to describe the function of the control, such as Aux, Pan and so on.

The first character is always @, followed by two additional characters. For example, 'Send@s#' will display as 'Send1,' 'Send2,' etc.

The second character is used to define the type of event label. For example: t=track, r=surround etc. While the list below is the most exhaustive I could find, it is possible that there are others in use.

Second character	Meaning
t	Track
r	Surround
s	Send slot
S	Number of sends
e	EQ band
E	Number of EQs
p	Insert slot
i	Instrument

The third character is used to define a value for the chosen event label type, such as the track number or name. For example: #=number of track.

Continued over...

From previous page..

Third character	Meaning
#	Number of above (track number, Send slot, EQ band, Insert slot)
n	Name of above
p	Name of parameter addressed by the assignment
P	Name of first parameter
o	Parameter offset, counted from 1
O	Maximum parameter offset, counted from 1
b	Parameter bank (= parameter offset/bank size), counted from 1
B	Total number of banks (= parameter offset/bank size), counted from 1

The following is beyond my personal experience and is a direct quote from Apple's own documentation.

Flip Group Field

Enter the same integer value for two assignments, to define a counterpart for Flip mode (for supported control surfaces that offer Flip mode). By setting a fader and an encoder to the same flip group, for example, they are coupled. To set 'none', enter a value of 0.

For unsupported devices, you need to set up two active assignments, both of which use the same flip group. One assignment needs to be absolute (using a fader, for example), the other relative (encoder, for example).

Exclusive Checkbox

When selected, the assignment deactivates all other assignments that have Exclusive turned off for the same control (on supported control surfaces). This limits the overwriting of a modeless assignment to particular modes. For example, faders normally control volume. To create a mode where faders control the send level, select Exclusive.

The **Flip Group** field and **Exclusive** checkbox are at the top right of the Expert View window. Typically, you would use a Flip button on a supported controller to assign a new function to the faders. Doing this has the advantage of allowing precise control of, for instance, Aux mixes. Where the controller has motorised faders, they also make updating automated moves much easier.

If you've been attempting to read this book from start-to-finish like a crime novel, you probably feel as if you're brain's about to explode by now! It's probably a good idea take a break before reading the next bit, because it details parts of the CA window you'll be using a lot when creating your own assignments.

Class is set using the pull-down menu. Apple describes Class as "the type of destination parameter controlled". If that doesn't exactly resonate with you, it's OK to think of Class as being "an important part of setting what a CA is actually going to do. Believe me, Class is a lot easier to understand when you actually get hands-on with creating some Assignments.

The Class Menu allows you to choose between seven types of class: Mode Change; Global; Channel Strip; Key; Key Command; Control Surface Group, and Automation.

Mode Change: A major use for a Mode Change is to switch between Modes that have

Continued over...

From previous page..

related functions. Let's say, for example, your controller has eight rotary controls, but you want to create CAs for all three parameters across the eight bands in the Parametric EQ plug-in. The first Mode you create might contain the CAs for cut/boost. The second Mode might be for the frequency sweeps, while the third mode would be for adjusting the Qs. Assigning a Mode Change to a button will allow you to switch between Modes, giving you control of all 24 parameters. This is the trick I briefly mentioned on p15.

Note: The Mode Change assignment *must* be located in the same Zone.

Global: As the name implies, this Class is used for CAs that affect the whole of the LPX main window, typically whereabouts on the timeline the Playhead is. Although the available Parameters are reasonably self-explanatory, there is a full list in the official Logic Pro User Guide, along with more detail on what each one does.

Channel Strip: There are a number of ways this Class can be used to specify which channel strip is being addressed. Arguably the most commonly used of these options from the pulldown menu is 'Selected track'.

When this is used, only the highlighted track/channel's controls are altered. This is how you can, say, adjust the EQ on Channel 6, while every other EQ remains static.

Another common option is to use a parameter such as 'Output', which allows channels to be specified by number. As we'll see in Chapter 7 – *24 channel mixer with graphic EQ*, It is possible to very quickly assign a group of controller faders to multiple faders in LPX's Mixer environment.

The complete list of possible Channel Strip parameters is found in the official Logic Pro User Guide.

Key: According to Apple, the Key Class enables you to use a CA to emulate any key on the QWERTY board using a MIDI controller. To be honest, Key Command already covers anything you're likely to want.

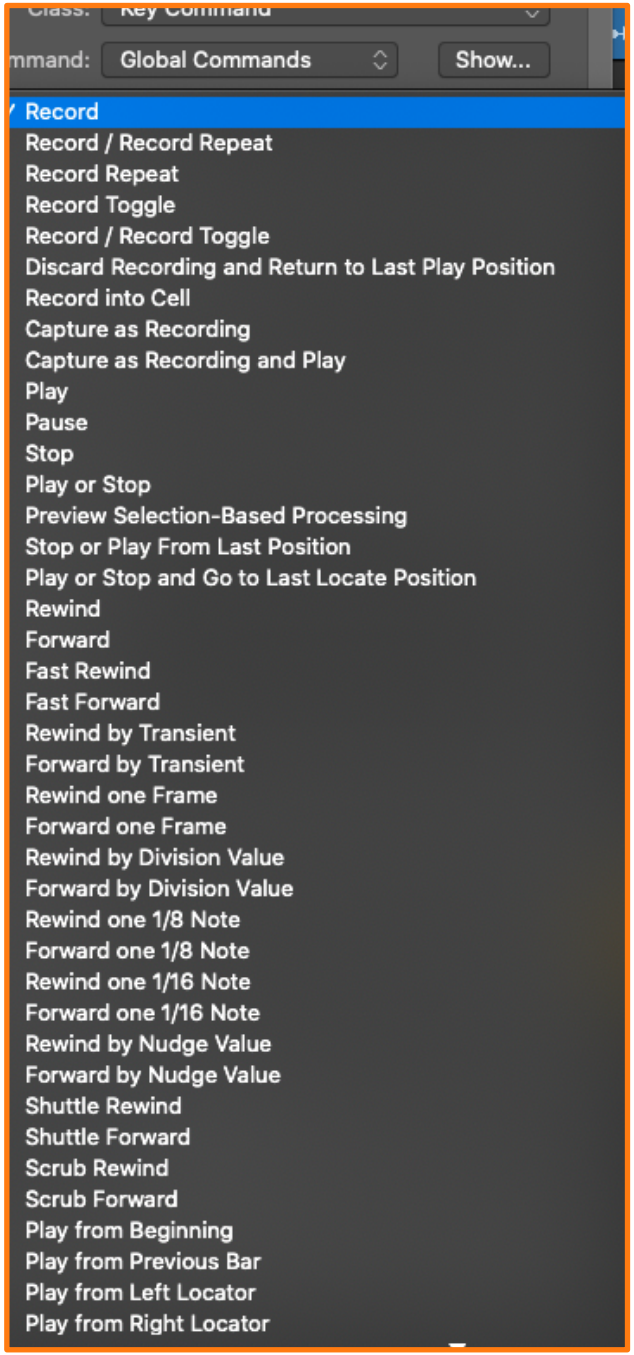
Key Command: Any function in LPX that can be given a QWERTY key command can be given an equivalent on an external controller. Common instances include assigning the transport controls to functions such as Play and Stop, as covered in Chapter 2 – *Transport control made easy*.

Actually, Key Commands are more powerful than that. Not only can you use them to control LPX, but also aspects of the Mac's functions outside of LPX, as I'll demonstrate.

As you can imagine, the total list of functions within the pull-down menus is huge. You want the A2 key on your musical keyboard to open the Piano Roll window? Be my guest. When you've finished having your mind blown by the seemingly endless list of possibilities, you're likely to notice that the Command pull-down menu has no fewer than 27 different types of Key Command, each with its own multiple option menu (see right).

By far the best way to get a feel for the scope and potential of these menus is to play with

Continued over...



From previous page..

them to see what they do. For instance, I don't find that much use for the Play buttons on my controllers, because I'm so used to tapping the space bar on my QWERTY key board for Play/Stop. Just for fun, I changed the Play CA from *Global Commands/Play* to *Tools Menu/Pencil Tool*. You got it: now every time I hit the Play button, I get the Pencil Tool instead of the Pointer Tool.

This is just an example, but it doesn't take a lot of imagination to see I could create CAs for all the tools is I wanted to. The further you get into this, the more you will see that you really can create custom controller configurations that are individual to you, and the way you like to work.

Note: You can also create CAs for Key Commands in the Key Commands window (Alt-K). Which one to choose and why is discussed in Chapter 5 – *Where Key Commands and Smart Controls Fit Into This*. Another way to access the Key Commands window is from the Controller Assignments Window. Simply press the Show button to the right of the Command menu.

I said I'd demonstrate that Key Commands can control some aspects of the way the Mac behaves beyond LPX. Better than that, I'll show you how to prove it for yourself. For this example, I've reconfigured our old friend the Play button using *Key Command/Windows Showing Audio Files/Show File(s) in Finder*. Try it for yourself.

When you've written the CA, go to the Arrange window and click on any audio file on the timeline. Then press Play. Now, instead of putting the arrangement into play, it opens a separate window, showing where the file actually is! If that folder happens to contain other audio files that are going to be used in the Project, it's quite a little time-saver.

Alternatively, holding Command and clicking on the title at the top of the window will reveal the entire path to the file; a trick I used to find very useful working on magazines with people who didn't always organise their work as carefully as you'd hope.

Control Surface Group: These are similar in concept to a Zone. Each Group comprises two or more control devices, which are going to act as a single control surface. A typical situation where this applies is if you have a controller and extension units, which you want to be a single mixing desk.

The device parameters you can set include Fader Bank Offset. This determines the physical faders that control given channels. (If you think about it, LPX doesn't 'know', which of three mix surfaces is the one in the middle, which one is on the right... unless you provide that information. In that respect, it's a bit like setting up two monitor screens, so that left and right are set correctly.)

While Control Surface Groups are powerful tools for getting the most from large-scale set-ups, they are well beyond the scope of this book. My aim is to get you up and running as easily as possible, rather than exploring every possible parameter in detail.

Automation Group: I would argue that for the vast majority of users, this feature is now like taking a sledgehammer to crack a nut. While it is obviously invaluable in a large mix to be able to group all drums under the control of a single fader, LPX provides several easier ways to achieve this, such as creating a Summing Stack. However, you may wish to investigate Automation Groups as a way of specifying which hardware fader is assigned to that Summing Stack fader.

I strongly suggest that you get hands-on experience with successfully creating more basic CA setups before you dive into control Surface and Automation groups.

MIDI Input Message

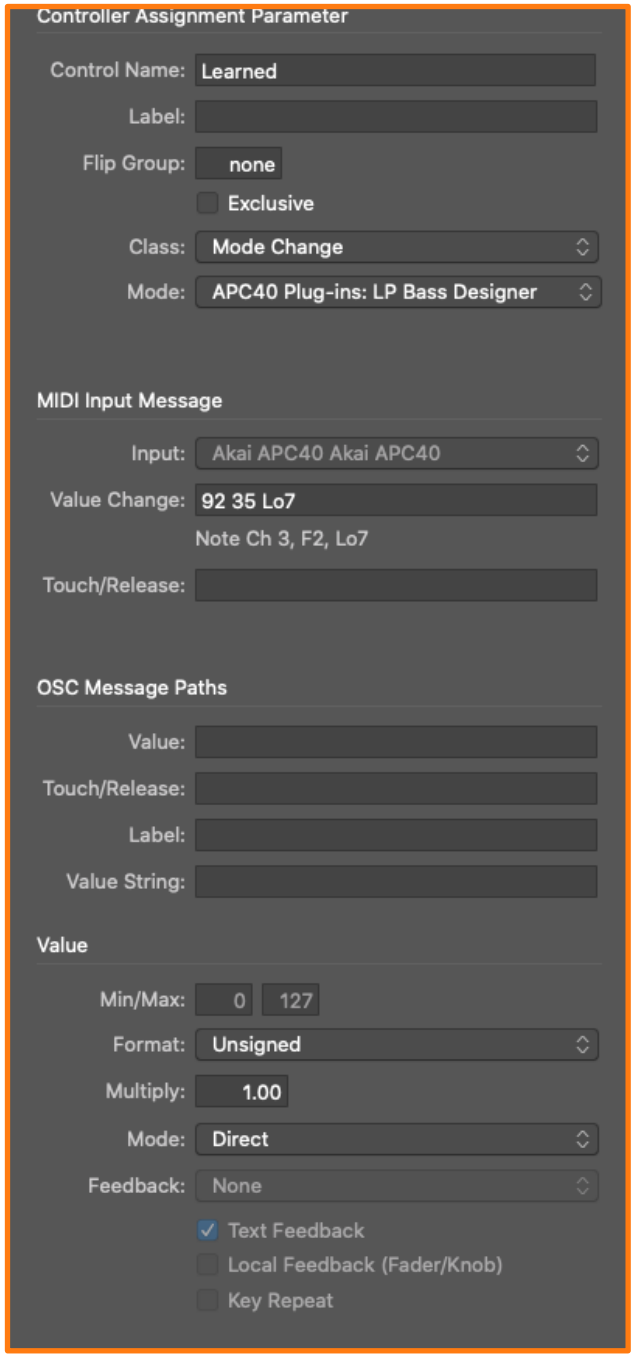
You may be relieved to know that this section of the CA window is relatively straightforward, certainly compared to the Controller Assignment Parameter section above it.

Input: This is simply the physical MIDI device you are sending the control message from. So, if you have more than MIDI device connected, you select the one you intend to use from the pulldown menu.

Value Change: This is the MIDI message that the controller will send to execute this CA. Apple helpfully provides a translation below the field, so if you didn't know 90 5B Lo7 was Ch1, G5, Lo7, the CA window spells it out for you.

On the other hand, if you know MIDI inside-out, you can actually enter the Value Change in manually. I very rarely do this, for fear of making a mistake, and rely on the Learn function. Some

Continued over...



From previous page..

experienced users actually prefer to write the Value Changes themselves.

Touch/Release: This applies only to the Channel Strip assignment class and to parameters that can be automated, so for many CAs, this field will be blank. If you enter a number, 0 signifies touched, and any whole number above that means released.

OSC Message Paths

Apple has this to say: "The OSC protocol allows for real time communication between software and hardware over a network, or when using USB or other serial interfaces. The current OSC implementation in Logic Pro supports UDP and IPv4 network connections only."

Most users need never know what any of that means, and should leave those fields blank.

Value

Min/Max: As with velocity on a MIDI keyboard, the value here can range between 0-127. Normally, that's what you'd want it to be, because if the upper value is less than 127, you'll generally that you can never reach the full range of an on-screen control, even when your physical control is maxed out.

Sometimes, if you've been a little sloppy with your physical movement when in Learn mode, LPX misinterprets your intentions and writes in, perhaps, 0-99. If that happens, it's usually better to enter the correct value in the field, rather than relearn the assignment.

Note – There are a few controllers that use the same message – but with different value ranges – for different controls. Please refer to your User Manual if your device works that way.

Format: Most of the time, you can leave this to Unsigned, which signifies that no negative values are possible. Again, there are controllers that use one of the other formats – 2's complement, 1's complement, or Sign Magnitude – and any information regarding this is likely to be in the manual.

Multiply: As the name implies, this will scale the incoming MIDI. Most of the time, you'll want it set to 1, meaning the MIDI value is unchanged. Apple cites two examples of times when a different value is appropriate:

- To set the automation mode to Write, set multiply to 4.0 and Mode to Direct.
- To decrement a parameter by 1 with a button press, set multiply to -1.00 and Mode to Relative.

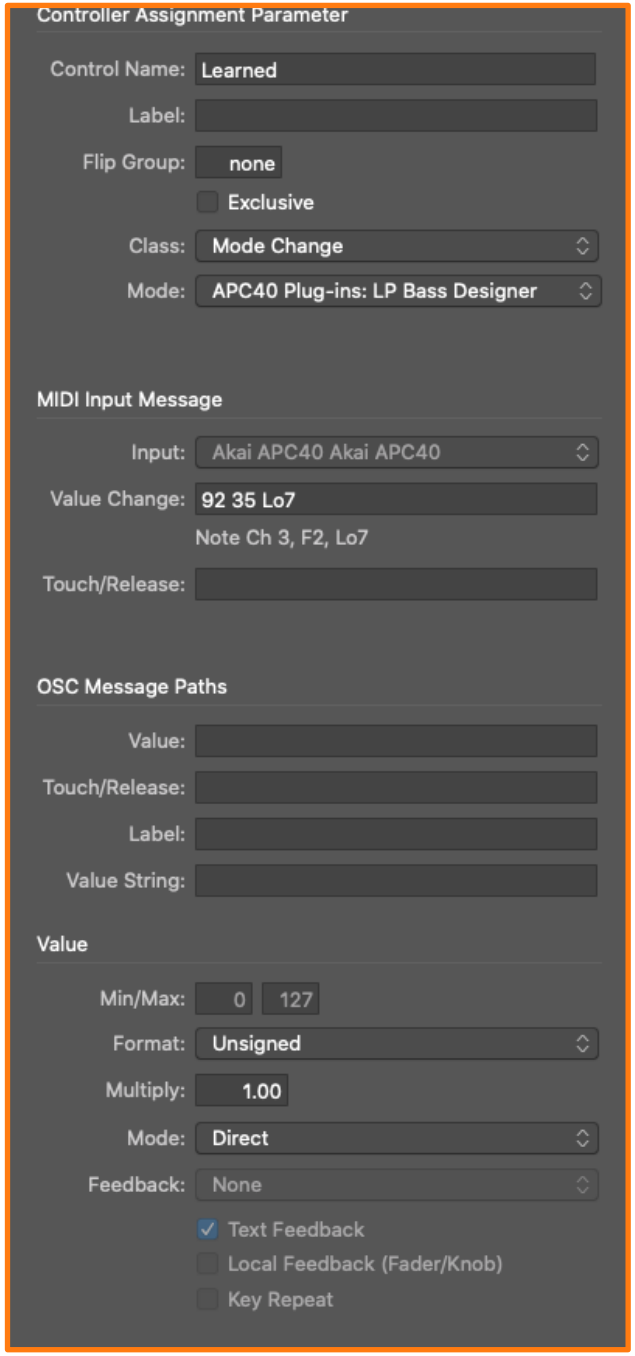
This brings us on to...

Mode: There are six possible Modes selectable from the pull-down menu. If you're ever unsure which one is appropriate, you can do no harm by trying each one in turn.

It's worth bearing in mind that the physical controller will also dictate what any given knob or button can do.

- Direct – the incoming value is the one used for the parameter.
- Toggle – Mutes and Solos are good examples where you want to be able to switch them on if they are currently off and vice versa. For on/off assignments, the mode is set to Toggle by default.
- Scaled – The incoming value is scaled to the range of the control, so that you get the

Continued over...



From previous page..

full range of a fader, for instance.

• Relative – This can be used for encoders, or buttons that you want to increment/decrement by a fixed amount. Here the incoming value is added to the existing value, so it's a +/- function, basically.

• Rotate – This is mainly used when you want to create a button that cycles between modes. Apple cites the various automation states (ie Read, Write etc) as an example, but there are plenty of plug-ins that have a list you might prefer to increment through, rather than control them from a rotary. The Output Model in the Tube EQ is an example.

• X-OR – This is apparently a Boolean operation, that can be useful for enabling or disabling single channel strip types in All view. If you are smart enough to use this one, let me know and I'll happily buy *your* book!

Feedback: I've always left this set to Automatic. When I move a rotary on screen, so does the LED 'skirt' on the related rotary on my Akai APC40 – and vice versa – so that one works for me. My Korg nanoKontrol doesn't display any feedback, but Automatic works fine.

However, Feedback isn't just about displaying names on readouts, and getting a few LEDs to dance around, so you may want to see the part about Key Repeat, even if your controller doesn't have any feedback displays.

Feedback types

None – No feedback is sent.

Single Dot/Line – For LED rings, only one LED. For LCDs: a single vertical line.

Left to Right Bar: A bar from the minimum to the current value.

Bar from Centre – A bar from the centre position to the current value.

Right to Left Bar – A bar from the current value to the maximum.

Q/Spread – Two equal bars from the center to the current value.

Ascending Bar – A bar from the bottom to the current value.

Descending Bar – A bar from the top to the current value.

Text Only – For LED rings, no feedback. For LCDs, no feedback as a graphics element.

Automatic – Depending on the currently assigned parameter, the most suitable feedback mode is used: Plug-in and Instrument parameters carry this information, Pan displays a single dot or line (Single Dot/Line), and all other parameters display a bar that runs from left to right (Left to Right Bar).

There are also three checkboxes:

• Text Feedback – Turn on to send a text representation of the current value to the control surface display. The control surface display capabilities determine the display position and number of characters used.

• Local Feedback (Fader/Knob) – Turn on to stop feedback while the parameter is in Touch mode. This is to stop motorised faders fighting against user movements.

• Key Repeat checkbox – Turn on to

Continued over...

From previous page..

repeatedly execute the assignment at the rate set by the Key Repeat slider in macOS Keyboard preferences. The duration the button must be held for before the assignment is repeated is set with the Delay Until Repeat slider in Keyboard preferences.

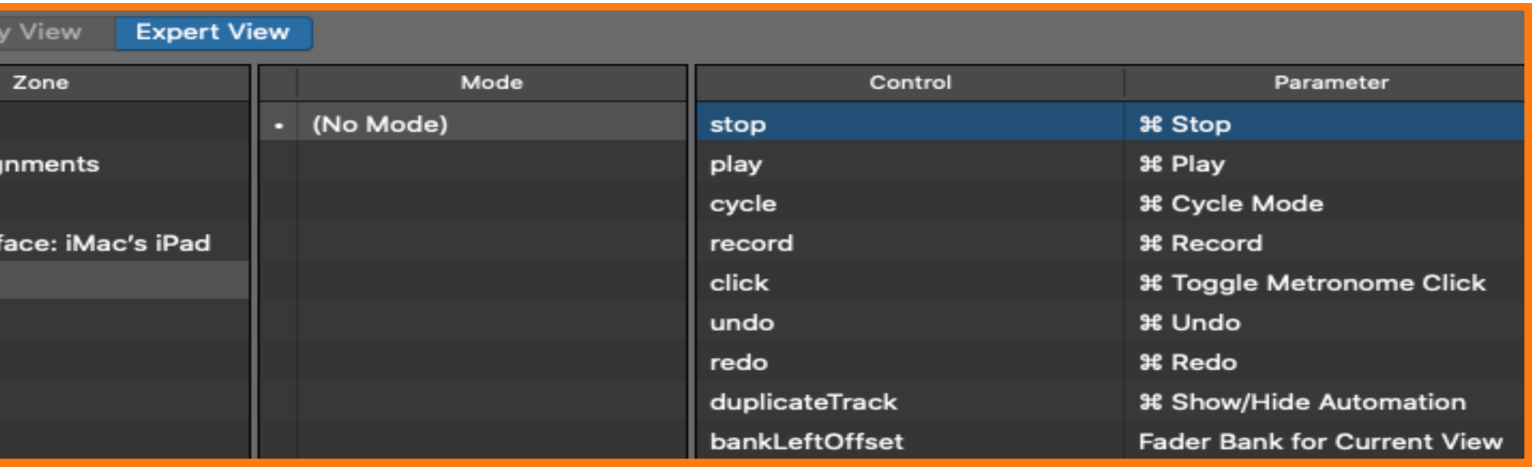
Apple states:
" Key Repeat is useful for the zoom function. For example, if you assign a Key Repeat command to the Mackie Control Zoom buttons, holding the Zoom In button will continuously zoom in the Logic Pro window until the button is released. The alternative is to repeatedly press the Mackie Control Zoom buttons to zoom in or out more than one level.

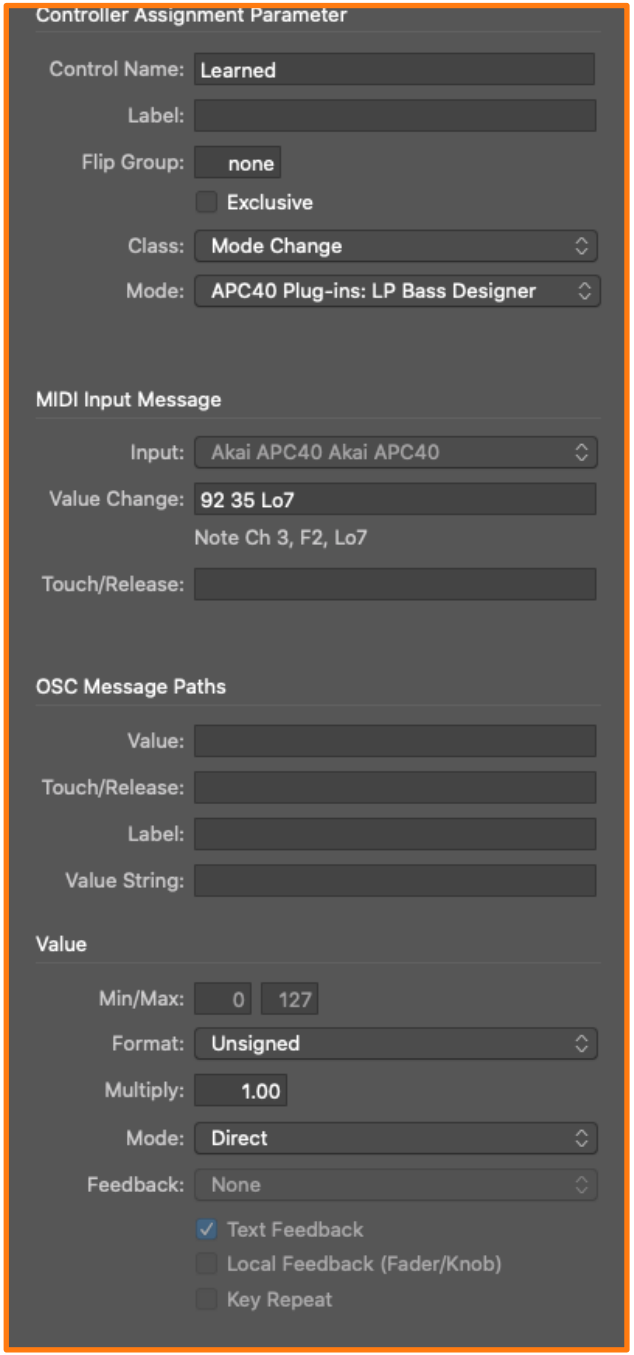
"Note: The Key Repeat checkbox is available only for key commands, key presses, and relative value changes. If any other assignment class is selected, the checkbox is dimmed.

"The default key command assignments support the Key Repeat function if useful or applicable to the control surface or device, making changes unnecessary to use this function. If you want to enable the Key Repeat function for your own assignments, you may need to use the re-learn option for the assigned message.

"Key Repeat messages must include the Lo7 byte, which provides information on the up (released) or down (pressed) state of the assigned button. Logic Pro guides you through the re-learning process. The current MIDI message is automatically cleared, Learn mode is activated, and a help tag prompts you to send the required MIDI message.

"Releasing the assigned button – after learning the MIDI message – automatically creates the Lo7 byte, and assigns the Lo7 value (for the button release message) to the Min parameter. The Lo7 value for the button pressed message is automatically assigned to the Max parameter. Typically, the value range of 1 to 127 is used for the button pressed message. The zero (0) value is generally used for the button release message." □





Where Key Commands and Smart Controls fit into this

What you'll learn in this chapter:

Discover how pretty-much every function in Logic Pro can be assigned a Key Command. (This can save you a lot of time if there are functions you use frequently.)

You'll also explore the Smart Controls, in terms of what they do, what the advantages are of using them and what their relationship is to Controller Assignments.

NB – Although you can get a long way without Smart Controls, you'll discover that they sometimes allow you to assign to plug-in controls that seem to be 'invisible' to the CA window. So, assign a Smart Control to them, then Learn the assignment. Clunky, maybe, but it can work.

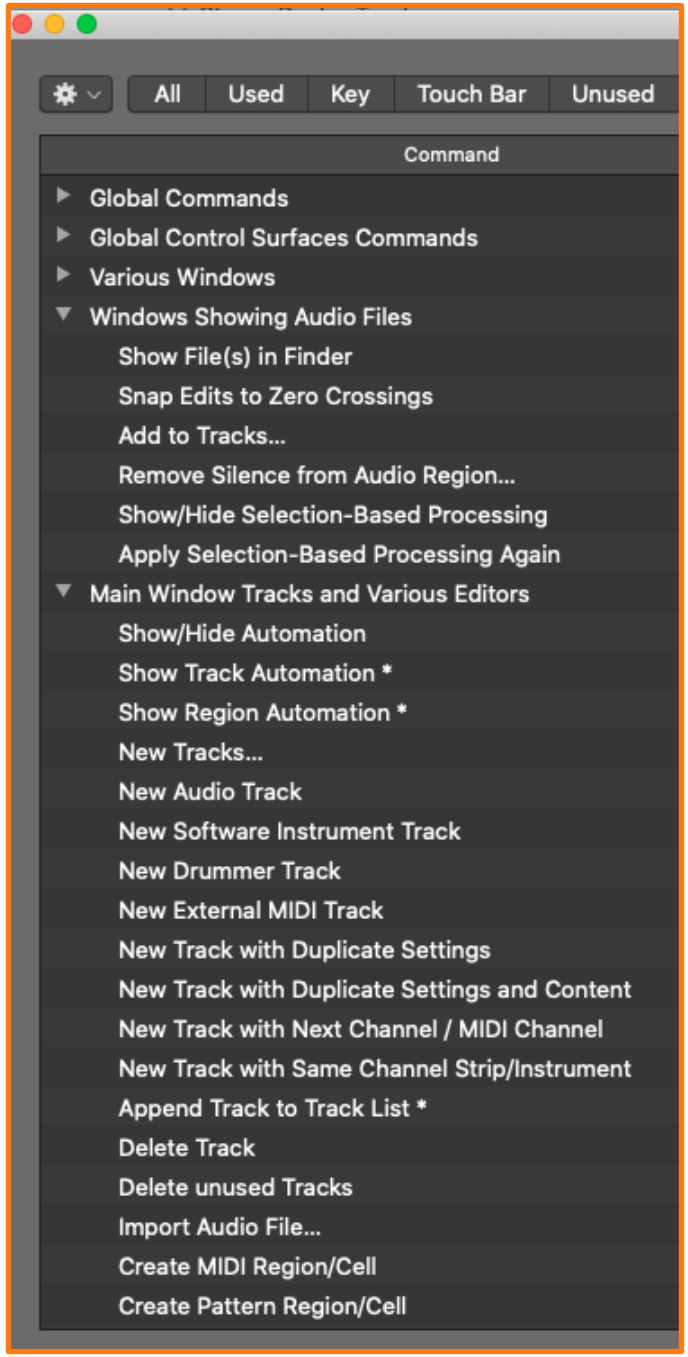
You've probably noticed that Logic usually provides you with at least two ways of achieving something. For instance, if you can't remember the Key Command for something, you can always use the drop-down menus, or press a button on screen. Similarly, if the Score Window doesn't tell you what you want to know, the Piano Roll shows the same information, but in a different way.

It's much the same with the relationship between the Controller Assignments window and the Key Commands edit window. In fact, we learned in the previous chapter that you can get to Key Commands from within the Controller Assignments window. That's a strong clue that they have closely related functions. However, there are also important differences.

Although there are a vast number of them of them, the Key Commands dialogue (Alt-K) shows exclusively the commands that it is possible to assign a Key Command to. These are divided as shown on the right, and each category contains anywhere between six commands, in the case of Windows Showing Audio Files, to hundreds in the case of Global Commands.

I don't know how many possible Key

Continued over...



From previous page..

Commands there are, but it must run into the thousands. Due to the sheer number, the dialogue window provides buttons for filtering the data according to what you are looking for. Note that it is possible to use more than one button to create more specific filtering criteria.

It is also possible to create assign a MIDI control to a Key Command, and the dialogue window includes buttons to Learn New Assignment and Delete Assignments. It also divides Commands into columns for Key, Touch Bar and Assignments. In addition, it is possible to import Key Commands, which is useful if transferring to a different computer.

In short, the Key Commands dialogue provides a comprehensive and flexible tool for managing those great productivity enhancers, but you cannot use it to write Controller Assignments for, say, a plug-in or fader.

On the flip side of the coin, the Controller Assignments window can only learn CAs from a MIDI device. QWERTY keystrokes are simply not recognised (in my experience).

So far, I haven't said anything about Smart Controls. That's because the relationship between that environment and the Controller Assignments window is more complicated, and therefore a bit harder to explain. While Key Commands do not show up the CA window, any assignments Learned in the Smart Control environment can also be seen in the CA window. If you delete an assignment from the CA window – and it was created using Smart Controls – it will cease to work in the Smart Controls environment.

What this tells us is that the External Assignment Learn function is actually writing CAs. However, it's only when we open the CA window and move the MIDI control in question that we discover where in the structure the CA has been placed. Usually, it's in No Zone. While it is obviously possible to use the CA environment to move assignments that were created within the Smart Controls environment, this adds a level of complexity that is probably best avoided. For that reason, we'll be working almost exclusively with the CA window in Expert Mode for the rest of this book.

Smart Controls have a value, mainly in letting you quickly adjust key parameters on plug-ins without having to open them. But, from the perspective of managing an hierarchical structure of multiple Zones, Modes and Controls, the CA window is the appropriate environment.

Should you wish to use Smart Controls with Controller Assignments, I'd recommend that you still use the CA window for managing and creating your CAs, but you'll also need the Smart Control Window to click on the control you want to assign to.

For some strange reason, if you write a CA to, say, a plug-in's Resonance parameter in Smart Controls, the same assignment will move the reciprocal Resonance control on the plug-in itself. That's what you'd expect. But if you write the CA direct to the Resonance control on the plug-in, the reciprocal Smart Control stays still, even though you can see the Resonance control on the plug-in move as you tweak the rotary on your hardware controller.

It's just the way it is: if you want to use Smart Controls with CAs, you have to assign to the Smart Controls, not the plug-in. □

Note that Smart Controls can address more than one plug-in in a channel strip, as shown below, where the SSL LMC+ and Sonsig REV-A are both controlled

Preparation and planning

I'm guessing that you're reading this book because you've tried writing CAs in the past, but it didn't quite go as planned.

Maybe you got some moments of elation as you saw the controls on screen respond to physical controls on your MIDI device. But then, as you tried to expand the scope of your CAs, things started to go wrong.

Some of your assignments probably ended up in the wrong place. So even though you started out in the Mode you intended, your CAs ended up in No Zone, or somewhere else that made no sense.

Then, when you went back to it a few days later, maybe it didn't work the way you thought it did, so you got fed up with the whole business.

After a few months of working with the QWERTY keyboard and mouse, you went back to trying to set up CAs, but this time, things made even less sense, and you started to question whether this part of LPX even worked properly. What if you're just wasting time? On a 'hiding to nowhere'.

If any that sounds vaguely familiar, that's because it's a fair summary of my first few experiences. Don't worry, it doesn't have to be that way.

The reason most first attempts at CAs don't really work out isn't because the users are all stupid, or CAs are too hard for anyone who isn't an experienced programmer. Not at all. I can barely program the central heating!

Hardware set-up

Because there are going to be times when you have multiple windows open, it is very helpful to have two – or more – monitor screens. If you only have one screen, so be it. You just have to be a little more careful about how you manage the real estate, and you probably find yourself switching windows a little more often.

Of course, you'll also need at least one MIDI device to act as controller(s). At the risk of pointing out the scaldingly obvious, the MIDI gear needs to be on, connected to USB ports and LPX needs to 'know' it's there. Sometimes, it's possible to get so focused on "he hard bit' – like creating your first Controller Assignments – that you overlook the basics. So if LPX doesn't seem to be responding to a MIDI device, you'll need to troubleshoot that before you can get any further.

Software set-up

On a related note, I'd seriously recommend you to download MIDI Monitor from Snoize. https://www.snoize.com/midimonitor/ This free utility will show you what MIDI Message, Channel and Data each key, button, fader and rotary control on your MIDI device is sending.

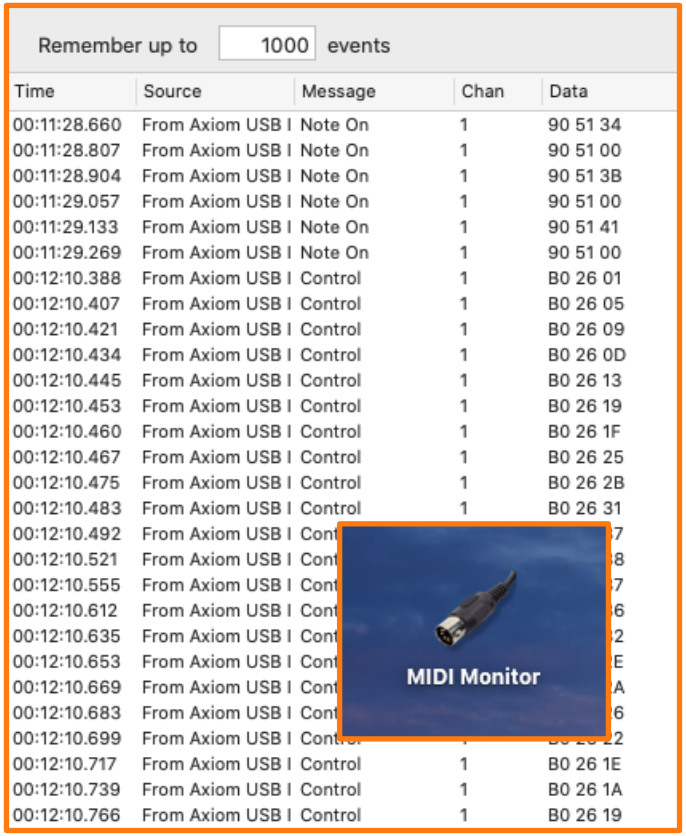
This is particularly valuable if you are using MIDI devices that you haven't owned from new. At the very least, you can find out whether

Continued over...

What you'll learn in this chapter:

Believe it or not, this is actually the most important chapter in this book. Why? Because it holds the keys to your CAs going just the way you hoped, giving you hardware control, just the way you want it.

MIDI Monitor is your friend…



Time	Source	Message	Chan	Data
00:11:28.660	From Axiom USB I	Note On	1	90 51 34
00:11:28.807	From Axiom USB I	Note On	1	90 51 00
00:11:28.904	From Axiom USB I	Note On	1	90 51 3B
00:11:29.057	From Axiom USB I	Note On	1	90 51 00
00:11:29.133	From Axiom USB I	Note On	1	90 51 41
00:11:29.269	From Axiom USB I	Note On	1	90 51 00
00:12:10.388	From Axiom USB I	Control	1	B0 26 01
00:12:10.407	From Axiom USB I	Control	1	B0 26 05
00:12:10.421	From Axiom USB I	Control	1	B0 26 09
00:12:10.434	From Axiom USB I	Control	1	B0 26 0D
00:12:10.445	From Axiom USB I	Control	1	B0 26 13
00:12:10.453	From Axiom USB I	Control	1	B0 26 19
00:12:10.460	From Axiom USB I	Control	1	B0 26 1F
00:12:10.467	From Axiom USB I	Control	1	B0 26 25
00:12:10.475	From Axiom USB I	Control	1	B0 26 2B
00:12:10.483	From Axiom USB I	Control	1	B0 26 31
00:12:10.492	From Axiom USB I	Cont		7
00:12:10.521	From Axiom USB I	Cont		8
00:12:10.555	From Axiom USB I	Cont		7
00:12:10.612	From Axiom USB I	Cont		6
00:12:10.635	From Axiom USB I	Cont		2
00:12:10.653	From Axiom USB I	Cont		E
00:12:10.669	From Axiom USB I	Cont		A
00:12:10.683	From Axiom USB I	Cont		6
00:12:10.699	From Axiom USB I	Cont		2
00:12:10.717	From Axiom USB I	Control	1	B0 26 1E
00:12:10.739	From Axiom USB I	Control	1	B0 26 1A
00:12:10.766	From Axiom USB I	Control	1	B0 26 19

From previous page..

every control is actually working. You can also discover which buttons are momentary (so send a Note On message when pressed, followed by a Note Off as soon as released) and which are latching (so only send a Note Off when pushed a second time).

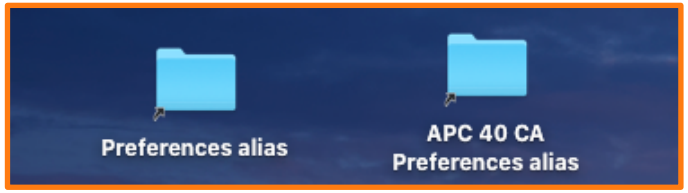
If you haven't already done so, now is a good time to go into *Logic Pro/Preferences/ Advanced Tools* and activate them all. (Not all of them are needed for writing Controller Assignments but if we're all working from the same starting point, there's less chance you'll find yourself saying: "That's odd. It doesn't look like that on mine...")

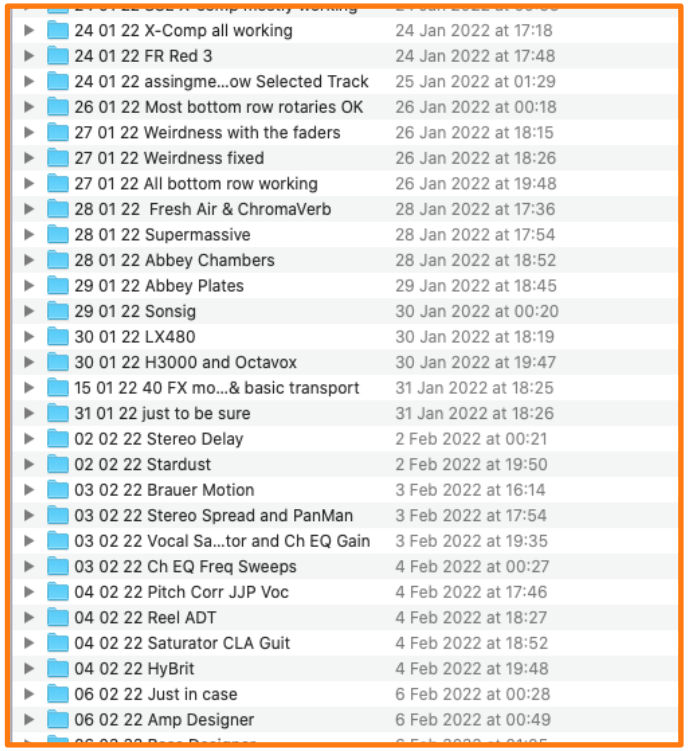
Finding Preferences

LPX stores Controller Assignments in the Preferences folder in the Library. The way to get here is to go *Finder/Go*, then hold down the Alt key. That makes Library appear on the list of options. When you are in the Library and have located the Preferences folder, I strongly recommend you to Right-Click and select Make Alias. Then drag that alias to your Desktop.

The file that stores CA is 'com.apple.logic.pro.cs'. You should delete this file (unless you've already made CAs you want to keep, in which I suggest you store the file somewhere for now). Removing this Preferences file forces LPX to create a new one the next time it is launched.

Later, I also strongly recommend you to make an alias of wherever you are going to





24 01 22 X-Comp all working	24 Jan 2022 at 17:18
24 01 22 FR Red 3	24 Jan 2022 at 17:48
24 01 22 assingme...ow Selected Track	25 Jan 2022 at 01:29
26 01 22 Most bottom row rotaries OK	26 Jan 2022 at 00:18
27 01 22 Weirdness with the faders	26 Jan 2022 at 18:15
27 01 22 Weirdness fixed	26 Jan 2022 at 18:26
27 01 22 All bottom row working	26 Jan 2022 at 19:48
28 01 22 Fresh Air & ChromaVerb	28 Jan 2022 at 17:36
28 01 22 Supermassive	28 Jan 2022 at 17:54
28 01 22 Abbey Chambers	28 Jan 2022 at 18:52
29 01 22 Abbey Plates	29 Jan 2022 at 18:45
29 01 22 Sonsig	30 Jan 2022 at 00:20
30 01 22 LX480	30 Jan 2022 at 18:19
30 01 22 H3000 and Octavox	30 Jan 2022 at 19:47
15 01 22 40 FX mo...& basic transport	31 Jan 2022 at 18:25
31 01 22 just to be sure	31 Jan 2022 at 18:26
02 02 22 Stereo Delay	2 Feb 2022 at 00:21
02 02 22 Stardust	2 Feb 2022 at 19:50
03 02 22 Brauer Motion	3 Feb 2022 at 16:14
03 02 22 Stereo Spread and PanMan	3 Feb 2022 at 17:54
03 02 22 Vocal Sa...tor and Ch EQ Gain	3 Feb 2022 at 19:35
03 02 22 Ch EQ Freq Sweeps	4 Feb 2022 at 00:27
04 02 22 Pitch Corr JJP Voc	4 Feb 2022 at 17:46
04 02 22 Reel ADT	4 Feb 2022 at 18:27
04 02 22 Saturator CLA Guit	4 Feb 2022 at 18:52
04 02 22 HyBrit	4 Feb 2022 at 19:48
06 02 22 Just in case	6 Feb 2022 at 00:28
06 02 22 Amp Designer	6 Feb 2022 at 00:49

A couple of aliases and archiving every time you create new CAs is all it takes to ensure you'll never 'lose the lot'

store copies of the 'com.apple.logic.pro.cs' and put that on the desktop. That way, making regular copies of your progress to date becomes easy.

I suggest you save a copy of the CA's Preferences file every time you make a significant change. My personal scheme is to save each new .cs file in a new folder with the date and a note of what I've changed. So '06 02 22 Amp Designer' indicates that the .cs file was copied on 6 Feb 2022, just after I'd created all the CAs for the Amp Designer plug-in.

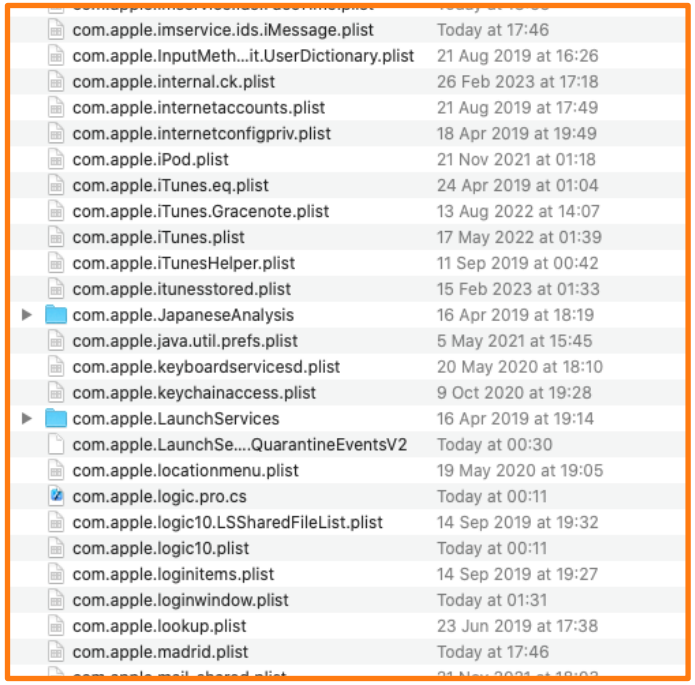
This is your insurance policy against loss of endless hours of your time! If the file becomes corrupt, you can revert to the last good copy, so you haven't lost everything. Also, you'll probably want to use some of the suggestions in this book as test pieces, and may not want to write them to your 'master' .cs file.

Why remove the .CS file?

There are two main reasons why it makes sense to remove the .cs file before starting your CA build project. One is that these files can become corrupt. Sometimes, this is caused when a Mode is deleted. If that happens, it can become impossible to write some CAs – even though they should be perfectly possible – and you may find buttons jumping to entirely different Modes than the ones you assigned.

The second reason to start with a 'clean slate' is a little harder to explain. I wrote this book using the OpenOffice word processor. Every time I write 'control', OpenOffice suggests 'controller'. That's because I've used the word 'controller' so many times, the word processor calculates my intention is 'controller', based on based on prior usage. That's predictive text, and we've all got it on our phones, word processors, and pretty much

Continued over...



com.apple.imservice.ids.iMessage.plist	Today at 17:46
com.apple.InputMeth...it.UserDictionary.plist	21 Aug 2019 at 16:26
com.apple.internal.ck.plist	26 Feb 2023 at 17:18
com.apple.internetaccounts.plist	21 Aug 2019 at 17:49
com.apple.internetconfigpriv.plist	18 Apr 2019 at 19:49
com.apple.iPod.plist	21 Nov 2021 at 01:18
com.apple.iTunes.eq.plist	24 Apr 2019 at 01:04
com.apple.iTunes.Gracenote.plist	13 Aug 2022 at 14:07
com.apple.iTunes.plist	17 May 2022 at 01:39
com.apple.iTunesHelper.plist	11 Sep 2019 at 00:42
com.apple.itunesstored.plist	15 Feb 2023 at 01:33
com.apple.JapaneseAnalysis	16 Apr 2019 at 18:19
com.apple.java.util.prefs.plist	5 May 2021 at 15:45
com.apple.keyboardservicesd.plist	20 May 2020 at 18:10
com.apple.keychainaccess.plist	9 Oct 2020 at 19:28
com.apple.LaunchServices	16 Apr 2019 at 19:14
com.apple.LaunchSe....QuarantineEventsV2	Today at 00:30
com.apple.locationmenu.plist	19 May 2020 at 19:05
com.apple.logic.pro.cs	Today at 00:11
com.apple.logic10.LSSharedFileList.plist	14 Sep 2019 at 19:32
com.apple.logic10.plist	Today at 00:11
com.apple.loginitems.plist	14 Sep 2019 at 19:27
com.apple.loginwindow.plist	Today at 01:31
com.apple.lookup.plist	23 Jun 2019 at 17:38
com.apple.madrid.plist	Today at 17:46

Tip – Open your .cs file once using the Xcode developer application and the icon will change. Easier to find!

From previous page..

everywhere text can be keyed.

LPX does something similar with CAs, second guessing the user based on their prior actions. The downside to this arrangement is that LPX's guesses can be very wide of the mark if your previous actions have been somewhat erratic!

This is why repeated sessions of 'playing about' with CAs can make LPX produce seemingly illogical assignments. If you've had assignments repeatedly end up in No Zone – despite the fact you were in a specific Mode when you wrote them – it's probably because you've written a lot of CAs in No Zone in the past.

Other than starting with a new .cs file, the best way you can correct LPX is to immediately cut-and-paste the CA into its correct position. If you do this repeatedly, you will discover the number of errors goes down. Eventually, you'll find errors are very rare indeed, so making logical assignments is very much a 'virtuous circle'. As the error count goes down, your productivity will increase greatly.

It's all in the planning

Combining the facts that it is possible to corrupt a .cs file through deletions, and LPX will make very few errors if you only make logical assignments, we start to see how important it is to work to a plan.

During the time I've been learning Controller Assignments (and, believe me, I'm still learning) I've gone through three stages with regard to planning:

1. Making it up as I go along, not writing it down anywhere then being very vague about how my CAs worked a few months later. Not only does this have all the downsides outlined above, the worst thing is that you stop using the assignments you've created as soon as you forget what they do! So, you put all that work in, but you're back to working with the QWERTY keyboard and mouse.

2. Documenting all my assignments. This was a vast improvement, because logging what I'd done mean I actually used the assignments I'd created, and started to gain hardware control over the LPX environment. But, if you think about it, keeping a note of what you've done after the event is still making it up as you go along! There's no actually planning ahead of time.

Note – You can only correct errors if you detect them, which is one of the many reasons to write CAs with the Controller Assignments window open. If you're not sure where in the hierarchy the CA you wrote has gone, touch the physic control you just assigned and the CA window will jump to it. You can then cut-and-paste it to the right place.

3. Creating a plan, then writing the assignments to the plan. This was a massive breakthrough for me. On one hand, it forced me to think about the complete structure of Zones, Modes, Controls and Parameters, along with how it could best be applied to the hardware I had. So I was making better decisions from the off. On the other hand, working to a plan meant the speed with which I was writing CAs went up and up. I've now got to the point where I can normally write all the CAs for a plug-in in just a few minutes.

The difficulty with going from Stage 2 to Stage 3 is you need to know what building blocks you can use to create your scheme. The 'plan' isn't meant to show all the assignments; it's just there to remind you what the knobs and buttons on the MIDI controller are going to do.

One of the reasons this chapter is probably the most important in the book is we're going

Continued over...

From previous page..

to look at the building blocks we'll be using in the rest of the book. Once you know what's actually possible, you'll be able to create your own plans. More importantly, you'll then be able to carry them out.

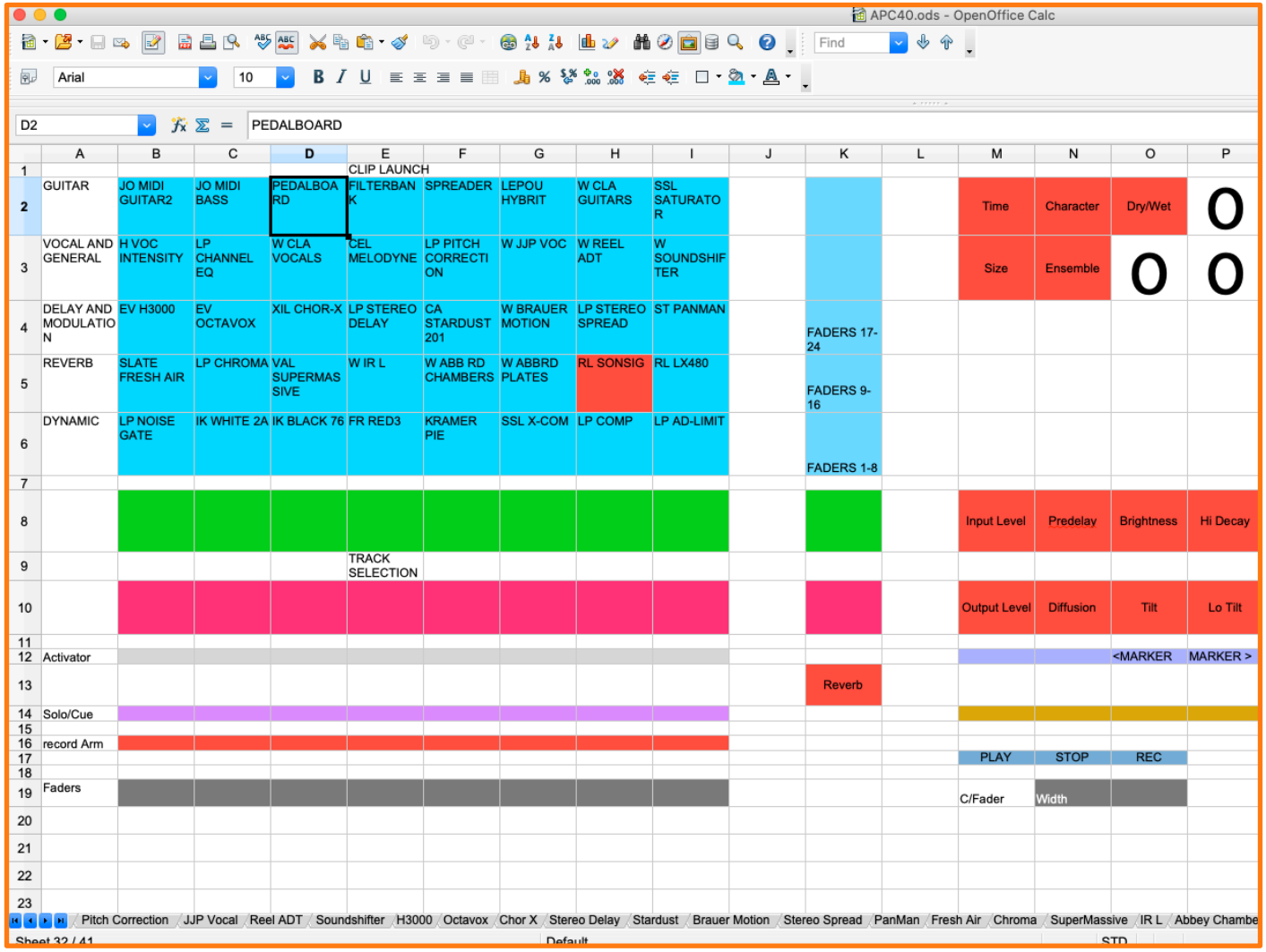
The planning tools

Of course, the most important planning tool is your mind. Only you know what you need as a musician, and it's your skill and imagination that will make it happen. But even the world's greatest monuments need tools and materials to get built.

I decided the best starting point was a spreadsheet, laid out as a visual representation of my main MIDI controller. As you can see, my spreadsheet looks like the physical layout of the Akai APC40. (Well, maybe. If you squint a bit and you're feeling generous.)

As a starting point, I decided that the central bank of 40 buttons would each route to a different Mode and that Mode would contain all the CAs needed for a particular plug-in. That meant I needed to copy the first page of my spreadsheet another 40 times, to show what all the other controls were going to do when I pushed one of the 40 buttons. If you take a look at a few examples, you'll soon see what I mean.

Using a graphical representation turned out to be a very good idea, because it made it really easy to plan what I was going to do, and how it would function in use. Doing it using a



Planning using a spreadsheet worked, but it was really hard to update. Unless you know how to build relational databases, using more than one table/spreadsheet, this isn't an easy road to go down

spreadsheet turned out to be a very bad idea. Here's why.

A number of times during the planning stage, I came up with a better idea, aka 'changed my mind'. If the change was at the top level – ie what those 40 buttons did – I had to edit 41

Continued over...

From previous page..

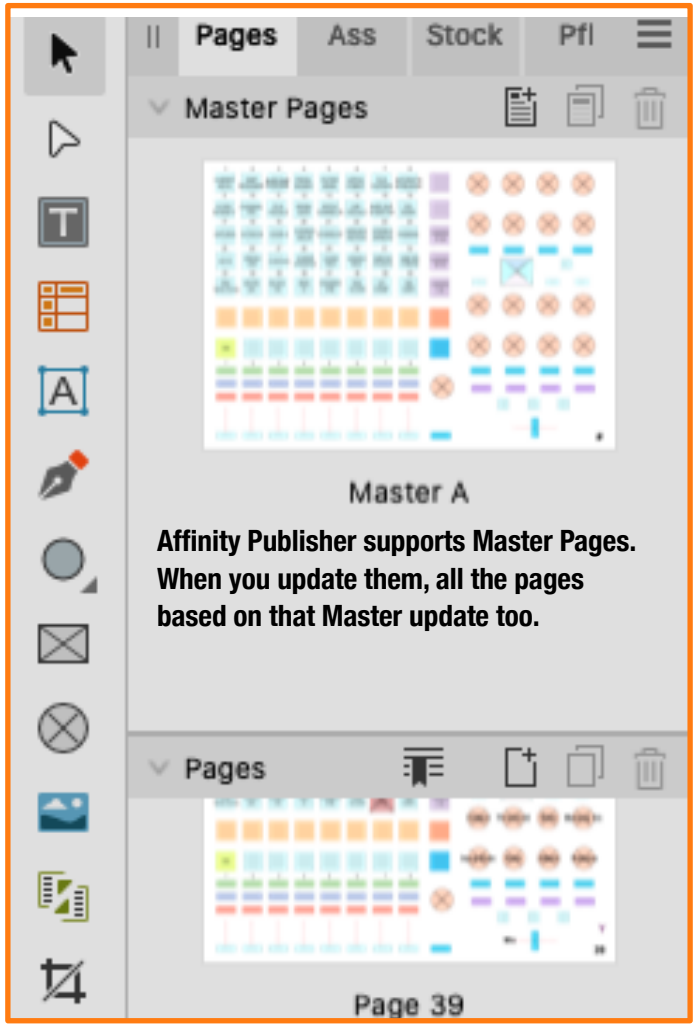
separate pages, which was a massive pain. Don't ask me where. You already know.

If you know your way round spreadsheets, you probably know that there's a way to link cells between spreadsheets. Had I done that from the first page to all the following ones, it would have saved me from later making edits to all pages manually, but even setting up the references between the pages seemed like a lot of work.

Because I have a background in magazine and newspaper production, I decided it would be better to use a page layout program, like Adobe InDesign, or Quark Xpress. Sadly, I've been out of the game for a while, so I didn't have either of those tools any more... and they are expensive options for anyone who isn't earning a living from them.

I ended up opting for Affinity Publisher (https://affinity.serif.com/en-gb/publisher/), which is £47.99 at the time of writing. It's actually remarkably good. Getting to grips with all the ins-and-oust of the program would be a steep learning curve if you haven't worked in the magazine business, but all we need to do for this exercise is learn how to create coloured circles, rectangles and to write in text boxes. On the other hand, if you're familiar with InDesign, Affinity Publisher won't be much of a mystery to you.

How's that better than using a spreadsheet? Well, the main benefit is that page design programs are based on Master pages, on which all other pages are based. So, if you



Affinity Publisher supports Master Pages. When you update them, all the pages based on that Master update too.

change the master, all of the other pages update (see below).

A secondary benefit is we can use these smart-looking layouts as the basis for a display system, to show what the controls do, as I explain in *Practical Projects Volume 2*. Yep, I'm talking about a display that changes when you push the buttons on your MIDI controller...

Design options

Although you will hopefully get good ideas from this book, and learn how to use them by copying, ultimately you'll want to design a scheme that gets what you want from the hardware you have. For those of us who don't have vast amounts of money to throw at this, that means making some smart design decisions.

For starters, you'll probably never have as many physical controls as the plug-ins etc you want to address. Then again, if you did have a selector button for every plug-in, you'd have a very large control surface, like a full-scale studio mixing console, so the design compromise isn't just about money.

One of the things that attracted me to the Akai APC 40 was it left me relatively rich in buttons and rotaries. But I soon came to see that my original scheme of one plug-in for each of the 40 buttons in the 'Clip Launch' section was actually very wasteful. For instance, some of the plug-ins that emulate classic compressors have only maybe four rotary controls to assign to.

When you've got 17 rotaries on the control surface, it's silly to leave 13 of them doing nothing some of the time. Once I realised that, I started doubling up (or more) the number of plug-ins each of the 40 buttons could control. Well, that's a bit of a sloppy way of saying it. What I actually did was start putting the CAs for more than one plug-in within some of the Modes.

What that means in practice is that a single

Continued over...

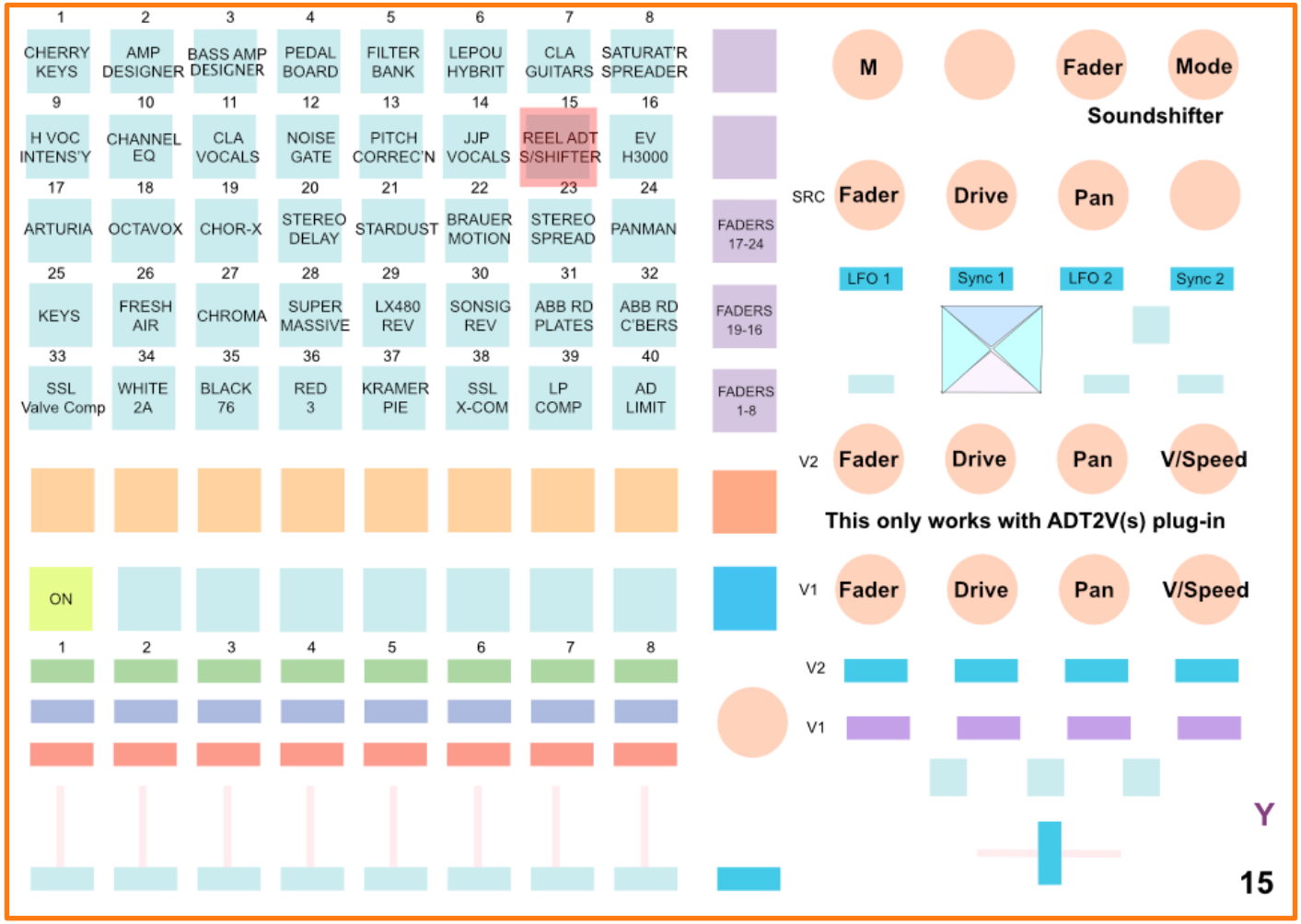
From previous page..

mode could contain CAs for four completely different plug-ins, each with four rotary controls on a different row. Even if you had all four plug-ins inserted in a single channel strip, there would be no interaction between the CAs, because the assignments for each plug-in are mutually exclusive. The screen-shot makes this easier to understand.

But what about plug-ins that will never be in the same channel strip? I'm thinking here of synth and other keyboard-type plug-ins. Providing your assignments are based on Class: Channel Strip/Channel Strip: Selected track, you could put all your synth, piano and organ plug-in assignments into a single Mode, addressed by a CA from a single button.

If the paragraph above has made your head spin, I suggest you read it again, then look at the screen-grabs, so you can see what this actually looks like in LPX. Once you've convinced yourself that you really can have multiple synths in one Mode/button assignment, take a few deep breaths and we'll have a recap on how our design options are expanding so far.

We started with the idea that each button on the controller could be assigned to a different Mode, and that Mode would contain all the CAs for a single plug-in. Then we looked at the idea of using any unused rotaries and switches within a mode for different plug-ins. So when we selected, say, button Five, the lower section of rotaries and switches might be assigned to one plug-in, while the upper section was a completely different plug-in.



In the example above, I've assigned some of the controls to the Soundshifter plug-in and others to the Reel ADT plug-in. I could further assign the same controls to completrely different plug-ins, providing they were never used in the same channel.

Then we introduced a new idea: we could use the same rotaries and switches for multiple plug-ins in a single Mode providing: a) none of the plug-ins would ever be used in the same channel as any of the others in the Mode, and b) the assignment was to 'Selected Track', meaning no other Track/channel strip would be affected. In other words, we were only working with the highlighted channel, and the setting on any other channel would stay unchanged.

Continued over...

From previous page..

Note – In real-life, you might not want to put all your synth, organ and piano assignments in a single Mode. The reason is it get's really confusing writing all your CAs into a single Mode! However, it is very efficient in use, as I'll explain later in this book (Chapter 11 – *CAs for classic Cherry Audio keys*). The screen grabs on the right show just some of the assignments used for the Cherry Audio Eight Voice and Memory Mode. OK, they're classic synths with a ton of on-screen controls, but imagine what that Mode is going to look like if we thow in every keyboard plug-in we've got.

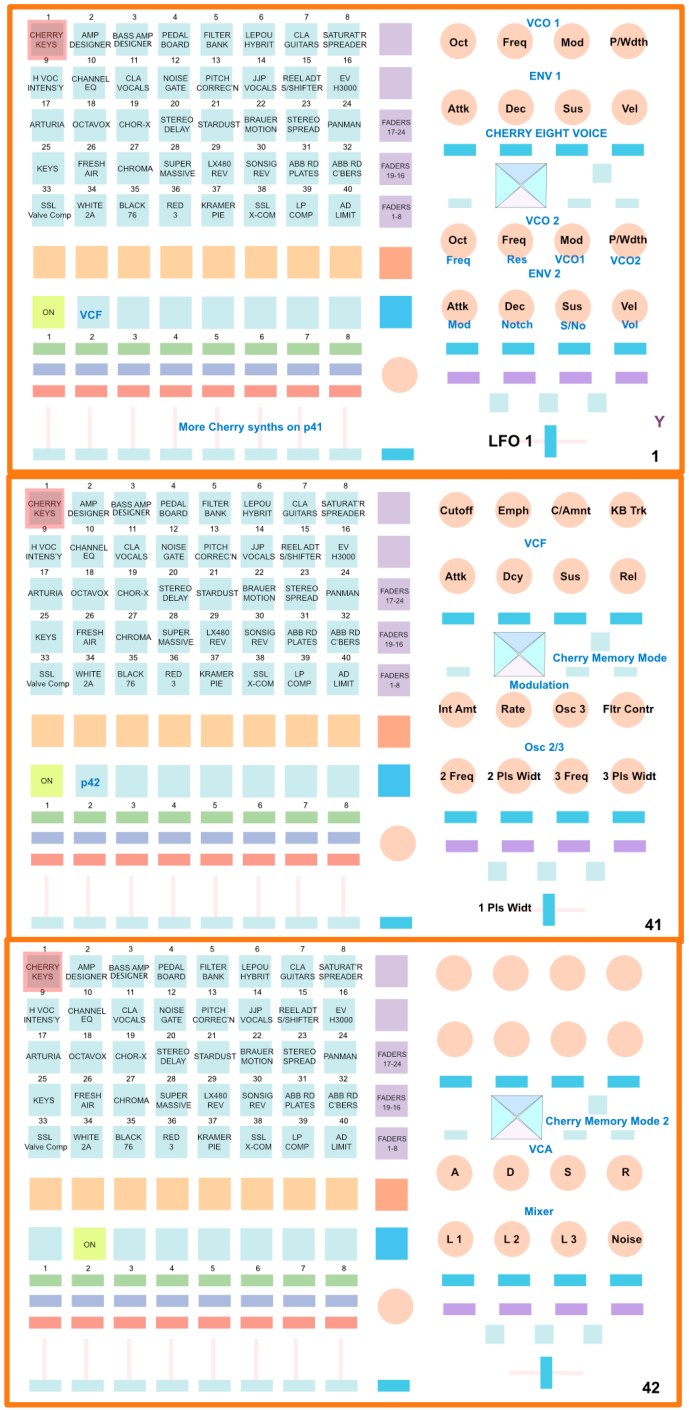
How each individual manages this is up to them, but I've got one Mode called Cherry Keys, another called Arturia Keys, and so on. Otherwise we'd be back to the chaos caused if we cram every letter a company receives or sends into one folder in the filing cabinet marked 'Letters'.

That dynamic display system is starting to sound more-and-more essential, isn't it?

Some MIDI controller-specific options

Some MIDI controllers have a feature that gives you even more design options. They perform a Scene Change, where some – sometimes nearly all – the controls send out a completely different set of MIDI messages.

You'll have already come across this idea if you have a musical keyboard with less that 88 keys, because it will be fitted with two buttons



that change the octave you're playing in. LPX doesn't 'know' you have only 49 keys; it simply responds to the incoming MIDI messages.

In Chapter 8 – *Live performance system with backing tracks and mixer*, we'll see how we can harness this Scene Change feature on a Korg nanoKontrol to access more and more songs, without using up more physical buttons.

In Chapter 12 – *A selection of Controller Assignments for analogue-style synths*, we'll see how a similar feature on the Akai APC40 means we can assign physical controls to almost every knob on even a classic 80s-style analogue synth.

How we document this – so we know what the heck's going on with our assignments – is really important. If we don't know what our CAs, we probably won't ever use them!

The building blocks

In the scheme I use in Chapter 9 – *Assigning to multiple plug-ins*, most of the assignments are in a Zone called APC40 Plug-ins. This contains 40 Modes, each one selected in turn by 40 physical buttons. The CAs for these are found in APC40 Plug-ins/(No Mode). The Class of these assignments is Mode Change.

When you use the Class Mode Change, you'll be presented with a pull-down menu of all the available Modes. This raises a key point that unless you create the Modes first, there won't be any Modes to choose from!

Generally, speaking, you'll create the Zone first, then populate it with the Modes, and

Continued over...

From previous page..

finally write the assignments for each Mode. Because nearly all the CAs I've used are of the type *Class: Channel Strip/Channel Strip: Selected Track,* they are only active on the highlighted channel strip.

(In LPX, the term 'track' is often used interchangeably with 'channel strip'. As I grew up in a world where the mixing console was one piece of hardware and the multitrack tape recorder another unit entirely, I tend to think of the audio regions in the Arrange window as being on horizontal 'tracks' and the vertical sections containing the processing in the Mixer window as being 'channel strips'.)

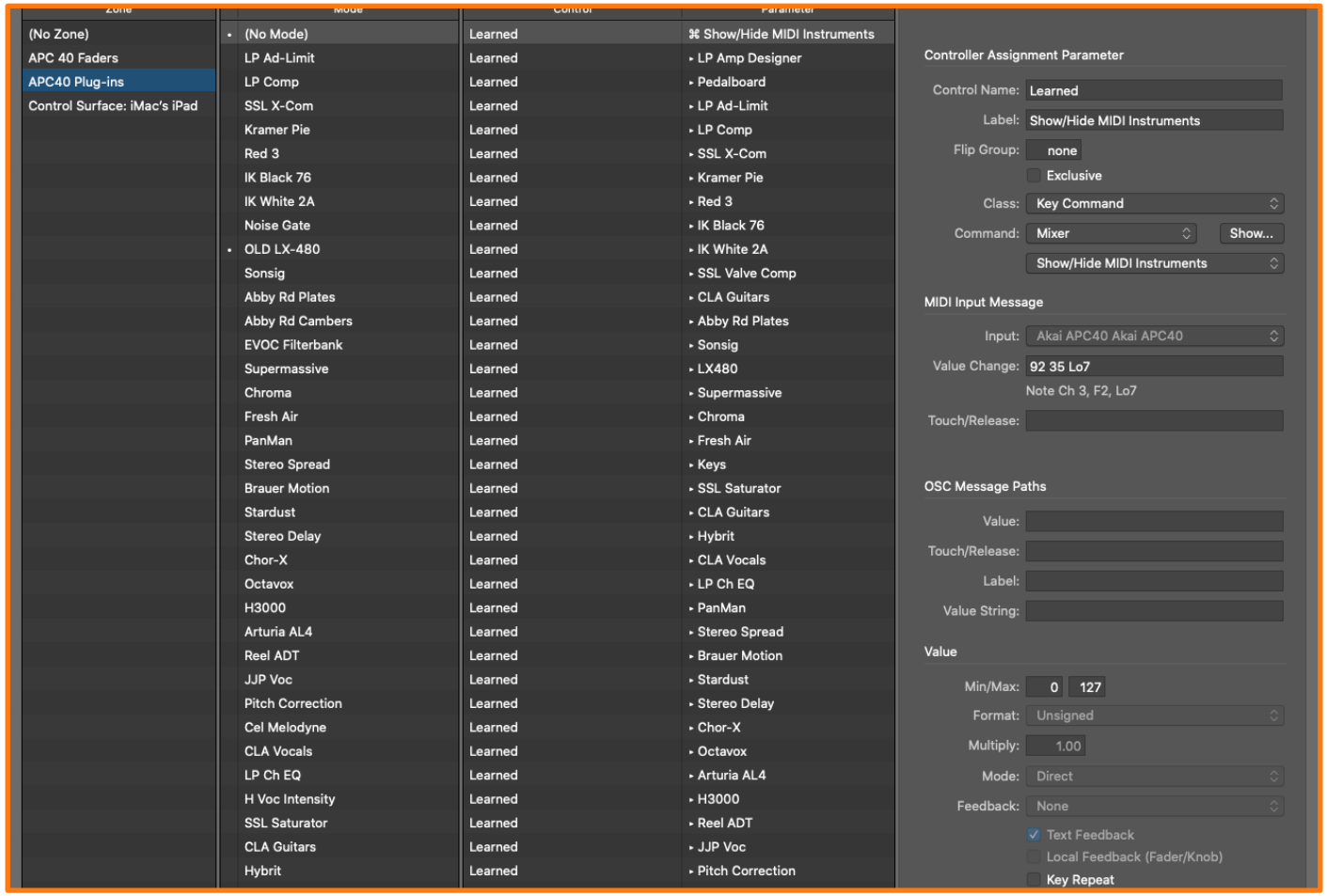
Doubling up the controller

At this point, you may be thinking: "Great, but my MIDI controller hasn't got 40 buttons, so I won't be able to control anything like that number of plug-ins." Not so. Here's the good news. There are at least two ways you can control more plug-ins than the number of buttons on your controller.

We've already discussed one approach, which is to simply write for CAs for more than one plug-in into a single Mode. This works fine, providing you can answer yes to *one* of these two questions. 1) Is each CA in the Mode using a different physical control on the MIDI unit? *Or*, 2) Are you absolutely sure you'll never want to use the more than one of the plug-ins in the Mode in the same channel strip?

As we discussed earlier, the second question is easy to answer when it comes to software instruments, because it's actually impossible to use more than one in the same track/channel strip. When it comes to audio processors, there are definitely times when you might want to use two compressors in the same channel strip, for example.

Here's another approach, which I used when writing plug-in CAs for my little Korg unit. You can write a Mode Change class of CA inside



Zone	Mode	Control	Parameter
(No Zone)	(No Mode)	Learned	⌘ Show/Hide MIDI Instruments
APC 40 Faders	LP Ad-Limit	Learned	LP Amp Designer
APC40 Plug-ins	LP Comp	Learned	Pedalboard
Control Surface: iMac's iPad	SSL X-Com	Learned	LP Ad-Limit
	Kramer Pie	Learned	LP Comp
	Red 3	Learned	SSL X-Com
	IK Black 76	Learned	Kramer Pie
	IK White 2A	Learned	Red 3
	Noise Gate	Learned	IK Black 76
	OLD LX-480	Learned	IK White 2A
	Sonsig	Learned	SSL Valve Comp
	Abby Rd Plates	Learned	CLA Guitars
	Abby Rd Cambers	Learned	Abby Rd Plates
	EVOC Filterbank	Learned	Sonsig
	Supermassive	Learned	LX480
	Chroma	Learned	Supermassive
	Fresh Air	Learned	Chroma
	PanMan	Learned	Fresh Air
	Stereo Spread	Learned	Keys
	Brauer Motion	Learned	SSL Saturator
	Stardust	Learned	CLA Guitars
	Stereo Delay	Learned	Hybrit
	Chor-X	Learned	CLA Vocals
	Octavox	Learned	LP Ch EQ
	H3000	Learned	PanMan
	Arturia AL4	Learned	Stereo Spread
	Reel ADT	Learned	Brauer Motion
	JJP Voc	Learned	Stardust
	Pitch Correction	Learned	Stereo Delay
	Cel Melodyne	Learned	Chor-X
	CLA Vocals	Learned	Octavox
	LP Ch EQ	Learned	Arturia AL4
	H Voc Intensity	Learned	H3000
	SSL Saturator	Learned	Reel ADT
	CLA Guitars	Learned	JJP Voc
	Hybrit	Learned	Pitch Correction

Controller Assignment Parameter
Control Name: Learned
Label: Show/Hide MIDI Instruments
Flip Group: none
Exclusive
Class: Key Command
Command: Mixer Show...
Show/Hide MIDI Instruments
MIDI Input Message
Input: Akai APC40 Akai APC40
Value Change: 92 35 Lo7
Note Ch 3, F2, Lo7
Touch/Release:
OSC Message Paths
Value:
Touch/Release:
Label:
Value String:
Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Direct
Feedback: None
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat

The Zone called APC40 Plug-ins contains 40 Modes, each one selected in turn by 40 physical buttons. The CAs for these are found in APC40 Plug-ins/(No Mode). The Class of these assignments is Mode Change

Continued over...

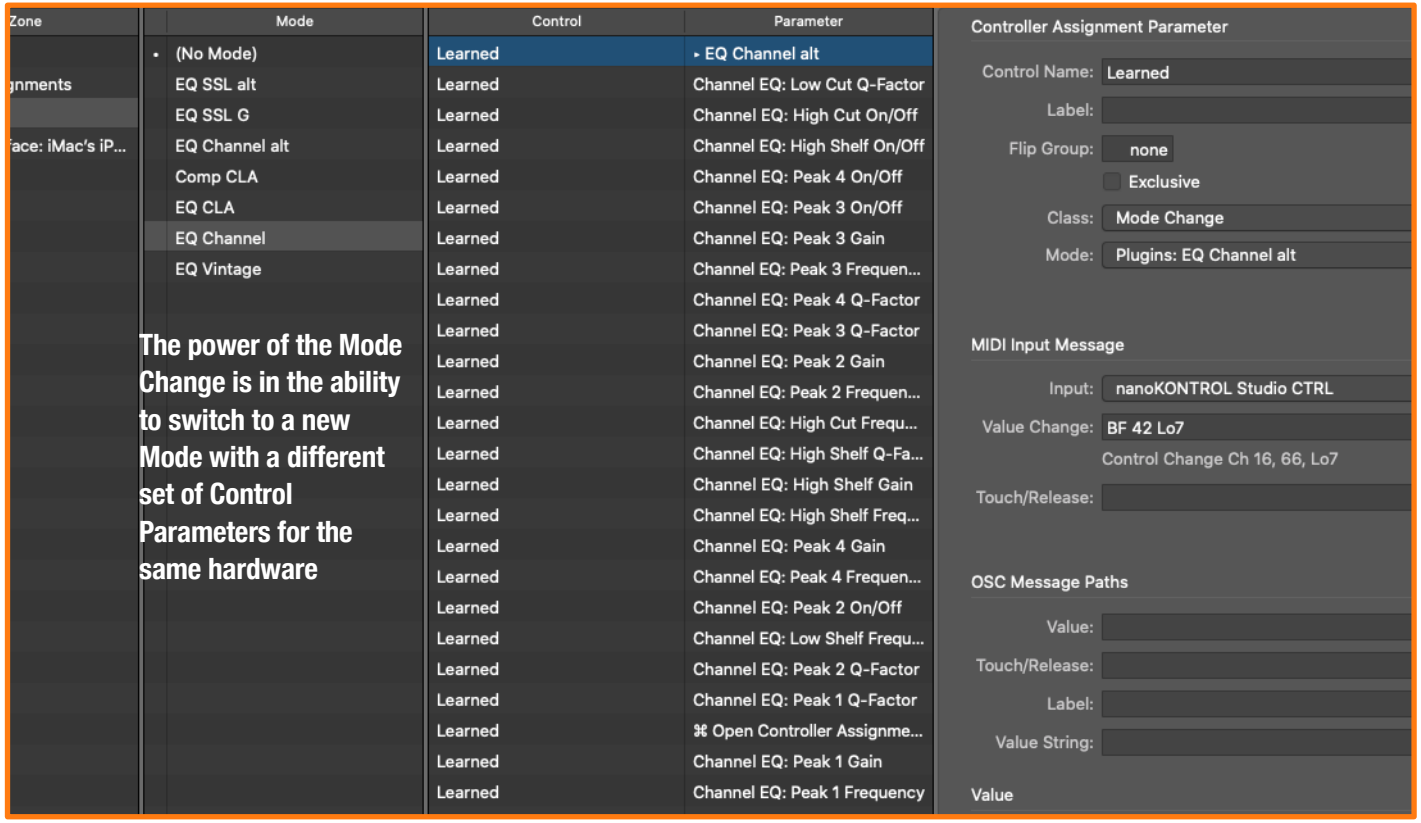
From previous page..

one mode, so that it switches to another. I first used this to get round the fact the Korg nanoKontrol had eight rotaries, plus eight faders, but you need 24 controls in total if you want to address all the parameters in the Linear EQ. (8 frequency, 8 cut/boost, 8 Q, or contour, equals 24.)

By introducing a Mode Change CA on one of the buttons, we can reuse either the faders or the rotaries to address the third set of parameters, with those CAs in a separate Mode. When I do this, I usually name the new Mode after the first one but add 'alt' at the end, to make it clear this is the alternative assignments for some of the controls.

Let's say my Mode 'Channel EQ' contains a Mode Change to 'Channel EQ alt', can I write a another Mode Change inside 'Channel EQ alt'? Absolutely, yes! In fact, you may choose to do exactly that was a way of going from 'Channel EQ alt' back to 'Channel EQ'. (Another way of doing it is to use a momentary switch for the Mode Change to 'Channel EQ alt', so it switches back to 'Channel EQ' when you take your finger off the button.)

Continued over...



The power of the Mode Change is in the ability to switch to a new Mode with a different set of Control Parameters for the same hardware

CAUTION – While deleting Control Parameters that haven't worked out is something you have to do as you go along, I would caution against deleting Modes, because it can corrupt the 'com.apple.logic.pro.cs'. This caution will appear multiple times, because corrupting the Preferences file can spoil your day, doubly so if you don't notice until a few days later, so have to spend time working out which version of the 'com.apple.logic.pro.cs' that you have saved is the 'clean' one.

The reason it can be hard to spot the file has corrupted is that it often affects a specific CA. For instance, I'd written 39 CAs for buttons selecting a Mode. It was only when I came to write Mode Change number 40 that I discovered it was impossible to make to go the the Mode I'd selected! While it would show the correct Mode in the pull-down menu, when you pushed the button, it went to a different mode entirely.

In the end, I had to simply accept the CA as it was, and change one of the other CAs, so it went to the 40th Mode.

If you end up with a Mode you don't need in your scheme – perhaps because your set-up has changed, or you've found a plug-in you like better – I'd suggest you simply leave the redundant Mode where it is and create a new one, instead of deleting the one you don't want.

From previous page..

How you use the power of the Mode Change is entirely up to you. After all, the purpose of this chapter is to help you to plan your own CA scheme. Copying some of my examples is a great way to learn, and to get up to speed with writing CAs, but there's every chance you've got different MIDI devices to me, different plug-ins and want to enhance your own ways of working.

I'll finish of this section with one more example of how you could use Mode Changes to go through a whole succession of Modes. Let's say you decided that one button on your controller will always be for LPX's EQs. You could write a (No Mode) CA, with a Mode Change to Channel EQ.

Then you could have another button with a CA inside the Channel EQ Mode, that switched it to Linear Phase EQ.

That same button could have a CA inside the Linear Phase EQ that had a Mode Change to a Match EQ Mode... and so on, until you had worked your way through the entire dropdown menu, including the three units in the Vintage EQ Collection.

Practice makes perfect

Just like playing a musical instrument, writing CAs takes a bit of practice before they go right nearly all of the time. One thing that really helps is setting both the physical controls and the on-screen controls to their lowest position (ie faders down, rotaries turned all the way anti-clockwise.)

That means, when you go into Learn mode and move a control, the pickup should be instant.

It's also best to move the control until the one on screen is at the end of its travel, then immediately take LPX out of learn. This minimises that chance of you writing a further CA you didn't want.

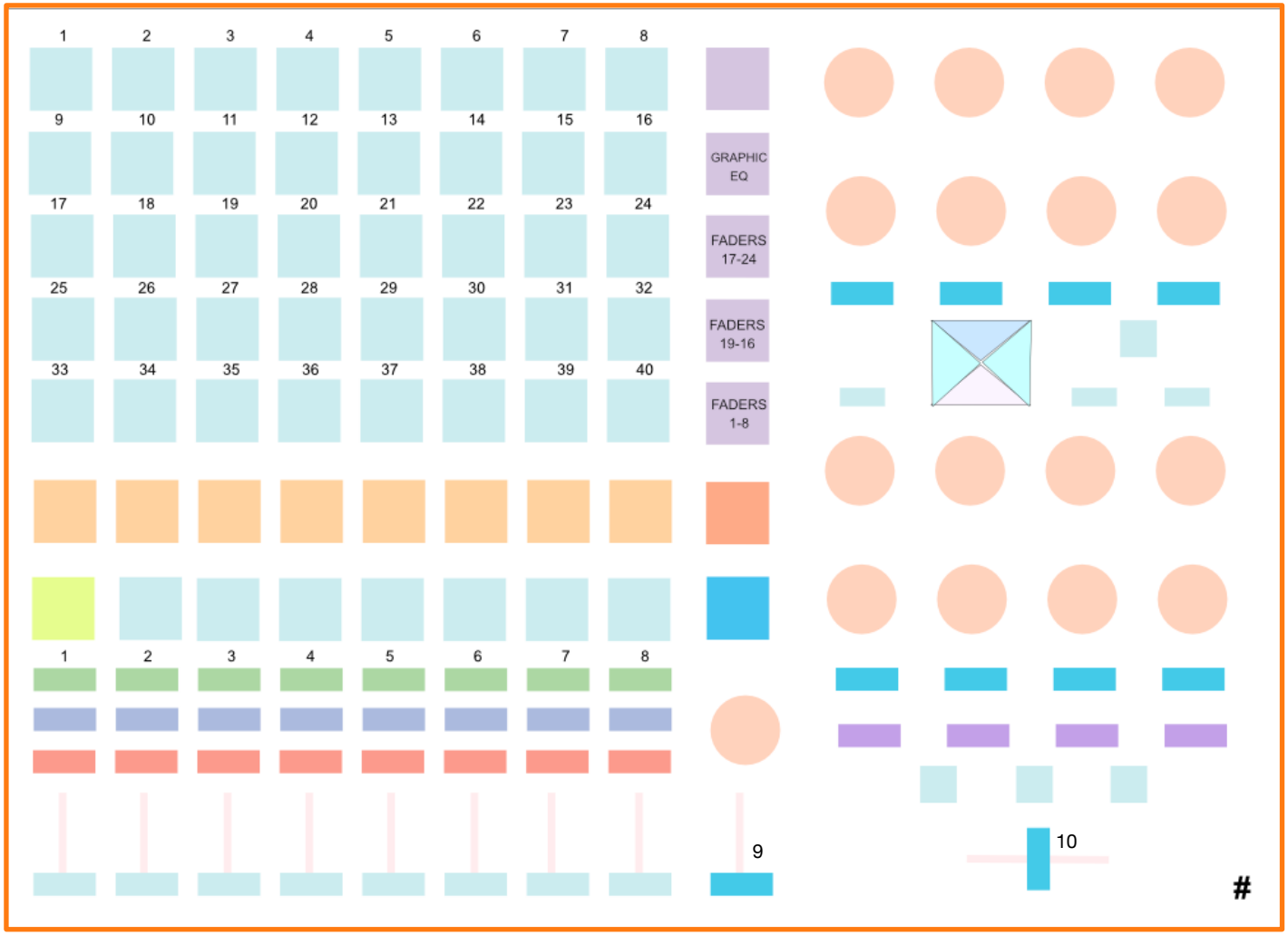
I always check in the CA window that my assignments are being written to the correct mode, and correct them using 'cut & paste' if not. Remember, the more you correct any mistakes, the less mistakes you will get in the future, so the whole process becomes faster and easier.

It's also worth checking that the Min/Max MIDI values range from 0-127 – assuming that was your intention – as LPX doesn't always write them correctly. If you have a very worn MIDI controller that is sending out somewhat random values, or you move the controls in an indistinct manner, things are more likely to go wrong.

Remember, it's not a race. Clear, smooth movements – not speed – help to make for accurately written QAs.

Have I covered all the 'building blocks' that can help you make useful CA set-ups. Er... no. That would make for a very long book. Besides, how am I supposed to get rich and famous if I don't write a Practical Projects Volume 2? □

Build a 24 channel mixer with graphic EQ



What you'll learn in this chapter:

This is a practical demonstration of how to get 10 faders and four buttons to address 24 input channels, Stereo Out, plus a 10-band graphic EQ.

The main purpose of this exercise is to familiarise you with using Mode Changes to use the same physical controls to assign to multiple functions, and to help you understand why this mixer is structured the way it is. That way, you can create your own assignments, by harnessing the same concepts to build schemes that build your personal needs.

The other purpose of this chapter is mainly to give you some hand-on experience, to show you how you can achieve a lot relatively easily, and to give you a better feel for what's possible. If you don't have as many faders as I've used, you can still adapt the basic methods to make a mixer that works with your MIDI hardware.

If you're currently using a Preferences file that has been only been used to 'mess about with' Controller Assignments in the past – or you'd refer to treat this chapter as a stand-alone exercise that won't alter your existing CAs – I suggest you remove it, so LPX creates a new one when you launch it. How to best manage that is covered on p26.

Before we get hands-on, I'd like to walk you through what the end result is supposed to do from a user perspective, and also why I've structured the CAs the way they are for this exercise.

Continued over...

From previous page..

Overview

The diagram on the previous page is a simplified version of the Akai APC40 control surface. We're only going to be using the 10 faders at the bottom (which includes the DJ-style crossfader), plus the four mauve buttons that have their intended function written on them. If your controller doesn't have as many faders, you can still make a very usable scheme, as I'll explain later.

Turning our attentions to the CA window, you'll see that there's a Zone called Mixer, which was created using the '+' button at the bottom of the Zone section. If you're using more than one MIDI controller, you might want to choose a Zone name that makes it clear which controller it uses.

In the Mode column, you'll see that there are three Modes relating to the faders, plus (No Mode). Why do we put Control Parameters into (No Mode)? Because we want these functions to be available all the time. For instance, the Volume parameter in (No Mode) relates to fader 9, which is marked Master on the controller itself.

We want this fader to be assigned to the stereo output fader in Logic, and we want this to be available, regardless of which bank of faders on screen physical faders 1-8 are assigned to. (Later on, we'll discover that this arrangement has its limitations, and we'll alter the scheme, so that the Master fader controls a frequency slider when the Graphic EQ is in use.)

Also in (No Mode) are four Mode Change parameters. These select: Faders 1-8; 9-16 and 17-24, plus the Graphic EQ. Again, these are functions we want available all the time, so that we can switch fader banks at will.

Before we can create our Mode Changes, we'll need to create some Modes for them to change to. You can see them in the Mode column and if we select one, we'll see it contains all the individual Volume parameters.

Step-by-step: fader volumes

If you were starting this build completely from scratch, step one would be to draw what the CAs are intended to do. Typically, this would involve creating a map like the one I made, showing the buttons do, the function of any rotaries etc. Unless you happen to be using an APC40 as your controller, I suggest you create a similar map for *your* MIDI controller.

When choosing which buttons on your controller to use for switching the fader banks, it's worth knowing that the ones I used are momentary (Note On when pushed down/Note Off when released). Before we start writing any

Continued over...

Note – As a working method, I'd suggest you generally create your individual fader Modes and populate at least some of them with your intended Parameters before you get to Mode Changes. During this stage of development, you can switch between Modes by simply selecting them in the CA window with your mouse/trackpad. There are a number of advantages to scheduling your build this way. One is that, regardless of the functions they address, Modes often use the same type of Parameter over-and over. It's much the same story with Mode Changes. Because of the element of repetition, you'll probably be amazed at how fast you can write parameters once you get into the swing of it. A second benefit is that, by writing your Modes before you worry about Mode Changes, you can focus on a reduced set of factors, meaning you're not trying to hold the whole scheme in your head at one time. That's one of the reasons I've learned to document what I'm going to do, then follow the plan as if I'm a builder, not the architect any more.

From previous page.

CAs, it helps greatly to have all your faders at their lowest position, so that Logic will Learn the full throw of each fader. (This also applies to the faders within the Mixer Window on-screen.)

Before you can start this exercise, you will need to create a Logic project file containing 24 consecutive audio tracks. These should be numbered 1-24 sequentially, to correspond with the CAs we're about to create.

Note – *If you later add, say a software instrument track between audio tracks 2 & 3, the assignments will still work, but operation will be confusing. This is because LPX will regard the new track as 3, meaning your 4th physical fader will now control the track on screen marked Audio 3, and so-on down the line. So, if you want to use the mixer we're creating for real, better to put all your instrument tracks etc after track 24, so the physical faders continue to relate to the right ones on screen.*

With the CA window open (Alt-Shift-k), click on the button at the bottom of the Zone column to create a new Zone. Name this 'Mixer', or whatever name will help you to identify its purpose in future.

Now, click on the button at the bottom of the Mode column, then name the new Mode Faders 1-8. This should also create a (No Mode) above it and it's here we will write our first assignment. With (No Mode) selected, enter the Learn state (Cmd-L) and move the Stereo Out fader on screen slightly. Now move the fader on your controller that you want to be the Stereo Out to the full extent of its travel. Press Cmd-L again to switch Learn off.

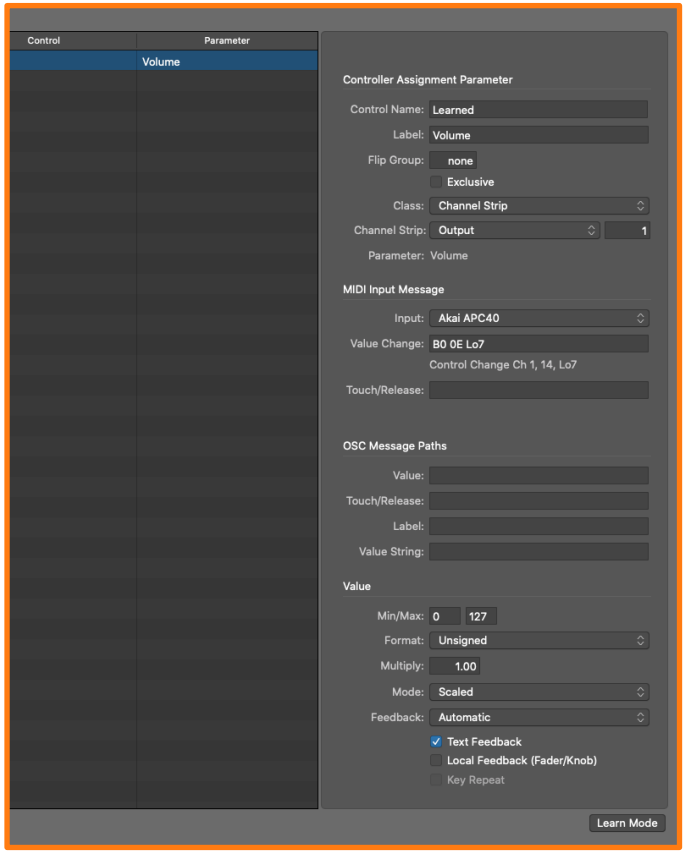
What you've created so far should look like the above. If it doesn't, you can write the correct values in manually (*without* Learn on). For instance, I made a hash of the fader movement, and had to write in the MIDI values of 0-127 before the fader worked correctly.

Next, we're going to select Mode 'Faders 1-8' and write in the CAs for these faders. It's tempting to think at this point that you've already grasped the Big Plan and can finish the whole thing by yourself. If you stick with me, I can save you a shed load of time and frustration, for reasons I'll demonstrate.

Go into Learn mode. Move the on-screen fader to Audio 1 slightly, then ease up the first fader on your controller. Switch Learn off. At this point you'll probably give your first fader on the controller a shove, seen Audio 1 on screen move with it and think: "I totally nailed this!"

Oh yeah? Let me show you something. In the Mixer window, click on the channel marked Audio 2. Now move the fader on the controller again. Oops! It's not assigned to Audio 1, it's assigned to Selected Track. So, no matter which track we click on, the fader will follow it. There's no point in having eight faders do that, because they're all doing the same thing. OK, here's the fix…

In the CA window, change *Class: Channel Strip* to Audio and make sure the number is 1. Now see that the first fader on your controller



does. See? No matter what track you highlight on screen, that fader only moves Audio 1. That means that when we've got all eight faders assigned we can use any of them at will.

At this point, you need to repeat the instructions for Audio 1 but apply them to the relevant faders on your controller and Audio 2-8 in Logic. Once you've completed all eight CAs, you'll no doubt want to have a play with you new toy.

If you do that with the CA window open,

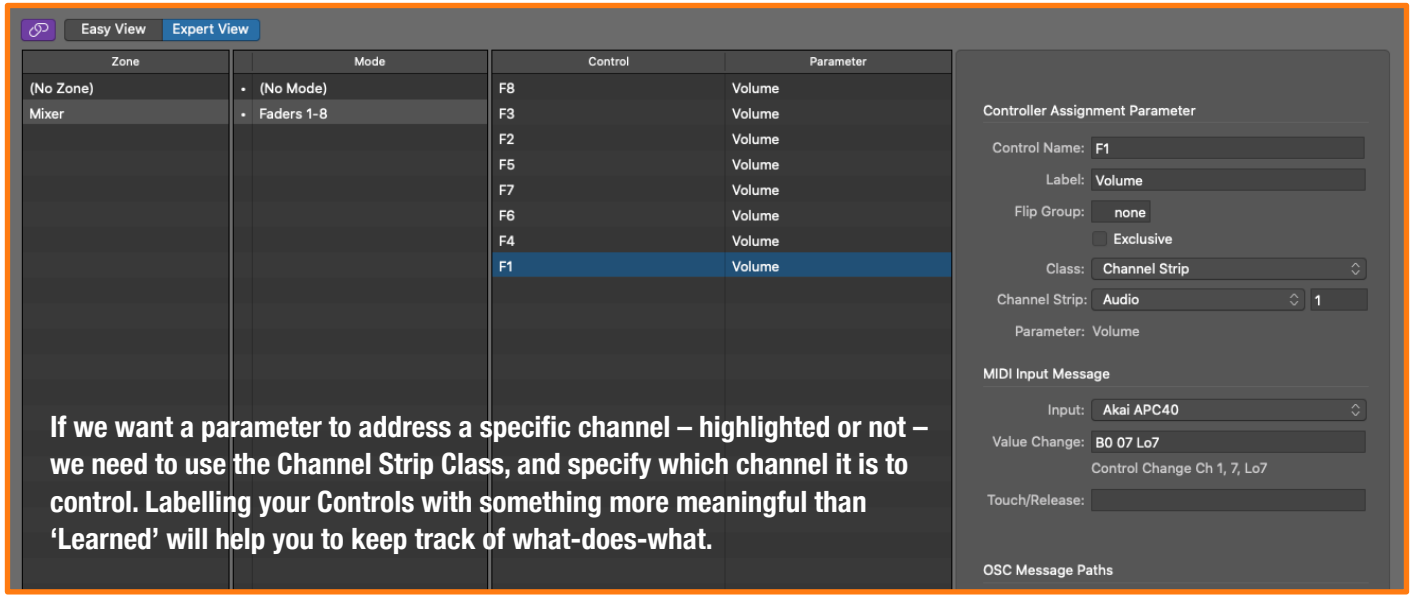
Continued over...

From previous page.

you'll notice the individual Assignments aren't necessarily in numerical sequence. If you want to make it easier you can change the Control from Learned to something a little more informative. In my case, I moved fader 1, clicked on the highlighted Control and renamed it F1, them moved to fader 2… until I got a scheme that looked like the one on the right. Should I want to change the function of any of these CAs in the future, it'll be a lot easier to see 'what's what'.

Now let's save ourselves some time. Instead of creating a new Mode for faders 9-16, and populating it with new CAs, we can simply highlight the Fader 1-8 Mode, go Cmd-C, Cmd-V to cut and paste, rename the duplicate Mode 'Faders 9-16', then repeat for 'Faders 17-24'.

Of course we have to put in a little bit of work after that. We need to go into 'Faders 9-16', and change the Channel Strip numbers to reflect the new track numbers they address. Fortunately, this is fairly straightforward if you named the Controls to reflect the physical faders the work with. In other words, you go into the Mode 'Faders 9-16', click on Control F1, then change the Channel Strip number from 1 to 9. Then F2 becomes 10 and so on.



If we want a parameter to address a specific channel – highlighted or not – we need to use the Channel Strip Class, and specify which channel it is to control. Labelling your Controls with something more meaningful than 'Learned' will help you to keep track of what-does-what.

Step-by-step: switching fader banks

We could go on and create the Volume assignments for the Graphic EQ, but let's create the Mode Changes that select each bank of eight fader to create our 24 channel audio mixer first. (Incidentally, the word 'bank' isn't really LPX CA terminology, but I've used it as a handy way to describe how our controller will be used from an audio mix perspective.)

For the rest of this section, we will be working exclusively in (No Mode) because we need these 'bank change' assignments to be available to us at all times. The first thing I suggest you do is change Control name of the one Parameter already in (No Mode) to 'Master'. Otherwise, you are likely to get very confused and make annoying mistakes.

Once you've done that, put CA into Learn and push the button that will assign to Faders 1-8. Turn Learn off. Congratulations! You've created a CA that does absolutely nothing useful. Let's fix that. First let's change the Control name from 'Learned' to 'Switch Faders 1-8' so we clearly understand what it's for.

Now change its Class from Channel Strip to Mode Change. Set the Mode to Mixer: Faders 1-8. (You don't switch Learn on when you're doing this.) In the CA dialogue window, click on the Mode called 'Faders 9-16'. With that highlighted, push the button on your controller that is to select faders 1-8. If the Mode immediately jumps to 'Faders 1-8', you've done it!

Repeat everything in the last two paragraphs, but apply it to new (No Mode) Controls you've created called 'Switch Faders 9-16' and 'Switch Faders 17-24'. Assuming all is well, you can now control 24 tracks of audio and the Stereo Out from hardware faders.

If some things don't work as expected, go

Continued over...

From previous page.
back over what you've done by selecting each mode and CA in the CA window, by clicking on it. Take a screen grab (Cmd-Shift-4) of a CA that works correctly, and compare it to the one that doesn't. Often, the difference is subtle, but once you've spotted it, it's usually easy enough to put right.

Creating your own configurations

Before we get hands-on with the Graphic EQ, I'm going to encourage you to solve a problem. This is what it is: I'm going to need every physical fader on the APC40 – including the stereo master and the as-yet unused crossfader, but…

Fader 9 is currently in (No Mode), meaning it's always the main Stereo Out. How are we going to assign it to the ninth band of the Vintage Graphic EQ (8kHz), without it also controlling the Stereo Out of the main mix? I'll show you that the exact problem on your own system later, but for now, you might want to think about how you would go about altering what we've done to solve the problem.

In fact, this little break in our step-by-step build is all about you. The chances are you don't own an APC40 controller, so what you learn from this book has to work for the hardware you use. Equally, this book has to give you the tools to create CA schemes that suit your needs, not mine. So how could you adapt what you've already learned to different hardware, different situations within Logic, and what you want to achieve?

Taking the 24 channel audio mixer as a starting point, you don't have to configure it as three banks of eight channel faders, plus Stereo Out. If you only have four faders but six suitable buttons, you could achieve control over 24 channels of level faders by writing four volume CAs, which you cut and paste a further five times, and assign to the relevant tracks, followed by six Mode Change CAs.

At the greatest extreme, you could have just one physical fader and 25 buttons, but the ergonomic value of that is questionable, as it would be easier to specify the Channel Strip as Selected Track. That way the fader would simply act on whatever Track you clicked on.

As we'll see when setting up the Graphic EQ, the Volume Parameter is not confined to setting the loudness of channel strips. It can be used on any rotary or slider you see on screen, and the physical control can be rotary or slider, regardless of how it's shown on screen.

In short, you now have the building blocks to control virtually anything you can see in the Mix window, including audio processor and instrument plug-ins.

Step-by-step: creating the EQ assignments

To complete the final part of this chapter, you'll need an instance of the LPX Vintage Graphic EQ inserted into one of the Mixer channels. It doesn't matter which one, because we'll be using Selected Track, meaning it will act on an instance of the EQ only if that channel is the one highlighted.

Next, we'll need to create a new Mode and name it Graphic EQ. Then we'll populate our new Mode, using 10 Volume CAs, in the same way as we did for faders 1-8 earlier on. Fortunately, the APC40 has 10 faders, if we include the Master and the crossfader. If it did not, we could either use rotaries on the controller, or we could use two banks of five faders and two buttons for Mode Changes. Which approach you take depends on the capabilities of your controller, and your own preferences. This is a good point to write your first eight EQ assignments. That's when you'll see the problem.

Unfortunately, the Master fader is already assigned to Stereo Out. Perhaps you've already guessed how we're going to deal with that… If you don't deal with the issue before making the CA for the 8kHz slider, you'll get up an alert asking if you want to keep the twin assignments, or cancel. Probably best to cancel, while we sort this out.

As you know, the Master Volume CA is permanently available, because it is in (No Mode). That's but fine until now, but it's clearly got to be moved. But if we put it in one of the other Modes, we'll only have a Master fader when, say, faders 1-8 are selected, which isn't much use.

The solution is as simple as a 'cut & paste'. Click on the Master CA, Cmd-X to delete it and use Cmd-V to paste an instance in all Modes, except the Graphic EQ. Try it – it works! Now

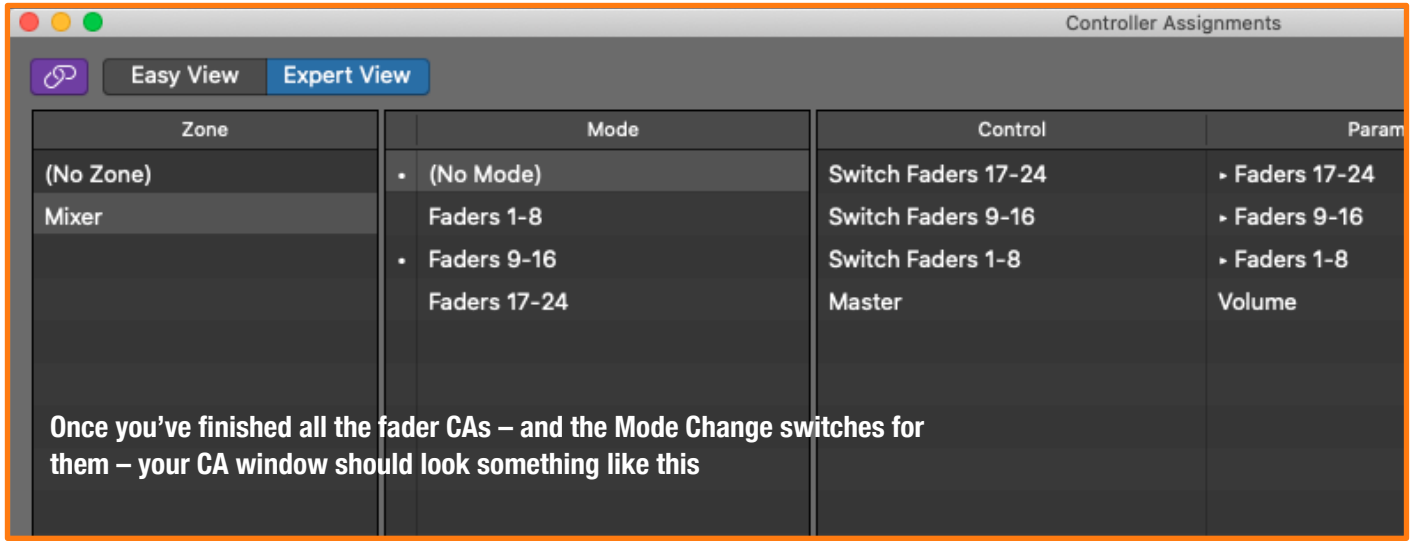
Continued over...

From previous page.

you can go back to the Graphic EQ Mode and write the CAs for the final two frequency bands.

Note – Despite the fact we've moved the Master CA, you may still get an alert to say this control is already assigned. Click the blue Reassign button and all will be fine.

After that, it's a question of creating one more Mode Change control in the (No Mode), and we're finished. Don't forget: when we create a Mode Change, we only use Learn to push the button we want to use. Then we turn Learn off and manually change Class to Mode Change and Mode to 'Mixer: Graphic EQ'.



Once you've finished all the fader CAs – and the Mode Change switches for them – your CA window should look something like this

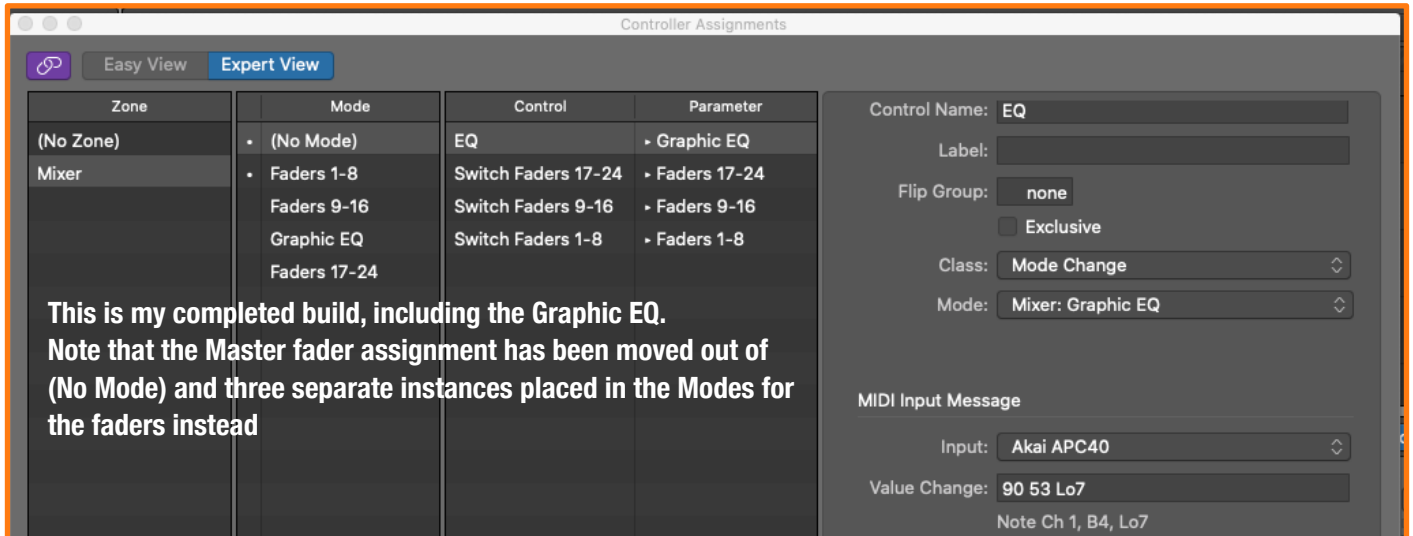
Summary

Well done. If you've followed this chapter carefully, you should now have a complete mixer, as described at the start. However, if there are any parts of your creation that don't work properly, I suggest you use the CA window to go through the parts that do work and compare with the ones that don't. Take screen shots and you can put them side-by-side until you spot where things have gone wrong.

Believe me, I had to do that plenty of times while writing this chapter, not to mention, having to change seemingly random MIDI values to 0/127 by writing the numbers in. Don't forget: the sooner you correct mistakes, the better the system understands your intentions, and the less mistakes you'll get. I'm sure you're much faster at writing QAs than you were at the start of the chapter. You'll get even faster as you get more practice.

Hopefully, it's not just the mechanics of writing QAs, but also how to plan and design your project, so you can design your own schemes.

The rest of this book assumes you've already worked through this chapter, so it won't be babying you by telling you every single step. Instead it will show you what software the scheme is for, how it's designed to work and what the QAs look like. Of course, your MIDI input source and value change will probably be different, but you should be able to adapt the designs for your own builds. □



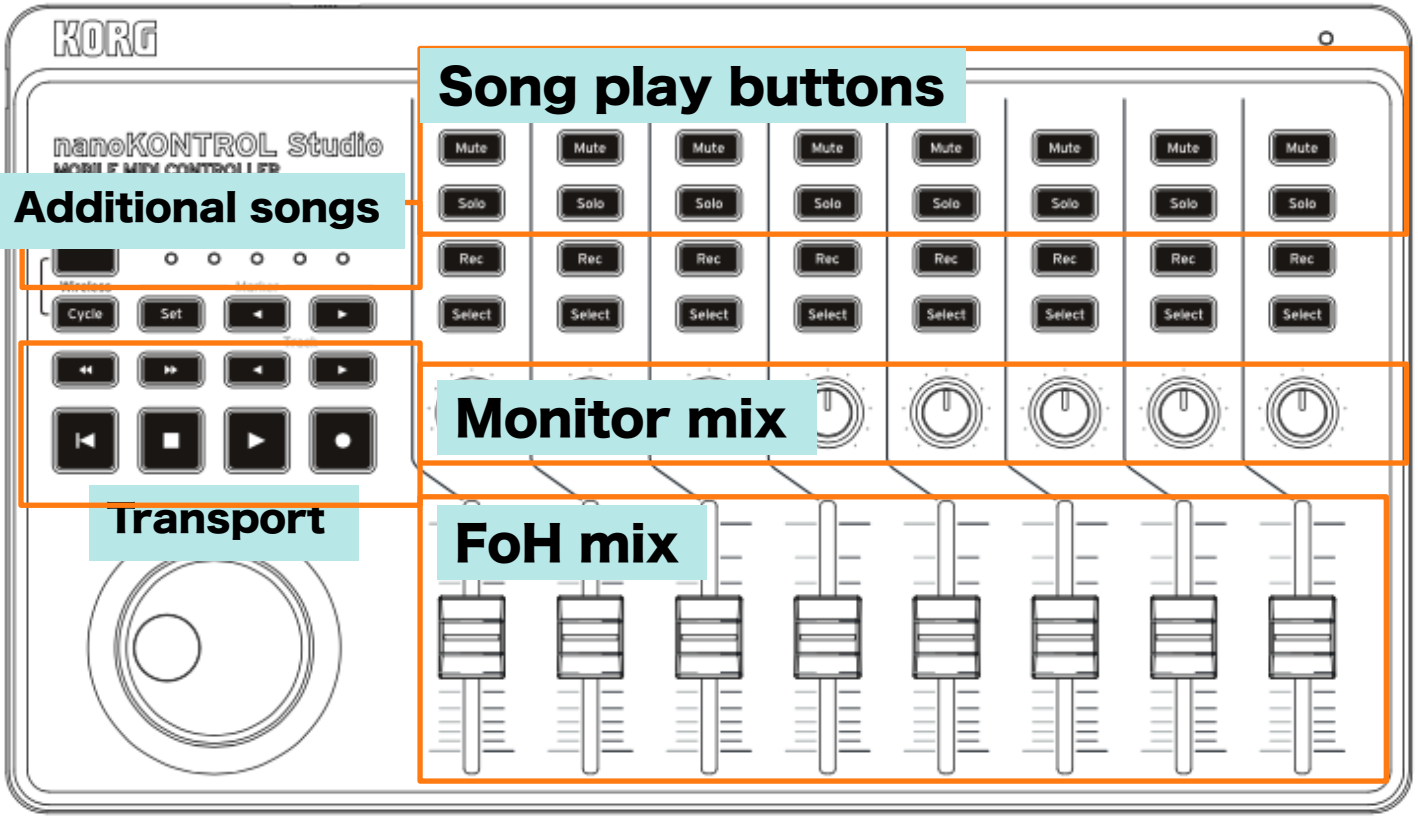
This is my completed build, including the Graphic EQ. Note that the Master fader assignment has been moved out of (No Mode) and three separate instances placed in the Modes for the faders instead

Live performance system with backing tracks and mixer

What you'll learn in this chapter:

How to create a multi-song playback system for live performance, with an eight channel Front of House mixer, plus monitor mix. You'll be using Markers and Go To Marker Parameters each song at the push of a button.

I set myself the challenge of creating an ultra-compact, low-cost, automated show assist system that could bring big-show technology to a duo paying small venues. C'mon, you can't fault me for lack of ambition…



Overview

I chose a Korg nanoKontrol as the MIDI controller*. This was partly because it's tiny, but also because I already owned one, so this exercise was going to cost me zippo. I loaded Logic onto a Mac Mini I picked up locally for £130 to drive the display system I developed to show the assignments on my Akai APC40. My Focusrite Saffire Pro 26 interface would give me all the outputs I needed, so the only thing I needed to buy was a tiny monitor screen. Designed for home security CCTV applications, it cost me about £60 on ebay. The monitor isn't actually needed to drive the simplified version of the automated system, but it's absolutely essential when setting up.

Continued over...

Say 'ahh": The tiny monitor is dwarfed by even a nanoKontrol

**If you too are using a nanoKontrol, you might want to consider using the lower two rows of buttons for the Marker assignments, because in the more advanced version of my scheme – which I explain at the end – the top two row of buttons are used for what it says on them, Mute and Solo.*

From previous page.

The ‘jukebox’ side of the system comprises a succession of sound files, all lined up on the same track, and each with a Marker at its head (see Right). A series of 16 buttons on the controller is used to give instant access to the first 16 songs, while the Scene button* allows access to songs 17-20, which is the highest number for which there are ‘Go To Marker Number…’ parameters available.

(There is a revised design, which will give you instant access to 128 songs, but it complicates the scheme, so I’ll deal with that at the end of the chapter. Meanwhile, you might want to figure that one out for yourself.)

Although I’ve only shown the system with example stereo files of songs, there is scope for an optional click/cue track, which I’ve routed to Bus 3 in Logic, along with the rest of the monitor mix. This mix is on the eight rotaries and is as follows: backing track; click track; guitar 1; MIDI guitar 1; vocal 1; guitar 2; MIDI guitar 2, and vocal 2.

The Front of House mix follows the same scheme, with the obvious exception of the click track, which you don’t want the audience to hear.

To avoid disappointment, I should point out that attempting to mix on faders with such a limited throw turned out to be a bit impractical in a live setting, especially as the faders do nothing until they have reached the same MIDI values as the fader setting in Logic. That means it’s quite easy to make something louder before you can turn it down!

What does really work is using Logic’s automation to write in mutes and effects settings. A big part of my scheme involved using the excellent Jam Origin MIDI guitar 2

Continued over...

**This function on the nanoKontrol no longer works under LPX 10.6.0, meaning pushing the Scene button has no effect on the MIDI messages coming from the buttons. It does however work under macOS 10.13.6/ LPX 10.4.8 on the Mac Mini.*

From previous page.

plug-in. The ability to automate when it was active, and which patches it was playing, effects settings etc, proved very powerful.

The only limitation with that is you may want to use Low Latency Mode to avoid unwanted processor delays. Doing that can limit the amount of plug-ins you're running.

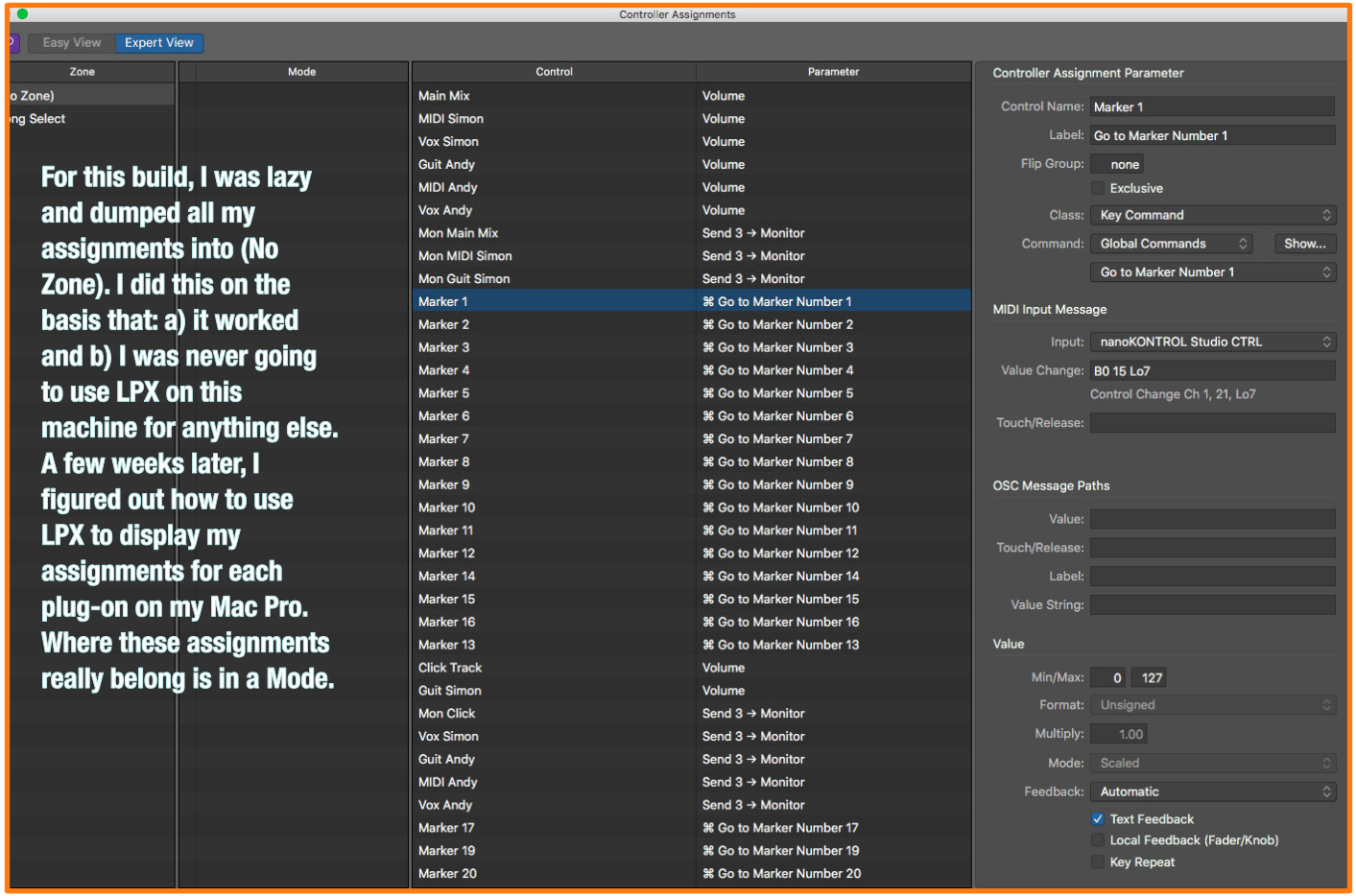
Solutions – How to get round the fact there are only 20 'Go To Marker Number…' parameters available. I'm really hoping you've come up with some possible ways round this yourself, but here are some pointers.

First pointer: Logic is a multitrack system (the latest version will actually support 1,000 tracks). As we probably only need one stereo track for the backing and perhaps a click/count-in, there's plenty of spare capacity there!

So if we create another two tracks, with songs 21-40 loaded, we can potentially play all 40 out on demand. To make this work, it's a good idea to space your sound files out along the timeline. That's because Marker 2 has to be the start of both Song 2 and Song 22.

If Song 2 is short and Song 22 is long, you need to leave enough space between your sound files to accommodate them both, so that the Marker for Songs 3/23 can line up to the song starts.

Second pointer: Under LPX 10.6.0, a Korg nanoKontrol is automatically recognised, and a premade set of CAs is inserted into the 'com.apple.logic.pro.cs' Preferences file. These make almost all the controls do what they say on the buttons. The chances are you're not



using a nanoKontrol, but you might like to see how the Mute button is configured.

Alternatively, if you not using click/cue tracks, you could use the Solo function to select the track you actually want, as opposed to muting the ones you don't. Given that it is possible to assign more than one CA to a single button, there must be a way togging the functions so that a single button press selects one track by unmuting all others, but I haven't managed to get that to work as yet.

Assuming you use eight tracks, each loaded with 16 songs, that's 128 songs under your fingertips, so working out how to streamline the track selection process is certainly worth the effort. If you get there before me, please share

If you are using a nanoKontrol, then you may

Continued over...

From previous page..

want to use the lower two rows of buttons for the CAs that select the songs. That way, the Mute and Solo buttons can still keep the functions that are printed on them.

Design considerations

My scheme will work with any MIDI controller than has eight or more faders, and ideally 16 or more buttons. If your controller is better equipped than that, you can hopefully adapt my basic ideas to get even more mileage out of them.

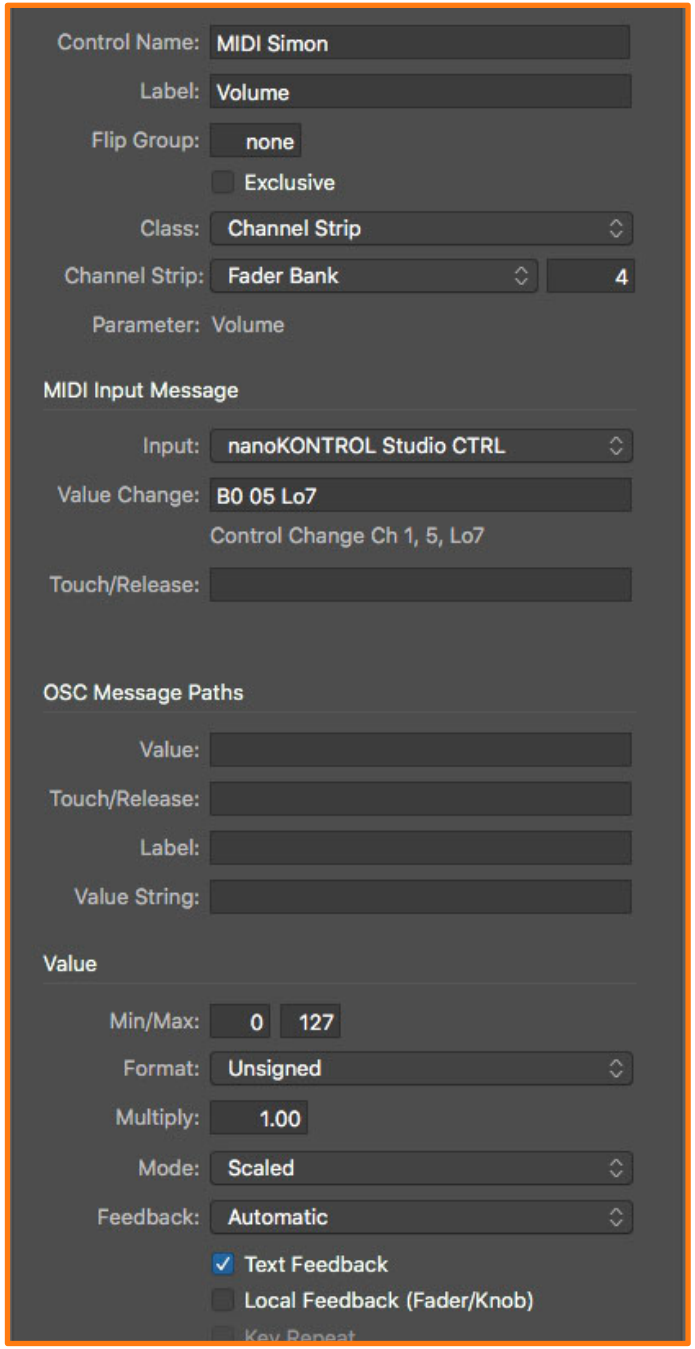
However, the more complex the scheme becomes, the more chances there are that something will go wrong, which obviously is what you very much want to avoid during a live show.

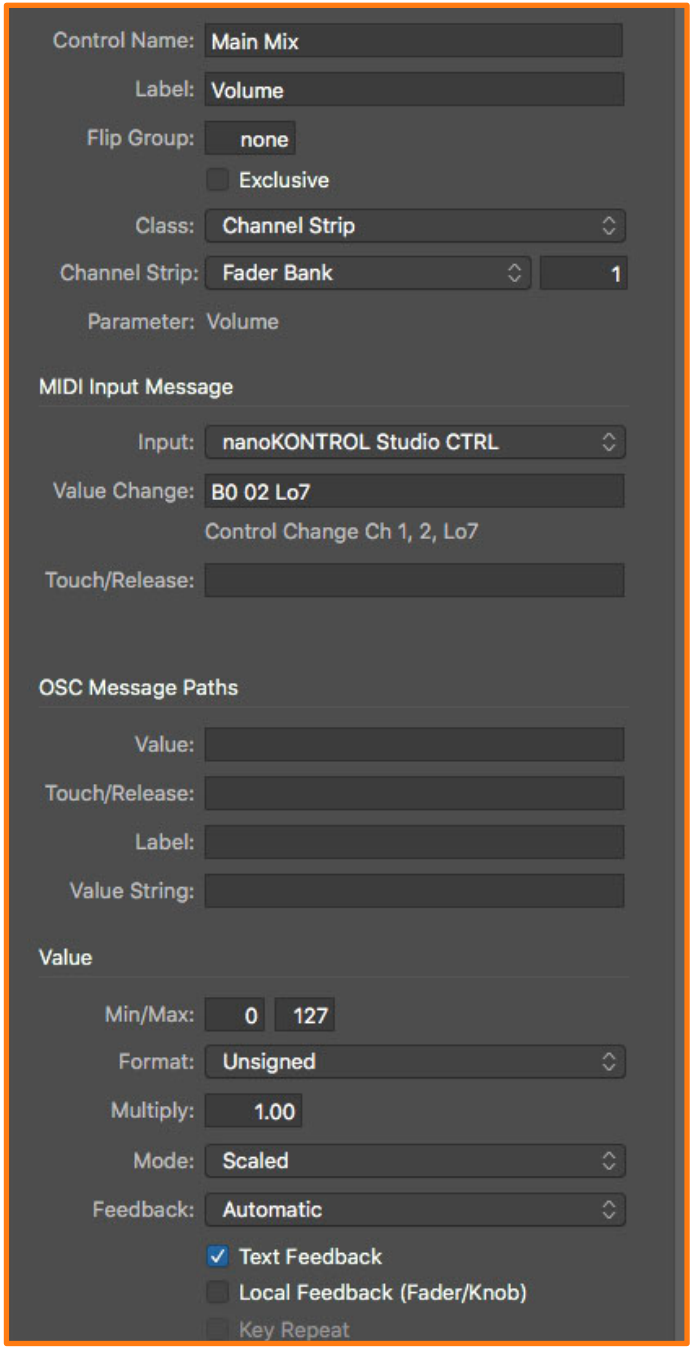
In particular, if you decide, to harness the Mute And Solo buttons to increase the number of available songs, the chances of you cueing the wrong song – or worse, playing two simultaneously – will also increase.

Although I wanted to fire your imagination and show you what *could* be done, I'm not suggestion it *should* be done. Personally, I like to keep things as simple as possible when performing.

My solution in the end was to forget Scene changes and simply put songs 17- 20 on the next row of buttons above the faders. After 20 songs, you're generally entitled to a break, which gives you plenty of time to load another LPX project file with the next 20 songs.

The rest of this chapter is screen grabs. □





Control Name: Click Track
Label: Volume
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Fader Bank 2
Parameter: Volume

MIDI Input Message
Input: nanoKONTROL Studio CTRL
Value Change: B0 03 Lo7
Control Change Ch 1, 3, Lo7
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Scaled
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)

Parameter
Solo
Solo
Solo
Solo
Solo
Solo
Solo
Solo
Mute
Mute
Mute
Mute
Mute
Mute
Mute
Mute
Record Enable
Record Enable
Record Enable
Record Enable
Record Enable
Record Enable
Record Enable
Record Enable
Select
Select
Select
Select
Select
Select
Select
Select

Control Name: Mute Button 1
Label: @tp
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Fader Bank
Parameter: Mute

MIDI Input Message
Input: nanoKONTROL Studio CTR
Value Change: BF 30 7F
Control Change Ch 16, 48, 1
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Toggle
Feedback: Automatic
Text Feedback
Local Feedback (Fader/K
Key Repeat

Control Name: Solo Button 1
Label:
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Fader Bank 1
Parameter:

MIDI Input Message
Input: nanoKONTROL Studio CTRL
Value Change: BF 20 7F
Control Change Ch 16, 32, 127
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Toggle
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat

Some example plug-in Controller Assignments

There isn't much text in this chapter, because it's mainly about screen shots you can copy to create CAs for specific Logic plug-ins. Similarly, I haven't shown the physical layout for each plug-in on the controller, because the chances are, you'll be using a different controller to me.

As a result, your MIDI Input Message section in the CA window is likely to look different to the ones shown. As long as you pick an appropriate type pf control (momentary button, latching button, rotary etc) you should achieve workable results.

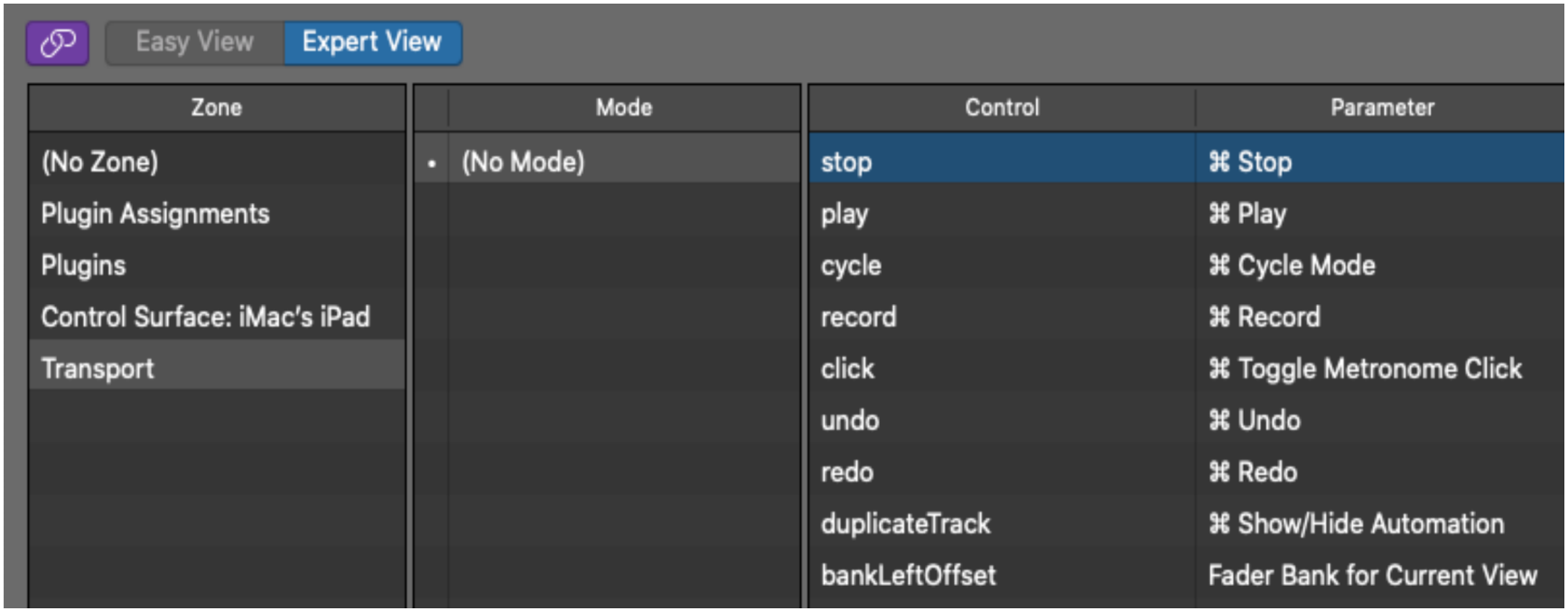
Some of these examples are a few years old now. While I believe they are all functional, I lost the use of a few plug-ins after a system upgrade, so I can't check them. They worked last time I used them!

Some of the ground covered on the next few pages is covered elsewhere in the book. This is because some people find it easier to learn from a written explanation, whereas other people may find a more graphical approach helpful.

In the end, I decided to limit the number of plug-ins shown in this chapter, because a lot of them are pretty much the same thing for a CA perspective; you see a rotary on screen you go Cmd-L twist a rotary on the controller and key Cmd-L again. Really, it's all in the planning. □

Mono and stereo are not the same

It's worth knowing that CAs written for the mono version of an LPX plug-in won't work for the stereo version, and vice versa. If you want to use both, you'll need to write CAs for both. However, you can use the same physical controls and store the CAs in the same Mode. This might seem a bit strange, but it's just the way it is.



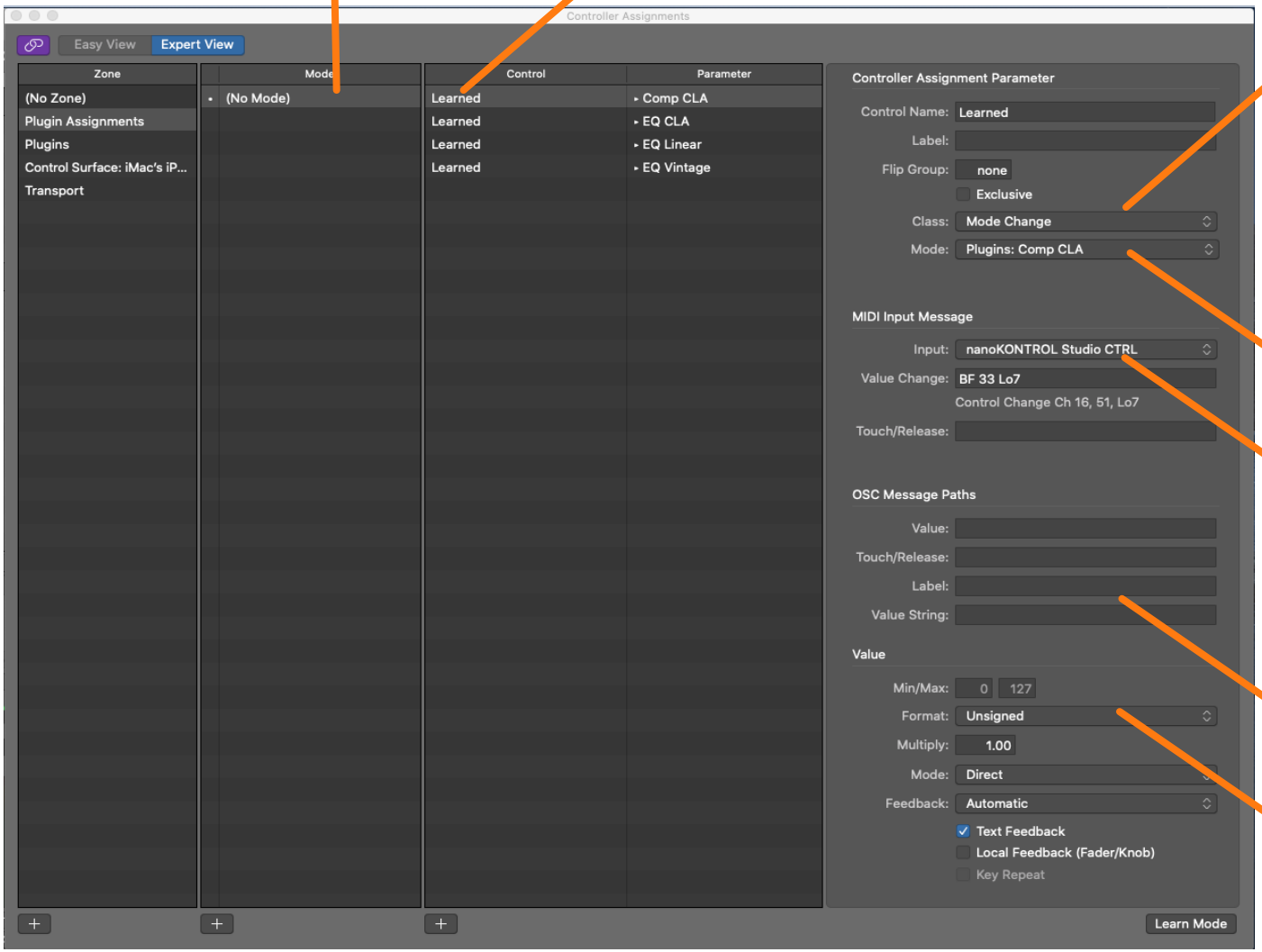
You will occasionally come across controls in plug-ins that it seems impossible to create CAs for. There are a couple of ways you may be able to get round this. One is to change View from Editor to Controls and assign to them. The other is to assign to Smart Controls instead of the plug-in.

How I'm switching between plugins: an overview before we get started

I've placed these Assignments in Plugin Assignments/(No Mode), because I want them to work regardless of which plugin Mode is currently active.

'Learning' is as simple as clicking on the Learn Mode button (bottom right), moving the desired button/knob on your controller, then making the Parameter changes explained next.

Make sure Learn Mode is *off* before you go any further!



Class needs to be set to 'Mode Change' from the pulldown menu.

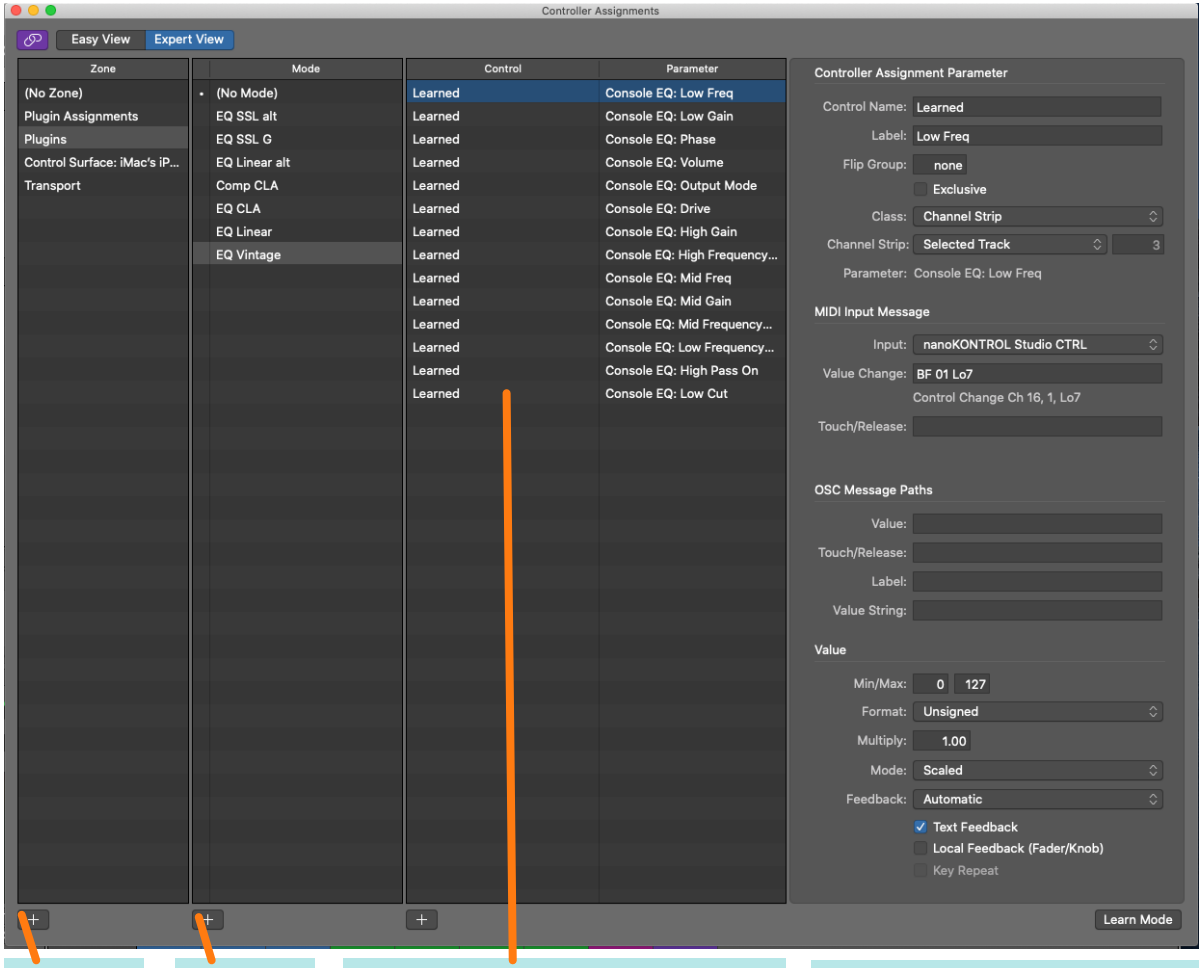
The **Mode** is chosen from the available plugin Modes, which I've created within the Plugins Zone. As the pull-down menu will only be populated once you've actually created some plugin Modes of your own, I suggest you read this overview, then continue reading until we get to the hands-on stage.

If things don't work as planned, one place to look is the **MIDI Input Message** section. If you accidentally nudged a different controller wile in Learn mode, the Input may not be the one you think it is. Equally, a little 'finger trouble' can lead to two Value Changes being entered.

This section should be left blank

You will probably need to change some of the **Values**. The correct ones – shown here – are available from the pull-down menus.

Let’s get cooking! Creating Assignments for the Vintage EQ



Press here to create a new **Zone**.

Press here to create a new **Mode**.

Providing you have a Mode selected, activating Learn Mode should automatically create a new Assignment in this window.

Learn Mode button turns blue when active. Make sure it has reverted to grey when you have assigned a control.

I haven’t assigned a physical button to the In switch, as it is unlikely to get enough use to be a priority. However, if you want to assign one, the structure of the Assignment is exactly the same as those used to switch in the various sections of the EQ (ie Low Cut, Low, Mid and High).

In order to create some consistency between plugins, I’ve stuck wherever possible to the orthodoxy that EQ gains are on the rotary controls (originally used by Korg as pans) with the frequency sweeps on faders (the mixer channel levels in Korg’s scheme).

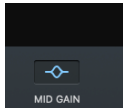
As you’ll see on the next pages, deciding how to assign the Output Model, Phase and Volume functions in the Output was a little more of a judgement call. Your solutions may vary from mine. Have fun!

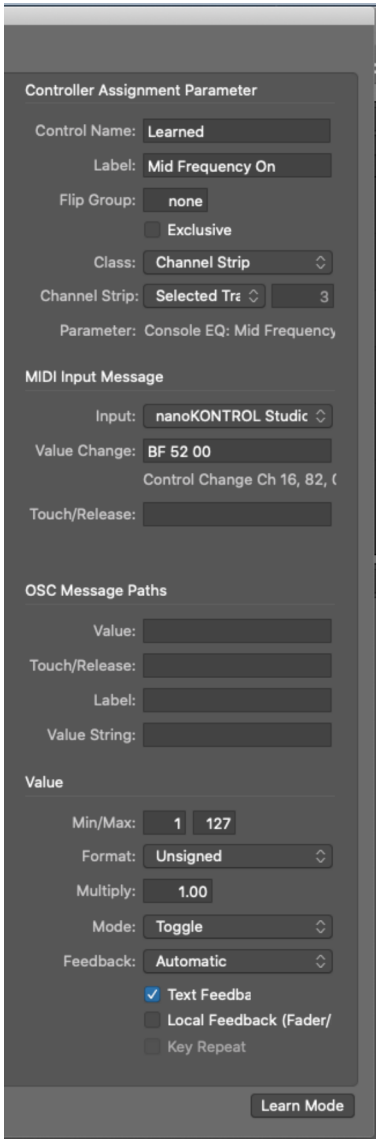
See next page for the full Vintage EQ walk-through with screen-grabs

NB – If you don’t manually create a new Control Assignment, the order in which they appear in the Controller Assignment window is somewhat haphazard, which is why mine don’t reflect the physical positions of the controls on screen. However, I have been very careful to ensure that the positions of the physical controls mirror those on-screen, to make operation more intuitive.

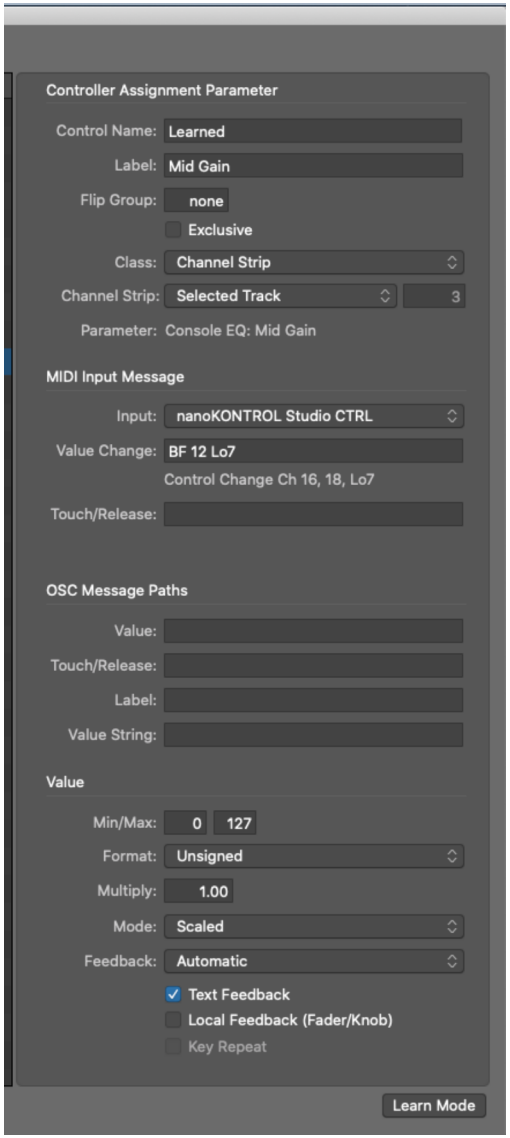
Creating Assignments for the Vintage EQ: the walk-through

This Assignment will work for all On/Off toggles

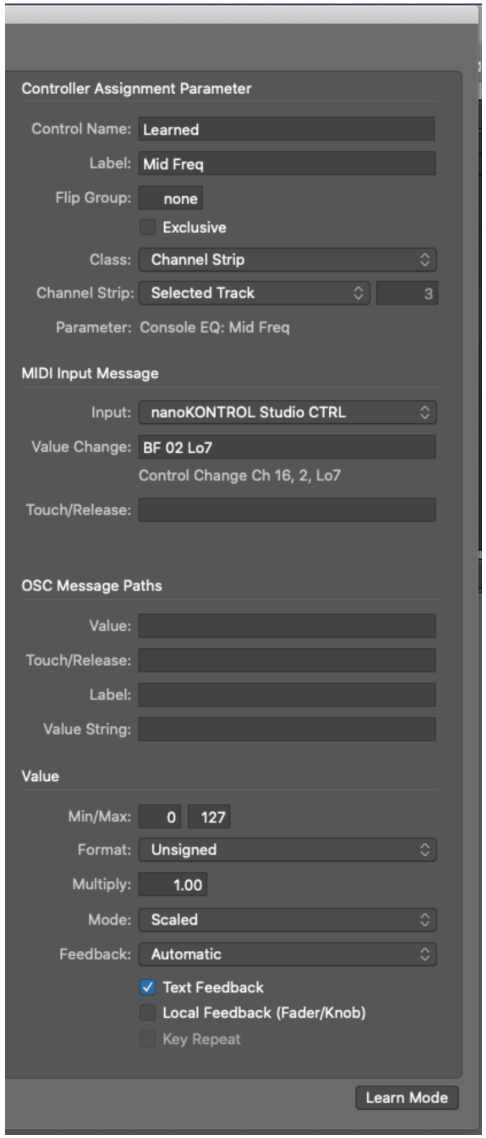




This Assignment will work for all Gains, Cuts etc



This Assignment will work for all Freq Sweeps



Notes

As a general observation: While it is certainly possible to cut 'n' paste Assignments, then edit the copy, I found it to be quicker to simply Learn a new Assignment.

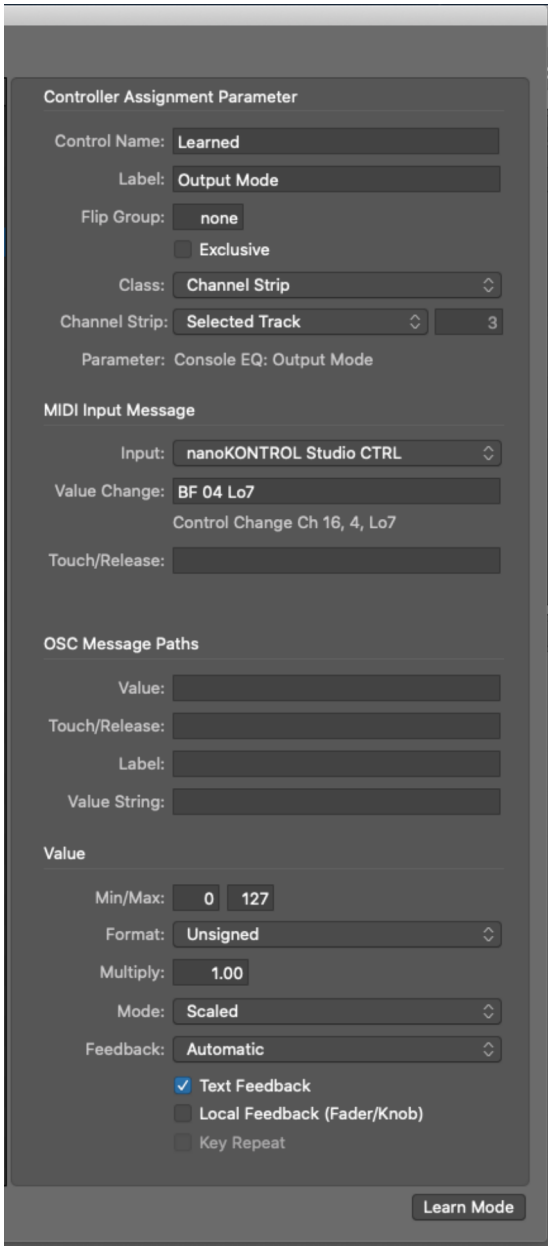
The difference between the Gain and the Freq Assignment. There isn't one! I just wanted to show they are basically identical whether you use a rotary or a fader.

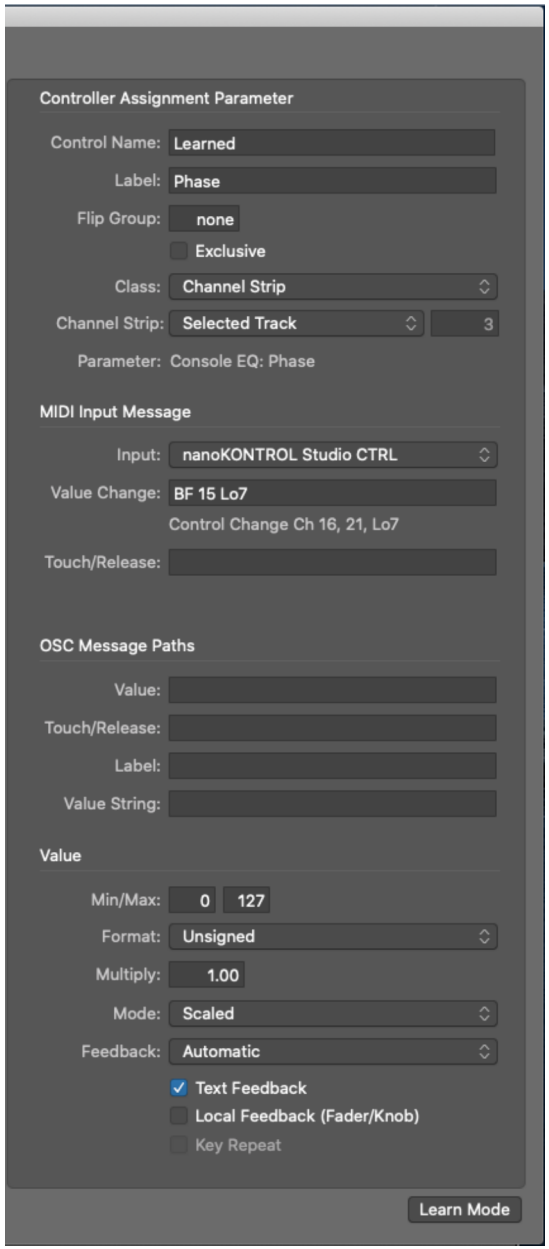
Class: Channel Strip
Channel Strip: Selected Track
Together, these define the general type of action and what it is acting on.

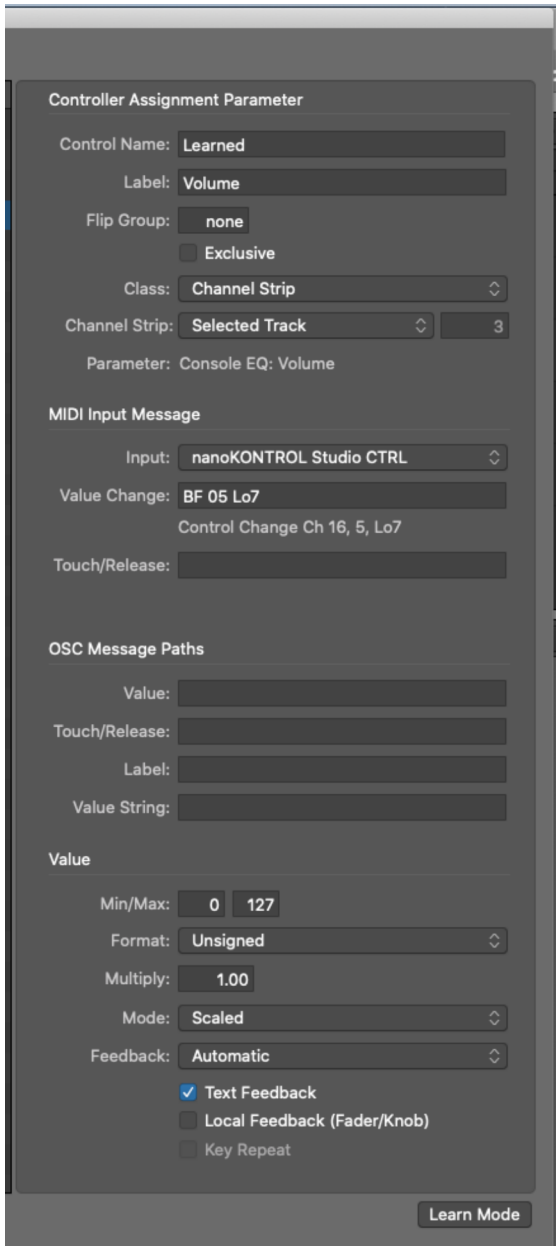
Value Change:
Obviously, the actual MIDI value will vary from case-to-case, but keep an eye out for unusually long numbers, because it may mean you've accidentally created more than one Value Change.

Value Min/Max:
Strictly speaking they should all be 0/127. For some reason, my nanoKontrol won't always Toggle unless the Min is set to 1.

Creating Assignments for the Vintage EQ: the walk-through







Notes

There is nothing in these Assignments that probably leaps out at you, but there is a reason I'm showing them to you.

Output Mode(l) uses a fader to scroll between four states: Silky, Punchy, Sooth and Off.

Phase uses a rotary knob to switch between Natural and Linear operation.

Volume is assigned to a fader.

This tells us that there is *no necessary correlation* between the way a control acts and looks on screen and the physical control we assign to it. You'll soon see why that matters when we apply what we've learned to LPX's full parametric EQ.

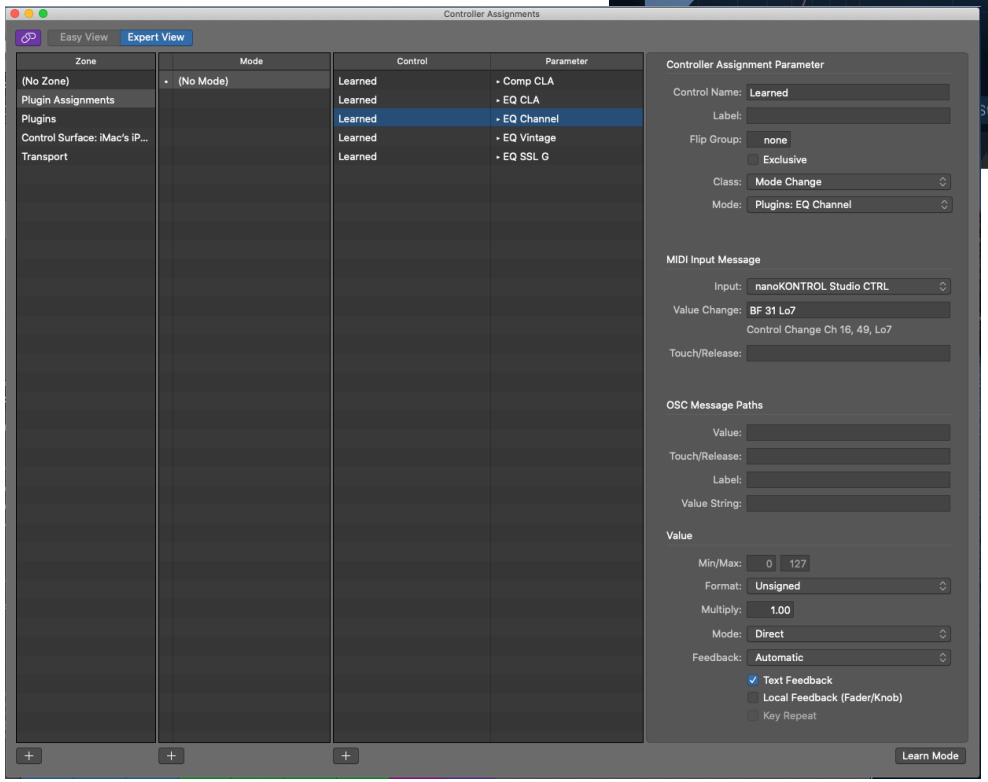
Creating Assignments for the Channel EQ/Linear EQ

The Assignments we covered for the Vintage EQ are very much applicable to the Channel EQ, but there is a complication: there are more parameters to address than I have physical rotaries and faders. Therefore, it is necessary to reassign some of the controls to a secondary Mode.

This is where I have an embarrassing confession to make. Although I've managed to assign all the EQ parameters, apart from the Gain, I don't fully understand how the mechanism I've created actually works.

It all starts off straightforwardly enough, with a button on my controller assigned to the Mode 'Plugins: EQ Channel'. Just like the Vintage EQ, this uses the Class: Mode Change.

But, take a look inside the EQ Channel Mode and you'll see it contains a further Mode Change, this time to the Mode 'EQ Channel alt'. As you'll see on the next page, the 'alt' mode is where all the additional Assignments are stored.


Controller Assignments
Easy View
Expert View
Zone
Mode
Control
Parameter
(No Zone)
Plugin Assignments
Plugins
Control Surface: iMac's iP...
Transport
(No Mode)
Learned
Comp CLA
EQ CLA
EQ Channel
EQ Vintage
EQ SSL G
Controller Assignment Parameter
Control Name: Learned
Label:
Flip Group: none
Exclusive
Class: Mode Change
Mode: Plugins: EQ Channel
MIDI Input Message
Input: nanoKONTROL Studio CTRL
Value Change: BF 31 Lo7
Control Change Ch 16, 49, Lo7
Touch/Release:
OSC Message Paths
Value:
Touch/Release:
Label:
Value String:
Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Direct
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat
Learned
Learned
Learned

Creating Assignments for the Channel EQ/Linear EQ

… From previous page

In the 'EQ Channel alt' mode, the faders that were previously assigned to frequency sweeps are now assigned to the Q parameter.

Note that there is also a Control to execute a Mode Change, this time linked back to the main 'Plugin: EQ Channel'.

How it works?

OK, let's just take a breath here, and talk through how this 'should' work in LPX.

The top row of buttons on my controller is dedicated to selecting plugins. From L-R, the first is the Vintage EQ, second is Channel EQ, and so on.

Let's assume my selected channel in Logic has a Channel EQ inserted. I click on it, and press the second button along on my controller.

Now, the buttons just above the rotaries toggle the individual bands on/off, the knobs control gain and the faders sweep the frequencies.

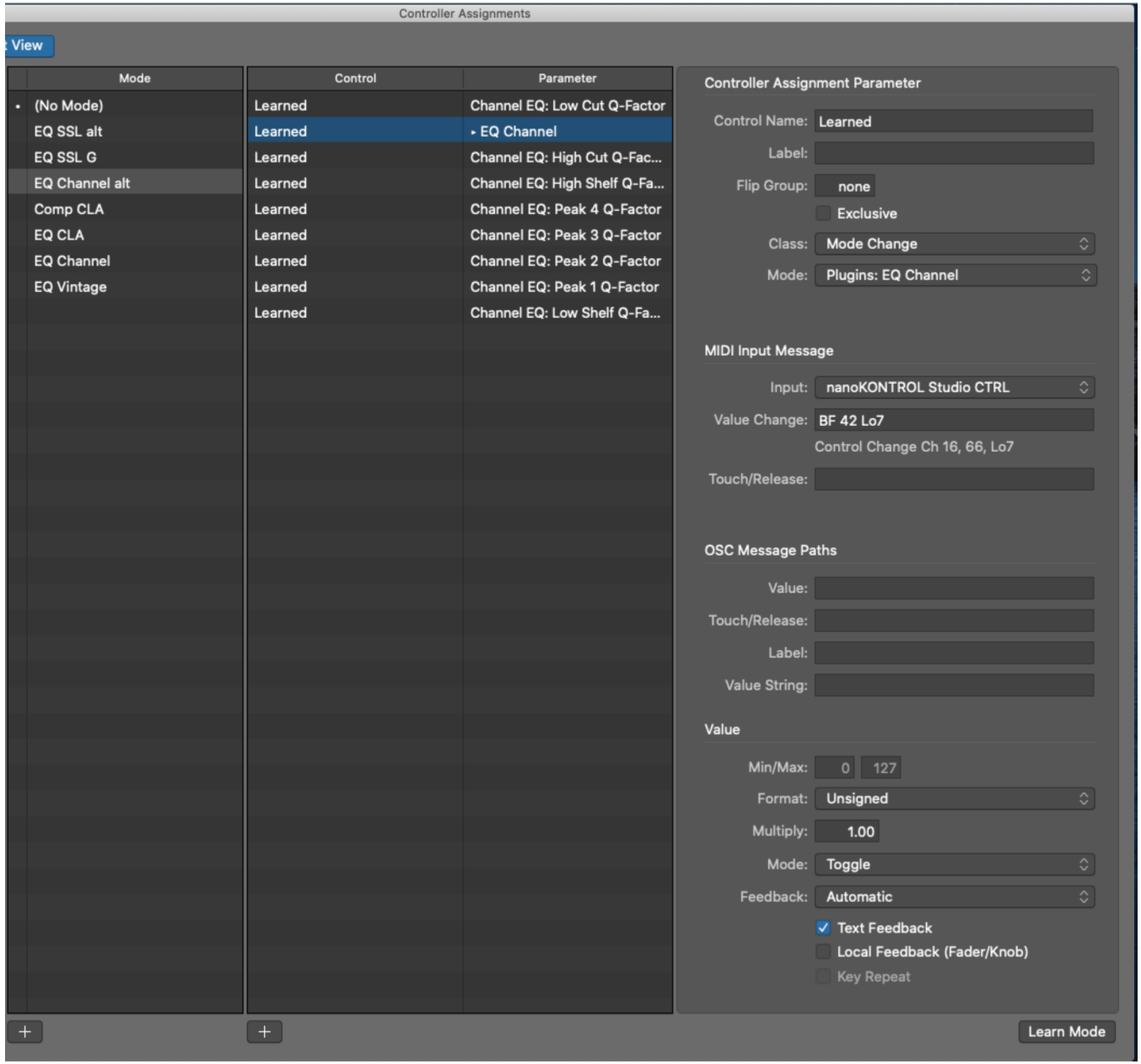
One button on the controller toggles the Mode Change, and now the faders control the Q values. Push that button again, and it toggles back to the main 'EQ Channel' functions.

How it works!

What it actually does is similar, but not quite what I was expecting.

When you hold the Mode Change button down, it momentarily assigns the faders to Qs, but if you release it and move a fader, it's straight back to the main Channel EQ mode.

Luckily, it's really usable that way – to the extent I may have actually chosen it – but I'm wary of anything that works accidentally! □



Creating Assignments for the Waves CLA MixHub

This plugin gets a different treatment to the earlier ones. The MixHub EQ section and the Dynamics sections' Assignments are stored in separate Modes, with each given a dedicated button on my controller.

We could map this synth by using multiple Modes, one for each section

On the bottom right of this page, I've shown which buttons I currently have my first three plugins assigned to. As we've already seen, I could have used a Mode Change Assignment *within* the main CLA Mode, treating the Dynamics section as 'CLA alt'. This is what we did with the Channel EQ.

I didn't do it that way, partly because I like the idea that the EQ and DYN sections sit side-by-side, right at the top of the controller. But also, I really wanted to test my assumption that it was possible to address *different sections within* a single plugin.

Right now, that concept might not seem terribly important, but consider this: without the ability to control a single plugin by Assigning multiple modes, even a MiniMoog takes up a large number of physical controls.

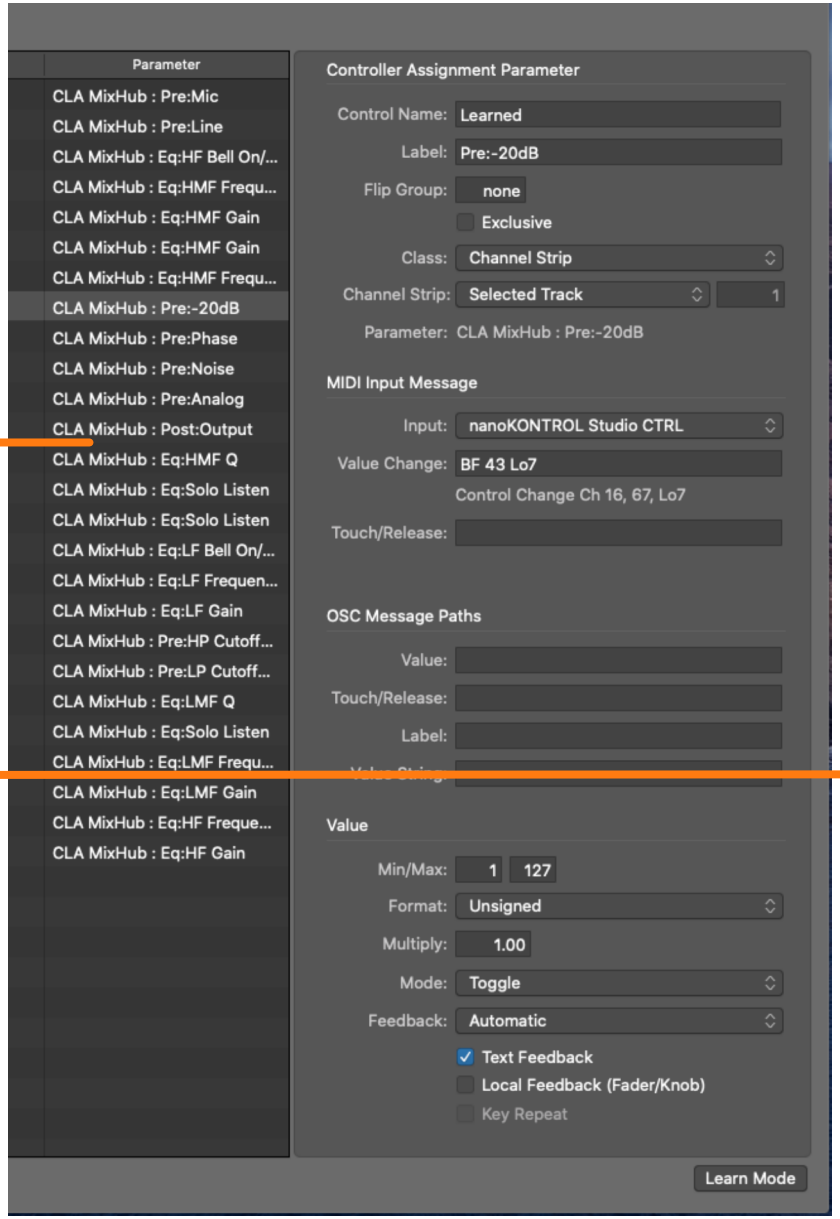
By the time you get to a polyphonic synth from the 1980s, things might get a little unmanageable with this controller. □

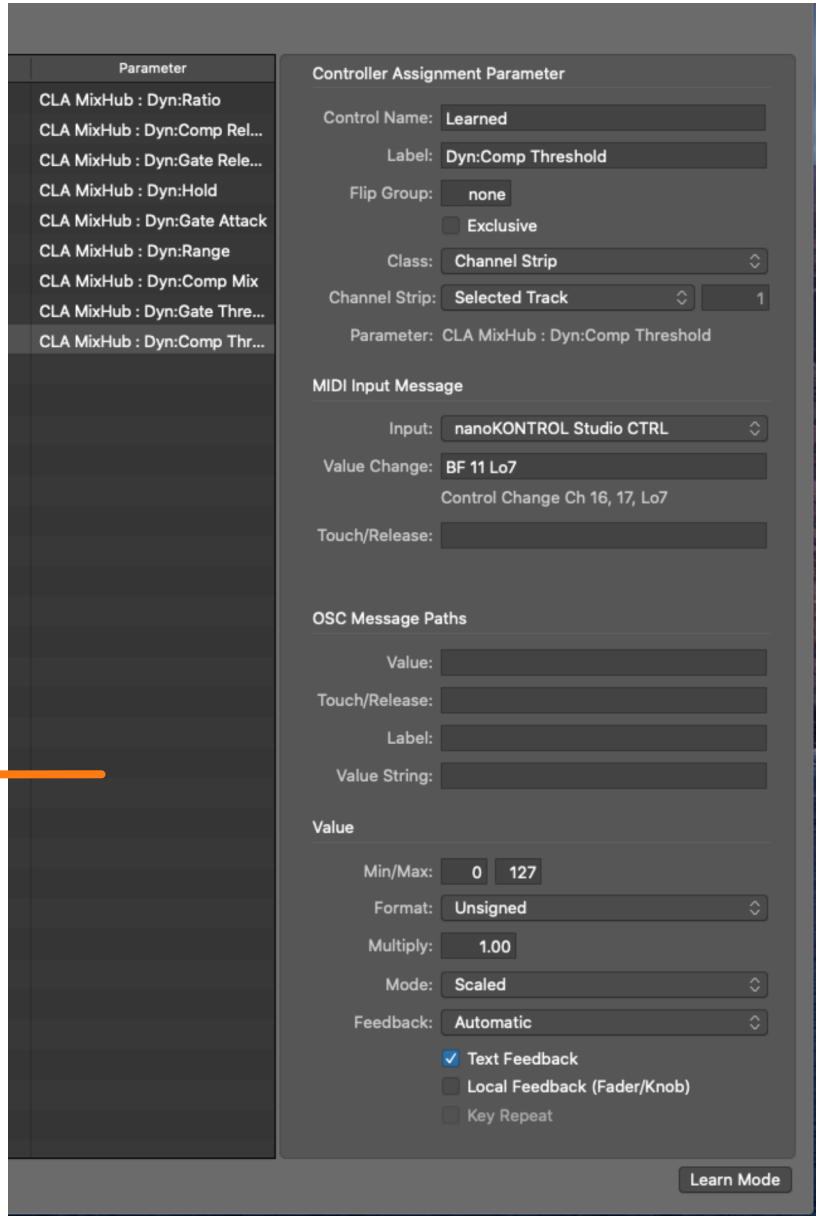
Creating Assignments for the Waves CLA MixHub

I haven't shown the Plugin Assignments for EQ CLA and Comp CLA, because they're basically the same as for the other plugins.

The EQ CLA Mode includes Assignments for the input section (input levels, filters etc), as well as the main fader.

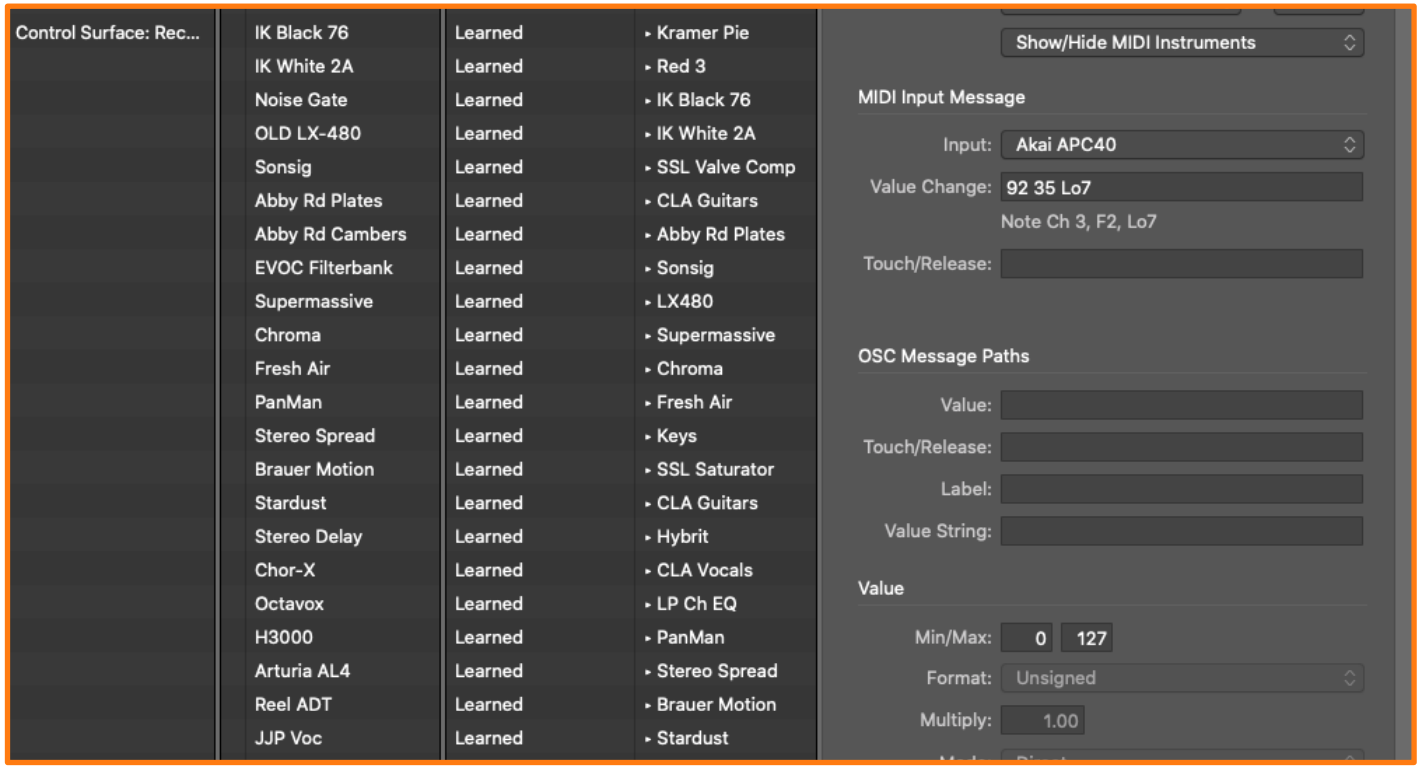
In contrast, I have yet to finish all the Assignments for the Dynamics section, because I'm still mulling over the best controls to use from an ergonomic perspective.

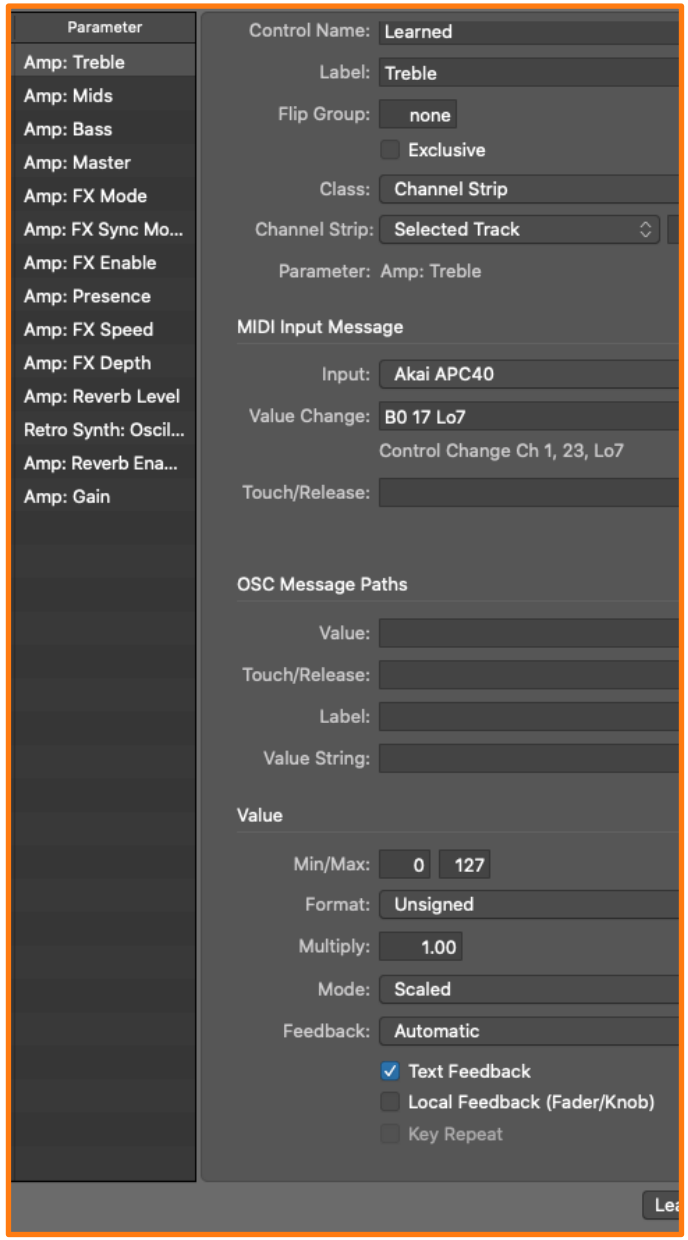




Creating Assignments for the LPX Amp Designer

Above: Once you've done CAs for one amp, it works for all of them. Below: This took me a while…





Control Name: Learned
Label: Mids
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Selected Track 1
Parameter: Amp: Mids

MIDI Input Message
Input: Akai APC40
Value Change: B0 16 Lo7
Control Change Ch 1, 22, Lo7
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Scaled
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat

Control Name: Learned
Label: Bass
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Selected Track 1
Parameter: Amp: Bass

MIDI Input Message
Input: Akai APC40
Value Change: B0 15 Lo7
Control Change Ch 1, 21, Lo7
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Scaled
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat

Label: Master
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Selected Track 1
Parameter: Amp: Master

MIDI Input Message
Input: Akai APC40
Value Change: B0 2F Lo7
Control Change Ch 1, 47, Lo7
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

Value
Min/Max: 0 127
Format: 2's Complement
Multiply: 1.00
Mode: Relative
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat

Label: Reverb Enable
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Selected Track 1
Parameter: Amp: Reverb Enable

MIDI Input Message
Input: Akai APC40
Value Change: 90 3A Lo7
Note Ch 1, A#2, Lo7
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Scaled
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat

Label: FX Enable
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Selected Track 1
Parameter: Amp: FX Enable

MIDI Input Message
Input: Akai APC40
Value Change: 90 3B Lo7
Note Ch 1, B2, Lo7
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Scaled
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat

Label: FX Sync Mode
Flip Group: none
Exclusive
Class: Channel Strip
Channel Strip: Selected Track 1
Parameter: Amp: FX Sync Mode

MIDI Input Message
Input: Akai APC40
Value Change: 90 3C Lo7
Note Ch 1, C3, Lo7
Touch/Release:

OSC Message Paths
Value:
Touch/Release:
Label:
Value String:

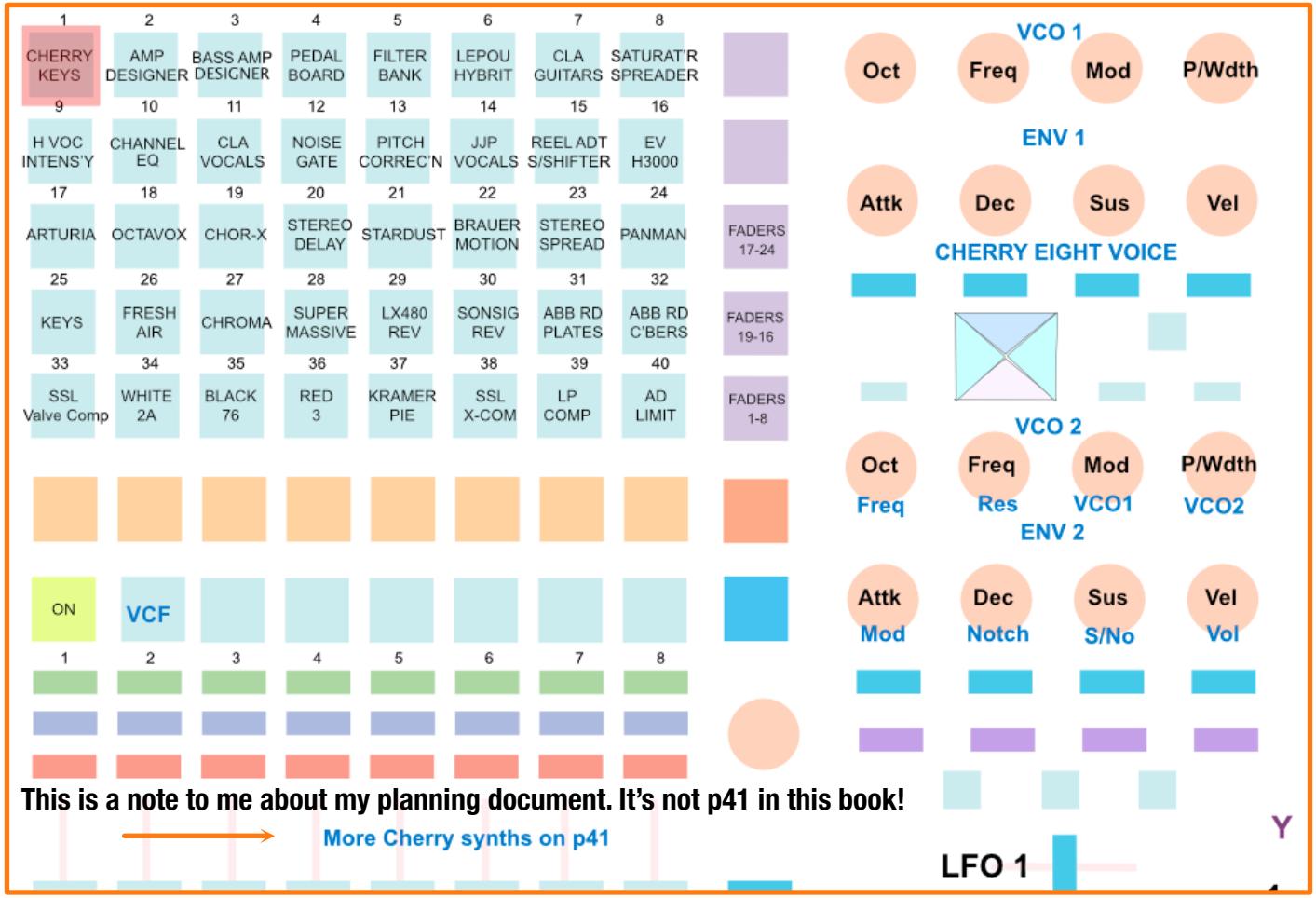
Value
Min/Max: 0 127
Format: Unsigned
Multiply: 1.00
Mode: Scaled
Feedback: Automatic
Text Feedback
Local Feedback (Fader/Knob)
Key Repeat

CAs for classic Cherry Audio keys bear fruit

What you'll learn in this chapter:

How to combine assignments for multiple synth plug-ins into one Mode. How to plan your build to get the results you want, so making your instruments more expressive and creative to play.

Well this is a challenge. How are we going to get one button to give us hardware control for several emulations of classic analogue synthesisers?

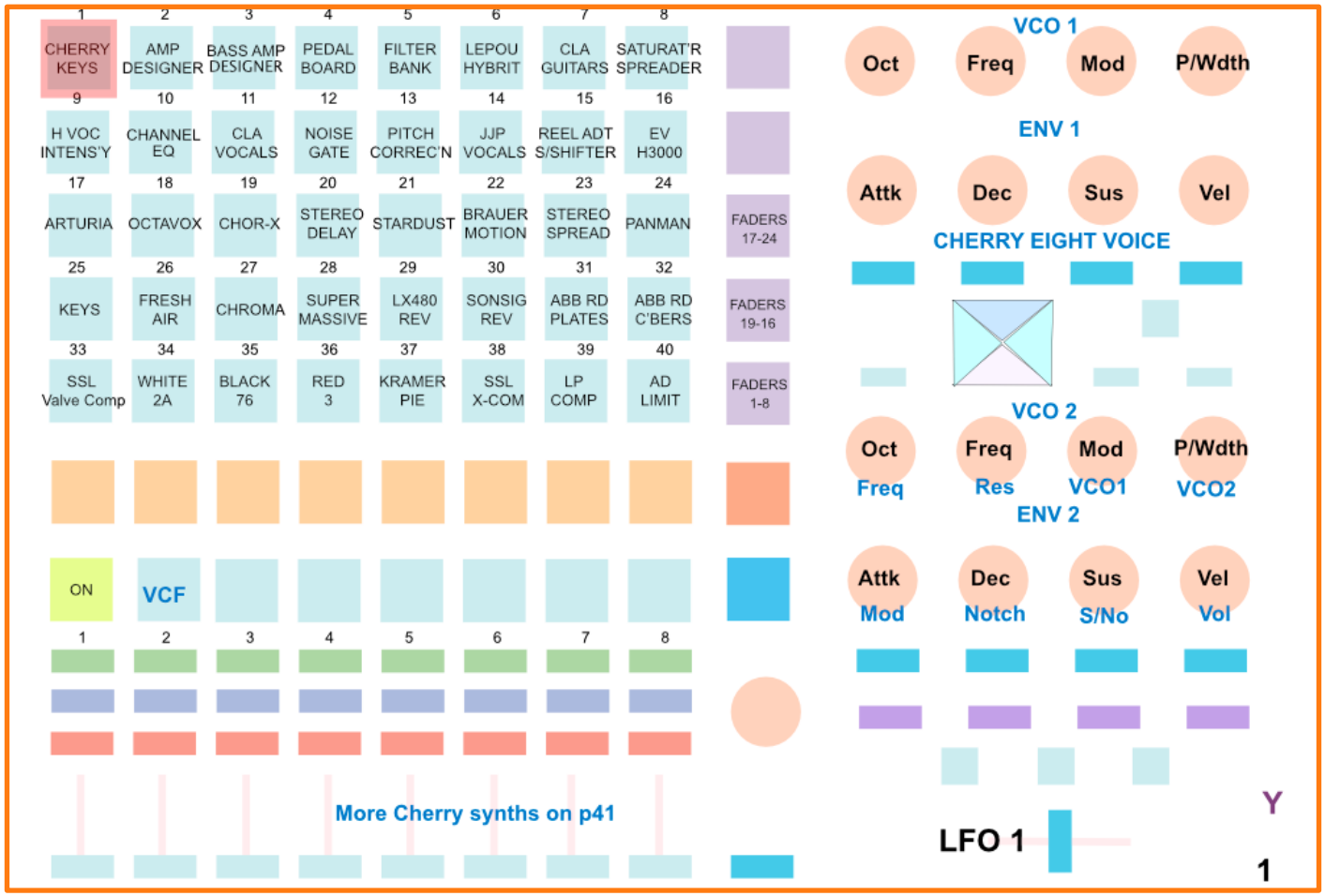


Welcome to my work in progress! This is the chapter where we go from doing difficult stuff to creating what might seem like impossible levels of real-time control over surfaces that were originally made of metal and were packed with knobs and switches. In short, we're going to make these synths live and breathe again as expressive performance instruments.

At the moment, I own five of Cherry Audio's fine classic 1980s emulations. You might be wondering: "So how can he make anything useful happen from one button/Mode for all five synths?" Good question. Part of the answer lies in the fact that the Selected Track will never contain more than one synth plug-in. That means we don't need individual Mode Changes in the CAs.

Continued over...

This is just a fraction of the CAs needed to control five different synths from one Mode.

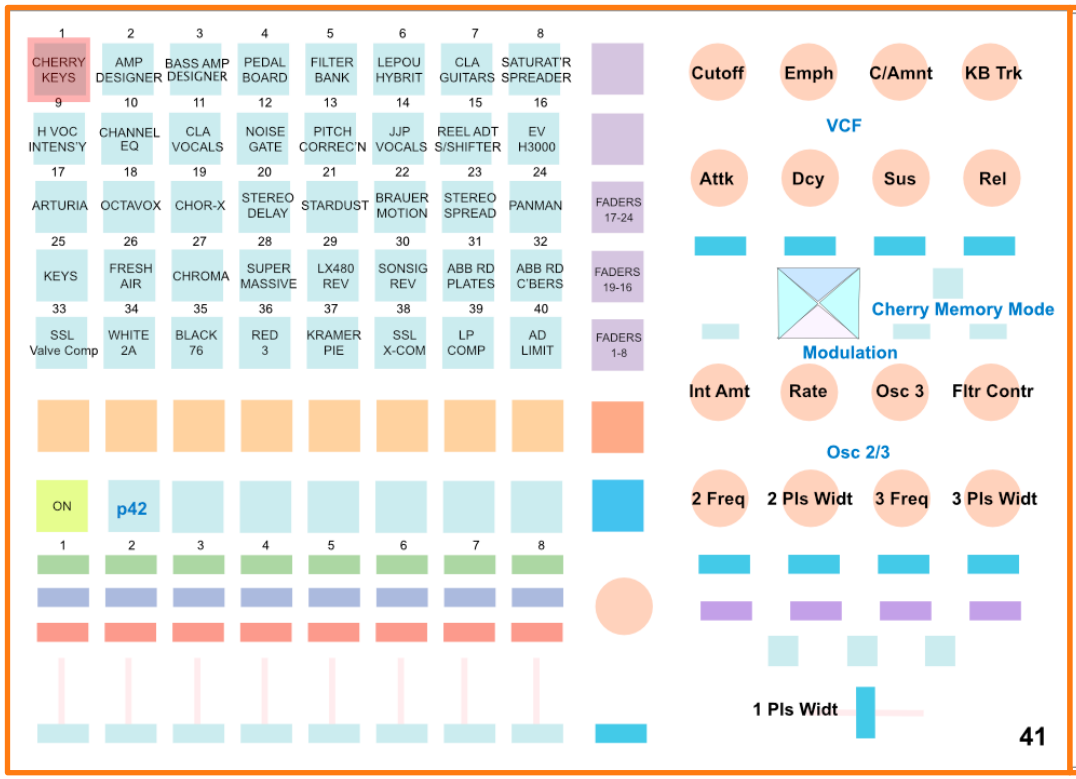


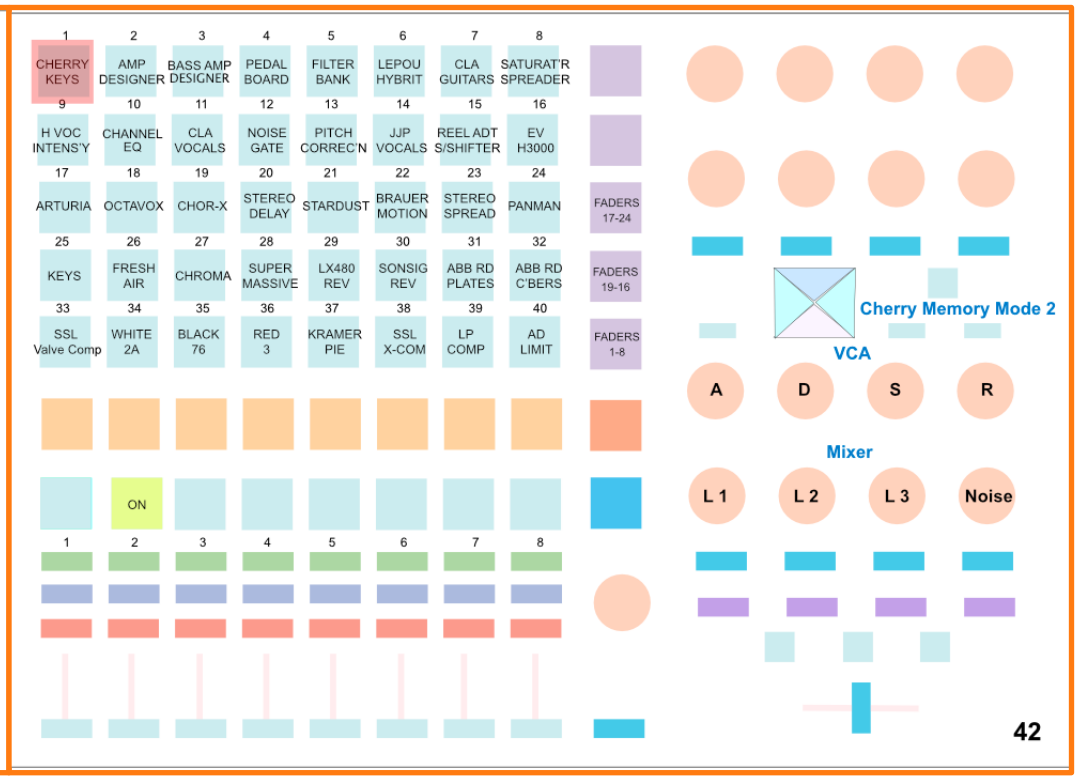
… From previous page

My two best tricks

Before we go any further, I should explain more about two strategies I adopted to get to maximise the number of controls at my disposal, plus the most synth plug-ins I could incorporate. Take a look at the starting with the one marked ON. These change the MIDI messages sent by the controls that on this page are assigned to VCO 2 and ENV 2. That means that instead of just eight CAs, these eight buttons give us 8 x 8, which equals 64. The four blue buttons below the rotaries also change MIDI message, so we've got 32 switches. Suddenly, we're in with a real chance of controlling an anlogue-style synth from hardware. That's quite a thing. On the

Continued over...





… From previous page

left, you'll see my CA window for the Cherry synths. The scheme isn't complete yet, but you can see the complexity in putting the CAs for more than one plug-in into the same Mode.

Design and planning

If ever there was a build where thinking your design through, and carefully documenting is vitally important, it's this one. The good news is thinking is free. No matter how much of it you do, you won't get a bill. Ever. That's good news, because the more you think your scheme through, the more successful it's likely to be in making your musical instruments more fun – and more expressive – to play.

I'm hoping you already have a graphic representation of your control surface(s), like the ones above. If you really can't make one, I suggest downloading a PDF of the owner's manual, because there's usually a graphic in there. Here are a few things I think you should consider in your design.

What are the controls you most want assignments for? It's best to prioritise, because writing CAs takes time, so at least you'll get the ones you need most before you run out of puff!

Should you try to mirror the layout of the synth? Normally with a plug-in, I do, but that isn't so realistic if you've only chosen to create assignments for a few key controls that are not together. A cluster of physical controls is more ergonomic than ones dotted all over the place, just because that's how they were on a synthesiser that was made in 1975.

How does this fit into your total scheme? In my case, I'd assigned all 40 buttons to effects processors before it dawned on me that I could free some up some of them by placing plug-ins that had few controls into the same Mode as another plug-in.

Continued over...

… From previous page

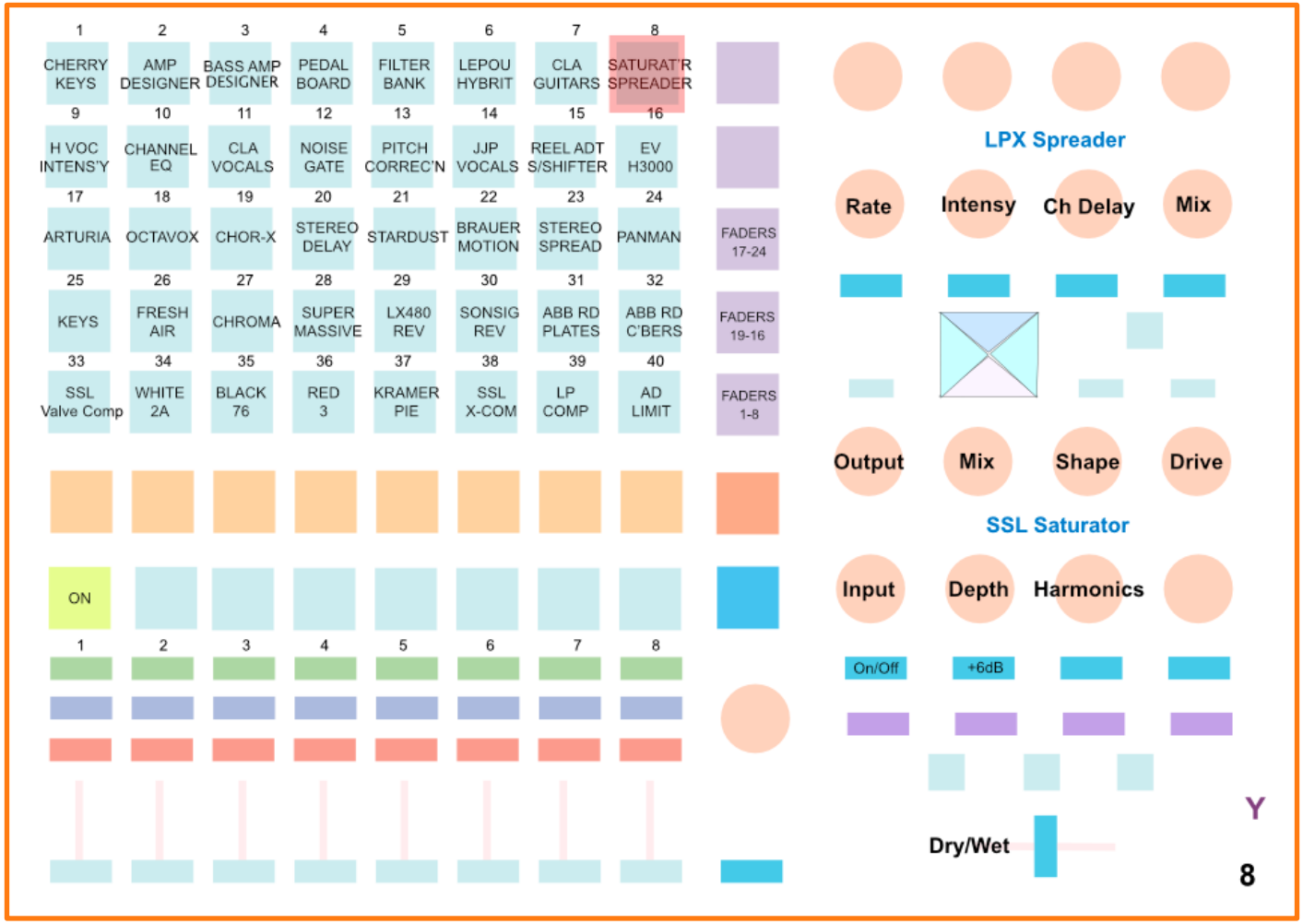
For instance, by putting the LPX Spreader plug-in on the top rotaries and SSL Saturator on the bottom eight, I was able to free up button Number 1, and use it for the Cherry Audio synths. This cause only minimal disruption to my existing scheme (see right).

What strategies are you going to use to get the best from your controller? For instance, if your controller has a lot of buttons but only one rotary, you're going to need a lot of Mode Change assignments before that rotary can address even a fraction of the knobs on most classic synths. At the other extreme, let's imagine you have a controller so extensive that you can assign a knob and switch to every function the synth shows on screen, without needing Mode Changes within your CAs, or Scene Changes on your controller that make the same controls send out new MIDI messages. This brings me to another issue…

*Just because you **can**, **should** you?* Just because you *could* assign every function to a physical control, it doesn't mean it's the right option for you.

Sometimes making the control surface, and your design, more complicated can actually make it less useful to you. This is especially true in performance situations, where it's better to have a few controls that you know for definite do what you want, than a vast number, the purpose of which you're struggling to recall.

In the end, these are your choices, but it's as well to think them through before you start your build, because it's going to get complicated! □



One-control-per-function is a more realistic option on some synths than others. On Cherry Audio's version of the Roland Jupiter-4, you could make it work, but you'll either need a lot of physical faders, or to assign those functions to rotaries.

Arturia V Collection the easy way

What you'll learn in this chapter:

You'll learn that, if you want real-time control over key parameters on 19 synths, six pianos and three organs in about 10 minutes, Arturia's Analog Lab is your friend!

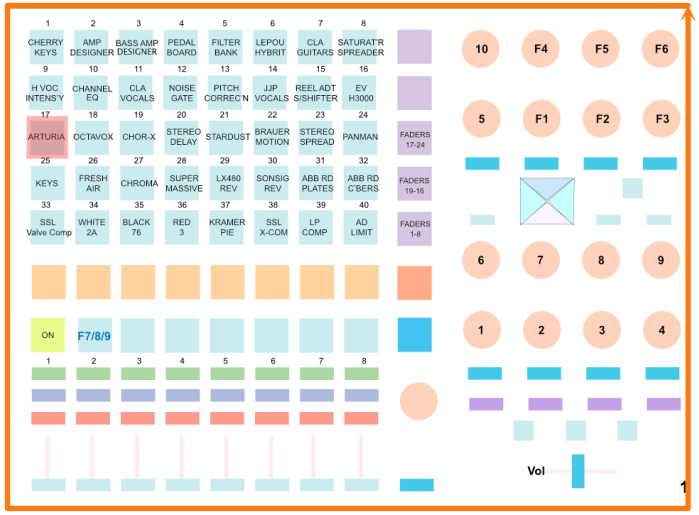
For those of you unfamiliar with Arturia's V Collection, it is a whole suite of emulations of old analogue synths.

While I think they're very good, writing assignments for each one individually would be a mammoth task, even if you were very selective about *which* controls you wrote CAs for. Luckily, Arturia has solved the problem for us by bringing all the individual keys together in a program known as the Analog Lab.

Here, you can browse through thousands of presets, many of which combine two synths in ways that would be hard work otherwise.

As you can see, Analog Lab also offers a reduced set of controls that work across all

Continued over...



My existing layout is quite usable, but with hindsight, I could use one of the mauve assignment buttons to free up the faders, so I can get a scheme better matched to Arturia's own screen

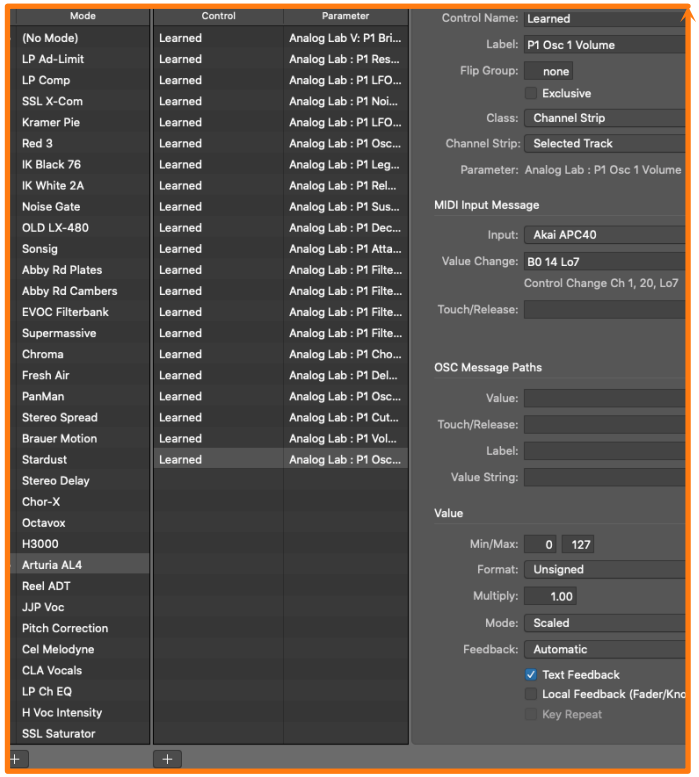
… From previous page

the instruments in the suite. That means one set of CAs will give you a useful degree of control over all the instruments. As a result, it is a really quick job.

If you've this far through the book, you shouldn't need me to tell you how the write the assignments. If you *do* need that help please go back to the start of the book to pick up the skills you need.

Design considerations

In general, my approach to this was to mirror the layout on-screen as far as possible. I'm starting to think I could have done this better,



You won't need to write a lot of assignments to get control of all the instruments in the suite

so time spent planning is definitely not time wasted.

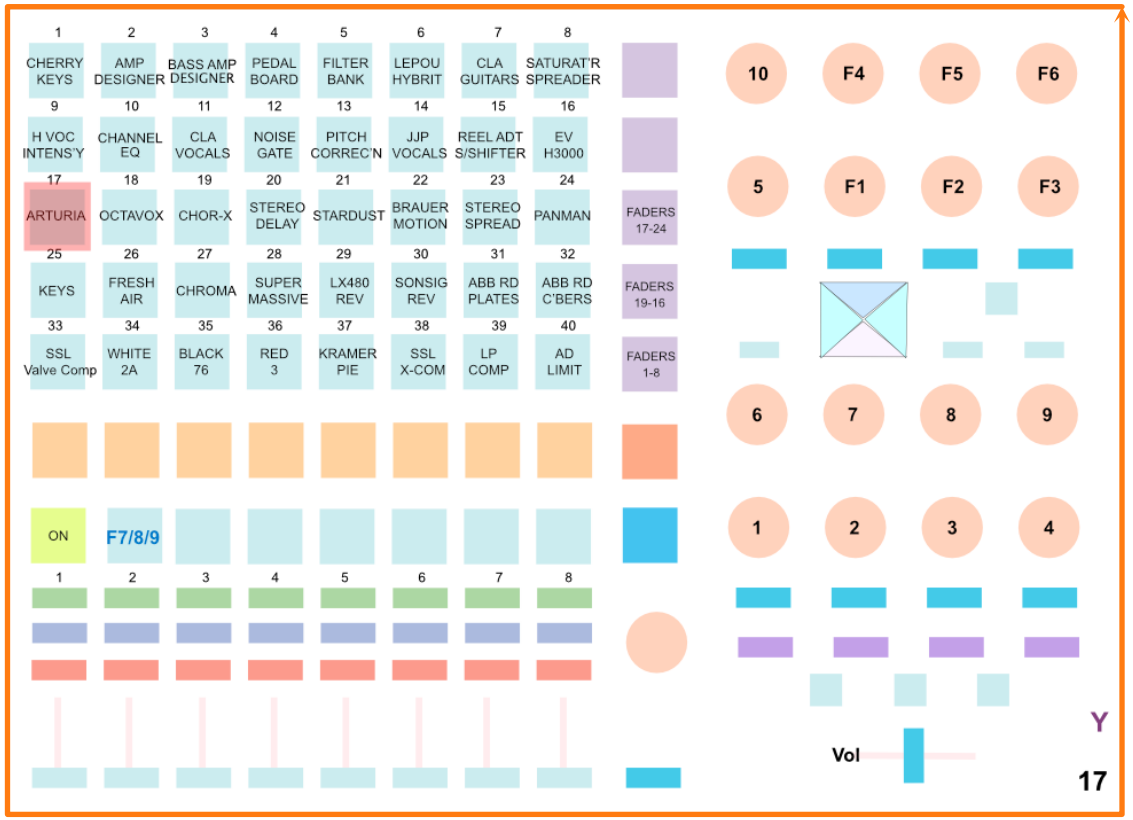
How you go about your build depends on your controller and how you like to work, but the fact there are nine faders on-screen, and nine faders on my main controller is a strong suggestion to me that I should be using the nine faders I've assigned to Channel and Master volumes.

It's not a difficult fix, but none of us buy into Logic Pro because we like writing Controller Assignments. We want to make music. □

If you'd prefer to write individual CAs for some of the synths in the suite, you can still do that, but some of them look quite challenging…

NOTE – I've only just got hold of Analog Lab V a few days ago. Although AL 4 was an easy nut to crack, writing them for AL V seems to be more difficult. I'll keep you posted.

A brief wrap-up until we meet again



I hope you find this first book on Controller Assignments useful. I'm already working on a second volume that will cover subjects such as networking audio and MIDI as part of a strategy to build a dynamic visual display system that will show what all the controls are doing on the physical device at any one time.

Although I have a prototype of that working, there are some limitations, which I'd like to iron out before unleashing it on the world!

If you have questions about your own build, the, er… logical place to post them is here: https://www.logicprohelp.com/forums/

You're sure to get informed answers, but they may not always be from me. I'm 66-years old now, (I kept quiet about that, right?) and if I step away from the world of writing Controller Assignments – for instance because I'd rather spend time actually *using* what I've built – I soon forget how the heck I did things. That means I have to read my own book!

In the meantime…

If you need a visual aid, so you know what all the assignments do for each plug-in, my best suggestion is to turn the visual template, which I hope you've been making *before* you start assigning, into a booklet.

I know it's not ideal, but without some sort of visual aid, the danger is that you'll simply stop using your controller, meaning all that time and effort is wasted.

That would be a shame, because making music is so much more fun when you can control things instantly. □