

Getting Started with pluggo



Cycling '74

1186 Folsom
San Francisco, CA 94103
tel (415) 621-5743 fax (415) 621-6563
info@cycling74.com www.cycling74.com



Copyright and trademark notices

pluggo, Plug-in Manager, and this manual are © 2000 Cycling '74. **pluggo** uses MAXplay, the runtime environment for Max, published by Opcode Systems, Inc. Max and MAXplay are copyright © 1990-99 Opcode Systems, Inc. and IRCAM, l'Institut de Recherche et Coordination Acoustique/Musique. **pluggo** also uses the runtime environment for MSP. MSP is copyright © 1997-2000 Cycling '74—All rights reserved. Portions of MSP are based on Pd by Miller Puckette, © 1997-2000 The Regents of the University of California. MSP and Pd are based on ideas in FTS, an advanced DSP platform © IRCAM.

pluggo, Plug-in Manager, and MSP are trademarks of Cycling '74. Max, Vision, Vision DSP, and Studio Vision are trademarks of Opcode Systems, Inc. VST and Cubase are trademarks of Steinberg Soft- und Hardware GmbH.

Contents

Introduction	5
About This Manual	5
Basic Features of Pluggo	5
System Requirements	6
Where to Get Help and Support	6
A Little Background Information on Pluggo	6
Installing Pluggo	7
What is Going to be Installed?	7
The Regular Pluggo Installer	7
The Installer for Max/MSP	9
Authorize Pluggo	13
Entering Your Code	14
Inserting Plug-ins Within Your Sequencer	17
Is Pluggo Asking for a Plug-in?	18
Inserting Plug-ins—Mono or Stereo	19
Using VST Instruments	28
Using the Plug-in Interface	33
Changing Parameters	40
The Parameter Change Pop-up Menu	41
Using the View Menu	42
Elements of Interface Views	43
Interface Hints and Labels	46
The Level Meter	46
Plug-in Manager	47
Starting Plug-in Manager	48
Enabling and Disabling Plug-ins	49
When Are the Files Moved?	50
Getting Information About a Plug-in	51
Saving Plug-in Sets	51
Working with Multiple Plug-in Folders	51
Synchronizing Plug-ins	55
Opening the Sync Example Document	55
Exploring the Synchronization Modes	55
Using PluggoSync	62
Using Modulator Plug-ins	66
Using Other VST Plug-ins within Pluggo	68

Using Other VST Plug-ins within MAS	68
The Pluggo plug-in	68
Features of Hosted Plug-ins	69
Modulating the Parameters of a Hosted Plug-in	70
Getting On the PluggoBus	73
PluggoBus Receivers	73
PluggoBus Senders	74
Plug-in Automation	76
Automation in Cubase	76
Automation in Logic Audio	78
Index	81

Introduction

About This Manual

Welcome to **pluggo**, the never-ending plug-in. This *Getting Started* manual covers installation of the program, operation of the interface features common to all **pluggo**-based plug-ins, information about using Plug-in Manager, synchronization, **pluggo** Modulators, opening other VST plug-ins within **pluggo**, PluggoBus and automation.

Do you need to read this entire manual? We definitely recommend reading through the description of the installation and authorization/registration process. The rest of this manual may cover some material with which you are already familiar. However, it's pretty short and discusses a few things that might not be obvious to the new user. For instance, did you know you can undo changes to parameters in **pluggo** plug-ins? Look in the *Plug-in User Interface* chapter for the details.

The documentation assumes that you are familiar with the basic operations of the sequencer program you'll be using to host **pluggo** plug-ins. We do review a few things related to inserting effects in Cubase, Vision, Digital Performer, and Logic Audio, but very little is said about creating or importing audio tracks or using audio I/O.

The *Pluggo Plug-in Reference Guide*, which describes each of the included plug-ins, is in electronic format. It's found in the Pluggo Stuff folder on the root level of the hard disk where you installed the software. In addition, all of the included plug-ins contain information on their use that you can access interactively. Plug-ins generally come with example preset effect programs (when it makes sense), as well as hints and other interactively accessible text describing parameters and basic operations.

This manual does not describe how to make your own plug-ins with Max and MSP. The *Pluggo Installer for Max/MSP Users* places plug-in development materials in the Pluggo Stuff folder.

Basic Features of Pluggo

- A lot of plug-ins (at least 74 of them)
- Opens any plug-in made with the plug-in development tools that are part of Max/MSP
- With VST 2.0 and MAS hosts, plug-ins support host synchronization. With other hosts, you can use the PluggoSync feature for synchronization.
- Plug-ins can send audio and/or control information to each other
- A Plug-in Manager that lets you keep track of enabled and disabled plug-ins
- Modulator plug-ins that control the parameters of other plug-ins
- Hosts any VST plug-in so you can change its parameters with a **pluggo** Modulator

System Requirements

pluggo requires:

- A PowerPC Macintosh or Mac OS clone. A 604 or G3 processor running at 150Mhz or faster is recommended.
- System 7.5.3 or later
- A sequencer application that hosts VST, VST 2.0, or MAS plug-ins.
- At least 64MB of memory, although the exact amount depends on the system version and the host sequencer you're using.
- At least 20MB of hard disk space

Where to Get Help and Support

If you encounter problems when using **pluggo**, the following resources are available:

- The Pluggo FAQ document, on-line at <http://www.cycling74.com/support/questions.html>
- e-mail to support@cycling74.com
- You may telephone our office in San Francisco, generally open from 10AM to 4PM Pacific time, but it's unlikely that anyone who handles technical support will actually be there. However, questions related to installation and authorization of the software can be answered.

Technical support is only available to customers who have purchased **pluggo**. If you did not purchase your copy of **pluggo** directly from Cycling '74, please send in the registration card in your package or register at www.cycling74.com/register.

A Little Background Information on Pluggo

pluggo is a runtime shell that uses the Max object-oriented graphical programming environment. **pluggo**'s signal processing capabilities are provided by the MSP audio extensions to Max, published by Cycling '74. Using Max and MSP in conjunction with **pluggo**, you can write your own audio and MIDI plug-ins.

The authors of Max and MSP are Miller Puckette and David Zicarelli. **pluggo**'s plug-ins and objects were written in Max/MSP by jhno, Adam Schabtach, David Zicarelli, Leslie Stuck, and Joshua Kit Clayton. Additional MSP objects for **pluggo** were written by Richard Dudas, who also drew the **pluggo** character. Plug-in Manager was written by Adam Schabtach. **pluggo** user interface design by Lilli Wessling and jhno. This manual was written by David Zicarelli and Gregory Taylor, with contributions from jhno and Adam Schabtach.

Installing Pluggo

Installing **pluggo** involves two steps. First, running the Pluggo Installer from the CD-ROM or your software download folder, and second, running the Authorize Pluggo application to authorize the software.

There are two versions of the Pluggo Installer. The *Regular Pluggo Installer* is what most people will use. The *Installer for Max/MSP Users* is for people who own both Max and MSP. It installs **pluggo** so that you can do plug-in development as well as use the plug-ins within your sequencer. If you don't own Max and MSP, use the Regular Pluggo Installer covered below.

What is Going to be Installed?

The Pluggo Installer will put files in the folder containing your sequencer application. **pluggo** has a couple of files that contain the support environment for its plug-ins that are placed in the same folder as the sequencer application. Additional support files (or aliases to these files) will be placed in a subfolder of the sequencer's application folder called Pluggo Support. Normally, sequencer effect windows show menus of plug-ins contained in the VstPlugIns folder. Since **pluggo** has so many plug-ins, you may not want to see all of them listed, so you'll get to choose whether you want to install them all. If you don't want to install them all, the installer creates a folder called VstPlugIns (disabled) containing plug-ins you can try out later.

If you're installing the MAS version of **pluggo**, a "stub" plug-in called Pluggo for MAS is installed in the Plug-ins folder inside the MOTU folder inside your Extensions folder. Unlike other MAS plug-ins you may have, however, the actual **pluggo** plug-ins themselves are not installed here. Instead, they are in a folder called VstPlugIns inside the application folder. If you'll be using **pluggo** with more than one MAS application, you'll need to install a copy for each application.

The Plug-in Manager application and additional documentation will be placed in a folder called Pluggo Stuff.

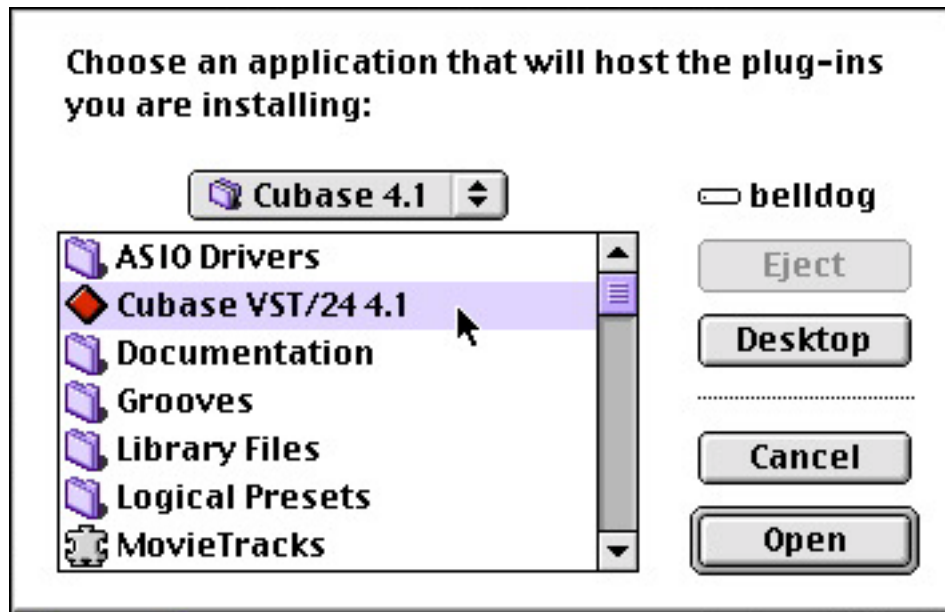
You can use Plug-in Manager to move plug-ins from the disabled plug-ins folder to the VstPlugIns folder. You can also make "sets" of enabled/disabled plug-ins. See the Plug-in Manager chapter of this manual for more information.

The Regular Pluggo Installer

When you run the *Regular Pluggo Installer*, you'll first see a license agreement covering the software and its documentation. Please read the agreement carefully and click Agree if you are willing to abide by its terms.

The screen appearing after the license agreement lets you choose between Easy Install and Custom Install. In Custom Install, you can selectively exclude certain items from being installed. We recommend Easy Install unless you know what you are doing and want to re-install a specific file.

If you clicked Easy Install, you'll be presented with the dialog box shown below and asked to select the application that will host the plug-ins. For example, if you'll be using **pluggo** with Cubase VST 4.1, open your Cubase folder and select this application file. The plug-ins and support files will be installed in the folder containing the application you choose.

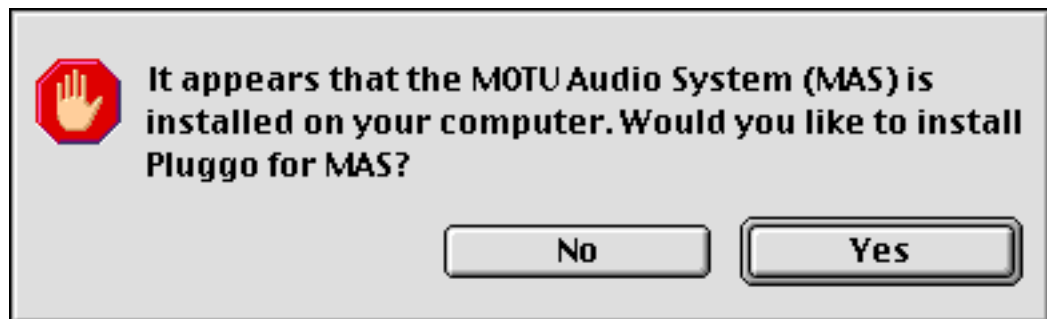


- Choose one of the folders shown and click OK. If you click Cancel, the installation will be stopped.

After determining a location for installation, the installer will copy the necessary files to your sequencer's application and plug-ins folder.

The **pluggo** Installer has no way of determining whether you chose a valid application to host the plug-ins. You'll find this out soon enough when you discover you can't use audio plug-ins with Microsoft Office.

If you have the MOTU Audio System installed, you'll see a dialog box asking whether you'd like to install **pluggo** for MAS.



Pluggo for MAS is a MAS plug-in file that allows **pluggo** to work with MAS-compatible host applications.

- Choose Yes if you want to install this file. Choose No if you are not using Digital Performer or another MAS-compatible host application for your plug-ins and do not want to install this file.

Note that if you want to use **pluggo** with your MAS-compatible application and you don't see this dialog, **pluggo** won't work. You should verify that you have a proper MAS installation.

You'll then be asked to choose one of the following Easy Install options:

- **Install a Few Plug-ins.** In this option, we select ten (10) of our favorite plug-ins to add to your sequencer's VstPlugIns folder and put the rest in the disabled plug-ins folder. You can then use Plug-in Manager to enable additional plug-ins, or use the *Pluggo* VST plug-in to open any plug-in file you want to use within the sequencer.
- **Install All Plug-ins.** In this option, the installer puts all the plug-ins that come with **pluggo** into the sequencer's VstPlugIns folder. You can use Plug-in Manager to disable some of these plug-ins if you find that having so many plug-in names appear in a menu is bothersome.

After you choose one of these options, the plug-ins will be installed. The installer also installs a folder called Pluggo Stuff in the disk or folder you selected for installation. Pluggo Stuff contains:

- the Authorize Pluggo application
- the *Pluggo Plug-in Reference Guide*
- the Plug-in Manager application
- possibly other random promotional material

Two of the files that are installed are placed in the same folder as your sequencer application. One is called *MaxPlugLib* and the other is called *Max Audio Library for Plugins*. If you are installing **pluggo** to be used with several host applications and want to save a small amount of disk space, you can move the *MaxPlugLib* and *Max Audio Library for Plugins* libraries to the Extensions folder and remove them from the folders of each host application.

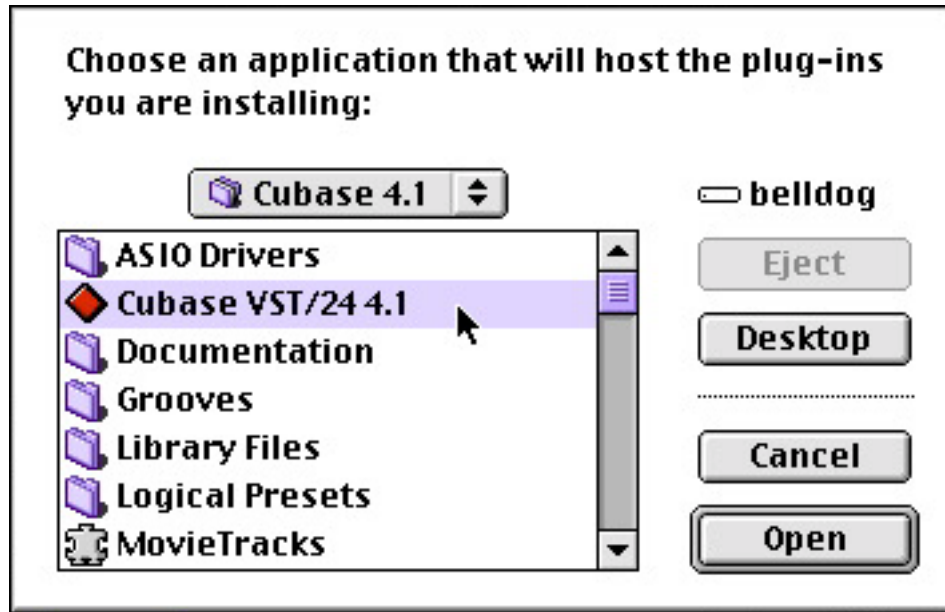
The Installer for Max/MSP

If you've installed **pluggo** with the *Regular Pluggo Installer*, you can skip this section and continue reading about the Authorize Pluggo application.

When you run the *Installer for Max/MSP Users*, you'll first see a license agreement covering the software and its documentation. Please read the agreement carefully and click Agree if you are willing to abide by its terms.

The next screen lets you choose between Easy Install and Custom Install. In Custom Install, you can selectively exclude certain items from being installed. We recommend Easy Install unless you know what you are doing and want to re-install a specific file.

If you clicked Easy Install, you'll be presented with the dialog box shown below and asked to select the application that will host the plug-ins. For example, if you'll be using Pluggo with Cubase VST 4.1, open your Cubase folder and select this application file. The plug-ins and support files will be installed in the folder containing the application you choose.

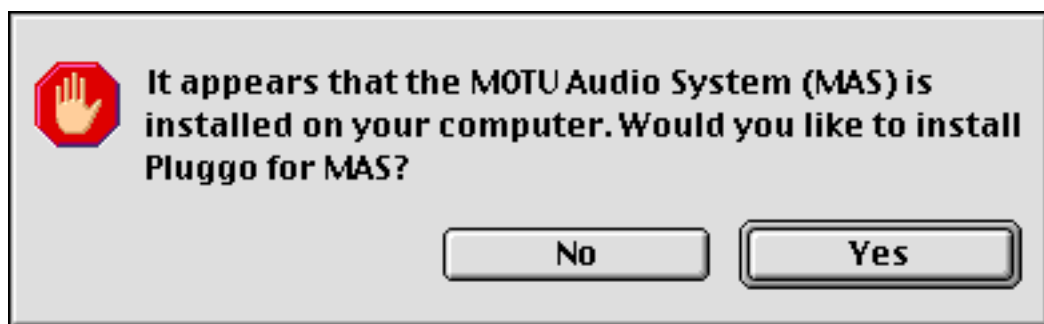


- Choose one of the folders shown and click OK. If you click Cancel, the installation will be stopped.

After determining a location for installation, the installer will copy the necessary files to your sequencer's application and plug-ins folder.

The **pluggo** Installer has no way of determining whether you chose a valid application to host the plug-ins. You'll find this out soon enough when you discover you can't use audio plug-ins with Microsoft Office.

If you have the MOTU Audio System installed, you'll see a dialog box asking whether you'd like to install **pluggo** for MAS.



Pluggo for MAS is a MAS plug-in file that allows **pluggo** to work with MAS-compatible host applications.

- Choose Yes if you want to install this file. Choose No if you are not using an MAS-compatible host application for your plug-ins and do not want to install this file.

Note that if you want to use **pluggo** with your MAS-compatible application and you don't see this dialog, **pluggo** won't work. You should verify that you have a proper MAS installation.

You'll then be asked to choose one of the following Easy Install options:

- Install a Few Plug-ins. In this option, we select ten (10) of our favorite plug-ins to add to your sequencer's VstPlugIns folder and put the rest in the disabled plug-ins folder. You can then use Plug-in Manager to enable additional plug-ins, or use the *Pluggo* VST plug-in to open any plug-in file you want to use within the sequencer.
- Install All Plug-ins. In this option, the installer puts all the plug-ins that come with **pluggo** into the sequencer's VstPlugIns folder. You can use Plug-in Manager to disable some of these plug-ins if you find that having so many plug-in names appear in a menu is bothersome.

After you choose one of these options, the plug-ins will be installed. The installer also installs a folder called Pluggo Stuff in the disk or folder you selected for installation. Pluggo Stuff contains:

The installer will update various external object files it finds in the Max/MSP folder. **pluggo** requires special versions of certain Max and MSP external objects. However, these objects will not behave differently for non-Pluggo uses. Help files for **pluggo** - specific objects such as **plugin~** and **pp** will also be installed.

The installer also installs a folder called Pluggo Stuff in the disk or folder you selected for installation. Pluggo Stuff contains:

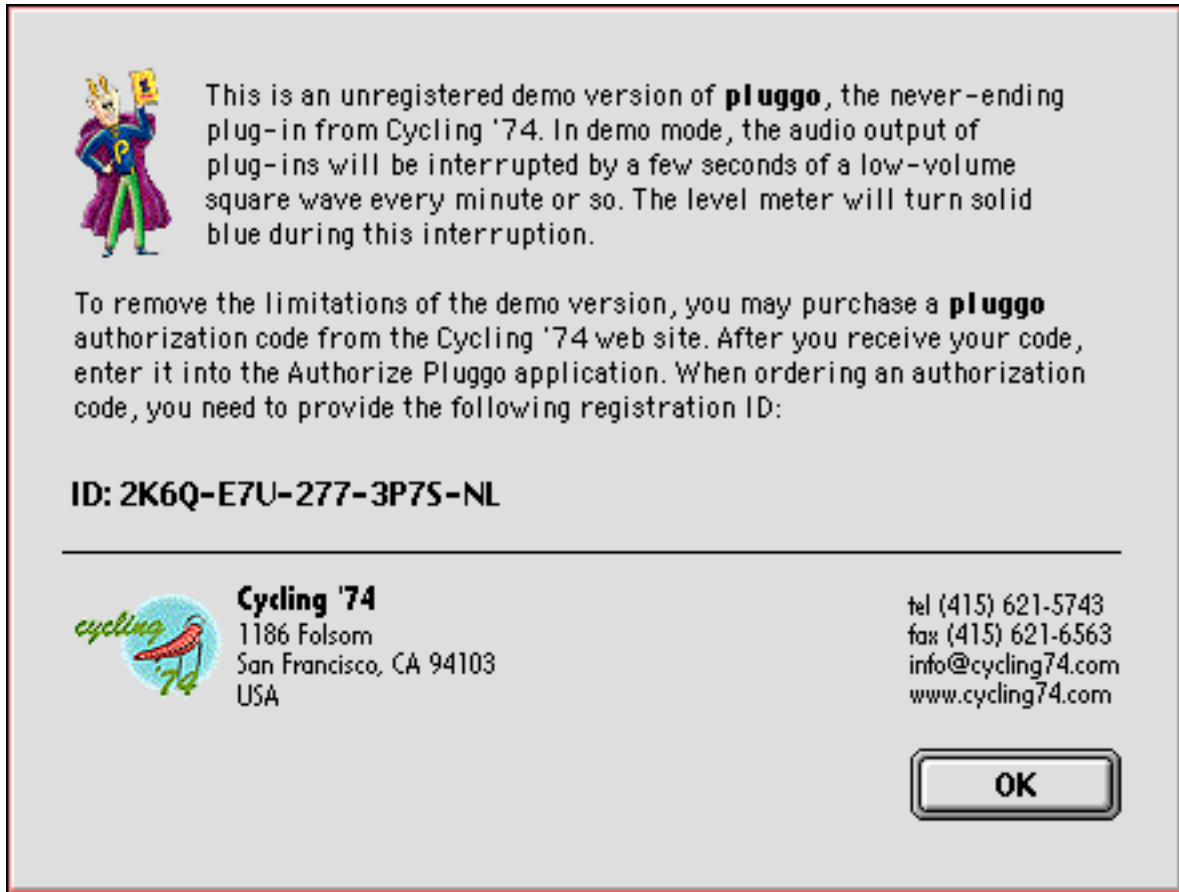
- the Authorize Pluggo application
- the *Pluggo Plug-in Reference Guide*
- the Plug-in Manager application
- information about developing your own plug-ins in Max/MSP
- possibly other random promotional material

Finally, the installer places an alias to your Max folder inside the VstPlugIns folder. This allows **pluggo** to find the external objects it needs to load the plug-ins.

Two of the files that are installed placed in the same folder as your sequencer application. One is called *MaxPlugLib* and the other is called *Max Audio Library for Plugins*. If you are installing **pluggo** to be used with several host applications and want to save a small amount of disk space, you can move the *MaxPlugLib* and *Max Audio Library for Plugins* libraries to the Extensions folder and remove them from the folders of each host application.

Authorize Pluggo

When you install **pluggo**, it's in demo mode. You'll see the following dialog box when you open the first **pluggo** plug-in within your sequencer:



In demo mode, the audio output of each plug-in is interrupted for a moment every minute or so.

In order to gain unlimited use of **pluggo**, you need to run the Authorize Pluggo application and enter a valid authorization code.

If you purchased the packaged version of **pluggo**, your authorization code and Registration ID can be found on a yellow sheet of paper included in the box. Skip to the Entering Your Code section below.

Please keep your authorization code, serial number, and Registration ID in a safe place.

If you downloaded **pluggo**, you can purchase an authorization code from Cycling '74. The most convenient way to do so is by using a credit card on our secure on-line order form. We will send you the authorization code after receiving your order. Here's the procedure:

- Run Authorize Pluggo, found in your Pluggo Stuff folder in the root directory of the drive where you installed **pluggo**. The opening screen will contain a message with a Registration ID at the bottom.
- Write down the Registration ID or click the Copy ID to Clipboard button so you can paste it into a form with your web browser later.
- Alternatively, if you're ready to jump right into **pluggo** itself, launch your sequencer and insert a **pluggo**-based plug-in. You'll see the dialog shown above with the Registration ID. Write down the ID and quit the sequencer.

If you prefer to purchase a packaged version of **pluggo**, visit your local retailer or use the order form on the Cycling '74 web site to buy one directly from us. No Registration ID is needed to purchase the packaged version—the ID is found inside the box, and you'll need to reinstall **pluggo** from the CD in the packaged version in order to authorize the software.

- In Authorize Pluggo, click the Go to Web Page button to launch your web browser or click Quit and visit <http://www.cycling74.com/regpluggo>. Fill out the information on the order form, and paste or type your Registration ID into the space provided.
- Once we process your order, you'll receive e-mail with an authorization code. Copy the authorization code text in the e-mail message to the clipboard so you can paste it into Authorize Pluggo.

Entering Your Code

- Run Authorize Pluggo. Click Continue at the first screen. Now you should see the following dialog for entering your authorization code:



If you see an error message instead, you may not be able to authorize the software. If you have problems, please contact Cycling '74 support.

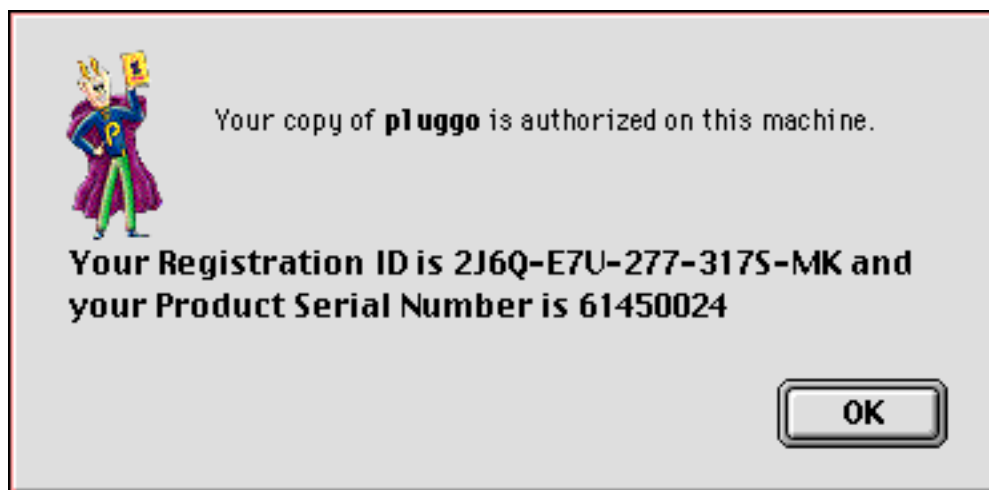
You can authorize **pluggo** on a removable drive or an external hard disk by selecting it from the pop-up menu above the space for authorization code. As long as you take your disk with you and connect it to a computer that will be running the software, you'll be able to use it, even if your copy of **pluggo** isn't on the authorized disk itself.

It's also possible that the Authorize button will not become enabled when you enter your authorization code. This is a bit of a frustrating interface since it's impossible to know exactly why Authorize Pluggo doesn't like the authorization code you entered. Here are some of the reasons we've discovered in the past:

- You entered the authorization code text incorrectly
- You formatted your hard disk with non-standard formatting software. For instance, RAID disk arrays are not supported. There are also potential incompatibilities with certain SCSI cards.
- You have a removable drive such as a Zip or Jaz drive connected to your computer. This is usually OK, but there appears to be a problem if a Zip or Jaz disk is inserted when you are trying to authorize the software on a drive other than your Zip or Jaz drive.
- There is something amiss with your hard disk. You may have to reformat. This is an extremely rare problem, but it has happened. Before reformatting, contact us and we might be able to provide alternatives.
- Some authorization codes may simply be invalid. Contact us and we'll test out the code and if necessary, send you a new one.

Let's assume you don't have any of these problems.

- Click Authorize after entering your authorization code. You should then see the following dialog box:



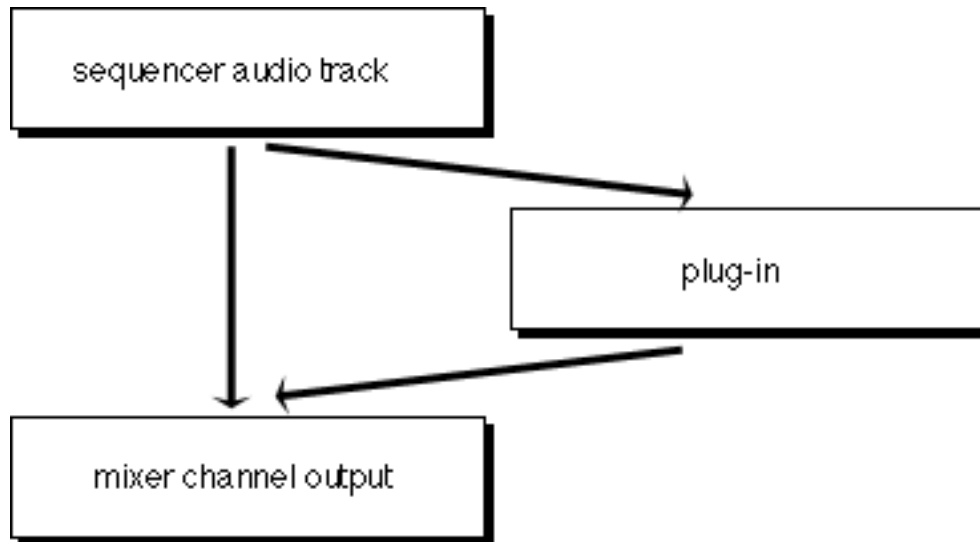
The **pluggo** authorization resides in a secret place on your hard disk, not in a particular **pluggo** file. If you want to use **pluggo** plug-ins with another sequencer on the same machine, you won't need to authorize again.

If you run Authorize Pluggo again and you've already registered, you'll see the above dialog again after the introductory screen. You won't have an opportunity to enter a different authorization code unless you hold down the shift and option keys after clicking OK at the introductory dialog.

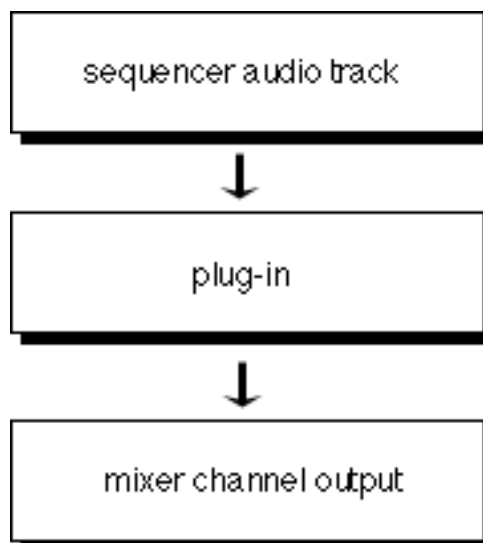
Inserting Plug-ins Within Your Sequencer

In this chapter, we'll cover the steps for inserting **pluggo**-based plug-ins with Cubase VST, Digital Performer, Vision DSP, Studio Vision Pro, and Logic Audio.

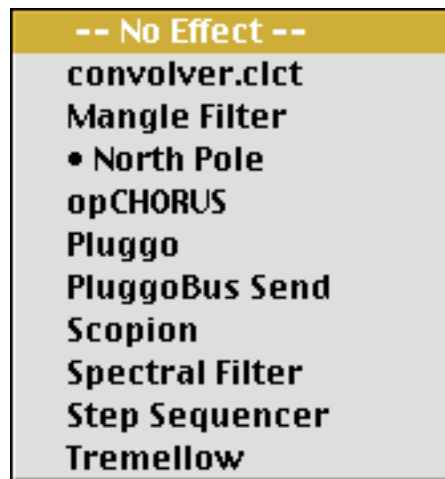
When we say “inserting” a plug-in, we mean placing it into the audio processing path of a mixer. There are two kinds of insertion. In one kind, used in the Channel Mixer in Cubase VST, the plug-in *adds* its signal to the input as a kind of bus. Cubase VST 4.x refers to these as Send Effects.



In another kind of insertion, used in Vision, Logic Audio, Digital Performer, and in the Master and Insert Effects of Cubase VST, the plug-in *replaces* its input with its output.



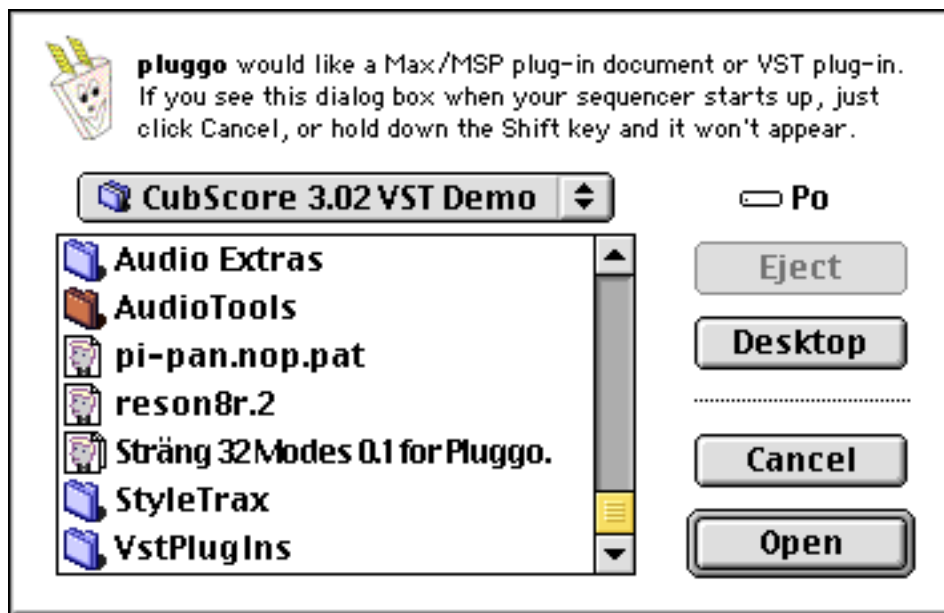
After installing **pluggo** you will find an assortment of plug-ins in your sequencer's VstPlugIns folder. All of these files will appear in a pop-up menu of plug-ins you can insert. For instance, here is the menu for Master Effects in Cubase VST 4.0.



Is Pluggo Asking for a Plug-in?

Note: This section does not apply to MAS host applications because the *Pluggo* plug-in is not used in MAS. MAS users may skip to the *Inserting Plug-ins: Mono or Stereo?* below.

One of the plug-ins in the pop-up menu is special, and it's worth mentioning because the first time you run your sequencer (or possibly every time you run it), you may see its initial dialog box shown below.



Pluggo is the name of a VST plug-in that loads other plug-ins by letting you choose files from a standard open file dialog. In that sense it's a bit like a joker in a deck of cards.

The reason you may see this dialog box when your sequencer starts up is that many sequencers load all of their VST plug-ins at startup to determine whether they are mono or stereo. Most are intelligent and store the information about plug-ins somewhere in a file on your hard disk, so after they open them on startup once, they don't have to do it again. But some programs may not store this information consistently, so you may see this dialog each time you launch the sequencer. To prevent this from happening, you can either hold down the Shift or Caps Lock key while the sequencer is launching, or take the *Pluggo* plug-in out of your VstPlugIns folder.

In any case, the correct choice, as stated in the directions for the dialog box, is to click Cancel to make the box go away.

You may also see the *Pluggo* plug-in's open file dialog box in other situations:

- You've created a startup document for your sequencer (what Cubase calls an Autoload Song) that contains the *Pluggo* plug-in. In this case, it's up to you whether you want to open a plug-in file or not.
- You're opening a document that contains a plug-in you opened with the *Pluggo* plug-in, but the plug-in can't be found. There are several reasons why this might be the case.
- Your sequencer (e.g., Cubase 3.0) may not support a feature recently added to the VST specification that **pluggo** uses to re-load plug-ins you opened. The solution for these sequencers is to place the plug-in into the VstPlugIns folder rather than opening it via the *Pluggo* plug-in.
- The plug-in file may have been moved or deleted. **pluggo** can often find plug-ins if you've moved or renamed them, but not always.
- There may be a problem with the *Pluggo Dictionary* file in the Preferences Folder in the System Folder. This file stores the locations of all the files opened by the *Pluggo* plug-in. It could be missing, incomplete, or damaged. Or, you may have created the sequencer document on one computer and opened it on another, where the *Pluggo Dictionary* file contains conflicting information about file locations.

In any of these cases, you may need to locate the plug-in again. If you used several plug-in files you originally opened with the *Pluggo* plug-in, this might be a little confusing, since you don't have any idea which request corresponds to which plug-in. However, after you go through the process once, the *Pluggo Dictionary* file will be updated, so your document should read in without any problems from then on.

Inserting Plug-ins—Mono or Stereo

pluggo-based plug-ins can be opened in either a mono or stereo context. However, a mono plug-in may not do what you want if you try to process a stereo signal—you may just hear processing on the left channel.

It's also possible in some sequencers to insert plug-ins into a context where they accept a mono input signal and produce a stereo output signal.

You can refer to the chart in the *Pluggo Plug-in Reference Guide* that lists information for each of the included plug-ins about whether it can be used effectively in mono or stereo situations (or both).

Typically, to insert a stereo plug-in, the channel in the mixer you're using for the plug-in has to be a stereo channel.

To make a stereo insertion in Cubase:

- Choose Master Effects from the Audio menu, and choose a plug-in from the pop-up menu to the right of an effect. Switch on the power button. Master Effects in Cubase are always stereo.



To make a mono insertion in Cubase:

- Choose Effects from the Audio menu (in Cubase 4.0, choose Audio Send Effects from the Panels menu; in Cubase 4.1, choose VST Send Effects from the Panels menu), and choose the name of a plug-in from the pop-up menu to the right of an effect.
- Switch on the power button and turn up the gain on the Effects Master knob corresponding to the effect. This gain knob is the master input send to the effect.



- Choose Monitor from the Audio menu (in Cubase 4.0, choose Audio Channel Mixer from the Panels menu; in Cubase 4.1, choose VST Channel Mixer from the Panels menu). Pick a channel on which you would like to hear the effect and click on the little FX button above the channel fader.



- The effect send and EQ window for Channel 1 appears. You'll see the name of the effect, in this case, *Generic Effect*, with an on-off button and a Send volume knob.

Turn up the gain on the Send knob and click on the on-off button to turn on the effect send. If the blue Pre button is lit, the effects send occurs before the channel fader, otherwise the channel fader will attenuate the effects send level.



If you are using a **pluggo**-based plug-in that ignores its audio input such as a synthesizer or sampler, you don't need to turn up the effect sends. The plug-in will be able to produce audio as long as its power button is on.

You can also make an Insert Effect in Cubase 4.0 that goes between the input (the audio track being played) and the rest of the channel's mixer settings. Unlike the bus effects just described, the Insert effect replaces its input. This is the nature of all effects in the other applications we'll cover in this chapter. You might think that an Insert Effect before a stereo pair of channels in Cubase will allow you to use a stereo plug-in, but in fact, you'll only hear the left channel of the plug-in's output.

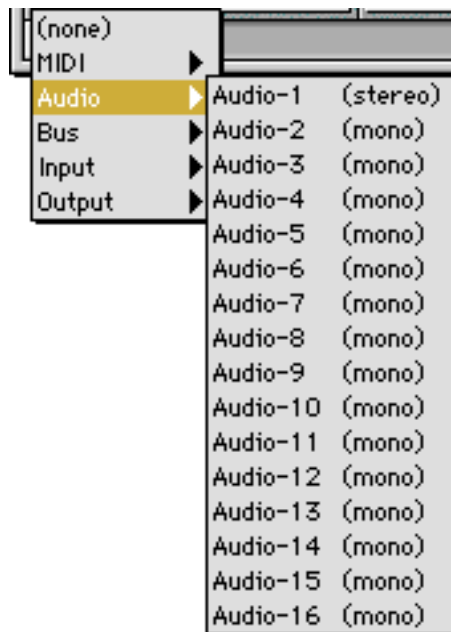
To make a stereo insertion in Vision:

- Open the Audio Instruments window by choosing Audio Instruments from the Windows menu. Make one of the existing instruments stereo by checking its box in the Stereo column.

Name	M	S	Stereo	Output	Sends	Plug-Ins	EQs
▼ Instruments							
Audio-1			✓	Snd Mgr	0	0	0
Audio-2				Snd Mgr	0	0	0

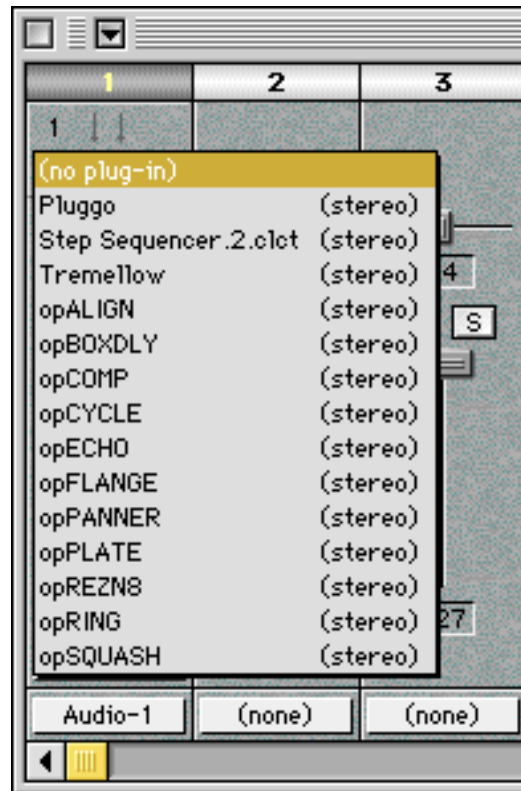
- Choose one of the four Consoles from the Windows menu (i.e., Console 1, or whatever you've named it).

- At the bottom of a fader, click on (none) and choose a stereo Audio Instrument from either the Audio submenu or the Input submenu. Audio selections play tracks in your sequence, while Inputs select live inputs to the mixer.



- After you do this, an output choice will appear above the input. It may be set automatically to something, or it may be labeled (no output). Click on (no output) to set an output for the fader. If you pick an Output Pair, the output will be stereo, otherwise it will be mono.
- Click on the downward-pointing triangle in the title bar of the Console window, and choose 1 Plug-in from the Plug-ins submenu.

- Click on (no plug-in) and choose the desired plug-in from the pop-up menu.



To make a mono insertion in Vision:

- Choose one of the four Consoles from the Windows menu (i.e., Console 1, or whatever you've named the Console).
- Click the bottom box below the fader, click on (none) and choose a mono Audio Instrument from the Audio submenu or the Input submenu. Audio selections play tracks in your sequence, while Inputs select live inputs to the mixer.
- After you do this, an output choice will appear above the input labeled (no output). Click on (no output) to set an output for the fader.
- Click on the downward-pointing triangle in the title bar of the Console window, and choose 1 Plug-in from the Plug-ins submenu.
- Click on (no plug-in) and choose the desired plug-in from the pop-up menu.

In Vision, a “channel” in a console represents a connection between an input and an output. Plug-ins are always inserted in the middle of this connection, so they always replace their input.

To make an insertion in Logic Audio:

- In Logic Audio, any mixer or bus channel can have a stereo output. You don't need to do anything special to set this up. However, only certain channels will have a stereo input. These are mixer channels associated with Audio objects that have stereo audio files playing back on them. Bus mixer channels always have mono inputs. There is no

way to tell which channels will have a stereo input and which will have a mono input except by context (and listening of course).

There are two ways to use plug-ins in Logic Audio. They can either be inserted directly into a mixer channel, or they can be inserted into a bus channel which gets its input from a mixer channel's send. Inserting plug-ins on a bus channel allows you to have multiple mixer channels going to the same plug-in(s), as well as making it possible to control the wet and dry levels separately in cases where a plug-in does not give you that option.

- For an insert effect, click once on one of the two boxes below Inserts in a mixer channel.



- All stereo plug-ins will be shown twice in the pop-up that appears. The plug-in's name followed by (m/s) indicates it will be given a mono input and a stereo output.

- Choose the desired plug-in and input version. After being loaded, its name will appear in green in one of the Insert slots.



- For a bus effect, simply insert a plug-in on a bus channel as you would on a mixer channel. If you can't find the bus channels you should make sure that your audio objects are correctly setup in your Environment – see the Logic Audio manual for details. Then return to the mixer channel and click and hold on one of the boxes under Sends. Then select the one of the 8 busses in which you've inserted an effect. When the bus is selected a knob will appear to its right indicating how much of that channel's level will be sent to the bus. Make sure the bus is assigned to currently active outputs or you won't hear any effect.

To make an insertion in Digital Performer:

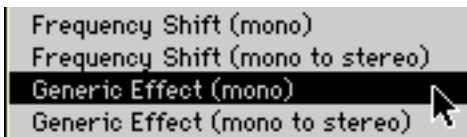
- Digital Performer lets you insert plug-ins in its Mixing Board window. In Digital Performer, any audio track can have a mono or stereo output. If you want to add an audio track with a stereo input to the Mixing Board, you will need to create a stereo audio channel.

To create a stereo audio channel in Digital Performer, the Sequence window must be active. Click on the Sequence window so that the title bar menus are visible. Choose Stereo Voice from the Add Audio Track submenu of the Sequence window's title bar menu. Although the new track will be added, you may need to add a Mixing Board track strip for it. Make the Mixing Board window active. On the left side of the mixing panel, you will see a track list. The new stereo voice you added will be grayed out. Click on the grayed out track icon, and a new track strip for your stereo voice will appear in the Mixing Board window.

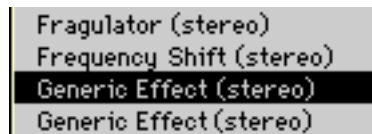
- Click on the arrow buttons for the effect inserts in an audio mixer channel. A pop-up menu listing the available plug-ins will appear.



- If you are inserting an effect on a mono audio channel, all the plug-ins will be shown twice in the pop-up menu that appears. The plug-in's name will be followed by its output type (either mono or mono to stereo) in parentheses. If you choose a mono to stereo effect, Digital Performer will automatically change the audio output to stereo (if you remove the effect, the track will revert to mono).



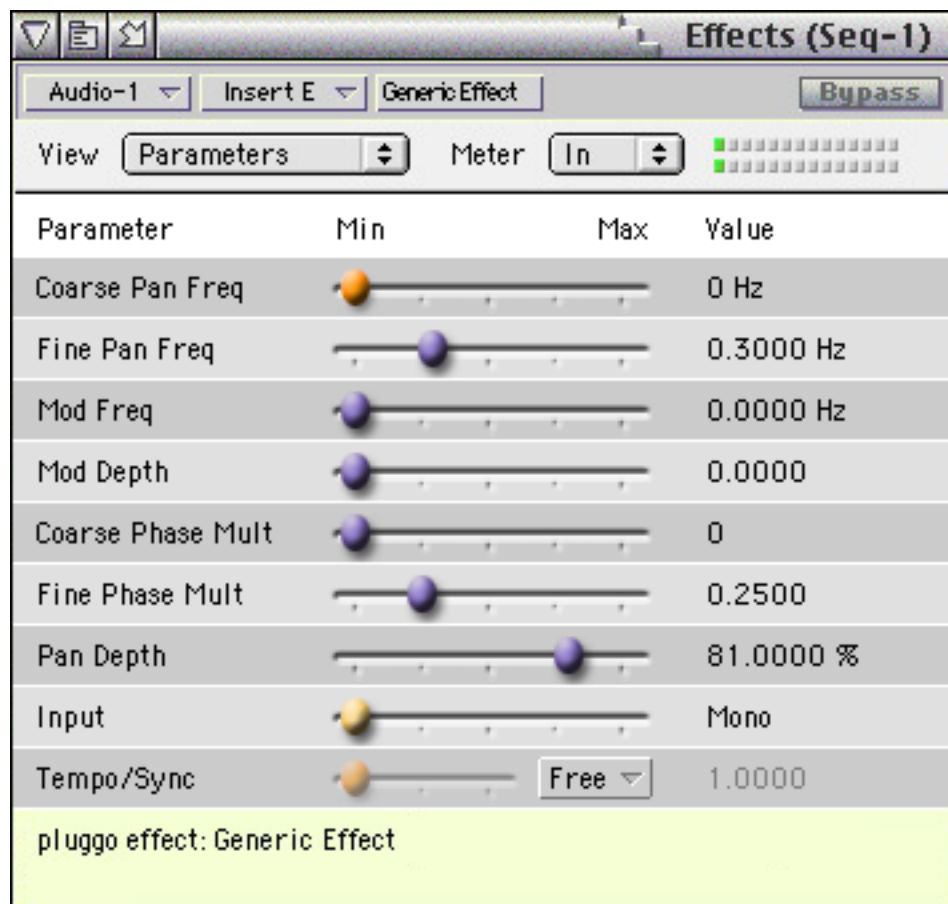
If you are inserting an effect on a stereo audio channel, you will only be shown the stereo to stereo plug-ins.



- Choose the desired plug-in and version (either mono or stereo, if applicable). After the plug-in loads, its name will appear in the window for the insert slot you have chosen.



The plug-in's edit window will also appear.



Using VST Instruments

The VST 2.0 plug-in format supports plug-ins that receive MIDI and generate only audio output. In Cubase 4.1, these plug-ins are called VST instruments. In Logic Audio 4.1, they are choices from a set of available Audio Instruments. Plug-ins of this type are not effects processors but synthesizers, samplers or drum machines.

Cubase 4.1 and Logic Audio 4.1 are the only host applications that at the time of the preparation of this manual support the use of VST Instruments. This chapter is only currently applicable to users of these applications, although other hosts may eventually support plug-ins with MIDI input and audio output. However, the host application that supports them may call them something other than VST Instruments.

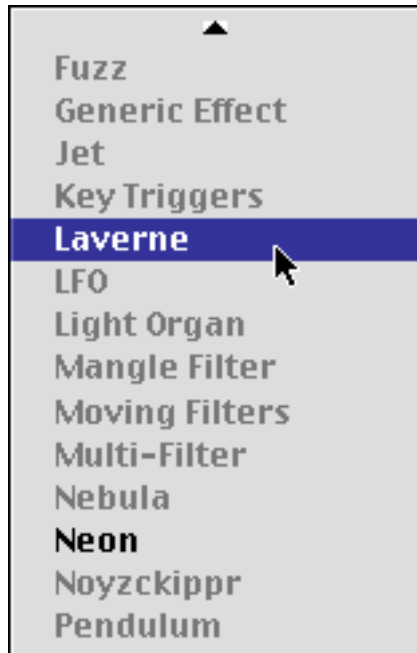
Earlier releases of **pluggo** included plug-ins that generated audio output such as *Synth*, but **pluggo** now allows plug-ins to receive MIDI and generate audio information, just like a “real” synthesizer. We’ve included a simple synthesizer plug-in called *Laverne* that you can try.

To use a VST Instrument in Cubase:

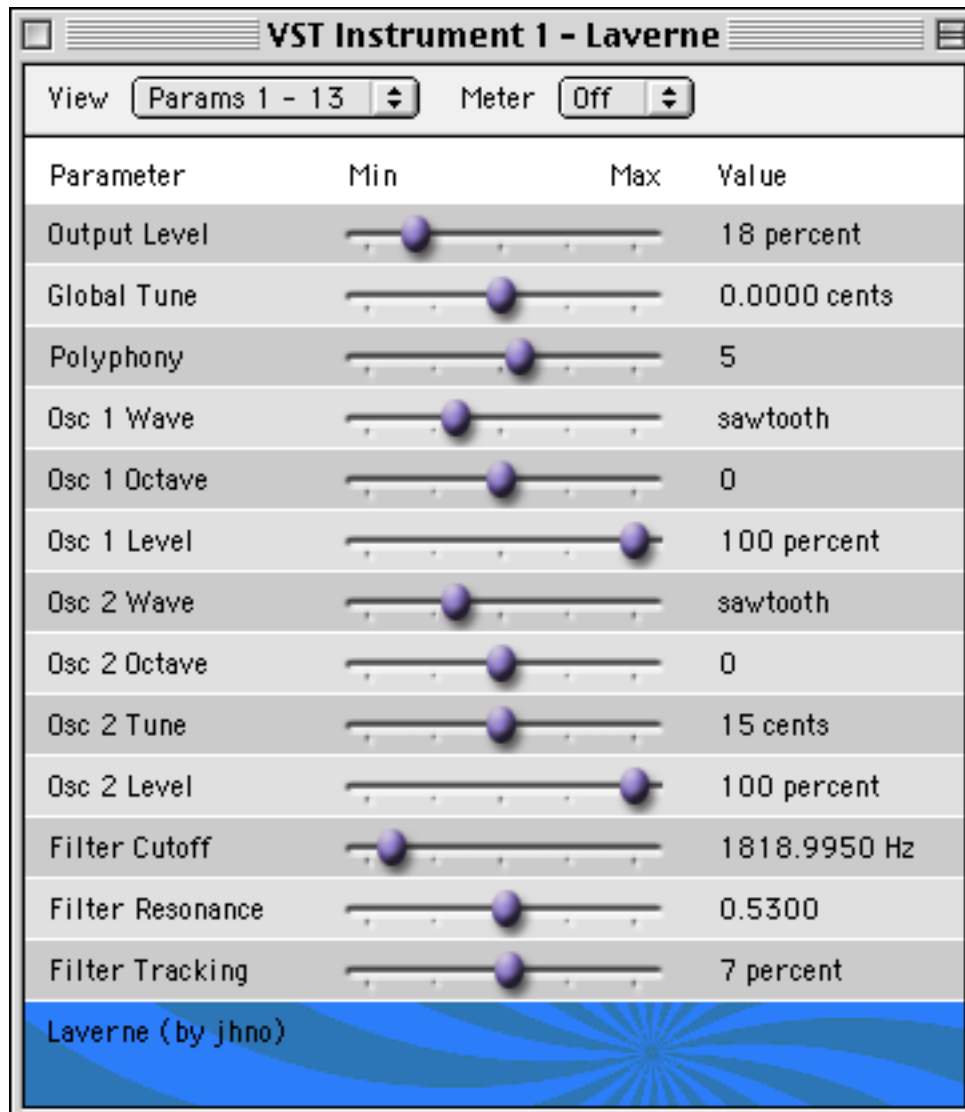
- Choose VST Instruments from the Panels menu. You will see the VST Instruments Panel appear.



- Click on No VST Instr. Choose the *Laverne* VST Instrument plug-in from the pop-up menu. Notice that while you'll see the names of the other plug-ins installed, everything that isn't a VST instrument plug-in will be grayed out.

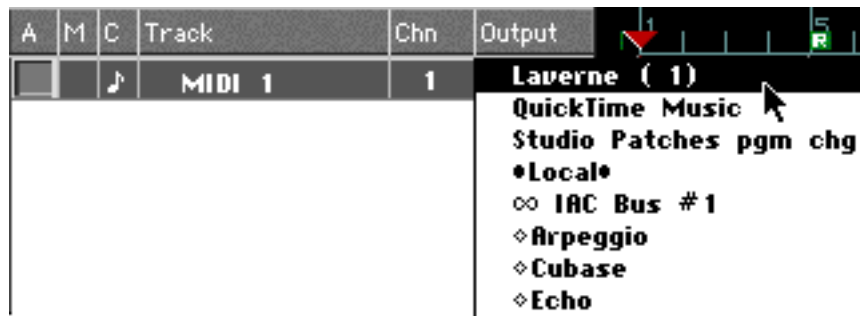


- After the plug-in loads, switch on its power button.
Click the plug-in's Edit button in the VST Instruments Panel. You will see the VST Instrument's control panel.



To route a MIDI Track to the VST Instrument in Cubase:

- If you have an existing MIDI track in your sequence, you can assign it to “play” *Laverne*. Click on the Output column for the desired track in the Arrange window and choose Laverne (1) from the pop-up menu.



MIDI output from the MIDI track will now be routed to the VST instrument you have selected.

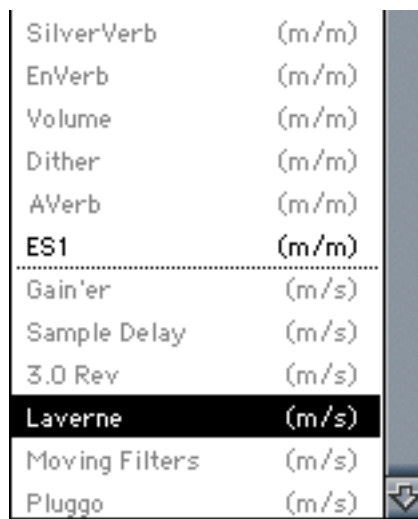
To use a VST Instrument in Logic Audio:

- In the Arrange window, there should be at least two tracks assigned to Audio Instruments. One is shown selected in the picture below.



- If not, click the title of any MIDI or audio track and choose an Audio Instrument (1 – 8) from the menu.
- Double-click the Audio Instrument track. You’ll be taken to the corresponding Audio Instrument object (channel strip) in the Environment (audio mixer) window.

- A blank rectangle appears at the top of the channel strip to set the source for the Instrument. Click on it and you'll get a pop-up menu with a lot of grayed-out items. Near the end of the list you'll see ES1, the Emagic synthesizer plug-in, plus any VST Instrument plug-ins, such as *Laverne*.



- Choose the desired plug-in. Its name appears at the top of the channel strip.



- You can now return to the arrange window and either record or draw notes into any tracks that use the Audio Instrument you've configured with the plug-in. The notes in the red (selected) track shown below will "play" *Laverne*.



Using the Plug-in Interface

In this chapter, we'll be using a mono plug-in called *Generic Effect* as our example. The Pluggo Installer added this plug-in to your VstPlugIns folder even if you chose the Install a Few Plug-ins option. We'll use *Generic Effect* to illustrate the features common to all, or at least most, **pluggo**-based plug-ins.

- Following the appropriate directions for your sequencer for inserting a mono plug-in, Choose *Generic Effect* from the pop-up menu of plug-ins.

All the plug-ins included with **pluggo** have their own edit windows; in other words, they don't use the host sequencer's interface for changing parameters.

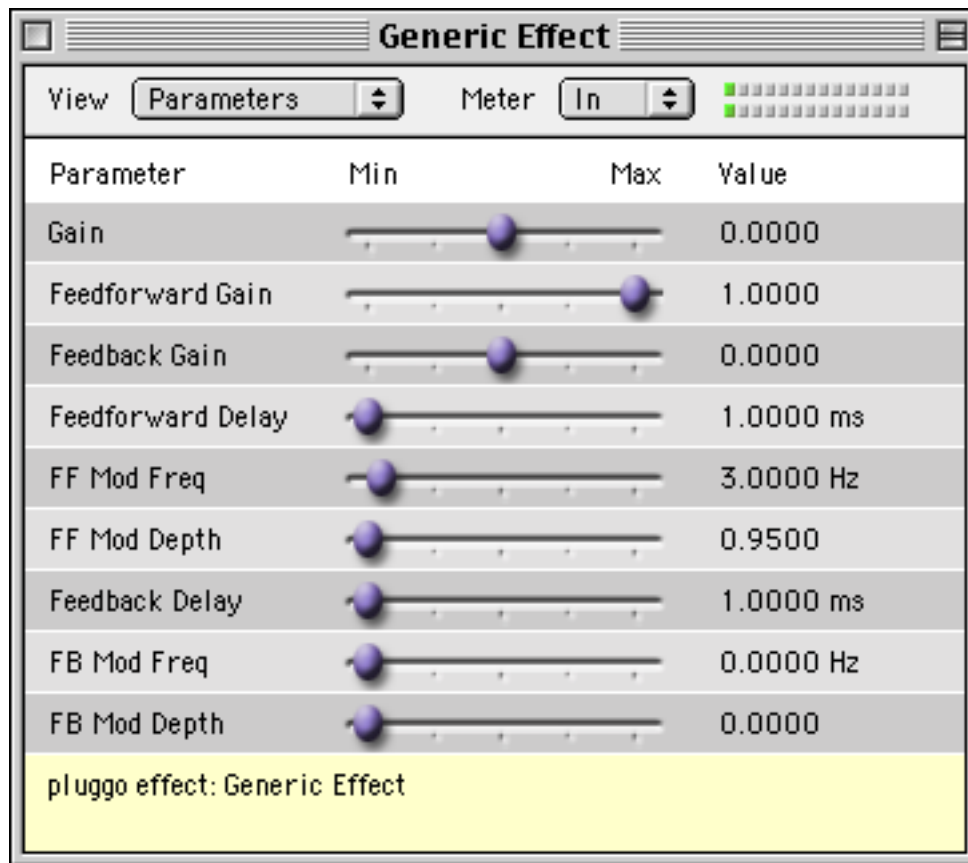
Using the Plug-in Interface in Cubase:

In Cubase VST, a plug-in with its own interface will appear as a *Rack Xpander*



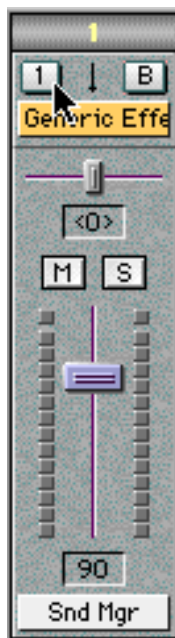
The Rack Xpander contains Prog buttons that let you change effect programs (collections of plug-in parameter settings). You cannot change programs from within a plug-in's edit window.

- To see the plug-in's edit window, click the Edit button. Here's the *Generic Effect* plug-in edit window in Cubase VST. You'll see an egg slider corresponding to each parameter of the effect.

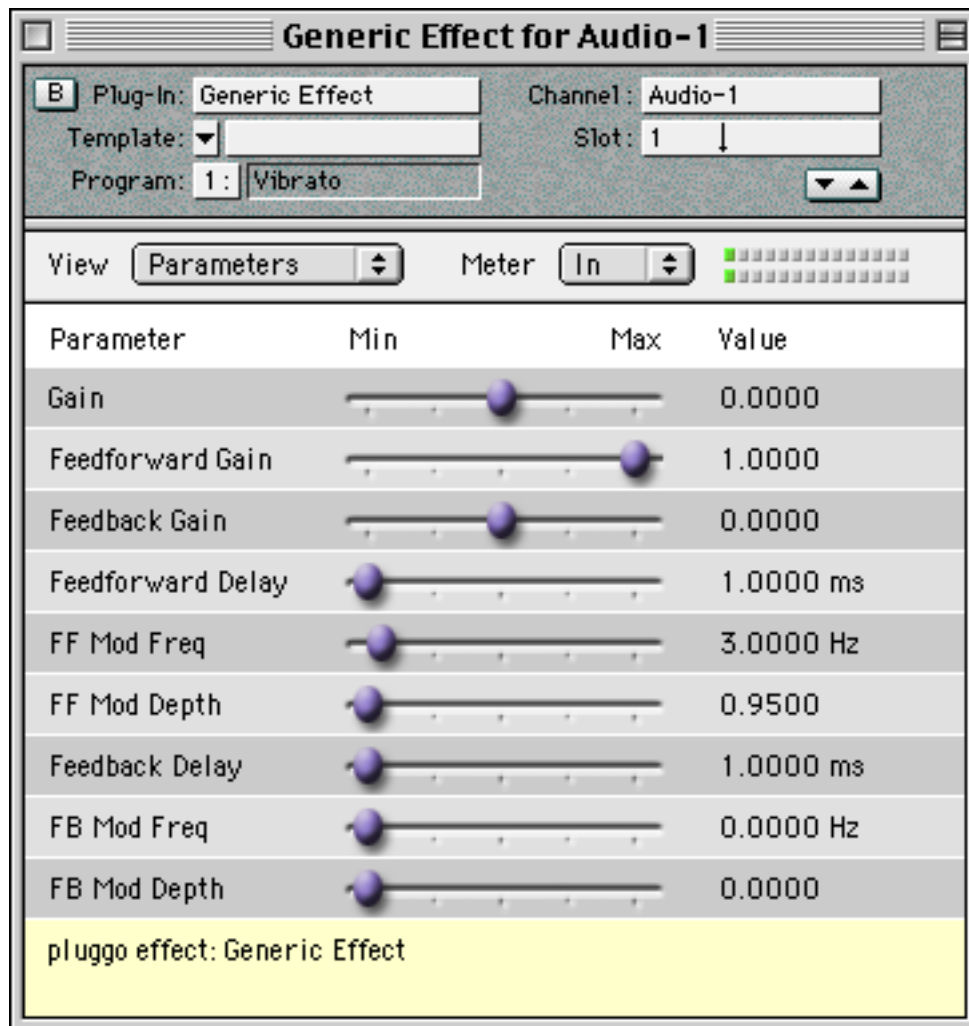


Using the Plug-in Interface in Vision:

In Vision, you click on the number of the plug-in to bring up its edit window.



Vision puts a pop-up menu of the effect programs along with mixer configuration information in a pane above the editing interface of the plug-in. Here's the *Generic Effect* plug-in edit window in Vision.



Using the Plug-in Interface in Logic Audio:

To open the plug-in's edit window in Logic Audio, double-click on the name as displayed in green.



Logic Audio adds plug-in controls beneath the plug-in's edit window. If the plug-in window is too narrow, these controls are sometimes cut off.

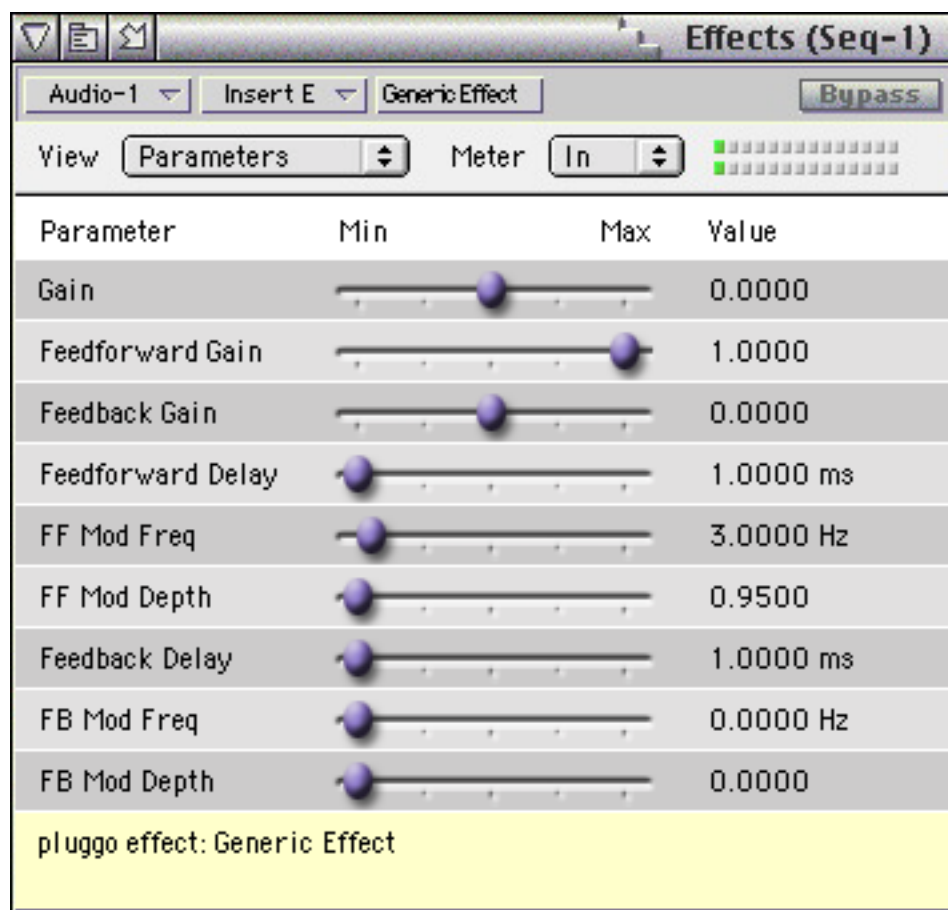
There is no way to change effect programs in Logic Audio. But you can use the Copy All from Program feature of the Parameter Change pop-up menu to listen to the presets included with each **pluggo**-based plug-in. Hold down the command key and click on any slider. Choose Copy All from Program on the pop-up menu that appears, and then choose an effect program. The settings for the program you select will be copied into the current program. You can Undo this action if desired.

Listening to the presets is strongly recommended—it will give you an idea of the range of sounds each plug-in is capable of producing.

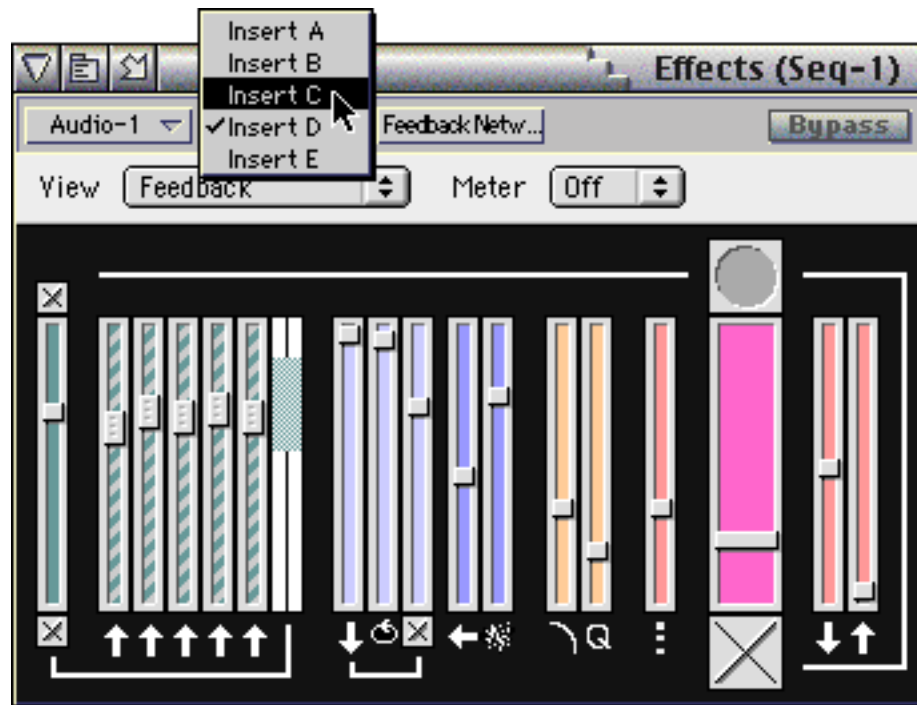
Logic Audio 4.0 allows you to use its generic interface for viewing plug-in parameters, which gives you a more compressed display than the one in **pluggo**. To see this display, choose Controls from the pop-up menu that displays Editor below the plug-in's edit window. To switch back to the normal **pluggo** interface view, choose Editor from this menu, which in the Controls display mode moves to the top of the plug-in's window.

Using the Plug-in Interface in Digital Performer:

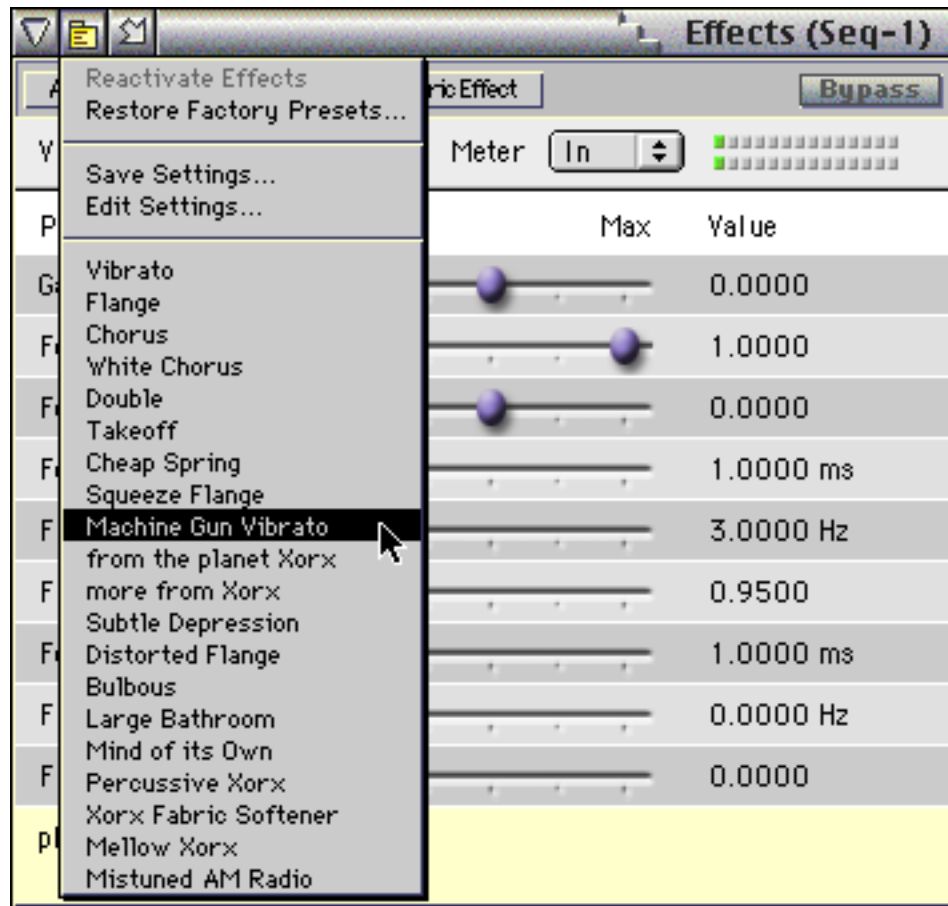
When you insert a plug-in in Digital Performer, its edit window appears automatically.



If you want to see a plug-in's edit window when it's not visible, choose its position from the Insert menu above the plug-in editing interface. For example, if the plug-in is third from the top, choose Insert C.



In Digital Performer 2.6 and later versions, you can access the preset effect programs (MOTU calls them *settings*) using the menu in the title bar of the Effects window.



However, settings work very differently in Pluggo running under MAS than they do under VST. Here is a summary of the differences:

- If you switch to another preset, Digital Performer does not warn you that you are about to lose any changes you may have made to the current preset. This is probably not so terrible; we'd guess the warning dialog would become irritating rather quickly. However, if you're used to the VST way of doing things, in which changes to any effect program are stored in the effect program when you switch to another program, you could lose some work.

To save your work, you have two choices. You can either write over an existing setting (probably not a great idea), or you can name a new setting that will be added to the menu at the top of the plug-in window. To save a new effect program, choose Save Settings... from the plug-in window's menu, name the settings, and click OK.

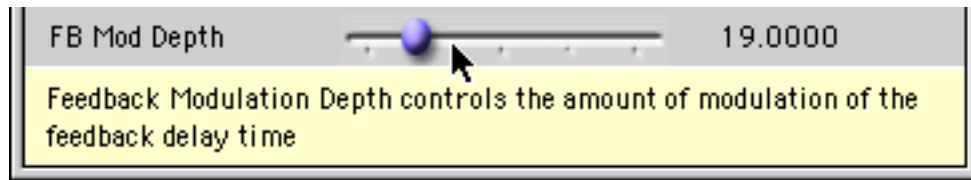
VST host applications limit the number of available effect programs to the number specified by the plug-in itself. With MAS, the number of settings is essentially unlimited.

In MAS, plug-ins do not have "program locations" based on the notion of a program change sent via MIDI to change a synthesizer to a stored state. A MAS plug-in's "settings" are collections of individual parameter values copied to a plug-in when the

user chooses a new preset from the menu at the top of the plug-in window. In MAS, Pluggo never knows when a Digital Performer user chooses a new “setting” or what the “current” setting is. This means that as far as the VST world is concerned, MAS-hosted plug-ins stay on the first program. New data from settings chosen by the user is simply copied into the first program location in the same way as would happen when you choose Copy All from Program from the Parameter Change pop-up menu.

Changing Parameters

As you’ve seen above, the initial interface for *Generic Effect* shows a set of sliders, one for each of the effect’s parameters. As you move the mouse over each slider, its knob (or its egg if you prefer) is highlighted and information about the parameter appears at the bottom of the window.



Changing parameters is more rewarding when they have an audible effect. Ensure that you have an audio signal coming into the plug-in, and that you are hearing the output of the plug-in. For *Generic Effect*, the initial effect preset program is a vibrato. This effect should be pretty obvious on most audio signals.

You can change effect parameters by using the sliders or in most cases by clicking directly on the displayed value of the parameter itself.

- To make coarse changes to an effect parameter using the sliders, click on the egg-shaped knob or anywhere along the length of the slider, then drag to the desired position. In coarse mode, there are about 100 different positions available.
- To make fine changes to an effect parameter using the sliders, double-click or option-click on the egg-shaped knob or anywhere along the length of the slider. When the slider is in fine mode, the egg turns red and the cursor disappears, because the slider moves much more slowly than your mouse is moving. There are 1024 different values possible from the left to right edges of the slider’s range. You can switch between fine and coarse modes by pressing and releasing the Option key within a single drag of the slider. The cursor will disappear and reappear according to whether you’re in fine or coarse mode.



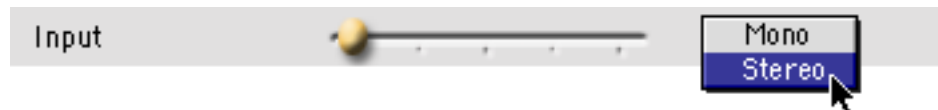
- To make changes to an effect parameter by clicking on its value, move the mouse over the value to the right of a slider. The value of the parameter will appear highlighted.



Changing a parameter by clicking on its value does not work for certain synchronization parameters, nor does it work for third-party VST plug-ins hosted by **pluggo**.

To change the value of the slider, move the mouse pointer over the digit in the value you want to change. When you click, this digit will turn yellow, and as you move the mouse, you will see the value of the parameter change by increments of this digit. Move the mouse up to increase the parameter and down to decrease it.

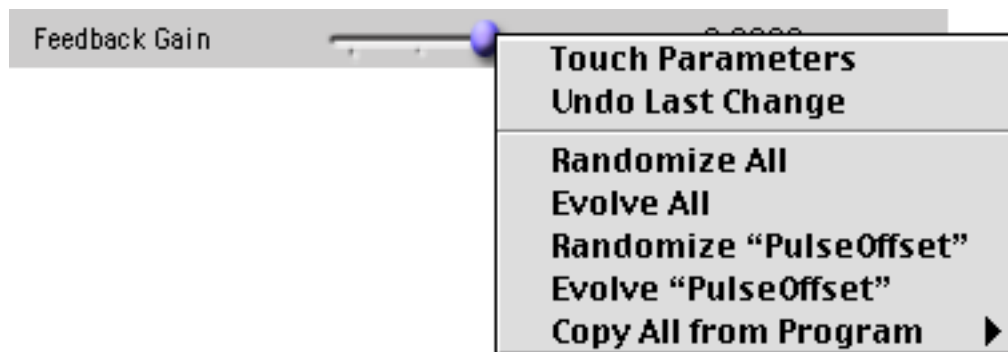
If a slider is used to select from a small number of possibilities (such as a mono/stereo select slider), you will see a pop-up menu appear instead. Use the menu to set the parameter's value.



When you change a parameter by moving a slider, the value is stored within the current effect program (displayed in the Rack Xpander in Cubase and above the plug-in's edit window in Vision and Digital Performer). With VST hosts, you don't have to save a program to store your changes as you might on a typical hardware effects unit. With Digital Performer, you *do* have to save any changes you make, and you aren't warned when you're about to lose them, for example when switching to another preset.

The Parameter Change Pop-up Menu

Hold down the command key and click on a slider. You'll get a pop-up menu that looks something like this:



The Parameter Change pop-up menu contains commands that deal with parameter values in the plug-in.

Touch Parameters is used to send out messages to the host sequencer that describe the current position of all the controls for a plug-in. It is used with Vision and Logic Audio to aid in plug-in automation. It tells the host that all parameters have been "changed" to their current value, so that host can record these changes. This command has no effect on the actual values of any plug-in parameters. For an example of using this feature, see the *Automating Pluggo* chapter.

Undo Last Change allows you to go back to the previous setting of a parameter that you changed with a slider. It can also return you to the state the plug-in was in before selecting a new program.

Choose Undo Last Change again to return to the change before you undid it. (There is only one level of undo available.)

It doesn't matter what slider you command-click on to activate the pop-up menu if you want to choose Undo Last Change. Only the most recent change to any slider is undoable, so the slider that was changed last will be reset.

Randomize All will set the parameters to random values. This can be useful in finding effect settings you never would have been able to come up if you had adjusted the sliders one at a time. Not all parameters may be affected however; the developer of the plug-in can keep certain parameters (such as output gain) from being randomized that would simply be irritating if they were changed.

Evolve All will nudge each parameter by a random amount up to five percent greater or less than its current value.

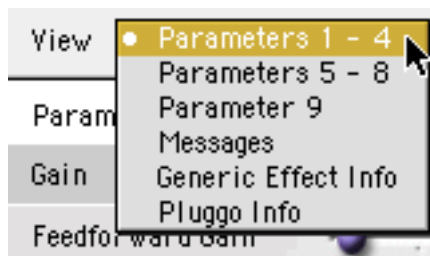
If you click on a particular slider, you'll get two additional menu items specific to the parameter controlled by the slider. Randomize "Parameter Name" will set the value of the parameter whose slider you clicked on to a random value. Evolve "Parameter Name" will nudge the parameter whose slider you clicked on by a random amount up to five percent greater or less than its current value.

Copy All from Program lists the names of all of the programs for this plug-in. When you choose a program name from this menu, you copy the settings from the selected program into the current program. You can Undo this action if desired.

Using the View Menu

At the top of the **pluggo** edit window, you'll see a menu labeled View. When you see the rows of egg sliders, the View menu displays the word Parameters. Click on this menu to see other pages or *views* of the plug-in's interface.

Some plug-ins contain more parameter sliders than will fit in a single page. As an example case, if a plug-in has four parameters on its initial screen, the View menu will contain an item labeled Parameters 1 - 4. A subsequent page of four parameters would be labeled Parameters 5 - 8.



Some plug-ins do not display sliders for their parameters as *Generic Effect* does; instead they present an Interface view unique to the plug-in, possibly displaying information in a graphical form. Other plug-ins will have both a Parameters view and an Interface view. The Interface view may not be called Interface in the View menu; the plug-in developer can name it anything he or she wants.

- Choose Generic Effect Info from the View menu. You'll see a screen that provides information about the plug-in and its developer unless he or she chooses to remain anonymous.
- Choose Pluggo Info from the View menu. This is **pluggo**'s About Box. Here's a secret: clicking on the **pluggo** character is the same as Randomize All from the Parameter Change pop-up menu. The Pluggo Info view will appear and behave slightly differently depending on the size of the plug-in.
- Some plug-ins provide a Messages view. This window may provide diagnostic information that could be helpful if the plug-in does not appear to be working. For instance, you might see a message something like this after opening a plug-in:

error: plugin~: no such object

This particular error indicates an incorrect or corrupted **pluggo** installation.

Note that the contents of the Messages view is common to all **pluggo** plug-ins. In other words, if you've already loaded a plug-in that reported information or errors in the Messages view, you'll see the messages it generated in the Messages view of any subsequent plug-in you insert.

Elements of Interface Views

Plug-ins may contain one or more custom Interface views as discussed above. In the Interface view, the Parameter Change pop-up menu is also available when you command click anywhere in the window. Plug-ins with Interface views will contain user interface elements from Max and MSP that you may not have seen before. These include:

The number box



To change a numerical value, click on the box and scroll up or down.

The horizontal or vertical slider



Using these sliders is self-explanatory, except that the cursor disappears while moving the knob of the slider. This allows fine changes similar to the “fine mode” of the egg slider discussed above.

The vertical gain slider

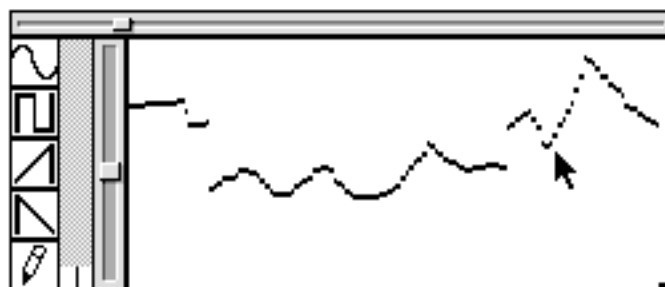


This slider is used to make smooth exponential fades to audio signals. The top value of the slider is usually +18 dB, 0 dB is where the knob of the slider is located in the above picture, and the bottom is -75 dB (essentially the same as a volume knob turned all the way down).

The multislider



The multislider is a collection of several sliders arranged from left to right. Drag the cursor over the entire picture to set several slider values. This user interface element has a variety of uses. Sometimes it will appear as an array of dots or squares, as in the waveform draw portion of the LFO plug-in shown below.



The range slider



This slider lets you draw out a range of values. Drag from one end of the range to the other.

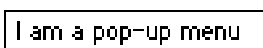
The toggle box



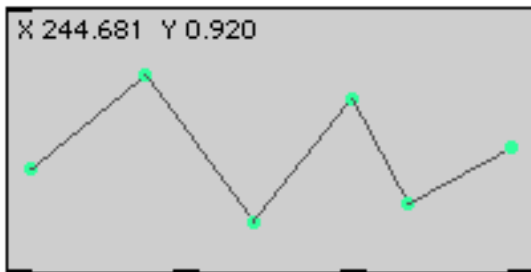
Click on the box to turn something on or off.

The pop-up menu

Pop-up menus look like this:

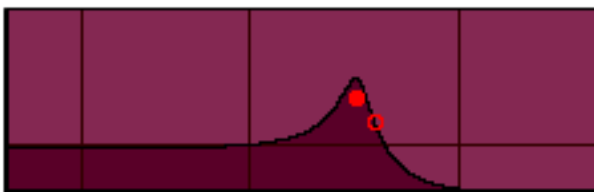


The breakpoint envelope editor



You can draw an envelope for various purposes with this box. Click on one of the existing breakpoints and dragging on it will move it to a new location. There is a display of the values as you drag the breakpoint. Some breakpoint envelope editors will allow you to add a new point by clicking anywhere on the envelope. You may also be able to delete points by shift-clicking on them.

The filter graph

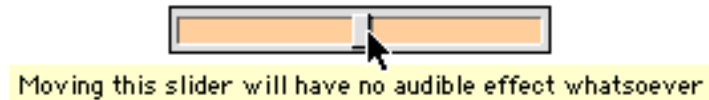


The filter graph lets you draw the frequency response of a filter in several different modes. The solid red circle is used to move the center or cutoff frequency. The hollow red circle, where applicable, determines the width of a bandpass filter. Often the filter graph will be accompanied by a set of buttons for choosing filter type: lowpass, bandpass, highpass, etc.

You might also see knobs and buttons of various kinds not described here. If you're mystified about how they work, the documentation for the specific plug-in effect in which they appear should clarify the situation.

Interface Hints and Labels

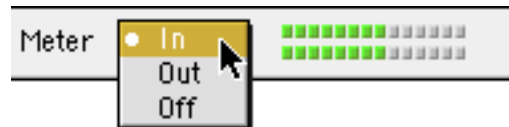
In addition to the text provided in the Parameters view when you move over a slider, the Interface View provides a hint in a small text box about the function of its control elements. Here's an example of a hint over a horizontal slider.



If you find the hints bothersome, command-click to get **pluggo**'s Parameter Change pop-up menu and choose Disable Hints (a menu item only available in an Interface view). This disables hints for all plug-ins. To re-enable hints, command-click to get the pop-up menu again, and choose Enable Hints.

The Level Meter

pluggo contains a handy level meter that tells you whether the plug-in is getting any input signal level or producing an output signal. Normally the meter is set to display the level of the input signal, as shown below:



- Choose Out to have the meter display the plug-in's output signal level.
- Choose Off to turn the meter off. Some plug-ins may turn the meter off by default. As with any on-screen level meter, the **pluggo** meter does consume a tiny amount of the CPU available for signal processing.

Plug-in Manager

Plug-in Manager is a standalone application that manages plug-ins for your sequencer and many other applications. It works very much like the Mac OS Extensions Manager. If you're not familiar with the Extensions Manager concept, here's what you need to know.

Plug-ins, like system Extensions, need to be placed in a specific folder in order to be used. For VST-compatible sequencers, this folder is called VstPlugIns. For system Extensions, the folder is called the Extensions folder and is found inside your System Folder. Take something out of either folder and the next time you restart your computer or launch the application, it won't be used.

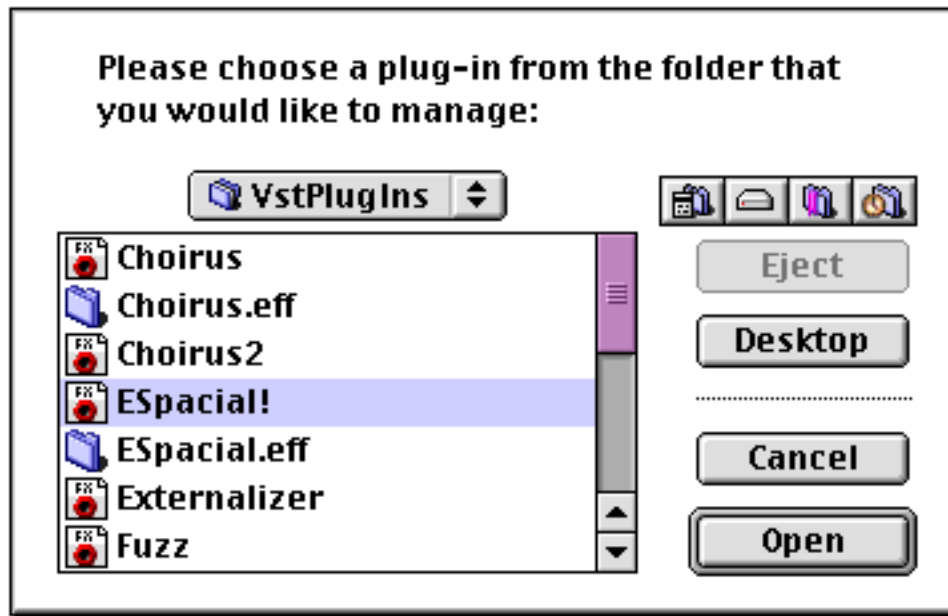
If your sequencer's plug-ins folder is called VstPlugIns, Plug-in Manager creates a folder called VstPlugIns (disabled) inside the folder containing your sequencer and moves plug-ins you don't want to use into this folder. But this file moving operation is transparent to you if you use the program. All you do to enable and disable a plug-in is click a checkbox next to the plug-in's name.

For MAS host applications, the **pluggo** installer creates VstPlugIns and VstPlugIns (disabled) folders containing all the plug-ins it will install. However, other non-**pluggo** MAS audio effect plug-ins are contained inside the Plug-ins folder inside the MOTU folder inside the Extensions folder. Unfortunately, Plug-in Manager has to manage these two folders in separate windows.

Plug-in Manager is a free (albeit copyrighted) application. You're welcome to give it to your friends. You just can't sell it.

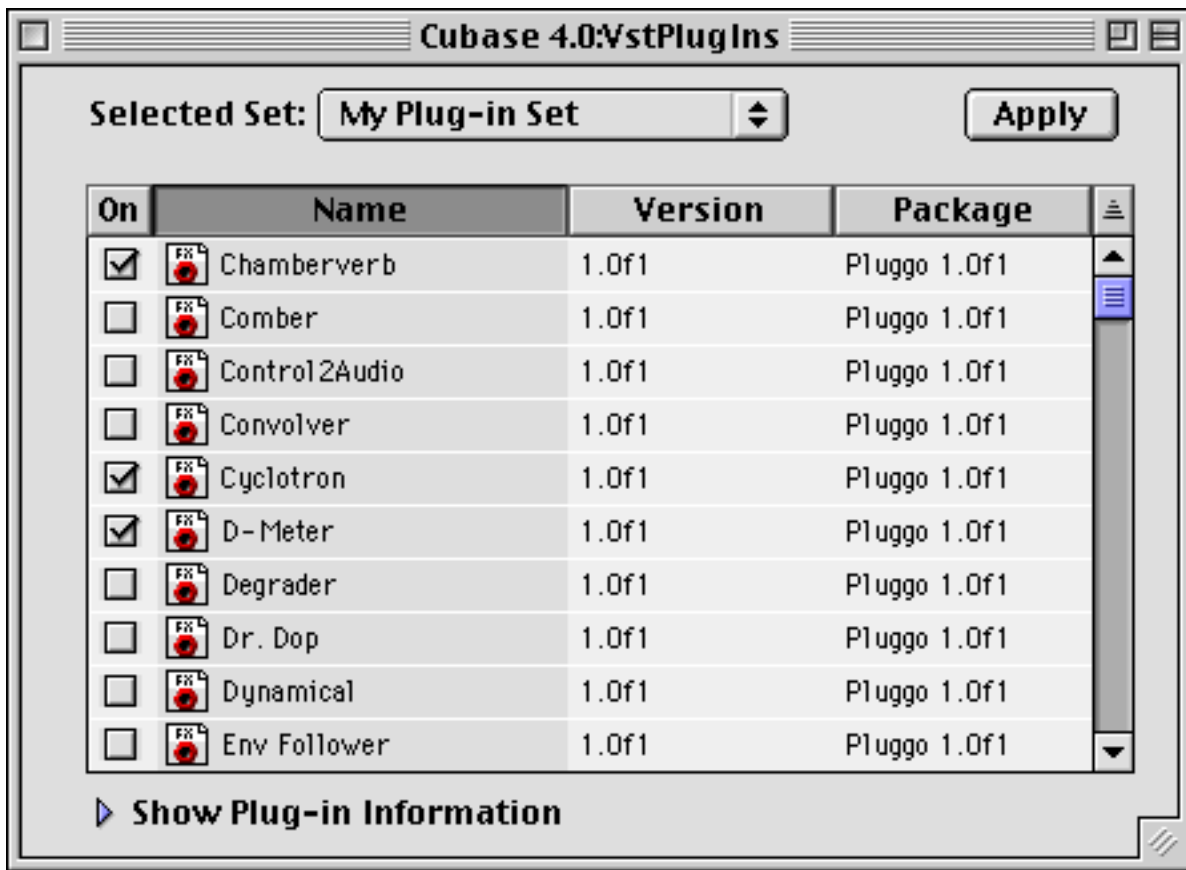
Starting Plug-in Manager

The first time you launch Plug-in Manager, you'll be asked to choose plug-in file inside the plug-ins folder you want to manage. Doing so tells Plug-in Manager both the type and location of the plug-ins.



Plug-in Manager can manage plug-ins for other applications—graphics programs, web browsers, etc. This use of the program is not discussed here. But if you need a plug-in manager for another application, give it a try.

When you've chosen a Plug-in Folder and plug-in type, Plug-in Manager displays a list of all the plug-ins of the chosen type inside the specified folder. An example is shown below.



Unless you've created a VstPlugIns (disabled) folder yourself, you'll find that initially all plug-ins will be enabled, so the "On" boxes to the left of their names will all be checked.

After you select this folder, it's saved in a file called Plug-in Manager Preferences in the Preferences Folder in your System Folder. You can choose other plug-in types in other folders by choosing New Folder from the File menu. Once a plug-in folder has been placed under management, it is listed in the Manage submenu, so you can switch back to working with it at any time.

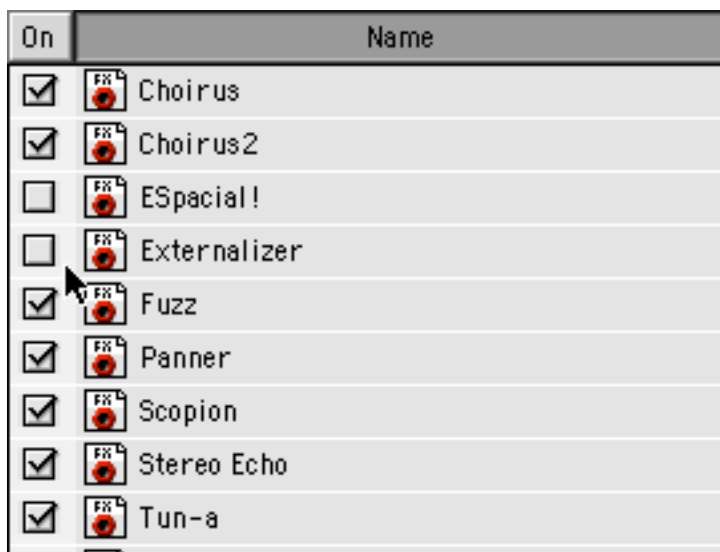
Enabling and Disabling Plug-ins

Before discussing *how* to disable a plug-in, we should first discuss *why* you might want to disable a plug-in. Here are some possible reasons:

- You aren't using it and you find its presence in a list of available plug-ins distracting
- You suspect it's guilty of destabilizing the host application program
- You've been overusing a plug-in lately and you need to find a fresh artistic approach.

A plug-in is considered On by Plug-in Manager if it is located in the plug-ins folder you're managing, and Off if it is in the plug-ins (disabled) folder. Plug-in Manager

displays the current state of these two folders. To disable a plug-in, simply uncheck the box to the left of its name.



Most applications, including those that use Plug-ins, only change the list of plug-ins they use when you start up the host application. So you'll need to quit the host application and restart it before the changes will take effect, just as you need to restart your computer before changes to extensions take place in the Extensions Manager. A message below the list of plug-ins will remind you of this as you enable and disable plug-ins with Plug-in Manager.

When Are the Files Moved?

A plug-in file that you enable or disable is not actually moved when you check or uncheck the box next to its name. Files are moved from the plug-ins folder to the plug-ins (disabled) folder (or vice versa) at the following times:

- when you click the Apply button
- when you close a window for a plug-in folder
- when you quit Plug-in Manager

After the files are moved, you can restart the host application and its plug-in list will be updated.

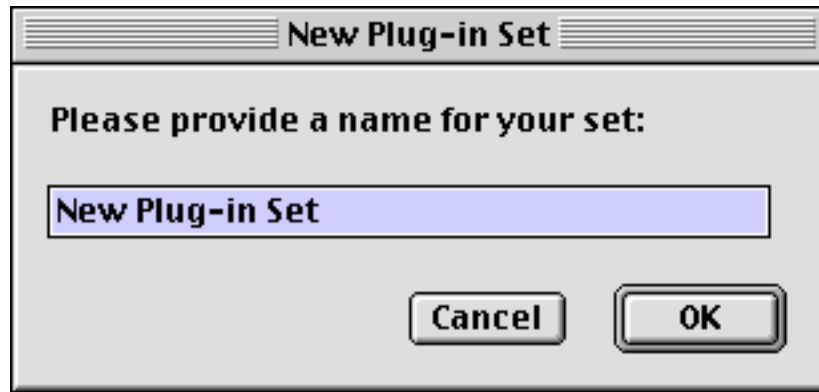
If you're managing plug-ins for Digital Performer, you can update the plug-in list more quickly by choosing MIDI Only from the Audio System submenu of the Basics menu in Digital Performer. Then choose MOTU Audio System from the Audio System of the Basics menu in Digital Performer. This reinitializes MAS. This technique will apply to MAS plug-ins in the Extensions folder as well as Pluggo plug-ins in the VstPlugIns folder.

Getting Information About a Plug-in

At the bottom of the Plug-in Manager window, a box shows you the date, version (if available), and additional information describing the plug-in. Not all plug-ins will have any additional information available.

Saving Plug-in Sets

You can create sets of plug-ins that you want to use together. A set describes for a given plug-in folder which plug-ins are enabled and which are disabled. Initially, a plug-in folder has one set, called My Plug-in Set. To create a new set, choose Create Set from the File menu. The Plug-in Manager will ask you to give a name to your plug-in set:



Once you've created a set, its name appears on the popup menu at the top of the Plug-in Manager window. The current state of each plug-in, enabled or disabled, is stored in your new set.

You can create as many sets as you need for a managed plug-in folder. To recall a set, choose its name from the popup menu. The enabled/disabled state of each plug-in is restored to match the states of the plug-ins as they were when you created the set.

Working with Multiple Plug-in Folders

Plug-in sets can be saved for each Plug-in Folder you manage. To open a new plug-in folder for management, choose its name from the Manage submenu of the File menu (for a previously managed folder) or choose New Folder from the File menu (for managing a new folder).

Plug-in Manager will open a new window for the folder. The pop-up menu at the top of the new window lists all of the plug-in sets that you have created for the Plug-in Folder. If you haven't created any sets yet, only the one called My Plug-in Set will be listed.

Working with Nested Plug-in Folders in Cubase

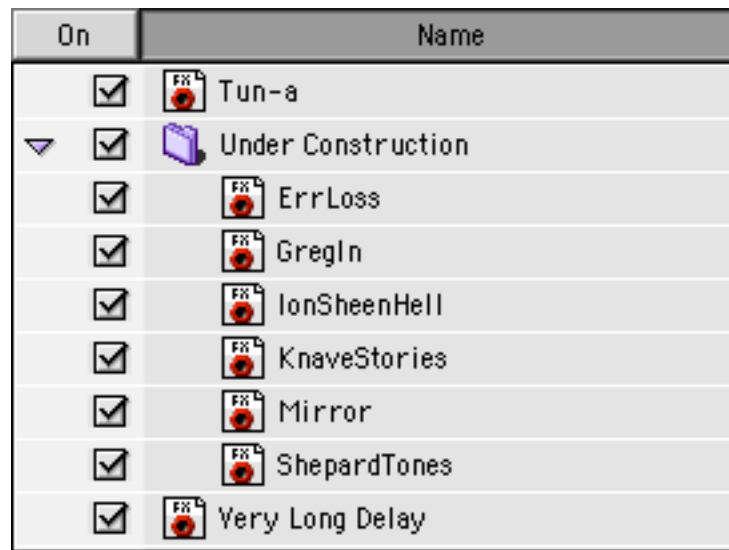
Cubase 4.1 and later versions let you organize the VST effects in your VstPlugIns folder by using subfolders. You can create any subfolders you want inside the VstPlugIns folder, and then move VST plug-ins into those folders. They will then appear as submenus in any effect selection pop-up menus.



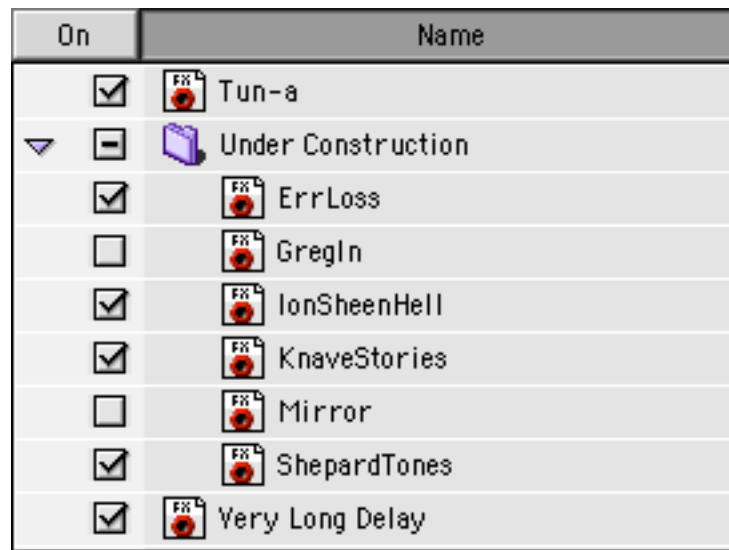
With this feature, you can create subfolders for different types of plug-ins (such as granular plug-ins, distortion plug-ins, or plug-ins that you're currently developing) and move the plug-ins into those folders. You can then view and select your plug-ins quickly and easily, even if you have a large number of plug-ins in the Cubase VstPlugIns folder.

You will need to create any subfolders in the VstPlugIns folder that you want to use manually using the Finder before you run Plug-In Manager. You cannot use Plug-In Manager to create subfolders and cannot move plug-ins from one folder to another.

When Plug-in Manager finds subfolders containing plug-ins, it adds the folder's name to the display along with a disclosure triangle. You can click on the disclosure triangle to the left of a folder's icon to view the plug-ins contained in that folder.



Plug-ins within a subfolder can be enabled/disabled individually. If a folder contains both enabled and disabled plug-ins, its checkbox will show a horizontal line. You can enable/disable all of the plug-ins within a subfolder by clicking the subfolder's checkbox.















Plug-In Manager adds subfolders to the disabled plug-ins folder to match subfolders in the corresponding plug-ins folder. If you add, rename, or move subfolders or plug-ins within a plug-in folder, Plug-In Manager will accommodate the changes and display the subfolders and plug-ins appropriately the next time it runs. However, any plug-in sets you created before rearranging the contents of your plug-in folder will probably not be useful any longer, since Plug-In Manager will not be able to restore your plug-ins to a previous state if they are no longer in the same place as they were when that state was saved with a plug-in set.

Do not attempt to move plug-ins or subfolders of the plug-in folder when Plug-In Manager is displaying the contents of this folder in one of its windows. It will become confused, and you know how dangerous a confused Manager can be.

Also, the preferences files used by Plug-In Manager 1.1 (included with **pluggo 2.0**) are not compatible with older versions of Plug-In Manager. If you have used Plug-In Manager 1.0, all of your preferences files—your saved plug-in sets for each application whose plug-ins you manage—will be converted to the new format when you run Plug-In Manager 1.1 for the first time. After this conversion, earlier versions of Plug-In Manager will not recognize your plug-in sets, and should not be used.

Working with Nested Plug-in Folders in Logic Audio

You can use folders within the Logic Audio VstPlugIns folder to organize your plug-ins. However, you will not see the folder names themselves in Logic Audio when you choose a plug-in.

On	Name
<input checked="" type="checkbox"/>	 Audio Rate Pan
<input checked="" type="checkbox"/>	 Audio2Control
<input checked="" type="checkbox"/>	 Average Injector
<input checked="" type="checkbox"/>	 Breakpoints
<input checked="" type="checkbox"/>	 Center Channel
<input checked="" type="checkbox"/>	 Chamberverb
<input checked="" type="checkbox"/>	 Comber
<input checked="" type="checkbox"/>	 Construction Zone
<input checked="" type="checkbox"/>	 Mirror
<input checked="" type="checkbox"/>	 ShepardTones
<input checked="" type="checkbox"/>	 Control2Audio
<input checked="" type="checkbox"/>	 Convolver

Instead, Logic Audio uses folders to determine the order in which plug-ins appear in its effects menu. Normally, Logic Audio displays plug-in names in alphabetical order. But the plug-in menu below shows the effect of having a Construction Zone subfolder shown in the window above. Note that Mirror and ShepardTones appear after Comber in the menu, rather than with plug-ins whose names begin with M or S. In other words, Logic Audio alphabetizes by the name of the folder instead of by the names of the plug-ins within the folder.

Synchronizing Plug-ins

The synchronization features of **pluggo** allow the operation of a plug-in to be tied to events going on in the outside world. Some, but not all, **pluggo** plug-ins support one or more forms of synchronization. The “slickest” synchronization mode uses the timing information from the sequence, and is available in applications that support the VST 2.0 or MAS plug-in formats. For sequencers without this capability, or for special types of synchronization that respond to elements of audio tracks themselves, there’s the *PluggoSync* plug-in, which can “listen” to an audio signal fed to its input and derive sync information from it that can be used by other **pluggo**-based plug-ins.

In the other synchronization modes, plug-ins use their own timing source that has no connection with anything going on outside.

Opening the Sync Example Document

To demonstrate synchronization, we’ve included example documents for Cubase 3.x, Cubase 4.0, Cubase 4.1, Vision, Logic Audio, and Digital Performer.

- Choose Open... from your sequencer’s file menu.
- Locate and select the Sync Example document appropriate for your sequencer. You’ll find it in the Pluggo Getting Started Examples folder within the Pluggo Stuff folder.
- In your sequencer’s transport window, click on the Play button, then when you’ve heard enough, click Stop.

This document contains samples of spoken numbers processed through the *Audio Rate Pan* plug-in. What we’ll be discussing in the context of this example is various ways we can set the rate of left-right panning.

The Audio Rate Pan edit window should appear when you open the document. Note that synchronization parameters appear in orange, a convention followed by most **pluggo**-based plug-ins.

Exploring the Synchronization Modes

There are four sync modes available in the *Audio Rate Pan* plug-in: Free, Host, Plug, and UDT. You set the sync mode by using the pop-up menu in the Tempo/Sync parameter (number 9). Not all plug-ins will organize sync options in exactly the same way, but *Audio Rate Pan* provides all the available options you’ll encounter. We’ll summarize each mode briefly here:

- *Free* mode allows you to determine rate and duration parameters using milliseconds or Hertz units. In this mode, the plug-in ignores synchronization information, or to put it another way, it is “free running.”
- *Host* mode locks the rate and duration parameters to the tempo of the sequence you’re playing through the plug-in. Typically, no change in these parameters will be heard unless the sequence is actually playing. You specify rate and duration

parameters in terms of note units, such as whole notes or quarter notes, plus a multiplier. Host mode can be selected in host environments that don't support direct synchronization, but it will be the same as specifying a duration of zero.

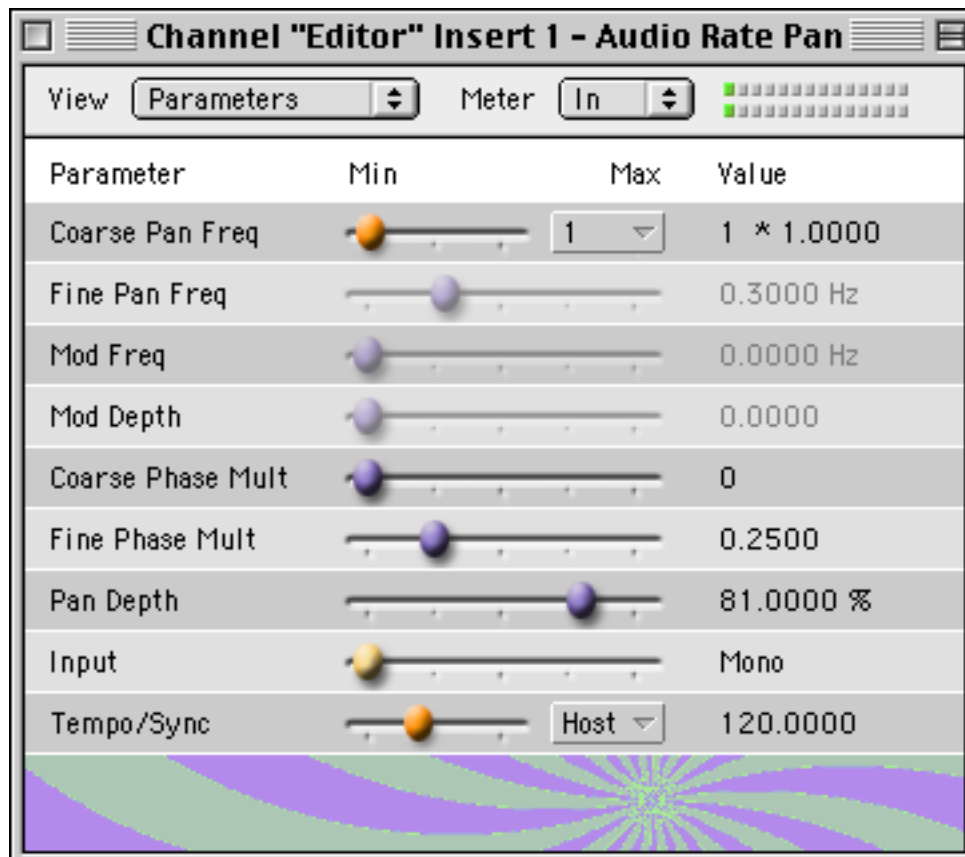
Plug mode is similar to Host mode except that the *PluggoSync* plug-in supplies the synchronization information instead of the host sequencer. This means that in order for Plug mode to work, you'll need to have a *PluggoSync* plug-in inserted someplace. We'll discuss how *PluggoSync* works below. Plug mode is available in all host applications.

UDT (which stands for User-Defined Tempo) mode is similar to Free mode in that no synchronization between the plug-in and the host occurs. It differs from Free mode in that you're allowed to specify rate and duration in terms of note units and a multiplier. However, the plug-in uses a tempo you specify rather than obtaining it from the host. In other words, UDT is simply an alternate method of specifying time that might be more musical for certain users than milliseconds or Hertz.

Host mode

In the example documents for Cubase 4.1 and Digital Performer, Host mode is initially selected in the *Audio Rate Pan* edit window. Users of applications that do not support this capability can skip this section (or read it with envy).

In Host mode, the modulating oscillator that pans the audio input from left to right is synchronized to the beat. The *Audio Rate Pan* edit window appears as shown below.

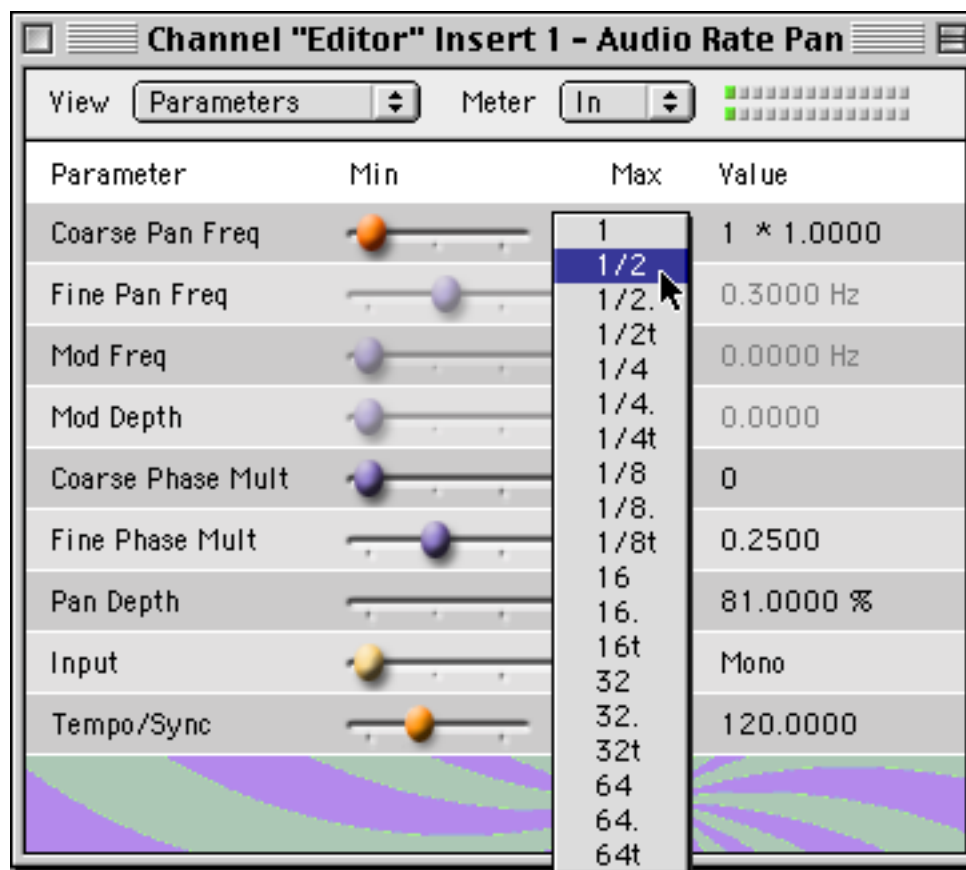


Note at the bottom that the Tempo/Sync parameter is set to Host and that you can't change the orange egg slider next to the pop-up menu. In Host mode, the slider acts as an *indicator* of the current tempo of the sequence that is playing. You'll also note that the Fine Pan Freq, Mod Freq, and Mod Depth parameters are grayed out. These are disabled in both Host and Plug modes. Instead, the information that controls the audio panning rate is derived from your sequencer application and you set the relationship of the panning frequency to the tempo by using the slider and pop-up menu combination shown for the Coarse Pan Freq parameter. In the example, the Coarse Pan Freq is $1.0000 * 1$, meaning 1.0 times a whole note (i.e., four beats if the time signature is 4/4).

- Start playing the example sound file and change the tempo value in the tape transport window. You will notice that the value of the Tempo/Sync slider changes when you alter the playback tempo of the sequence. You should hear the pan rate change in accordance with the tempo as well.



- Click on the pop-up menu in the Coarse Pan Freq parameter. This menu lets you specify note units that subdivide the beat at the current tempo.



These options represent abbreviations for various timing values of musical notes. Included are whole note (1), half note (1/2), quarter note (1/4) and so on. A note value with a period after it represents a dotted note. A note value with a "t" after it represents a triplet.

- Select the half note value from the pop-up menu. You will now hear the audio rate sweep at twice the rate of speed (i.e. four times per second).



So far, the Coarse Pan Freq parameter's egg slider has been set to a value of one. You can use the slider to set a value to multiply the note unit chosen in the pop-up menu. For instance, you can use this to set much longer durations of synchronized change.

- Set the Coarse Pan Freq slider to a value of 4.0000, and set the pop-up menu value to a whole note, as shown:



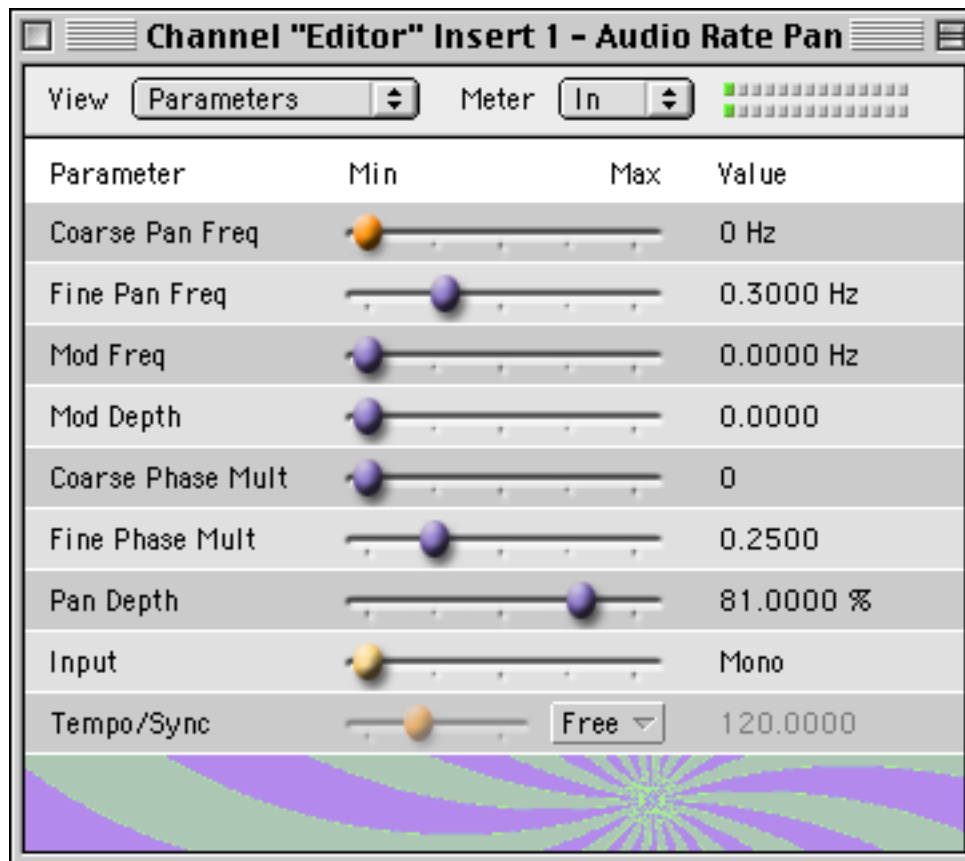
You will now hear the audio pan from left to right once every two seconds. You can use the slider and the note value pop-up menu to create complex timing relationships.

Free mode

Free mode is supported by all host applications. It's the mode initially selected in the Cubase 4.0, Vision, and Logic Audio example documents.

- Choose Free from the Tempo/Sync parameter pop-up menu.

In Free mode, the plug-in uses its own built-in clock or oscillator to derive rates of modulation (in this case, the Coarse and Fine Pan Freq sliders). You'll notice that the Tempo/Sync egg slider is grayed out. That's because the modulation rate is not set in terms of a relationship to a beat; rather, it is set in terms of absolute time. If you were a user of **pluggo** 1.0, you'll recognize this as the way the *Audio Rate Pan* plug-in functioned.

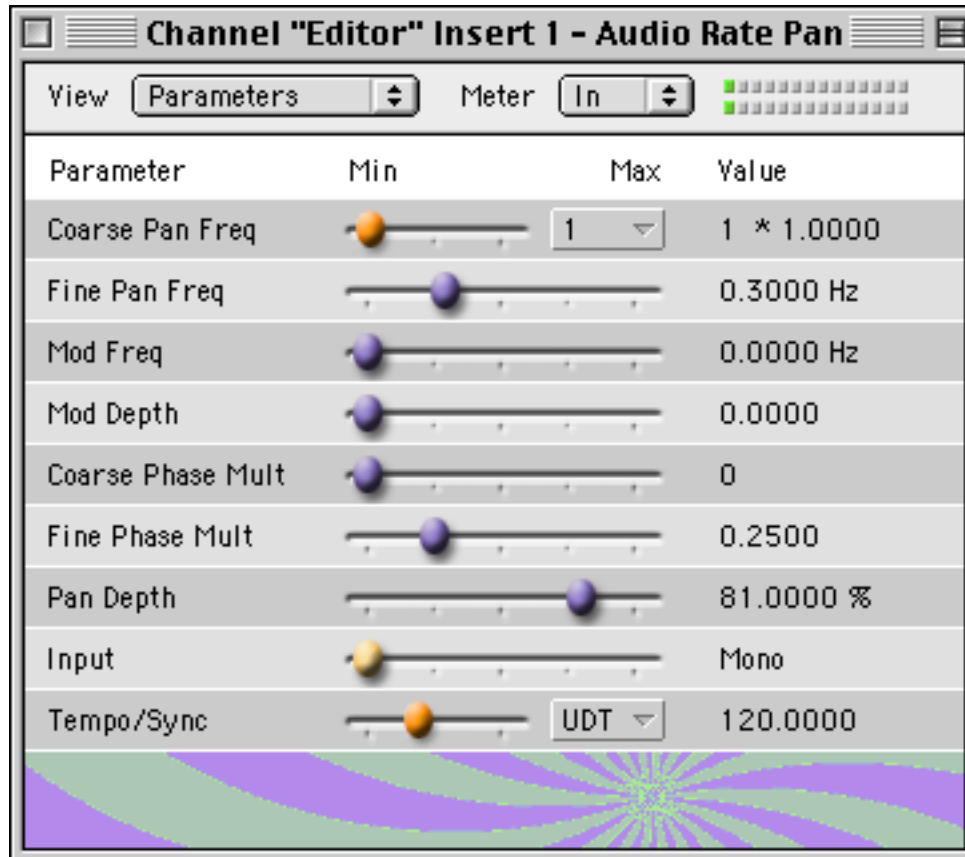


UDT mode

- Choose UDT from the Tempo/Sync parameter's pop-up menu.

As mentioned before, UDT stands for User-Defined Tempo. This mode allows you to specify a tempo, note unit value, and multiplier to determine the rate of panning. These values are independent of the tempo of the host sequence.

The Fine Pan Freq, Mod Freq, and Mod Depth parameters are available in UDT mode.



- Click on the displayed value of the Tempo/Sync parameter and change the tempo to a lower value. You should hear the rate of panning slow down accordingly.



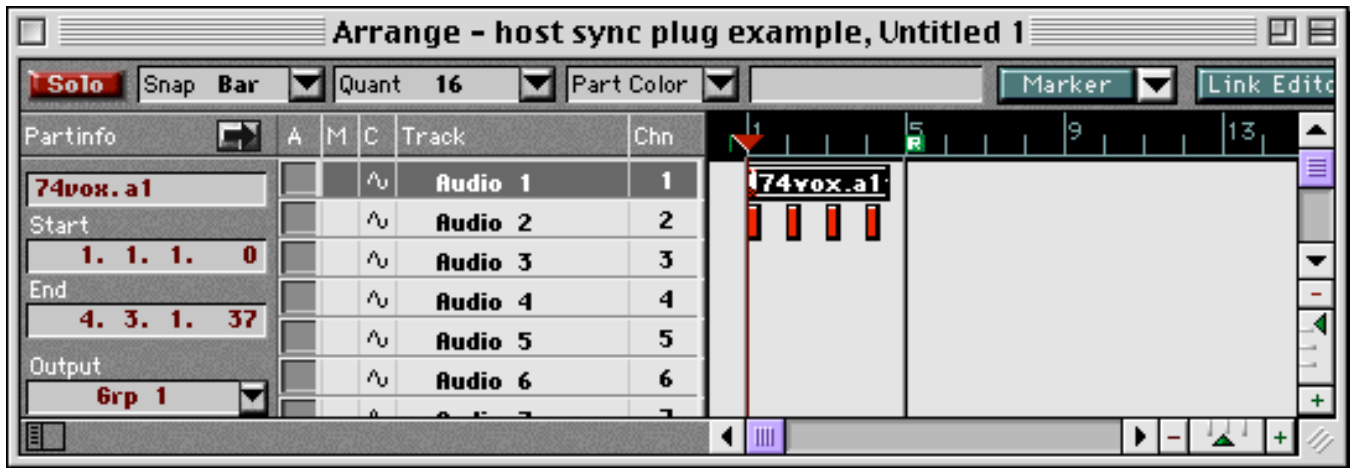
You can also use the Coarse Pan Freq note value pop-up menu and multiplier slider to change the panning rate.

Plug mode

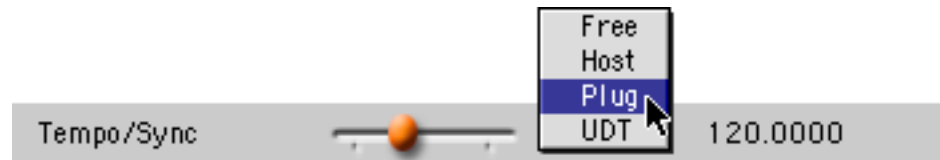
Prior to the release of host applications with synchronization, **pluggo** used a plug-in called *PluggoSync* to provide a way to synchronize plug-ins with the tempo of your music. *PluggoSync* uses a “click track” that feeds the *PluggoSync* plug-in, and the plug-in then outputs synchronization to other plug-ins. We’ll discuss the *PluggoSync* plug-in in more detail in the next section.

This technique is still applicable to those host applications that do not have synchronization features, but it’s also useful in those that do. You can use the *PluggoSync* plug-in to respond to cues other than the timing information supplied by your host application (for example, you can use it to insert cues in your song that control your plug-ins independent of timing and tempo).

We've included a second audio track in the Sync Example document containing a “sync sample” audio file that is fed to the audio input of the PluggoSync plug-in. The sync audio file has been placed at the beginning of every beat. This is the click track. You'll also see the first track that contains the audio that's being panned. Here's what the song looks like in Cubase.

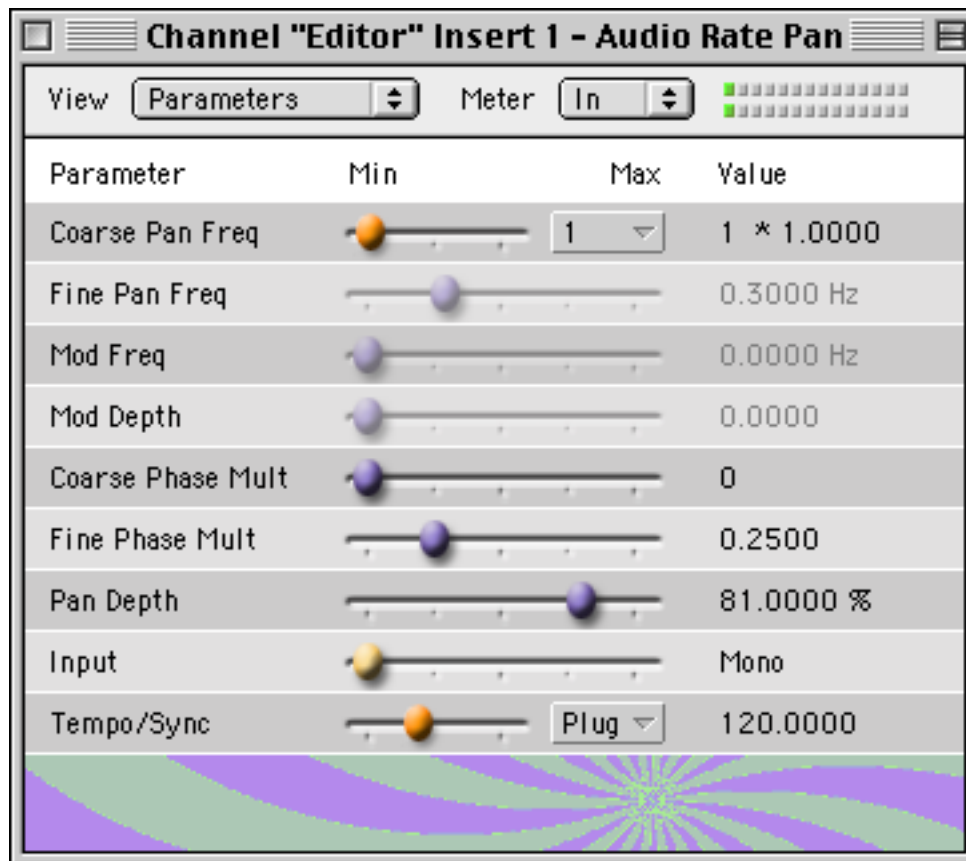


- In the *Audio Rate Pan* plug-in, choose Plug mode from the Tempo/Sync parameter's pop-up menu.



In the *Audio Rate Pan* plug-in, Plug mode functions similarly to Host mode. You specify the relationship between the panning rate and the tempo derived from *PluggoSync* using the note value pop-up menu and multiplier slider in the Coarse Pan Freq parameter.

Note that the Fine Pan Freq, Mod Freq, and Mod Depth parameters are grayed out, as in Host mode. The information that controls the audio panning rate is derived from *PluggoSync*.



- Play the sequence. You should hear the *Audio Rate Pan* plug-in panning the audio track in sync with the click track on Audio 1. This click track is sending its audio signal to the *PluggoSync* plug-in instead of using the host application.
- Try using your sequencer's arrangement window to change the spacing of the sync track units at odd intervals and listen to what happens.
- Return to the edit window of *Audio Rate Pan*. While the sequence is playing, try changing the Coarse Pan Freq note value pop-up menu and multiplier sliders to change the panning rate.

In the next section, we'll take a closer look at how the *PluggoSync* plug-in works.

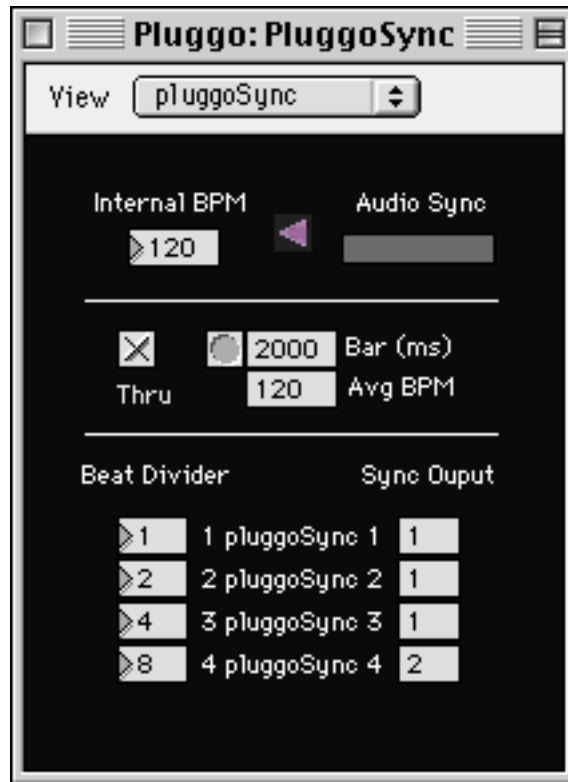
Using PluggoSync

If your sequencer doesn't support host sync, the *PluggoSync* plug-in provides a way to synchronize plug-ins with the tempo of your music. With modulation plug-ins such as *LFO*, this lets you change any parameter value regularly on the beat.

As we saw in the last section, you may still want to use the *PluggoSync* plug-in to respond to certain elements of an audio track even if your sequencer does support host sync.

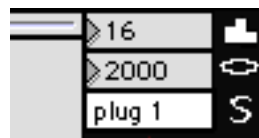
Synchronization using the *PluggoSync* plug-in requires that you create a “click track” that feeds the *PluggoSync* plug-in. The synchronized plug-ins then “listen” to one of *PluggoSync*’s synchronization sources.

- Open the edit window of *PluggoSync*.



PluggoSync has a number of features we aren’t using in our example document. The Plug mode of *Audio Rate Pan* uses the basic beat information generated from *PluggoSync* when it is set to Audio Sync mode. (The other mode, Internal BPM, is discussed below.)

At the bottom of the *PluggoSync* edit window, you’ll see a Beat Divider feature that allows you to subdivide the basic beat information being generated. For example, a division of 8 means that 8 “sync pulses” are generated for every beat. Certain plug-ins, such as *Synth*, can be set to listen to one of the four beat division selections as shown below where it’s been set to listen to *PluggoSync* output 1.



This feature is not used with *Audio Rate Pan*, which has its own note value and multiplier specifications that are somewhat more powerful than the ones inside *PluggoSync*.

Feeding Audio to PluggoSync

To add synchronization capabilities to an existing song, you need to add a click track that feeds the *PluggoSync* plug-in. You do this in two steps:

- Import the audio file *sync.aiff* found in the Pluggo Stuff folder onto an audio track, and make copies of it so that it plays once every bar of the sequence.
- Next, insert the *PluggoSync* plug-in and send it the audio from the click track.

Note that if you don't want to hear the click track in your mix, you have several options:

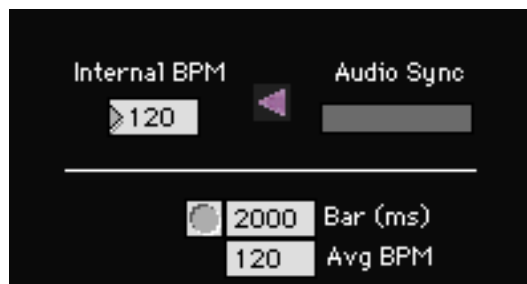
- In Cubase 4.0, Vision, Logic Audio, and Digital Performer, you can use *PluggoSync* as an Insert Effect. Since *PluggoSync* will not echo its audio input to its audio output unless you have the Thru toggle checked in its edit window, the click track is silenced.
- In any version of Cubase, you can use a Send Effect and enable the Pre (for pre-fader send) button for the effect send. Then turn down the channel fader for the click track. The clicking sample is sent only to the plug-in, not the mix.



Check the meter below the words Audio Sync in the *PluggoSync* edit window to make sure *PluggoSync* is receiving the audio, and you will see it “lock up” to the click track.

The more devious among you may have already noticed that you can use any audio signal as a click track, but we don't guarantee the results.

- To set *PluggoSync* to run on its own internal clock, ignoring its audio input, click on the purple triangular button so it points to Internal BPM. Then you can set a tempo with the number box.



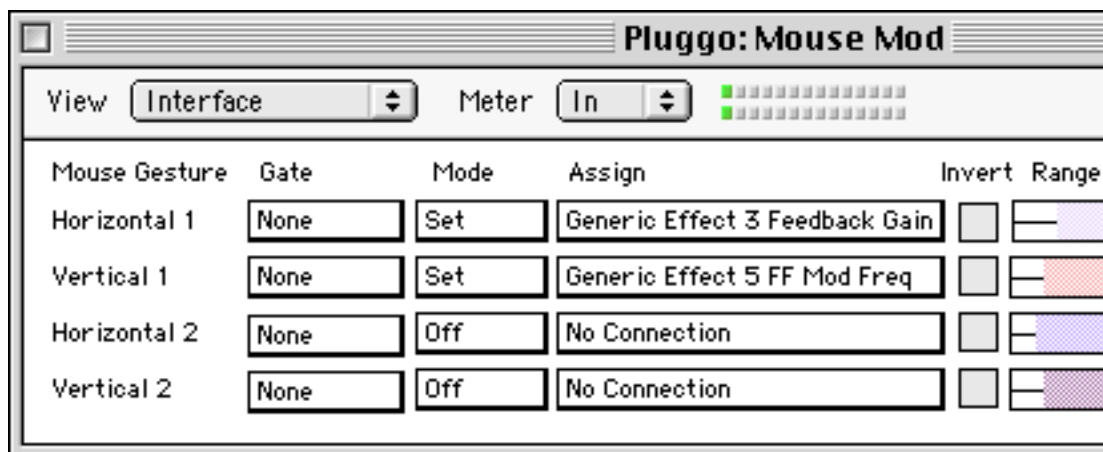
One use of the internal clock is to keep your plug-ins in sync with each other without a click track. *PluggoSync*'s internal clock will simply run freely at the designated BPM and it will run whether or not your sequencer is playing a song. In other words, *PluggoSync* is functioning as a sequencer inside your sequencer.

Using Modulator Plug-ins

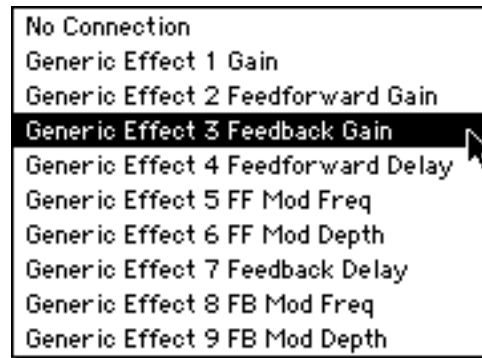
pluggo's Modulators are plug-ins that do not process any audio. Instead they are designed to modify the parameters of other plug-ins. **pluggo** provides a number of different Modulators. We're going to look at one of the simpler ones called *Mouse Mod*. It lets you move the cursor around the screen (without clicking) to change up to four different parameters at once. Since you can have several *Mouse Mod* plug-ins operating at the same time, even more parameters can be changed if you like this sort of thing. Other Modulator plug-ins, such as *LFO*, generate automatic control signals, allowing modulation effects to be accomplished without the danger of repetitive strain injury. *LFO* can also be synchronized to *PluggoSync* so the repeating changes can be tied to the metric structure of your music.

Modulator plug-ins pass audio signals through them; indeed, most ignore their audio inputs completely. So you can insert them anywhere without affecting the signal path of your mixer.

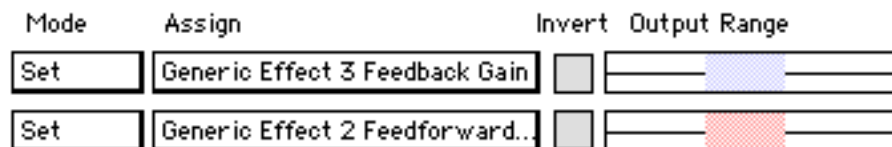
- Set a channel in your sequencer's mixer to handle an audio signal and insert both the *Mouse Mod* and *Generic Effect* plug-ins on that channel. You can put *Mouse Mod* before or after *Generic Effect* in the effect chain, it doesn't matter.
- Open the *Mouse Mod* edit window.



- Click on the top menu (corresponding to Horizontal 1) in the Assign column, and choose Generic Effect 3 Feedback Gain. Then choose Set from the Mode menu immediately to its left. This assigns the horizontal position of the mouse to modify *Generic Effect's* feedback gain coefficient.



- Click on the menu corresponding to Vertical 1 in the Assign column, and choose Generic Effect 2 Feedforward Gain. Then choose Set from the Mode menu immediately to its left. This assigns the vertical position of the mouse to modify *Generic Effect's* feedforward gain coefficient.
- Open the *Generic Effect* plug-in's edit window. You should see the Feedforward and FB Mod Depth parameters moving around as you move the mouse on the screen, and you should also hear a change in the audible result. Experiment with different Output Ranges in the *Mouse Mod* window (just click to drag out a range); this will affect the ratio of mouse position to parameter change. For these two gain settings, a value that's too high will cause distortion, so it's best to limit the Output Range to the middle of the total range, like this:



Next, we'll apply *Mouse Mod* to modulate the settings of a VST plug-in that isn't included with **pluggo**.

Using Other VST Plug-ins within Pluggo

pluggo has the ability to open and host VST plug-ins, including those that were not written using Max/MSP. This feature works differently depending on whether you are using a VST or MAS host environment. Please refer to the section below that is appropriate for your environment.

Using Other VST Plug-ins within MAS

If you're using an MAS host, **pluggo** makes it extremely easy to use other VST plug-ins and Max/MSP patches. Simply drop the plug-ins and patches into the VstPlugIns folder contained within your MAS host's application folder, and restart the application. In Digital Performer, you'll see the plug-ins listed in the pop-up menu of effects in the Mixing Board window.

One obvious benefit of **pluggo** for MAS users is that it serves as a way to use VST plug-ins within MAS applications. Another benefit is that non-**pluggo** VST plug-ins can have their parameters modulated by **pluggo** Modulator plug-ins.

In addition, **pluggo** adds its interface enhancements to the VST plug-ins it hosts, including undo, parameter randomization, and display of all plug-in parameters as egg sliders with coarse/fine control.

For more information on these features, skip to the section below entitled *Features of Hosted Plug-ins*.

pluggo cannot host MAS plug-ins, nor can it modulate the parameters of MAS plug-ins.

The Pluggo plug-in

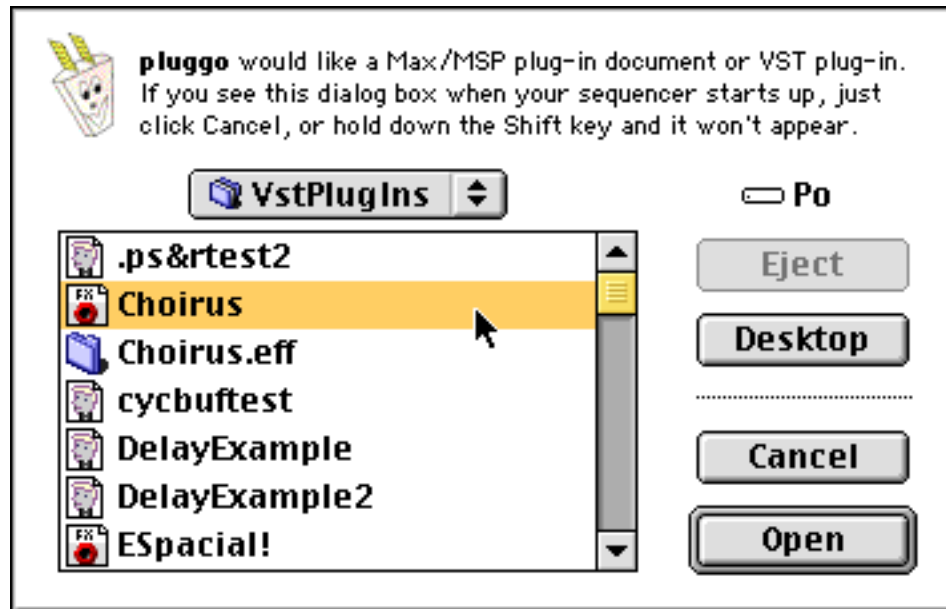
The *Pluggo* plug-in has the ability to open and host other VST plug-ins. You can use it to open VST plug-ins made with Max/MSP, which is just like loading them from a menu in the host sequencer. You can also use it to open Max/MSP documents directly without having to make them into VST plug-ins, a feature typically only useful to plug-in developers. Most importantly, however, you can use the *Pluggo* plug-in to open VST plug-ins that were not made with Max/MSP.

Sorry, fans of self-reference: the one VST plug-in that cannot be opened by the *Pluggo* plug-in is the *Pluggo* plug-in itself.

Why would you want to open a VST plug-in using *another* plug-in when you can just insert it directly within the sequencer? The main benefit is being able to modulate a plug-in's parameters using a **pluggo** Modulator plug-in. Other benefits include the user interface enhancements of **pluggo** plug-ins, such as undo, parameter randomization, and

display of all plug-in parameters as egg sliders with coarse/fine control. Here's how you do it:

- Insert the *Pluggo* plug-in, and when the open file dialog appears, choose the appropriate plug-in or Max/MSP document. Most of your plug-ins will probably be in your sequencer's VstPlugIns folder. But *Pluggo* allows you to open a plug-in file that's located anywhere.



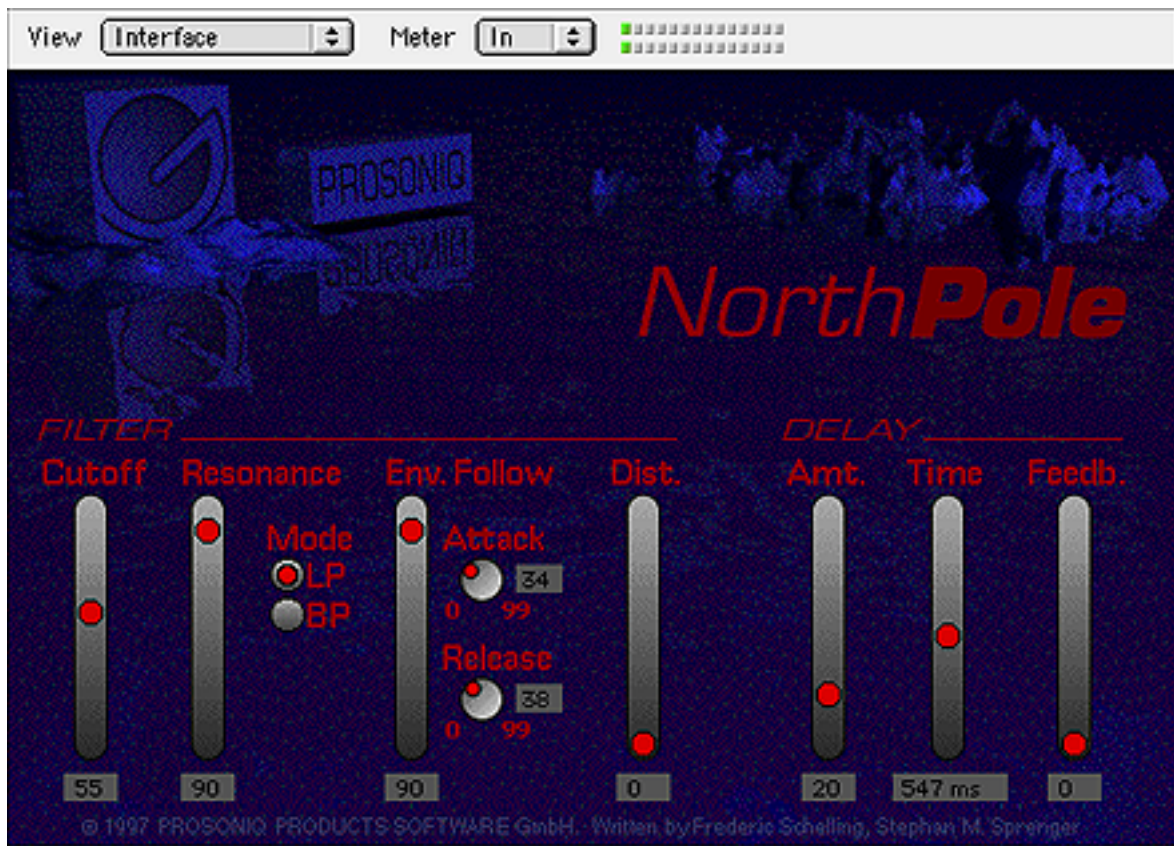
Some VST plug-ins don't work correctly if they're not in the VstPlugIns folder of the host application you're currently using. Some may require auxiliary files to be in specific locations such as a specially named subfolder of the VstPlugIns folder.

Features of Hosted Plug-ins

If a plug-in has its own edit window, this interface is displayed first when you open its **pluggo**-hosted edit window. If not, you'll see all of the plug-in's parameters displayed as sliders. For VST plug-ins with their own edit windows, you can use the **pluggo** View menu to see the parameters.

Since VST plug-ins with their own edit windows never had their parameters exposed by name before, many do not provide names for their parameters, so the names you see (P1 - P10 for example) may not be very helpful or even erroneous. Contact the publisher of your favorite plug-in and ask them to add parameter names. It will be very simple for them to do.

Here is Prosoniq's *North Pole* filter plug-in being hosted by the *Pluggo* plug-in within Cubase.



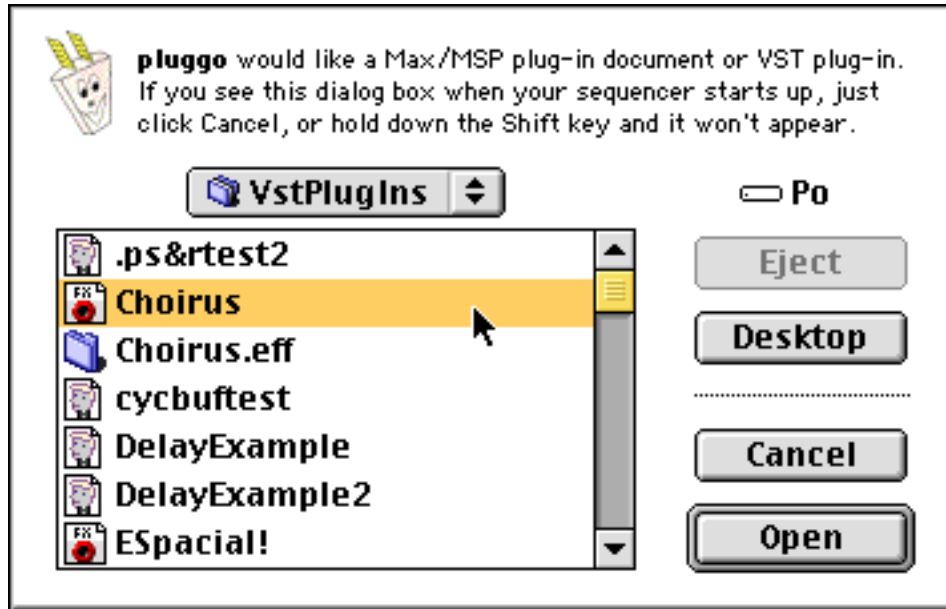
To access the undo feature for a hosted plug-in, command click on plug-in's edit window to get the Parameter Change pop-up menu and choose Undo Last Change. This menu also contains the parameter randomization commands described in the *Using the Plug-in Interface* chapter.

Modulating the Parameters of a Hosted Plug-in

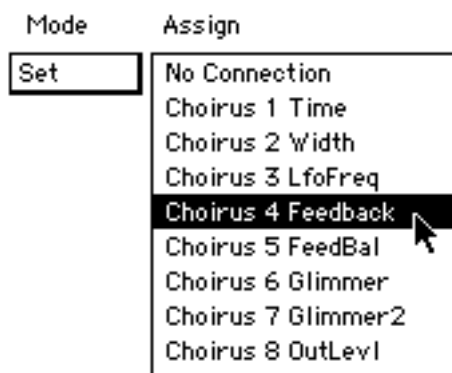
As we mentioned above, the main reason you'll want to host VST plug-ins with **pluggo** is to modulate their parameters with a Modulator plug-in. Here's an example where we will modulate parameters of a chorus-type effect with mouse movement. For Cubase, we'll use the *Chorus* plug-in, and for Vision, we'll use the *opCHORUS* plug-in. (Logic Audio and Digital Performer don't ship with any VST plug-ins, so if you're a Logic Audio or Performer user and you have some VST plug-ins, feel free to use them instead of the chorus effects discussed here.)

- If you're using Digital Performer or another MAS application, move the chorus-type VST plug-in to the `VstPlugIns` folder that's a subdirectory of the application. You must quit and restart your MAS host application to update the effects pop-up menus after making any changes to this folder.
- Insert the *Mouse Mod* plug-in somewhere if it's not already present.

- If you're using an MAS host application, simply insert the chorus-type VST plug-ins you wish to modulate by choosing its name from the pop-up menu of available effects.
- If you're using a VST host, insert the *Pluggo* plug-in, and when the open file dialog appears, insert the appropriate chorus plug-in, which you should find in your sequencer's VstPlugIns folder.



- Ensure that you have some audio going through the chorus effect.
- For *Chorus*: Open the *Mouse Mod* window, click on the top menu in the Assign column, and choose to modulate the Feedback parameter. Then choose Set from the Mode menu in the same row.



- For *opCHORUS*: Open the *Mouse Mod* window, click on the top menu in the Assign column, and choose to modulate the Feedback parameter. Then choose Set from the Mode menu in the same row.
- For another effect: Open the *Mouse Mod* window, click on the top menu in the Assign column, and choose to modulate a parameter you think will have some audible effect.

- Move the mouse so the cursor changes horizontally. You should hear an audible difference as the modulated parameter changes. If you open the edit window for *Choirus* or *opCHORUS*, you will see the Feedback slider change when the mouse moves.

Getting On the PluggoBus

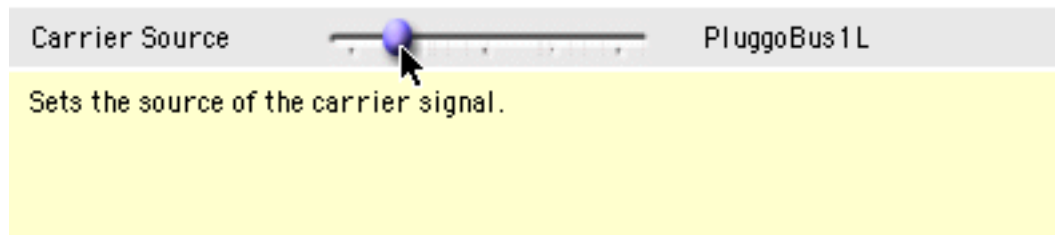
In audio mixer terminology, a bus is a signal path that can be both an input and output for different elements of the mixer. For instance, the output bus on a mixer is the sum of all the individual channels feeding into it. Other types of busses (yes, that's the way it's pluralized) combine effects or input channels, but you don't necessarily listen to them directly in a final mix.

pluggo features its own audio bus, which is called, logically enough, the *PluggoBus*. It's designed so that plug-ins made with Max/MSP can send audio signals directly to each other. Why is this necessary when most sequencers today offer flexible "virtual" bus architectures? One reason is demonstrated by **pluggo**'s *Vocoder* plug-in, which requires both a carrier source and modulator in order to function effectively. The source is often a synthesizer, while the modulator that filters it is often a vocal track. But it's also interesting to experiment with other types of sounds as sources, some of which may be generated by plug-ins, and some of which might come from audio files playing into your mixer.

In any typical mixer setup, the inputs to effect plug-ins are tied together as a stereo pair. But in this case, we want two inputs that are completely different from one another.

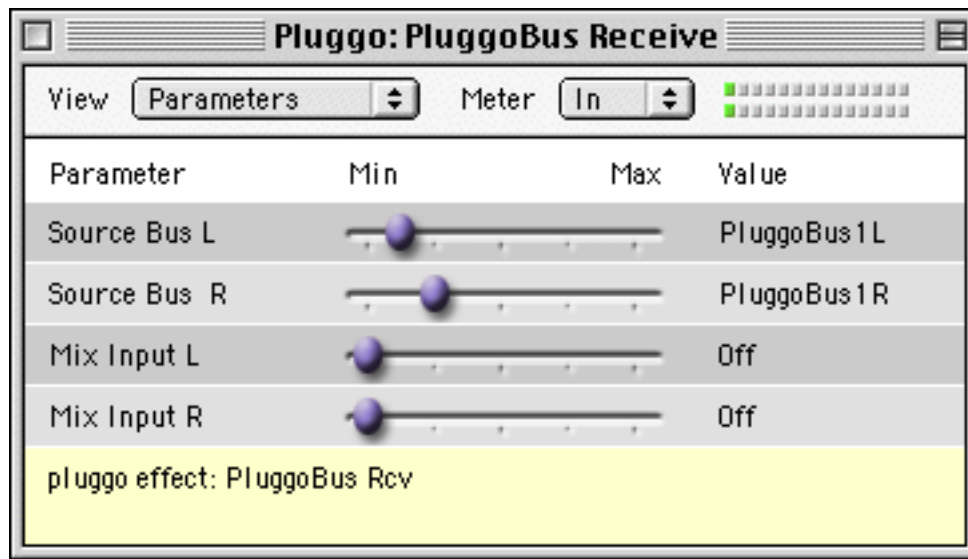
PluggoBus Receivers

Below, we've shown a slider in the *Vocoder* interface that lets you select the Carrier Source.



You can choose PluggoBus 1L through 4R as carrier sources instead of using *Vocoder*'s internal synthesizer. These assign whatever is on the bus as the carrier source for vocoding (the modulator is always the audio signal fed into the input of the *Vocoder* plug-in).

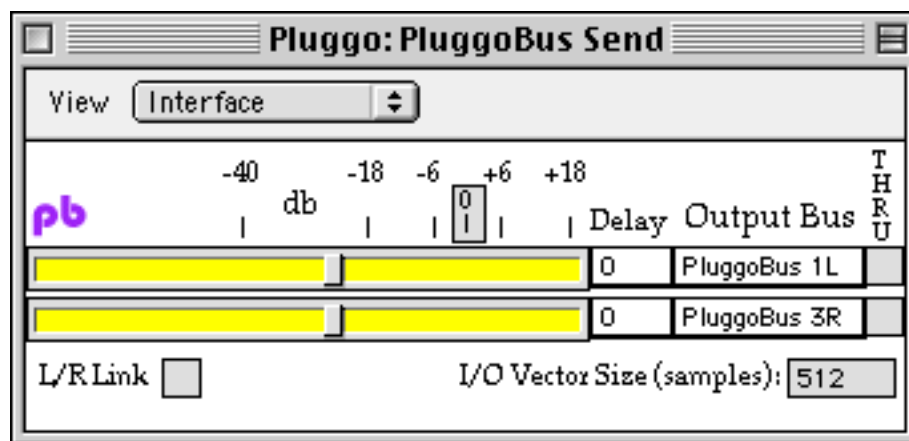
In addition to specific plug-ins that accept information from the bus, there is also a *PluggoBus Rcv* plug-in. This can be used to feed an effect inserted below it with the contents of the bus.



The Source Bus L and Source Bus R choose what you want to send to the plug-in's output. You can also mix a desired level of the plug-in's input with what's on the bus with the Mix Input sliders.

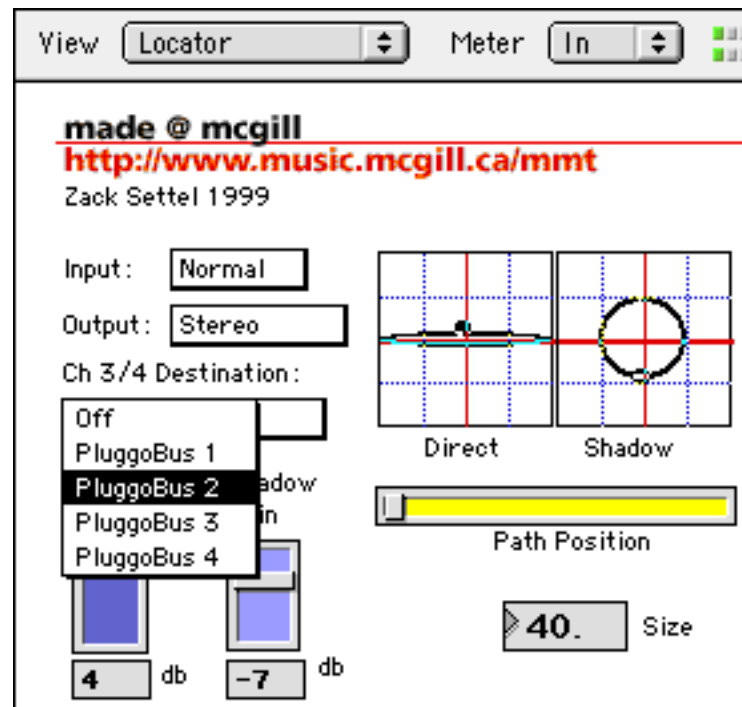
PluggoBus Senders

Now we've seen how to get things off of the bus, let's show you how you can put things on the bus. The *PluggoBus Send* plug-in takes its audio input signal and assigns it to one or two PluggoBus outputs.



The large yellow sliders set the input level fed to the bus. The input can also be delayed by a multiple of the number of samples the sequencer's mixer processes at one time. This allows you to solve synchronization problems that arise when you try to send audio from a plug-in that comes later in the mixer's processing chain to one that comes earlier.

Other PluggoBus sources will include plug-ins that generate more than two output channels. For instance, the *Dr. Dop* plug-in can create a four-channel spatialization of its input. You can send its outputs to the plug-in's outputs of course, but that's only two channels. The other channels (3 and 4) are obvious candidates for the PluggoBus.



Plug-in Automation

Some sequencers allow you to capture changes you make to **pluggo** plug-ins while the music is playing back, then replay these changes in time with the music. The most common application of this feature is *automated mixdown*—meaning you essentially get more hands to move mixing board controls while the music plays back. Here we'll show you how to use the automation features of Cubase with **pluggo**.

For more information on automation, refer to the manual of your host application. Our discussion of the topic here is not intended to replace the host application's documentation of its automation features.

pluggo 2.0 does not support the automation features in the forthcoming 2.7 version of Digital Performer; however, we anticipate releasing an update to **pluggo** that will support automation.

Vision and Studio Vision 4.5.1 support plug-in automation, but we have been unable to get them to automate **pluggo** reliably at the time this manual was printed. We therefore decided to omit information about using plug-in automation in these programs.

Please check <http://www.pluggo.com> for updates and more current information on plug-in automation.

Automation in Cubase

You can use the Write function of the VST Channel Mixer to capture changes you make to **pluggo** control panels—as well as volume, panning, muting and soloing in the VST mixer itself—into a special Audio Mix part. You can add additional changes in subsequent “passes” to this Audio Mix part, and undo any passes you're not happy with. When you play back the Audio Mix part, all captured fader movements or button pressing will be repeated just as you performed them, complete with exciting dancing on-screen faders.

To Record Your Actions:

- Open the VST Channel Mixer.
- Using the Windows menu, open the effects panel corresponding the effects you want to automate (either Send Effects or Insert Effects).
- Click on the Edit button corresponding to the plug-in you want to automate.



- Click on the Write button in the upper left corner of the Channel Mixer.



- Start the music playing. Every change you make on the Channel Mixer or a plug-in control panel will be recorded when the Write button is active.

When the Write button is on, you can also stop playback on the transport control. Any changes you make to your mixer parameters when stopped are recorded at the current Song Position. You can use this to add exciting abrupt changes in your mix, pause to bring up new plug-in edit windows, or to set initial settings for your plug-ins or Mixer Channels.

If you check the Arrange window, you will see that clicking on the Write button creates a new Mixer Track called Audio Mix.



This track will contain a record of all your actions. Each time you click the Write button while playing the sequence back, another pass is added to the Audio Mix part containing the new changes.

Since you can create an automated mixdown by layering, you'll find it easiest to work with a single plug-in or a single channel at a time. If you turn off the Write button at the end of your mixdown pass, you can undo your most recent pass if you don't like it by choosing Undo from the Edit menu.

- Stop playback.
- Click on the Write button in the Channel Mixer, disabling recording of your actions.

To Listen to the Actions You've Recorded:

- Click on the Read button in the Channel Mixer.



You can turn on the Write and Read buttons at the same time. You would do this if you wanted to listen to the actions you've already recorded while you are working with a new Channel or plug-in during another record pass.

- Start playback. The faders and controls will move automatically, following your recorded actions.

If you aren't happy with what you just recorded, you can undo your pass by choosing Undo from the Edit menu. Repeat as necessary.

Automation in Logic Audio

Logic Audio Version 3.7 or later allows automation of console changes and plug-in parameters. Generally, this is done by recording using the A-Playback channel (displayed between the Audio and MIDI channels in the song window).

Although Logic Audio can record any changes you make to a plug-in, it does not store the initial position of a plug-in's controls when you start recording. To overcome this limitation, you can use the Touch Parameters command in the **pluggo** plug-in Parameter Change pop-up menu. Touch Parameters sends out the plug-in's current values to the host sequencer. This allows you to start your automation sequence from a known state.

To Record Your Actions:

- Make sure the plug-in's edit window is open by double-clicking on the name of the plug-in in the Environment (audio mixer) window.



- In the Arrange window, Record enable the A-Playback track by selecting it.



- Click and hold the Record button until you see the Record menu. Choose Record Toggle from the menu. (There is also a Record Toggle key command you can use.) This will drop you into record without having the transport start moving.
- Hold down the command key and click in the plug-in's edit window to see the Parameter Change pop-up menu. Choose Touch Parameters. All of the current settings for this plug-in will be written at the beginning of the track you're recording.



- Click the Record button and hold it down, select Record Toggle again from the menu (or use the Record Toggle key command). This will take you out of record. You will see a block displayed in the track display window at the beginning of the A-Playback track.



- Your plug-in will now revert to its initial state whenever you record subsequent passes during your mix. If you're working with a plug-in that uses delay lines, you may have to pause between passes to let the delay lines empty out.

If you're planning on automating changes to a lot of plug-ins, you may want to use Logic Audio's event editing to create a single large block of data that initializes all the plug-ins you want to work with.

- Click the Record button to begin recording. Move sliders or other controls in the plug-in's edit window as needed. Every change you make in the audio mixer or a plug-in control panel will be recorded.

For finer control of the sliders, you can hold down the option key while you're moving the fader. Note however that Logic Audio only records your changes with a resolution of 127 steps.

- When you have finished recording, click the Stop button twice to stop recording and rewind to the beginning. You will see the changes you added displayed in the track display window. This new block contains a record of all your actions during the last mixdown pass.



To Listen to the Actions You've Recorded:

- Click the Play button to start playback. The sliders in the plug-in edit window should move automatically, following your recorded actions. If you are unsatisfied with your mixdown pass, you can delete it and try again. Remember that the first block of data shown for the A-Playback track is the initialization data for your plug-in and probably shouldn't be deleted.

Index

- alias, 11
- A-Playback, 79
- Audio Instruments window, 21
- audio sync, 64
- authorization code, 13, 14, 15
- Authorize button, 15
- authorizing
 - entering a different code, 16
 - problems, 15
- automation, 76
 - Logic Audio, 78
 - Touch Parameters, 41
- beat divider, 63
- breakpoint envelope editor, 45
- bus, 73
- bus effects, 21
- Carrier Source, 73
- Choirus* plug-in, 70
- click track, 61, 64
- coarse mode, 40
- command key, 41, 79
- Copy All from Program, 37, 42
- Copy ID to Clipboard, 14
- Create Set, 51
- Cubase, 18, 52, 61, 64, 70
 - assigning VST instrument, 31
 - Audio Mix part and automation, 76
 - automated mixdown, 76
 - changing effect programs, 33
 - effect programs, 40
 - Effects Master knob, 20
 - insert effects, 21
 - inserting mono plug-ins, 20
 - inserting stereo plug-ins, 20
 - master effects, 20
 - nested plug-in folders, 52
 - plug-in edit window, 34
 - Read button and automated mixdown, 78
 - synchronization, 55
 - VST channel mixer, 76
 - VST instruments, 28
 - VST instruments, 28
 - Write button and automated mixdown, 77
- Custom Install, 8
- demo mode, 13
- Digital Performer
 - automated mixdown, 76
 - changing effect programs, 39
 - creating a stereo audio channel, 25
 - effect programs, 40
 - hosting MAS plug-ins, 68
 - hosting other VST plug-ins, 68
 - inserting plug-ins, 25
 - plug-in edit window, 37
 - saving effect programs, 39, 40
 - synchronization, 55
- Disable Hints, 46
- Dr. Dop* plug-in, 75
- Easy Install, 8
- edit window, 34, 69
- effect programs, 5, 33, 36, 40
 - copying, 42
- effect sends, 21
- Effects Master knob, 20
- Enable Hints, 46
- Evolve All, 41
- filter graph, 45
- fine mode, 40
- free synchronization mode, 55, 58
- gain slider, 43
- Generic Effect*, 33, 66, 67
- Go to Web Page button, 14
- hints
 - enabling and disabling, 46
- host synchronization mode, 55, 56
- hosting MAS plug-ins, 68
- insert effect, 17, 21, 64
- Install a Few Plug-ins, 9, 11
- Install All Plug-ins, 9, 11
- installation, 7
- Installer for Max/MSP Users*, 9
- Interface view, 42
- internal clock, 65
- Jaz drive, 15
- level meter, 46

- LFO*, 66
- license agreement, 7
- Logic Audio, 54, 78
 - assigning VST instruments, 32
 - automated mixdown, 76
 - automated mixdown playback, 80
 - automated mixdown recording, 79
 - bus channel, 24
 - bus effect, 25
 - changing effect programs, 37
 - generic plug-in interface, 37
 - insert effect, 24
 - inserting stereo plug-ins, 23
 - mixer channel, 24
 - nested plug-in folders, 54
 - plug-in edit window, 36
 - Touch Parameters command, 78
 - VST instruments, 28
 - VST instruments, 31
- Manage submenu, 51
- MAS
 - hosting MAS plug-ins, 68
 - hosting other VST plug-ins, 68
 - Pluggo plug-in, 18
 - plug-in folders, 47
- master effects, 17, 20
- Max, 5, 6
- Max Audio Library for Plugins*, 9, 12
- Max folder, 11
- MaxPlugLib*, 9, 12
- Messages view, 42
- MIDI
 - assigning VST instruments, 31
 - VST instruments, 28
- mixdown automation, 76
- modulators, 66, 70
 - and audio input, 66
- mono plug-ins, 19
- Mouse Mod*, 66, 70
 - Assign menu, 67
 - Mode menu, 67
 - output range, 67
- MSP, 5, 6
- multislider, 44
- My Plug-in Set, 51
- New Folder, 51
- North Pole* filter plug-in, 70
- note-unit multiplier, 58
- note-unit pop-up menu, 57
- number box, 43
 - changing parameters, 40
- opCHORUS* plug-in, 70
- opening a sequencer document, 19
- packaged version, 14
- Parameter Change pop-up menu, 41, 46
- parameter sliders
 - coarse mode, 40
 - fine mode, 40
- parameters
 - changing, 39
 - changing numerically, 40
 - modulating, 66, 70
 - nudging, 42
 - pop-up menus, 40
 - randomizing, 41
 - undoing changes, 41
- Parameters view, 42
- Performer. *See* Digital Performer
- plug synchronization mode, 56, 60
- Pluggo Dictionary* file, 19
- Pluggo FAQ, 6
- Pluggo Info view, 42
- Pluggo Installer, 7
 - for Max/MSP Users, 7
 - Regular, 7
- Pluggo plug-in, 19, 68, 70
 - hosting VST plug-ins, 70
 - in MAS, 18
 - loading MSP patches, 69
 - loading non-pluggo plug-ins, 69
 - modulating non-pluggo plug-ins, 68
- Pluggo Plug-in Reference Guide*, 5
- Pluggo Plug-in Reference Guide*, 20
- Pluggo Stuff, 7, 9, 11, 64
- PluggoBus*, 73
 - input level, 74
 - sources, 75
 - synchronization, 75
- PluggoBus Rcv* plug-in, 74
- PluggoBus Send* plug-in, 74

- PluggoSync*, 55, 56, 60, 62, 63, 64, 65
- plug-in folder, 49, 51
- Plug-in Manager, 47
 - disabling plug-ins in a folder, 53
 - nested plug-in folders, 52
 - nested plug-in folders in Logic Audio, 54
 - preferences files, 54
 - viewing plug-in folders, 53
- plug-in sets
 - naming, 51
 - recalling, 51
 - saving, 51
- plug-ins
 - automating, 76
 - diagnostic messages, 42
 - edit window, 33
 - enabling and disabling, 49
 - getting info about, 42, 50
 - inserting mono, 23, 26
 - inserting, 17
 - inserting stereo, 20, 23, 26
 - modulation of, 66
 - mono and stereo, 19
 - version information, 50
- pop-up menus
 - in Interface view, 45
 - parameter change, 41
- pre-fader send, 64
- purchasing
 - downloadable version, 14
 - packaged version, 14
- Randomize All, 41
- range slider, 44
- Registration ID, 14
- repetitive strain injury, 66
- saving plug-in sets, 51
- send effects, 17, 64
- startup, 19
- stereo plug-ins, 19
- Studio Vision Pro. *See* Vision
- sync mode menu, 57
- sync.aiff audio file, 64
- synchronization, 55
 - of pluggoBus audio, 75
- synchronization modes, 55, 56, 57, 58, 59, 60
- system requirements, 6
- technical support, 6
- Tempo/Sync parameter, 57
- toggle box, 44
- Touch Parameters, 41, 79
- UDT synchronization mode, 56, 59
- undo, 41
- Undo Last Change, 41
- user interface elements, 43
- user-defined tempo, 56, 59
- vertical slider, 43
- View pop-up menu, 42, 69
- Vision, 70
 - automated mixdown, 76
 - changing effect programs, 36
 - effect programs, 40
 - inserting mono plug-ins, 23
 - inserting stereo plug-ins, 21
 - plug-in edit window, 35
- Vocoder* plug-in, 73
- VST instruments
 - assigning MIDI channels, 31
 - using, 28, 31
- VstPlugIns (disabled), 47
- VstPlugIns folder, 7, 18, 19, 47, 69, 71
- Zip drive, 15