

Getting Started with

pluggo



# Table of Contents

Copyright and Trademark Notices .....	4
Introduction .....	5
About This Manual .....	5
Basic Features of Pluggo.....	5
System Requirements .....	6
Windows .....	6
Macintosh .....	6
Where to Get Help and Support.....	6
A Little Background Information on Pluggo.....	7
Installing Pluggo .....	8
What is Going to be Installed? .....	8
Windows Systems .....	8
Macintosh Systems.....	8
The Pluggo Installer .....	9
Authorizing Pluggo .....	9
Challenge/Response Authorization.....	10
Authorizing an iLok .....	11
Deauthorizing and Challenge/Response .....	11
Lose Your Response?.....	11
Using Pluggo.....	12
Inserting Plug-ins Within Your Sequencer .....	12
Using Virtual Instruments .....	13
Using Virtual Instruments in Max/MSP.....	13
Plug-in Automation .....	14
Using the Generic Plug-in Interface .....	14
Changing Parameters .....	15
The Parameter Change Pop-up Menu .....	16
Using the View Menu.....	17
Elements of Interface Views .....	18
The Number Box.....	19
The Horizontal or Vertical Slider.....	19
The Vertical Gain Slider .....	19
The Multislider .....	19
The Range Slider .....	19
The Toggle Box .....	19
The Pop-up Menu .....	19
The Breakpoint Envelope Editor .....	19
The Filter Graph.....	20
Interface Hints and Labels.....	20

The Level Meter .....	20
Synchronizing Plug-ins .....	22
Opening the Sync Example Document .....	22
Exploring the Synchronization Modes .....	22
Host Mode .....	23
Free Mode .....	25
UDT Mode .....	25
Plug Mode .....	26
Using PluggoSync .....	28
Feeding Audio to PluggoSync .....	29
Using Modulator Plug-ins .....	31
Getting On the PluggoBus .....	33
PluggoBus Receivers .....	33
PluggoBus Senders .....	34
Index .....	35

# Copyright and Trademark Notices

**pluggo** and this manual are © 2005 Cycling '74. **pluggo** uses Max/MSP, published by Cycling '74. Max is copyright © 1990-2005 Cycling '74 and IRCAM, l'Institut de Recherche et Coordination Acoustique/Musique. MSP is copyright © 1997-2005 Cycling '74—All rights reserved. Portions of MSP are based on Pd by Miller Puckette, © 1997-2005 The Regents of the University of California. MSP and Pd are based on ideas in FTS, an advanced DSP platform © IRCAM.

**pluggo** and MSP are trademarks of Cycling '74. VST and Cubase are trademarks of Steinberg Soft- und Hardware GmbH. Digital Performer, Performer, and Mark of the Unicorn are trademarks of Mark of the Unicorn, Inc. Peak and Vbox are trademarks of BIAS, Inc. Digidesign and Pro Tools are trademarks of Digidesign, a division of Avid Technology, Inc. Ableton and Live are trademarks of Ableton AG.

# 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** plug-ins, synchronization, **pluggo** Modulators and the PluggoBus.

Do you need to read this entire manual? We definitely recommend reading through the description of the installation, authorization and registration processes. 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 program you'll be using to host **pluggo** plug-ins. We do mention a few things that are specific to using **pluggo** with applications such as Cubase, Pro Tools, Digital Performer, Logic Audio, Live, Peak and Spark, 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 Applications folder on the Applications folder on the hard disk. 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 places plug-in development materials in the Pluggo Applications folder.

## Basic Features of Pluggo

- A lot of plug-ins (over 100 of them)
- The ability to open any plug-in made with the plug-in development tools that are part of Max/MSP
- Support for host synchronization in VST, RTAS, and Audio Units applications. With other hosts, you can use the *PluggoSync* feature for synchronization.
- Plug-ins that can send audio and/or control information to each other
- Modulator plug-ins that control the parameters of other plug-ins
- Instrument plug-ins that respond to sequenced and live MIDI input

## System Requirements

### Windows

**pluggo** requires:

- A Windows machine equipped with a 1 Ghz P3 or better (a 2.0 Ghz P4 or equivalent is recommended). An ASIO-compatible sound card is recommended for optimum audio performance.
- Windows XP (Home or Pro)
- A sequencer application that hosts VST or RTAS plug-ins.
- At least 256 MB of memory, although the exact amount depends on the system version and the host sequencer you're using.
- At least 55MB of hard disk space

### Macintosh

**pluggo** requires:

- A Macintosh equipped with a G3 processor running at 500 Mhz or faster is recommended.
- Mac OS X 10.2 or later
- A sequencer application that hosts VST, RTAS, or Audio Units plug-ins.
- At least 256 MB of memory, although the exact amount depends on the system version and the host sequencer you're using.
- At least 55MB 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/questionspl.html>
- The Pluggo troubleshooting guide, on-line at  
<http://www.cycling74.com/products/pluggo/troubleshooting.html>
- e-mail to [support@cycling74.com](mailto:support@cycling74.com)
- You may telephone our customer support line at (415) 869-2812
- If you have any questions related to installation and authorization of the software, you may telephone our office at (415) 974-1818, generally open from 10AM to 4PM Pacific time.

E-mail or telephone 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](http://www.cycling74.com/register).

## A Little Background Information on Pluggo

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

The authors of Max and MSP are Miller Puckette and David Zicarelli. The **pluggo** plug-ins and objects were written in Max/MSP by jhno, Adam Schabtach, David Zicarelli, Leslie Stuck, Joshua Kit Clayton, Marc Sirguy, Darwin Grosse, and Gregory Taylor. Additional MSP objects for **pluggo** were written by Richard Dudas, who also drew the **pluggo** character. Macintosh RTAS support was by Adam Schabtach. **pluggo** user interface design by Lilli Wessling Hart and jhno. The Windows version of **pluggo** was written by Rob Sussman and David Zicarelli, and the Windows plug-in versions were done by Richard Dudas and Andrew Pask. This manual was written by David Zicarelli and Gregory Taylor, with contributions from jhno, Darwin Grosse, and Andrew Pask, 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.

## What is Going to be Installed?

### Windows Systems

If you are installing **pluggo** on a Windows system, the Pluggo Installer will ask you to select a destination where you want your plug-ins to be installed. When you select a destination (such as *C:\Program Files\Steinberg\vstplugins*, which is common), the plug-ins will be installed in a subfolder named “Pluggo” at that destination.

**pluggo** places support files in the *C:\Program Files\Common Files\Cycling '74* folder. This folder and its subfolders should not be moved or renamed.

The Authorize Pluggo application is installed into the *C:\Program Files\Cycling '74* folder. In addition, documentation and plug-in development information is placed in the Pluggo Applications folder.

If you're installing the RTAS version of **pluggo** on a Windows system, a “stub” plug-in called Pluggo-RTAS is installed here:

*C:\Program Files\Common Files\Digidesign\DAE\Plug-Ins*

**Note:** If you move the **pluggo** plug-ins from their original install location, **pluggo** will no longer work in RTAS. If you want to change the location where your plug-ins are installed, you will need to uninstall **pluggo** and then re-install at the new location.

### Macintosh Systems

If you are installing **pluggo** on a Macintosh, the Pluggo Installer will put the plug-ins in the *Library/Audio/Plug-ins/VST/Pluggo* folder. **pluggo** places support files in *Library/Application Support/Cycling '74* and */Library/CFMSupport*. None of these folders should be moved or renamed.

Audio Unit support is added to the *Library/Audio/Plug-Ins/Components* folder.

The Authorize Pluggo, Pluggo and AU Scan applications are installed into the *Pluggo Applications* folder inside the Applications folder. In addition, documentation and plug-in development information is placed in the *Pluggo Applications* folder.

If you're installing the RTAS version of **pluggo** on a Macintosh, a “stub” plug-in called Pluggo-RTAS is installed here:

*Library/Application Support/Digidesign/Plug-ins*

## The Pluggo Installer

When you run the 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.

If you click Install, the plug-ins and support files will be installed in the folder containing the application you choose.

- Choose the application and click the Choose or Open buttons. If you click Cancel, the installation will be stopped, and you will need to quit the installer and relaunch it if you want to install again. After you choose the application, the plug-ins and supporting files will be installed.

After the Pluggo Installer runs, it launches an installer for the copy protection. You'll be asked for your administrative password, because the copy protection installs files into the System folder. These copy protection files will not interfere with the operation of your system in any way.

If you've already installed the copy protection once (InterLok 5 extensions), you don't need to do so again. Just cancel when asked for your password, and click OK at the next dialog.

## Authorizing Pluggo

When you install **pluggo**, it's in demo mode. **pluggo** will operate in a fully functional demo mode after you first install it, occasionally producing an unobtrusive sound rather than the audio output you expect. If you decide that you'd like to buy a copy of **pluggo** after trying it out, you can purchase the software and obtain what is known as a *Challenge Response authorization* from Cycling '74. The Challenge/Response authorization process involves the exchange of nonsense phrases: you provide us with a "challenge" phase displayed by the software when you start it up, and we generate a "response" that you type into the dialog box displaying the challenge. After you do this successfully your software will be authorized.

Response phrases are machine-unique. In other words, after you authorize one computer, you can't copy your software to another computer and expect it to be authorized. Normally we'll respond to two challenges per customer, but with an adequate explanation of the circumstances, we'll be happy to respond to additional challenges.

If you purchased the packaged version of **pluggo**, your serial number, authorization code and Registration ID are inside the **pluggo** box. If you purchased a **pluggo** upgrade package, your information can be found on a sticker inside the manual cover. Skip to the Entering Your Code section below.

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

## Challenge/Response Authorization

- Launch the Authorize Pluggo application. On Windows systems, the Authorize Pluggo application is located in the *C:\Program Files\Cycling '74* folder. On Macintosh system, the Authorize Pluggo application is located in the *Pluggo Applications* folder inside the Applications folder.
- The opening screen will ask you whether you want to purchase a **pluggo** authorization code, or use an authorization code you have received to authorize **pluggo**. When you select the option to purchase an authorization code and click Next, you will see a screen with a Registration ID.
- If you click on the option to purchase an authorization code and click the Web Page button or press the return key, the Authorize Pluggo program will start up your web browser and automatically take you to the online **pluggo** purchase form. The program will also automatically enter your Registration ID on the online purchase form. Enter the additional relevant information on the form and click to purchase an authorization code. • You'll see the Pluggo Authorization dialog. The exact words in the challenge displayed in this dialog will be different, but as an example, it might be something like LEAF TIRE CUTS CLAY JAWS BACK ECHO REST INCA.
- Click Copy Challenge to copy the challenge string to the clipboard or click Save Challenge... to create a file with the information. By default this file is called *pluggo.ilr*.
- If you used Copy Challenge, open your web browser and go the following URL:

*<http://www.cycling74.com/auth/cr/pluggo.html>*

If you used Save Challenge..., attach the file *pluggo.ilr* to an e-mail message that includes your full name and your Max/MSP serial number to *challenge@cycling74.com*. You're done: just wait for a return e-mail within a day or so from Cycling '74.

- If you are using the web challenge form, paste the challenge into the space provided for Max/MSP challenges and fill out the other information requested on the form.
- You'll receive a response via e-mail or fax within a day or so from Cycling '74. Note that it's extremely important that we receive the exact challenge string, or we will be unable to generate a proper response string.
- If you don't have immediate web access you can e-mail your challenge to *challenge@cycling74.com* or phone us at +1 (415) 974-1818 during business hours to obtain a response. Most people find the whole process to be much easier over e-mail than over the phone.
- Once you receive your response via e-mail, you'll see a file *pluggo.ilf* attached to the message. Simply double-click on this file and the software should be authorized—you shouldn't see any demo or authorization dialog.
- If the e-mail file attachment technique doesn't work for you, write us back and we'll send you the response string as text in an e-mail message. Copy the response text, launch the Authorize Pluggo application again, go to the Authorization dialog, and paste the response into the space provided below the challenge string. Click Next. You'll either see an error message is the response was incorrect, or a message saying

you've successfully authorized the software. If you were told the response is incorrect, double-check that the text you entered matches exactly what you were sent via e-mail. If you simply can't get it to work, contact us.

## **Authorizing an iLok**

Pluggo also supports authorization via a PACE iLok hardware dongle. Use this form on our web site to request an iLok authorization:

*<http://www.cycling74.com/auth/ilok/pluggo.html>*

You'll need an account at [ilok.com](http://ilok.com) in order to receive an authorization. For more information visit this page:

*<http://www.cycling74.com/support/questionsilok.html>*

## **Deauthorizing and Challenge/Response**

If you authorized your hard disk via challenge/response, don't worry about deauthorizing. It's possible (but unlikely) that you may lose your authorization if you do something like reformat your hard disk or install a new System. To reauthorize, you can either double-click your *pluggo.ilf* file again, or save your response text and paste it into the application when it's requested.

It's possible that the challenge you see when reauthorizing your hard disk may be slightly different than the one you saw originally. Don't panic: your response will probably still work. If it doesn't, contact us.

## **Lose Your Response?**

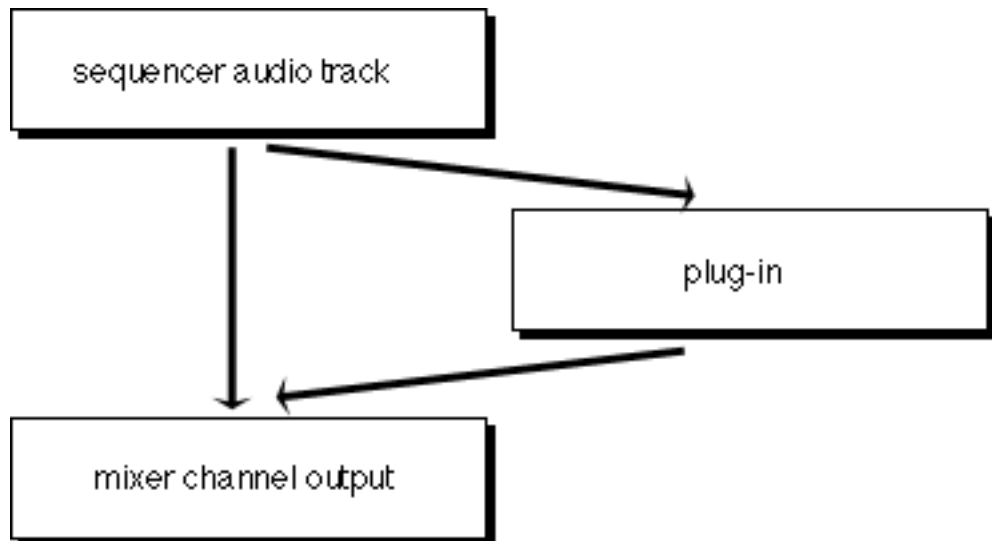
If you've lost your challenge/response authorization, contact us: we store them in our database. Or, we can just generate a new response from your challenge string.

# Using Pluggo

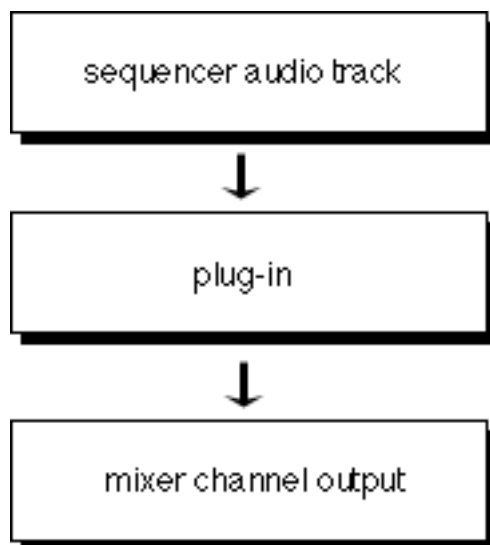
## Inserting Plug-ins Within Your Sequencer

In this chapter, we'll cover the steps for inserting **pluggo**-based plug-ins with Cubase, Digital Performer, Live, Logic Audio, Max/MSP, Peak, Pro Tools, and Spark.

When we say “inserting” a plug-in, we mean placing it into the audio processing path of a mixer. There are two kinds of insertions. In one kind of insertion, the plug-in adds its signal to the input as a kind of bus. Cubase Send Effects are an example of this kind of insertion.



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



After installing **pluggo** you will find an assortment of new plug-ins available to your host.

**pluggo**-based plug-ins can be opened in either a mono, stereo, or multi-channel 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 or multi-channel 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, stereo, or multi-channel situations (or all three).

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. For more specific details, see the documentation for your specific host application.

## Using Virtual Instruments

The **pluggo** collection includes a selection of virtual instruments—plug-ins that receive MIDI and generate only audio output.

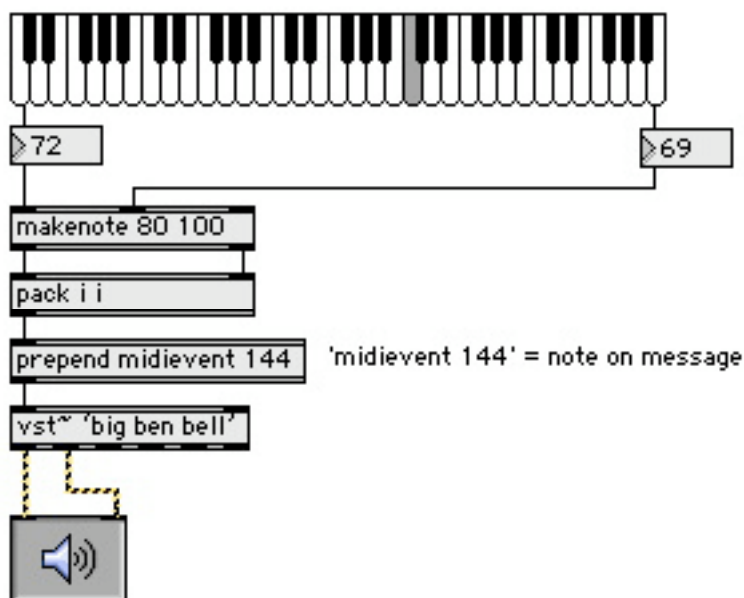
These virtual instruments are accessible to host sequencer applications in different ways which vary according to the specific host application. In Cubase, these plug-ins are called VST instruments, and have their own Instrument panel. In Logic Audio, virtual instruments are choices from a set of available Audio Instruments. Digital Performer uses MIDI instrument tracks to host virtual instruments. In Pro Tools, virtual instruments appear on the plug-in menus along with audio plug-ins. **pluggo** also includes the M2M plug-in, which lets you control **pluggo** plug-ins using MIDI input on some applications.

Typically, you'll need to create a new track for your virtual instrument, and then route MIDI data from your synthesizer or a pre-recorded MIDI track or file to the new virtual instrument. For more specific details, see the documentation for your specific host application.

## Using Virtual Instruments in Max/MSP

You can insert and use **pluggo** virtual instrument plug-ins in Cycling '74's Max/MSP environment in the same way that you use audio plug-ins. The MSP **vst~** object loads **pluggo** virtual instrument plug-ins in the same way as any other VST plug-in, and virtual instrument parameters can be changed in Max/MSP via messages to the **vst~** object. (For more general information about loading plug-ins in Max/MSP and accessing their parameters, see the section on inserting plug-ins in Max/MSP in the previous chapter of this manual.)

The `midievent` message to the `vst~` object is used to send MIDI information to a virtual instrument, as shown in the following simple Max/MSP patch.



For more information about the `midievent` message, see the `vst~` object manual page in the MSP manual.

## 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.

These sequencers let you automate effects parameter changes either by recording fader changes in real time or by inserting and graphically editing control envelopes for **pluggo** effects parameters. For more information on automation, refer to the manual of your specific host application.

## Using the Generic Plug-in Interface

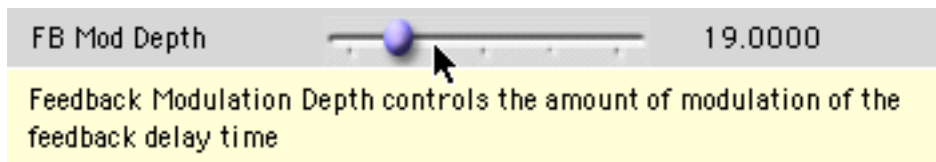
In this section, we'll be using *Generic Effect* plug-in to illustrate the features common to all, or at least most, **pluggo**-based plug-ins. This plug-in features the **pluggo** egg slider interface common to many plug-ins in the **pluggo** collection.

- 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. For more specific details about using the plug-in interface, see the documentation for your specific host application.

## 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 (Macintosh) or Alt-click (Windows) 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 (Macintosh) or Alt key (Windows) 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 numerical value does not work for certain synchronization parameters.

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.

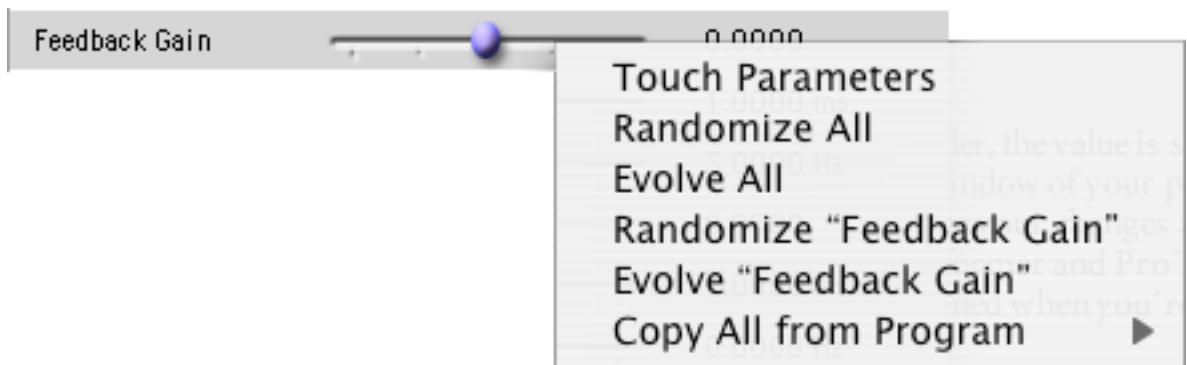
If a slider is used to select from a small number of possibilities (such as a mono/stereo select slider), you will see a pop-up menu appear instead. Use the menu to set the parameter's value.



When you change a parameter by moving a slider, the value is stored within the current effect program (displayed in the edit window of your plug-in). 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 and Pro Tools, 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 on Macintosh or the Control key on Windows 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 Digital Performer, Pro Tools 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 (Macintosh) or Control-click (Windows) 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 with 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.

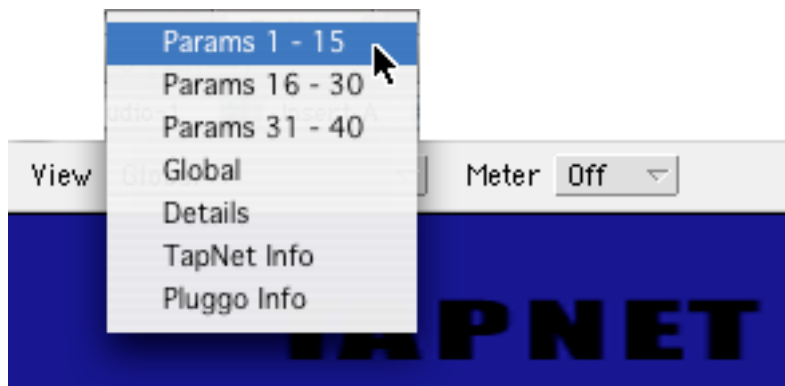
Copy All from Program lists the names of all of the programs for this plug-in. When you choose a program name from this menu, you copy the settings from the selected program into the current program. You can Undo this action if desired.

## 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 or to access a plug-in's presets.

Some plug-ins contain more parameter sliders than will fit in a single page, while others present an Interface view unique to the plug-in, possibly displaying information in a graphical form. Some plug-ins will have both a Parameters view and an Interface view (the following example shows you the TapNet plug-in, which has both kinds of interfaces—unlike Generic Effect). If a plug-in has fifteen parameters on its initial screen, the View menu will contain an item labeled Parameters 1 - 15. A subsequent page of four parameters would be labeled Parameters 16-30, and so on).

On some host applications, the View menu also may also contain a listing of the plug-in presets.



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.
- The Generic Effect plug-in provides a Messages view. This window may provide diagnostic information that could be helpful if a 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.

---

### 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.

---

### The Range Slider



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

---

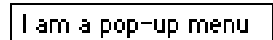
### The Toggle Box



Click on the box to turn something on or off.

---

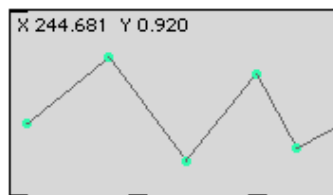
### The Pop-up Menu



Click on the pop-up menu to see a number of choices

---

### The Breakpoint Envelope Editor

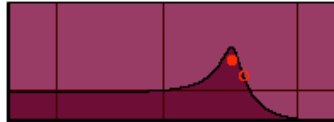


You can draw an envelope for various purposes with this box. Click on one of the existing breakpoints and dragging on it will move it to a new location. There is a display of the values as you drag the breakpoint. Some breakpoint envelope editors will allow you to add a new point by clicking anywhere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

---

---

## The Filter Graph



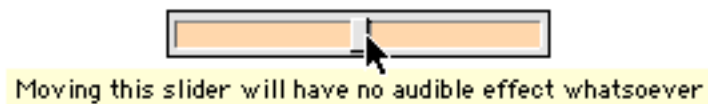
The filter graph lets you draw the frequency response of a filter in several different modes. The solid red circle is used to move the center or cutoff frequency. The hollow red circle, where applicable, determines the width of a bandpass filter. Often the filter graph will be accompanied by a set of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

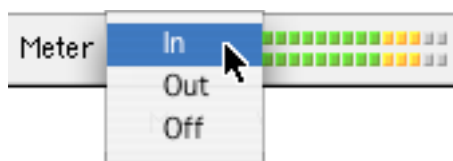


If you find the hints bothersome, Command-click (Macintosh) or Control-click (Windows) to get **pluggo**'s Parameter Change pop-up menu and choose Disable Hints (a menu item only available in an Interface view). This disables hints for all plug-ins. To re-enable hints, Command-click (Macintosh) or Control-click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

Disabling hints may eliminate extra clicks when using plug-in edit windows in Ableton Live version 2. Later versions of Live do not have this problem.

## 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.

# 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 and Audio Unit 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, Logic, Digital Performer, and Pro Tools.

- 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 Sync 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.


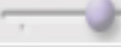
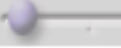



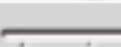


Host mode can be selected in host environments that don't support direct synchronization, but it will be the same as specifying a duration of zero.

- 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

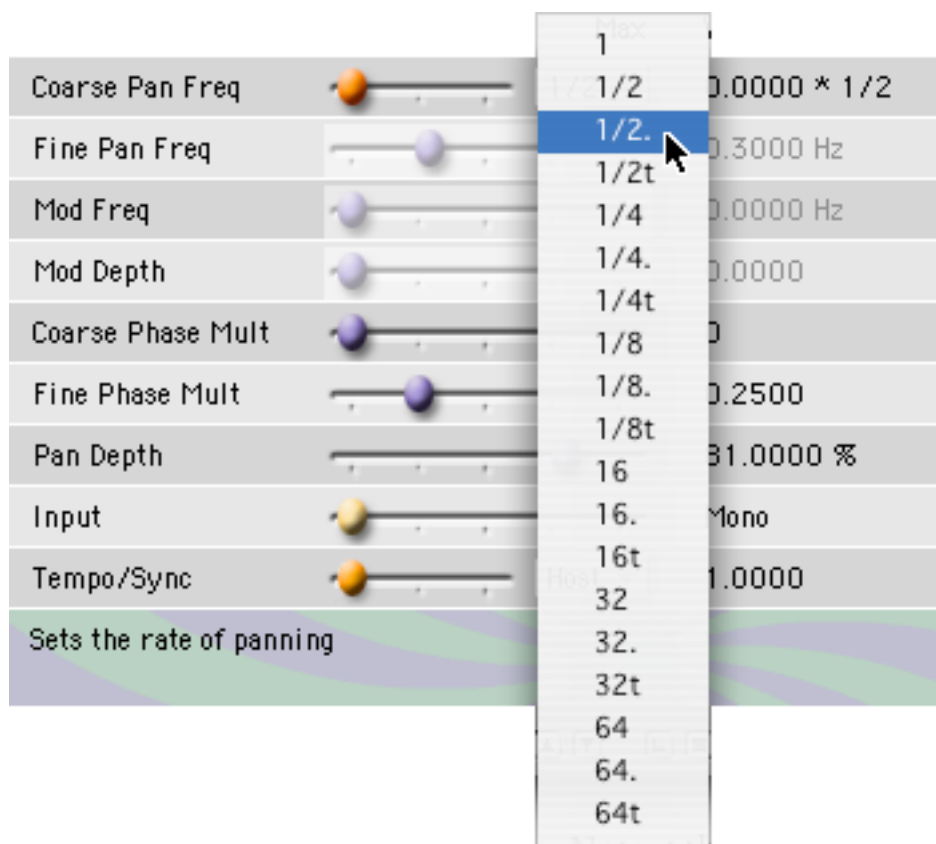
In the example documents for Cubase, Digital Performer, and Logic, Host mode is initially selected in the *Audio Rate Pan* edit window. Users of applications that do not support this capability can skip this section (or read it with envy).

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.

Parameter	Min	Max	Value
Coarse Pan Freq		1 ▾	1 * 1.0000
Fine Pan Freq			0.3000 Hz
Mod Freq			0.0000 Hz
Mod Depth			0.0000
Coarse Phase Mult			0
Fine Phase Mult			0.2500
Pan Depth			81.0000 %
Input			Mono
Tempo/Sync		Host ▾	120.0000

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.

- 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.

Parameter	Min	Max	Value
Coarse Pan Freq			0 Hz
Fine Pan Freq			0.3000 Hz
Mod Freq			0.0000 Hz
Mod Depth			0.0000
Coarse Phase Mult			0
Fine Phase Mult			0.2500
Pan Depth			81.0000 %
Input			Mono
Tempo/Sync		Free ▾	120.0000

## 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.

Parameter	Min	Max	Value
Coarse Pan Freq		1 ▾	0.0000 * 1
Fine Pan Freq			0.3000 Hz
Mod Freq			0.0000 Hz
Mod Depth			0.0000
Coarse Phase Mult			0
Fine Phase Mult			0.2500
Pan Depth			81.0000 %
Input			Mono
Tempo/Sync		UDT ▾	1.0000

- 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.



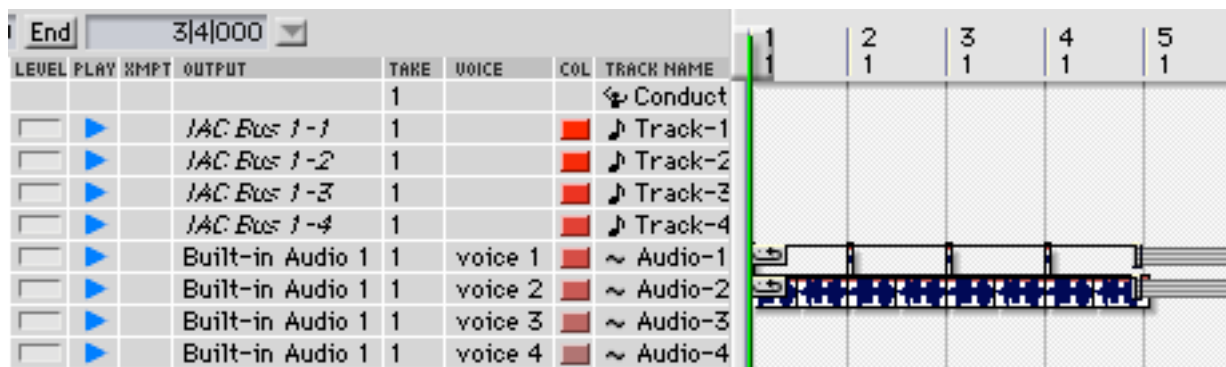
You can also use the Coarse Pan Freq note value pop-up menu and multiplier slider to change the panning rate.

## 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 Digital Performer.



- 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*.

Parameter	Min	Max	Value
Coarse Pan Freq		1	0.0000 * 1
Fine Pan Freq			0.3000 Hz
Mod Freq			0.0000 Hz
Mod Depth			0.0000
Coarse Phase Mult			0
Fine Phase Mult			0.2500
Pan Depth			81.0000 %
Input			Mono
Tempo/Sync		Plug	1.0000

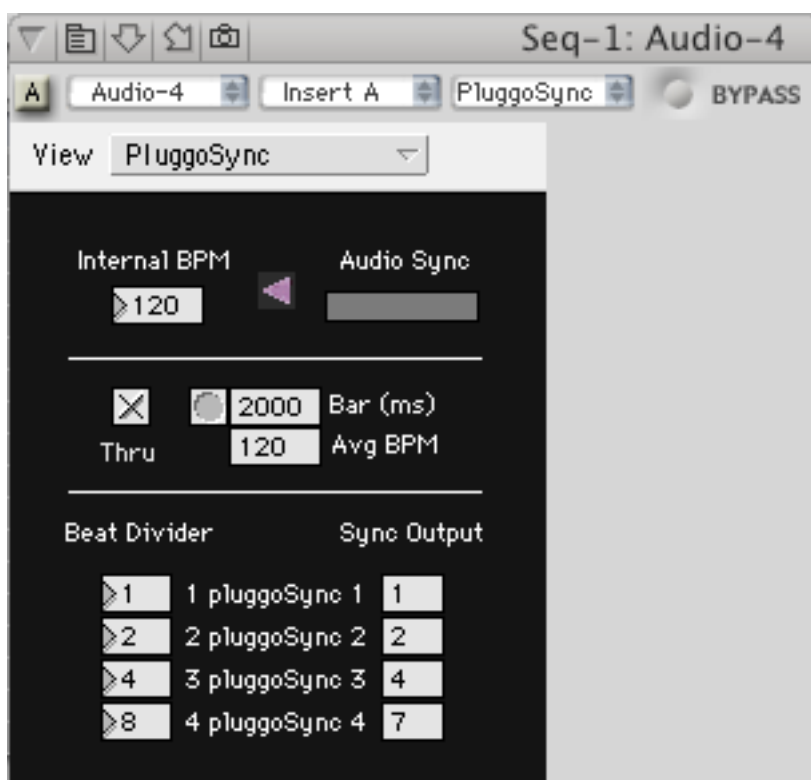
- Play the sequence. You should hear the *Audio Rate Pan* plug-in panning the audio track in sync with the click track on Audio 1. This click track is sending its audio signal to the *PluggoSync* plug-in instead of using the host application.
- Try using your sequencer’s arrangement window to change 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

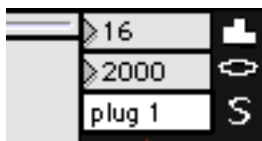
Synchronization using the *PluggoSync* plug-in requires that you create a “click track” that feeds the *PluggoSync* plug-in. The synchronized plug-ins then “listen” to one of *PluggoSync*’s synchronization sources.

- 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:

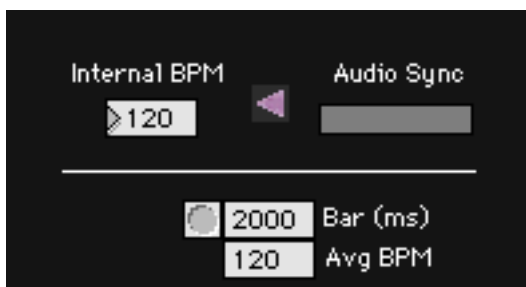
- Import the audio file *sync.aiff* found in the *Pluggo Stuff* folder onto an audio track, and make copies of it so that it plays once every bar of the sequence.
- Insert the *PluggoSync* plug-in and send it the audio from the click track.

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

- In Cubase, Logic, Digital Performer, and Pro Tools, 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.

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.



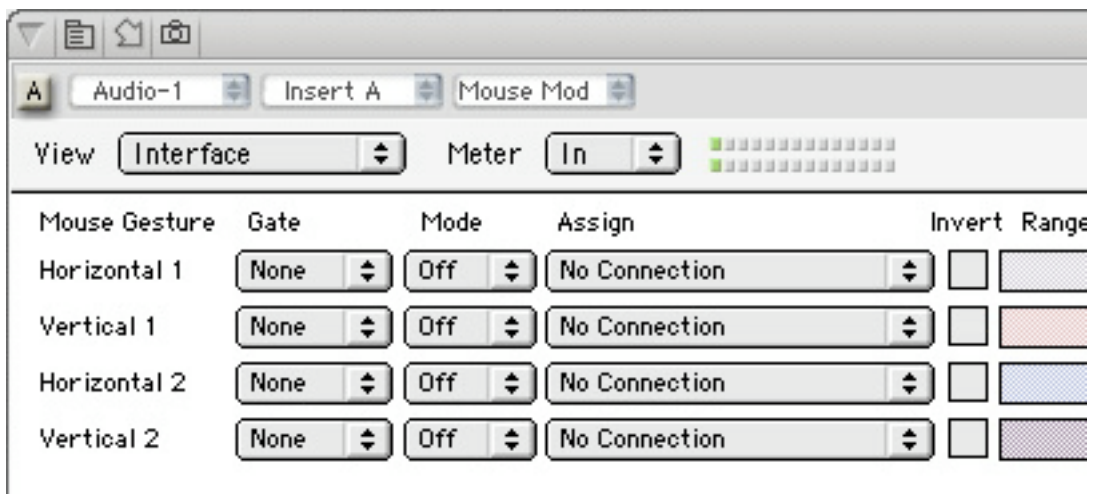
One use of the internal clock is to keep your plug-ins in sync with each other without a click track. *PluggoSync*'s internal clock will simply run freely at the designated BPM and it will run whether or not your sequencer is playing a song. In other words, *PluggoSync* is functioning as a sequencer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

# Using Modulator Plug-ins

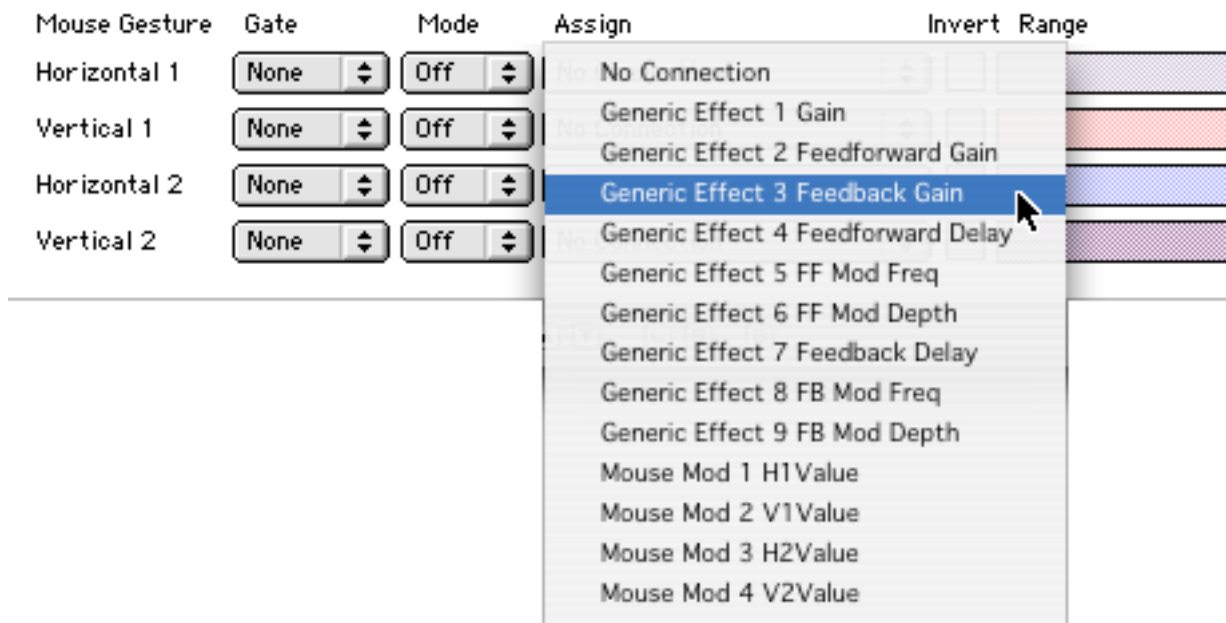
**pluggo**'s Modulators are plug-ins that do not process any audio. Instead they are designed to modify the parameters of other plug-ins. **pluggo** provides a number of different Modulators. We're going to look at one of the simpler ones called *Mouse Mod*. It lets you move the cursor around the screen (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 stress 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:



# Getting On the PluggoBus

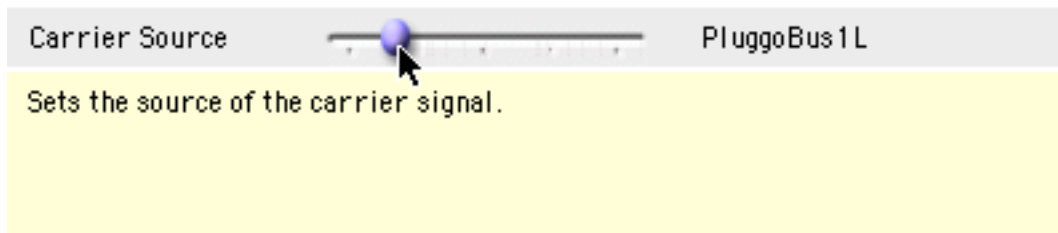
In audio mixer terminology, a bus is a signal path that can be both an input and output for different elements of the mixer. For instance, the output bus on a mixer is the sum of all the individual channels feeding into it. Other types of busses (yes, that's the way it's pluralized) combine effects or 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.

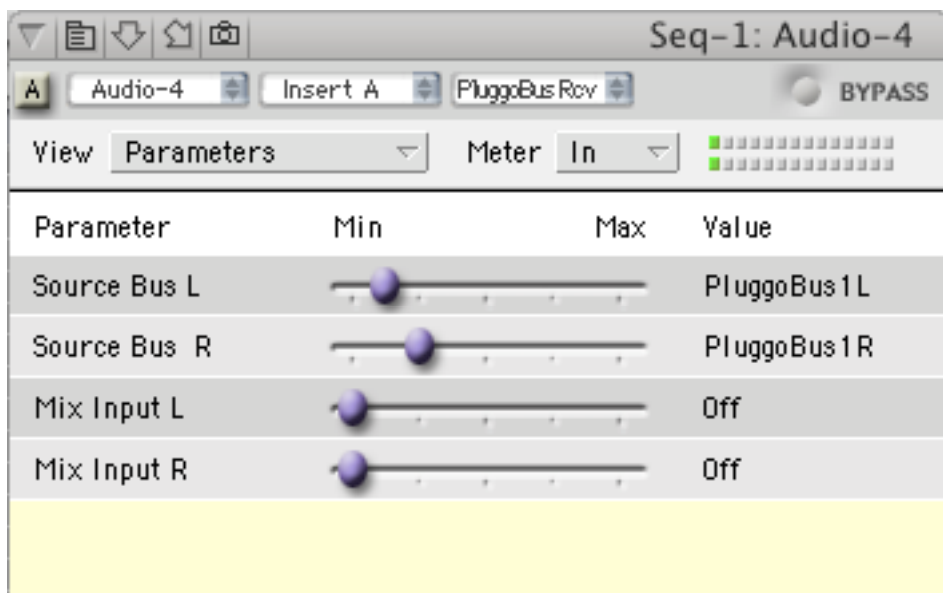
## PluggoBus Receivers

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).

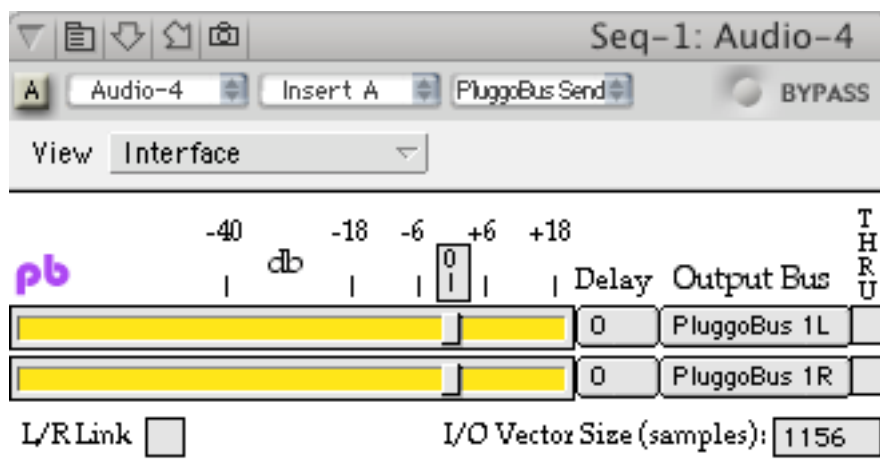
In addition to specific plug-ins that accept information from the bus, there is also a plug-in called *PluggoBus Rcv*. This plug-in can be used to feed an effect inserted below it with the contents of the bus.



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.

# Index

- authorization code, 9
- automation
  - Touch Parameters, 16
- Beat Divider, 29
- breakpoint envelope editor, 19
- bus, 33
- Carrier Source, 33
- click track, 27, 29
- Cubase
  - master effects;master effects (Cubase), 12
  - automated mixdown
- automated mixdown
  - automated mixdown;automation, automating, 14
- demo mode, 9
- Copy All from Program
  - copying, 17
- effect programs, 5
- filter graph, 20
- gain slider, 19
- Generic Effect, 31
- Parameter Change pop-up menu;Disable hints,Enable hints
  - enabling and disabling, 20
- horizontal slider, 19
- inserting plug-ins
  - Max/MSP
- inserting plug-ins
  - inserting plug-ins, 13
- installation, 8
- level meter, 20
- LFO, 31
- license agreement, 9
- Max, 5, 7
- modulators, 31
- Mouse Mod, 31
  - Mode menu
  - Assign menu, 32
    - output range, 32
- MSP, 5, 7
- multislider, 19
- note-unit multiplier, 24
- note-unit pop-up menu, 24
- number box, 19
  - changing parameters
- changing numerically, 15
- coarse mode
  - coarse mode, 15
- changing effect parameters
  - changing, 15
- parameters
  - nudging, 17
  - modulating
- modulation
  - and audio input, 31
  - pop-up menu, 16
- Randomize All
  - Randomize All, 17
- Parameters view, 17
- Pluggo FAQ, 6
- Pluggo Info view, 18
- Pluggo Installer, 8
- Pluggo Plug-in Reference Guide, 5, 13
- Pluggo Stuff folder, 29
- PluggoBus, 33
  - input level, 34
- PluggoBus Rcv, 34
- PluggoBus Send, 34
- Pluggo-RTAS plug-in, 8
- PluggoSync, 22, 23, 26, 28
- Messages view
  - diagnostic messages, 18
- mono plug-ins;stereo plug-ins;multi-channel plug-ins
  - mono, stereo, multi-channel, 12
- plug-ins
  - edit window, 14
  - inserting, 12
  - inserting stereo, 13
- plugs
  - getting information about, 18
- pop-up menu
  - parameter change, 16

- pop-up menus
  - Interface view, 19
- pre-fader send, 29
- range slider, 19
- Registration ID, 9
- repetitive strain injury, 31
- sync mode menu, 23
- sync.aiff audio file, 29
- free synchronization mode
  - free, 22
- host synchronization mode
  - host, 22
- plug synchronization mode
  - plug, 23
- UDT synchronization mode
  - UDT, 23
- free synchronization mode
  - free, 25
- host synchronization mode
  - host, 23
- plug synchronization mode
  - plug, 26
- synchronization modes, 22
- UDT synchronization mode
  - UDT, 25
- synchronizing plug-ins, Cubase
  - synchronization, Logic Audio
- synchronization, Digital Performer
  - synchronization, ProTools
  - synchronization, 22
- System Requirements, 6
- technical support, 6
- Tempo/Sync parameter, 23
- toggle box, 19
- Undo, 16
- Undo Last Change, 16
- user interface elements, 18
- user-defined tempo, 25
- vertical slider, 19
- View pop-up menu, 17
- Vocoder, 33
- VST instruments, MIDI
  - VST instruments, 13