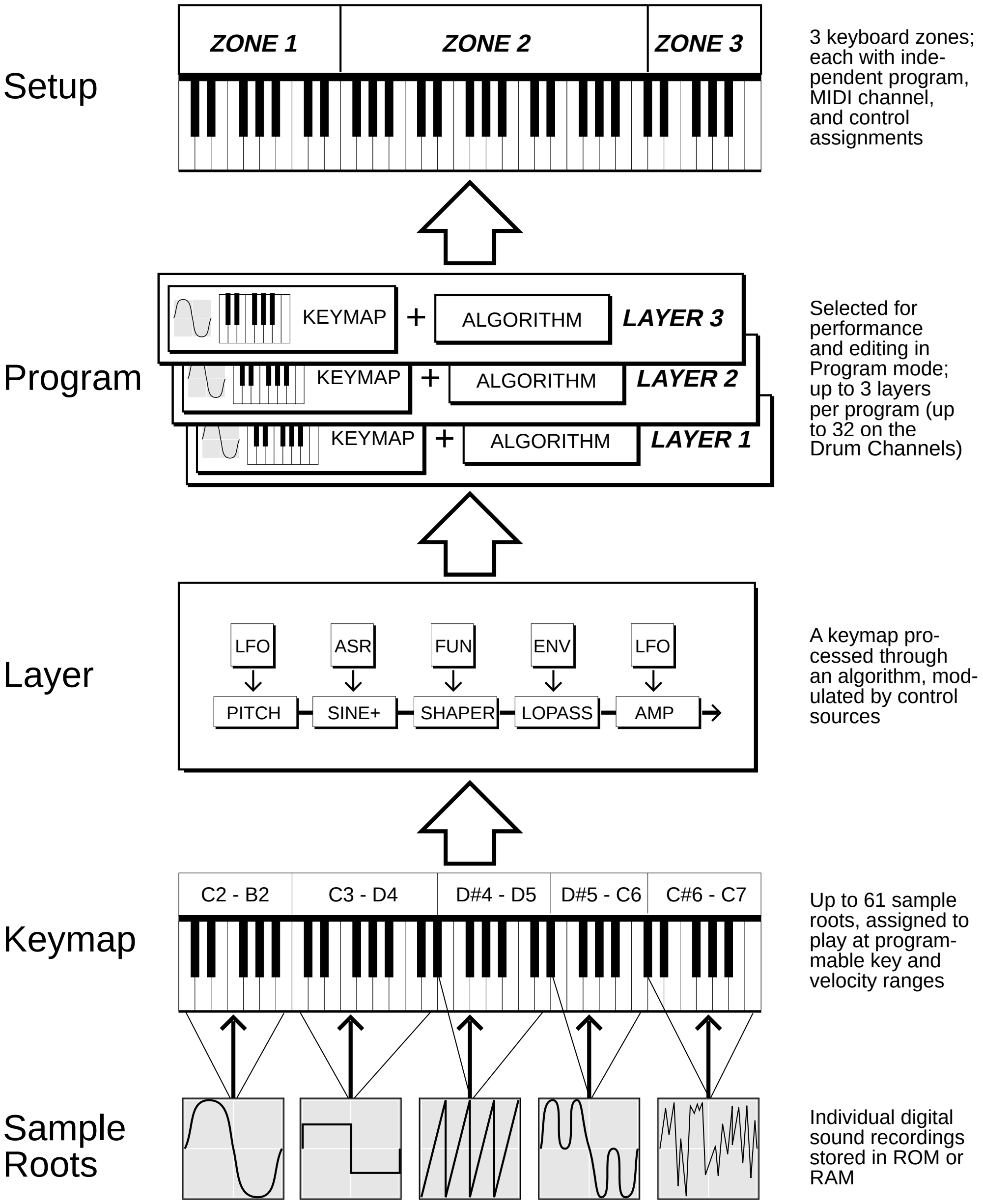
Chapter 6
Program Mode and the Program Editor

Program mode is the heart of the K2500, where you select programs for performance and editing. The K2500 is packed with great sounds, but it's also a synthesizer of truly amazing depth and flexibility. When you're ready to start tweaking sounds, the Program Editor is the place to start. But first there's a bit more general information about Program mode that wasn't covered in Chapter 2. Refer to the illustration below as you read the sections that follow.

K2500 Program Structure


ZONE 1
ZONE 2
ZONE 3
Setup
3 keyboard zones; each with independent program, MIDI channel, and control assignments
KEYMAP
+
ALGORITHM
LAYER 3
LAYER 2
LAYER 1
Program
Selected for performance and editing in Program mode; up to 3 layers per program (up to 32 on the Drum Channels)
Layer
LFO
ASR
FUN
ENV
LFO
PITCH
SINE+
SHAPER
LOPASS
AMP
A keymap processed through an algorithm, modulated by control sources
C2 - B2
C3 - D4
D#4 - D5
D#5 - C6
C#6 - C7
Keymap
Up to 61 sample roots, assigned to play at programmable key and velocity ranges
Sample Roots
Individual digital sound recordings stored in ROM or RAM

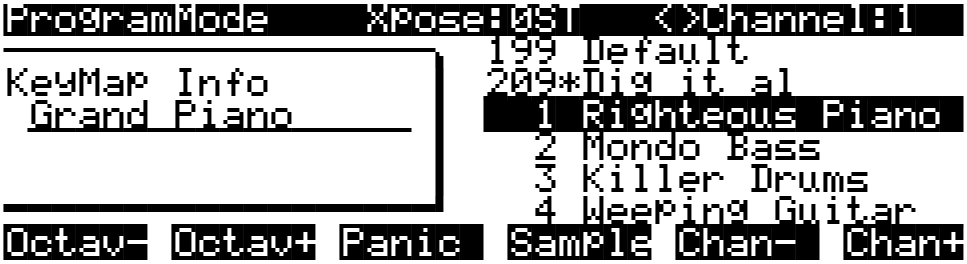
K2500 Program Structure

Programs are the K2500's performance-level sound objects. They're preset sounds that can be played on any of the 16 MIDI channels. Other synths refer to them as patches, presets, voices, multis, etc. Each program consists of from one to three ***layers*** (or up to 32 layers for a drum program - see below), each of which in turn consists of a keymap and an algorithm for processing the samples that make up the keymap. Samples are stored in the K2500's ROM, or loaded into optional RAM via Disk mode, MIDI standard sample transfer, SMDI sample transfer, or by your own sampling efforts. Each sample is a separate digital recording of an instrumental note, a drum, a waveform or a sound effect. The individual samples are assigned to specific key ranges and are assigned to be triggered at specific attack velocities. These assignments constitute the keymap.

When you trigger a note, the K2500 looks to the keymap of each ***layer*** of the currently active program(s) to determine which samples to play. The sound engine then fetches the requested samples and generates a digital signal that represents the timbres of the samples. This signal first passes through the five DSP functions that make up the algorithm, then through the global effects processor (if assigned to an audio output pair that uses effects), then finally appears at one or more of the audio outputs.

The layer is the K2500's basic unit of polyphony, that is, each layer constitutes one of the 48 voice channels the K2500 can activate at any time. If you have a program that consists of two layers covering the note range from C 0 to C 8, each note you trigger will trigger two voice channels.

The Program Mode Page



```
ProgramMode    Xpose:0ST    <>Channel:1
                          199 Default
KeyMap Info               209*Dig it al
 Grand Piano                1 Righteous Piano
                            2 Mondo Bass
                            3 Killer Drums
                            4 Weeping Guitar
Octav-  Octav+  Panic  Sample  Chan-  Chan+
```

The top line of the Program mode entry level page shows your location, the present MIDI transposition, and the current MIDI channel.

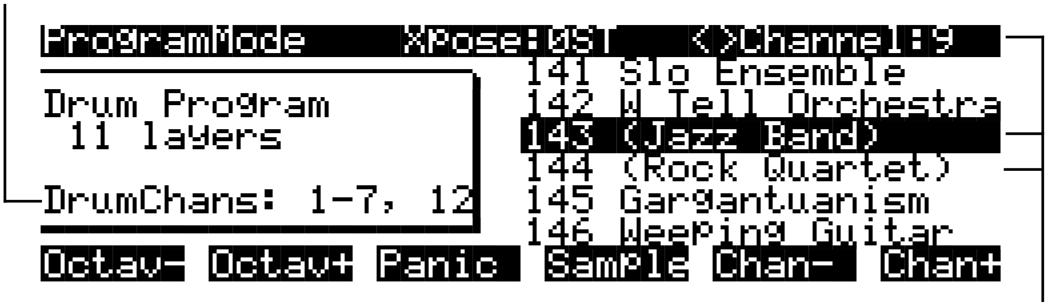
The box at the left of the Program mode page tells you which keymap is assigned to each layer. Layer 1 is at the top. Layers 2 and 3, if any, are listed below layer 1. If a ***drum program*** is selected, the box tells you how many layers it has. If a drum program is selected, and the current MIDI channel is not a drum channel, the program's name will appear in parentheses in the display, and the program will not play (more on this in the next section). The box at the left of the display will tell you which channels are drum channels. (See the Drum Channel parameter in Chapter 11).

The line beneath the name of the keymap indicates the keyboard range of the layer. On the Program mode page above, for example, the layer extends from C 0 to C 8—the default range. The representation of these layer ranges is approximate; they're intended to let you know if you have a layered keyboard (lines overlapping) or a split keyboard (lines not overlapping).

What are these programs in parentheses?

While you are scrolling through different programs on various MIDI channels, you may occasionally see a program that is in parentheses and doesn't make any sound. The parentheses tell you that you have selected a drum program without being on one of the drum channels. Unlike normal programs on the K2500, which can contain up to 3 layers, drum programs can have as many as 32 layers.

Current drum channels are 1-7 and 12.



```
ProgramMode     Xpose:0ST    <>Channel:9
                         141 Slo Ensemble
Drum Program             142 W Tell Orchestra
 11 layers               143 (Jazz Band)
                         144 (Rock Quartet)
DrumChans: 1-7, 12       145 Gargantuanism
                         146 Weeping Guitar
Octav- Octav+ Panic  Sample Chan-  Chan+
```

Current MIDI channel is 9. Parentheses around drum program names show that they won't sound on this channel.

Even though drum programs don't have to have drum sounds assigned to them, we use the name drum program because that is the most likely purpose for a program with 32 layers. The only limitation for a drum program is that it must be played on a MIDI channel that has been designated as a drum channel. A drum channel can play any program - it does not have to play a drum program, but a drum program must be played on a drum channel.

You can have up to 8 drum channels on the 2500. Channels 1 through 8 are drum channels by default, but you can change this to be channels 1 through 7 plus the channel of your choice. To set that channel, go to the Master page and set the Drum Channel parameter to any value between 9 and 16 to select that channel plus 1 through 7. In the example above, DrumChan on the Master page has been set to 12.

Control Setup

Version 2 of the K2500 software provides a new control setup feature. You specify the control setup in the CtlSetup parameter on the MIDI XMIT page; by default it's 99 DefaultSetup. Zone 1 of the control setup (which can be any available setup) defines the functions of the various controllers on the K2500 for all programs. This is especially important on keyboard models of the K2500, since they have eight programmable sliders, two ribbon controllers, and a number of assignable buttons. So for example, if you want to reprogram the long ribbon to send to a destination other than AuxBend2, you would either change the control setup or edit the current control setup.

Two important things to remember about the control setup are:

- The current control setup will be used by all programs. You cannot change the control setup from within program mode.
- The control setup does not affect the sound of a program. Samples and keymaps are still assigned within the program editor. The local program(s) assigned to a setup are ignored by programs that use the setup as a control setup.

Complete information on setups is in Chapter 7 of this manual.

The Soft Buttons in Program Mode

If the value of the Local Keyboard Channel parameter (on the RECV page in MIDI mode) matches the transmission channel of your MIDI controller, you can use the **Octav-** or **Octav+** soft buttons for quick transposition. The top line will reflect the change. This will transpose the K2500 as well as any MIDI devices connected to the K2500's MIDI Out port. Press both Octave soft buttons simultaneously to reset the transposition to zero. Changing the transposition with the soft buttons will also change the corresponding setting on the MIDI XMIT page. If the Local Keyboard Channel parameter is not set to match the transmission channel of your MIDI controller, the Octave soft buttons will have no effect.

Pressing the **Panic** soft button will send an All Notes Off message and an All Controllers Off message on all 16 MIDI channels.

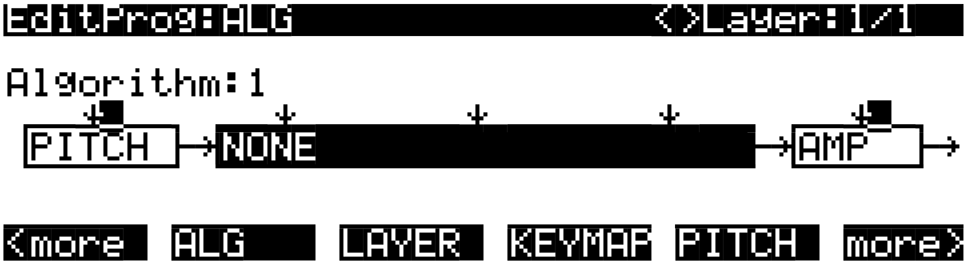
Press the **Sample** soft button to enter the K2500's sampler. Refer to Chapter 15 for complete information on the sampler.

Use the **Chan-** and **Chan+** soft buttons to change the current MIDI channel. This changes the MIDI channel the K2500 uses internally, as well as the channel you're using to send information to other synths connected to the K2500's MIDI Out port (MIDI slaves). Changing the current MIDI channel with the soft buttons also changes the corresponding setting on the MIDI XMIT page.

Using the Program Editor

The Program Editor is where you begin to modify the K2500's resident sounds, and to build your own sounds around samples (ROM or RAM) and/or waveforms. There's virtually no limit to the sounds you can create using the tools in the Program Editor.

To enter the Program Editor, start in Program mode and press EDIT. The Program mode LED will go out, and the ALG (Algorithm) page will appear.



EditProg:ALG <>Layer:1/1

Algorithm:1

The top line of the display gives you the usual reminder of your location. It also tells you which layer you're viewing, and how many layers there are in the program. You can use the CHAN/ BANK buttons to scroll through the layers, if the program has more than one.

Here's a method for jumping quickly to a specific layer in a program that is especially useful in multi-layered drum programs. Hold the ENTER button and strike a key on your controller. The display will change to the layer assigned to that key. If more than one layer is assigned to the same key, repeatedly striking the key (while continuing to hold the ENTER button) will cycle through all layers assigned to that key. This method will work in most places within the Program Editor, but there is an exception: if the parameter you have highlighted has a note number or control source as its value, then holding ENTER and striking a note will call up that note or control source (as described in the "Intuitive Entry" section on page 3-6). For all other parameters, however, this method will switch between layers.

The Soft Buttons in the Program Editor

The Program Editor's soft buttons are labeled by the words that appear in the bottom line of the display. These buttons have two important jobs in the Program Editor: selecting pages, and selecting specific functions. If a soft button is labeled in all uppercase letters, pressing it will take you to the page it describes. If the button is labeled in mixed uppercase and lowercase letters, pressing it will execute the software function described by the label. Pressing the **PITCH** soft button, for example, will select the PITCH page, while pressing the **Save** soft button will initiate the process for saving the currently selected program.

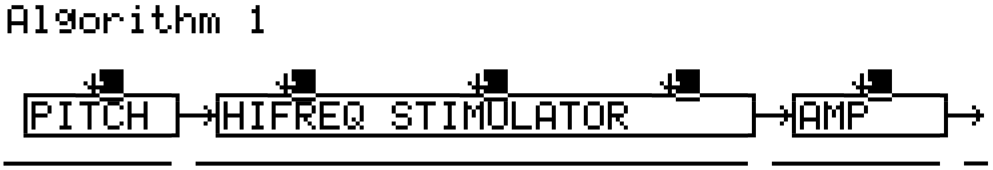
There are more pages and functions in the Program Editor than there are soft buttons. Therefore, two of the soft buttons are dedicated to scrolling through the list of pages and functions. If you don't see the button for the page or function you want to select, press one of the soft buttons labeled **<more>**, and the labels will change. This doesn't change the currently selected page, it merely changes the selection of available soft buttons.

Five of the soft buttons in the Program Editor are special cases. They're the soft buttons that select the editing pages for the five control input pages for the ***DSP functions***. One of these soft buttons is always labeled "PITCH," since the first DSP function in every algorithm is the pitch control. The remaining four vary somewhat depending on the DSP functions you choose for the currently selected algorithm, but they always have the prefixes F1, F2, F3, and F4, and they always take you to the pages for the four DSP functions that follow the pitch control function.

Algorithm Basics

The basic definition: an algorithm is the "wiring" (signal path) of a sample to the audio outputs, through a series of digital signal processing (DSP) functions that you select. The K2500's algorithms are the core of Variable Architecture Synthesis Technology. The DSP functions are synthesis tools (filters, oscillators, etc.) that you assign to the various stages of the algorithm. The DSP functions you choose determine the type of synthesis you use.

Each of the 31 available algorithms represents a preset signal path. You can't change the path of the algorithms, but you can select different algorithms, and assign a wide variety of DSP functions to the individual stages of each algorithm's signal path. Take a look at Algorithm 1 in the diagram below. It's one of the simplest algorithms.



The DSP functions are represented by the rectangular blocks. The horizontal arrows indicate the flow of the digital signal from left to right. They represent what we call the "wire" of the algorithm, the actual physical path that the signal follows through the algorithm. Selecting different algorithms can be compared to connecting different DSP functions with different wiring diagrams.

Think of the left side of each block as its input, and the right side as its output. Depending on the algorithm, the signal may split into two wires, enabling part of the signal to bypass certain portions of the algorithm. Split wires may rejoin within the algorithm, or they may pass all the way through as split signals. If the last block has two wires at its output, we call it a double-output algorithm. If it has one wire, it's a single-output algorithm, even if there are two wires in earlier portions of the algorithm.

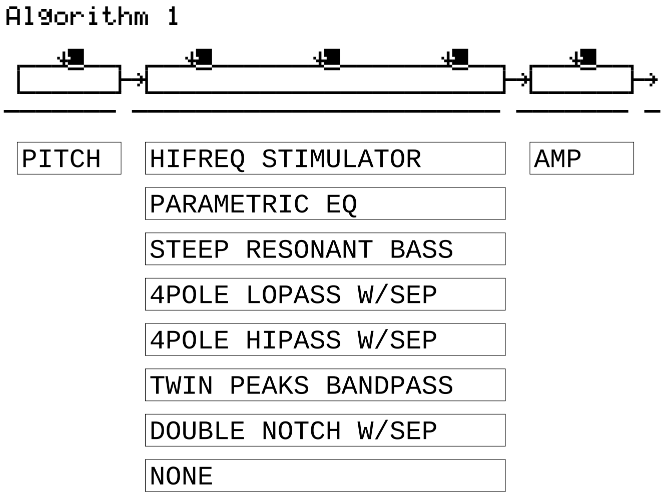
The five downward-pointing arrows indicate the five realtime control inputs to the DSP functions. There are usually five inputs, each of which has its own page within the Program

Editor. (Algorithms 26-31, which use hard sync oscillation, have only four inputs; you can read about hard sync functions on page 14-51.) Each of these pages has several parameters that can modulate its related DSP function. Often a single DSP function will have more than one input. That’s why some blocks are larger, and have more than one arrow pointing to them. Each function can be independently controlled by a variety of sources (the Control Source list), including LFOs, ASRs, envelopes, programmable functions, and external MIDI.

In Algorithm 1, the signal flows first through a one-stage DSP function that controls the pitch of the samples in the keymap. In fact, the first DSP function in each algorithm *always* controls pitch, even though it doesn’t apply in every instance. Similarly, the last DSP function always controls the final amplitude of the signal. It can be a one-stage or two-stage function. In Algorithm 1, it’s a one-stage function.

The second, larger block indicates a single three-stage DSP function, meaning that it has three control inputs that can be adjusted to modulate the signal.

Once again, you can’t change the wiring path of an algorithm—you can simply select a different algorithm to get a different path. And within each algorithm, you can assign a large number of different DSP functions to each of the five control inputs. The diagram below, for example, shows Algorithm 1 with all the possible values for each DSP function lined up under the blocks that represent the DSP functions.



Notice that PITCH is the only value available for the first block, and AMP is the only value available for the last block. The center, three-stage block, however, allows you to choose from seven DSP functions. An eighth value, NONE, deactivates the block.

Common DSP Control Parameters

The type of DSP function available for any function block depends on the algorithm. Some of the specialized functions like the PANNER are always located just before the final AMP function. Others, like the three-input functions, appear only in algorithms that are structured for three-input functions.

You can change the nature of each layer of a program simply by assigning different DSP functions to the layer’s algorithm. Your level of control goes much deeper than that, however. Each DSP function has one or more inputs to which you can patch a variety of control sources to modify the behavior of the DSP functions themselves. These control inputs are represented by the arrows pointing down at the blocks that make up the algorithm. For each input arrow, there’s a corresponding control input page that you can select with the five special soft buttons we mentioned above (PITCH, and F1–F4). All of the DSP functions have at least one control input, but many of them have two or even three inputs.

The parameters on the various control input pages are very similar; in fact, there are six parameters that appear on almost every page. Consequently we refer to them as the common DSP control parameters. Although the parameters on the control input pages differ slightly from function to function, you can expect to see some or all of the common DSP control parameters whenever you select the control input page for any of the DSP functions. They are:

Initial Setting Parameters

These have no input, but set the overall level for the function, the starting point from which the other parameters modulate the function.

Coarse adjust (abbreviated “Coarse”)

Fine adjust (Fine)

Hard-wired Parameters

These always take their input from your MIDI controller; the note number and the attack velocity values of each Note On event.

Key tracking (KeyTrk)

Velocity tracking (VelTrk)

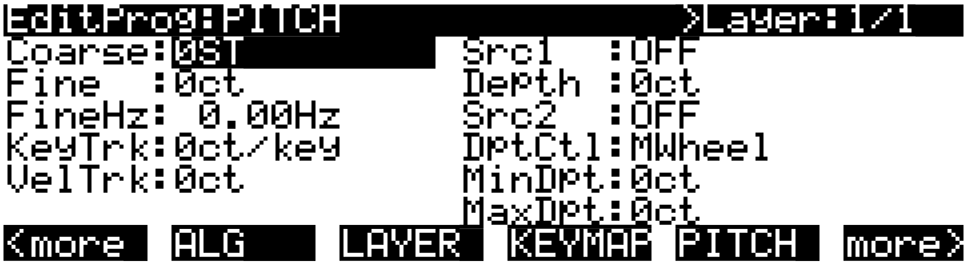
Programmable Parameters

These can accept any control source as their input, and have related parameters for further control.

Source 1 (Src1)

Source 2 (Src2)

Take a look at the PITCH page, as an example—we’ll look at how these six control parameters are used in the pitch control function. If you’re not already on the PITCH page, you can get there by pressing the soft button labeled **PITCH**. If you don’t see **PITCH** on the bottom line of the display, press one of the **<more>** buttons until it appears.



```
EditProg:PITCH                  >Layer:1/1
Coarse:0ST            Src1   :OFF
Fine   :0ct           Depth :0ct
FineHz: 0.00Hz        Src2   :OFF
KeyTrk:0ct/key        DptCtl:MWheel
VelTrk:0ct            MinDpt:0ct
                      MaxDpt:0ct
```

You’ll recognize the common DSP control parameters, along with several other parameters. Keep in mind that there’s a set of common control parameters for each of the DSP functions; in this case we’re describing them only as they apply to the pitch control function.

Coarse Adjust

The Adjust parameter (sometimes coarse and fine adjust) is the fixed amount of adjustment you add to any DSP function. On the PITCH page, the Coarse Adjust parameter will change the pitch in semitone increments. Use this as a starting point to set the pitch where you want it to be normally. This will shift the pitch of the currently selected layer, and will affect the playback rate of sampled sounds. Sampled sounds have an upper limit on pitch adjustment. It’s normal for the pitches of sampled sounds to “pin” (stop getting higher) when you adjust the pitch upward in large amounts. The oscillator waveforms can be pitched higher. Any sound can be pitched downward without limit.

The primary use of the Adjust parameters (Coarse *and* Fine) is to offset the cumulative effects of the other parameters on the control input pages. For example, you might set a high value for key tracking (defined below) for a dramatic change in effect across the keyboard. The effect might be too much at one end of the keyboard, however, so you could use one of the Adjust parameters to reduce the initial amount of that effect.

The K2500 always uses real values of measurement, rather than just arbitrary numbers, for adjustable parameters. This means that you specify pitch in semi-tones and cents, amplitude in dB, and filter cutoff frequency in hertz.

Remember that the parameters on the control input pages are cumulative—they can add to or subtract from the effects of the other parameters on the page, depending on their values. For example, even if you've adjusted the pitch of a sample so high that it pins, the effects of the other parameters may bring the pitch back down to a workable range.

Fine Adjust

You can add slight detuning to the pitch by changing the fine adjust parameter. Notice that there are actually two fine adjust parameters on the PITCH page: one that changes the pitch in cents (100ths of a semitone), and one that changes it according to its frequency (in increments of Hertz—cycles per second). Since we're discussing the universal control sources here, and not specifically pitch, we'll move on for now, as the "Fine Hz" parameter applies only to pitch-related functions. See "The PITCH Page," later in this chapter, which describes Fine Hz more thoroughly.

Key Tracking

This is a quick way to get additional control based on the MIDI note number of each note you trigger. Key tracking applies a different control signal value for each note number. In the case of pitch, key tracking enables you to change the tuning of each note relative to its normal pitch.

Middle C is the zero point. Regardless of the key tracking value, there is no effect on Middle C. If you set a non-zero value for key tracking, the effect increases for each note above or below Middle C. In the case of pitch, for example, say you assign a value of 5 cents per key for the key tracking parameter. Triggering Middle C (C 4 on the K2500) will play a normal C 4. Triggering C# 4 will play a note 5 cents higher than C# 4. Triggering D 4 will play a note 10 cents higher than D 4, and so on. Notes below Middle C will be tuned lower than their normal pitches. If you set a negative value for key tracking, notes above Middle C will be tuned lower than their normal pitches.

Keep in mind that key tracking on the PITCH page works in conjunction with the key tracking parameter on the KEYMAP page. This is why you can set the KeyTrk parameter on the PITCH page to 0ct/key, and the K2500 will still increases in pitch by 100 cents/key as you go up the keyboard. It's because the KeyTrk parameter on the KEYMAP page is already set at 100 cents/key.

Velocity Tracking

A positive value for velocity tracking will raise the pitch as you trigger notes with higher attack velocities. This is great for getting a trace of detuning based on your attack velocity, especially in drum programs, where you can make the pitch of the drum samples rise slightly with higher-velocity Note Ons, just as drums do when you strike them harder. Negative values will lower the pitch as you increase the attack velocity.

Source 1

This parameter takes its value from a long list of control sources (you can find it in the *Reference Guide*—it's called the Control Source list) including every MIDI control number, a host of LFOs, ASRs, envelopes and other programmable sources.

Src1 works in tandem with the parameter beneath it on the page: Depth. Choose a control source from the list for Src1, then set a value for Depth. When the control source assigned to Src1 is at its maximum, the pitch will be altered to the full depth you set. For example, if you set Src1 to "MWheel," and set Depth to 1200 ct, the pitch will rise as you push the Mod Wheel up on your MIDI controller, reaching a maximum of 1200 ct (12 semitones, or one octave).

Source 2

This one's even cooler. Like Src1, you choose a control source from the list. But instead of setting a fixed depth, you can set a minimum and maximum depth, then assign another control source to determine how much depth you get. Try this example. (Make sure Src1 is set to OFF first, so the two sources don't interact). Start with Program 199, and press EDIT. Press the **PITCH** soft button to select the PITCH page. Set the Src2 parameter to a value of LFO1, then set the Minimum Depth parameter to 100 ct, and Maximum Depth to 1200 ct. Then set the Depth Control parameter to MWheel. This lets you use your MIDI controller's Mod Wheel to vary the depth of the oscillation in pitch generated by the LFO.

Now, when the Mod Wheel is down, the pitch will oscillate between a semitone (100 ct) up and a semitone down (the default waveform for LFO1 is a sine wave, which goes positive and negative—if this perplexes you, see the *Reference Guide*, where there's an explanation of how the K2500 generates and interprets control source signals). With the Mod Wheel up, the pitch will oscillate between an octave up and an octave down.

Since the Mod Wheel is a continuous control, you can achieve any amount of depth control between the minimum and maximum. If you had set the Depth Control to Sustain, for example, then you'd get only two levels of depth control: the maximum (1200 cents) with your MIDI controller's sustain pedal down, or the minimum (100 cents) with the sustain pedal up.

Summary of Common DSP Control Parameters

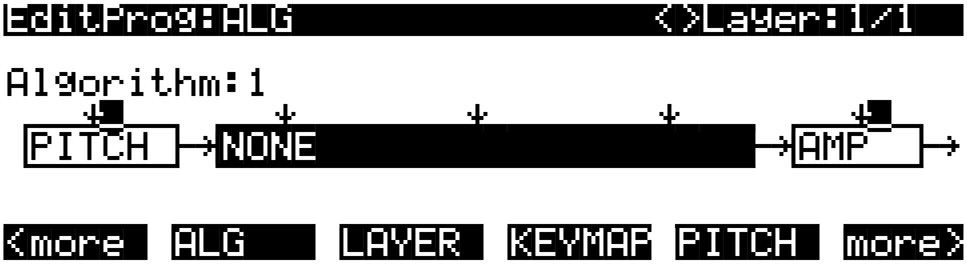
These six control source parameters are just a few of the control sources available throughout the Program Editor. We've given them special attention because they appear on *all* the pages relating to the DSP functions, not just on the PITCH page.

As with the PITCH parameters, you can go to each of the DSP functions' control input pages, and set a similar set of parameters to control each of those functions as well. The units of measurement may differ, but you'll almost always find one or two adjustment parameters, key and velocity tracking, and two programmable control sources. And remember, we've been talking about one layer in one program here. You can add one or two more layers to your program, and start all over with another identical set of control sources for each layer, each of which can be programmed independently.

On any given page, the settings for the control parameters are added to each other before the signal leaves the DSP function. Depending on the values you set, they may cancel each other out, or they may add up to huge amounts of modulation. If things get out of control, the easiest way to get a handle on the situation is to set some of the parameters to values of 0 or OFF. Adjust the value for one parameter at a time to hear the effect of that one parameter.

Program Editor—Algorithm (ALG) Page

The ALG page is the first page you see when you enter the Program Editor. It enables you to select from among the 31 possible algorithms, and assign the DSP functions within the current algorithm.



```
EditProg:ALG                    <>Layer:1/1

Algorithm:1

PITCH →NONE                        →AMP    →

<more  ALG    LAYER  KEYMAP PITCH  more>
```

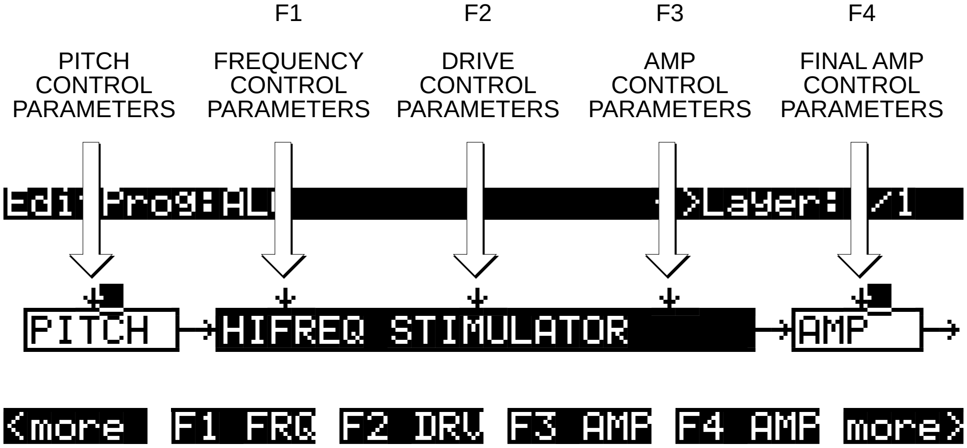
The top line of the display gives you the usual mode reminder, and tells you which layer you’re looking at, as well as how many layers are in the current program (in the diagram above, it’s the first layer of a one-layer program). You can view the ALG pages of any other layers in the program by using the CHAN/BANK buttons.

The central portion of the page shows the algorithm for the currently selected layer. You see the number of the algorithm (from 1 to 31) and a graphic representation of the signal path, as well as the currently selected DSP functions within the signal path.

To use a different algorithm, select the Algorithm parameter and use any data entry method to select a different one. To change the DSP function within an algorithm, move the cursor to the block you want to change, then use the Alpha Wheel or Plus/Minus buttons. There’s a staggering number of combinations of algorithms and DSP functions alone, not to mention the numerous controls that can be used to modify the DSP functions. the *Reference Guide* contains a list of all 31 algorithms and the DSP functions available for each one.

NOTE: Changing a layer’s algorithm can affect the layer’s sound drastically. It’s a good idea to bring down the volume of your K2500 or your sound system before changing algorithms.

The five downward-pointing arrows represent inputs to the DSP functions that are available for the current algorithm. Each input arrow has its corresponding page. The first arrow points to the PITCH function. The soft button for the PITCH page is already visible. Press it to view the parameters affecting pitch for the currently selected layer. The buttons for the other four DSP functions are not visible when you first enter the Program Editor. To see them, press the **more>** soft button (on the right side of the page). You’re still on the same page, but the soft buttons’ labels change to let you select a different set of pages, as the diagram below shows.

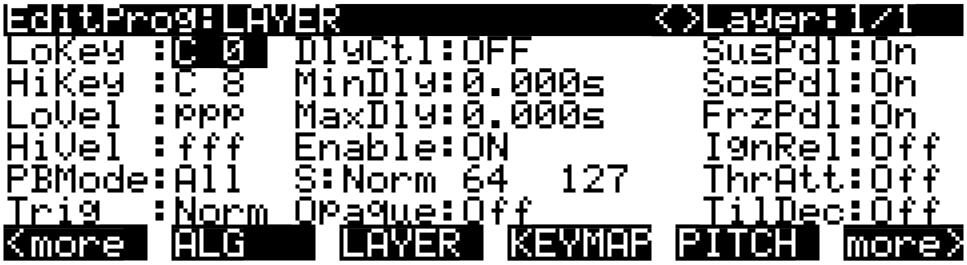


```
Edi Prog:AL                      >Layer: /1

PITCH →HIFREQ STIMULATOR           →AMP    →

<more  F1 FRQ F2 DRV F3 AMP F4 AMP more>
```

The pages (F1–F4) that control the DSP functions are described later in this chapter. “Algorithm Basics” on page 6-5 gives general information on algorithms, while Chapter 14 gives a thorough description of each the DSP functions and the parameters found on their editing pages.

Program Editor—LAYER Page

Press the **LAYER** soft button to call up the LAYER page. Here you'll set a number of parameters that affect the current layer's keyboard range, attack and release characteristics, and response to various controls.



```
EditProg:LAYER                <>Layer:1/1
LoKey :C 0  DlyCtl:OFF        SusPdl:On
HiKey :C 8  MinDly:0.000s     SosPdl:On
LoVel :ppp  MaxDly:0.000s     FrzPdl:On
HiVel :fff  Enable:ON         IgnRel:Off
PBMode:All  S:Norm 64  127    ThrAtt:Off
Trig  :Norm Opaque:Off        TilDec:Off
<more   ALG   LAYER   KEYMAP   PITCH   more>
```

PARAMETER	RANGE OF VALUES	DEFAULT
LOW KEY	C -1 to G 9	C 0
HIGH KEY	C -1 to G 9	C 8
LOW VELOCITY	ppp to fff	ppp
HIGH VELOCITY	ppp to fff	fff
PITCH BEND MODE	Off, Key, All	All
TRIG	Normal, Reversed	Normal
DELAY CONTROL	Control Source list	OFF
MINIMUM DELAY	0 to 25 seconds	0
MAXIMUM DELAY	0 to 25 seconds	0
LAYER ENABLE	Control Source list	ON
ENABLE SENSE	Normal, Reversed / Min. / Max.	Normal / 64 / 127
OPAQUE LAYER	Off, On	Off
SUSTAIN PEDAL	Off, On, On2	On
SOSTENUTO PEDAL	Off, On	On
FREEZE PEDAL	Off, On	On
IGNORE RELEASE	Off, On	Off
HOLD THROUGH ATTACK	Off, On	Off
HOLD UNTIL SUSTAIN	Off, On	Off

Low Key (LoKey)

This sets the lowest active note for the current layer. This parameter's value cannot be set higher than the value for HiKey. The standard MIDI key range is C -1—G 9 (0-127). Middle C is C 4 (ISP).

High Key (HiKey)

Here you set the highest active note for the current layer. This parameter's value cannot be set lower than the value for LoKey.

Low Velocity (LoVel)

With this parameter you define the lowest attack velocity at which the layer will be enabled (generate a sound). The values for this parameter and the next are expressed in the standard musical dynamics markings, similar to the values available for the velocity maps. Attack velocities that are below this velocity threshold will not trigger notes. If you set this parameter's value higher than the HiVel value, the layer will not play at all.

High Velocity (HiVel)

Similarly, this will set the highest attack velocity at which the layer will be enabled. Attack velocities above this velocity will not trigger notes in this layer.

Pitch Bend Mode (PBMode)

This determines how Pitch bend control messages will affect the current layer. A value of “All” bends all notes that are on when the Pitch bend message is generated. A value of “Key” bends only those notes whose triggers are *physically* on when the Pitch bend message is generated (notes held with the sustain pedal, for example, won’t bend). This is great for playing guitar solos on top of chords—play a chord, hold it with the Sustain pedal, then play your licks and bend them all you want; the chord won’t bend with it. A value of “Off” disables Pitch bend for the current layer.

Trigger (Trig)

Set Trig to "Rvrs" to have notes triggered on key-up. The initial velocities of notes triggered this way are determined by the release velocities of the keys that trigger them. The default setting is "Norm"

Delay Control (DlyCtl)

Here you select, from the Control Source list, a control source that will delay the start of all notes in the current layer. The length of the delay is determined by the following two parameters. You’ll assign a continuous control like MWheel for the DlyCtl parameter when you want to vary the delay time, and a switch control if you want the delay to either be its minimum value (switch off), or its maximum (switch on). The delay control will affect only those notes triggered *after* the delay control source is moved; the delay time is calculated at each note start, based on the status of the delay control source at that time.

Minimum Delay (MinDly), Maximum Delay (MaxDly)

The length of the delay is determined by these two parameters. When the control source assigned to DlyCtl is at its minimum, the delay will be equal to the value of MinDly. The delay will be equal to the value of MaxDly when the control source is at its maximum. If DlyCtl is set to OFF, you get the minimum delay. If it’s set to ON, you get the maximum delay. This doesn’t change the note’s attack time, just the time interval between the Note On message and the *start* of the attack. The delay is measured in seconds.

Enable

This selects a control source to activate or deactivate the layer. When the assigned control source is on (or above the midpoint of 64 for continuous controls) the layer will be active. The layer will not sound when the control source is off or below its midpoint. For example, if you wanted to create an octave doubler that would kick in on demand, you could create a program with a second layer, transpose it up or down an octave, and set its Enable parameter to any control source, like MWheel. Then whenever your MIDI controller’s Mod Wheel is above its midpoint, you’ll hear the second layer. Some local control sources (e.g., KeyNum and AttVel) are not valid for the Enable parameter. In these cases, you should use the global equivalent (e.g., GKeyNum and GAttVel).

Enable Sense (S)

This lets you reverse the orientation of the MIDI control you’ve assigned to enable the layer. A value of Normal has no effect on the Enable parameter, while Reversed will activate the layer when the Enable control source is off or below its midpoint, and deactivate the layer when the Enable control source is on or above its midpoint. You could use this parameter to set up a two-layer program that would let you use a MIDI control to switch between layers, say a guitar sound and a distorted guitar. Both layers would have their Enable parameters set to the same control source, say MWheel. One would have its Enable Sense parameter set to Normal, and the other would have it set to Reverse. Then the first layer would play when your MIDI controller’s Mod Wheel was above its midpoint, and the second layer would play when the Mod Wheel was below its midpoint.

This is also the place to set the minimum and maximum values at which the layer will be enabled. The two numbers to the right of the Enable Sense parameter are the Minimum and

Maximum values for the Enable parameter. .The range for each is -128-127. When the controller is at a value between the Minimum and Maximum amount, the layer will be activated or deactivated, (depending on the setting of the Enable Sense parameter). For example, you could create a program with several layers, assign Mod Wheel for the Enable parameter on each layer, and set different Minimum and Maximum amounts for each layer. Then as you move the Mod wheel, you could trigger the different layers

Opaque

An opaque layer blocks all higher-numbered layers in its range, allowing only the opaque layer to play. This is an easy way to change a small range of notes in a program, leaving the original sound playing above and below the new sound. Just create a new layer (Layer 2), set its range (say, C 3 to D 3), assign it the keymap you want, and set its Opaque parameter to On. Then duplicate the original layer, so its number is higher than that of the new layer (the duplicate layer is Layer 3). You'll now have a three-layer program. Delete Layer 1 (the original layer), and the new Layer will become Layer 1, while the duplicated original layer becomes Layer 2. Now the new layer will block out the original layer at the notes C 3–D 3.

Sustain Pedal (SusPdl)

When this parameter is on, the layer will respond to all sustain messages (MIDI 64) received by the K2500. When off, the current layer will ignore sustain messages. On2 means that the sustain pedal will not catch the release of a note that is still sounding when the sustain message is received; this can be very useful in a program that uses amplitude envelopes with a long release time.

Sostenuto Pedal (SosPdl)

When Sostenuto is on, the layer will respond to all sostenuto messages (MIDI 66) received by the K2500. When off, sostenuto messages are ignored by the layer.

Freeze Pedal (FrzPdl)

Activate or deactivate the layer's response to Freeze pedal messages (MIDI 69). The Freeze pedal control causes all notes that are on to sustain without decay until the Freeze pedal control goes off. If a note is already decaying, it will freeze at that level.

Ignore Release (IgnRel)

When on, the layer will ignore all Note Off messages received by the K2500. This should be used only with sounds that decay naturally, otherwise the sounds will sustain forever. When IgnRel is off, the layer responds normally to Note Off messages. This parameter can come in handy when your K2500 is slaved to a drum machine or sequencer, which sometimes generates Note Ons and Note Offs so close together that the envelope doesn't have time to play before the note is released. You'll also want to use this parameter when you're playing staccato, and the sound you're playing has a long amplitude envelope. This parameter should be used only with notes that eventually decay to silence. Sustaining sounds will sustain forever.

Hold Through Attack (ThrAtt)

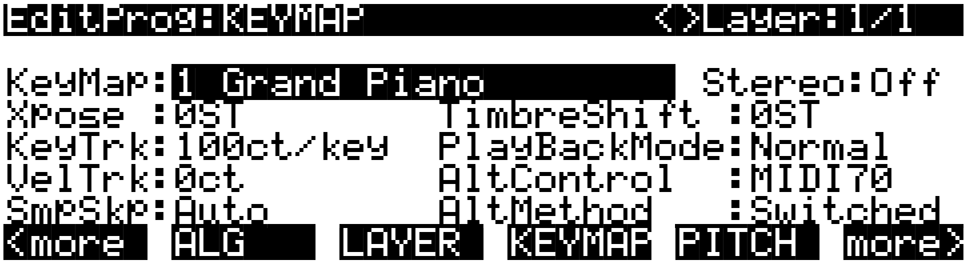
When on, this parameter causes all notes in the layer to sustain through the entire first attack segment of their amplitude envelopes, even if the notes have been released. If you have a sound with a slow attack, or an attack that's delayed with the delay control, setting this parameter to On will make sure your notes reach full amplitude even if you're playing fast. When set to Off, notes will release as soon as you release the note (generate a Note Off). If the first attack segment of the layer's amplitude envelope is very short, you probably won't notice a difference between values of On and Off.

Hold Until Decay (TilDec)

When on, this parameter causes all notes in the layer to sustain through all three attack segments in their amplitude envelopes even if the notes have been released. Looped amplitude envelopes will not loop, however, if the notes are released before reaching the end of the final attack segment. Notes will go into their normal releases if they are released after the envelope has looped. When set to Off, notes will release as soon as a Note Off message is generated.

Program Editor— KEYMAP Page

Press the **KEYMAP** soft button to call up the KEYMAP page. The parameters on this page affect sample root selection—which samples are played on which keys.



```
EditProg:KEYMAP              <>Layer:1/1

KeyMap:1 Grand Piano      Stereo:Off
Xpose :0ST        TimbreShift :0ST
KeyTrk:100ct/key  PlayBackMode:Normal
VelTrk:0ct        AltControl  :MIDI70
SmpSkp:Auto       AltMethod   :Switched
<more   ALG    LAYER  KEYMAP  PITCH  more>
```

PARAMETER	RANGE OF VALUES	DEFAULT
KEYMAP	**Keymap list**	**1 Grand Piano**
TRANSPOSE	**± 60 semitones**	**0**
KEY TRACKING	**± 2400 cents per key**	**100**
VELOCITY TRACKING	**± 7200 cents per key**	**0**
SAMPLE SKIPPING	**Auto, Off, On**	**Auto**
STEREO	**Off, On**	**Off**
TIMBRE SHIFT	**± 60 semitones**	**0**
PLAYBACK MODE	**Norm, Rvrs, Bidirectional, Noise**	**Normal**
ALT CONTROL	**Control Source list**	**OFF**
ALT METHOD	**Switched, Continuous**	**Switched**

Keymap

Assign a ROM or RAM keymap to the current layer. Keymaps are collections of samples assigned to note and velocity ranges. There are nearly 200 ROM keymaps to choose from. You'll find a list of them in the *Reference Guide*.

Transpose (Xpose)

Transpose the current keymap up or down as much as 60 semitones (5 octaves).

Key Tracking (KeyTrk)

This is one of the six common DSP control parameters. On the KEYMAP page, key tracking affects the interval between notes. The default value of 100 cents (hundredths of a semitone) gives you the normal semitone interval between each note. Higher values increase the interval; lower values decrease it. Negative values will cause the pitch to decrease as you play higher notes. You can create a mirror-image piano by setting the key tracking to –100 and transposing the layer up 4 semitones. When you make changes to this parameter, you'll need to keep in mind that KeyTrk on the KEYMAP page works in conjunction with KeyTrk on the PITCH page. Therefore, you'll need to check the KeyTrk value on both pages to see how key tracking works within a program.

Velocity Tracking (VelTrk)

This is another common DSP control parameter. As with the other parameters on the KEYMAP page, this shifts the position of the keymap. Different attack velocities will play different pitch shifts of the sample root assigned to that note range. If the shift is great enough, the next higher or lower sample root will be played, which in some cases (like the drum programs) will play an entirely different sound. Positive values will play higher pitches of the sample root when you use hard attack velocities (they shift the keymap downward), while negative values will play lower pitches.

Sample Skipping (SmpSkp)

Sample skipping is a new feature in V2 K2500 software that allows for increased upward transposition of samples. By using a new sample playback algorithm, the K2500 increases the maximum playback rate of a sample from 96 Khz to a maximum of 192 Khz. Needless to say, though, there is a tradeoff. Unwanted artifacts may be added to a note's sound as the result of sample skipping. Therefore, the Auto value for this parameter is usually the best choice. Auto means that the keymap in this layer will only employ sample skipping for those notes whose upward transposition can be increased. Notes below a certain point would not benefit from sample skipping and, therefore, Auto ensures that these notes will not use the feature unnecessarily. The only disadvantage to using Auto sample skipping is that you cannot pitchbend a note from below the cutoff point into the range of the sample skipped notes.

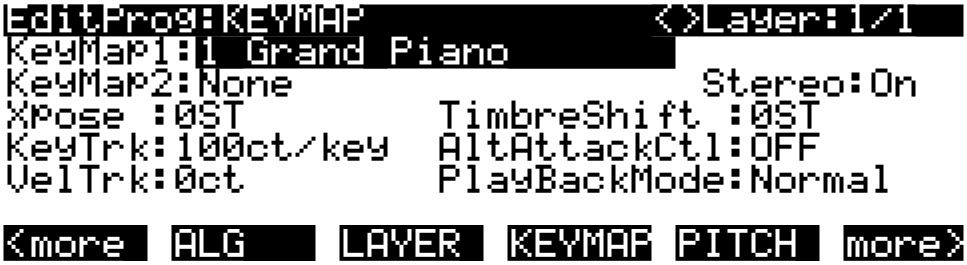
On means that sample skipping will be employed throughout the range of the keymap. This eliminates the pitchbend limitation described above, but may add some artifacts to the sound. Creative types may appreciate this form of distortion, however, so we've made it available.

Off means that sample skipping will not be used at all. Essentially, this means that sample fetching will be done the same way it was done in pre-V2 software.

Stereo

You'll use this parameter when you're working with stereo samples. When you use the optional stereo piano programs (included with the RMB-2 daughter board) or load stereo samples from disk, the K2500 views both sides of the sample as a single sample object. When you select a stereo sample as the value for the Sample parameter, you'll see the letter "S" as part of the sample name (for example, "204*StratoBlaster E3 S".

When you set this parameter to On, the KEYMAP page changes slightly:



```
EditProg:KEYMAP                <>Layer:1/1
KeyMap1:1 Grand Piano
KeyMap2:None                     Stereo:On
Xpose  :0ST        TimbreShift  :0ST
KeyTrk:100ct/key   AltAttackCtl:OFF
VelTrk:0ct         PlayBackMode:Normal
```

An additional Keymap parameter appears. The two keymap parameters are distinguished as Keymap 1 and Keymap 2. The KEYMAP page parameters will affect both keymaps. When the Stereo parameter is on, the OUTPUT page for the current layer will show an additional pair of Pan parameters.

To get the samples to play together, set the Stereo parameter to On, and select the keymap as the value for *both* the Keymap 1 and Keymap 2 parameters. The K2500 automatically uses the left side for Keymap 1, and the right side for Keymap 2. Then go to the OUTPUT page and set the panning for each sample as desired. Keep in mind that using stereo keymaps reduces the polyphony of the program. For example, if you had a two-layer program with stereo keymaps in each layer, each note you play would use 4 of your 48 voices, allowing a total of 12 notes before all the voices have been used.

This parameter is unnecessary if you're not using stereo samples, and should be set to Off in that case.

Timbre Shift

This parameter works only on multi-sample keymaps, and changes the root selection for each key you play. With this parameter you can radically alter the current layer's timbre (basic sound characteristics). The nature of the change depends on the timbre itself, so this parameter calls for experimentation. Basically, timbre shifting changes a note’s timbre by imposing different harmonic qualities onto the note. A timbre-shifted note retains its original pitch, but its harmonics are those of the same timbre at a higher or lower pitch. Positive values for this parameter tend to brighten a sound, while negative values darken.

Here’s an example. If you shift the timbre up 4 semitones, then playing C 4 will result in the *pitch* C 4, but will actually play the sample normally assigned to G# 3, and shift its pitch up 4 semitones. This will increase the playback rate of the sample, so although the pitch remains normal, the timbre is brighter. You’d get the same effect by setting the Xpose parameter on the Keymap page to -4 semitones, then setting the Adjust on the PITCH page to + 4 semitones. For multi-sample layers with narrow key ranges, large amounts of timbre shifting will cause different sample roots to be played back.

Playback Mode

This gives you four options for manipulating the samples in the current layer as you trigger them. Normal leaves the samples unaffected, while Reverse plays them in reverse. At a value of Reverse, the samples will continue to loop as long as notes are sustained. To play them just once in reverse, you would adjust the length of the layer’s amplitude envelope (explained later in this chapter). BiDirect (bidirectional) causes the samples to loop infinitely, alternating between normal and reversed playback. Finally, Noise replaces the samples with a white noise generator.

Alternative Switch (AltControl and AltMethod)

You can assign a control source to change the sound by using an alternate start point or alternate end point for the current keymap. Whether or not it is an alternate start or alternate end depends on the position of the Alt parameter for the sample (set in the Sample Editor). When set before the end point, it is used as an alternate start (the Alt point can be before or after the normal Start point). When set after the end it is used as an alt end.

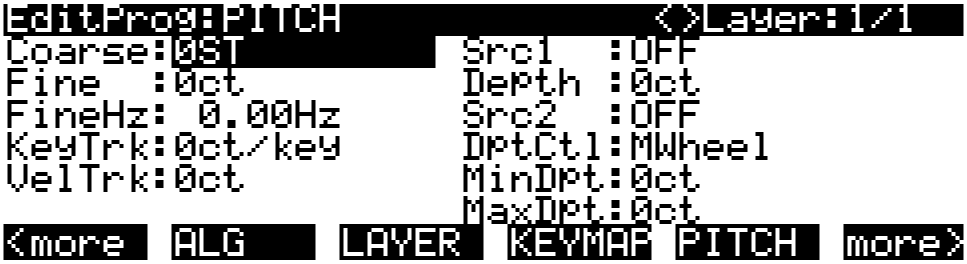
If you place the Alt point after the initial attack transients of the sample, then you can use the Alt Switch to emulate legato playing in an acoustic instrument. As an example, set the Keymap to "#14 Flute". Now set the AltControl parameter to Chan St (Channel State). Now if you play notes separately, the initial breathy chiff will be heard. But if you play the notes legato (connecting them smoothly), the Alt point is used and you do not hear the chiff. This is because the Chan St is turned on as long as any note is being held. Most of the K2500’s ROM samples have their Alt points set for purposes of legato play. In most cases the difference in attacks is subtle, but for some sounds, like drums, the difference can be more noticeable.

The AltMethod parameter lets you choose between continuous or switched interpolation. Continuous interpolation will set the alternate start or end point based on the value (0-127) of the control source. Switched means that the control source will simply switch between using the alternate or normal start or end point, depending on whether the control source transmits a value greater or less than 64.

For more information, refer to the discussion of the TRIM page’s Alt point on page 15-17.

Program Editor— PITCH Page

Press the **PITCH** soft button, and the PITCH page will appear. These parameters adjust the pitch (playback rate) of the samples after the root has been selected by the keymap.



```
EditProg:PITCH                  <>Layer:1/1
Coarse:0ST            Src1  :OFF
Fine  :0ct            Depth :0ct
FineHz: 0.00Hz        Src2  :OFF
KeyTrk:0ct/key        DptCtl:MWheel
VelTrk:0ct            MinDpt:0ct
                      MaxDpt:0ct
<more   ALG    LAYER  KEYMAP  PITCH  more>
```

PARAMETER	RANGE OF VALUES	DEFAULT
COARSE ADJUST	–120 to 60 ST (semitones)	0
FINE ADJUST (cts)	± 100 cents	0
FINE ADJUST (Hz)	± 6 Hz	0
KEY TRACKING	± 2400 cents per key	0
VELOCITY TRACKING	± 7200 cents per key	0
SOURCE 1	Control Source list	OFF
DEPTH	± 7200 cents	0
SOURCE 2	Control Source list	OFF
DEPTH CONTROL	Control Source list	MWheel
MINIMUM DEPTH	± 7200 cents	0
MAXIMUM DEPTH	± 7200 cents	0

These parameters were described in the section on common DSP control parameters earlier in this chapter, so we won’t repeat them here, but a word about the Fine Hz parameter is in order.

Fine Hz

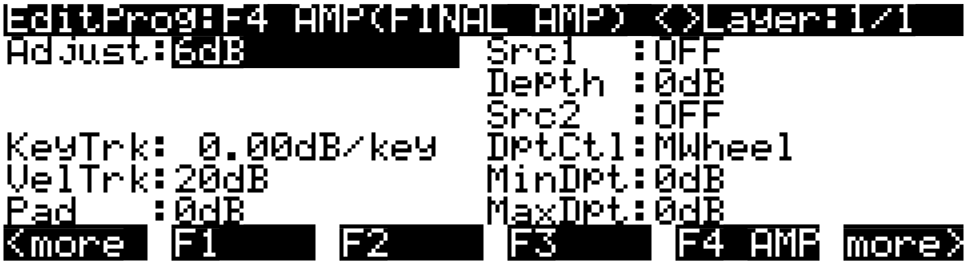
This measures pitch adjustment by the relative frequency (in Hertz) of each note. This is useful for controlling the beat frequency between layers in a multi-layered program. Using this parameter to detune chorusing layers will keep the beat frequency constant across much of the keyboard. Although the ratio of frequencies between each layer remains constant, the detuning will increase at lower pitches, and can become extreme. The K2500 automatically limits the amount of detuning when it becomes extreme, so you’ll notice the beat frequencies moving out of sync when you play low pitches.

Program Editor—F1–F3 Pages

These pages are reached by pressing the **F1**, **F2**, and **F3** soft buttons, respectively. They contain the parameters governing the three variable DSP functions in each algorithm. The pages vary depending on the DSP functions selected for the three middle DSP control inputs, represented by the downward-pointing arrows on the ALG page. See the *Reference Guide* for a complete list of the algorithms and their available DSP functions.

Program Editor—F4 AMP Page

Press the **F4 AMP** soft button to call up this page, which features five of the six common DSP control parameters, in this case controlling the final amplitude of the current layer before it reaches the audio outputs. There's also a parameter that enables you to pad (attenuate) the current layer's signal before its final amplification.



```
EditProg:F4 AMP(FINAL AMP) <>Layer:1/1
Adjust:6dB           Src1  :OFF
                     Depth :0dB
                     Src2  :OFF
KeyTrk: 0.00dB/key   DptCtl:MWheel
VelTrk:20dB          MinDpt:0dB
Pad   :0dB           MaxDpt:0dB
<more   F1     F2     F3     F4 AMP  more>
```

PARAMETER	RANGE OF VALUES	DEFAULT
ADJUST	**–96 to 48 dB**	**6**
KEY TRACKING	**± 2 dB**	**0**
VELOCITY TRACKING	**± 96 dB**	**20**
PAD	**0, 6, 12, 18 dB**	**0**
SOURCE 1	**Control Source list**	**OFF**
DEPTH	**± 96 dB**	**0**
SOURCE 2	**Control Source list**	**OFF**
DEPTH CONTROL	**Control Source list**	**MWheel**
MINIMUM DEPTH	**± 96 dB**	**0**
MAXIMUM DEPTH	**± 96 dB**	**0**

Adjust

Adjust the overall amplitude (gain) of the currently selected layer. In multi-layer programs, this parameter adjusts the amplitude of the layers relative to each other. This is the final output control for the layer (Post-amp pad). Be careful not to set this too high! If one of your layers is too loud, it's generally better to cut its level than to boost the others. This will keep distortion to a minimum.

Key Tracking

This uses the MIDI note numbers of the notes you play as a control source affecting the individual amplitudes of each note in the current layer. Positive values increase the amplitude as you play higher-pitched notes. For example, if the key tracking is .20 dB/key, then C#4 will be .20 dB louder than C 4 if triggered with the same attack velocity. If the value for this parameter were negative, C#4 would have less amplitude than C 4. A word of caution here: values above 0.30 dB/key (or below –0.30) can generate extremely high amplitude levels. If you set this parameter that high without lowering the value of the Adjust parameter to -12 dB or lower, your sound may clip, which can be useful, but it isn't necessarily what you want.

Velocity Tracking

This uses the MIDI attack velocity value of the notes you play as a control source affecting the individual amplitudes of each note in the current layer. This is the primary parameter to use for adjusting the dynamics of a layer. At a value of 0, every note in the current layer would have the same amplitude, regardless of its attack velocity. When the value is positive, note amplitude increases as attack velocity increases. When the value is negative, note amplitude *decreases* as attack velocity increases. Larger values increase the range between minimum and maximum amplitude, so with a large positive value, the amplitude will be low when you play softly. Small values decrease the range between min and max, so with a small positive value, you'll get nearly full amplitude even with light attack velocities.

Pad

Select one of four attenuation levels for cutting the amplitude of the current layer *before* the final amp stage (Pre-amp pad). Use the pad if the layer's sound distorts when played. Note: clipping can occur in earlier algorithm blocks as well. If this is the case, you'll probably want to try to remove the clipping in the earlier block, if possible.

Sources 1 and 2, Depth Controls

These are common DSP control parameters, which in this case let you assign control sources to affect the amplitude of the current layer. The functions of common DSP control parameters are explained in their own section earlier in this chapter.

Program Editor—OUTPUT Page

This page is reached by pressing the **OUTPUT** soft button. This is where you route the signal to the eight separate outs, and to the MIX outputs, with or without passing through the effects processor on the way. If you're using the MIX outputs, select Output Group A if you want the layer to appear at the MIX outputs with effects, or Output Group B, C, or D if you want the layer to appear dry at the MIX outputs.

Before we look at the OUTPUT page, we'll take a quick look at the K2500's audio outs. You will see there are ten, configured as pairs labeled Mix, A, B, C, & D. It is important to understand that even though there are ten jacks, there are only 8 routable outs. Here is the way it works: When you route a signal, you choose the A, B, C, or D pairs and the appropriate panning position. But all audio signals come out of the Mix pair, until you physically plug a cable into a separate out. At that point, any signal routed to that out is removed from the Mix and comes out of that particular out. So if you plug cables into all of the separate outs, there will be no signal coming from the Mix outs.

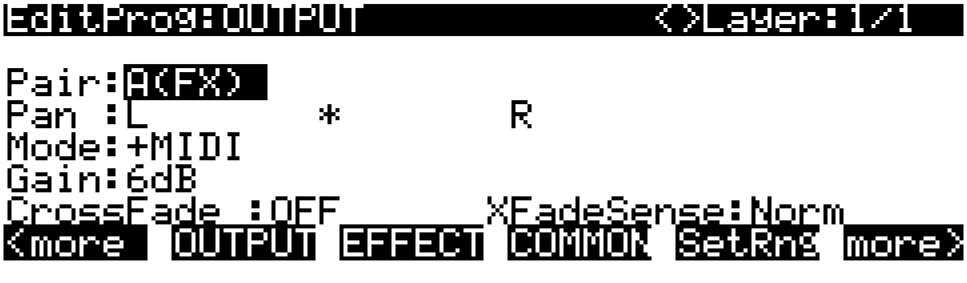
To route something through the standard effects processor in your 2500, you need to set the output to A. But the effects actually only come out of the MIX outs. So if you plug cables into the A outs then, you will not be able to use the effects. For this reason, a typical way users will wire their units is to plug into the MIX, B, C, and D pairs.

The OUTPUT page gives you broad control over the audio signal. You can adjust the output routings of every layer in every program, enabling you to take maximum advantage of the K2500's flexible audio output capabilities.

There are actually four different configurations of the OUTPUT page. The one you see depends on whether the current layer uses a stereo keymap, and whether it uses a double-output algorithm. A double-output algorithm is one whose signal path is split into two parts before final amplification.

Regardless of the page's configuration, there are parameters for adjusting the Output Group, the Pan position, the Output Mode, the Gain, the Crossfade control, and the Crossfade sense. Layers that use stereo keymaps, or that use double-output algorithms, have additional sets of Output Group and Pan parameters on their OUTPUT pages.

The following page is for a layer with one keymap and a single-path algorithm.



```
EditProg:OUTPUT                 <>Layer:1/1

Pair:A(FX)
Pan :L        *        R
Mode:+MIDI
Gain:6dB
CrossFade :OFF        XFadeSense:Norm
<more  OUTPUT  EFFECT  COMMON  SetRng  more>
```

PARAMETER	RANGE OF VALUES	DEFAULT
PAIR	**A(FX), B(DRY), C(DRY), D(DRY)**	**A(FX)**
PAN	**Left to Right (15 positions)**	**Center**
MODE	**Fixed, +MIDI, Auto, Reverse**	**+MIDI**
GAIN	**–12 to 30 dB (6 dB increments)**	**6 dB**
CROSSFADE CONTROL	**Control Source list**	**OFF**
CROSSFADE SENSE	**Normal, Reversed**	**Normal**

Pair

This parameter defines the Output Group of the current layer—that is, which group of audio outputs the layer uses. If the layer is assigned to Pair A, for example, its audio signal will appear at the Group A outputs. The signal will also appear at the mixed outputs, with effects applied, if there are no cables inserted into the Group A outputs.

Pan

Use this parameter to position the current layer’s audio signal between the left and right outputs of whichever Output Group they’re assigned to.

Mode

When the mode is Fixed the pan position remains as defined with the Pan parameter, ignoring MIDI pan messages. When the mode is +MIDI, MIDI pan messages (MIDI 10) will shift the sound to the left or right of the Pan parameter setting. Message values below 64 shift it left, while those above 64 shift it right. A setting of Auto assigns the pan setting of each note based on its MIDI note number. In this case, Middle C (MIDI note number 60) is equivalent to the Pan parameter’s setting. Lower notes shift increasingly left, while higher notes shift increasingly right. A setting of Reverse shifts low notes right, and high notes left. MIDI pan messages will also affect the pan position when values of Auto and Reverse are selected.

NOTE: If you’re using the PANNER DSP function in the algorithm for any of the layers in a program, that layer will respond to MIDI pan messages even if the Mode parameter is set to a value of Fixed. That’s just the way it is.

Gain

Boost (or cut) the amplitude of the current layer. For layers using double-output algorithms, the gain is divided evenly between the two signal paths. Since this gain is not affected by the layer’s amplitude envelope, you can use it to add a constant amount of gain to a layer.

Crossfade, Crossfade Sense (XFadeSense)

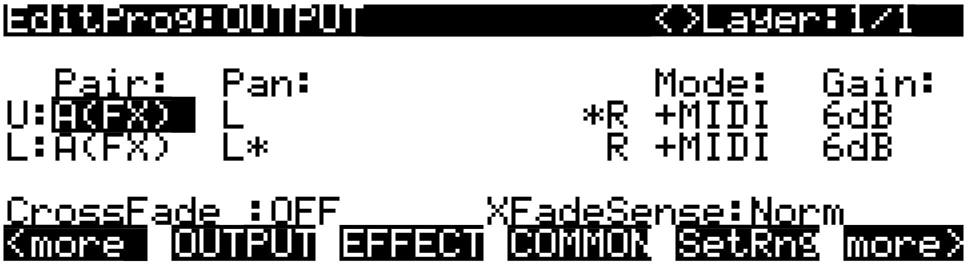
The Crossfade parameter lets you select a control source to fade the current layer’s amplitude from zero to maximum. When Crossfade Sense is Normal, the layer is at full amplitude when the Crossfade control is at minimum. With Crossfade Sense set to Reverse, the layer is at zero amplitude when the Crossfade control is at minimum.

This parameter is similar to the Src1 and Depth parameters on the F4 AMP page, but the attenuation curve for the Crossfade parameter is optimized specifically for crossfades.

To crossfade two layers in the same program, assign the same control source for the CrossFade parameters in both layers, then set one of their XFadeSense parameters to a value of Norm, and the other’s to Rvrs.

Other OUTPUT Page Configurations

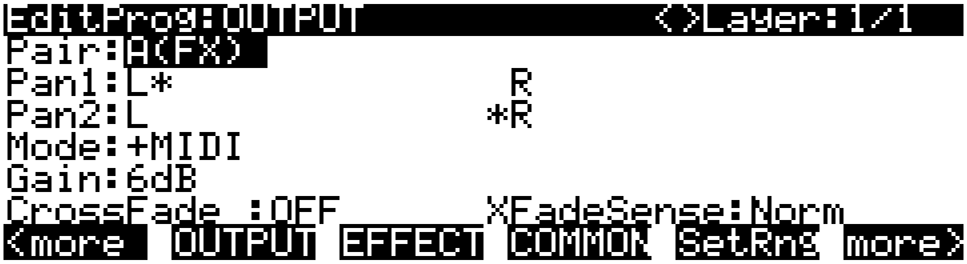
The following page is for a layer with one keymap and a double-output algorithm. The U and L stand for the upper and lower wires (signal paths). You have independent control of the output parameters for each wire.



```
EditProg:OUTPUT                  <>Layer:1/1

  Pair:   Pan:                   Mode:   Gain:
U:A(FX)   L                   *R +MIDI   6dB
L:A(FX)   L*                   R +MIDI   6dB

CrossFade :OFF          XFadeSense:Norm
<more  OUTPUT  EFFECT  COMMON  SetRng  more>
```

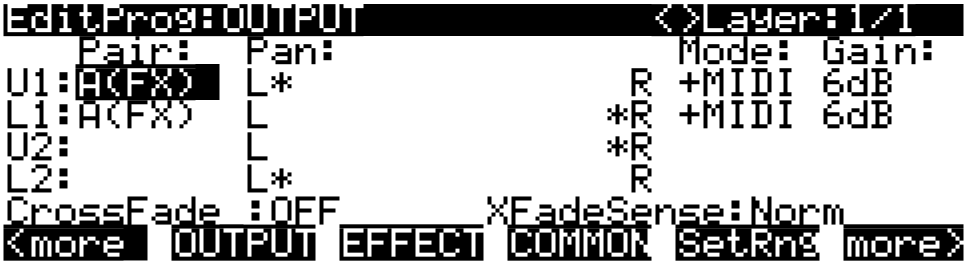
Next are the two page configurations for layers with stereo keymaps: the first one uses a single-output algorithm, and the second uses a double-output algorithm.

With a single-output algorithm, stereo keymap layers let you adjust the pan position of each keymap, but all other parameters are identical for both keymaps.



```
EditProg:OUTPUT                  <>Layer:1/1
Pair:A(FX)
Pan1:L*              R
Pan2:L              *R
Mode:+MIDI
Gain:6dB
CrossFade :OFF          XFadeSense:Norm
<more  OUTPUT  EFFECT  COMMON  SetRng  more>
```

When a stereo keymap layer uses a double-output algorithm, both keymaps are split between the upper and lower wires. In other words, both wires carry the signal from each of the keymaps. The Output Group (Pair), Output mode, and Gain level of Keymap 1 are mimicked by Keymap 2 (that's why these parameters aren't displayed for Keymap 2 on the OUTPUT page). You can, however, set the pan positions independently for the upper and lower wires of both keymaps.

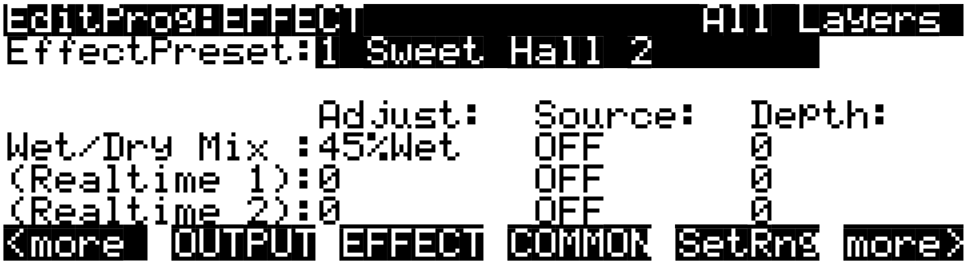


```
EditProg:OUTPUT                  <>Layer:1/1
   Pair:   Pan:                   Mode: Gain:
U1:A(FX)   L*                   R +MIDI 6dB
L1:A(FX)   L                   *R +MIDI 6dB
U2:        L                   *R
L2:        L*                   R
CrossFade :OFF          XFadeSense:Norm
<more  OUTPUT  EFFECT  COMMON  SetRng  more>
```

Program Editor—EFFECT Page

On the EFFECT page you'll adjust the parameters that control the preset effect that the global effects processor applies to the currently selected program. Whether or not you will hear the effect that you assign on this page is determined by the settings of the FX Mode and FX Chan parameters, found in the Effects Mode; they're described on page 9-2.

Press the **EFFECT** soft button and the EFFECT page appears:



```
EditProg:EFFECT                 All Layers
EffectPreset:1 Sweet Hall 2

              Adjust:    Source:    Depth:
Wet/Dry Mix :45%Wet      OFF        0
(Realtime 1):0           OFF        0
(Realtime 2):0           OFF        0
<more   OUTPUT  EFFECT  COMMON  SetRng  more>
```

The top line of the page gives you the usual mode reminder, and shows that the preset effect you select will apply to all layers of the program. Keep in mind that these parameters are global—that is, they affect every part of the program's sound that goes through the global effects processor. The display reminds you of this by showing "All Layers" in the top line.

The Effect Preset parameter determines which preset effect will be selected when you select the program in Program mode.

The remaining three parameters—Adjust, Source, and Depth—apply to three inputs that can be used to modify the preset effect in real time, using any of the control sources. The inputs are the Wet/Dry Mix, and two other realtime inputs that vary with the preset effect—chorus delay, reverb time, and other similar multi-effects DSP functions.

The Adjust parameter sets the initial level for the three inputs. This defines the status of the effect when no control source signal is applied to the inputs. For the Wet/Dry Mix, the range of levels is 0 to 100%. For the two variable inputs, the ranges are –128 to 127.

This Source parameter lets you define which control source is used to modify each of the inputs. The values are taken from the Control Source list.

The Depth parameter defines how much you can modify each input with its assigned control source. The range for each input is –128 to 127.

Wet/Dry Mix

The Wet/Dry Mix parameter determines how much of the preset effect is applied to the program. A value of 0% leaves the program dry; no effect is heard. A value of 100% removes all of the dry signal, and you hear a full effect level on the sound. For best results, start with values of near 50%, and tweak them from there.

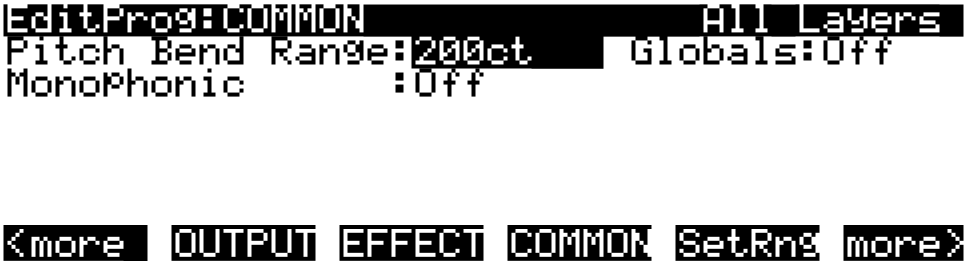
Realtime 1 and 2

These two inputs differ with each preset effect. When you select different preset effects, the names of these two inputs will change to reflect the effect parameter that can be modified via the control source you've assigned with the Source parameter.

When you're recording sequences and including realtime effects changes in the sequence, be sure that the realtime effects messages occur at least 200 milliseconds after program or setup changes. It takes the K2500 approximately 200 milliseconds to switch preset effects, and realtime effects events can interfere with the selection of the new effect. The most likely result is that the realtime effects messages will be ignored.

Program Editor—COMMON Page

Here’s where you find six frequently-used parameters that affect the entire current program, not just the current layer. The COMMON page is reached by pressing the **COMMON** soft button in the Program Editor.



```
EditProg:COMMON                         All Layers
Pitch Bend Range:200ct      Globals:Off
MonoPhonic      :Off
```

Notice that when the Monophonic parameter is set to its default value of Off, the four monophonic parameters do not appear on the page.

PARAMETER	RANGE OF VALUES	DEFAULT
PITCH BEND RANGE	± 7200 cents	200 cents
MONOPHONIC	Off, On	Off
(LEGATO PLAY)	Off, On	Off
(PORTAMENTO)	Off, On	Off
(PORTAMENTO RATE)	1 to 3000 keys per second	70
(ATTACK PORTAMENTO)	Off, On	On
GLOBALS	Off, On	Off

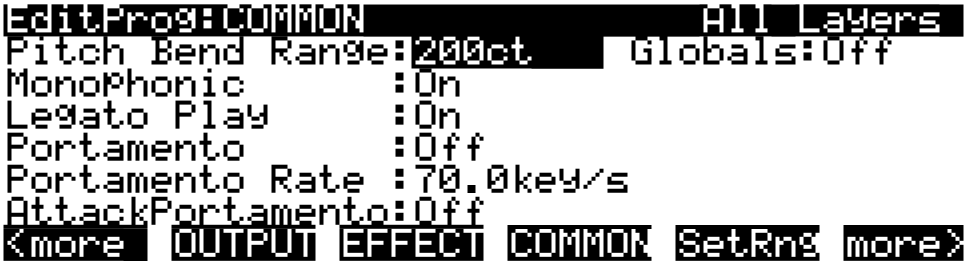
Pitch Bend Range

Use this parameter to define how much the pitch will change when you move your MIDI controller’s Pitch Wheel. Positive values will cause the pitch to bend up when the Pitch Wheel is pushed up, while negative values will cause the inverse. Large positive values can cause samples to bend to their maximum upward pitch shift before the Pitch Wheel is fully up. This will not happen when bending pitch down.

Monophonic

When off, the program is polyphonic—it can play up to 48 notes at a time. Notice that when the Mono Mode parameter is off, the three Portamento parameters do not appear on the COMMON page. This is because only monophonic programs can use portamento.

When On, the program will play only one note at a time. This makes it possible to use portamento, so the remaining parameters appear.



```
EditProg:COMMON                         All Layers
Pitch Bend Range:200ct      Globals:Off
MonoPhonic      :On
Legato Play     :On
Portamento      :Off
Portamento Rate :70.0key/s
AttackPortamento:Off
```

Legato Play

When Legato Play is on, a note will play its attack only when all other notes have been released. This is useful for realistic instrumental sounds.

Portamento

This parameter is either on or off. The default value of Off means that portamento is disabled for the current program.

Portamento is a glide between pitches. On actual acoustic instruments like violin and bass, it's achieved by sliding a finger along a vibrating string. On most keyboards that offer portamento, it's achieved by holding down a key that triggers the starting note, then striking and releasing other keys. The pitch glides toward the most recently triggered note, and remains at that pitch as long as the note remains on. The K2500 gives you two ways to get portamento. See the Attack Portamento parameter below.

When you're applying large amounts of portamento to multi-sampled sounds (Acoustic Guitar, for example), the K2500 will play more than one sample root as the pitch glides from the starting pitch to the ending pitch. This may cause a small click at each sample root transition. You can reduce the number of clicks you'll hear by entering the Program Editor and adjusting the KeyTrk parameter on both the KEYMAP and PITCH pages. The quickest way is to set the KeyTrk value on the KEYMAP page to 0, and to 100 on the PITCH page. This will stretch the sample root that plays at C 4 across the entire keyboard. Now any amount of portamento will play only one sample root, and the clicks will disappear.

There's a tradeoff here, since many sounds will change in timbre as these single sample roots are pitch-shifted during the portamento. This will be most noticeable for acoustic instrument sounds, and may not be noticeable at all for single-cycle waveforms like sawtooth waves. Furthermore, some samples will not glide all the way up to the highest notes—there's a limit to the amount of upward pitch-shifting that can be applied to samples. If this doesn't work for you, you can compromise between the number of clicks and the amount of timbre change by further adjusting the KeyTrk parameters on the KEYMAP and PITCH pages.

As long as the combined values of the KeyTrk parameters on both pages add up to 100, you'll have normal semitone intervals between keys. If you set both parameters to values of 50, for example, the sound will still play normally, and you'll have several sample roots (about half the number of the original sound) stretched evenly across the keyboard, instead of just one. This will give you fewer clicks than in the original sound, but not as much change in timbre as setting the KEYMAP KeyTrk value all the way to 0. Set the KEYMAP KeyTrk parameter higher to reduce the change in timbre, or set the PITCH KeyTrk value higher to reduce the number of clicks. Just make sure the combined values add up to 100, to preserve the normal intervals between notes.

Portamento Rate

The setting for Portamento rate determines how fast the current note glides from starting pitch to ending pitch. The value of this parameter tells you how many seconds the note takes to glide one semitone toward the ending pitch. At a setting of 12 keys/second, for example, the pitch would glide an octave every second. The list of values is non-linear; that is, the increments get larger as you scroll to higher values.

Attack Portamento

This parameter toggles between two types of portamento. When set to On, the K2500 remembers the starting pitch so you don't have to hold a note on to achieve portamento. The pitch always glides to each new note from the previously triggered note. When set to Off, the pitch will glide to the most recently triggered note only when the previous note is still on (in other words, you must use legato fingering).

Globals

This is another toggle, which affects LFO2, ASR2, and FUNs 2 and 4. When off, these four control sources are local; they affect each individual note in the layers that use them as a control source. They begin operating each time a note in that layer is triggered.

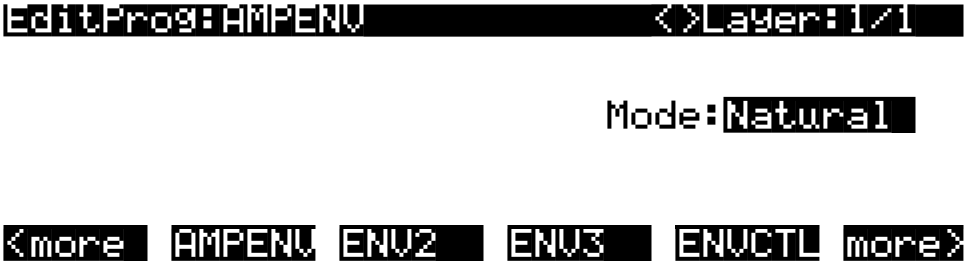
When the Globals parameter is set to On, these control sources become global, that is they affect every note in every layer of the current program, not just the one to which they're applied. When these control sources are global, they begin operating as soon as the program is selected. When Globals are on, LFO2, ASR2, and FUNs 2 and 4 will appear on their respective pages preceded by the letter "G" to indicate that they're global.

You'll use global control sources when you want to affect each note in a given layer uniformly, and local control sources when you want to affect each layer's note independently. For example, you'd use a global LFO controlling pitch to create a ***Leslie effect*** on an organ sound, since you want the affect applied to all the notes you play. You'd use a local LFO controlling pitch to create a vibrato for a solo violin, since you want to be able to vary the rate and depth of the vibrato for each note.

Program Editor—Amplitude Envelope (AMPENV) Page

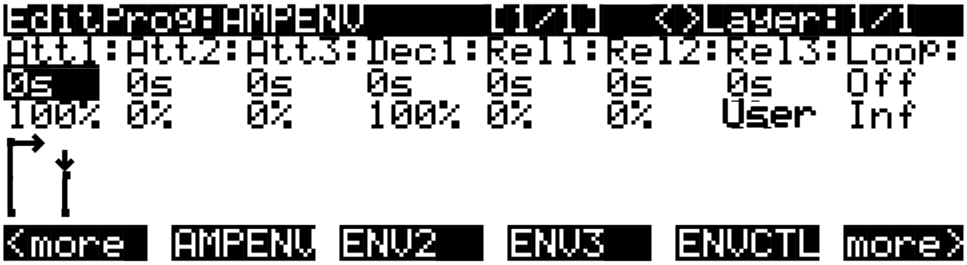
Amplitude envelopes have three sections: attack, decay, and release. The attack section determines how long each note takes to reach its assigned amplitude level after you trigger a Note On event. The decay section determines how quickly and how much a sustained sound fades before a Note Off is triggered. The release section determines how quickly a sound fades to silence *after* a Note Off is triggered.

Press the **AMPENV** soft button to reach the Amplitude Envelope page. For many programs, it will look like the diagram below, which tells you that the amplitude for the current layer is the default, "natural" ROM amplitude envelope that's applied to each sample and waveform during its original development process. You'll leave the amplitude envelope in Natural mode when you don't want to change the way the current layer's loudness develops.



```
EditProg:AMPENV                 <>Layer:1/1

                         Mode:Natural
```

If you want to build your own amplitude envelope, just turn the Alpha Wheel a click. The word "Natural" will change to "User," and a set of AMPENV parameters will appear. The sound will change when you do this, because the default settings for the User envelope, as shown in the diagram below, take effect as soon as you leave Natural mode. Returning to Natural mode applies the original amplitude envelope once again.



```
EditProg:AMPENV      [1/1]   <>Layer:1/1
Att1:Att2:Att3:Dec1:Rel1:Rel2:Rel3:Loop:
0s   0s   0s   0s   0s   0s   0s   Off
100% 0%   0%   100% 0%   0%   User Inf
```

You'll tweak the parameters on the AMPENV page when you want to shape the amplitude characteristics of your sounds. A graphic view of the amplitude envelope will appear on the display to give you a visual sense of the envelope's characteristics. The dots along the envelope graphic indicate the breakpoints between the envelope's various segments. The small horizontal arrow represents the end of the decay section. The small downward-pointing arrow represents the beginning of the release section.

Because the K2500's ROM samples are stored in a compressed format, applying an altered amplitude envelope can change more than just the amplitude of your sound, since it also changes the rate at which the samples are decompressed for playback. When the samples are made to play back with altered envelopes, the timbres can evolve in new and interesting ways.

The AMPENV page's top line gives you the usual location reminder, points out the currently selected layer, and tells you the relative scale of the envelope's graphic view. The envelope graphic shrinks in scale as the segment times get longer. This auto-zoom feature maximizes the available display space. Try lengthening one of the segment times. The envelope graphic will stretch to fill the display from left to right. When it fills the display, it will shrink to half its size, and the top line will indicate that the scale has changed (from [1/1] to [1/2], for example).

Each parameter on this page has two values, as listed below. For the envelope segments, the first (upper) value is the duration of the segment, and the second is the amplitude level at the completion of the segment. For the Loop parameter, the values define how the envelope loops, and how many times the loop cycles.

PARAMETER		RANGE OF VALUES
ATTACK SEGMENT 1, 2, 3	TIME	0 to 60 seconds
	LEVEL	0 to 100 %
DECAY SEGMENT	TIME	0 to 60 seconds
	LEVEL	0 to 100 %
RELEASE SEGMENT 1, 2, 3	TIME	0 to 60 seconds
	LEVEL	0 to 100 %
LOOP	TYPE	Off, Forward, Bidirectional
	# OF LOOPS	Infinite, 1 to 31 times

Attack Segment Times

These indicate how long it takes for the current layer's amplitude to reach its final level from its starting level.

Attack Segment Levels

These are the final levels that each segment achieves at completion. The levels are expressed as percentages of the maximum possible amplitude for the current layer. Attack segment 1 always starts at zero amplitude, and moves to its assigned level in the time specified by its time value. So the default settings of 0 seconds and 100% mean that the first segment of the attack section moves instantly from zero amplitude to 100% amplitude. Increase the time of Attack segment 1 if you want the sound to ramp up more slowly.

Attack segments 2 and 3 affect the sound only when you set a non-zero value for time. They will then move to their assigned levels in the time specified. Their starting levels are equal to the final levels of the preceding segment.

Decay Segment

The decay section has only one segment. It has values for time and level, just as for the attack section. The decay section begins as soon as the attack section has been completed. It starts at the same amplitude level as the attack segment preceding it, and moves to its assigned level in the time specified. You'll hear a note's decay section only when the attack section is completed before a Note Off message is generated for that note.

To create a sustaining envelope, simply set the Decay segment to a non-zero value.

Release Segments

Like the attack and decay sections, each of the three segments in the release section has values for time and level. Each segment reaches its assigned level in the time specified for that segment. Release segment 1 starts at the Note Off event for each note, at the current amplitude level of that note—whether it's in the attack section or the decay section. It then moves to its assigned level in the time specified. Release segments 2 and 3 start at the final levels of the segments before them. Release segments 1 and 2 can be set to any level from 0 to 100%. Release segment 3 always has a level of 0%, so you can't adjust its level. In place of its Level parameter you see a parameter that lets you toggle between User envelopes and the sound's preprogrammed "natural" envelope.

Loop Type

There are seven different values for Loop type.

A value of Off disables looping for the current layer's amplitude envelope.

Values of seg1F, seg2F, and seg3F are forward loops. In each case, the amplitude envelope plays through the attack and decay sections, then loops back to the beginning of the first, second, or third attack segments, respectively.

Values of seg1B, seg2B, and seg3B, are bidirectional loops. The amplitude envelope plays through the attack and decay sections, then reverses and plays backward to the beginning of the first, second, or third attack segment, respectively. When it reaches the beginning of the assigned attack segment, it reverses again, playing forward to the end of the decay section, and so on.

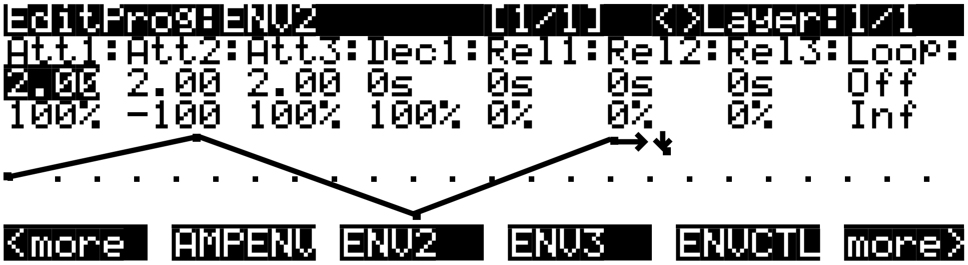
Number of Loops

A value of Inf makes the amplitude envelope loop until a Note Off is generated. Values of 1 through 31 indicate how many times the loop will repeat after the amplitude envelope has played once through its normal cycle.

Regardless of the loop type and the number of loops, each note goes into its release section as soon as its ***Note State*** goes off (that is, when a Note Off is generated). The envelope will continue to loop as long as Note State remains on, whether it's held on by a pedal, by the IgnRel parameter (described in the section entitled "The LAYER Page"), or whatever.

Program Editor—Envelopes 2 and 3

The K2500 offers two envelopes in addition to the amplitude envelope. Like the amplitude envelope, Envelopes 2 and 3 can be assigned like any other control source. The only difference between these two envelopes and the amplitude envelope is that Envelopes 2 and 3 can be bipolar. This means that you can set negative values for them. (Obviously, you can’t have an amplitude less than zero, so the amplitude envelope is unipolar—the values range from 0 to 100%). A bipolar envelope controlling pitch, for example, could modulate the pitch both above and below its original level.

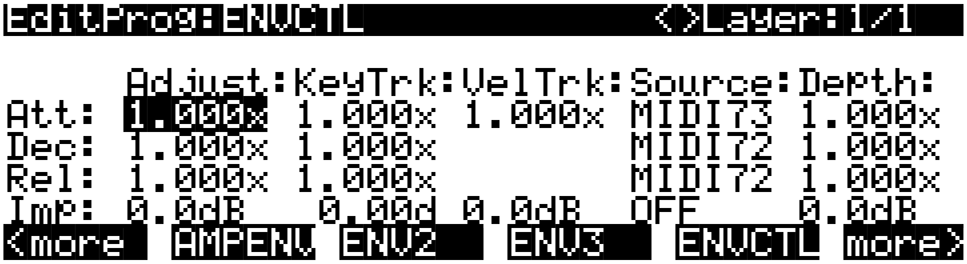


```
EditProg:ENV2           [1/1]   <>Layer:1/1
Att1:Att2:Att3:Dec1:Rel1:Rel2:Rel3:Loop:
2.00 2.00 2.00 0s   0s   0s   0s   Off
100% -100 100% 100% 0%   0%   0%   Inf
```

The pages for Envelopes 2 and 3 are reached with the soft buttons **ENV2** and **ENV3**. When you select these pages, you’ll find a display that looks very much like the AMPENV page. The only differences are that you can program an amount for Rel3, and in the envelope graphic, which has a dotted line running horizontally across the display. This is the zero level line; negative level values for the various envelope segments will cause the envelope graphic to dip below this line.

Program Editor—Envelope Control (ENVCTL) Page

Envelopes are control sources with outputs that evolve over time without repeating (unless you want them to). You can make the envelopes even more powerful by using envelope control. This gives you realtime control over the rates of each section of the envelopes. Press the **ENVCTL** soft button to reach the ENVCTL page.



```
EditProg:ENVCTL                 <>Layer:1/1
     Adjust:KeyTrk:VelTrk:Source:Depth:
Att: 1.000x 1.000x 1.000x MIDI73 1.000x
Dec: 1.000x 1.000x        MIDI72 1.000x
Rel: 1.000x 1.000x        MIDI72 1.000x
Imp: 0.0dB   0.00d 0.0dB  OFF    0.0dB
<more  AMPENV ENV2   ENV3   ENVCTL more>
```

The display's top line reminds you of the current layer. The first line of text in the center of the display shows five of the common DSP control parameters: Adjust, Key tracking, Velocity tracking, and Source/Depth.

This page is a table showing the five envelope control parameters, and their values for each of the three sections of the envelopes. Additionally, the line above the soft buttons lets you make use of the Impact feature, which adds an amplitude overshoot to the first 20 milliseconds of a note's attack. It's important to keep in mind that if you set up an envelope control source, it affects Envelopes 2 *and* 3, as well as the amplitude envelope (Natural *or* User). Furthermore, the values for the various parameters are cumulative. With the exception of Impact, though, ENVCTL does not affect the attack sections of natural envelopes.

The parameters and values in the following list apply to *each* of the three envelope sections—attack, decay, and release. We'll describe them only once, since their functions are largely the same for each envelope section. The only difference is with velocity tracking, which is hard-wired to control only the attack sections of the envelopes (you can assign attack velocity as the value for the Source parameter in each of the sections, however).

The values of each of these parameters multiply the *rates* of the envelope sections they control. Values greater than 1.000x make the envelope sections run *faster* (they *increase* the rate), while values less than 1.000x make the envelope sections run *slower*. Say for example that on the current layer's AMPENV page you had set the Decay section's time at 2.00 seconds, and its level at 0%. This sets the layer's amplitude to fade to silence two seconds after the completion of the last attack segment. The decay *time* is two seconds; the decay *rate* is 50% per second. Now if you select the ENVCTL page and set the Decay Adjust parameter to a value of 2.000x, you've increased the decay *rate* by a factor of two. The rate increases to 100% per second, and the decay time is now one second instead of two.

PARAMETER	**RANGE OF VALUES**
	(Att, Dec, Rel)
ADJUST	**0.018 to 50.000x**
KEY TRACKING	**0.018 to 50.000x**
VELOCITY TRACKING	**0.018 to 50.000x**
SOURCE	**Control Source list**
DEPTH	**0.018 to 50.000x**

Adjust

This is the familiar Coarse adjust found on many other pages. Use it here to change the rate of one of the envelope sections without reprogramming the envelope itself. This parameter doesn't give you realtime control over the envelope. It is, however, a good way to adjust the natural envelopes without switching to a User envelope and trying to approximate the Natural envelope.

Key Tracking

This uses the MIDI note number of each key as the control input for the current layer's corresponding envelope section. When the value of this parameter is greater than 1.000x, notes above C 4 will make the envelope section run faster, while notes below C 4 will make it run slower. When the value of this parameter is less than 1.000x, notes above C 4 will make the envelope section run slower, and notes below C 4 will make it run faster. This gives you realtime envelope control right from your MIDI controller. You might use it, for example, to cause an acoustic guitar sound to decay quicker at the high end (set the key tracking to a positive value).

Velocity Tracking

Use your attack velocity as the control input for the current layer's attack section (this parameter doesn't apply to decay or release). When the value of this parameter is greater than 1.000x, attack velocities greater than 64 make the attack section run faster, and attack velocities below 64 make it run slower. This gives you realtime attack control over the envelope.

Source, Depth

These two parameters work together to let you assign a control like the Mod Wheel to affect the current layer's envelopes in realtime. The value of the Source parameter defines which control affects the envelope section, and the value of the Depth parameter defines how much the rate is multiplied when the control is at its maximum.

Impact

Impact punches the volume during the first 20 milliseconds of the attack of an envelope. Use this feature to get maximum "thump" from your bass and drum sounds. Programs you create that use Impact will not work on a K2000.

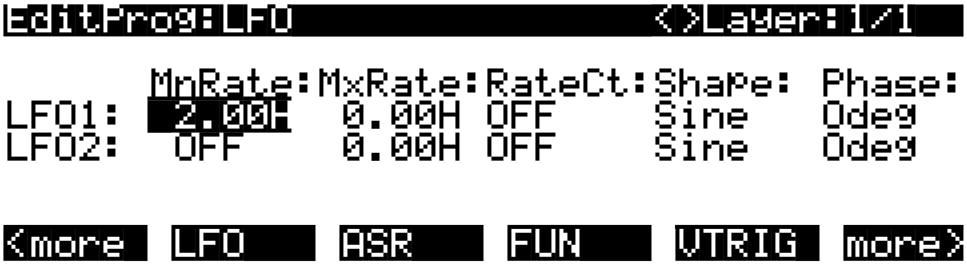
Program Editor—LFO Page

These are low-frequency oscillators. You'll use the LFO page to define the behavior of the two LFOs available to each layer. LFOs are periodic (repeating) control sources. The basic elements are the rate and shape, which define how frequently the LFO repeats, and the waveform of the modulation signal it generates.

With the K2500, you can set upper and lower limits on each LFO's rate, and assign a control source to change the LFO's rate in realtime, if you wish.

Because of its periodic nature, the LFO is perfect for creating effects like vibrato (cyclic variation in pitch) and tremolo (cyclic variation in amplitude). When you're editing LFOs, or any control source, remember that it must be assigned to control some parameter before you'll hear the effects of your edits.

LFO1 is always local, meaning that it's triggered with each Note On event, and runs independently for each note in the layer. LFO2 is local by default, but can be made global. This is done on the COMMON page, by setting the Globals parameter to On, which causes LFO2, ASR2, FUN2 and FUN4 all to become global. Global controls uniformly affect every note in each layer.



```
EditProg:LFO                  <>Layer:1/1

      MnRate:MxRate:RateCt:Shape: Phase:
LFO1:  2.00H  0.00H OFF    Sine   0deg
LFO2:  OFF    0.00H OFF    Sine   0deg

<more  LFO    ASR    FUN    VTRIG  more>
```

The top line of this page gives the usual mode reminder and tells you which layer you're looking at. There are five parameters for each of the LFOs.

PARAMETER	RANGE OF VALUES	DEFAULT
MINIMUM RATE	**0 to 24 Hz**	**2.00 (OFF for LFO2)**
MAXIMUM RATE	**0 to 24 Hz**	**0.00**
RATE CONTROL	**Control Source list**	**OFF**
LFO SHAPE	**LFO Shape list (Ref. Guide)**	**Sine**
LFO START PHASE	**0, 90, 180, 270 degrees**	**0**

Minimum Rate

This is the slowest rate at which the LFO will run. When its Rate control is set to OFF, or when the control source assigned to it is at its minimum, the LFO runs at its minimum rate.

Maximum Rate

This is the fastest possible rate for the LFO. When its Rate control is set to ON, or when the control source assigned to it is at its maximum, the LFO runs at its maximum rate.

Rate Control

Assign any control source in the list to modulate the LFO's rate between its minimum and maximum. A continuous control like the Mod Wheel is a natural choice, enabling you to get just about any rate between min and max. But you can use a switch control too, to get just the min or max with nothing in between. Assigning MPress as the rate control for an LFO vibrato gives you an easy way to increase the vibrato rate in realtime, as you can on many acoustic instruments.

LFO Shape

The shape of the LFO waveform determines the nature of its effect on the signal its modulating. There are diagrams of each LFO shape in the *Reference Guide*; these will give you an idea of how each LFO shape affects the signal. An easy way to check the effects of the different LFO shapes is to set LFO1 as the value for the Src1 parameter on the PITCH page, and set the Depth for Src1 to 400 cents or so. Then go to the LFO page, set the Min and Max rates for LFO1 at 0.00 Hz and 4.00 Hz or so, and set the Rate control to MWheel. Now play your MIDI controller and you'll hear the LFO's rate change when you move its Mod Wheel. Select different LFO Shapes and check out the effect on the pitch.

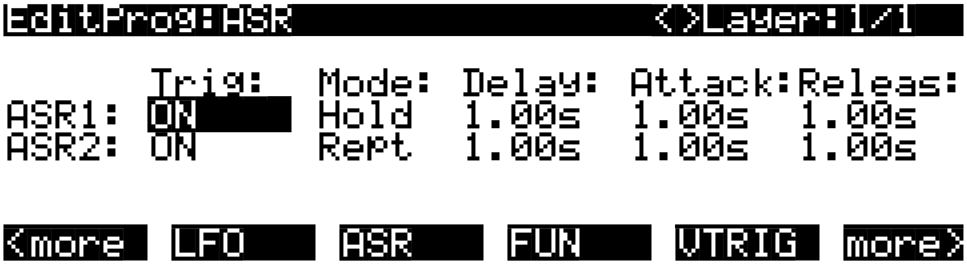
LFO Phase

Use this parameter to determine the starting point of the LFO's cycle. One complete cycle of the LFO is 360 degrees. 0 degrees phase corresponds to a control signal value of 0, becoming positive. Each 90-degree increment in the phase represents a quarter-cycle of the LFO.

When an LFO is local, the phase parameter gives you control over the starting point of the LFO for each note (for example, you could make sure every vibrato started below the pitch you played instead of at the pitch you played). The LFO's phase also affects global LFOs, although it's often indistinguishable, since global LFOs start running as soon as the program containing them is selected, even if you don't play any notes.

Program Editor—ASR Page

ASRs are three-section unipolar envelopes—attack, sustain, and release. The K2500’s ASRs can be triggered by a programmable control source, and can be delayed. ASR1 is always a local control. ASR2 is local by default, but becomes global if the Globals parameter on the COMMON page is set to On. ASRs are frequently used to ramp the depth of pitch or amplitude in a vibrato or tremolo, enabling delays in those effects. Chapter 19 gives an example of creating a delayed vibrato. The ASR page consists of two rows of five parameters, one row for each of the ASRs.



```
EditProg:ASR                    <>Layer:1/1

       Trig:   Mode: Delay: Attack:Releas:
ASR1:  ON      Hold  1.00s  1.00s  1.00s
ASR2:  ON      Rept  1.00s  1.00s  1.00s

<more  LFO  ASR  FUN  VTRIG  more>
```

PARAMETER	RANGE OF VALUES
TRIGGER	Control Source list
MODE	Normal, Hold, Repeat
DELAY	0 to 60 seconds
ATTACK	0 to 60 seconds
RELEASE	0 to 60 seconds

Trigger

This defines the control source that starts the current layer’s ASRs. The ASR starts when the trigger switches from off to on. If the Trigger parameter is set to ON, global ASRs will run as soon as the program containing it is selected. Switch controls are better suited for ASR triggers because of their binary (on/off) nature. A continuous control will trigger the ASRs when its signal value is above its midpoint.

Mode

This parameter sets the sustain section of the ASR. The ASR’s mode determines what the ASR does when it finishes its attack section. If the Mode parameter is set to Normal, the ASR will run directly from its attack section to its release section (no sustain). At a setting of Repeat, the ASR will cycle through the attack and release sections, then loop forward and cycle through again until the ASR’s trigger switches off. If the mode is set to Hold, the ASR maintains its position at the end of the attack section until the ASR’s trigger switches off. The ASR then goes into its release section. If the ASR’s trigger switches off before the attack section is complete, the ASR goes directly to its release section.

Delay

When the ASR’s trigger switches on, the ASR will start immediately if this parameter is set to zero. Non-zero values will cause a corresponding delay between the ASR trigger and the start of the ASR.

Attack

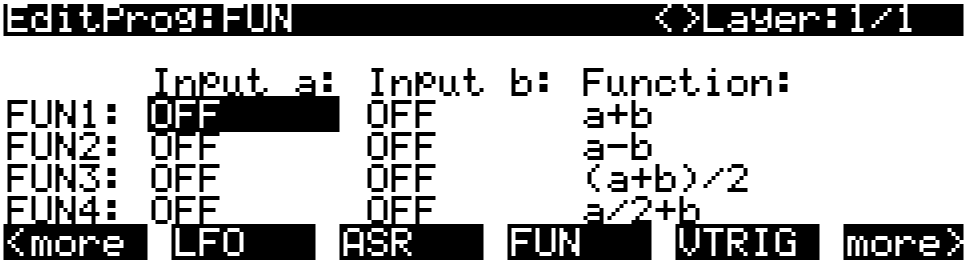
This defines how long the ASR takes to ramp up from minimum to maximum effect on whatever it’s patched to.

Release

This defines how long the ASR takes to fade to minimum from its maximum. If the ASR’s trigger switches off before the ASR has reached maximum, the ASR releases from that level.

Program Editor—FUN Page

“FUN” is short for “function.” The K2500’s four FUNs greatly extend the flexibility of the control sources. Each FUN accepts input from any two control sources, performs a selectable function on the two input signals, and sends the result as its output, which can be assigned like any other control source. Using the FUNs involves defining them on the FUN page, then assigning one or more of them as control sources. The FUN page looks like this:



```
EditProg:FUN                      <>Layer:1/1

       Input a:  Input b:  Function:
FUN1:  OFF       OFF       a+b
FUN2:  OFF       OFF       a-b
FUN3:  OFF       OFF       (a+b)/2
FUN4:  OFF       OFF       a/2+b
```

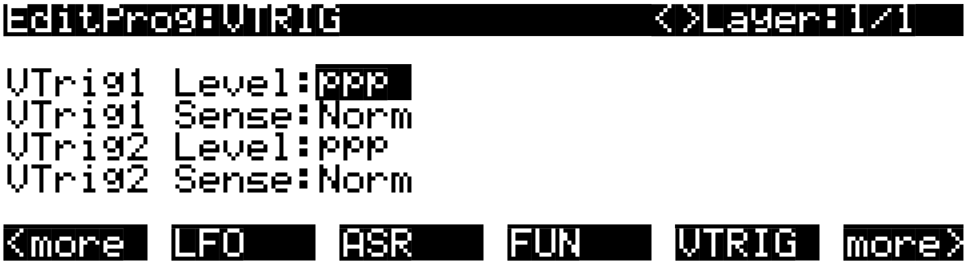
There are three parameters for each FUN. Inputs **a** and **b** can be any control source from the Control Source list. The control sources you want to combine are the ones you’ll assign as the values for these parameters.

The Function parameter determines what mathematical function is applied to the two inputs. When a FUN has been assigned as a control source, the K2500 reads the values of the two control sources defined as Inputs **a** and **b**. It then processes them according to the setting for the Function parameter, and the resulting value is the FUN’s output.

Chapter 16 describes each of these functions, and provides a few diagrams to give you a hint of the immense control (as well as some chaos) that these functions make possible.

Program Editor—VTRIG Page

The velocity triggers base their operation on the attack velocity of each note you play. To use a VTRIG, you simply set its velocity level (threshold), then set it to switch on or off when your attack velocities exceed that threshold. Then assign it as a control source for some other parameter. They’re handy for triggering ASRs, for example.



```
EditProg:VTRIG                  <>Layer:1/1

VTrig1 Level:ppp
VTrig1 Sense:Norm
VTrig2 Level:ppp
VTrig2 Sense:Norm
```

PARAMETER	RANGE OF VALUES	DEFAULT
VEL. TRIGGER LEVEL	ppp to fff	ppp
VEL. TRIGGER SENSE	Normal, Reversed	Normal

The velocity trigger’s level is expressed in terms of the standard dynamic markings of western music—ppp, pp, p, mp, mf, f, ff, and fff. The K2500 converts each attack velocity value it receives into one of these eight levels. When a velocity trigger has been assigned as a control source, the K2500 compares the velocity trigger’s level and sense with the attack velocity values it receives. If the sense is Normal and the attack velocity value is greater than the velocity trigger’s level, the trigger switches on. When the velocity trigger’s sense is reversed, the trigger switches on when the attack velocities it receives are lower than the velocity trigger’s level. Keep in mind that you won’t hear the effect of editing the VTRIG page until you’ve assigned a VTRIG as a control source for some other parameter.

Function Soft Buttons

The remainder of this chapter describes the soft buttons that perform specific functions, as opposed to selecting programming pages. The descriptions below are arranged in the order in which you would see the soft buttons if you pressed the more> button repeatedly. You can always get to these buttons, regardless of which page is currently selected.

Set Range (SetRng)

The **SetRng** soft button gives you a quick way to set the lowest and highest notes in the currently selected layer. Press this button, and the K2500 will prompt you to trigger the note you want to set as the low note for the layer. Press the **Cancel** soft button if you change your mind. Otherwise, trigger the desired note on your MIDI controller. When you trigger a note, the K2500 prompts you to trigger the note you want to be the highest in the layer. When you trigger another note, the previously selected page returns, and the notes you triggered will be recorded as the new values for the LoKey and HiKey parameters on the LAYER page. You'll notice that the higher of the two notes you triggered is entered as the HiKey value, regardless of the order in which you triggered the two notes.

Name

Call up the page that enables you to change the name of the current program.

Save

Start the process of saving the current program.

Delete

Delete the current program from RAM. You can also delete any other RAM program by scrolling through the list that appears when you press the **Delete** soft button, then pressing Delete again when the desired program is selected. If you attempt to delete a ROM program, the K2500 will say it's deleting the program, but it doesn't actually do it.

Dump

Send a MIDI System Exclusive dump of the current program's settings. See the *Reference Guide* for more information about System Exclusive messages.

New Layer (NewLyr)

Create a new layer, numbered one above the highest existing layer. The new layer's parameters are those of the single layer in Program 199, called "Default Program." When you press this button, the K2500 will tell you that it is creating a new layer, then will return to the page you were on. The new layer becomes the current layer. If the current program already has its maximum number of layers, the K2500 will tell you that you can't add any more. If the current keyboard channel is a drum channel, you can add up to 31 additional layers, for a total of 32. Otherwise, the maximum is a total of three.

Program 199 makes a good template for programs that you build from the algorithm up. You might want to edit Program 199 to adjust one or more parameters to values you want to use in your template program. If you like the settings of the default layer as they are, however, remember not to make any permanent changes to Program 199.

Duplicate Layer (DupLyr)

Create a copy of the current layer, duplicating the settings of all its parameters. The copy becomes the current layer, and is numbered one higher than the original.

Import Layer (ImpLyr)

Copy a specific layer from another program into the current program. This button brings up a dialog that prompts you to select a layer number and a program number. The dialog tells you the currently selected layer, and the total number of layers in the program. Use one of the two leftmost soft buttons (or the up/down cursor buttons) to change the layer number. If the current program has only one layer, pressing these buttons will have no effect. Use one of the two center soft buttons (or the left/right cursor buttons) to change the program number.

While you are on this screen, you can listen to the layer you are selecting to import, along with all other layers in the current program. If you want to hear the layer to be imported by itself, you must mute the other layers.

When you have selected the desired layer from the desired program, press the **Import** soft button, and the selected layer will be copied from the selected program, becoming the current layer. Importing layers is a convenient alternative to creating layers from scratch. If you have a favorite string sound, for example, and you want to use it in other programs, just import its layer(s) into the program you're building. This will preserve the envelopes and all the control settings so you don't have to reprogram them.

Delete Layer (DelLyr)

Delete the current layer. When you press this button, the K2500 asks you if you want to delete the layer; press the **Yes** soft button to start the deletion process, or the **No** soft button to cancel it. This prompt prevents you from accidentally deleting a layer.