Chapter 6
Program Mode and the Program Editor

Programs are the K2600's performance-level sound objects. They're preset sounds equivalent to the patches, presets, voices, or multis that you find on other synths.

Program mode is the heart of the K2600, where you select programs for performance and editing. The K2600 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.

In Chapter 2 we briefly discussed the difference between VAST programs and KB3 programs. You'll remember that VAST programs contain up to 32 layers, each of which contains a keymap, which in turn consists of a number of samples assigned to a particular keyboard range—typically five or six notes, depending on the samples.

We mentioned drum programs, which are VAST programs with more than three layers. There's no real difference between "normal" VAST programs and drum programs—consequently this chapter doesn't make any further distinctions between them.

OK, one further distinction: there's no keymap information about drum programs in the info box on the Program-mode page—there simply isn't room for information about more than three layers. If you're wondering why we even *have* the concept of a drum program, it's actually a carryover from the K2000, which had less processing power than the K2600, and required a special channel to handle more than three layers—and you need lots of layers, each with a different sound and keyboard range, to make a convincing drum program. The name stuck.

You'll also recall from Chapter 2 that KB3 programs use a much different architecture: no layers or algorithms, just a bunch of oscillators that start running as soon as you select a KB3 program. This keeps the K2600's sound engine rather busy, and that's why there's a special channel dedicated to KB3 programs; "regular" channels don't have the processing ability to generate that many voices on a constant basis. By default, Channel 1 is the KB3 channel, but you can make any channel the KB3 channel (with the KB3Chan parameter on the Master-mode page).

Background

There's a lot to digest in this chapter, so it might help to explain how we've set it up. The next two sections give more detailed descriptions of the differences in structure between VAST programs and KB3 programs. Then, since there are several performance features (and a few issues) unique to KB3 programs, we'll talk about those (*KB3 Mode* on page 6-4). After that, there are descriptions of the Program-mode features that are common to both types of programs.

Then it's on to the Program Editor. When it comes to editing, there are more differences between the two types of programs than there are similarities, so there are two separate sections: *Editing VAST Programs* on page 6-11, and *Editing KB3 Programs* on page 6-48.

VAST Program Structure

You might want to take a look at Figure 6-1 on page 6-3, which depicts the hierarchy of a VAST program, from individual samples all the way up to setups, which can contain up to eight programs.

Every VAST program contains at least one layer. A layer consists of a keymap and an algorithm for processing the samples contained in the keymap. Samples are stored in the K2600's ROM, or are loaded into Sample 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 some kind of sound: musical, vocal, industrial, any sound at all. Individual samples are assigned to specific key ranges (from A 2 to D 3, for example), and are also assigned to be triggered at specific attack velocities. These assignments constitute the keymap.

When you trigger a note, the K2600 looks to the keymap of each layer of the currently active VAST program(s) to determine which samples to play. The sound engine then fetches the requested samples and generates a digital signal representing the sound of the samples. This signal first passes through the five DSP functions that make up the algorithm. It then passes through the KDFX effects processor, and finally appears—with some level of effects applied to it—at one or more of the audio outputs.

The layer is the VAST program's basic unit of polyphony, that is, each layer constitutes one of the 48 voice channels the K2600 can activate at any time. If you have a program that consists of two layers covering the note range from A 0 to C 8, each key you strike triggers two voice channels.

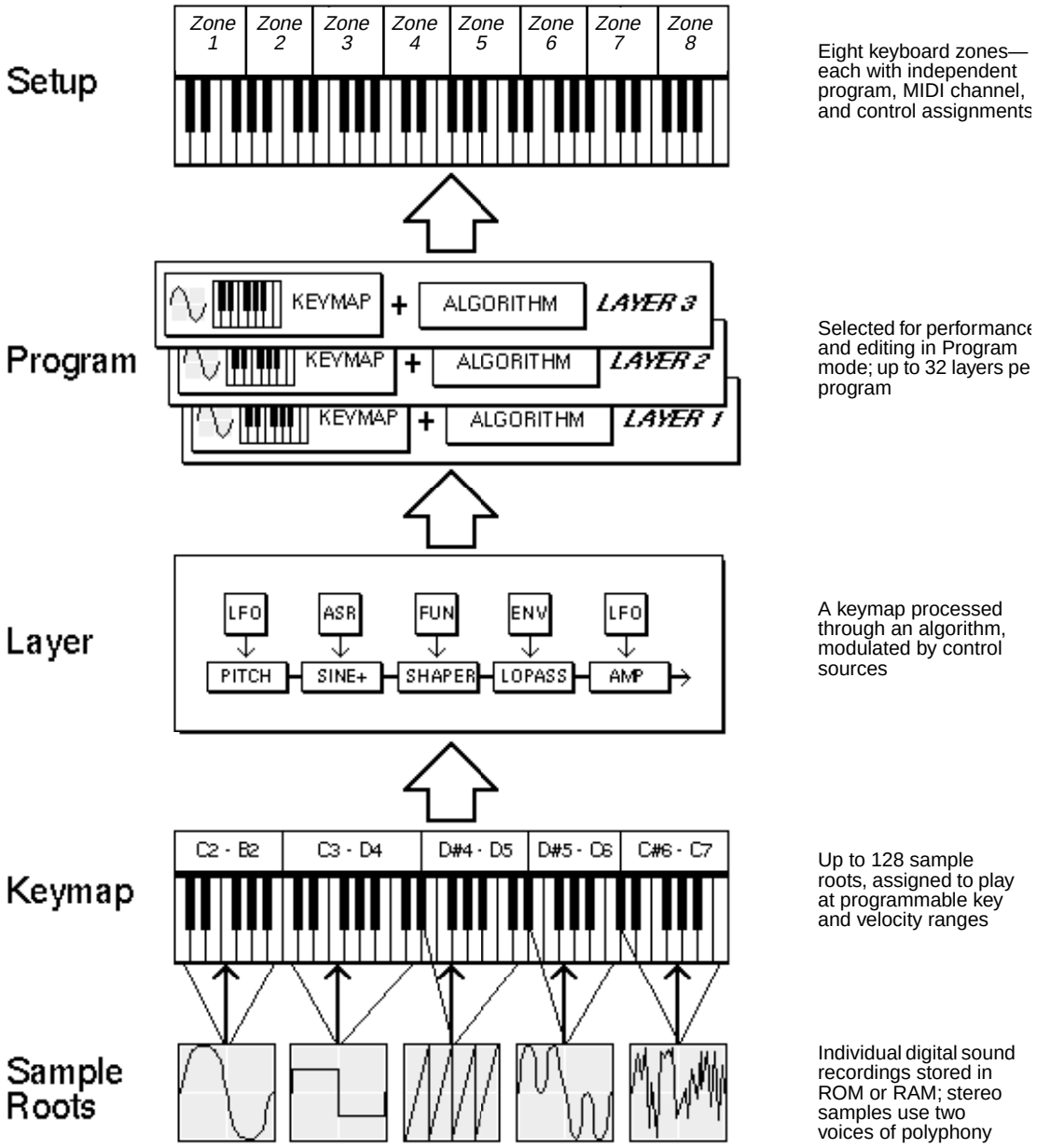


Figure 6-1 VAST Program Structure

KB3 Program Structure

There's nothing quite like the sound of the classic Hammond™ B-3 tone wheel organ, especially when played through a Leslie™ rotating speaker system. We've done extensive testing and analysis with several tone wheel organs, and created our own models to emulate the unique tone wheel sound. We even took into account the way that older organs start to sound different (and arguably better) as their capacitors begin to leak—and we included a parameter that lets you vary the amount of grunge (leakage) in your sound.

We also recruited some very talented organ players to try out KB3 programs, and we've used their feedback to make the real-time controls as convenient and realistic as possible.

KB3 programs use oscillators to emulate the tone wheel sound. Each oscillator operates independently, and has its own pitch and amplitude control. You can control how many oscillators are used for a KB3 program. There are two oscillators per voice, for a total of 96. You can use up to 95 of them in a KB3 program (the 96th is reserved to produce key click). Because the oscillators start running as soon as you select a KB3 program, there are always voices available—unlike VAST programs, which start "stealing" notes when you reach the 48-voice polyphony limit. In other words, with a KB3 program, you can play and sustain more than 48 notes, and the K2600 will continue to play then all. With VAST programs, once 48 notes are on (for example, when you play and sustain a four-note chord in a 12-layer program), each new note that you play replaces one of the notes that was already on.

The oscillators—we'll call them tone wheels from here on—are divided into an upper and lower group. By default, the upper tone wheels use the samples in the K2600's keymaps (including your own RAM keymaps if you want) to generate sound, while the lower tone wheels use waveforms (like sine, square, or sawtooth). You can switch this around if you like, for even more variety.

KB3 Mode

KB3 programs are different enough from VAST programs that we use the term KB3 mode to describe what's going on when you play a KB3 program. There are a few important points to consider if you want to get the most out of KB3 mode.

KB3 Channel

As we mentioned in Chapter 2, you can play KB3 programs only on the KB3 channel, which you define on the Master-mode page. When you're in Program mode, this means that the current MIDI channel must match the KB3 channel, and when you're in Setup mode, any zone that uses a KB3 program must use the KB3 channel. If this isn't the case, the KB3 programs won't make any sound. If this happens in Program mode, all KB3 program names appear in parentheses, and the info box reminds you that you're not on the KB3 channel. If it happens in Setup mode, the display looks normal, but if you go into the Setup Editor, the LocalPrg parameter shows the KB3 program name in parentheses in every zone that's not on the KB3 channel.

Note: *If you're using a Kurzweil PC88 to control your K2600, you shouldn't use Channel 1 as the KB3 channel. The PC88 sends MIDI Controller 90 on Channel 1 to select effects. In KB3 mode, the K2600 maps Controller 90 to internal controller 90, which controls the emulation of leakage that we mentioned earlier. You probably don't want your leakage level fluctuating every time the PC88 sends Controller 90.*

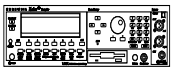
Real-time Controls in KB3 Mode

Owners of keyboard models of the K2600 have real-time control over many components of KB3 programs, directly from the front panel. The sliders emulate the drawbars that are so essential to the tone wheel sound, while the buttons above them (they're called the **Mute** buttons, because they normally mute and solo zones in Setup mode) can control the KB3 effects: Leslie, vibrato, chorus, and percussion (key click).

When you're in Program mode, the **Mute** buttons always control KB3 effects. In a setup containing a KB3 program, if you want the **Mute** buttons to control KB3 effects, you'll have to edit the setup, because in Setup mode, the **Mute** buttons mute and unmute zones by default.

1. Go to Setup mode, and select the setup you want to edit. Press **Edit**.
2. Press either **more** soft button until you see the **COMMON** soft button. Press it, and your display should look like this:

3. Select the Mutes parameter and change its value to **KB3 Control**.
4. Don't forget to save.



A note to K2600R owners: while you don't have a set of sliders or **Mute** buttons on your instrument, you can control most of the KB3 features from your MIDI controller. There are also dedicated controllers, like the Voce™ MIDI Drawbar controller, that give you convenient real-time control of KB3 features. See *MIDI Control of KB3 Programs* on page 6-6 for information about controlling KB3 programs via MIDI.

Playing KB3 Programs

One of the standard performance features of many tone wheel organs is the set of drawbars for emulating the stops on a pipe organ. Moving the drawbars controls the amplitude of either the fundamentals or the harmonics of the notes (out to increase amplitude, in to decrease it).

On keyboard models, the sliders and Mod Wheel serve as the nine drawbars found on most tone wheel organs. Pushing the sliders up is the equivalent of pushing the drawbars in (removing fundamentals or harmonics). The Mod Wheel is the other way around, since you're probably used to the Mod Wheel being off when it's down, and on when it's up. So remember, for the Mod Wheel, down (off) is like pushing the drawbar in (decreasing amplitude), and up (on) is out (increasing amplitude).

Subharmonics		Fundamental	Harmonics					
16'	5 1/3'	8'	4'	2 2/3'	2'	1 3/5'	1 1/3'	1'
Slider A	Slider B	Slider C	Slider D	Slider E	Slider F	Slider G	Slider H	Mod Wheel

Table 6-1 Standard Drawbar Settings for the Hammond B3

KB3 Mode Buttons (Mute Buttons)

When the **Mute** buttons are enabled for KB3 control, their LEDs indicate the status of the various effects for the current KB3 program. This status is saved as part of each program. You can change the effects in real time by pressing the buttons (or by sending the appropriate MIDI Controller values from your MIDI controller).

In normal operational modes, using the **Mute** buttons to change a program's KB3 effects doesn't affect the program; the effects return to their programmed settings the next time you select the program. If, however, you're in an editor when you change the effects, you're actually editing the program. If you like the changes, you can save the program with the new KB3 effects settings. If you don't like the changes, you can exit without saving, and the program will revert to its previous settings.

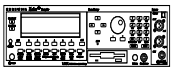
The **Mute** buttons also send MIDI Controller information to the K2600's MIDI Out port. See Column 2 of Table 6-3 to check which Controller numbers the buttons send.

Of course, you can change the programmed settings for the KB3-mode buttons. For each of the buttons, there's a corresponding parameter in the Program Editor.

	Effect Category	Button Name	Corresponding Page and Parameter	Comments
1	Rotary	Fast / Slow	MISC: SpeedCtl	
2	Vibrato	On / Off	MISC: VibChorCtl	
3		Chorus / Vibrato	MISC: VibChorSel	Disabled if Button 2 is off
4		Depth 1 / 2 / 3	MISC: VibChorSel	Disabled if Button 2 is off
5	Percussion	On / Off	PERC: Percussion	
6		Volume Loud / Soft	PERC: Volume	Disabled if Button 5 is off
7		Decay Fast / Slow	PERC: Decay	Disabled if Button 5 is off
8		Pitch High / Low	PERC: Harmonic	Disabled if Button 5 is off

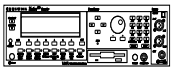
Table 6-2 KB3 Mode Buttons and Corresponding Parameters

MIDI Control of KB3 Programs



When you're playing a KB3 program from an external MIDI source, there are two things to keep in mind:

- Certain MIDI Controller numbers always control specific KB3 features
- The value of the LocalKbdCh parameter affects how KB3 programs respond to MIDI Controller messages



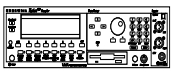
Controller Numbers

Table 6-3 lists the MIDI Controller numbers that control KB3 features. The first column lists the Controller numbers that KB3 programs always respond to (the K2600 also sends these Controller numbers to its MIDI Out port when you're using the local keyboard channel—we'll say more about that on page 6-7). KB3 programs also respond to the Controller numbers in the second column; these are the Controller numbers that the Voce™ MIDI Drawbar Controller uses to control common tone wheel organ features. Whatever kind of external MIDI source you're using, you can use the MIDI controller numbers in either the second or third column to control

the corresponding KB3 feature in the first column. For example, to control Drawbar 1, you can send either MIDI 6 or MIDI 12.

KB3 Program Feature	MIDI Controller Number	
	K2600	Voce
Drawbar1	6	12
Drawbar2	22	13
Drawbar3	23	14
Drawbar4	24	15
Drawbar5	25	16
Drawbar6	26	17
Drawbar7	27	18
Drawbar8	28	19
Drawbar9	1	20
Expression Pedal	4	8
Percussion On/Off	73	N.A.
Percussion High/Low	72	72
Percussion Loud/Soft	71	71
Percussion Fast/Slow	70	70
Rotating Speaker Slow/Fast	68	68
Vibrato/Chorus On/Off	95	95
Vibrato/Chorus Selector	93	93
Key Click Level	89	89
Leakage Level	90	90

Table 6-3 KB3 MIDI Controller Assignments



Local Keyboard Channel

The local keyboard channel enables the K2600 to receive MIDI information on a single channel, then rechannelize that information so you can play and control all eight zones of a setup, even if your MIDI source transmits on only one channel. When you're in Program mode, the local keyboard channel remaps incoming information to the K2600's current channel (the one shown in the top line of the display).

The LocalKbdCh parameter (on the RECEIVE page in MIDI mode) defines the local keyboard channel. When you're in Program mode, and playing a KB3 program, you may want to leave LocalKbdCh set to **None**, which is its default value. In this case the MIDI Controller messages for KB3 control listed in Table 6-3 are certain to work.

There are some possible disadvantages to this, however. First, the K2600 doesn't relay incoming MIDI to its MIDI Out port. Perhaps more importantly, if you change the channel on your MIDI source, the K2600 plays the program on the channel used by your MIDI source—regardless of the K2600's current channel. For example, if your MIDI source transmits on Channel 1, and you set the K2600's current channel to 2, you'll still play the program assigned to Channel 1. If that's the way you like it, there's no problem.

You may find it more convenient to use the local keyboard channel. In this case, the K2600 remaps incoming MIDI to the K2600's current channel, so in Program mode, you'll always play the program on the K2600's current channel. Incoming MIDI also gets sent to the K2600's MIDI

Out port. On the other hand, in this case your MIDI source's transmitting channel must match the K2600's local keyboard channel for anything to work. Furthermore, for KB3 programs, some of the MIDI Controller numbers listed in Table 6-3 won't necessarily work.

Things are a bit different for playing setups. In this case, you *must* use the local keyboard channel to be able to play and control all of the setup's zones. Set LocalKbdCh to match the channel your external MIDI source is using (so if, for example, your MIDI source transmits on Channel 1, set LocalKbdCh to **1**). All MIDI information that the K2600 receives on the local keyboard channel gets remapped to the channels and control destinations used by the zones in the setup.

The K2600 also remaps certain MIDI Controller messages that it receives on the local keyboard channel, so that they correspond (in most cases) to the default assignments for the K2600's physical controllers (Mod Wheel, sliders, ribbons, etc.). While this ensures that the physical controllers work in a consistent and relatively standard fashion for most setups and VAST programs, it necessitates a few adjustments to make incoming MIDI Controller messages control the KB3 features listed in Table 6-3. Without these adjustments, some of the KB3 features won't respond to MIDI Controller messages—this is true when you're playing programs as well as when you're playing setups.

To make everything work properly, you need to make sure that all the appropriate physical controllers are assigned for KB3 control. Physical controller assignments are handled by setups, and are defined by parameters on several pages in the Setup Editor (Table 10-1 on page 10-8 lists the physical controllers that get remapped by the local keyboard channel). Each zone of a setup has its own controller assignments. Programs don't have controller assignments, so they "borrow" them from a special setup that's reserved for that purpose. This setup is called the control setup; it's determined by the value of the CtlSetup parameter (on the TRANSMIT page in MIDI mode). You can read about control setups in detail on page 6-10.

When you're playing a setup on the local keyboard channel, each zone that uses a KB3 program must have the appropriate physical controller assignments. When you're playing a KB3 program, Zone 1 of the *control setup* must have the appropriate physical controller assignments.

There are two ways to configure a setup properly for KB3 control:

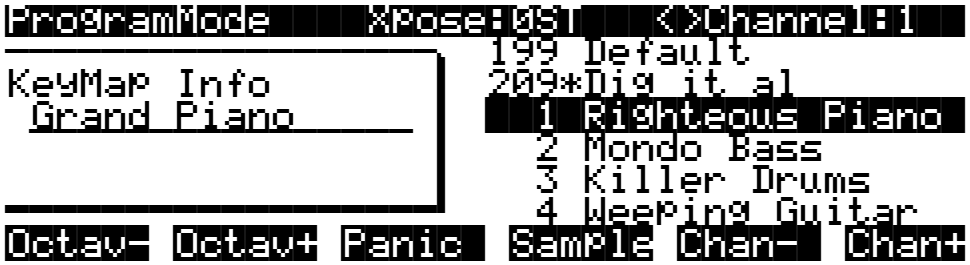
- Edit an existing setup, adjusting some of the physical controller assignments (see page 20-11).
- Use the KB3 setup that we've provided for your convenience. It's in a file on one of the accessory disks that came with your K2600 (see page 20-12).

KB3 Control: A Summary

Whenever you want to play a KB3 program, make sure that the KB3 setup is assigned as the control setup. When you want to play a setup containing a KB3 program, make sure that the zone that uses the KB3 program has the same physical controller assignments as the KB3 setup. When you're creating a setup that will use a KB3 program, use the KB3 setup as your starting point.

One final word—for now—about using the local keyboard channel: all the MIDI information received on the local keyboard channel also gets sent—*after being remapped*—to the K2600's MIDI Out port. There's a thorough discussion of the local keyboard channel beginning on page 10-7.

The Program Mode Page



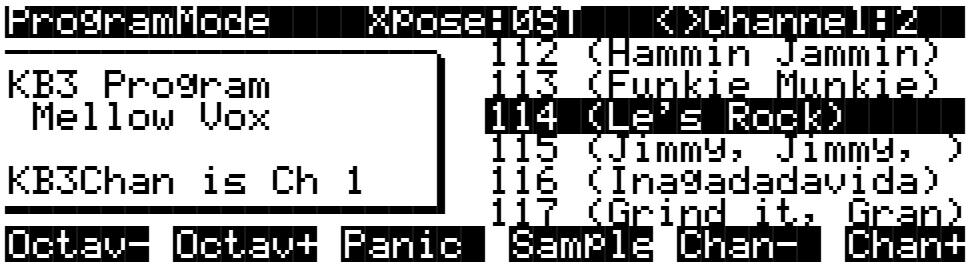
The top line of the Program-mode entry-level page shows your location, the present MIDI transposition, and the current MIDI channel.

The info box at the left of the Program-mode page gives you information about the current program. For VAST programs of up to three layers, the info box shows the keymap assigned to each layer (Layer 1 on top, with additional layers below). The line beneath the name of the keymap indicates the keyboard range of that layer. In the diagram above, for example, there's one layer that 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).

For drum programs (VAST programs of more than three layers), the info box shows the number of layers in the program. For KB3 programs, the info box shows the keymap used for the upper tone wheels (or the lower tone wheels, if you have the Upper/LowerSwap parameter set to **On**).

Program Names in Parentheses

While you are scrolling through different programs on various MIDI channels, you may occasionally encounter a program that doesn't make any sound, and whose name is in parentheses. The parentheses tell you that you have selected a KB3 program without being on the KB3 channel. KB3 programs use a different program architecture, and require many more voices to operate. Consequently, they a special channel with enough throughput to handle those voice requirements. If you select a KB3 program without being on a KB3 channel, the K2600 cannot play the program. As shown in the following illustration, the KB3 channel is 1, while the K2600's current channel is 2. The selected program is disabled.



The Program-mode page illustrates this in two ways: the program names are in parentheses, and the box at the left of the page includes the message "KB3 Chan is Ch 1." To fix this, you could either change the K2600's MIDI channel (with the **Chan/Bank** buttons), or make Channel 2 the KB3 channel (using the KB3Chan parameter in Master mode).

You can play any program on the KB3 channel, but you can play KB3 programs *only* on the KB3 channel.

If you've used a K2000 or K2500, you'll remember that you would occasionally see parentheses around the names of drum programs, as well. The K2600 can play drum programs—up to 32 layers—on any channel. The Drum Channel parameter no longer exists.

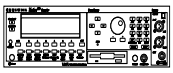
Control Setup

The control setup defines what the K2600's physical controllers (wheels, sliders, pedals, etc.) do while you're in Program mode. It's a convenient way to apply the controller assignments in your setups globally. Just choose an existing setup to be the control setup, using the CtlSetup parameter on the MIDI-mode TRANSMIT page. Then while you're in Program mode, many of the controller assignments for Zone 1 of the control setup also apply to the programs you play (this is true for MIDI control messages as well, unless you have turned off MIDI control).

If you don't like the way the physical controllers work in Program mode, you can either select a different control setup, or edit the existing one. Any changes you make to the current control setup will also affect the way that setup works in Setup mode.

There are a few important points to remember about the control setup:

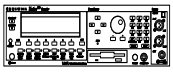
- The current control setup is used by *all* programs in Program mode.
- You cannot change the control setup from within Program mode.
- The control setup doesn't affect the *sound* of a program, only the assignments of certain physical controllers. The samples and keymaps assigned to a program are unaffected by the control setup. While you're in Program mode, the K2600 ignores the programs assigned to the setup that you choose as the control setup.
- Almost all of the VAST programs in the K2600 are designed to respond to the controller assignments in the default control setup (**97 ControlSetup**). Therefore you'll want to use **97 ControlSetup** as the control setup in most cases, with two exceptions. When you're using the local keyboard channel (that is, when the value of the LocalKbdCh parameter is anything but **None**) and playing a KB3 program from an external MIDI source, use a control setup that's configured for KB3 control, as described on page 6-8 (if LocalKbdCh is **None**, **97 ControlSetup** is fine for playing KB3 programs from an external MIDI source). If you want to change the controller assignments for any program or set of programs (either VAST or KB3), use a control setup that has the controller assignments you want.



See page 7-4 for a table listing the parameters that affect Program-mode controller assignments.

The Soft Buttons in Program Mode

The **Octav-/Octav+** buttons are a shortcut for quick transposition in 12-semitone increments. You can use them to transpose the entire K2600 as much as three octaves up or down. The top line of the display shows the current amount of transposition (Xpose). Pressing both **Octave** buttons simultaneously returns the transposition to zero.



If you're using a K2600R, the **Octav-/Octav+** buttons are functional only if the value of the Local Keyboard Channel parameter (LocalKbdCh) matches the current MIDI channel of your MIDI controller. In this case, the K2600R will apply transposition to the notes it receives on the local keyboard channel (but not on other channels). The LocalKbdCh parameter is on the RECEIVE page in MIDI mode.

The **Octave** buttons transpose the K2600, as well as any MIDI devices connected to the K2600's MIDI Out port. Changing the transposition with the soft buttons also changes the corresponding setting on the MIDI-mode TRANSMIT page.

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

Press the **Sample** soft button to enter the K2600's sampler. Refer to Chapter 14 for complete information on the sampler.

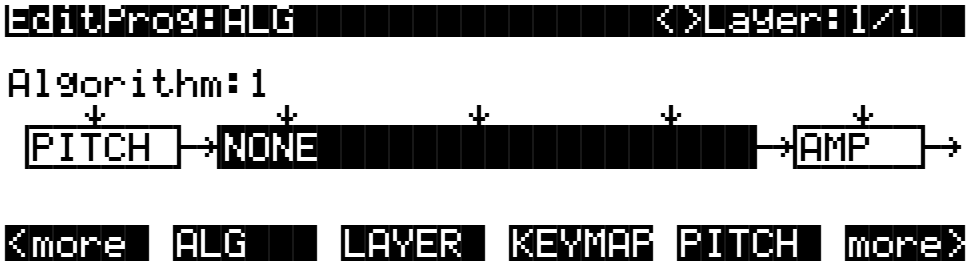
Use the **Chan–** and **Chan+** soft buttons to change the current MIDI channel. This changes the MIDI channel the K2600 uses internally, as well as the channel you're using to send information to other synths connected to the K2600's MIDI Out port (MIDI slaves). Changing the current MIDI channel with the soft buttons also changes the corresponding setting on the MIDI-mode TRANSMIT page.

Editing VAST Programs

The Program Editor is where you begin to modify the K2600'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.

This section describes the Program Editor as it applies to VAST programs. See *Editing KB3 Programs* on page 6-48 for information about editing KB3 programs.

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.



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—it's especially useful in multi-layered drum programs. Hold the **Enter** button and strike a key. The display will show the layer(s) 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 *Intuitive Data Entry* on page 3-7). 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 K2600'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 real-time 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 16-55.) 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.

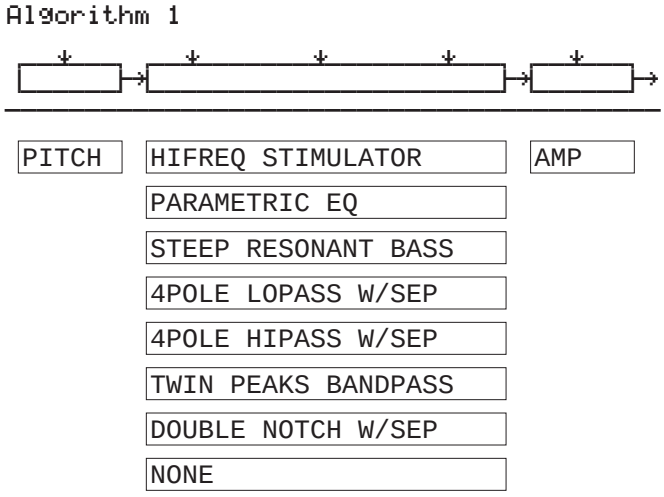


Figure 6-2 Algorithm Wiring

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.

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 (Coarse)
- Fine adjust (Fine)

Hard-wired Parameters

These always take their input from MIDI events (either the K2600 or an external MIDI controller)—specifically the note number and the attack velocity values of each Note On event.

- Key tracking (KeyTrk)
- Velocity tracking (VelTrk)

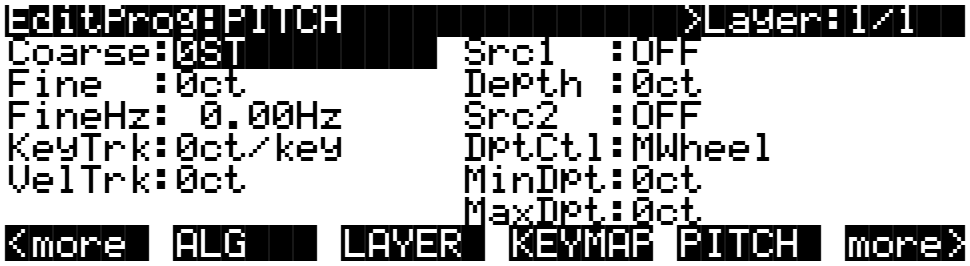
Programmable Parameters

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.



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 K2600 always uses real values of measurement, rather than just arbitrary numbers, for adjustable parameters. This means that you specify pitch in semitones and cents, amplitude in dB, and filter cutoff frequency in Hz.

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 with 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* on page 6-27 for a more thorough description of Fine Hz.

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 nonzero 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 K2600) 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 notes still increase 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 per 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 (Src1)

This parameter takes its value from a long list of control sources (you can find it beginning on page 4-7 in the *Musician's Reference*—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 K2600 or MIDI controller, reaching a maximum of 1200 ct (12 semitones, or one octave).

Source 2 (Src2)

This one's even more programmable. 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 the 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 *Musician's Reference*, where there's an explanation of how the K2600 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

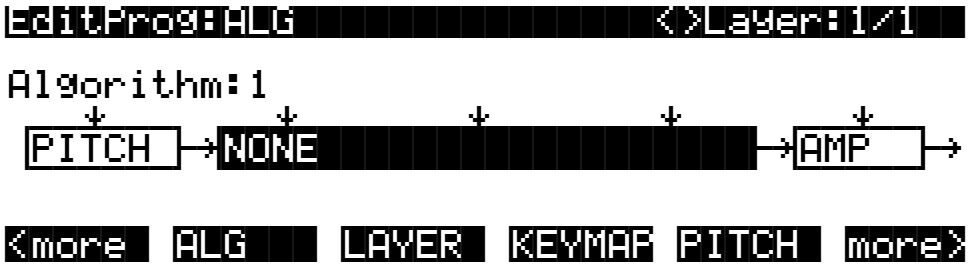
These six control source parameters are just a few of the control sources available throughout the Program Editor. We've given them special attention because they appear on *all* the pages relating to the DSP functions, not just on the PITCH page.

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.

The 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.



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 *Musician's Reference* 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 K2600 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.

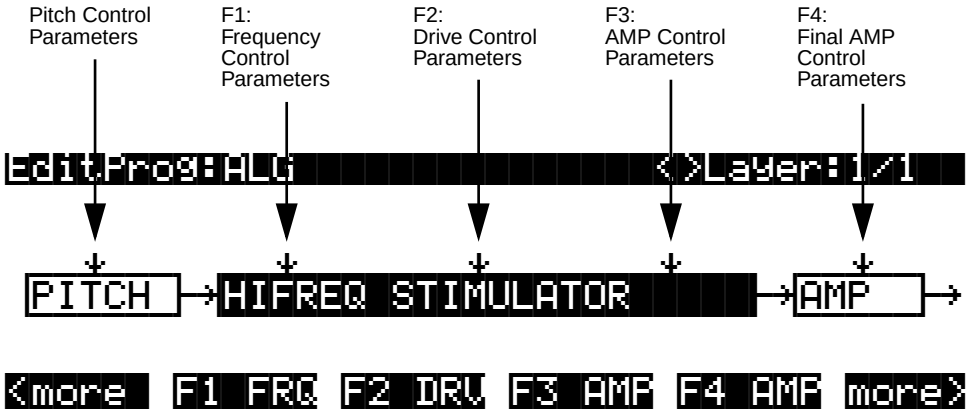
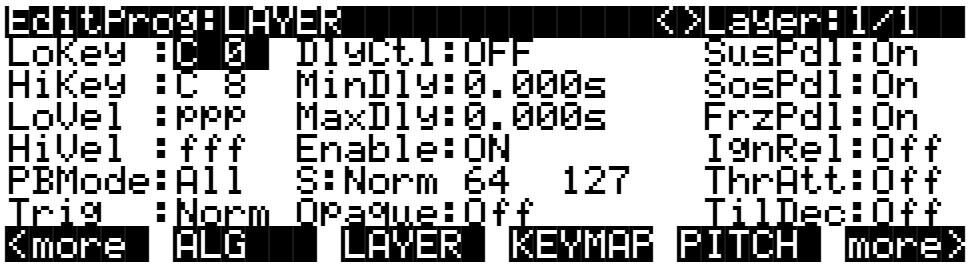


Figure 6-3 DSP Function Inputs

The pages (F1–F4) that control the DSP functions are described later in this chapter. *Algorithm Basics* on page 6-12 gives general information on algorithms, while Chapter 16 gives a thorough description of each the DSP functions and the parameters found on their editing pages.

The 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.



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 Enable Min Enable Max	Normal, Reversed -128 to 127 -128 to 127	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 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 threshold will not trigger notes in this layer.

Using LoVel and HiVel, you can set up velocity switching between up to eight layers. If you need even more, you can do it using the Enable and Enable Sense (S) parameters (page 6-21).

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 MinDly and MaxDly (described below). 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 assigns a control source to activate or deactivate the layer. When the value of the assigned control source is between the minimum and maximum thresholds set by the Sense (S) parameter, the layer is active. When the value of the assigned control source is below the minimum or above the maximum, the layer is inactive. By default, many layers have the Enable parameter set to **ON**, so the minimum and maximum thresholds don't matter. They're relevant only when Enable is set to a specific control source (like MWheel).

Some local control sources (KeyNum and AttVel, for example) are not valid for the Enable parameter. In these cases, you should use the global equivalent (GKeyNum and GAttVel in this example).

Enable Sense (S)

This parameter determines how and when a layer is enabled by the control source assigned for the Enable parameter. Enable Sense has three values: orientation, minimum, and maximum.

Suppose for a moment that you're editing a program, and in the current layer you've set the value of Enable to **MWheel**, which causes the Mod Wheel to control whether the layer is active. The default values for Enable Sense are as follows: orientation is **Norm**; minimum is **64**, and maximum is **127**. This means that when the Mod Wheel is less than halfway up, the layer is disabled. The layer plays only when the Mod Wheel is more than halfway up.

Change the orientation to **Rvrs**, and the layer plays only when the Mod Wheel is *less* than halfway up. Change the orientation back to **Norm**, and change the minimum to **127**. Now the layer plays only when the Mod Wheel is *all* the way up.

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 layer would have its Enable Sense orientation set to **Norm**, and the other would have it set to **Rvrs**. Both layers would have their Enable Sense minimums set to 64, and their maximums to 127. The first layer would play when your Mod Wheel was above its midpoint, and the second layer would play when the Mod Wheel was below its midpoint. (You could achieve the same effect by having the Enable Sense orientation in both layers set to **Norm**, and the min and max values set as follows: min **0** and max **63** for one layer; min **64** and max **127** for the other.)

Using this parameter in conjunction with the Enable parameter, you can easily create velocity-switching for as many layers as you have in your program. This is useful for drum programs, since you can define a different velocity-trigger level for each of the 32 layers available in drum programs.

First, set the Enable parameter for the Layer 1 to a value of **GAttVel** (global attack velocity). This causes the layer to play based on the attack velocity of your keystrikes. Then set the Enable Sense (S) parameter to a value of **Norm**, and adjust its minimum and maximum values (the two numerals to the right of **Norm**) to a narrow range. Don't use negative values, since they don't apply when you're using GAttVel as the layer enabler.

Repeat this for each layer in the program. Bear in mind that if you want to set up 32 different velocity levels for a program, with equal intervals between each layer, then you have a range of 4 for each level (Layer 1 is 0–3, Layer 2 is 4–7, and so on). It won't be easy to play precisely enough to trigger the layer you want. On the other hand, if you're using Song mode or an external sequencer, you can edit attack velocity levels, and get exactly the results you want.

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.

Start with a one-layer program, and create a new layer (Layer 2) with the **NewLyr** soft button. On the KEYMAP page for Layer 2, select the keymap you want to use, then on the LAYER page, set Layer 2's range (say, C 3 to D 3), and set its Opaque parameter to **On**. Then go to Layer 1, and duplicate it (with the **DupLyr** soft button); the duplicate layer becomes Layer 3. You now have a three-layer program. Delete Layer 1 (the original layer); Layer 2 (the new layer you created) becomes Layer 1, and Layer 3 becomes Layer 2. Now Layer 2 blocks out Layer 3 (the duplicate of 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). 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). When off, the layer ignores sostenuto messages. Sostenuto, as you may know, is a feature found on pianos that have three pedals. Pressing the Sostenuto pedal on a piano (usually the middle pedal) sustains the notes whose keys you were holding down when you pressed the pedal. Notes played after the pedal is already down do not get sustained.

Freeze Pedal (FrzPdl)

This parameter activates or deactivates 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 K2600. 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 K2600 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.

The 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.

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	0
Sample Skipping	Auto, Off, On	Auto
Stereo	Off, On	Off
Timbre Shift	± 60 Semitones	0
Playback Mode	Norm, Rvrs, Bidirectional, Noise, KDS 1—8	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 *Musician's Reference*.

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** (a cent is a hundredth 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. Unless you're looking for nonstandard note intervals, the values of the KeyTrk parameters on the PITCH and KEYMAP pages should add up to 100 cents.

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 (many drum programs, for example) 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 allows for increased upward transposition of samples. This is done by using a special sample playback algorithm, which enables the K2600 to increase the maximum playback rate of a sample from 96 Khz to a maximum of 192 Khz (thereby enabling the sample to be played at a higher pitch).

There is a tradeoff, however. Unwanted artifacts may creep into 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 employ sample skipping only for those notes whose upward transposition can be increased. Notes below a certain cutoff 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.

A value of **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.

A value of **Off** means that sample skipping will not be used at all.

Stereo

You'll use this parameter when you're working with stereo samples. When you use the stereo piano programs or load stereo samples from disk, the K2600 views both sides of the sample as a single sample object. When you select a stereo sample (by setting the value of the Sample parameter in the Keymap Editor), 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:

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 set to **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 same keymap as the value for *both* the Keymap1 and Keymap2 parameters. (With some imported sample formats, such as Akai, you'll have two keymaps for a stereo sample—for example, Piano L and Piano R. In this case, select the left keymap as the value for Keymap1, and the right keymap as the value for Keymap2.) If you select a keymap as the value for both Keymap1 and Keymap2, the K2600 automatically uses the left side for Keymap1, and the right side for Keymap2.

Once you have the keymaps assigned, 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.

If you're not using stereo samples, you should set this parameter's value to **Off**.

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 four 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 numerous 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. **Noise** replaces the samples with a white noise generator.

KDS In 1 through **KDS In 8** are for use if you have the digital I/O option, in which case you can use an external digital signal to the layer's sample. This signal then gets processed by the layers' DSP algorithm and sent to the layer's output. Using this value disables most of the other KEYMAP-page parameters for the layer.

Using the KDS inputs bypasses the PITCH page in the current program, so nothing on the PITCH page has an effect. If the current program uses a natural amplitude envelope, the signal at the KDS In uses the current keymap's amplitude envelope and volume adjust parameters.

Alternative Switch (AltControl and AltMethod)

You can assign a control source to change the sound by using an alternative start point or alternative end point for the current keymap. Whether it is an alternative start or alternative 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 alternative 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.

Use the AltControl parameter to specify a control source that will cause the sample to begin or end at the Alt point. Then use the AltMethod parameter to choose between switched and continuous calculation of the Alt point. If the value of AltMethod is **Switched**, the K2600 will use the Alt point when the relevant control source is at a value greater than 64 at Note Start. If AltMethod is **Continuous**, the Alt point will vary depending on the value of the relevant control source at Note Start.

As an example, suppose you're editing a two-second sample. You've set the Start point at **0.000**, and the Alt point at **1.000** (this is done on the TRIM page in the Sample Editor). Now you return to the KEYMAP page in the Program Editor, and you set AltControl to **MWheel**. If you set AltMethod to **Switched**, the sample will begin at the 1-second point if the Mod Wheel is at least halfway up at Note Start. If you set AltMethod to **Continuous**, the K2600 will interpolate the sample's starting point based on the position of the Mod Wheel. If the Mod Wheel is halfway up (64) at Note Start, the sample will begin at the half-second point. If the Mod Wheel is 75% up (96), the sample will begin at the .75-second point, and so on.

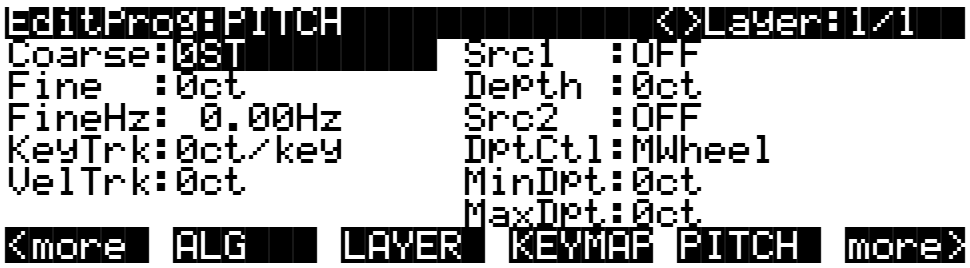
Emulating Legato Play

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 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 K2600'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.

For more information, refer to the discussion of the TRIM page's Alt point on page 14-16.

The 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.



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	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

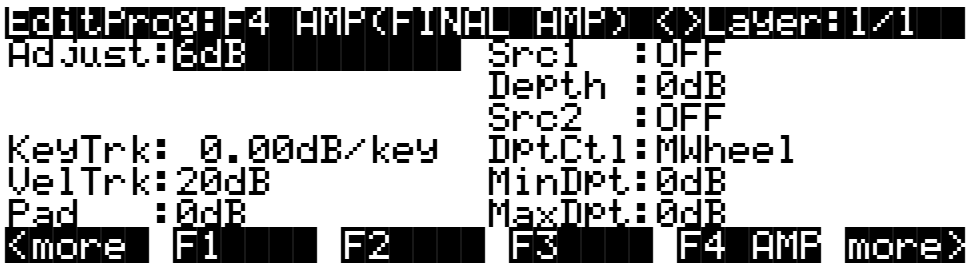
This measures pitch adjustment by the relative frequency (in Hertz) of each note. This is useful for controlling the beat frequency between layers in a multi-layered program. Using this parameter to detune chorusing layers will keep the beat frequency constant across much of the keyboard. Although the ratio of frequencies between each layer remains constant, the detuning will increase at lower pitches, and can become extreme. The K2600 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.

The 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 *Musician's Reference* for a complete list of the algorithms and their available DSP functions.

The 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.



Parameter	Range of Values	Default
Adjust	–96 to 48 Db	6
Key Tracking	± 2 Db per key	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 (preamp 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.

Source 1 and Source 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.

The OUTPUT Page

Press the **OUTPUT** soft button to get to the OUTPUT page, where you set the layer's output routing. This doesn't assign the layer's audio signal to the *audio* outputs, as you might expect. It assigns the signal to a KDFX *input*, which determines what effects, if any, get applied to the signal before the actual audio output. The routing of the signal from KDFX to the audio outputs happens on the OUTPUT page in the Studio Editor. See page 9-15, as well as Chapter 19, for more information about audio output.

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.

Parameter	Range of Values	Default
Pair	KDFX-A, KDFX-B, KDFX-C, KDFX-D	KDFX-A
Pan	Left to Right (15 Positions, indicated by *)	Center
Mode	Fixed, +MIDI, Auto, Reverse	+MIDI
Gain	–12 to 30 Db (6 Db Increments)	18 Db
Crossfade Control	Control Source List	OFF
Crossfade Sense	Normal, Reversed	Norm

Pair

This parameter sets the output routing of the current layer—not the final audio output, but the KDFX (effects processor) input to which the layer's signal will flow. Setting a value here is like connecting an audio output to the effects send on a mixer. The actual audio output of the K2600 is determined by the output settings for KDFX.

In a nutshell, the audio signal gets routed from the K2600's sound engine into KDFX using this parameter. A separate KDFX parameter routes the signal from KDFX to the physical audio outputs on the K2600's rear panel.

Pan

Use this parameter to position the current layer's audio signal from left to right. An additional pan parameter (Pan2) appears if you have the Stereo parameter on the KEYMAP page set to a value of **On**.

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**. In this case, leave Mode set to **Fixed**, and set Pan1 fully left, and Pan2 fully right.*

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. This is a good way to make a layer louder without the voices on that layer clipping. If you do this, however, keep in mind that increasing the gain reduces your headroom, so if you're playing a lot of voices, you may get some channel clipping (as opposed to voice clipping).

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.

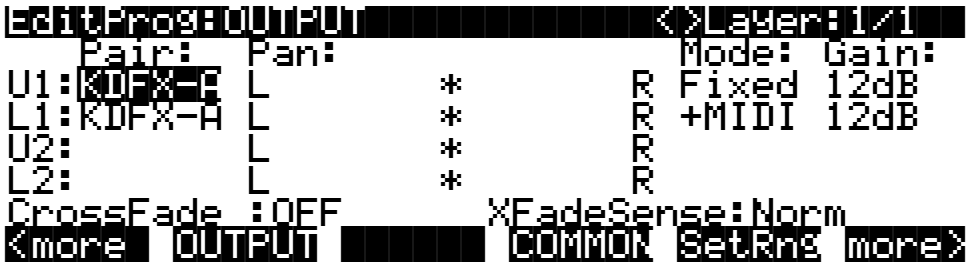
```
EditProg:OUTPUT              <>Layer:1/1

  Pair:  Pan:                   Mode:  Gain:
U:KDFX-A L        *          R Fixed  12dB
L:KDFX-A L        *          R +MIDI  12dB

CrossFade :OFF       XFadeSense:Norm
<more  OUTPUT       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:KDFX-A
Pan1:L        *          R
Pan2:L        *          R
Mode:+MIDI
Gain:12dB
CrossFade :OFF       XFadeSense:Norm
<more  OUTPUT       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.



The COMMON Page

Here's where you find eight 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.

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
OutPair	KDFX-A, KDFX-B, KDFX-C, KDFX-D	KDFX-A

Pitch Bend Range

Use this parameter to define how much the pitch will change when you move your 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.

Setups include a parameter called Bend Range (on the BEND page in the Setup Editor), which in most setups does not affect the program-level Pitch Bend Range setting. This is because in most setups the default value for Bend Range is **Prog**, which gives control of the pitch bend range to each program in the setup. If you change the value of Bend Range to anything but Prog, however, the setup will control the pitch bend range for all programs in the setup.

Setup 97 (the default control setup for defining the controller setting while you're in Program mode) has its Bend Range parameter set to **Prog**, so unless you change the value, or use a different control setup, individual programs control the pitch bend range while you're in Program mode. For more about the Bend Range parameter, see page 7-20.

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 Legato Play parameter and 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.

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 K2600 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 K2600 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 nonlinear; 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 K2600 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, FUNs 2 and 4, and program output routing to KDFX. 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.

OutPair

When Globals are off, the OUTPUT page for each layer determines the KDFX input to which the layer sends its audio output. This enables you to apply a different effect to each layer. Turning Globals on overrides the settings on each OUTPUT page, causing every layer in the program to be routed to the KDFX input you specify for the Outpair parameter. This is a quick way to apply the same effect to every layer in the program.

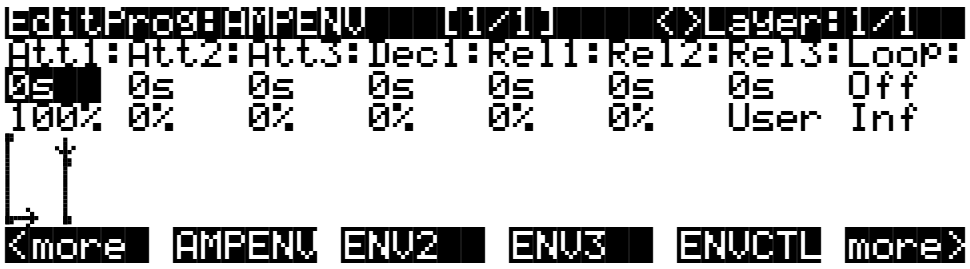
The Amplitude Envelope (AMPENV) Page

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.

If you want to build your own amplitude envelope, just turn the Alpha Wheel a click. The value **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.

Many programs feature User envelopes with appropriate envelope settings. This is usually the case for programs that use samples of acoustic instruments, since it provides a convenient starting point for you to adjust the envelopes.



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 K2600'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 Group	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 nonzero 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's level to a nonzero 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

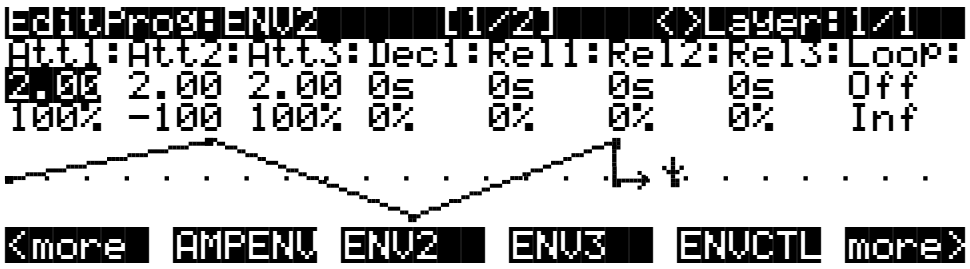
A value of **Inf** makes the amplitude envelope loop until a Note Off is generated. Values of **1** through **31** indicate how many times the loop will repeat after the amplitude envelope has played once through its normal cycle.

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* on page 6-19), or whatever.

The Envelope 2 (ENV2) and Envelope 3 (ENV3) Pages

The K2600 offers two envelopes in addition to AMPENV. Like AMPENV, ENV2 and ENV3 can be assigned like any other control source. Unlike AMPENV, however, ENV2 and ENV3 can be bipolar. This means that you can set negative values for them. (Obviously, you can't have an amplitude less than zero, so AMPENV 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.

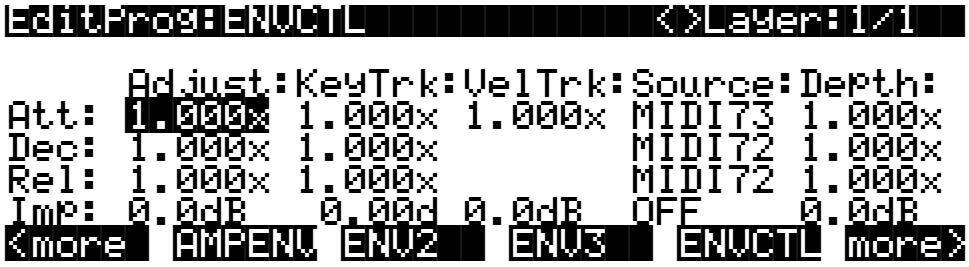
Another difference is that AMPENV *always* controls the amplitude of the layer, so even if you use it as a control source for other functions, it will still affect the layer's amplitude. ENV2 and ENV3 affect only those layers that have them assigned as a control source. Also, AMPENV uses an exponential attack (the amplitude rises much faster at the end of the attack segment than it does at the beginning), while ENV2 and ENV3 use linear attacks (the attack segment increases at the same rate from start to finish).



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.

The 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.



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 Group (Available for each of Att, Dec, Rel, Imp)	Range of Values
Adjust	0.018 to 50.000x (-24.0 to 24.0 dB for Imp)
Key Tracking	0.018 to 50.000x (-2.00 to 2.00 dB for Imp)
Velocity Tracking	0.018 to 50.000x (Not available for Dec or Rel; -24.0 to 24.0 dB for Imp)
Source	Control Source List
Depth	0.018 to 50.000x (-24.0 to 24.0 dB for Imp)

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.

This feature is incompatible with the K2000. K2000s running operating system version 3.54 or earlier won't even load programs that use Impact. K2000s running operating system version 3.8 or higher *can* load programs that use Impact. The Impact feature won't work, but the programs will play normally otherwise.

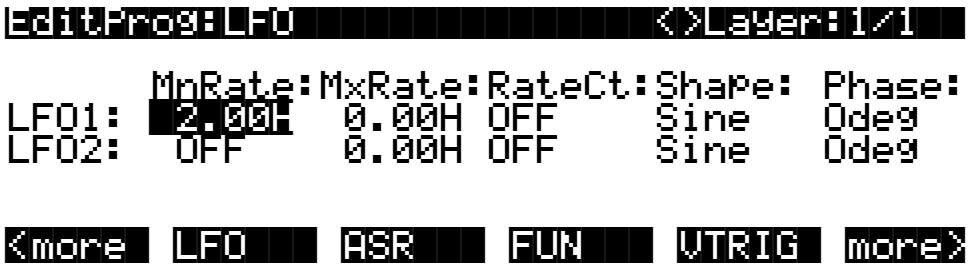
The LFO Page

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 K2600, 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.



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 Group (Available for each of LFO1 and LFO2)	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 runs. 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 *Musician's Reference*; 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.

The ASR Page

ASRs are three-section unipolar envelopes—attack, sustain, and release. The K2600'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 20 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.

Parameter Group (Available for each of ASR1 and ASR2)	Range of Values	Default Values (for both ASR1 and ASR2)
Trigger	Control Source List	Off
Mode	Normal, Hold, Repeat	Normal
Delay	0 to 60 seconds	0 seconds
Attack	0 to 60 seconds	0 seconds
Release	0 to 60 seconds	0 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**, a global ASR starts running immediately when you select a program that contains it. A *local* ASR starts running as soon as you trigger a note in the layer that contains it. 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. Nonzero values will cause a corresponding delay between the ASR trigger and the start of the ASR.

Attack

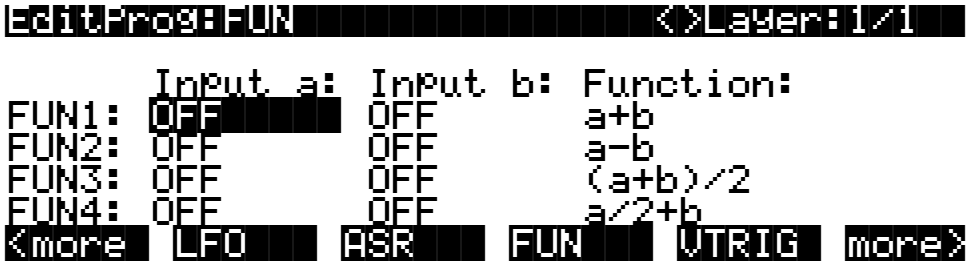
This defines how long the ASR takes to ramp up from minimum to maximum effect on whatever it's patched to.

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.

The Function (FUN) Page

FUN is short for function. The K2600'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:



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 K2600 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 17 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.

The Velocity Trigger (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.

Parameter Group (Available for each of VTrig1 and VTrig2)	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 K2600 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 K2600 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.

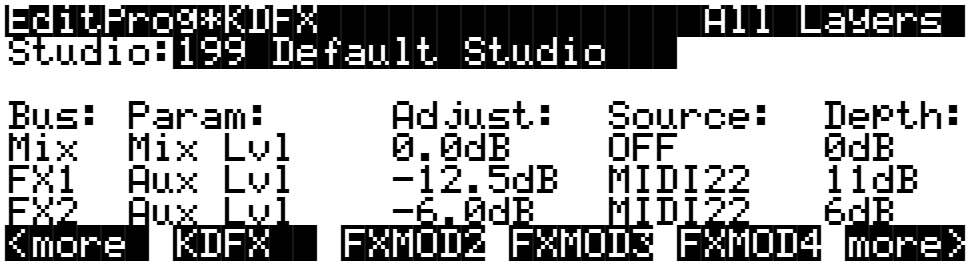
The KDFX Page

This is where you assign a studio to be used with the current program (depending on the settings for the FX Mode and FX Chan parameters—see *FX Mode* on page 9-4, and *Effects Channel (FX Chan)* on page 9-5).

The KDFX page is the first of four pages containing parameters that enable programs to control their associated studios in real time. For example, you can create (or edit) a program that uses the Mod Wheel or a slider to control the Wet/Dry mix of the signal that goes through FXBus1. The possibilities are almost limitless.

There are 18 sets of these real-time studio-control parameters, which we call FXMods. Each FXMod consists of five parameters. An important point to remember about FXMods is that they are components of a *program* or a *setup*, and they don't permanently affect the studios they control. They simply enable you to make real-time, temporary changes to the studio settings,

using the physical controller of your choice. FXMods provide a convenient way to gain serious performance flexibility.



The top line of the display indicates that you're on the KDFX page in the Program Editor, and that any changes you make will affect all layers of the current program. The second line consists of the Studio parameter; this identifies the studio associated with the current program, and enables you to change that studio.

The next line identifies the five parameters that make up each FXMod. Every FXMod uses these same five parameters, although the available values for some parameters depend on the specifications of the current studio, or on the values of other parameters in the FXMod—or both.

Bus Indicates which bus the FXMod will affect. Every FXMod is associated with a single KDFX bus. This can be any of the KDFX buses, which include Input A (**InA**) through Input D, FXBus1 (**FX1**) through FXBus4, the Aux bus (**Aux**), and the Mix bus (**Mix**).

Param Determines which aspect of the current studio the FXMod will affect. The available values for this parameter depend on both the value of the Bus parameter, and the specifications of the current studio.

Adjust Like the Adjust parameter on other editor pages, this enables you to set a "starting point" for your real-time control over the specified aspect of the studio. For example, you might want to zero a Wet/Dry mix before using a slider to vary it, or change a cutoff frequency to enhance the harmonics in a particular sound or effect. The available values for this parameter depend on the value of the Param parameter. The values you set for the Adjust parameter are not cumulative, as they are in other editors; they override the programmed settings for the studio used by the program.

Source Specifies which control source you'll use to modify the studio in real time. You can use any global control for the source.

Depth The amount that the specified aspect of the studio will be modified by the specified source. This is in addition to the amount specified by the Adjust parameter. The available values for Depth depend on the value of the Param parameter.

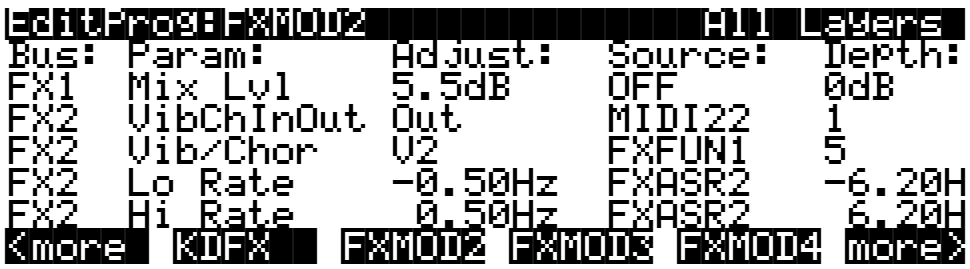
On the Program-editor page above, the first FXMod affects the level on the Mix bus, overriding whatever mix level is defined for the studio.

The second FXMod cuts the level of the send from FXBus1 to the Aux bus by 12.5 dB. Slider B (MIDI 22) boosts the Aux level by 11 dB at the top, and returns it to the Adjust value at the bottom.

The third FXMod does pretty much the same thing as the second, except that it does it to FXBus2, cuts the signal less, and boosts it less when you move Slider B up.

The FXMOD2–FXMOD4 Pages

These pages are a continuation of the KDFX page. Unlike the KDFX page, these pages each list five FXMods, and they don't indicate the current studio. Otherwise, they're identical to the KDFX page, listing the five parameters that make up each FXMod.

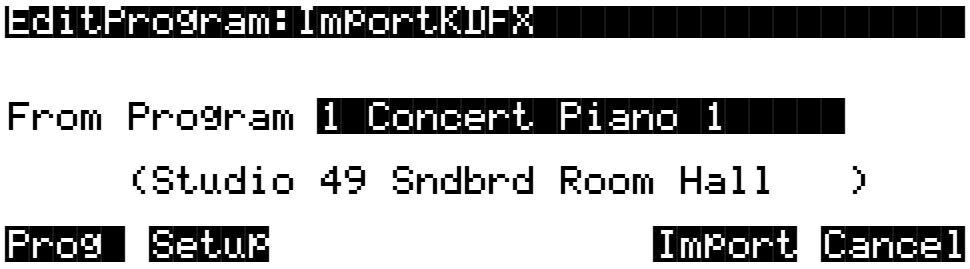


The FXLFO, FXASR, and FXFUN Pages

If you've read the descriptions of LFOs, ASRs, and FUNs that begin on page 6-40, you already know most of what you need to know about these three pages. They contain the parameters for the two LFOs, two ASRs, and four FUNs that the K2600 provides for KDFX control. With two exceptions, these control sources operate like the "regular" LFOs, ASRs, and FUNs. The differences are that the FX versions are global (they affect all layers in a program instead of individual layers), and they're available only for control of KDFX. Otherwise they're the same as the regular LFOs, ASRs, and FUNs (the regular versions can be used to control KDFX as well). In fact the descriptions for the regular LFOs, ASRs, and FUNs apply to their KDFX counterparts. Please see pages 6-40 through 6-44 if you need more information.

The ImportKDFX Page

You can import studios and FXMods from existing programs or setups. Press the **ImpFX** soft button to bring up the ImportKDFX page.



There's just one parameter, which tells you the program or setup from which to import. Use the **Prog** or **Setup** soft button to toggle between the list of programs and the list of setups. In this case, we're about to import the studio and FXMods from Program **1 Concert Piano 1** (which uses Studio **49 Sndbrd Room Hall**). As you scroll through the list of programs or setups, the display indicates which studio is assigned to the program or setup.

When you press the **ImpFX** soft button, the K26000 copies the studio and FXMod settings from the specified program or setup, and applies them to the current program. You then return to the page you were on when you pressed the **ImpFX** soft button.

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 K2600 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 the K2600 keyboard or a MIDI controller. When you trigger a note, the K2600 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. See page 5-3.

Save

Start the process of saving the current program. See page 5-3.

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 K2600 will say it's deleting the program, but it doesn't actually do it. See page 5-6.

Dump

Send a MIDI System Exclusive dump of the current program's settings. See the *Musician's Reference* 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 K2600 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, and is the highest-numbered layer in the program. If the current program already has its maximum number of layers, the K2600 will tell you that you can't add any more.

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 the highest-numbered layer in the program.

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 the **Layer–** or **Layer+** 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 **Prog–** or **Prog+** soft buttons (or the **Left/Right** cursor buttons) to change the program number.

While you are in this dialog, 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 K2600 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.

Editing KB3 Programs

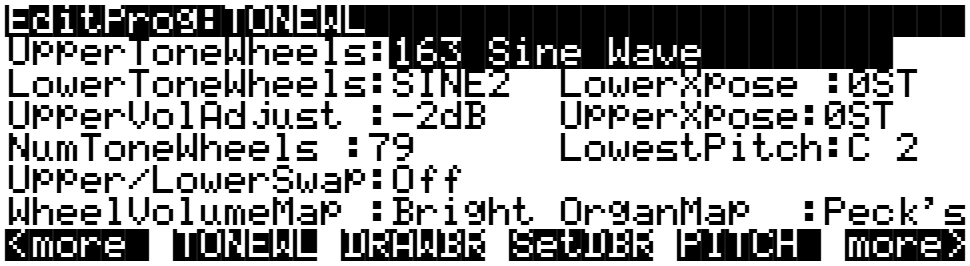
You can edit a wide assortment of any KB3 program's parameters. You can also create your own KB3 programs, though you must start with an existing KB3 program to do this. A regular K2500 program cannot be turned into a KB3 program. If you're not sure whether the current program is a KB3 program, check the information box on the left side of the program display; it will indicate "KB3 Program" if that's what the program is.

Enter the KB3 program editor by pressing the Edit button while a KB3 program is selected in program mode. You'll quickly see that the KB3 editor differs from the standard VAST program editor.

The TONEWL Page

By default, KB3 Mode uses DSP-generated waveforms for the lower half of its tone wheels and samples for the upper half of its tone wheels. Using the parameters on the TONEWL page, you can specify the waveforms and samples you wish to use, the number of tone wheels (which will affect how many other voices are available to you), and other related settings. You can even

switch the tone wheels, so that samples are used for the lower tone wheels and waveforms are used for the upper ones.



Parameter	Range of Values
Upper Tone Wheels	Sample List
Lower Tone Wheels	Sine, Sine2, Saw, Square
Upper Volume Adjust	-96 dB to 96 dB
Number of Tone Wheels	24–95
Upper/Lower Swap	Off, On
Wheel Volume Map	Equal, Bright, Mellow, Junky
Lower Transposition	-120 to 127 Semitones (-168 to 79 if Upper/Lower Swap is **On**)
Upper Transposition	-176 to 79 Semitones (-128 to 127 if Upper/Lower Swap is **On**)
Lowest Pitch	C 0 to C 7
Organ Map	Equal, Peck's, Bob's, Eric's

Upper Tone Wheels

Use this parameter to indicate the keymap (and thereby the samples) to use for the upper tone wheels. You can use any ROM or RAM keymap, though you must specify a keymap that uses looped samples for KB3 Mode to work correctly. When in Program mode, the keymap assigned to the program appears in the info box.

Lower Tone Wheels

Here you can specify the waveform to use for the lower half of the tone wheels. Choose from SINE, SINE2, SAW, and SQUARE. SINE2 is an improved version of SINE, with less distortion.

Upper Volume Adjust

Since sample volumes can vary, while the volume of DSP-generated waveforms will remain consistent, you may find it necessary to adjust the level of the sample-based tone wheels. This parameter lets you adjust the amplitude of the upper (sample-based) tone wheels relative to amplitude of the waveform-generated tone wheels.

Number of Tone Wheels

This parameter lets you specify the number of tone wheels used by a KB3 program. The classic tone wheel organs used 91 tone wheels, though the lowest 12 were for the pedals only. Therefore, you may find 79 a good number of tone wheels to specify for realistic organ

emulations. This would leave you eight voices for other programs. You can specify up to 95 tone wheels.

Here's how to do the math to calculate polyphony: the number of K2500 voices used by a KB3 program is (number of tone wheels+1)/2, rounded to the next highest whole number if the result is a fraction. So, for example, with 79 tone wheels specified you would use 40 voices. Keep in mind that these voices are permanently allocated and running while the KB3 program is selected, and cannot be stolen. The additional voice used by KB3 programs, by the way, is for keyclick.

Upper/Lower Swap

Use this parameter to swap the upper and lower tone wheel groups (that is, to use sample-based tone wheels for the lower positions and waveform-generated tone wheels in the upper positions). Setting this parameter to **On** changes the available values for upper and lower transposition.

Wheel Volume Map

The wheel volume map determines the volume level for each tone wheel. We've provided several tone wheel volume maps here, based on measurements we've taken on different organs. **Equal** is a map with all tone wheels at the same volume. It's not based on a real B3. **Bright** is a good normal map, based on a B3 in good condition. **Junky** is based on a B3 with an uneven, rolled-off response. **Mellow** is somewhere between **Bright** and **Junky**.

You can also apply EQ to control wheel volumes base on the frequencies of each tone wheel. See page 6-59.

Organ Map

The organ map controls the relative amplitude of each key, per drawbar. Like the wheel volume maps, these maps are based on measurements we've made on actual organs. **Equal** uses the same volume for each key and drawbar, and is not base on a real B3. **Peck's** is a good normal map, from a B3 in good condition. **Eric's** is a bit more idealized; it's smoothed out, but less realistic. **Bob's** is more uneven, based on an old B3.

Lower Transpose / Upper Transpose

These two parameters let you transpose the upper and/or lower tone wheels in semitone steps away from their default tunings. The available ranges of values for these parameters depends on the setting of the Upper/LowerSwap parameter.

Lowest Pitch

Here you can specify the pitch of the lowest tone wheel, which is typically set to C2. The rest of the tone wheels—as many as you have specified in the NumToneWheels parameter—will be tuned in semitone steps above this pitch.

The DRAWBR Page

Press the **Drawbr** soft button to view the DRAWBR Page. This page lets you edit KB3's drawbars. Remember that your K2600's sliders function as Drawbars 1-8, while the Mod Wheel is Drawbar 9.

Mode

When you set Mode to **Preset**, the preset drawbar settings on this page will be installed at program selection. The drawbar values will immediately change, however, as soon as you move the corresponding drawbar. Set Mode to **Live** if you want the drawbar volume settings at program selection to be determined by the positions of the drawbar controllers (sliders and Mod Wheel). With either setting, the drawbar controllers will affect drawbar volumes subsequent to program selection.

Steps

This parameter lets you specify the increments by which drawbar volumes will change. Choose either **0–8**, to approximate the drawbar settings on actual organs, or choose **0–127** for a finer degree of resolution.

Volume

This parameter only appears only if you've set Mode (see above) to **Preset**. Use the Volume parameter to set the preset volume of each of the nine drawbars. The available values will be **0–8** or **0–127**, depending on the setting of the Steps parameter.

Tune

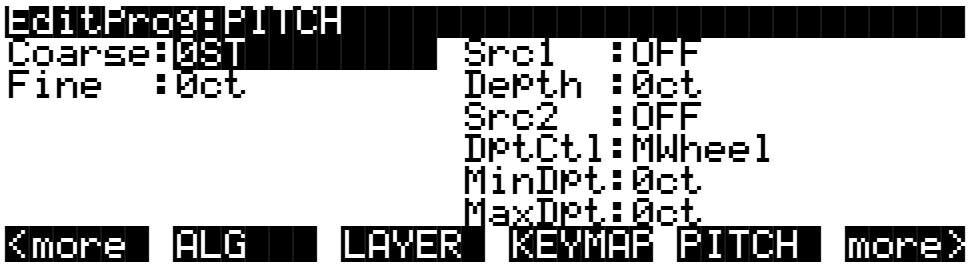
This parameter lets you tune each of the nine drawbars up or down in semitone steps. The values for the Tune parameter on the DRAWBR page shown above represent standard drawbar settings on a real B 3, as shown in Table 6-1 on page 6-5.

The SetDBR Soft Button

Press the **SetDBR** soft button to capture the current position of the drawbars, and use those positions as the preset drawbar positions on the DRAWBR page.

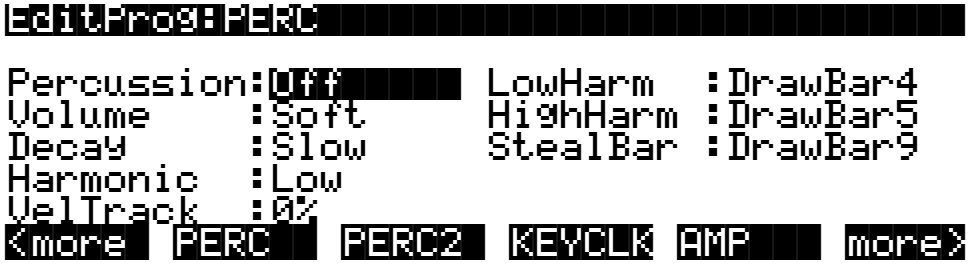
The PITCH Page

The PITCH page for KB3 programs is much like the PITCH page for VAST programs. The only difference is that for KB3 programs, there are no FineHz, KeyTrk, or VelTrk parameters. For a full description of the PITCH-page parameters, see page 6-27.



The PERC Page

Percussion is a characteristic feature of tone wheel organs. It's especially useful while soloing, since percussion adds an extra plink (actually an extra tone at a defined harmonic) to the attack. You can reach the percussion parameters by pressing the **Perc** and **Perc2** soft buttons.



Parameter	Range of Values
Percussion	Off, On
Volume	Soft, Loud
Decay	Slow, Fast
Harmonic	Low, High
Velocity Tracking	0–100%
Low Harmonic	Drawbar 1–9
High Harmonic	Drawbar 1–9
Steal Bar	None, Drawbar 1–9

Percussion

This is where you turn the percussion effect on or off. Percussion is created by a decaying envelope applied to one of the nine drawbars. The percussion effect is "single-triggered," which means that once it's triggered, it won't trigger again until all keys (or whatever you're using to trigger notes) go up. So if no keys are down, and you play a chord, percussion gets applied to all notes in the chord (and in fact, to all notes that are triggered during the short duration of the percussion envelope). Once the envelope runs its course, any notes you play while at least one

key is held down get no percussion. On keyboard models, you can turn percussion on or off by pressing Assignable Controller Button **5** (**Mute** button **5**).

Volume

This parameter switches between loud and soft percussion settings. The actual amplitude is set on the PERC2 page. On keyboard models, you can toggle between loud and soft by pressing Assignable Controller Button **6** (**Mute** button **6**).

Decay

This parameter switches between fast and slow percussion settings. The actual decay rate is set on the PERC2 page. On keyboard models, you can toggle between slow and fast decay by pressing Assignable Controller Button **7** (**Mute** button **7**).

Harmonic

This parameter switches between high and low harmonic percussion settings. The actual pitch is controlled by the LowHarm and HighHarm parameters. On keyboard models, you can toggle between low and high harmonics by pressing Assignable Controller Button **8** (**Mute** button **8**).

VelTrack

Here is where you specify the degree to which key velocity controls percussion volume. A value of zero corresponds to no velocity tracking, which is like a real tone wheel organ. Other values add velocity tracking, so that increased velocity results in louder percussion.

LowHarm

Controls which drawbar is used as the basis for the percussion when Harmonic is set to **Low**. On an actual tone wheel organ, this is Drawbar 4 (2nd harmonic). The actual pitch obtained depends on the drawbar tuning.

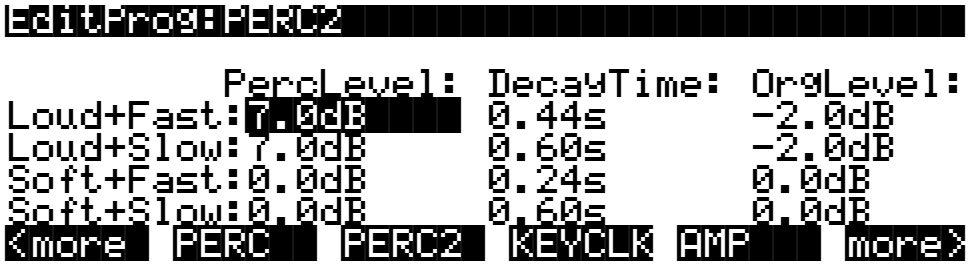
HighHarm

Controls which drawbar is used as the basis for the percussion when Harmonic is set to **High**. On an actual tone wheel organ, this is Drawbar 5 (3rd harmonic). The actual pitch obtained depends on the drawbar tuning.

StealBar

Controls which drawbar is disabled (if any) when the percussion effect is turned on. On an unmodified tone wheel organ, the ninth drawbar is the one disabled. Any drawbar can be selected, including **None**.

The PERC2 Page



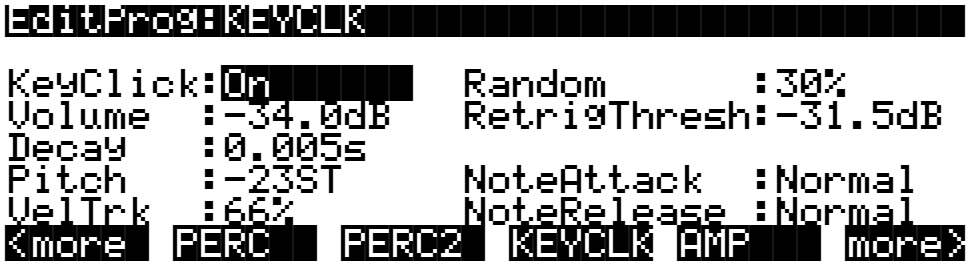
Parameter Group (Available for each combination of the Volume and Decay parameters on the PERC page)	Range of Values
Percussion Level	0 to 24.0 dB
Decay Time	0.10 to 5.10 seconds, in .02-second increments
Organ Volume Level	-12.0 to 12.0 dB

PercLevel, DecayTime, OrgLevel

With these parameters you can control the amplitude and decay time of the percussion effect for all combinations of the Volume and Decay parameters (on the PERC page). You can also adjust the level of the organ relative to the percussion, for accurate emulation of classic organs.

The KEYCLK Page

The Key Click feature adds a decaying burst of pitched noise to the attack of notes. Unlike the percussion, the key click is "multi-triggered," which means that every new note will trigger it. The parameters on this page primarily control the decay, volume, and pitch of the key click.



Parameter	Range of Values
Key Click	Off, On
Volume	-96.0 to 0.0 dB, in .5-dB increments
Decay Time	0.005 to 1.280 seconds, in .005-second increments
Pitch	-129 to 127 Semitones
Velocity Tracking	0–100%
Random	0–100%
Retrigger Threshold	-96.0 to 0.0 dB, in .5-dB increments
Note Attack	Normal, Hard, PercHard
Note Release	Normal, Hard

KeyClick

This is where you turn Key Click on or off.

Volume

This parameter sets the level of the keyclick; the noise decays from the level you set here. This level is scaled by the drawbar levels, as well as the expression pedal level.

Decay

Sets the basic decay time of the noise envelope. Smaller values produce a shorter burst.

Pitch

Sets the basic pitch of the key click noise, relative to the highest tone wheel's pitch. The pitch is actually controlled by a steep lowpass filter applied to white noise.

VelTrk

Controls the degree to which key velocity affects the key click volume. A value of zero means that the key velocity has no effect on the key click volume (which is like a real tone wheel organ). Other values add volume as the velocity increases.

Random

Controls the degree to which a random amount of amplitude variation is added to the key click.

ReTrigThresh

This parameter lets you set the volume level below which key click must decay before it will be retriggered.

Note Attack

Controls the attack characteristic of notes. **Normal** provides a smoothed attack, while a setting of **Hard** has an instant attack and will produce an audible click, in addition to any amount of key click specified with the other parameters on this page (you might prefer not to specify any additional key click when you use this setting). **PercHard** sets a hard attack level for percussion only; notes without percussion use a normal attack.

Note Release

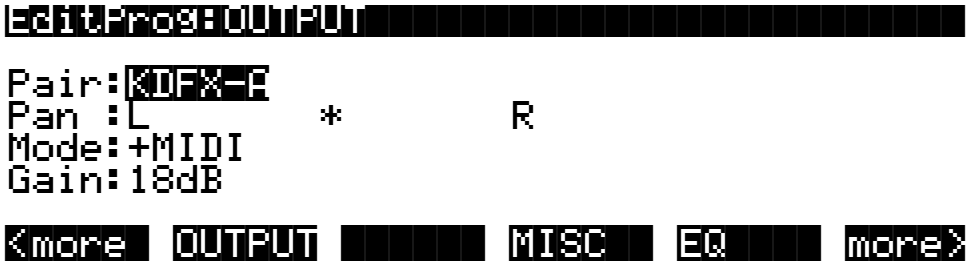
Controls the release characteristic of notes. A setting of **Normal** has a smoothed release, while a setting of **Hard** has an instant release. **Hard** will produce an audible click.

The AMP Page

Assign amplitude controllers on this page, which is similar to the F4 AMP page for VAST programs (see page 6-28 for a full description. The only difference between the AMP pages for KB3 programs and VAST programs is that the KB3 version doesn't have the KeyTrk, VelTrk, and Pad parameters.

The OUTPUT Page

Use this page to route the programs signal from the sound engine to KDFX. This page is similar to the VAST-program version; the only difference is that it doesn't have the CrossFade and XFadeSense parameters. See page 6-29 for a description of the OUTPUT-page parameters.



The MISC Page

The MISC page contains an assortment of control parameters, including Leslie speed control and vibrato/chorus selection.

Parameter	Range of Values
Preamp/Expression Response	Off, On
Leakage	-96.0 to 0.0 dB, in .5-dB increments
Leak Mode	None, Type A, Type X, Type Y, Type Z
Speed Control	Slow, Fast
Vibrato/Chorus Control	Off, On
Vibrato/Chorus Type Selection	Vib1, Vib2, Vib3, Chor1, Chor2, Chor3
Volume Adjust	-96 to 96 dB
Bend Range	-7200 to 7200 Cents
Sustain	Off, On
Sostenuto	Off, On

PreampResp

Set this parameter **On** or **Off** to enable or disable the preamp+expression pedal part of the KB3 model. Turning this **On** (the default) makes KB3 programs function like stock organs. The expression pedal in this case is more than a volume pedal; it actually functions like a "loudness control," varying the frequency response to compensate for the ear's sensitivity at different volumes. In addition, the preamp provides a deemphasis curve to compensate for the built-in tone wheel volume preemphasis. Turning preamp response **Off** emulates organs that have been modified to have a direct out (before the preamp and expression pedal).

Leakage

Controls the level of the simulated crosstalk and signal "bleed" of adjacent tone wheels in the model. This is provided to help "dirty up" the sound to make it a bit more realistic. A setting of -96 dB gives the purest tones; other values add more simulated leakage. This level is scaled by the drawbar levels, as well as the expression pedal level.

LeakMode

Selects between different leakage models, determining which leakage harmonics are emphasized. **TypeA** provides an overall tone wheel leakage, with all tone wheels leaking a small amount. **TypeX**, **TypeY**, and **TypeZ** emulate different degrees of drawbar leakage, where the leakage components correspond to the nine drawbars, instead of all the tone wheels.

SpeedCtl

Select either **Fast** or **Slow** to choose the speed of the rotary speaker emulation. On keyboard models, you can toggle between fast and slow speed using Assignable Controller Button **1** (**Mute** button **1**).

When you select a KB3 program, the K2600 sends several MIDI Controller messages both locally and to the MIDI Out port. One of those messages is Controller number 68, with a value corresponding to the value of SpeedCtl (**Slow** = 0, **Fast** = 127).

VibChorCtl

Choose **On** or **Off** to turn on or off either vibrato or chorus (as selected with the VibChorSel parameter). On keyboard models, you can toggle between on and off using Assignable Controller Button **2** (**Mute** button **2**).

When you select a KB3 program, the K2600 sends several MIDI Controller messages both locally and to the MIDI Out port. One of those messages is Controller number 95, with a value corresponding to the value of VibChorCtl (**Off** = 0, **On** = 127).

VibChorSel

Choose the vibrato or chorus program (there are three of each) you wish to use with this KB3 program. Note that you must set VibChorCtl (also on the MISC page) to **On** to hear the effect. On keyboard models, you can select the vibrato or chorus you want using Assignable Controller Buttons **3** and **4** (**Mute** buttons **3** and **4**.

When you select a KB3 program, the K2600 sends several MIDI Controller messages both locally and to the MIDI Out port. One of those messages is Controller number 93, with a value corresponding to the value of VibChorSel (**Vib1** = 0, **Vib2** = 36, **Vib3** = 58, **Chor1** = 79, **Chor2** = 100, and **Chor3** = 122).

VolAdjust

Provides an overall volume adjust for the KB3 model. Use this parameter to “normalize” KB3 programs with other programs.

BendRange

Controls the pitch bend range of this KB3 program.

Sustain

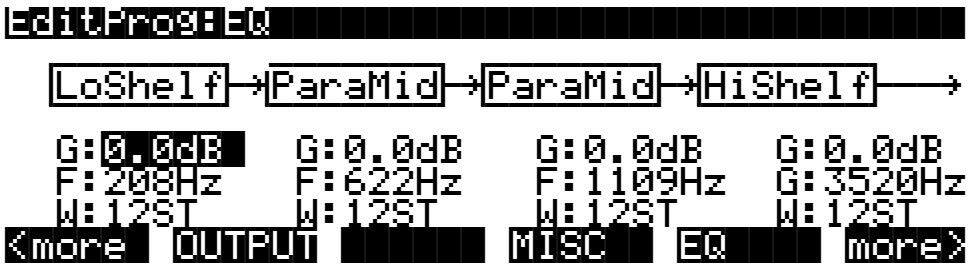
Set **On** or **Off** to enable or disable response to MIDI sustain (MIDI 64).

Sostenuto

Set **On** or **Off** to enable or disable response to MIDI sostenuto (MIDI 66).

The EQ Page

When you press the **EQ** soft button, you'll see a page that at first glance looks quite similar to the ALG page for a VAST program. The four blocks on this page, however, represent two shelving bands of equalization and two parametric bands. The KB3 EQ offered here, though, is not implemented as a true EQ section, instead it adjusts the volume of the tone wheels based on frequency. If the tone wheels are based on sine waves, then this acts a lot like a real EQ.



Parameter Group (Available for each EQ Block)	Range of Values
Gain	-24.0 to 24.0 dB, in 0.2-dB increments
Frequency	0 to 25088 Hz, in varying increments
Width	-128 to 128 Semitones, in 2-semitone increments

Each EQ section has Gain (G), Frequency (F), and Width (W) controls. Frequency controls the center frequency of the band. Width controls the bandwidth. Gain controls the amount of boost or cut.

All the Other Pages

The rest of the pages—LFO, ASR, FUN, etc.— are the same for KB3 programs as they are for VAST programs, so we won't describe them again here. Begin on page 6-40 to find descriptions of these pages.

Programming Tips

This section provides some starting points for creating your own KB3 programs. Remember that you'll have to start with one of the existing KB3 programs.

As described below, the most prominent difference between organ vintages is the number of tone wheels used. Keep in mind, however, that the sound of an actual tone wheel organ will depend not only on its age, but also on how well it has been maintained.

Octave folding, where an octave (or part of an octave) is repeated at the top or bottom of the keyboard, is handled automatically by KB3 Mode, emulating the folding done on actual tone wheel organs.

Early Tone Wheel Organs. Instruments of this period had 91 tone wheels. To get this sound, go to the TONEWL page, select 91 tone wheels, and set lowest pitch to C 1. Start with the **Junky** Wheel Volume Map and **Bob's** Organ Map. You may also want to increase the Key Click level, since this tends to become louder on older organs.

Middle Period Organs. To model one of these instruments, set 82 tone wheels and a low note of A 1. Use the **Mellow** Wheel Volume Map and **Eric's** Organ Map. Set Key Click to a moderate level.

The Classic B-3. For this sound, choose 79 tone wheels and set the low note to be C 2. The best settings here are the **Bright** Wheel Volume Map and **Peck's** Organ Map. You may also want to reduce the Key Click level.

Using a KB3 Program in a Setup

To get the rotary speaker effect to work properly, you'll need to make a few adjustments. On the Effects-mode page, make sure that FX Mode is **Auto**, and FX Chan is **Current**. Then go into the Studio Editor, and assign a KB3 program to one of the setup's zones. Then import the KDFX settings from a program that uses the rotary effect you want. If the setup has only one zone, you're finished. If the setup has more than one zone, you'll need to check each zone to see where the audio output for its program is going. If you don't want the rotary effect applied to the other programs in the setup, set those programs' outputs to a KDFX input that's different from the output of the KB3 program (you'll need to look at the setup's studio, find which FXBus uses the rotary-effect FX preset, then set the output of each non-KB3 zone so that it goes to a KDFX input that isn't routed to the FXBus using the rotary-effect FX preset).

Note: *Many of the factory KB3 programs use Studio **160 KB3 V/C -> Rotary**, which in turn uses FX Preset **779 KB3 FXBus** on FXBus1, and also uses the Aux Bus for the full rotary-speaker effect. Since FX Preset **779** uses 4 PAUs, there are no PAUs free for use on the other FXBuses. Consequently, when you use this studio, there's no way to route other zones through different effects. If you want a setup that uses a KB3 program with rotary-speaker effects, but you also want other programs in the setup that don't use the same effect, use a studio other than Studio **160**.*

Emulating a Two-Manual Organ

You can use Setup Mode to emulate a two-manual keyboard. However, you can only use a KB3 program in one zone of the setup (use a regular K2600 organ program in the other zone). Also, you will have limited polyphony on the non-KB3 channel. The actual number of voices that will be available on the non-KB3 channel depends on the number of tone wheels used by the KB3 program. (You may be able to reduce the number of tone wheels used, depending on the note range of the zone.)

Programs Using 2PARAM SHAPER

If you are running a KB3 program, you will not be able to simultaneously play K2600 programs that use the 2PARAM SHAPER DSP function on another channel. (Actually, the program will still play, but the 2PARAM SHAPER DSP function will be bypassed.)

Shaper AMP (!AMP)

Although the !AMP DSP function is not part of the KB3 editor, we have used the !AMP to help "dirty-up" several programs. This allows us to recreate the tube distortion characteristics of classic drawbar organ pre-amps.

Using the VAST Program Editor on a KB3 Program

KB3 Mode lets you use certain VAST-related parameters— such as the !AMP DSP function mentioned above—that are not found within the KB3 editor. There's a "back door" that takes you to these non-KB3 parameters: if you mark a page (with the Mark button) while editing a VAST program, you can jump to that page (with the Jump button) while you're editing a KB3 program.

When editing a KB3 program, we recommend that you edit only those parameters on the KB3-editing pages. We realize, however, that some power users (you know who you are!) will want to hear the results of applying VAST parameters to KB3 programs. Before you start experimenting, please be aware of the following:

- **Turn your volume down!** In KB3 mode, most resources are being used in much different ways than they in a typical VAST program; the results of applying VAST parameters to KB3 programs can be extremely unpredictable.
- Most changes made to these parameters do not take effect until save the program and reselect it.
- You cannot import a KB3 program layer into a regular VAST program.