K2600

Musician's Guide

KURZWEIL

Music Systems

Part Number: 910330 Rev. A

CAUTION

RISK OF ELECTRIC SHOCK

DO NOT OPEN

CAUTION: TO REDUCE THE RISK OF ELECTRIC SHOCK,
DO NOT REMOVE THE COVER
NO USER SERVICEABLE PARTS INSIDE
REFER SERVICING TO QUALIFIED SERVICE PERSONNEL

The lightning flash with the arrowhead symbol, within an equilateral triangle, is intended to alert the user to the presence of uninsulated "dangerous voltage" within the product's enclosure that may be of sufficient magnitude to constitute a risk of electric shock to persons.

The exclamation point within an equilateral triangle is intended to alert the user to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the product.

IMPORTANT SAFETY & INSTALLATION INSTRUCTIONS

INSTRUCTIONS PERTAINING TO THE RISK OF FIRE, ELECTRIC SHOCK, OR INJURY TO PERSONS

WARNING: When using electric products, basic precautions should always be followed, including the following:

1. Read all of the Safety and Installation Instructions and Explanation of Graphic Symbols before using the product.
2. This product must be grounded. If it should malfunction or break down, grounding provides a path of least resistance for electric current to reduce the risk of electric shock. This product is equipped with a power supply cord having an equipment-grounding conductor and a grounding plug. The plug must be plugged into an appropriate outlet which is properly installed and grounded in accordance with all local codes and ordinances.

   DANGER: Improper connection of the equipment-grounding conductor can result in a risk of electric shock. Do not modify the plug provided with the product - if it will not fit the outlet, have a proper outlet installed by a qualified electrician. Do not use an adaptor which defeats the function of the equipment-grounding conductor. If you are in doubt as to whether the product is properly grounded, check with a qualified serviceman or electrician.
3. **WARNING:** This product is equipped with an AC input voltage selector. The voltage selector has been factory set for the mains supply voltage in the country where this unit was sold. Changing the voltage selector may require the use of a different power supply cord or attachment plug, or both. To reduce the risk of fire or electric shock, refer servicing to qualified maintenance personnel.
4. Do not use this product near water - for example, near a bathtub, washbowl, kitchen sink, in a wet basement, or near a swimming pool, or the like.
5. This product should only be used with a stand or cart that is recommended by the manufacturer.
6. This product, either alone or in combination with an amplifier and speakers or headphones, may be capable of producing sound levels that could cause permanent hearing loss. Do not operate for a long period of time at a high volume level or at a level that is uncomfortable. If you experience any hearing loss or ringing in the ears, you should consult an audiologist.
7. The product should be located so that its location or position does not interfere with its proper ventilation.
8. The product should be located away from heat sources such as radiators, heat registers, or other products that produce heat.
9. The product should be connected to a power supply only of the type described in the operating instructions or as marked on the product.
10. This product may be equipped with a polarized line plug (one blade wider than the other). This is a safety feature. If you are unable to insert the plug into the outlet, contact an electrician to replace your obsolete outlet. Do not defeat the safety purpose of the plug.
11. The power supply cord of the product should be unplugged from the outlet when left unused for a long period of time. When unplugging the power supply cord, do not pull on the cord, but grasp it by the plug.
12. Care should be taken so that objects do not fall and liquids are not spilled into the enclosure through openings.
13. The product should be serviced by qualified service personnel when:
    - A. The power supply cord or the plug has been damaged;
    - B. Objects have fallen, or liquid has been spilled into the product;
    - C. The product has been exposed to rain;
    - D. The product does not appear to be operating normally or exhibits a marked change in performance;
    - E. The product has been dropped, or the enclosure damaged.
14. Do not attempt to service the product beyond that described in the user maintenance instructions. All other servicing should be referred to qualified service personnel.
15. **WARNING:** Do not place objects on the product's power supply cord, or place the product in a position where anyone could trip over, walk on, or roll anything over cords of any type. Do not allow the product to rest on or be installed over cords of any type. Improper installations of this type create the possibility of a fire hazard and/or personal injury.

RADIO AND TELEVISION INTERFERENCE

WARNING: Changes or modifications to this instrument not expressly approved by Young Chang could void your authority to operate the instrument.

IMPORTANT: When connecting this product to accessories and/or other equipment use only high quality shielded cables.

NOTE: This instrument has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This instrument generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this instrument does cause harmful interference to radio or television reception, which can be determined by turning the instrument off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the instrument and the receiver.
- Connect the instrument into an outlet on a circuit other than the one to which the receiver is connected.
- If necessary consult your dealer or an experienced radio/television technician for additional suggestions.

NOTICE

This apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

AVIS

Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class B prescrites dans le Reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

SAVE THESE INSTRUCTIONS

Young Chang Distributors

Contact the nearest Young Chang office listed below to locate your local Young Chang/ Kurzweil representative.

Young Chang America, Inc.
P.O. Box 99995
Lakewood, WA 98499-0995
Tel: (253) 589-3200
Fax: (253) 984-0245

Young Chang Co.
178-55 Gajwa-Dong
Seo-Ku, Inchon, Korea 404-714
Tel: 011-82-32-570-1380
Fax: 011-82-32-570-1218

Young Chang Akki Europe GmbH
Industriering 45
D-41751 Viersen
Germany
Tel: 011-49-2162-4491
Fax: 011-49-2162-41744

Young Chang Canada Corp.
250 Shields Court, Unit #11
Markham, Ontario L3R 9W7
Tel: (905) 948-8052
Fax: (905) 948-8172

Contents

Chapter 1 Introduction

Chapter 2 Startup

Chapter 3 User Interface Basics

Chapter 4 The Operating Modes

Chapter 5 Editing Conventions

Chapter 6 Program Mode and the Program Editor

Chapter 7 Setup Mode and the Setup Editor

Chapter 8 Quick Access Mode and the Quick Access Editor

Chapter 9 Basic Effects Mode

Chapter 10 MIDI Mode

Chapter 11 Master Mode

Chapter 12 Song Mode and the Song Editor

Chapter 13 Disk Mode

Chapter 14 Sampling and Sample Editing

Chapter 15 Advanced KDFX

Chapter 16 DSP Functions

Chapter 17 FUNS

Chapter 18 Other Editors

Chapter 19 Audio Outputs

Chapter 20 Programming Examples

Appendix A K2600 Boot Loader

Index

Chapter 1 Introduction

Welcome!

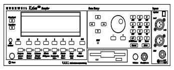
Congratulations, and thanks for purchasing a Kurzweil/Young Chang K2600 Series instrument. There are seven K2600 Series models, for a wide range of applications:

K2600	76 keys
K2600S	K2600 with built-in sampler
K2600X	88 keys
K2600Xs	K2600X with built-in sampler
K2600R	Rack-mount model
K2600Rs	K2600R with built-in sampler
K2600 AES	Top-end model, with built-in sampler, digital I/O, DMTi, and sample libraries

Whichever model you have, you're looking at one of the most powerful music workstations available. The K2600 Series instruments are packed with great acoustic, electric, and synth sounds—combined with some of the most advanced synthesis features available, which you can use to create almost any sound imaginable. The K2600 incorporates most of the features of the award-winning K2000 and the K2500, and provides several new features as well.

This manual and the accompanying *Musician's Reference*, as well as the video tutorial, will get you started with your new instrument. You'll definitely want to keep the manuals handy as you become an advanced user, too.

For K2600R Owners



Throughout the *Musician's Guide* and *Musician's Reference* we've simplified things by saying just "K2600" any time we're referring to features that are common to all instruments in the K2600 Series. Obviously, though, there are some differences between the rack and keyboard models; we'll point them out where they occur and mark them with a handy K2600R margin symbol like the one to the left of this paragraph. If you own a keyboard model of the K2600, you may want to read the K2600R-specific information anyway, since much of it also applies when your K2600 is a MIDI slave (when using a computer-based sequencer, for example).

Overview of the K2600

The K2600 has been designed to be a versatile instrument both for performance, and for multi-timbral sequencing and recording. Its Variable Architecture Synthesis Technology lets you build sounds from realistic instrumental samples and sampled synth waveforms—then modify the nature of those sounds through a wide variety of digital signal-processing (DSP) functions. The K2600 also generates its own synth waveforms, which can be combined with the samples or used on their own. The K2600 packs 12 Megabytes of onboard sound ROM. You can add up to four optional ROM sound blocks (there are two currently available), and you can load samples from disk into optional sample RAM.

Before we get into explaining Variable Architecture Synthesis, here are a few of the features that by themselves make the K2600 an impressive stage and studio machine. It's fully multi-timbral—different programs can be played on each MIDI channel. It's 48-note polyphonic, for a full sound no matter how many chords you play. There's an on-board digital effects processor, providing up to five simultaneous effects, including real-time effects control, internally or via MIDI.

In addition to the standard stereo audio output pair, there are eight separate outputs that can be configured as stereo pairs, or as individual mono outputs. These are balanced outputs, which you can connect to either balanced or unbalanced inputs.

The K2600 offers two SIMM sockets (single, in-line memory modules) so you can add optional Sample RAM, where you can store samples that you've loaded from disk. You can add up to 128 megabytes of Sample RAM. (Sample RAM is not battery-backed, so RAM samples are erased from memory when you power down.)

For offline storage, there's also a floppy drive and two ***SCSI*** ports, so you can store files on floppies or on an external hard disk or load them from a CD-ROM drive. The two SCSI ports make it easy to chain multiple SCSI devices together. There's also provision for an internal SCSI hard disk. You'll find all this storage potential extremely useful for saving and loading samples, which can be transferred to and from the K2600 using the standard MIDI sample transfer format, or the faster, parallel SMDI sample transfer format (SCSI Musical Data Interchange). See the *Musician's Reference* for information about MIDI and SMDI sample transfers.

The K2600's battery-backed program RAM can store about 800 of your own programs, or about 60,000 notes recorded in the sequencer. This sequencer (Song mode) lets you play back MIDI type 0 or 1 sequences, record and play back your own songs, and record multi-timbral sequences received via MIDI. For more onboard storage you can add the P/RAM option, which will increase your battery-backed RAM to about 1500K (1.5M), enough to store hundreds of additional programs, setups, songs, and other ***objects***. (Without the P/RAM option, the K2600 has about 500K of program RAM.)

The Local Keyboard Channel feature enables you to use the K2600's multi-zone setups even if you're using a MIDI controller that can transmit on only one channel. The K2600 will also rechannelize incoming MIDI information and send it to its MIDI Out port, enabling you to control additional synths on eight different channels.

An optional sampling feature is available, allowing you to make your own mono or stereo samples using analog or digital inputs. With the sampling option, you can also use Live mode, which enables you to take an input signal (at the analog sampling input), and route it through the K2600's VAST algorithms—so you can apply Kurzweil DSP and effects to any sound The sampling option also gives you digital audio output (stereo optical or unbalanced stereo XLR).

There's also the digital input/output (I/O) option, which provides eight channels for digital audio input and output (as well as a stereo digital output).

And, of course, there's the incomparable Kurzweil sound. The K2600 comes to you with over 400 programs (called patches, presets, voices, etc. on other synths)—238 in ROM and 200 more in RAM. There are also about 200 multi-zone performance setups. Many of these setups use note triggers to play factory-recorded songs that provide grooves and arpeggiation that make great templates for performance or recording.

VAST Synthesis

Variable Architecture Synthesis Technology gives the K2600 its unprecedented flexibility. While many other synthesizers offer a fixed set of ***DSP*** tools (typically filtering, pitch, and amplitude modulation) the K2600's Variable Architecture lets you arrange a combination of any five DSP functions from a long list of choices. The functions you choose define the type of synthesis you use.

Each layer of every program has its own DSP architecture, which we call an ***algorithm***. Within each algorithm, you can select from a variety of DSP functions. Each function can be independently controlled by a variety of sources including LFOs, ASRs, envelopes, a set of unique programmable functions (FUNs), as well as any MIDI control message. The many different DSP functions and the wealth of independent control sources give you an extremely flexible, truly vast collection of tools for sound creation and modification. When you're ready to jump in and start creating and editing programs, turn to Chapter 6.

V▸A▸S▸T▸

Variable **A**rchitecture **S**ynthesis **T**echnology

KB3 Tone Wheel Emulation

In addition to VAST synthesis, the K2600 offers many oscillator-based programs that give you the classic sound of tone-wheel organs like the Hammond B 3™. KB3 mode, as we call it, is completely independent of VAST, and has its own set of editing procedures. You'll find details in Chapter 6.

How the K2600 Works

The K2600 integrates three MIDI-driven components: a MIDI controller (the keyboard, or an external MIDI controller, a sound engine, and a global effects processor (KDFX). The sound engine responds to the MIDI events generated by the MIDI controller, and turns them into sounds that are processed within the variable architecture of the algorithms—or by oscillators for KB3 programs. The resulting sound can then be routed through KDFX and to the MIX and separate audio outputs.

How to Use This Manual

This manual includes the following:

- How to connect and power up your K2600, getting around the front panel, and a brief description of the operating modes
- Basic editing, including the normal operations of each operating mode
- The advanced programming features that make the K2600 so powerful—a number of familiar synthesis tools and quite a few new ones

Even if you're a complete techie, you should read Chapter 3, *User Interface Basics*. Here you'll get a tour of the K2600's front panel and learn how to navigate through the major functions.

Chapter 4, *The Operating Modes* describes the concept and operation of the K2600's eight operating modes, with a brief description of each. Chapters 6 through 13 describe each mode in detail, including the editors contained within them. Chapters 14 through 17 discuss the advanced editing features. Chapter 19 shows you how to use the multiple audio outputs to suit your needs. Chapter 20 is a programming tutorial, giving you specific examples of many of the K2600's programming functions. Some are basic, some are advanced. By working through the tutorials, you'll become familiar with many synthesis techniques, and you'll get a first-hand look at how to unleash the power of the K2600.

When manual text appears in boldface italic (***like this***), you'll find it described in the Glossary in the *Musician's Reference*. Only the first one or two occurrences of these words are highlighted.

You'll also see text in bold face (**like this**). We use bold face to indicate parameter values, K2600 objects (like programs and setups), filenames (like songs and samples), and front-panel buttons (like the **Program** button and the soft buttons).

The best way to read this manual is with your K2600 in front of you. By trying the examples we give to illustrate various functions, you can get a quick understanding of the basics, then move on to the more advanced features.

If you like to figure out your equipment for yourself, and normally use manuals only as reference sources, you'll probably get the most use out of the *Musician's Reference* that accompanies this manual, which contains brief descriptions of the K2600's major operating features, and all sorts of useful lists—***programs***, ***keymaps***, ***algorithms***, effects, ***control sources***, and more. A glossary and complete specifications for the K2600 are also included in the *Musician's Reference*.

Do I Have Everything?

Your K2600 shipping carton should include the following in addition to your instrument:

- Power cable
- Sustain pedal (keyboard models only)
- Instructional video
- Two manuals: the *Musician's Guide* (this manual) and the *Musician's Reference*
- A set of accessory disks

If you don't have all of these components, please call your Kurzweil/Young Chang dealer.

The Accessory Disks

The set of accessory disks that came with your K2600 should include the following:

K2600 Base ROM Objects

This disk contains two files. The first file is named OBJKBXXX.K26 (where XXX is the version number). The second file is named OBJKLXXX.K26 (where XXX is the version number). These disks provide a backup of the ROM Objects installed in your K2600 at the factory.

You will not normally need to use these disks, however you should keep them in a safe place in case you need to reinstall this version of the operating system. For example, if you have installed a newer version of the objects (we make new versions available for download from our web site) and you encounter any problems, you might need to go back and install the version on these disks.

OBJKB contains all of the objects in a stock K2600 without sampling. OBJKL contains the Live Mode objects, which are included in a unit with the sampling option installed.

When you install a new version of the operating system, you'll need to load all the object files for the options you have. This includes the object files for the Orchestral and Contemporary ROM blocks, if you have them. Load the system files first, then all of the object files, then exit the Boot Loader by pressing the **Run System** soft button. See Appendix A if you need help with operating the Boot Loader.

K2600 General MIDI Compatibility

This is a two-disk set, intended to make the K2600 compatible with General MIDI (GM) programs. The K2600 is not a General MIDI instrument. GM was intended as a least common denominator and is normally found on low-end instruments, not instruments intended for the high-end professional.

Nonetheless, we have created a GM compatibility file that can be loaded into the K2600. It contains samples and therefore must be loaded into the unit each time you power up.

There are three different GM files. Which file you use depends on which ROM blocks you have in the K2000. The disk that comes with your unit is for a K2600 without any additional ROM blocks. If you purchase the Orchestral or Contemporary ROM block, you will receive two other files—one for a unit with just the Orchestral ROM added, and one for a unit with the Orchestral and Contemporary ROM added. You can also download these files from the Downloads page of our web site:

http://www.youngchang.com/kurzweil/html/downloads.html

For complete documentation on how to use the GM file, please see *QA Banks and General MIDI Files* on page 10-15.

K2600 Demos

This disk contains several directories of demo song files, which show off some of the capabilities of your K2600. In addition, the disk contains two other files. **KDFXTUTR.K26** contains objects used in the tour of Effects mode that starts on page 9-12. **VIDEO.K26** is referred to in the Disk mode section of the video tutorial.

Most of the song demos can simply be loaded into the K2600 and played from Song mode. However, two demo files require that your K2600 have the sampling option installed. The demo files **LMFBDMO1.K26**, and **LMFBDMO2.K26** demonstrate internal feedback loops using a chain of VAST->KDFX->sampler->VAST->KDFX—known as Live mode. Because a Master table is included, your SampleMode-page settings will be set automatically when you load the demo file.

***Caution**: Programs that cause internal feedback can get extremely loud, so it's important to be careful after loading these objects. Live-mode programs are inherently unstable when you're in Live mode and the sampling source (the Src parameter in the Samplemode page) is set to **Int**. These programs are capable of producing tremendously loud feedback loops with a relatively small amount of gain. In modes where the provided studio is not the active studio, the compressor that keeps these feedback chains under control isn't operating, and the loop can grow out of control very quickly. If you intend only to listen to the demo, and not to examine the programming, we recommend that you delete the song file (and its dependent objects) from your K2600 when you are done listening. We also recommend that you go to the SampleMode page and set the Src parameter to **Ext** or the Mode parameter to something other than **LiveIn**, to help to avoid any unintended feedback. If you want to look at the program settings, lower the volume on your instrument, and read the more detailed description of Live mode beginning on page 14-41.*

K2600 Farm & Extras

This disk contains over 1000 extra programs and effects, all cultivated from the "farm" where we grow our programs and effects.

The **K2500FRM** directory contains over 1000 extra programs, organized in files by type of sound. This set of programs was originally developed for the K2500, hence the name.

The **KDFXFARM** directory contains two subdirectories. **LIVEMODE** contains a set of files containing extra programs using Live mode (which requires having the sample option in your K2500). **MOREKDFX** contains some FX Presets and a number of Programs that use KDFX, as well as a couple of blank studios.

PIANOFRM contains extra programs that specifically use the 4-megabyte stereo piano.

VOCODER contains programs and setups that are designed to be used with the Vocoder feature. You can read about the Vocoder in detail on page 11-4.

K2600 Additional Programs and Factory RAM Banks

This disk contains even more programs from our program farm.

The **MOREPRGS** directory contains several files of extra programs.

The **RAMBANKS** directory contains a backup file of the programs that were loaded into the 300s and 400s banks when your unit was shipped from the factory.

K2500 Compatibility

This disk contains a file of all the ROM objects in the K2500. You would use this file if you have songs created originally for a K2500. See *Loading Older Setup Versions* on page 7-2 for information about using older objects with the K2600.

K2000 Compatibility

These two disks contain a file of all the ROM objects in the K2000. You would use this file if you have songs created originally for a K2000. Again, see *Loading Older Setup Versions* on page 7-2.

Chapter 2 Startup

If hooking up new gear is familiar to you, and you just want to get going, here's a quick description of all the basic things you need to cover to get started with your K2600. The procedure is pretty much the same whether you have a rack or a keyboard model. If you need more information, thorough descriptions of each step follow. In either case, check out *Playing the Presets* on page 2-5.

Quick Startup Checklist

All Models

1. Most likely, the lithium "coin" battery that preserves user-defined programs and other data is already installed, but it may be packaged separately. In that case, see *Battery Replacement* on page 8-2 of the *Musician's Reference*, and install the battery.

2. If you have one of the keyboard models, set it on a hard, flat, level surface. If you have a K2600R, mount it in a standard 19-inch MIDI rack, or set it on a hard, flat, level surface. Make sure to leave plenty of room for ventilation.

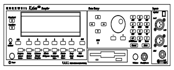
3. Connect the power cable.

4. Make sure your sound system is at a safe volume level.

5. Plug in a pair of stereo headphones or run standard (1/4-inch) audio cables from your amplifier or mixer to the MIX audio outputs on the K2600. (Use the MIX L out for mono.)

K2600R Owners Only

Your K2600R needs to be able to receive and interpret MIDI messages from whatever MIDI controller you're using.



1. Connect a MIDI cable from the MIDI Out port of your MIDI controller to the MIDI In port of the K2600R.

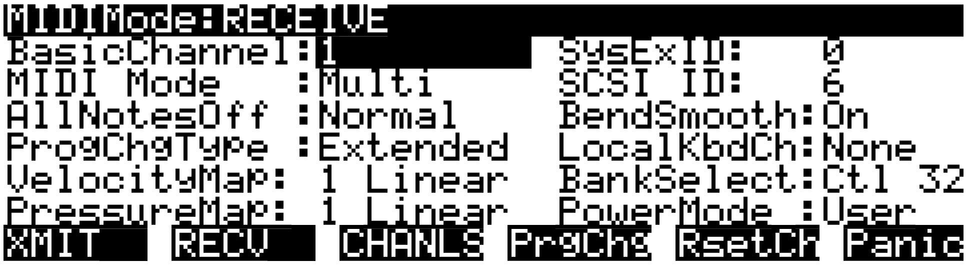
   To get setups (and other features) to work properly, you may need to make an adjustment to the K2600R.

2. Press the **MIDI** mode button (it's in the bottom row of buttons under the LCD display) to enter MIDI mode.

3. Look for the word RECV in the display, then press the button directly beneath it. The top line of the display now says MIDIMode:RECEIVE, which is where you configure the K2600R to respond to your MIDI controller.

   The following figure shows you what the display looks like. Notice how one of the ***parameters*** (probably BasicChannel) has highlighting (reversed color) around its value. This highlighting is called the cursor, and it indicates which parameter (or value) is selected for editing. You can select different parameters using the cursor buttons (the ones just to the right of the display; they're labeled with **Up**/**Down**/**Left**/**Right** arrows). Press one of the cursor buttons, and the cursor moves correspondingly. In our example below, the BasicChannel parameter is selected; its value is **1**.

4. Select the LocalKbdCh parameter (that stands for Local Keyboard Channel), then use the dial (we call it the Alpha Wheel) to change the value so that it matches the channel that your controller is sending on.



```
MIDIMode:RECEIVE
BasicChannel:1          SysExID:    0
MIDI Mode   :Multi      SCSI ID:    6
AllNotesOff :Normal     BendSmooth:On
ProgChgType :Extended   LocalKbdCh:None
VelocityMap: 1 Linear   BankSelect:Ctl 32
PressureMap: 1 Linear   PowerMode :User
XMIT    RECV    CHANLS  PrgChg  RsetCh  Panic
```

5. Press the button labeled **Exit** to get out of MIDI mode. You're ready to go. See page 10-7 for more information about the local keyboard channel.

Start Jamming!

1. Power up your K2600 and check out some of the programs and setups. The K2600 starts up in either Program mode or Setup mode—whichever of the two modes it was in when it was last shut down (or given a soft reset). Press the button labeled **Program** or **Setup** to switch modes.

2. If you hear distortion, reduce the gain on your mixing board, or use the pad if it has one.

3. Scroll through the program list with the Alpha Wheel.

4. Press the **Quick Access** mode button and use the alphanumeric buttons to select from programmable banks of ten programs or setups.

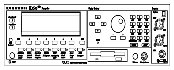
5. If you don't hear anything, review these steps, or check the Troubleshooting section in the *Musician's Reference*.

Startup—the Details

This section walks you through the hookup of your K2600. We'll take a look at the rear panel, then describe the power, audio, and other cable connections.

Before You Start...

Don't connect anything until you make sure your K2600 is properly and safely situated. Also, if your K2600 has been out in the cold, give it time to warm up to room temperature before starting it, since condensation may have formed inside the K2600.



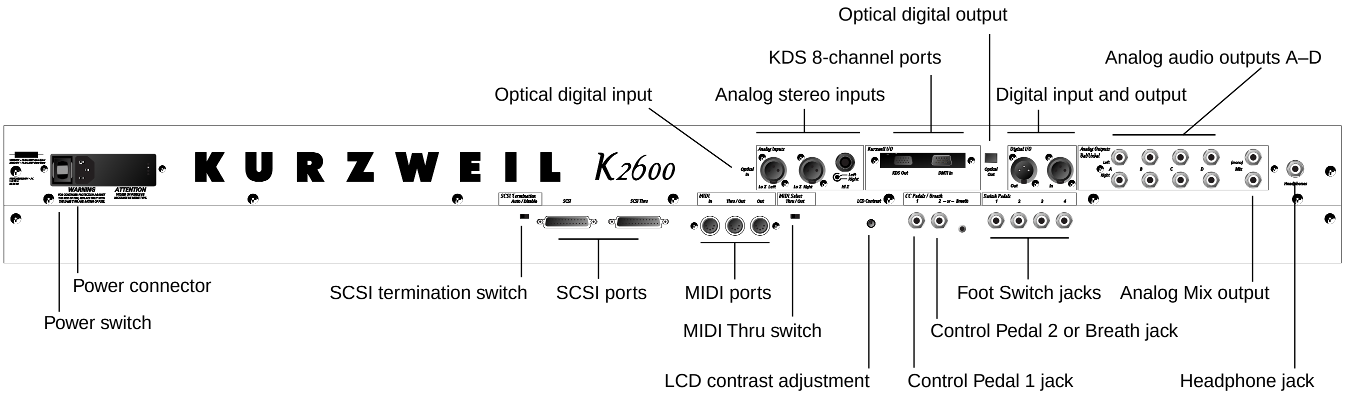
You'll probably be mounting a K2600R in a standard 19-inch MIDI rack (it takes up three rack spaces). If you will be transporting the rack containing the K2600R, be sure to support the back of the K2600R within the rack. If you're not installing the K2600R in a rack, it should rest on a hard flat surface. In this case, it *must* rest on its rubber feet (supplied in the packing carton), and *not* on the bottom panel.

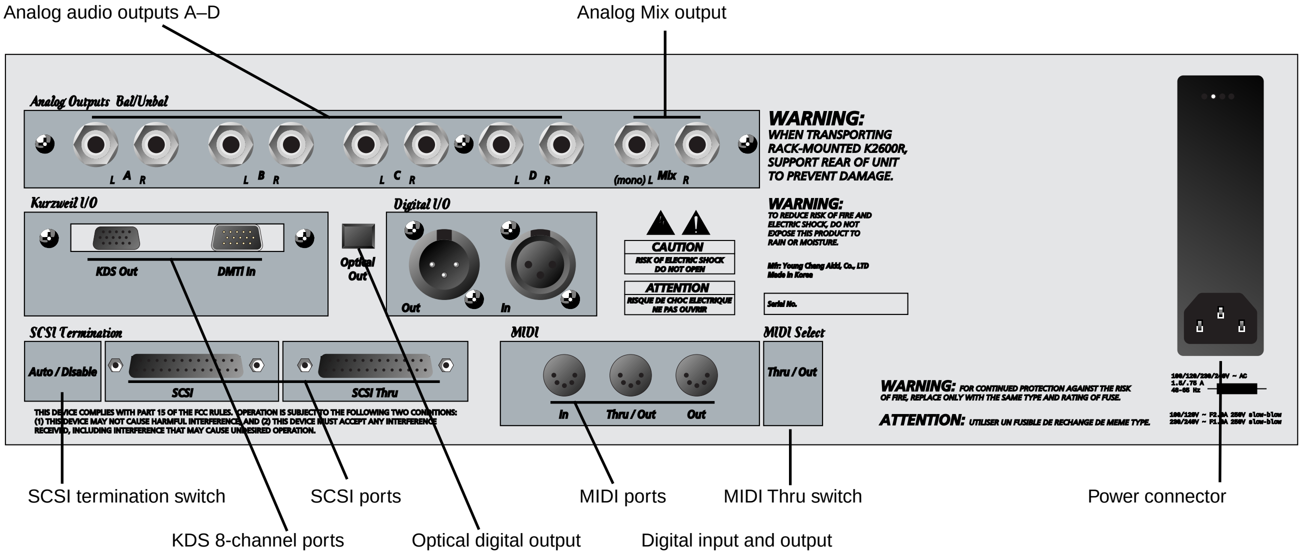
Never block the ventilation openings on the bottom or rear panels; doing so can cause overheating and serious damage. To provide adequate ventilation, the rear panel should be at least four inches from any vertical surface.

Connecting the Power Cable (Line Cord)

The K2600 runs on AC power: 100, 120, 220, or 240 volts at 50–60 Hz. Your dealer will set the voltage switch to match the voltage in your area. The voltage level is set with a selector on the rear panel of the keyboard models of the K2600. You shouldn’t have to adjust this, and we recommend that you don’t.

When you’ve connected the cable at the K2600 end (as you face the back of the K2600, the power connection is at the left—on the K2600R, it’s on the right), plug it into a grounded outlet. If your power source does not have the standard three-hole outlet, you should take the time to install a proper grounding system. This will reduce the risk of a shock.




Analog audio outputs A–D
Analog Mix output
Analog Outputs Bal/Unbal
L A R
L B R
L C R
L D R
(mono) L Mix R
WARNING:
WHEN TRANSPORTING RACK-MOUNTED K2600R, SUPPORT REAR OF UNIT TO PREVENT DAMAGE.
Kurzweil I/O
KDS Out
DMTI In
Optical Out
Digital I/O
Out
In
WARNING:
TO REDUCE RISK OF FIRE AND ELECTRIC SHOCK, DO NOT EXPOSE THIS PRODUCT TO RAIN OR MOISTURE.
CAUTION
RISK OF ELECTRIC SHOCK DO NOT OPEN
ATTENTION
RISQUE DE CHOC ELECTRIQUE NE PAS OUVRIR
Mfr: Young Chang Akki Co., LTD
Made in Korea
Serial No.
SCSI Termination
Auto / Disable
SCSI
SCSI Thru
MIDI
In
Thru / Out
Out
MIDI Select
Thru / Out
WARNING: FOR CONTINUED PROTECTION AGAINST THE RISK OF FIRE, REPLACE ONLY WITH THE SAME TYPE AND RATING OF FUSE.
ATTENTION: UTILISER UN FUSIBLE DE RECHANGE DE MEME TYPE.
THIS DEVICE COMPLIES WITH PART 15 OF THE FCC RULES. OPERATION IS SUBJECT TO THE FOLLOWING TWO CONDITIONS: (1) THIS DEVICE MAY NOT CAUSE HARMFUL INTERFERENCE, AND (2) THIS DEVICE MUST ACCEPT ANY INTERFERENCE RECEIVED, INCLUDING INTERFERENCE THAT MAY CAUSE UNDESIRED OPERATION.
SCSI termination switch
SCSI ports
MIDI ports
MIDI Thru switch
Power connector
KDS 8-channel ports
Optical digital output
Digital input and output

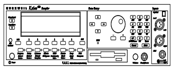
Connecting the Audio Cables

After you've turned down the level on your sound system, connect the K2600's analog audio outputs to your sound system using a pair of stereo or mono audio cables. Mono cables will always work, but if you're going into balanced inputs, use stereo cables for a better signal-to-noise ratio. The K2600's analog outputs are balanced, and generate a "hotter" signal than previous Kurzweil instruments.

You'll find ten 1/4-inch jacks near the top of the rear panel. For now, connect one end of each audio cable to your mixing board or PA system inputs, and connect the other end to the jacks marked MIX L and R on the rear panel of the K2600. If you have only one input available, use the K2600's MIX L output to get the full signal in mono. You'll find more about audio configurations—including digital output—in Chapter 19.

Connecting MIDI

The simplest MIDI configuration uses a single MIDI cable: either from the MIDI Out port of your K2600 to the MIDI In port of another instrument, or from the MIDI Out port of your MIDI controller to the MIDI In port of the K2600 or K2600R. There are all sorts of possible configurations, including additional synths, personal computers, MIDI effects processors, and MIDI patch bays. Depending on your system, you may want to use the K2600's MIDI Thru port to pass MIDI information from a MIDI controller to the K2600 and on to the next device in your system. You can also connect MIDI devices to the K2600's MIDI Out port, which can send channelized MIDI information from the keyboard or through the K2600 from your MIDI controller. See the discussion of the Local Keyboard Channel parameter on page 10-7.



Note: You can perform a quick check of the K2600R and your audio system without a MIDI controller connected to the K2600R. Hold down the **Cancel** button on the alphanumeric pad and press any other alphanumeric button, and you should hear notes. See the discussion of the Panel Play feature on page 3-10 for more information.

Connecting SCSI

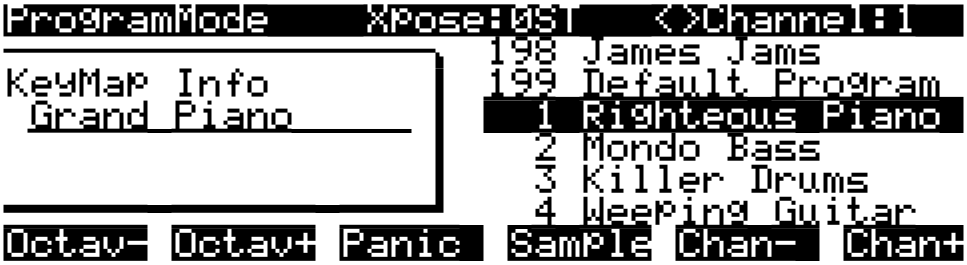
You may not have a hard disk or other SCSI device to connect to your K2600 right away, but if you do, you can connect it to either of the SCSI ports. Please read the following information carefully; it's very important. Also, there's a collection of SCSI tips in Chapter 6 of the *Musician's Reference*.

CAUTION: *If you plan to connect any SCSI devices (a hard disk, for example) to the K2600, you must terminate your SCSI chain properly. Turn to Chapter 13 and read the section called SCSI Termination. You can lose data if your system isn't terminated properly.*

Switching On the Power

- Keyboard models: the power switch is on the rear panel. If you're facing the keyboard, it's just to the right of the power cable connection.
- Rack-mount models: the power switch is located at the lower left of the front panel.

When you power up, the display briefly show some startup information. The Program mode display then appears. It looks like the diagram below (the programs shown in the diagram don't necessarily exist).



```
ProgramMode     Xpose:0ST    <>Channel:1
                         198 James Jams
KeyMap Info              199 Default Program
 Grand Piano               1 Righteous Piano
                           2 Mondo Bass
                           3 Killer Drums
                           4 Weeping Guitar
Octav- Octav+ Panic  Sample Chan-  Chan+
```

The first time you power up (or after a reset), your instrument will be set to operate on MIDI Channel 1 (as shown in the top line of the diagram). After that, it will power up on whatever channel you were on when you powered down. If you'd rather have the K2600 reset certain parameters to default values when you power up, you can do that, too. See *Power Mode* on page 10-9.

Set the volume at a comfortable level. You'll get the best signal-to-noise ratio if you keep the K2600 at full volume, and adjust the level from your mixing board. You may also want to adjust the display contrast. On keyboard models, there's a small knob labeled **LCD Contrast** on the rear panel, between the MIDI ports and the Control Pedal jacks. On the rack-mount model, it's on the left side of the front panel, just above the power switch.

On the K2600R, as you trigger notes from your MIDI controller, you'll see the MIDI LED flash. If it doesn't flash, check your MIDI cables and connections.

Playing the Presets

There are three things you'll want to check out right away: programs, setups and Quick Access banks. In performance situations, you'll be selecting your sounds using one of these three methods. There's a mode (and a mode button) corresponding to each method.

Getting Around

In all three of these modes, the bottom line of the display identifies the function of each of the buttons beneath the display. We call these buttons *soft* buttons, because they do different things depending on what's currently showing in the display.

In Program and Quick Access modes, you can change MIDI channels with the **Chan-** and **Chan+** buttons under the display. In Program mode, you can also change channels using the **Chan/Bank** buttons to the left of the display. (In Quick Access mode, the **Chan/Bank** buttons change Quick Access banks; more about that on page 2-9.)

In all three of these modes, you can transpose an octave at a time by pressing the **Octav** buttons under the display. With the K2600R, this works only if the LocalKbdCh parameter on the RECEIVE page in MIDI mode matches the transmitting channel of your MIDI controller—see Chapter 10.

There are two more soft buttons that appear in these three modes: Panic and Sample.

The **Panic** button sends an All Notes Off message and an All Controllers Off message—both to the K2600 and over all 16 MIDI channels. You won't need it often, but it's nice to have.

The **Sample** button takes you to Sample mode (the SampleMode page), which, as you've probably guessed, is where you make your own samples—if you have the sampling option in your instrument.

Programs

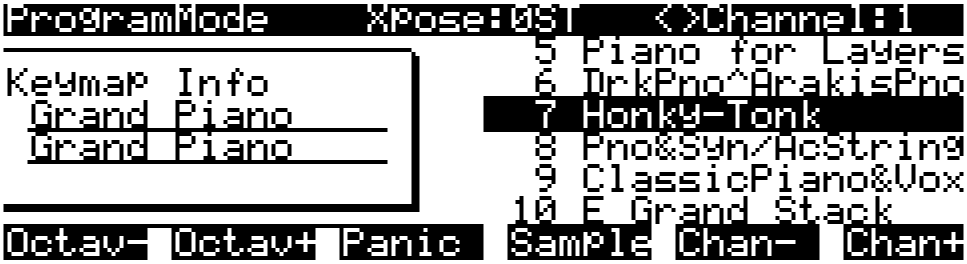
The K2600 powers up in Program mode, where you can select and play programs stored in ***ROM*** or ***RAM***. Programs are preset sounds composed of up to 32 ***layers*** of samples or waveforms. If you've left Program mode, just press the **Program** mode button or **Exit** button to return.

Take a minute to familiarize yourself with the Program-mode display. It gives you some helpful basic information, like the MIDI transposition, what MIDI channel you're on, and which program is currently selected.

There's a box at the left side of the display. The info box, as it's called, displays information about the current program (there's also an info box for Setup mode). The following diagrams shows how the information differs depending on the type of program you've selected.

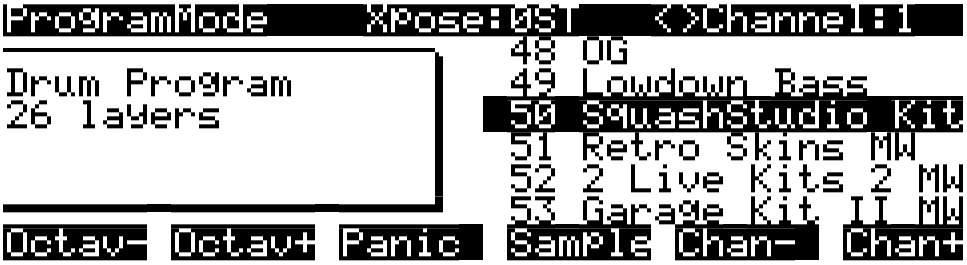
VAST Programs

First is a "normal" VAST program, which is what most of the factory programs are. They have from one to three layers. In this case, there are two. For programs of up to three layers, the info box contains one line per layer, indicating the keymap used in that layer. The line under the keymap name indicates the layer's keyboard range. In this case, both layers extend across the entire keyboard (A 0 to C 8).



```
ProgramMode     Xpose:0ST    <>Channel:1
                             5 Piano for Layers
Keymap Info                  6 DrkPno^ArakisPno
  Grand Piano                7 Honky-Tonk
  Grand Piano                8 Pno&Syn/AcString
                             9 ClassicPiano&Vox
                            10 E Grand Stack
Octav- Octav+ Panic  Sample Chan-  Chan+
```

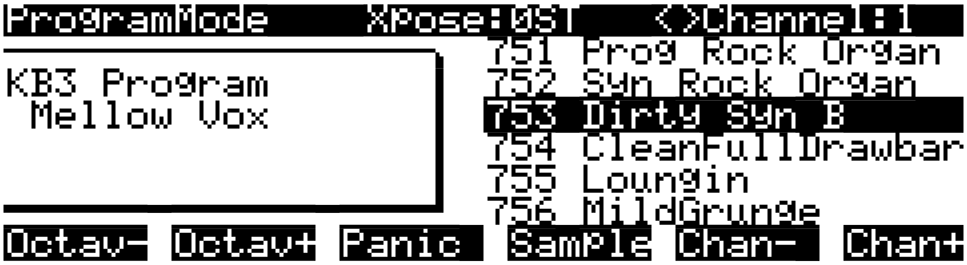
VAST programs with more than three layers are known as *drum* programs, as shown in the following diagram. Drum programs can use any sound you like, but the most common use for programs with more than three layers is to create a number of different sounds across the keyboard—which is perfect for combinations of percussion sounds (hence the name). For drum programs, the info box simply indicates the number of layers in the program—in this case, 26.



```
ProgramMode     Xpose:0ST    <>Channel:1
                            48 OG
Drum Program                49 Lowdown Bass
26 layers                   50 SquashStudio Kit
                            51 Retro Skins MW
                            52 2 Live Kits 2 MW
                            53 Garage Kit II MW
Octav- Octav+ Panic  Sample Chan-  Chan+
```

KB3 Programs

The architecture of KB3 programs is different from that of VAST programs. KB3 programs don’t have layers; they rely on oscillators that mimic the tone wheels used in many popular organs. Consequently, the info box shows only the waveform used in the program.

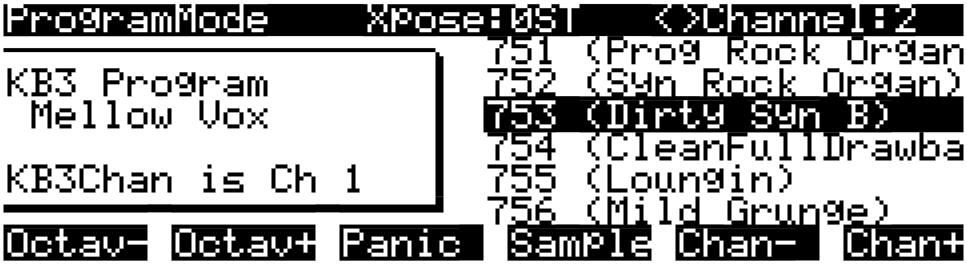


```
ProgramMode     XPose:0ST     <>Channel:1
                       751 Prog Rock Organ
KB3 Program            752 Syn Rock Organ
 Mellow Vox            753 Dirty Syn B
                       754 CleanFullDrawbar
                       755 Loungin
                       756 MildGrunge
Octav- Octav+ Panic  Sample Chan-  Chan+
```

Because of their architecture, KB3 programs require different processing within the K2600, and they don’t work on “regular” channels. They require a special channel to handle the KB3 program’s voices. You can choose any of the 16 MIDI channels to be the KB3 channel, but you can have only one KB3 channel, and KB3 programs play *only* on that channel (VAST programs work just fine on the KB3 channel, by the way).

By default, Channel 1 is designated as the KB3 channel (you can change it in Master mode). If the current channel isn’t the same as the KB3 channel, and you select a KB3 program, you won’t get any sound, because a non-KB3 channel can’t handle a KB3 program.

In the following diagram, Channel 2 is the current channel, but Channel 1 is the KB3 channel. The info box tactfully lets you know what the KB3 channel is, and parentheses appear around the names of all KB3 programs, to further remind you that KB3 programs aren’t available on the current channel. To get Program **753** to work in this case, you’d have to change the current channel back to Channel 1, or go to Master mode and change the KB3 channel to Channel 2.



```
ProgramMode     XPose:0ST     <>Channel:2
                       751 (Prog Rock Organ
KB3 Program            752 (Syn Rock Organ)
 Mellow Vox            753 (Dirty Syn B)
                       754 (CleanFullDrawba
KB3Chan is Ch 1        755 (Loungin)
                       756 (Mild Grunge)
Octav- Octav+ Panic  Sample Chan-  Chan+
```

Live Mode Programs

Programs 740–749 are designed expressly for Live mode, a feature that enables you to take a signal from the Analog sampling input (this requires the sampling options, of course), and route it through the DSP algorithms. Programs 740–749 will not play unless you’re receiving a signal at the Analog sampling input. See page 14-41 for more about Live mode.

Selecting Programs

When you want to change programs, you have several options.

The easiest method is the Alpha Wheel. Turning it left or right will scroll through the program list. You can also change programs using the cursor buttons (the arrow buttons to the right of the display), or the **Plus/Minus** buttons under the Alpha Wheel. To save time, you can enter the program number directly from the alphanumeric buttonpad to the right of the Alpha Wheel, pressing the **Enter** button to complete the selection. If you make a mistake, press **Clear**, then start over.

The K2600 has various settings for responding to MIDI Program Change commands from external sources. These are explained in Chapter 10, so we won’t go into them here. You should be able to change programs by sending Program Change commands from your MIDI controller.

K2600R owners, be sure to try whatever performance controls your MIDI controller has: the Pitch Wheel, Mod Wheel, and switch or control pedals—different programs respond to them in different ways. Attack velocity and aftertouch also have varying effects. Check the list of programs in the *Musician’s Reference*, describing how each program responds to specific MIDI control messages.

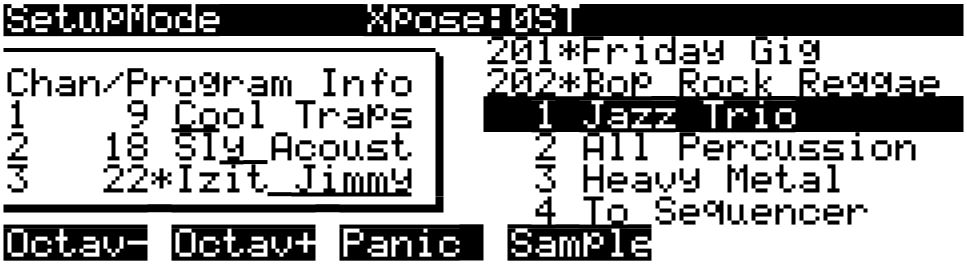
If you don’t hear anything, see the troubleshooting section in the *Musician’s Reference*. When you’re ready to start doing your own programming, check out Chapter 6.

Setups

Setups are preset combinations of programs. Setups can have up to eight zones, each of which can be assigned to any range of the keyboard (overlapping or split). Each zone can have its own program, MIDI channel, and MIDI control assignments.

If you’re a K2600R owner, you can make use of setups even if your MIDI controller can transmit on only one MIDI channel at a time. See *K2600R Owners Only* on page 2-1, or the discussion of the parameter called Local Keyboard Channel in Chapter 10.

Press the **Setup** mode button to the left of the display. Its LED will light, telling you that you’re in Setup mode. Notice that the Setup-mode display is similar to the Program-mode display. If the setup has three or fewer zones, the box at the left shows you the programs assigned to each of the setup’s three zones, and which MIDI channel is used for each program. If the setup is composed of more than three zones, then the box displays a series of horizontal lines illustrating the approximate key ranges of the zones. See page 7-1 for a more detailed description.



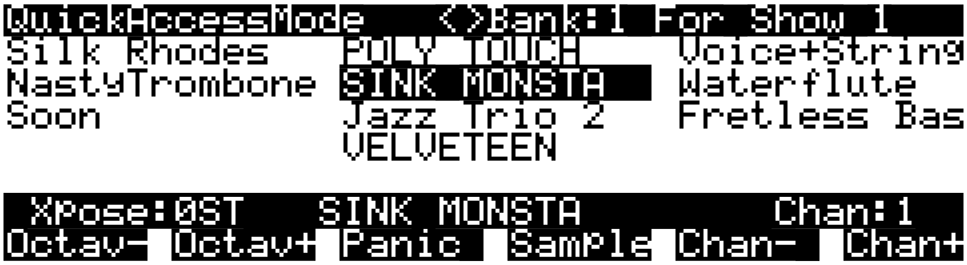
```
SetupMode       Xpose:0ST
                        201*Friday Gig
Chan/Program Info       202*Bop Rock Reggae
1    9 Cool Traps         1 Jazz Trio
2   18 Sly Acoust         2 All Percussion
3   22*Izit Jimmy         3 Heavy Metal
                          4 To Sequencer
Octav-  Octav+  Panic   Sample
```

Many setups include arpeggiation and note-triggered songs to create some pretty amazing grooves that you can use as is, or as templates for your own material. As you play with these setups, experiment with the sliders and other controllers for a wide range of effects. Some of these grooves keep playing after you’ve released the keys that got them going. When you want to stop them, select another setup, or press the **Setup** mode button.

Quick Access

A really convenient way to select programs and setups is to use Quick Access mode, where you select a Quick Access ***bank*** from a list of factory preset or user-programmed banks. Each bank contains ten memory slots, or entries, where you can store any combination of programs or setups. While you're in Quick Access mode, you can select any program or setup in the bank with buttons **0** through **9**.

The K2600 comes with a few Quick Access banks already programmed so you can get an idea of how they work. You'll probably create your own Quick Access banks to help you select programs and setups with a minimum of searching. Press the **Quick Access** mode button to the left of the display. Its LED lights, to tell you you're in Quick Access mode. You'll see a display that looks like this:



```
QuickAccessMode    <>Bank:1 For Show 1
Silk Rhodes    POLY TOUCH    Voice+String
NastyTrombone  SINK MONSTA   Waterflute
Soon           Jazz Trio 2   Fretless Bas
               VELVETEEN

 Xpose:0ST    SINK MONSTA         Chan:1
Octav- Octav+ Panic  Sample Chan-  Chan+
```

The top line of the display tells you which Quick Access bank is selected. Use the **Chan/Bank** buttons (to the left of the display) to scroll through the banks. The names of each of the ten entries in the bank are listed in the center of the display. Many of their names will be abbreviated. The currently selected entry's full name is shown near the bottom of the display. The amount of transposition is displayed to the left of the entry name. If the current entry is a program, you'll see the current keyboard (MIDI) channel displayed to the right of the entry's name. If it's a setup, you'll see the word Setup.

The entries on the Quick Access page are arranged to correspond to the layout of the numeric buttons on the alphanumeric pad. On the page above, for example, the program **SINK MONSTA** is entry 5, and can be selected by pressing **5** on the alphanumeric pad.

When you're ready to create your own Quick Access banks, turn to Chapter 8 to learn about the Quick Access Editor.

The Other Modes

There are seven other modes, five of which have mode buttons on the front panel (Sample mode and Live mode don’t have front-panel buttons). See page 3-2 and Chapter 4 for more detailed descriptions of each mode.

Effects mode Create and modify effects presets, and define how they’re applied to the K2600’s programs and setups.

MIDI mode Configure the K2600 for sending and receiving MIDI information.

Master mode Define performance and control settings.

Song mode Record and edit sequences (songs); play Type 0 and Type 1 MIDI sequences.

Disk mode Load and save programs, setups, samples, and other objects.

Sample mode Edit ROM samples; create and edit RAM (user-defined) samples.

Software Upgrades

Part of the beauty of the K2600 is the ease with which you can upgrade its operating system and objects (programs, setup, etc.) using the boot loader to install upgrades into flash ROM.

At Kurzweil and Young Chang, we have a long history of support for our instruments; the K250, K1000, K2000, and K2500 have been repeatedly enhanced, and these improvements have always been made available to instrument owners in the form of software upgrades.

Upgrading your K2600’s software is simple, painless, and—generally—free! As upgraded software becomes available, you can either get new disks from your Young Chang dealer or download the new stuff from the Web.

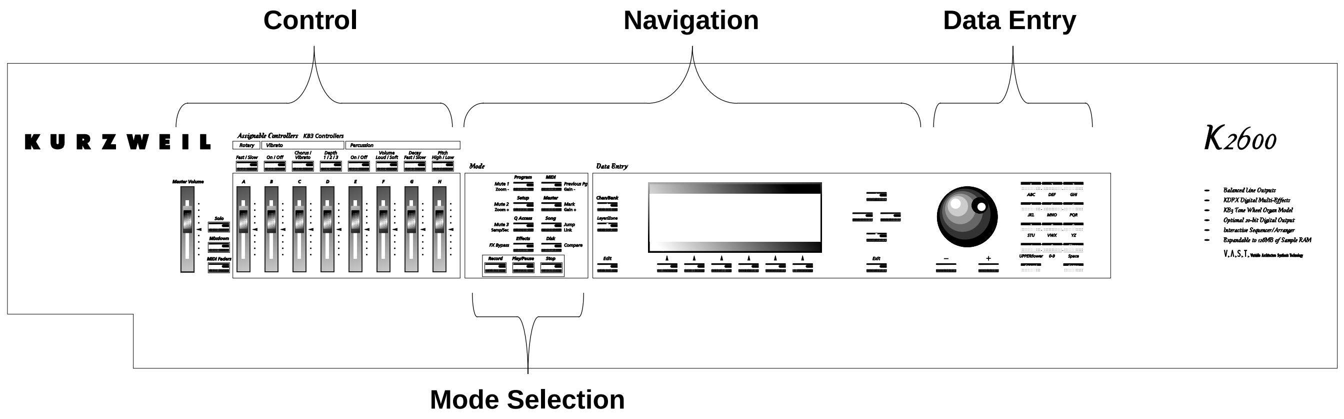
To stay in touch, check out our Web site:

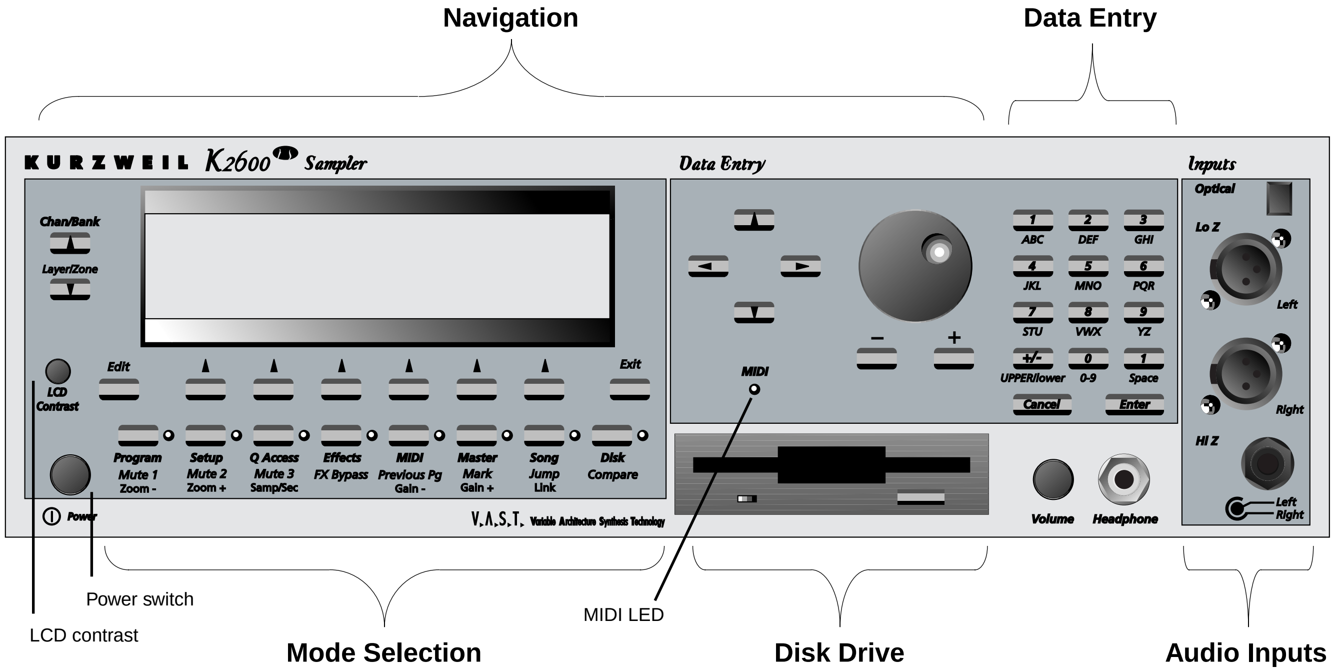
http://www.youngchang.com/kurzweil/

When you’ve acquired an upgrade, you can install it yourself in a matter of minutes. See Appendix A for details.

Chapter 3
User Interface Basics

This chapter will show you how to get around the front panel of your K2600. Your interactions can be divided into three primary operations: mode selection, navigation, and data entry. Keyboard models also have an assignable control section.





Mode Selection

The K2600 is always in one of eight primary operating modes. Select a mode by pressing one of the mode buttons—on keyboard models they're to the left of the display, and on the rack-mount model they're the lower row of buttons under the display. Each mode button has an LED that lights to indicate the current mode. Only one mode can be selected at a time.

Program mode Select and play programs, and modify them with the ***Program Editor***. Rearrange and modify samples in the Keymap and Sample Editors.

Setup mode Select and play setups (eight keyboard zones with independent MIDI channel, program and control assignments), and modify them with the Setup Editor.

Quick Access mode Select from a list of preset banks, each containing a list of ten programs and/or setups that can be viewed in the display for easy selection. Modify the preset banks and create your own with the Quick Access Editor.

Effects mode Define the behavior of the on-board global effects processor. Modify the preset effects and create your own with the Effects Editor.

MIDI mode Define how your K2600 sends and receives MIDI information, and configure each channel to receive independent program, volume, and pan messages that override the normal Program-mode settings.

Master mode Define performance and control characteristics for the entire K2600.

Song mode Use the K2600's sequencer to record and play back your keyboard performance, play Type 0 and Type 1 MIDI sequences, and record multi-timbral sequences received via MIDI.

Disk mode Interface with the K2600's floppy disk drive, an external SCSI device, or an optional internal SCSI disk to load and save programs, setups, samples, and more.

There are two more modes that don't have dedicated buttons on the front panel: Sample mode and Live mode. In Sample mode, you can create and edit samples (if you have the sampling option). To get to Sample mode, press the **Sample** soft button in Program, Setup, Quick Access, or Master modes. See Chapter 14 for details.

Live mode lets you route an input signal through the K2600's DSP algorithms. See page 14-41.

Mode Buttons

The mode buttons are labeled in white. When you press a mode button, its LED lights up to indicate that the mode has been selected. If pressing a mode button does not light its LED, press the **Exit** button one or more times, then try again.

The colored labeling for each mode button indicates special functions that relate to some of the K2600's editors. These functions are described on page 5-7.

Navigation

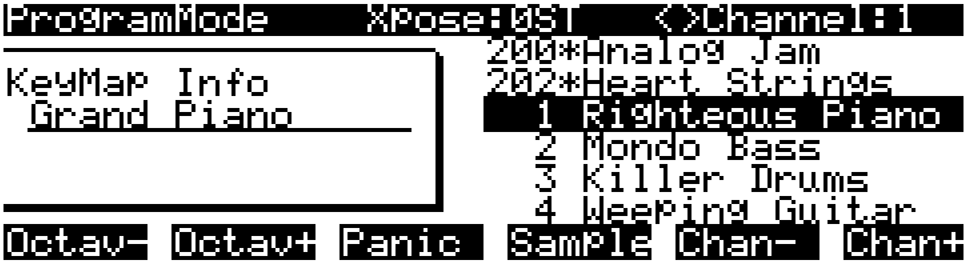
The navigation section of the front panel consists of the display and the buttons surrounding it. These navigation buttons will take you to every one of the K2600's programming parameters.

The Display

Your primary interface with the K2600 is its backlit graphic display. As you press various buttons, this fluorescent display reflects the commands you enter and the editing changes you make. The ample size of the display (240-by-64 ***pixels***) enables you to view lots of information at one time.

Pages

Within each mode, the functions and parameters are organized into smaller, related groups that appear together in the display. Each one of these groups of parameters is called a ***page***. Each mode has what we call an entry-level page; it's the page that appears when you select that mode with one of the mode buttons. Within each mode and its editor(s), the various pages are selected with the navigation buttons. There are many pages, but there are a few features common to each page. The diagram below shows the entry-level page for Program mode.



```
ProgramMode     Xpose:0ST    <>Channel:1
                        200*Analog Jam
KeyMap Info             202*Heart Strings
 Grand Piano              1 Righteous Piano
                          2 Mondo Bass
                          3 Killer Drums
                          4 Weeping Guitar
Octav-  Octav+  Panic  Sample  Chan-  Chan+
```

The Top Line

On the top line of most pages, there's a reminder of which mode you're in and which page you're on. Many pages display additional information in the top line, as well. The Program-mode page above, for example, shows you the current amount of MIDI transposition and the currently selected MIDI channel. The top line is almost always "reversed"—that is, it has a white background with blue characters.

The Bottom Line

The bottom line is divided into six (sometimes fewer) sets of reversed characters that serve as labels for the six buttons directly beneath the display. These labels—and the functions of the buttons—change depending on the currently selected page. Consequently the buttons that select these functions are called "soft" buttons.

The Soft Buttons

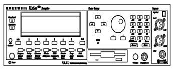
The soft buttons are called "soft" because their functions change depending on the currently selected mode. Sometimes they perform specific functions, like changing MIDI channels in Program mode. In the Program Editor and other editors, they're also used to move to different pages of programming parameters. If a soft button's label is in all capital letters (**KEYMAP**, for example), pressing the corresponding soft button takes you to a page of parameters. If the soft button is labeled in lower-case or mixed-case letters (**Save**, for example), the soft button performs some kind of function.

The Cursor Buttons

To the right of the display are four buttons arranged in a diamond fashion. These are called the cursor buttons. They move the cursor around the currently selected page, in the direction indicated by their labels. The cursor is a highlighted (reversed) rectangle (sometimes it's an underscore). It marks the value of the currently selected parameter.

Programming the K2600 involves selecting various parameters and changing their values. Select parameters by highlighting their values with the cursor. You can change the highlighted value with any of the data entry methods described in the data entry section below.

The MIDI LED



Below the cursor buttons is a red LED labeled MIDI. This LED flashes whenever the K2600 receives MIDI information from your MIDI controller.

The Chan/bank Buttons

To the left of the display are two buttons labeled **Chan/Bank**. Their function is related to the two small arrows—◄►—that appear in the top line of the display when there are multiples of the current page—for example, the LAYER pages in the Program Editor. When you see these arrows, you can use the **Chan/Bank** buttons to scroll the values of the parameter that appears to the right of the arrows. In Program mode, for example, they shift through the MIDI channels, showing the program assigned to each channel.

When you're in the Program Editor, the **Chan/Bank** buttons let you view each layer in the program. You can see the corresponding parameters in each layer by scrolling through the layers with these buttons. In the Setup Editor, the **Chan/Bank** buttons scroll through the zones in the current setup. In Quick Access mode, they scroll through the Quick Access banks, and in Song mode they scroll through recording tracks.

We'll let you know, when applicable, what the **Chan/Bank** buttons do.

The Edit Button

The **Edit** button activates each of the K2600's editors, and acts as a shortcut to many pages within the Program Editor. Pressing the **Edit** button tells the K2600 that you want to change some aspect of the object marked by the cursor. For example, when a program is selected and you press **Edit**, you enter the Program Editor. If a setup is selected, you enter the Setup Editor.

There are editors accessible from just about every operating mode. To enter an editor, choose one of the modes (mode selection), and press **Edit**. An editing page for that mode will appear. You can then select parameters (navigation) and change their values (data entry). If the value of the selected parameter has its own editing page, pressing the **Edit** button will take you to that page. For example, in the Program Editor, on the PITCH page, you might see **LFO1** assigned as the value for Pitch Control Source 1. If you select this parameter (the cursor will highlight its value—**LFO1** in this case), then press the **Edit** button, you'll jump to the page where you can edit the parameters of LFO1. Naturally, you can find every page in the current editor by using the soft buttons, but often it's easier to use the **Edit** button shortcut.

The Exit Button

Press **Exit** to leave the current editor. If you’ve changed the value of any parameter while in that editor, the K2600 will ask you whether you want to save your changes before you can leave the editor. See page 5-3 for information on saving and naming. The **Exit** button also takes you to Program mode if you’re on the entry level page of one of the other modes. If at some point you can’t seem to get where you want to go, press **Exit** one or more times to return to Program mode, then try again.

Data Entry

The data entry section of the front panel includes the Alpha wheel, the **Plus/Minu**s buttons, and the 14-button alphanumeric pad.

The Alpha Wheel

The Alpha Wheel is especially useful because it can quickly enter large or small changes in value. If you turn the Alpha Wheel one click to the right, you’ll increase the value of the currently selected parameter by one increment. One click to the left decreases the value by one increment. If you turn it rapidly, you’ll jump by several increments. You can also use the Alpha Wheel to enter names when you’re saving objects.

The Plus/Minus Buttons

These buttons are located just under the Alpha Wheel. The **Plus** button increases the value of the currently selected parameter by one, and the **Minus** button decreases it by one. These buttons are most useful when you’re scrolling through a short list of values, or when you want to be sure you’re changing the value by one increment at a time. One press of the **Plus** or **Minus** button corresponds to one click to the right or left with the Alpha Wheel. These buttons will repeat if pressed and held.

Pressing the **Plus** and **Minus** buttons simultaneously will move you through the current list of values in large chunks instead of one by one. Often this is in even increments (10, 100, etc.). Don’t confuse these buttons with the **+/-** button on the alphanumeric pad. This button is used primarily for entering negative numeric values and switching from uppercase to lowercase letters (and vice versa).

The Alphanumeric Pad

As its name implies, this set of 14 buttons lets you enter numeric values, and to enter names one character at a time. Depending on where you are, the K2600 automatically enters letters or numerals as appropriate (you don’t have to select between alphabetic or numeric entry).

When you’re entering numeric values, press the corresponding numeric buttons, ignoring decimal places if any (to enter 1.16, for example, press **1**, **1**, **6**, **Enter**). The display will reflect your entries, but the value won’t actually change until you press **Enter**. Before pressing **Enter**, you can return to the original value by pressing **Cancel**. Pressing **Clear** is the same as pressing **0** without pressing **Enter**.

When entering names, you can use the **Left/Right** cursor buttons or the <<< / >>> soft buttons to move the cursor to the character you want to change. Use the labels under the alphanumeric buttons as a guide to character entry. Press the corresponding button one or more times to insert the desired character above the cursor. The **Cancel** button is equivalent to the >>> soft button, and **Enter** is the same as **OK**. The **Clear** button replaces the currently selected character with a space. The **+/-** button ***toggles*** between uppercase and lowercase letters.

There's also a convenient feature called keyboard naming, which lets you use the keyboard to enter characters in names. See page 5-5.

Double Button Presses

Pressing two or more related buttons simultaneously executes a number of special functions depending on the currently selected mode. Make sure to press them at exactly the same time.

In this mode or editor…	…pressing these buttons simultaneously…	…does this:
Program mode	Octav-, Octav+	Reset MIDI transposition to 0 semitones. Double-press again to go to previous transposition.
	Chan–, Chan+	Set current MIDI channel to 1.
	Plus/Minus	Step to next Program bank (100, 200, etc.)
Master mode	Chan/Bank	Enables Guitar/Wind Controller mode.
Song mode	Left/Right cursor buttons	Toggle between Play and Stop.
	Up/Down cursor buttons	Toggle between Play and Pause.
	Chan/Bank	Select all tracks on any TRACK page in Song Editor.
Disk mode	2 leftmost soft buttons	Issue SCSI Eject command to currently selected SCSI device.
	Chan/Bank	Hard format SCSI device. List selected objects when saving objects.
	Left/Right cursor buttons	Select all items in a list. Move cursor to end of name in naming dialog.
	up/down cursor buttons	Clear all selections in a list. Move cursor to beginning of name in naming dialog.
Program Editor	Chan/Bank	Select Layer 1.
Keymap Editor	Plus/Minus	With cursor on the Coarse Tune parameter, toggles between default Coarse Tune of sample root and transposition of sample root.
Sample Editor	2 leftmost soft buttons	Toggle between default zoom setting and current zoom setting.
	Plus/Minus buttons	Set the value of the currently selected parameter at the next *zero crossing.*
Any Editor	Plus/Minus	Scroll through the currently selected parameter's list of values in regular or logical increments (varies with each parameter).
	2 leftmost soft buttons	Reset MIDI transposition to 0 semitones. Double-press again to go to previous transposition.
	Center soft buttons	Select Utilities menu (MIDIScope, Stealer, etc.).
	2 rightmost soft buttons	Sends all notes/controllers off message on all 16 channels (same as Panic soft button).
	Left/Right cursor buttons	Toggle between Play and Stop of current song.
	Up/Down cursor buttons	Toggle between Play and Pause of current song.
Save Dialog	Plus/Minus buttons	Toggle between next free ID and original ID.

Table 3-1 Double Button Presses

Intuitive Data Entry

Many parameters have values that correspond to the standard physical controls present on most MIDI controllers (primarily keyboards). In many cases, you can select these values “intuitively,” rather than having to scroll through the ***Control Source*** list. This is done by selecting the desired parameter, then holding the ENTER button while moving the desired physical control.

For example, on the LAYER page in the Program Editor, you can set the range of the currently selected layer as follows: use the cursor buttons to move the cursor to the value for the LoKey parameter, press (and hold) the **Enter** button, then use your MIDI controller to trigger the note you wish to be the lowest note for the currently displayed layer. The note you triggered (it has to be between C 0—C 8) will appear as the value for the LoKey parameter. Repeat the process for the HiKey parameter.

Another example: select Program 199 while in Program mode. Press **Edit** to enter the Program Editor. Press the PITCH soft button to select the PITCH page. Move the cursor to the Src1 parameter. Hold the **Enter** button, and move your controller’s Pitch Wheel. **PWheel** will be selected as the value for Src1.

You can also use the keyboard of your MIDI controller to choose control sources, since most key numbers correspond to a value on the control source list. If you have a certain control source that you use over and over (for example, LFO1), this can be the quickest way to enter its value. To do this: highlight a parameter which uses a value from the control source list, hold down **Enter**, then strike the key corresponding to the control source you want to choose. LFO1, for example, is assigned to B5. You’ll find a complete chart of these keyboard shortcuts at the end of Chapter 4 of the *Musician’s Reference.*

Also, for almost every parameter, holding the **Enter** button links the K2600 to your MIDI controller’s Data slider, if it has one. Moving the Data Slider will run through the range of values for the currently selected parameter. This is not as precise as the Alpha Wheel, but much faster.

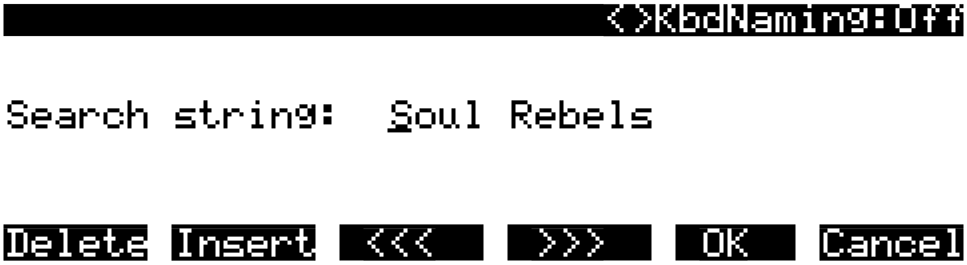
Changing the Current Layer in Multi-Layer Programs

When editing a multi-layer program (including drum programs), you can quickly switch between layers by holding the **Enter** button, then striking a key. The K2600 will change the current layer to that key’s layer. If the key is part of more than one layer, subsequent key strikes will cycle through each layer that has that key in its range.

Note: *This method for changing the current layer in a multi-layer program will NOT work if the currently highlighted parameter has a note number or control source for its value. In this case, the key you strike will function as described in Intuitive Data Entry, above.*

Search

There's a convenient way to find any string of characters within the currently selected list, or range of values. Hold the **Enter** button and press any of the numeric buttons. A ***dialog*** like the one below appears.



<>KbdNaming:Off

Search string: Soul Rebels

Type in the string of characters you want to find. For example, if you're looking at the program list and you want to find all programs containing the word "Horn," you would type h-o-r-n. This function is not case-sensitive; it will find upper and lower case characters regardless of what you type.

When you've typed the string of characters you want to find, press **Enter**. The K2600 searches through the current list of objects or values, finds all items that match the string of characters you typed, and displays the first one it finds. Hold **Enter** and press one of the **Plus/Minus** buttons to search for the next higher- or lower-numbered object that contains the string of characters.

The string you select remains in memory. You can store and select a string of characters with each of the numeric buttons. Hold **Enter** and press one of the numeric buttons at any time to select that string for a search. When the string appears, you can change it, or just press **Enter** to find that string.

Renaming Multiple Objects

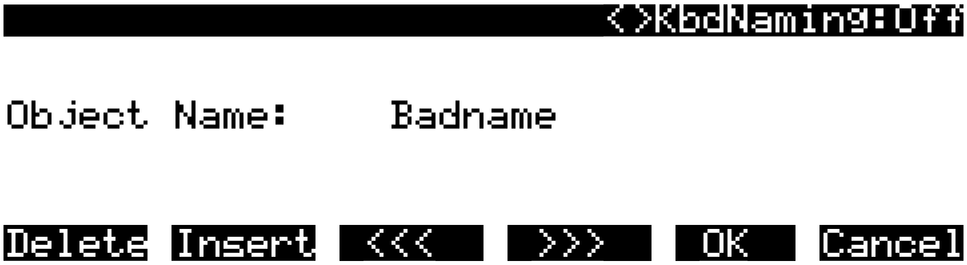
You can automatically rename multiple RAM objects with one operation, by substituting one string of characters with another.

1. Press the **Master** mode button to enter Master mode.

2. Press the **Object** soft button, then press the **Name** soft button. The NAME page appears, showing a list of all RAM objects.

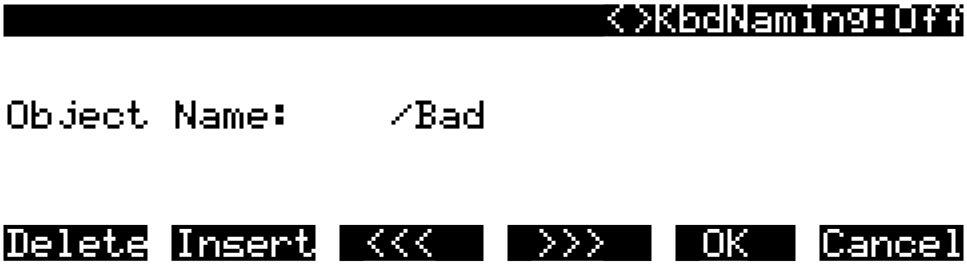
3. Use the Alpha Wheel or **Plus/Minus** buttons to highlight objects. When an object whose name you want to change is highlighted, press the **Select** soft button to select the object. Note the asterisk that appears between the object's ID and its name.

4. When you've selected all the objects you want to rename, press **OK**. The following dialog appears. The name you see is the name of the last object you selected.


<>KbdNaming:Off

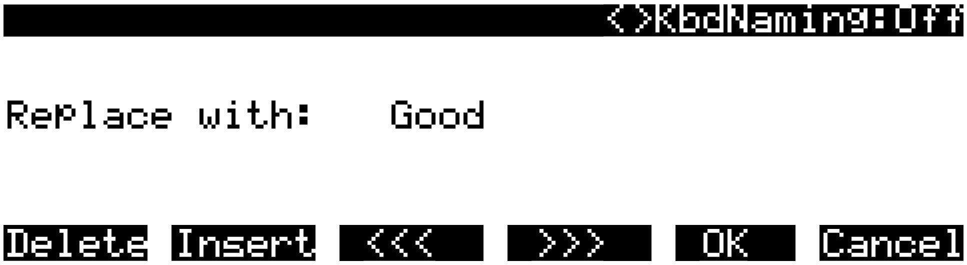
Object Name: Badname

5. Use your favorite method to enter a slash (/), followed by the string of characters that you want to replace. This function is case-sensitive, so you can enter both capitals and lower-case characters. In this example, we're going to change "Bad" to "Good."


<>KbdNaming:Off

Object Name: /Bad

6. When you've entered the desired string of characters, press **OK**, and the dialog changes to prompt you to enter the replacement string. Notice that we entered only the characters **B-a-d**, and not the entire name of the object. Enter the new string, as shown below.


<>KbdNaming:Off

Replace with: Good

7. Press **OK**. The K2600 asks you if you're sure (unless you have confirmations turned off). Press **Yes**. The NAME page reappears. Every RAM object whose name contained **Bad** now contains **Good** instead.

Panel Play Feature (Rack Models)

You can play notes on the K2600R even if you don't have a MIDI controller connected to its MIDI In port. Using the alphanumeric pad, you can play the octave from C 4 to C 5 and send the notes to the audio outputs and Headphones jack.

While in any mode, press and hold the **Cancel** button on the alphanumeric pad. Press one of the other alphanumeric buttons to play various notes: **1** is C 4, **2** is C# 4, and so on up to **Enter**, which plays C 5. Notes are sent to the K2600's sound engine on the current MIDI channel.

Pressing the **Up** or **Down** cursor buttons while holding **Cancel** will transpose up or down an octave while using this feature. This transposition will be remembered the next time you use this feature, but does not affect the K2600's normal MIDI transposition.

Notes are played at approximately the mf velocity level. You can adjust the attack velocity of the notes using the **Plus/Minus** buttons while holding down the **Cancel** button. The velocity values will continue to change if you hold the **Plus/Minus** buttons down.

You can press the **Right** cursor button while holding **Cancel** to sustain notes. The **Left** cursor button releases notes.

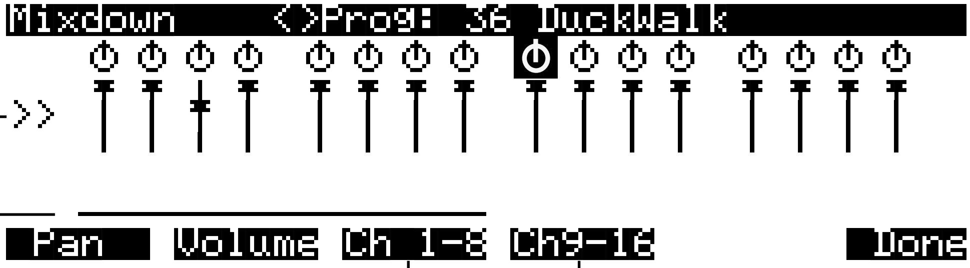
Mixdown and MIDI Faders Pages

On keyboard models, there are two buttons below the **Solo** button, labeled **Mixdown** and **MIDI Faders**. These call up pages that let you use the sliders to control MIDI Pan and Volume, or any MIDI Controller that you choose. This is useful for real-time control over setups.

K2600R owners can use front-panel buttons for the same purpose. On any page (except in Quick Access mode), press **Clear** to view the Mixdown page, then use the cursor buttons to select a pan or volume control, and the Alpha Wheel or **Plus/Minus** buttons to change values. Press **Clear** again to view the MIDI Faders page, where you can change values similarly. Press **Clear** a third time (or press **Exit**) to return to normal operation.

Mixdown Page

This page temporarily turns the K2600's 8 sliders into MIDI pan and volume controls. You can get to this page from any mode by pressing the **Mixdown** button, located below the **Solo** button. (K2600R owners can press the **Clear** button one or more times to call up these pages.) Press **Mixdown**, and the display looks similar to what you see on the Song-mode MIX page. However, there are a few differences. On the Mixdown page, the sliders relate to MIDI channels instead of tracks. If you highlight 1-8 for example, moving slider D will affect volume or pan for MIDI Channel 4 while you're on this page.



>> shows whether sliders control pan or volume

Shows which channels are affected by sliders.

Soft buttons for indicating which channels are affected by sliders.

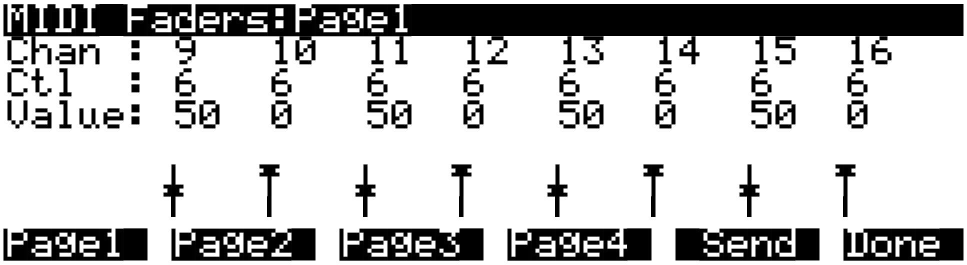
On the Mixdown page, pan and volume messages are sent to the internal K2600 sounds, to MIDI Out, or both. All Mixdown-page channels have the same destination (local, MIDI, or both), which is determined by the Control parameter on the MIDI-mode TRANSMIT page. You can assign the sliders to control either volume or pan. The soft buttons at the bottom of the Mixdown page let you select pan or volume control, on channels 1-8 or 9-16, allowing you to temporarily use the 8 sliders (or the navigation and data entry buttons on the K2600R) as pan or volume faders. Keep in mind that MIDI volume is a channel-specific message. Therefore, if you have more than one zone in a setup assigned to the same channel, moving any one of the associated sliders will affect all those zones.

MIDI Faders Pages

The button just below **Mixdown** is called **MIDI Faders** (on rack models, press the **Clear** button one or more times to view this page. The MIDI Faders page allows you to create a preset configuration of any Controller assignment for each slider, along with an initial preset value for that Controller. For each slider, you assign the MIDI channel, MIDI Controller number, and a value. You can enter the value using any normal data entry method, or by moving the corresponding slider on keyboard models. You can also assign several sliders to the same channel, but assign each one to a different Controller number.

There are four different configurations (pages 1–4), which you select with the soft buttons. Each configuration remembers the last value you set for each slider. If you want to send those values without moving the slider, press the **Send** button.

The available values for the Channel parameter on this page are **1–16** (local and MIDI), **1L–16L** (local only, no MIDI), and **1M–16M** (MIDI only, no local).



```
MIDI Faders:Page1
Chan : 9    10   11   12   13   14   15   16
Ctl  : 6    6    6    6    6    6    6    6
Value: 50   0    50   0    50   0    50   0

Page1   Page2   Page3   Page4    Send   Done
```

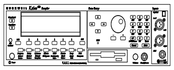
Quick Song Recording and Playback

On keyboard models, there are three buttons—labeled **Record**, **Play/Pause**, and **Stop**—below the mode selection buttons. They control the recording and playback of songs from any mode; you don’t have to be in Song mode to record or play back.

Using these buttons affects the current track of the current song—that is, the song and track that were selected the last time you were in Song mode. When you record, the recording track and recording mode are determined by the current settings in Song mode; likewise for the playback mode when you’re playing a song.

When the sequencer status is STOPPED (neither the **Record**-button LED nor the **Play/Pause**-button LED is lit or flashing), press **Record** to put the sequencer in REC READY status. The **Record**-button LED lights (red). Then press **Play/Pause** to start recording. The **Play/Pause**-button flashes (green) to indicate the tempo. Any countoff is determined by the current Song-mode setting for the CountOff parameter. Press **Play/Pause** or **Stop** to end recording and go to the Save dialog, where you can save the song, or trash it.

When the sequencer status is STOPPED, press **Play/Pause** to begin playing the current song. Press **Play/Pause** again to pause playback, and again to resume. Press **Stop** to end playback.



With the K2600R, you can’t record in this fashion (use Song mode for recording), but you can simultaneously press the **Up/Down** or **Left/Right** cursor buttons to start and stop playback of the current song.

Chapter 4
The Operating Modes

In this chapter we'll discuss the theory behind the mode concept, and we'll describe the basic operating features of each mode.

What the Modes Are

The modes exist to make the K2600 logical to work with. With as many performance and programming features as the K2600 has, it's helpful to break them into groups. These groups are called modes. There are eight primary modes (plus Sample mode and Live mode); they're described briefly in the section called *Using the Modes* on page 4-3. Chapters 6 through 13 are dedicated to explaining each primary mode in turn. Chapter 14 describes Sample mode and Live mode.

Each mode is named for the kind of operations you perform while in that mode, and each mode's editor (if any) contains all of the parameters related to editing the type of ***object*** found in that mode. In Setup mode, for example, you select setups (and only setups) for performance or editing. All of the setup-editing parameters are grouped together on the Setup-Editor page, which is accessible through Setup mode.

Selecting Modes

When the K2600 is on, it's almost always operating in one of the eight primary modes represented by the LED-highlighted buttons beneath the display—or in one of the editors corresponding to the current operating mode. There are two exceptions: Sample mode (and the Sample Editor), and Live mode. Pressing one of the mode buttons selects the corresponding mode. This is the mode's entry level. At the entry level, the LED of the selected mode is lit. Only one mode can be selected at a time.

From any primary mode, you can get to any other primary mode simply by pressing one of the mode buttons. If you're in an editor, however, you must press **Exit** to return to the mode's entry level before selecting another mode.

All of the modes except Disk mode and Live mode give you access to one or more editors for changing the values of the parameters within that mode. Press the **Edit** button to enter the editor of the currently selected mode. When you do this, the mode LED goes out.

It's possible to enter another mode's editor without leaving the currently selected mode. For example, if you press **Edit** while in Setup mode, you'll enter the Setup Editor. The Setup-Editor page will appear, and the LocalPrg parameter will be highlighted by the cursor. If you press **Edit** again, you'll enter the Program Editor, where you can edit the currently selected program. While you can edit and save programs as you normally would, you're still in Setup mode, and you can't select another mode at this point. When you exit the Program Editor, you'll return to the Setup-Editor page. Press **Exit** again, and you'll leave the Setup Editor, returning to the Setup-mode page.

This method of entering editors works for any parameter whose value is an editable object (program, setup, keymap, FX preset, etc.)

The following table lists the procedures for moving between modes and editors. Note that the **Exit** button won't always take you where the table says it will; it often depends on how you got where you are. The table assumes that you've entered a given editor via its corresponding mode. You'll always return to Program mode eventually if you press **Exit** repeatedly.

Current Mode/ Editor Status	Available Modes/ Editors	How to Get There
Any mode	All other modes	Press corresponding mode button
Program mode	Program Editor	Press **Edit**
	Sample mode	Press **Sample** soft button
Program Editor	Program mode	Press **Exit**
	Studio Editor	On KDFX page: select Studio parameter; press **Edit**
	Keymap Editor	On KEYMAP page: select Keymap parameter, press **Edit**
Keymap Editor	Program Editor	Press **Exit**
	Sample mode	Press the **MIDI** mode button
	Sample Editor	Select Sample parameter; press **Edit**
Sample Editor	Keymap Editor	Press **Exit**
Setup mode	Sample mode	Press **Sample** soft button
	Setup Editor	Press **Edit**
Setup Editor	Setup mode	Press **Exit**
	Program Editor	On CH/PRG page: select LocalPrg parameter; press **Edit**
	Studio Editor	On KDFX page: select Studio parameter; press **Edit**
	Song Editor	On COMMON page, select Song parameter; press **Edit**
Quick Access mode	Sample mode	Press **Sample** soft button
	Quick Access Editor	Press **Edit**
Quick Access Editor	Quick Access mode	Press **Exit**
	Program Editor	Set value of Type parameter to **Program**; select program; press **Edit**
	Setup Editor	Set value of Type parameter to **Setup**; select setup; press **Edit**
Effects mode	Studio Editor	Press **Edit**
Studio Editor	Previous mode	Press **Exit**
	FX Preset Editor	Select FX preset block; press **Edit**
FX Preset Editor	Studio Editor	Press **Exit**
MIDI mode	Program Editor	On CHANNELS page: select Program parameter; press **Edit**
	Setup Editor	On TRANSMIT page: select CtlSetup parameter; press **Edit**
	Velocity Map Editor	On TRANSMIT or RECEIVE page: select Veloc(ity)Map parameter; press **Edit**
	Pressure Map Editor	On TRANSMIT or RECEIVE page: select Press(ure)Map parameter; press **Edit**
Master mode	Sample mode	Press **Sample** soft button
	Velocity Map Editor	Select VelTouch parameter; press **Edit**
	Pressure Map Editor	Select PressTouch parameter; press **Edit**
	Intonation Table Editor	Select Intonation parameter; press **Edit**
Song mode	Song Editor	Select CurSong parameter; press **Edit**
	Program Editor	Select Program parameter; press **Edit**
Sample mode	Sample Editor	Select Sample parameter; press **Edit**
	Previous mode	Press **Exit**
Most editors	Previous mode or editor	Press **Exit**

Nested Editors

Starting at the Program-mode level, there are three “nested” editors, each related to the parameters that make up different components of a program. The first is the Program Editor, which you enter when you press **Edit** while in Program mode. Programs consist, among other things, of ***keymaps***; they determine which samples play on which keys. Keymaps can be edited as well. The Keymap Editor is entered from within the Program Editor, by selecting the KEYMAP page with the soft buttons, then pressing **Edit**.

Similarly, keymaps consist of ***samples***, which also can be edited. The Sample Editor is entered from the Keymap Editor, by selecting the Sample parameter and pressing **Edit**. When you enter the Sample Editor, you’ve worked through three nested levels of editors, all related to the components that make up a program. And in fact, you’re still in Program mode (if that’s where you started from). Pressing **Exit** while in the Sample Editor will return you to the Keymap Editor. Pressing **Exit** again will return you to the KEYMAP page of the Program Editor. Once more, and you’re back to Program mode’s entry level.

*Note: You can also get to the Sample Editor directly from Sample mode (by pressing **Edit**). If you’re editing a single sample, this might make more sense, since it’s more direct. If you’re planning on editing a number of samples within a given keymap, however, you’ll find it more convenient to enter the Sample Editor through the Program and Keymap Editors, because once you’ve finished with one sample, you can easily return to the KEYMAP page and select another sample within the keymap.*

Finding Square One

If, at any time, you don’t know where you are, and the mode LEDs are all unlit, press **Exit** one or more times. This will return you to the entry level of whatever mode you were in, and if you press **Exit** enough times, you will always return to Program mode, the startup mode. If you’ve made any changes, you’ll be asked whether you want to save before leaving any editor. Press the **No** soft button or the **Exit** button if you don’t want to save. If you want to save, press the **Rename** or **Yes** soft button, and you’ll see the Save dialog, which is described in *Saving and Naming* on page 5-3.

Using the Modes

You can play your K2600 regardless of the mode you’re in. In fact, the only times you can’t play it are when you’re in the middle of a disk operation (loading, saving, formatting) or a SMDI sample transfer. With these two exceptions, the K2600’s MIDI response is almost always active. Even so there are three modes that are more performance-oriented than the others. These are Program, Setup, and Quick Access modes. We’ll describe each of the eight modes briefly in this section.

Program Mode

The K2600 starts up in Program mode, where you can select, play, and edit programs. The Program-mode entry-level page shows the currently selected program, as well as a small segment of the program list. Program mode gives you access to Sample mode via the **Sample** soft button.

The Program, Keymap, and Sample Editors are nested within Program mode. They take you to the core of the K2600’s sound editing parameters. We’ll discuss them in Chapters 6 and 14.

Setup Mode

Setup mode lets you select, play, and edit setups. Setups consist of up to eight separate zones, split or overlapping, each having its own program, MIDI channel, and control parameters. Setups are great for performance situations, whether you're playing multiple K2600 programs or controlling additional synths connected to the K2600's MIDI Out port. Chapter 7 describes Setup mode and the Setup Editor in detail.

If you're using a MIDI controller, you can make use of Setup mode even if your MIDI controller can transmit on only one MIDI channel at a time. To do this, go to the RECEIVE page in MIDI mode (by pressing the **RECV** soft button while in MIDI mode), and set the Local Keyboard Channel parameter to a value that matches the transmit channel of your MIDI controller. When you select Setup mode, the K2600 will interpret incoming MIDI information according to the settings for the currently selected setup. See the discussion of the Local Keyboard Channel parameter in Chapter 10 for details.

Setup mode also gives you access to Sample mode via the **Sample** soft button.

Quick Access Mode

Another feature for live performance, Quick Access mode enables you to combine programs and setups into banks of ten entries. Each of these programs or setups can be selected with a single alphanumeric button. Different banks are selected with the **Chan/Bank** buttons. There's a selection of factory preset banks, and you can use the Quick Access Editor to create your own banks and store them in RAM. There's a full description in Chapter 8.

Quick Access mode gives you access to Sample mode via the **Sample** soft button.

You can use Quick Access banks as a way to remap incoming or outgoing Program Change commands. See *Program Change Formats* on page 10-11.

Effects Mode

Effects mode sets the behavior of the KDFX effects processor. The Effects-mode page lets you tell the K2600 how to select preset studios (programmed effects configurations) when you change programs or setups, or lets you choose a preset studio that's applied to *every* K2600 program. The Studio Editor allows you to tweak the preset studios, and create your own. Chapters 9 and 15 show you how. You can also listen to the sounds of various effects while in Effects mode, without selecting different programs.

MIDI Mode

You'll use MIDI mode to configure the K2600's interaction with other MIDI instruments, by setting parameters for transmitting and receiving MIDI. You'll also use it to configure your K2600 for multi-timbral sequencing. On the CHANNELS page, you can assign a program to each channel, and enable or disable each channel's response to three types of MIDI control messages: Program Change, volume and pan. You can also override program output assignments, and adjust overall program gain. See Chapter 10.

Master Mode

Master mode, described in Chapter 11, contains the parameters that control the entire K2600. Global settings for tuning, transposition, velocity and aftertouch sensitivity, and other preferences are adjusted here. You can also get to the SampleMode page from here, using the **Sample** soft button.

Song Mode

Song mode enables you to play sequences (songs) stored in the K2600's RAM, and provides a fully featured sequencer that you can use to record songs. You can also record multi-timbrally via MIDI, or load standard MIDI files (Type 0 or 1). The Song Editor enables you to modify existing RAM sequences, do step recording, and create arrangements (by linking two or more songs together). See Chapter 12.

Disk Mode

Finally, Disk mode is used to load and save programs and other objects using the K2600's internal floppy disk drive, an optional internal SCSI disk, or an external SCSI disk (or CD-ROM drive) connected to either of the K2600's SCSI ports. Chapter 13 has the details.

Chapter 5
Editing Conventions

Introduction to Editing

Programming (editing) the K2600 always involves three basic operations: mode selection, navigation, and data entry.

First, select the mode that relates to the object you want to edit—a program, a setup, etc. Then select the object you want to edit, and press the **Edit** button to enter the editor within that mode. An editor contains all the parameters that define the object you're programming.

Next, you navigate around the editor's page(s) with the soft buttons, and select parameters with the cursor (arrow) buttons. When you've selected a parameter (its value is highlighted by the cursor), you can change its value with one of the data entry methods. When you change a value, you'll normally hear its effect on the object you're editing. The K2600 doesn't actually write your editing changes to memory until you save the object you're working on. It then allows you to choose between writing over the original object, or storing the newly edited version in a new memory location.

What's an Object?

If you've been wondering what we mean by the term "object," it's an expression we use for anything that can be named, saved, deleted, or edited. Here's a list of all the types of objects:

Samples	Digital recordings of instrumental sounds or waveforms. Samples actually have two separate parts: the actual sample data and the sample header information, which contains start, alternative start, loop, and end points, as well as other information like tuning and volume.
Keymaps	Collections of samples assigned to specific velocity ranges and/or keys.
Programs	Factory-preset or user-programmed sounds stored in ROM or RAM. A program is one or more layers of sound, with programmable DSP functions applied to the keymaps within each layer.
Setups	Factory-preset or user-programmed MIDI performance presets consisting of up to eight zones, each with its own program, MIDI channel, and controller assignments, and (optionally) arpeggiation specifications.
Songs	Sequence files loaded into RAM, or MIDI data recorded in Song mode.
Studios	Factory-preset or user-programmed configurations of the K2600's onboard digital audio effects processor.
FX presets	Factory-preset or user-programmed data—including effects algorithms, levels and balance—that define the nature of studios.
Quick Access banks	Factory-preset or user-programmed banks of ten entries each, that store programs and setups for single-button access in Quick Access mode.
Velocity maps	Factory-preset or user-programmed curves that affect the K2600's response to, and MIDI transmission of, attack velocity values.

Pressure maps Factory preset or user-programmed curves that affect the K2600’s response to, and MIDI transmission of, pressure (aftertouch) values.

Intonation tables Factory preset or user-programmed tables that affect the intervals between the twelve notes of each octave.

Master tables The values that are set for the global control parameters on the Master-mode page, as well as the settings for the parameters on the CHANNELS page in MIDI mode, and the programs currently assigned to each MIDI channel.

Fader tables MIDI Controller assignments defined on the MIDI Faders pages.

Name tables Contains a list of dependent objects needed by the other objects in a file at the time the file was saved.

Macros List of disk files to be loaded into the K2600’s memory at start-up time.

Object Type and ID

The K2600 stores its objects in RAM using a system of ID numbers that are generally organized into banks of 100. Each object is identified by its object type and object ID; these make it unique. An object’s type is simply the kind of object it is, whether it’s a program, setup, song, or whatever. The object ID is a number from 1 to 999 that distinguishes each object from other objects of the same type. For example, within the 200s bank, you can have a setup, a program, and a preset effect, all with ID 201; their object types distinguish them. You can’t, however, have two *programs* with ID 201.

Object Type	Object ID	Object Name
Program	201	Hot Keys
Setup	404	Silicon Bebop
Velocity Map	1	Linear
Sample	3	Hey Moe

ROM (factory preset) objects have ID numbers in the Zeros bank (1–199) and the 700s bank. If you’ve bought the Contemporary and Orchestral ROM block options, you’ll also have ROM programs in the 800s and 900s bank. When you save objects that you’ve edited, the K2600 will ask you to assign an ID. If the original object was a ROM object, the K2600 will suggest the first available ID, starting with 200. If the original object was a RAM object, you’ll have the option of saving to an unused ID, or replacing the original object.

Objects of different types can have the same ID, but objects of the same type must have different IDs to be kept separate. When you’re saving an object that you’ve edited, you *can* assign the same ID to an existing object of the same type, but if you do, the new one will be written over the old one. For example, if you assign an ID of 1 to a program you’ve edited, the K2600 will ask you if you want to “replace” the ROM program currently stored with that ID. We’ll discuss this further in *Saving and Naming* on page 5-3.

Many parameters have objects as their values—the VelTouch parameter on the Master-mode page, for example. In this case, the object’s ID appears in the value field along with the object’s name. You can enter objects as values by entering their IDs with the alphanumeric pad. This is especially convenient for programs, since their ID numbers are the same as their MIDI program change numbers (when you're using the Extended or Kurzweil Program Change format—see *Program Change Formats* on page 10-11).

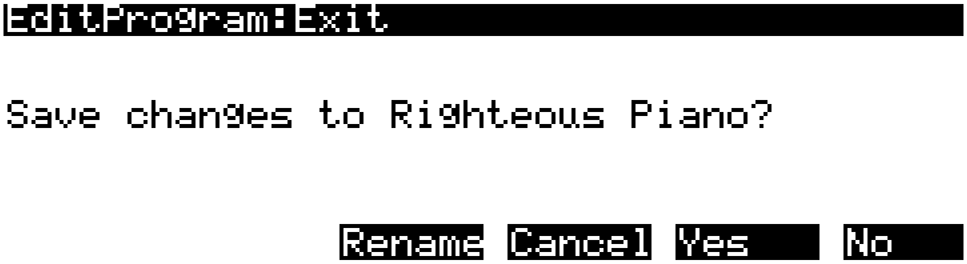
The object type and ID enable you to store hundreds of objects without losing track of them, and also to load files from disk without having to replace files you've already loaded. See *Memory Banks* on page 5-6 for more information on object type and ID.

Saving and Naming

When you've edited an object to your satisfaction, you'll want to store it in RAM. There's a standard procedure for saving and naming, which applies to all objects.

You can press the **Save** soft button, of course, but it's easier to press the **Exit** button, which means "I want to leave the current editor." If you haven't actually changed anything while in the editor, you'll simply exit to the mode you started from. If you *have* made changes, however, the K2600 will ask you if you want to save those changes. This is the first Save dialog. A dialog is any display that asks a question that you need to answer before the K2600 can proceed.

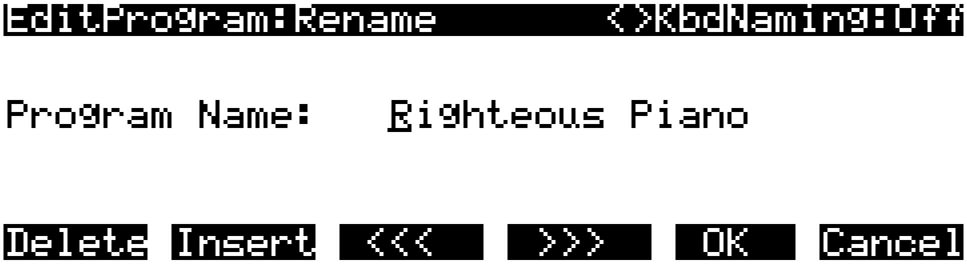
The following diagram shows the Save dialog for the Program Editor. The top line tells you that you got to this dialog by pressing Exit. If you had pressed the Save soft button instead of Exit, you'd see Save in the top line, instead of Exit. If you were in another editor, the top line would indicate that as well.



EditProgram:Exit

Save changes to Righteous Piano?

The best way through this process is to press the **Rename** soft button. This takes you immediately to the naming dialog, where you assign a name to the object you're saving. You haven't saved yet, but you'll be able to after you've named the program.



EditProgram:Rename <>KbdNaming:Off

Program Name: Righteous Piano

Like the Save dialog, the top line identifies the current editor and function, and also indicates whether the keyboard naming feature is enabled (see *Keyboard Naming* on page 5-5).

The cursor underlines the currently selected character. Press the <<< or >>> soft buttons to move the cursor without changing characters. Press an alphanumeric button one or more times to enter a character above the cursor. The characters that correspond to the alphanumeric buttons are labeled under each button. If the character that appears is not the one you want, press the button again. Press the **+/-** button on the alphanumeric pad to switch between upper and lower case characters.

Press **0** one or more times to enter the numerals 0 through 9. Press **Clear** (on the alphanumeric pad) to erase the selected character without moving any other characters. Press the **Delete** soft button to erase the selected character. All characters to the right of the cursor will move one space left. Press the **Insert** soft button to insert a space above the cursor, moving all characters to the right of the cursor one space to the right.

Press the **Cancel** soft button if you decide not to name the object. Press **OK** when the name is set the way you want to save it.

In addition to the letters and numerals, there are three sets of punctuation characters. The easiest way to get to them is to press one of the alphanumeric buttons to select a character close to the one you want, then scroll to it with the Alpha Wheel. Here’s the whole list:

! “ # $ % & ’ ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9

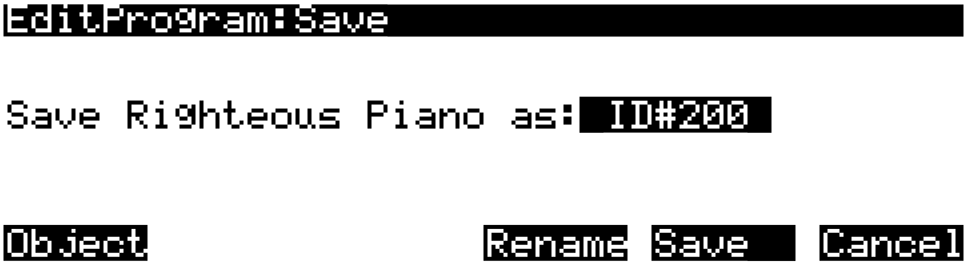
: ; < = > ? @ A through Z

[\ ] ^ _ ` a through z. (space).

Pressing the **Plus/Minus** buttons simultaneously is a short cut to the following characters:
0, **A**, **a** and (space).

If you’re wondering how we came up with this sequence of characters, it’s composed of ASCII characters 33 through 122.

When you press **OK**, the final Save dialog appears, where you assign an ID to the edited object. If you change your mind about the name, press the **Rename** soft button for another try.



EditProgram:Save

Save Righteous Piano as: ID#200

ROM Objects

If the object you started from was a ROM (factory preset) object, the K2600 will automatically suggest the next available (unused) ID as the ID for the edited object. If that’s the ID you want, press the **Save** soft button, and the object will be stored in RAM with that ID. Otherwise, you can select any ID from 1 to 999. This page also gives you the opportunity to return to the naming dialog (as described in the previous section) or, by pressing the **Object** soft button, to access the Object Utilities (described in Chapter 13).

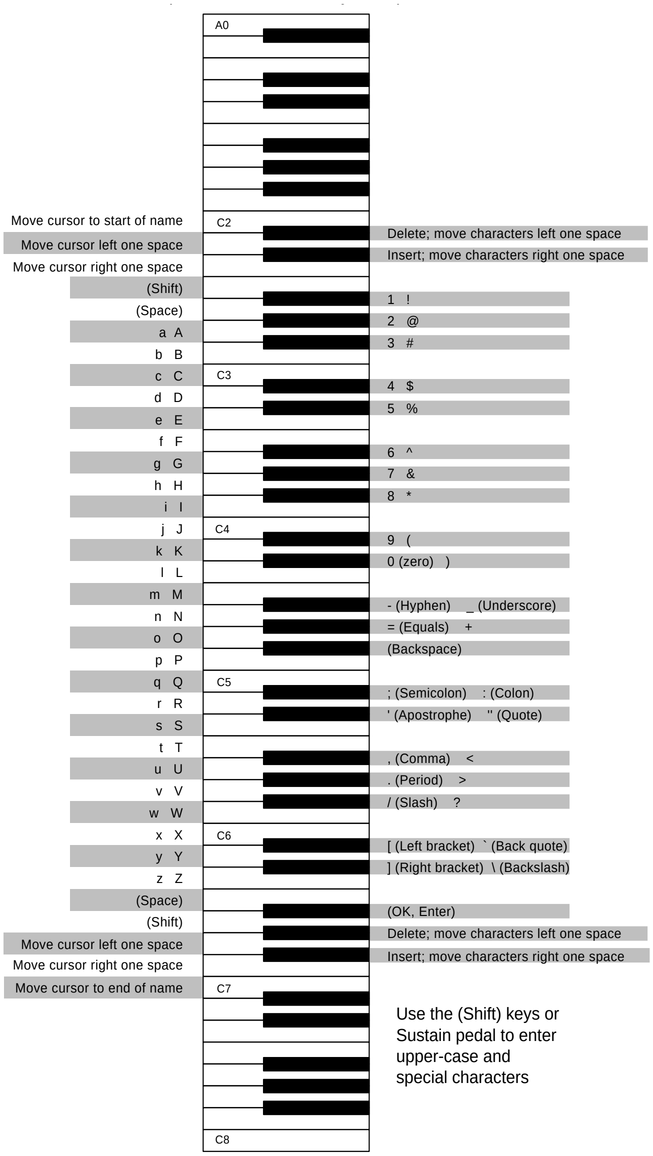
If you select an ID that’s already in use, the K2600 will tell you that you’re going to replace the ROM object that’s already been assigned that ID. If you don’t want to do that, you can select a different ID. Or you can press the **Plus/Minus** buttons simultaneously to toggle between the ID that the K2600 suggested and the original ID. Or press the **Cancel** soft button to cancel the operation.

If you decide not to cancel or change the ID, and you press the **Replace** soft button, the K2600 will write your newly edited object over the existing ROM object. Actually, it only appears that way, since you can't truly write to ROM. The ROM object will reappear if you delete the newly edited object (there are soft buttons in each editor for deleting objects).

RAM Objects

If the original object was a RAM object, the K2600 will assume you want to replace it, and will suggest the same ID as the original object (if it has an asterisk—*—between its ID and its name, it's a RAM object). As with ROM objects, you can cancel, replace, or change the ID and save to an unused ID. If you replace a RAM object, however, it's definitely gone!

Keyboard Naming

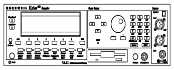

A0
Move cursor to start of name
C2
Delete; move characters left one space
Move cursor left one space
Insert; move characters right one space
Move cursor right one space
(Shift)
1 !
(Space)
2 @
a A
3 #
b B
c C
C3
4 $
d D
5 %
e E
f F
6 ^
g G
7 &
h H
8 *
i I
j J
C4
9 (
k K
0 (zero))
l L
m M
- (Hyphen) _ (Underscore)
n N
= (Equals) +
o O
(Backspace)
p P
q Q
C5
; (Semicolon) : (Colon)
r R
' (Apostrophe) " (Quote)
s S
t T
, (Comma) <
u U
. (Period) >
v V
/ (Slash) ?
w W
x X
C6
[(Left bracket) ` (Back quote)
y Y
] (Right bracket) \ (Backslash)
z Z
(Space)
(OK, Enter)
(Shift)
Delete; move characters left one space
Move cursor left one space
Insert; move characters right one space
Move cursor right one space
Move cursor to end of name
C7
Use the (Shift) keys or Sustain pedal to enter upper-case and special characters
C8

The keyboard naming feature makes naming objects convenient (and musical!). With keyboard naming enabled, you can use the keyboard (or your MIDI controller) to enter the name of the object you're modifying.

There are three keyboard naming states: **Off** (disabled), **On**, and **Adv** (Advance). When set to **On** or **Adv**, the keys (MIDI note numbers, actually) correspond to all the characters shown on page 5-4. There are also equivalents to the cursor movement, insertion, deletion, and **OK** (**Enter**) buttons.

On requires you to move the cursor to enter each letter, just as you have to do when using the normal data entry methods for naming. **Adv** automatically moves the cursor one space to the right each time you strike a key, just like a typewriter or computer keyboard. This is the most convenient setting.

When you're in the Rename dialog, use either of the **Chan/Bank** buttons to change the keyboard naming state.



Keyboard naming works on the K2600R as well, as long as the LocalKbdCh parameter on the RECEIVE page in MIDI mode is set to the same channel as the one your MIDI controller is transmitting on.

Deleting Objects

Within most editors, there are soft buttons for deleting objects. When you want to delete an object, press the **Delete** soft button, and the K2600 will ask you if you want to delete the object. (At this point in the dialog, you can select another object with any of the data entry methods.) Press **OK** if you want to delete it, or press **Cancel** if you don't. Although it seems that you can delete ROM objects, you can't actually do it. The K2600 will behave as if it's deleting the ROM object, but it will still be there the next time you select it. (What actually happens is that the ROM object is copied to RAM as soon as you press **Edit**, and when you "delete" the ROM object, you're actually deleting the RAM copy. The original ROM object remains in memory.)

RAM objects, on the other hand, are *gone* when you delete them! If you've "replaced" a ROM object by saving a RAM object with the same ID, the ROM object is invisible, but still there. Deleting the RAM object stored at the same ID will restore the ROM object.

You'll often delete objects to gain RAM space, or to organize the ***memory banks*** before saving objects to disk. To delete multiple objects, use the Delete Objects utility available in Master mode. It's described on page 11-17.

Dependent Objects

A dependent object is an object that's linked in memory with at least one other object. For example, if you create a setup that uses a program that you also created, that program is a dependent object of the setup.

When you start to delete an object that has dependent objects, the Delete dialog gives you a choice: Delete dependent objects? If you press **Yes**, the K2600 will delete the object and all its dependent objects when you execute the Delete function. In our example, if you were deleting the setup you created, and you chose to delete dependent objects, the dependent program would get deleted as well. If you press **No** at the Delete dependent objects? prompt, the K2600 deletes only the object, but keeps the dependent objects. In our example, the setup would get deleted, but the dependent program would remain.

When deleting objects and their dependents, the K2600 deletes only those dependent objects that aren't dependent on other objects that you're *not* deleting. For example, suppose you have two setups that contain the same program. If you delete one of the setups, and delete dependent objects with it, the setup gets deleted, but the program that's contained in the other setup remains in memory.

Memory Banks

To help organize the storage of your edited objects, the K2600's memory is divided into ten banks, each of which stores objects having IDs within a certain range. Objects within the same range of IDs are stored in the same memory bank, regardless of their types. The banks are in increments of 100, that is, objects with IDs from 1 through 99 are stored in the first bank, IDs from 100 to 199 in the second bank, and so on. We refer to them simply as the "Zeros bank," "100s bank," "200s bank," etc. If you save an object as ID 203, for example, it's in the 200s bank.

You can store up to 100 objects of each type in each memory bank, The number of objects of a given type that can be saved in a memory bank depends on its type. For example, you can store 20 Quick Access banks in each memory bank. As you begin to save objects that you've edited, you'll notice that the IDs suggested by the K2600 sometimes increase in large chunks—from 219 to 300, for example. This is due to the limit to the number of objects of a given type that can be stored in a single memory bank. This limit can be important in terms of organizing your objects for storing to disk. Check out the section called *Storing Objects in the Memory Banks* on page 13-36 for lists of how many objects of each type can fit into each memory bank.

You'll want to think about organizing your edited objects in the memory banks when you start using the floppy disk drive or a SCSI device to store your programs, samples, and other objects. Objects that are stored in the same memory banks can be easily stored in the same ***file*** on disk—which generally is a good idea. You can also store all the banks to one file by selecting "Everything" in the Disk-mode Bank dialog.

The memory banks work automatically, that is, you don't have to select the different banks to gain access to the objects stored in them. The K2600 selects the appropriate bank when you enter the object ID you want to work with. To select Program 201 while in Program mode, for example, just press **2**, **0**, **1**, **Enter** on the alphanumeric pad. The 200s bank is automatically selected, and the program list will show programs numbered in the 200s. If your MIDI controller can send program change commands from 0 to 127 or 1 to 128 only, you'll probably want to adjust the way the K2600 responds to program change commands. See the discussion of the Program Change Type parameter (ProgChgType) in Chapter 10.

When you do a save operation in Disk mode, you're creating a file to be saved to disk. This will save either individually selected objects or an entire bank of objects from the K2600's RAM. If you choose to save a complete bank, then all objects with IDs in the range of the selected bank, regardless of type, are saved as part of the file. For example, if you save the 200s bank (objects with IDs from 200–299), then every object with an ID from 200 to 299 will be saved to the file.

This system makes it easy for you to keep track of everything you save. The first program you save, for example, will have an ID of 200 (unless you specify another ID). The first setup you create will also have an ID of 200 (since they're different *types* of objects, the IDs can be the same). If you were to save the 200s bank, both your program and your setup would be saved to the same file.

Saving and Loading Files—Disk Mode

Saving to disk simply involves selecting objects or a complete bank of objects to be stored as a single file. All objects with IDs within that range will be saved to the file. When you *load* a file, the K2600 asks you which bank will receive the file. You can load a file into any of the ten banks, regardless of the bank it was saved from. The K2600 will automatically reassign the object IDs. A file saved from the 200s bank, for example will be stored on disk with its objects numbered from 200–299. If you load it back into the 300s bank, its objects will be renumbered from 300–399.

See Chapter 13 for more information on loading and saving files.

Special Button Functions

The Mode buttons and the **Chan/Bank Down** button have additional functions, depending on the mode or editor you're in. When you're in the Program or Setup Editor, they function according to the blue labeling under each button. They also work as track mutes on the MIX page of Song mode.

When you're in the Sample Editor, the **Program**, **Setup**, **Q Access**, **MIDI**, **Master**, and **Song** mode buttons function according to the orange labeling near each button. Table 5-1 describes all of the special button functions.

Button	Mode or Editor			
White **Blue** **Orange**	**Program Editor (Blue)**	**Setup Editor (Blue)**	**Song Mode**	**Sample Editor (Orange)**
Program Mute 1 Zoom-	Mutes Layer 1 of current program, or mutes current layer of current drum program	Mutes Zone 1 of current setup if 3 or fewer zones; mutes current zone of current setup if more than 3 zones	On MIX page, mutes Track 1 or 9	On TRIM and LOOP pages, decreases horizontal dimension of current sample in display
Setup Mute 2 Zoom+	Mutes Layer 2 of current program, or solos current layer of current drum program	Mutes Zone 2 of current setup if 3 or fewer zones; solos current zone of current setup if more than 3 zones	On MIX page, mutes Track 2 or 10	On TRIM and LOOP pages, increases horizontal dimension of current sample in display
Q Access Mute 3 Samp / Sec	Mutes Layer 3 of current program, or solos current layer of current drum program	Mutes Zone 3 of current setup if 3 or fewer zones; solos current zone of current setup if more than 3 zones	On MIX page, mutes Track 3 or 11	Toggles between units used to identify location within sample— either number of samples from start, or time in seconds from start
Effects FX Bypass	Bypasses (mutes) current program's FX preset (plays program dry)	Bypasses (mutes) current setup's studio (plays studio dry)	On MIX page, mutes Track 4 or 12	
MIDI Previous Pg Gain -	Successive presses take you back to four most recent editor pages; 5th press takes you to ALG page	Successive presses take you back to four most recent editor pages; 5th press takes you to CH/PRG page	On MIX page, mutes Track 5 or 13	On TRIM and LOOP pages, decreases vertical dimension of current sample in display
Master Mark Gain +	"Remembers" current editor page, so you can recall multiple pages with Jump button; asterisk appears before page name to indicate that it's marked; unmark pages by pressing Mark when page is visible	Same as for Program Editor; pages common to both editors are marked or unmarked for *both* editors	On MIX page, mutes Track 6 or 14	On TRIM and LOOP pages, increases vertical dimension of current sample in display
Song Jump Link	Jumps to marked pages in order they were marked	Jumps to marked pages in order they were marked	On MIX page, mutes Track 7 or 15	Preserves interval between Start, Alt, Loop, and End points of current sample; press again to unlink
Disk Compare	Negates effect of unsaved edits and plays last-saved (unedited) version of object being edited	Same as for Program mode; display reminds you that you're comparing; press any button to return to edited version	On MIX page, mutes Track 8 or 16	
Chan / Bank Layer / Zone	In Program Editor, these two buttons scroll through layers of current program; in Effects Editor, scroll through FX presets; in Keymap Editor, scroll through velocity levels of current keymap; in Setup Editor, scroll through zones of current setup; in Quick Access mode, scroll through entries in current Quick Access bank		Change recording track	
Edit	Whenever cursor is highlighting an editable object or parameter, takes you to corresponding editor or programming page			

Table 5-1 Special Button Functions

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.

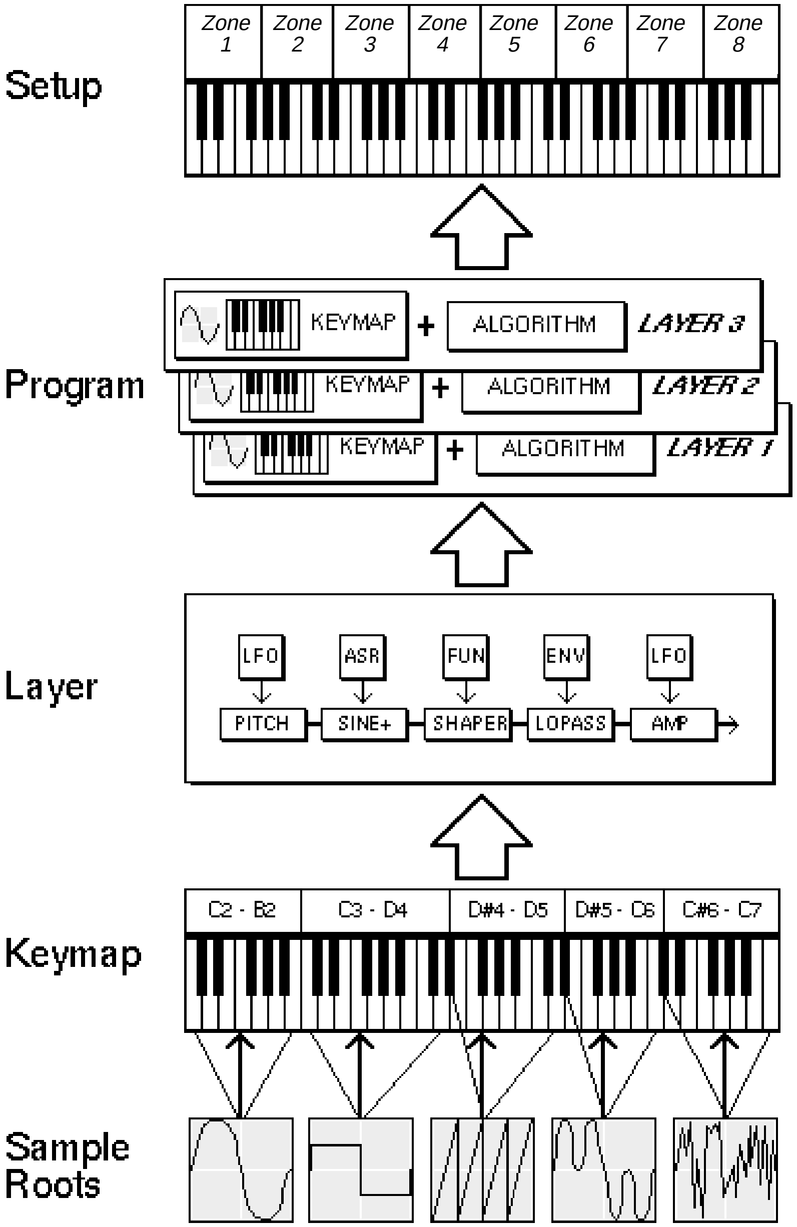
VAST Program Structure

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.



Eight keyboard zones—each with independent program, MIDI channel, and control assignments

Selected for performanc and editing in Program mode; up to 32 layers pe program

A keymap processed through an algorithm, modulated by control sources

Up to 128 sample roots, assigned to play at programmable key and velocity ranges

Individual digital sound recordings stored in ROM or RAM; stereo samples use two voices of polyphony

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.

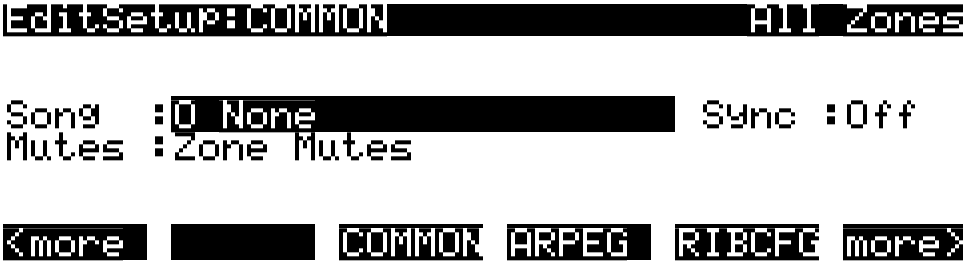
Note: *If you're using a Kurzweil PC88 to control your K2600, you shouldn't use Channel 1 as the KB3 channel. The PC88 sends MIDI Controller 90 on Channel 1 to select effects. In KB3 mode, the K2600 maps Controller 90 to internal controller 90, which controls the emulation of leakage that we mentioned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



```
EditSetup:COMMON                        All Zones

Song  :0 None                      Sync :Off
Mutes :Zone Mutes

<more          COMMON  ARPEG   RIBCFG  more>
```

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:

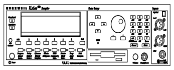
- Certain MIDI Controller numbers always control specific KB3 features
- The value of the LocalKbdCh parameter affects how KB3 programs respond to MIDI Controller messages

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:

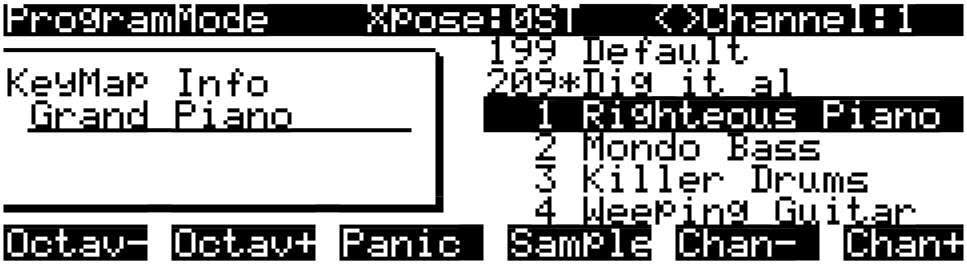
- Edit an existing setup, adjusting some of the physical controller assignments (see page 20-11).
- Use the KB3 setup that we've provided for your convenience. It's in a file on one of the accessory disks that came with your K2600 (see page 20-12).

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



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

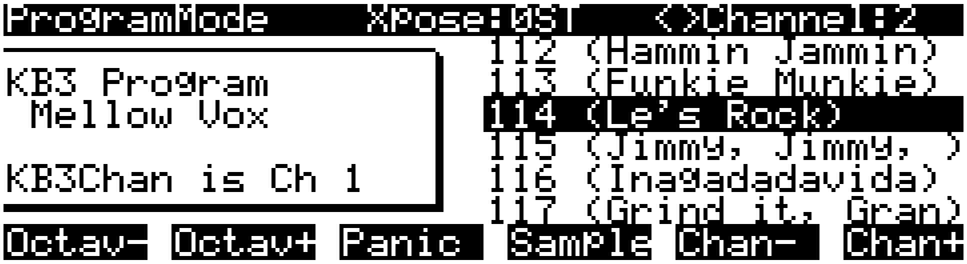
The top line of the Program-mode entry-level page shows your location, the present MIDI transposition, and the current MIDI channel.

The 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.



```
ProgramMode     Xpose:0ST    <>Channel:2
                          112 (Hammin Jammin)
KB3 Program               113 (Funkie Munkie)
 Mellow Vox               114 (Le's Rock)
                          115 (Jimmy, Jimmy, )
KB3Chan is Ch 1           116 (Inagadadavida)
                          117 (Grind it, Gran)
Octav-  Octav+  Panic   Sample  Chan-   Chan+
```

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.

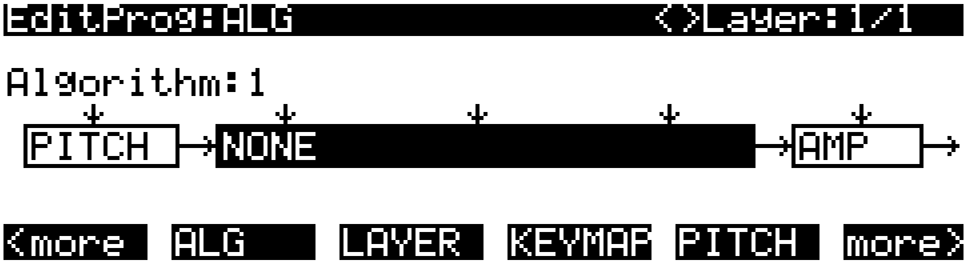
Use the **Chan–** and **Chan+** soft buttons to change the current MIDI channel. This changes the MIDI channel the K2600 uses internally, as well as the channel you're using to send information to other synths connected to the K2600's MIDI Out port (MIDI slaves). Changing the current MIDI channel with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



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

Algorithm:1
```

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

Here's a method for jumping quickly to a specific layer in a program—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.

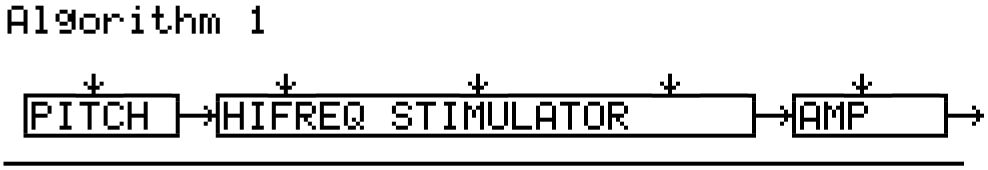
There are more pages and functions in the Program Editor than there are soft buttons. Therefore, two of the soft buttons are dedicated to scrolling through the list of pages and functions. If you don’t see the button for the page or function you want to select, press one of the soft buttons labeled **<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.

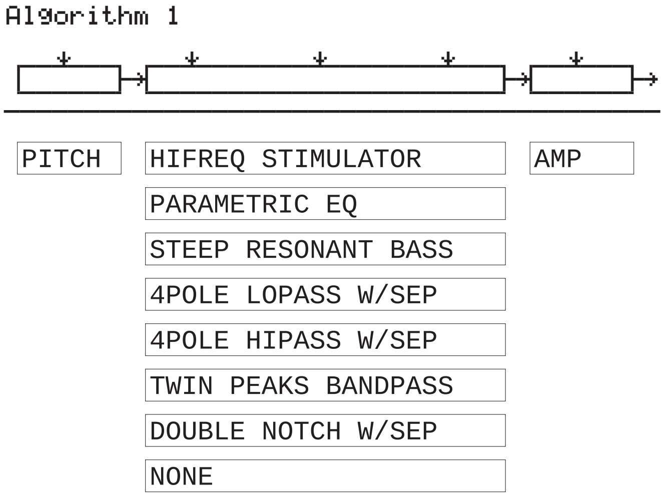


Figure 6-2 Algorithm Wiring

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.

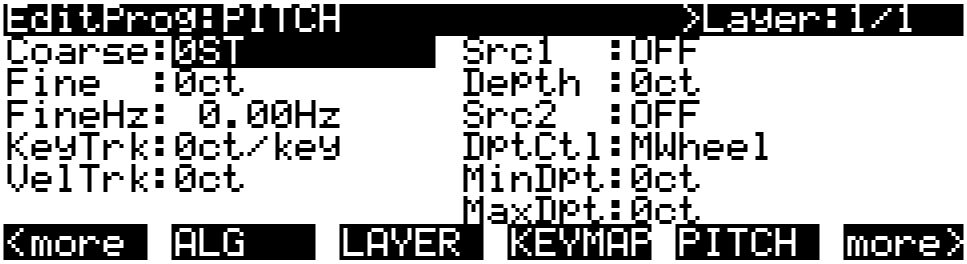
- Key tracking (KeyTrk)
- Velocity tracking (VelTrk)

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.



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

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

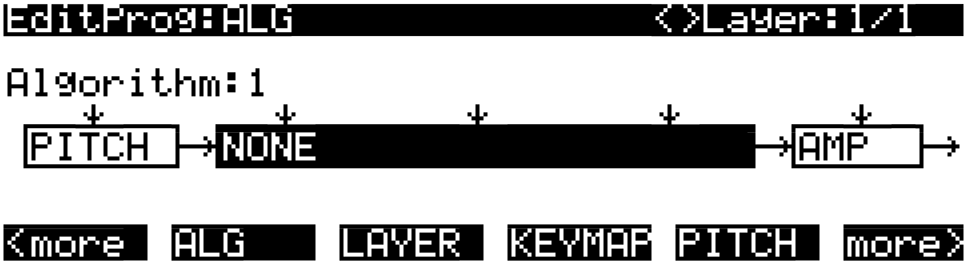
These six control source parameters are just a few of the control sources available throughout the Program Editor. We've given them special attention because they appear on *all* the pages relating to the DSP functions, not just on the PITCH page.

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.



EditProg:ALG <>Layer:1/1

Algorithm:1

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.

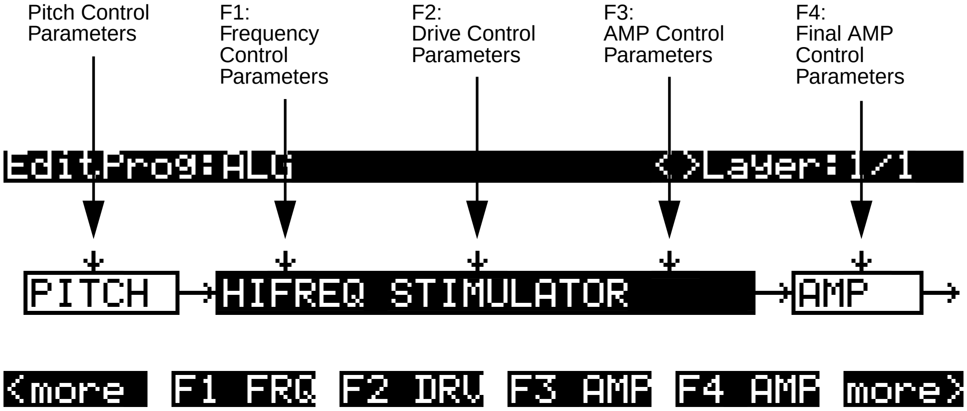
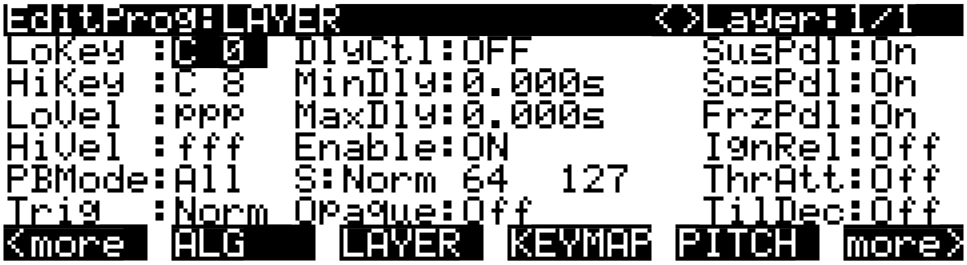

Pitch Control Parameters
F1: Frequency Control Parameters
F2: Drive Control Parameters
F3: AMP Control Parameters
F4: Final AMP Control Parameters
EditProg:ALG
<>Layer:1/1
PITCH
HIFREQ STIMULATOR
AMP
<more
F1 FRQ
F2 DRV
F3 AMP
F4 AMP
more>

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.



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

Parameter	Range of Values	Default
Low Key	C -1 to G 9	C 0
High Key	C -1 to G 9	C 8
Low Velocity	ppp to fff	ppp
High Velocity	ppp to fff	fff
Pitch Bend Mode	Off, Key, All	All
Trig	Normal, Reversed	Normal
Delay Control	Control Source list	Off
Minimum Delay	0 to 25 seconds	0
Maximum Delay	0 to 25 seconds	0
Layer Enable	Control Source list	On
Enable Sense 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)

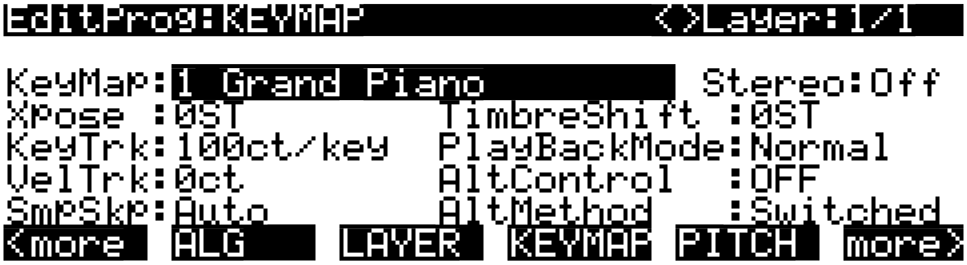
When on, this parameter causes all notes in the layer to sustain through the entire first attack segment of their amplitude envelopes, even if the notes have been released. If you have a sound with a slow attack, or an attack that's delayed with the delay control, setting this parameter to On will make 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.



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

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

Parameter	Range of Values	Default
Keymap	Keymap List	1 Grand Piano
Transpose	± 60 Semitones	0
Key Tracking	± 2400 Cents per key	100
Velocity Tracking	± 7200 Cents	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.

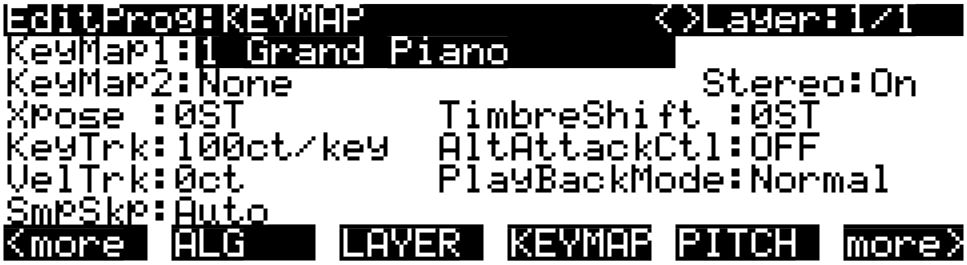
A value of **On** means that sample skipping will be employed throughout the range of the keymap. This eliminates the pitchbend limitation described above, but may add some artifacts to the sound. Creative types may appreciate this form of distortion, however, so we've made it available.

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:



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

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.

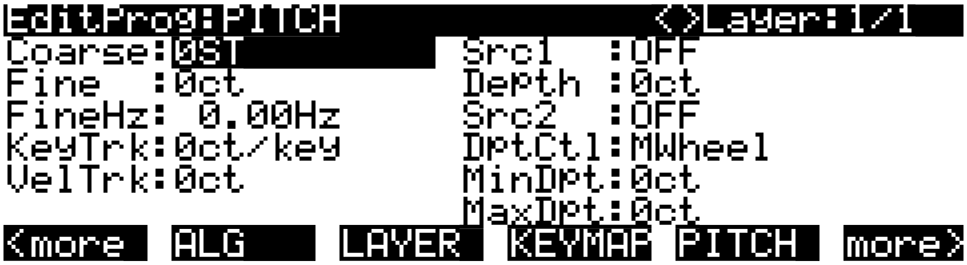
Emulating Legato Play

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.



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

Parameter	Range of Values	Default
Coarse Adjust	–120 to 60 ST (Semitones)	0
Fine Adjust (Cts)	± 100 Cents	0
Fine Adjust (Hz)	± 6 Hz	0
Key Tracking	± 2400 Cents per key	0
Velocity Tracking	± 7200 Cents	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

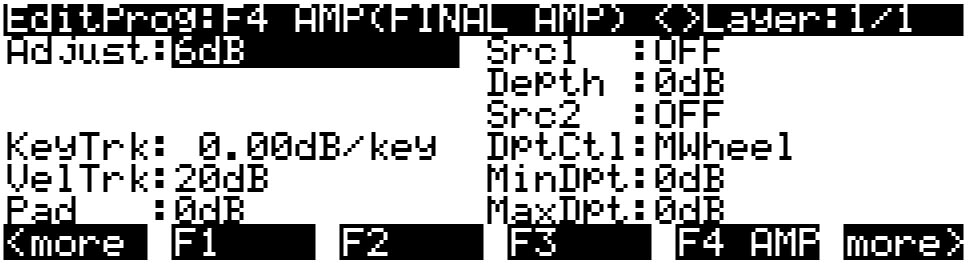
This measures pitch adjustment by the relative frequency (in Hertz) of each note. This is useful for controlling the beat frequency between layers in a multi-layered program. Using this parameter to detune chorusing layers will keep the beat frequency constant across much of the keyboard. Although the 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.



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

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

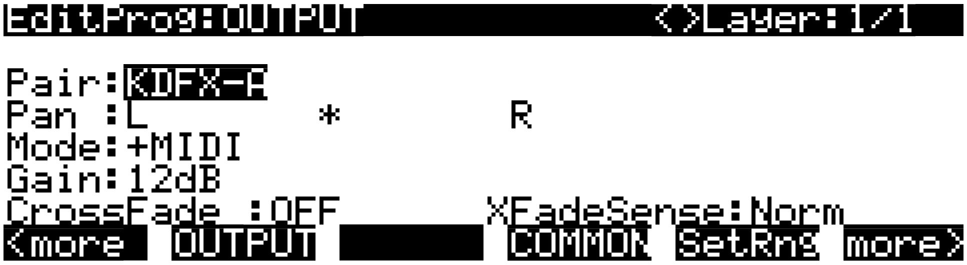
The OUTPUT Page

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.



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

Pair:KDFX-A
Pan :L       *       R
Mode:+MIDI
Gain:12dB
CrossFade :OFF       XFadeSense:Norm
```

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)

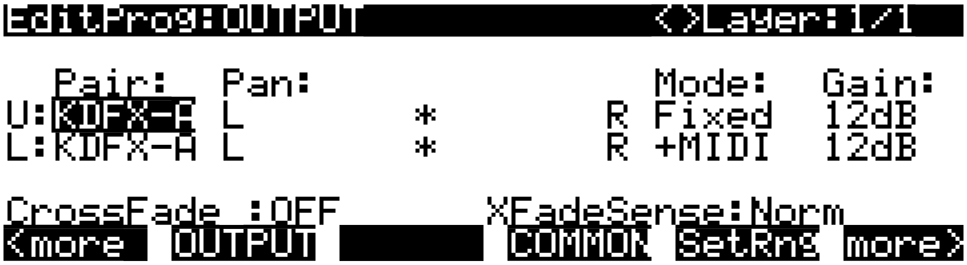
The Crossfade parameter lets you select a control source to fade the current layer's amplitude from zero to maximum. When Crossfade Sense is **Normal**, the layer is at full amplitude when the Crossfade control is at minimum. With Crossfade Sense set to **Reverse**, the layer is at zero amplitude when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

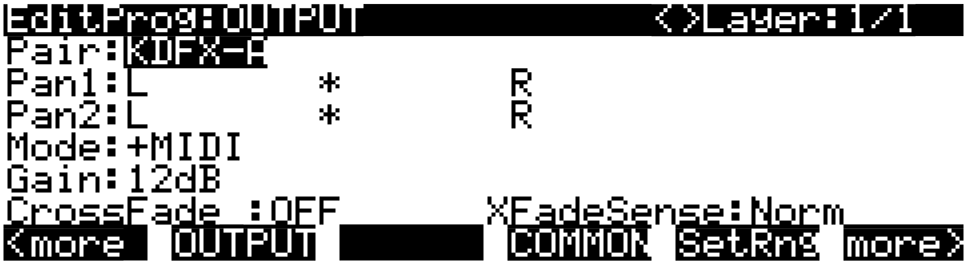
The following page is for a layer with one keymap and a double-output algorithm. The U and L stand for the upper and lower wires (signal paths). You have independent control of the output parameters for each wire.



```
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>
```

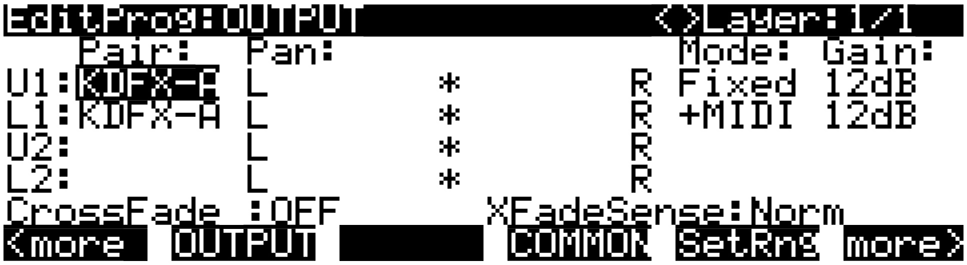
Next are the two page configurations for layers with stereo keymaps: the first one uses a single-output algorithm, and the second uses a double-output algorithm.

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>
```

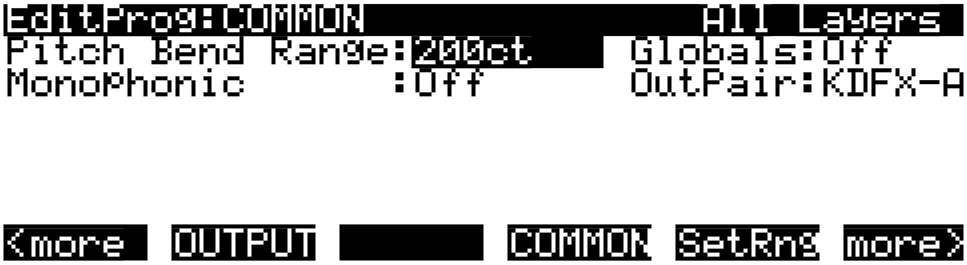
When a stereo keymap layer uses a double-output algorithm, both keymaps are split between the upper and lower wires. In other words, both wires carry the signal from each of the keymaps. The Output Group (Pair), Output mode, and Gain level of Keymap 1 are mimicked by Keymap 2 (that's why these parameters 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.



```
EditProg:OUTPUT             <>Layer:1/1
   Pair:   Pan:               Mode:  Gain:
U1:KDFX-A L        *       R Fixed  12dB
L1:KDFX-A L        *       R +MIDI  12dB
U2:       L        *       R
L2:       L        *       R
CrossFade :OFF      XFadeSense:Norm
<more   OUTPUT          COMMON  SetRng  more>
```

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.



```
EditProg:COMMON                  All Layers
Pitch Bend Range:200ct     Globals:Off
Monophonic      :Off       OutPair:KDFX-A



<more   OUTPUT          COMMON  SetRng  more>
```

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.

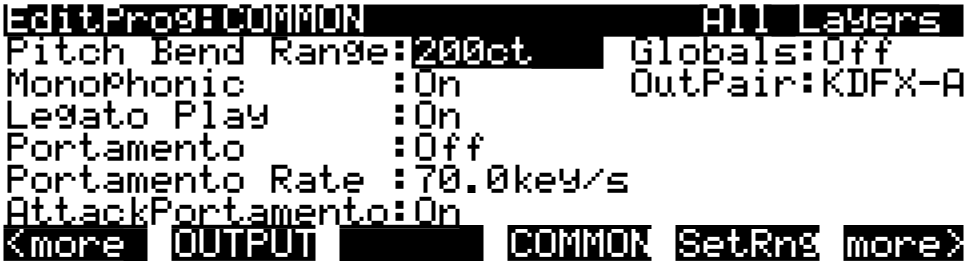
Setups include a parameter called Bend Range (on the BEND page in the Setup Editor), which in most setups does not affect the program-level Pitch Bend Range setting. This is because in most setups the default value for Bend Range is **Prog**, which gives control of the pitch bend range to each program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



```
EditProg:COMMON                All Layers
Pitch Bend Range:200ct     Globals:Off
MonoPhonic      :On        OutPair:KDFX-A
Legato Play     :On
Portamento      :Off
Portamento Rate :70.0key/s
AttackPortamento:On
<more   OUTPUT          COMMON  SetRng  more>
```

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.

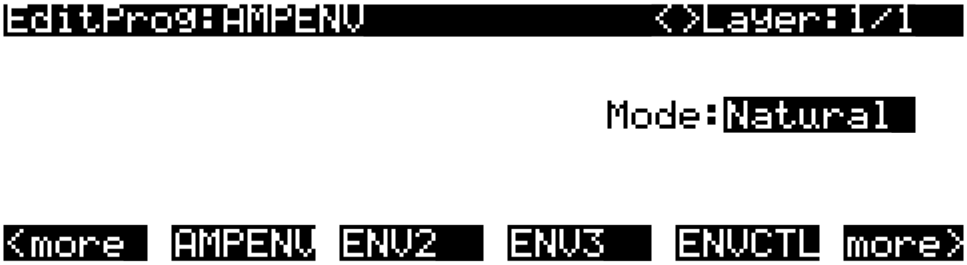
OutPair

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.

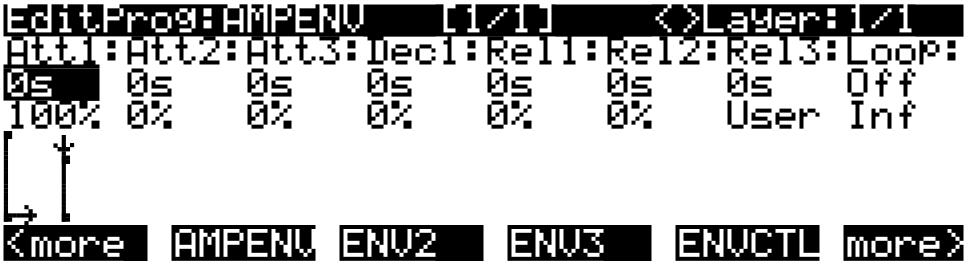
Press the **AMPENV** soft button to reach the Amplitude Envelope page. For many programs, it will look like the diagram below, which tells you that the amplitude for the current layer is the default, "natural" ROM amplitude envelope that's applied to each sample and waveform during its original development 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.



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

                               Mode:Natural
```

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.



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

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

Because the 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

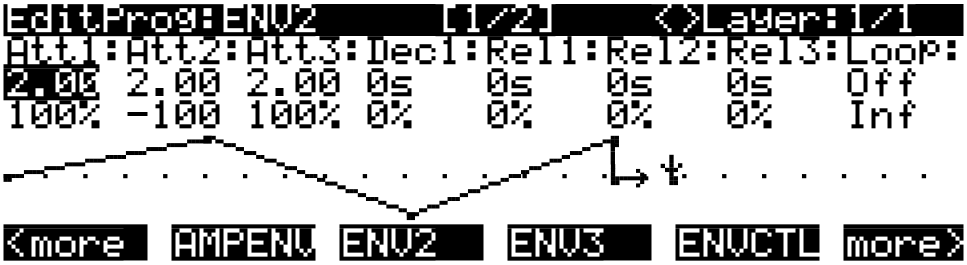
A value of **Inf** makes the amplitude envelope loop until a Note Off is generated. Values of **1** through **31** indicate how many times the loop will repeat after the amplitude envelope has played once through its normal cycle.

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).

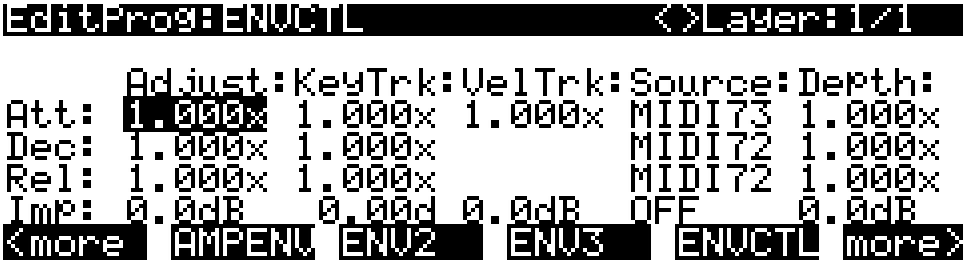


```
EditProg:ENV2      [1/2]     <>Layer:1/1
Att1:Att2:Att3:Dec1:Rel1:Rel2:Rel3:Loop:
2.00 2.00 2.00 0s   0s   0s   0s   Off
100% -100 100% 0%   0%   0%   0%   Inf
```

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

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.



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

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

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

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

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

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

Parameter 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.

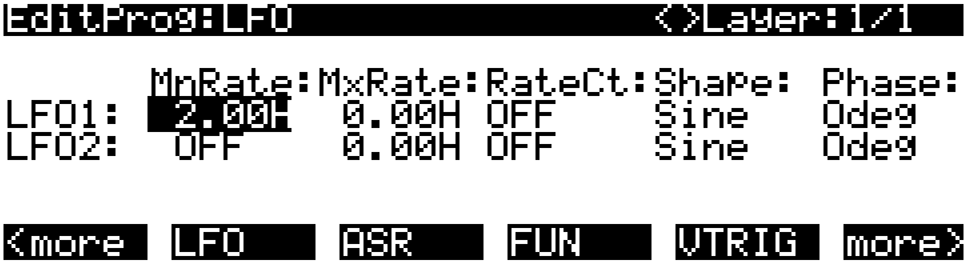
The LFO Page

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.



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

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

<more   LFO     ASR     FUN     VTRIG   more>
```

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

Parameter 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.

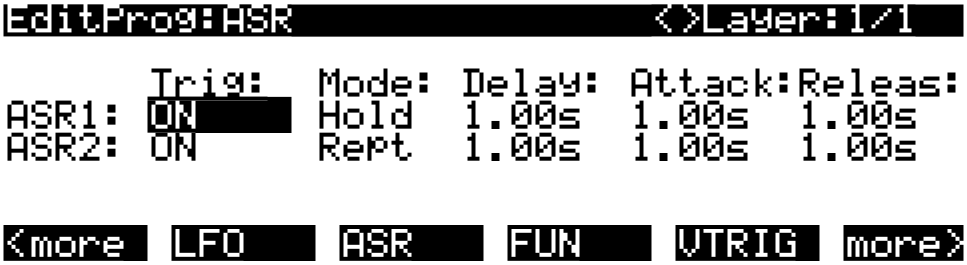
LFO Phase

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.



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

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

<more   LFO     ASR     FUN     VTRIG   more>
```

Parameter 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

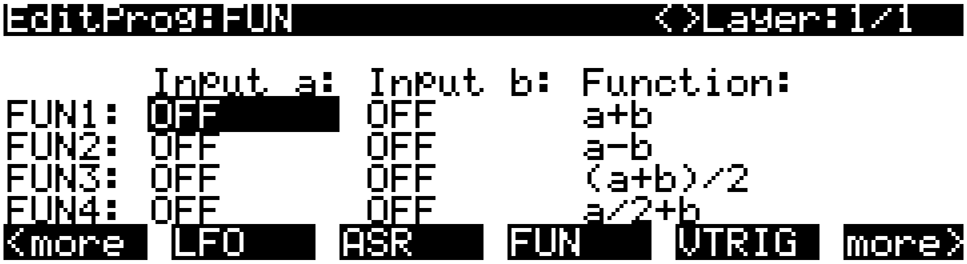
This defines how long the ASR takes to ramp up from minimum to maximum effect on whatever it's patched to.

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:



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

      Input a:  Input b:  Function:
FUN1: OFF       OFF       a+b
FUN2: OFF       OFF       a-b
FUN3: OFF       OFF       (a+b)/2
FUN4: OFF       OFF       a/2+b
<more   LFO     ASR     FUN     VTRIG   more>
```

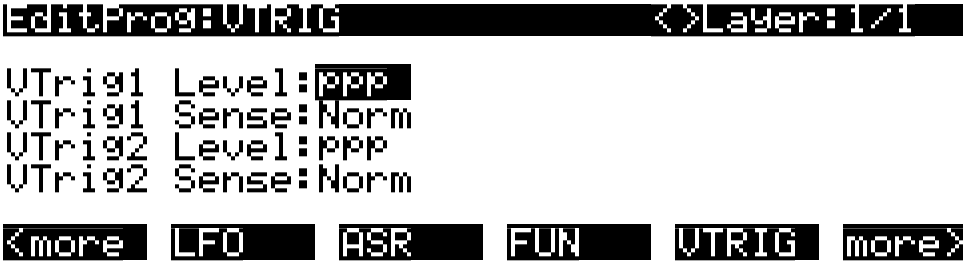
There are three parameters for each FUN. Inputs **a** and **b** can be any control source from the Control Source list. The control sources you want to combine are the ones you'll assign as the values for these parameters.

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.



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

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

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

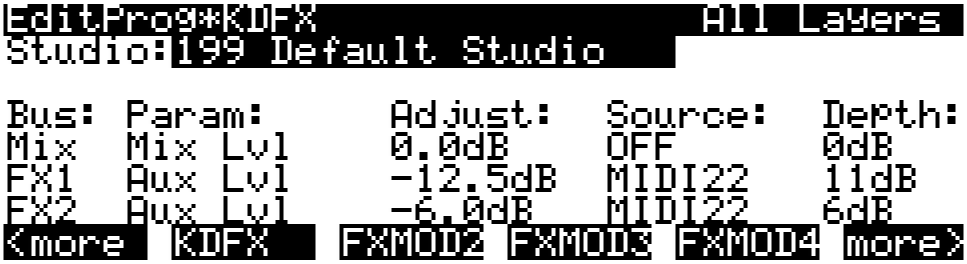
The velocity trigger's level is expressed in terms of the standard dynamic markings of western music—ppp, pp, p, mp, mf, f, ff, and fff. The K2600 converts each attack velocity value it receives into one of these eight levels. When a velocity trigger has been assigned as a control source, the K2600 compares 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.



```
EditProg*KDFX                     All Layers
Studio:199 Default Studio

Bus: Param:     Adjust:  Source:  Depth:
Mix  Mix Lvl    0.0dB    OFF      0dB
FX1  Aux Lvl    -12.5dB  MIDI22   11dB
FX2  Aux Lvl    -6.0dB   MIDI22   6dB
<more   KDFX   FXMOD2 FXMOD3 FXMOD4 more>
```

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.

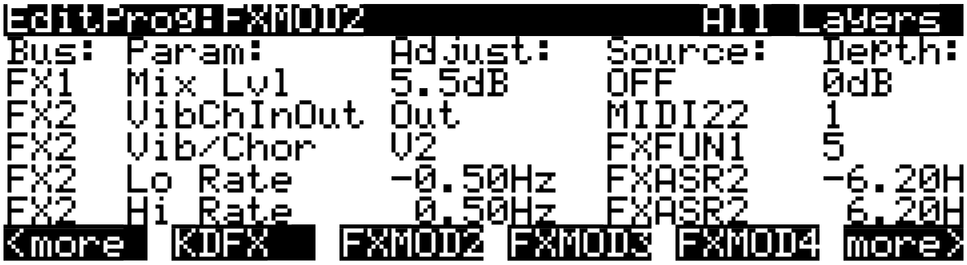
On the Program-editor page above, the first FXMod affects the level on the Mix bus, overriding whatever mix level is defined for the studio.

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.



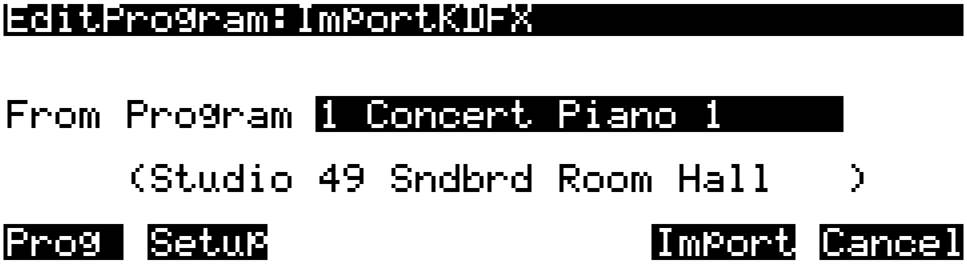
```
EditProg:FXMOD2                 All Layers
Bus: Param:      Adjust:   Source:   Depth:
FX1  Mix Lvl     5.5dB     OFF       0dB
FX2  VibChInOut  Out       MIDI22    1
FX2  Vib/Chor    V2        FXFUN1    5
FX2  Lo Rate     -0.50Hz   FXASR2    -6.20H
FX2  Hi Rate      0.50Hz   FXASR2     6.20H
```

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.



```
EditProgram:ImportKDFX

From Program 1 Concert Piano 1

     (Studio 49 Sndbrd Room Hall   )
```

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.

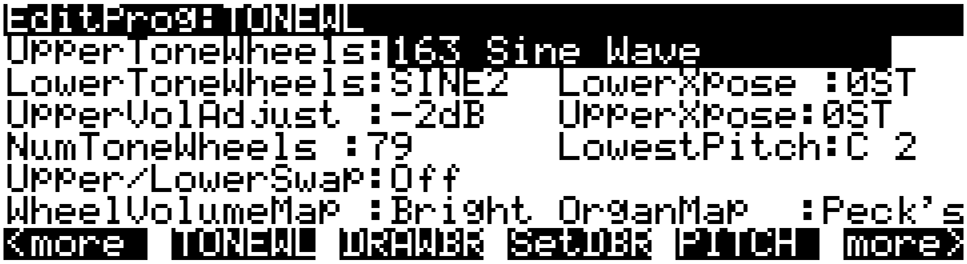
Editing KB3 Programs

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.



```
EditProg:TONEWL
UpperToneWheels:163 Sine Wave
LowerToneWheels:SINE2  LowerXpose :0ST
UpperVolAdjust :-2dB   UpperXpose:0ST
NumToneWheels :79      LowestPitch:C 2
Upper/LowerSwap:Off
WheelVolumeMap :Bright OrganMap   :Peck's
<more  TONEWL  DRAWBR  SetDBR  PITCH  more>
```

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

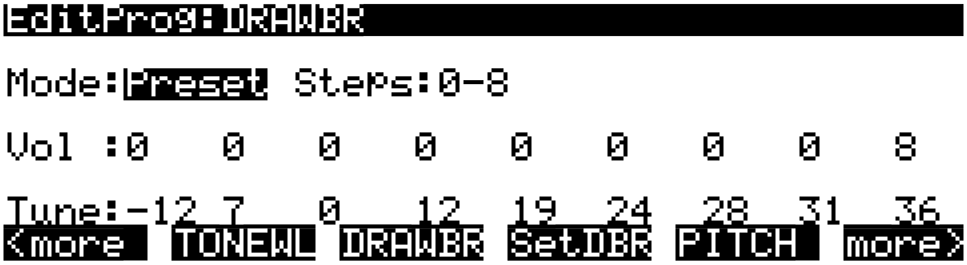
These two parameters let you transpose the upper and/or lower tone wheels in semitone steps away from their default tunings. The available ranges of values for these parameters depends on the setting of the Upper/LowerSwap parameter.

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.



```
EditProg:DRAWBR
Mode:Preset Steps:0-8
Vol :0   0   0   0   0   0   0   0   8
Tune:-12 7   0   12  19  24  28  31  36
<more   TONEWL  DRAWBR  SetDBR  PITCH   more>
```

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

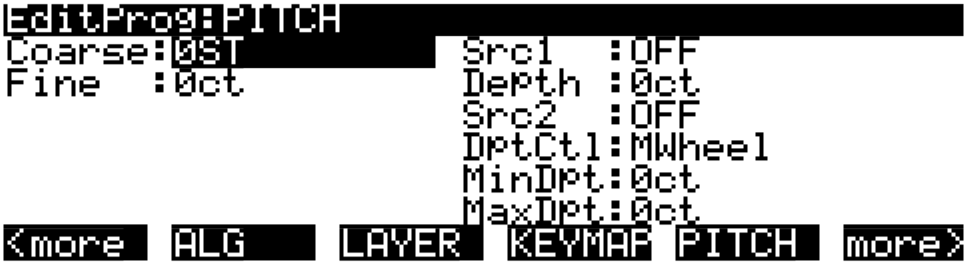
This parameter lets you tune each of the nine drawbars up or down in semitone steps. The values for the Tune parameter on the DRAWBR page shown above represent standard drawbar settings on a real B 3, as shown in Table 6-1 on page 6-5.

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

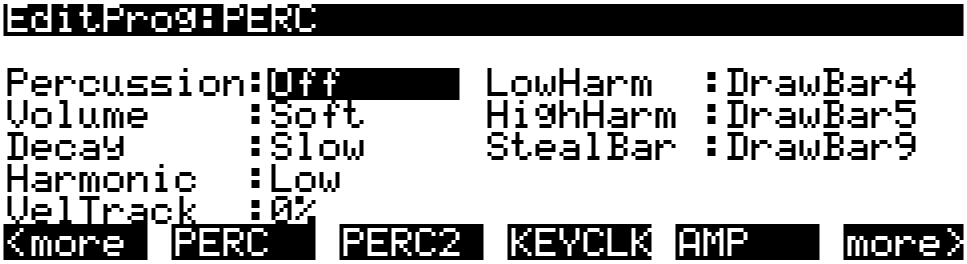
The PITCH page for KB3 programs is much like the PITCH page for VAST programs. The only difference is that for KB3 programs, there are no FineHz, KeyTrk, or VelTrk parameters. For a full description of the PITCH-page parameters, see page 6-27.



```
EditProg:PITCH
Coarse:0ST                Src1  :OFF
Fine  :0ct                Depth :0ct
                          Src2  :OFF
                          DptCtl:MWheel
                          MinDpt:0ct
                          MaxDpt:0ct
```

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.



```
EditProg:PERC

Percussion:Off        LowHarm  :DrawBar4
Volume    :Soft       HighHarm :DrawBar5
Decay     :Slow       StealBar :DrawBar9
Harmonic  :Low
VelTrack  :0%
```

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.

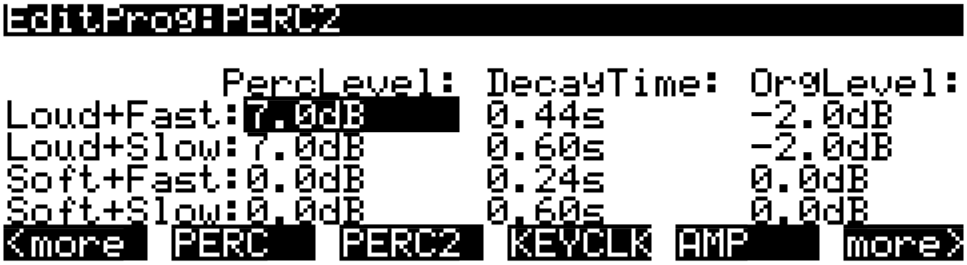
HighHarm

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



```
EditProg:PERC2

          PercLevel: DecayTime: OrgLevel:
Loud+Fast:7.0dB      0.44s      -2.0dB
Loud+Slow:7.0dB      0.60s      -2.0dB
Soft+Fast:0.0dB      0.24s      0.0dB
Soft+Slow:0.0dB      0.60s      0.0dB
```

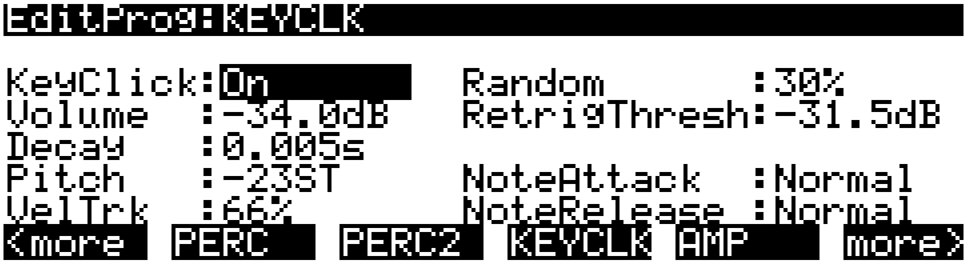
Parameter Group (Available for each combination of the Volume and Decay parameters on the PERC page)	Range of Values
Percussion Level	0 to 24.0 dB
Decay Time	0.10 to 5.10 seconds, in .02-second increments
Organ Volume Level	-12.0 to 12.0 dB

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.



```
EditProg:KEYCLK

KeyClick:On         Random      :30%
Volume  :-34.0dB    RetrigThresh:-31.5dB
Decay   :0.005s
Pitch   :-23ST      NoteAttack  :Normal
VelTrk  :66%        NoteRelease :Normal
```

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

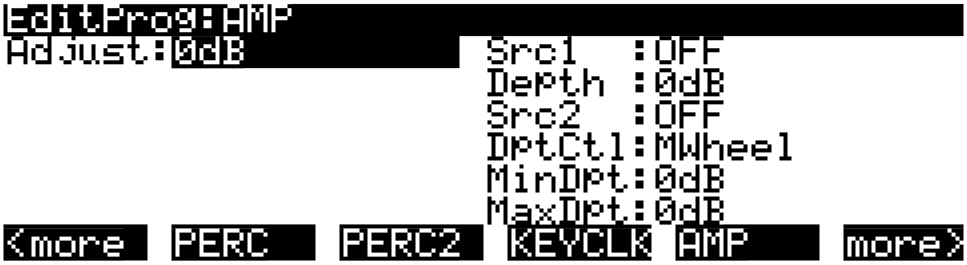
Controls the attack characteristic of notes. **Normal** provides a smoothed attack, while a setting of **Hard** has an instant attack and will produce an audible click, in addition to any amount of key click specified with the other parameters on this page (you might prefer not to specify any additional 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

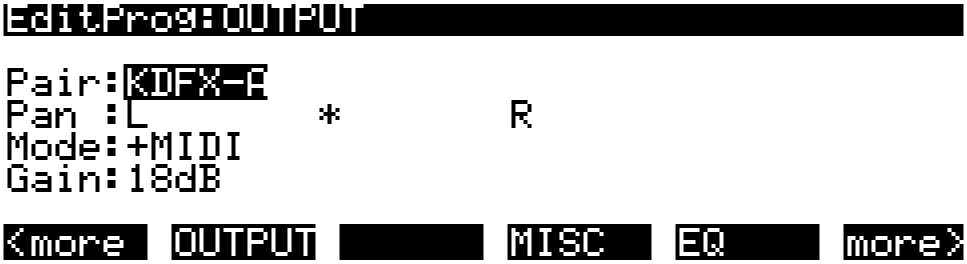
Assign amplitude controllers on this page, which is similar to the F4 AMP page for VAST programs (see page 6-28 for a full description. The only difference between the AMP pages for KB3 programs and VAST programs is that the KB3 version doesn’t have the KeyTrk, VelTrk, and Pad parameters.



```
EditProg:AMP
Adjust:0dB             Src1   :OFF
                       Depth  :0dB
                       Src2   :OFF
                       DptCtl:MWheel
                       MinDpt:0dB
                       MaxDpt:0dB
```

The OUTPUT Page

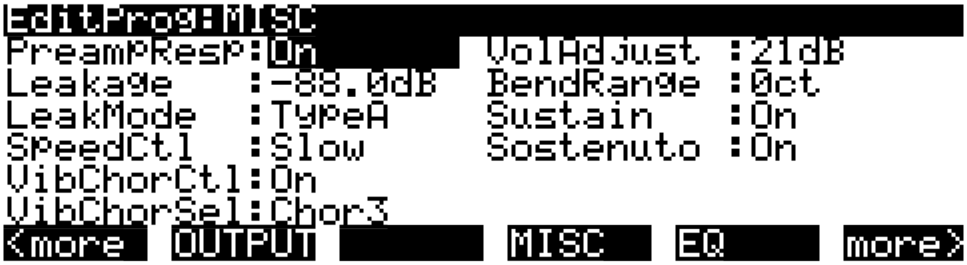
Use this page to route the programs signal from the sound engine to KDFX. This page is similar to the VAST-program version; the only difference is that it doesn’t have the CrossFade and XFadeSense parameters. See page 6-29 for a description of the OUTPUT-page parameters.



```
EditProg:OUTPUT

Pair:KDFX-A
Pan :L        *        R
Mode:+MIDI
Gain:18dB
```

The MISC Page

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



```
EditProg:MISC
PreampResp:On        VolAdjust :21dB
Leakage   :-88.0dB   BendRange :0ct
LeakMode  :TypeA     Sustain   :On
SpeedCtl  :Slow      Sostenuto :On
VibChorCtl:On
VibChorSel:Chor3
```

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

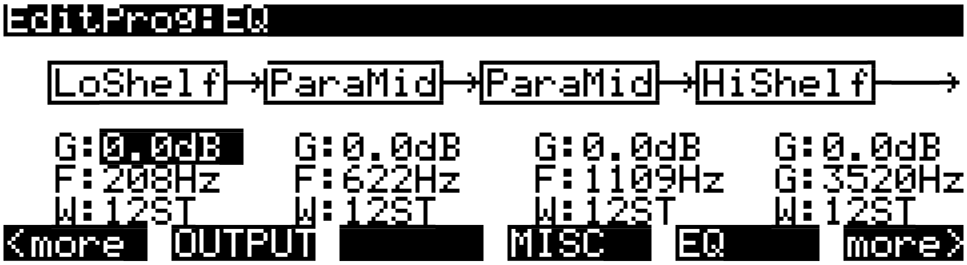
Set **On** or **Off** to enable or disable response to MIDI sustain (MIDI 64).

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.



```
EditProg:EQ

LoShelf → ParaMid → ParaMid → HiShelf →

G:0.0dB    G:0.0dB    G:0.0dB    G:0.0dB
F:208Hz    F:622Hz    F:1109Hz   G:3520Hz
W:12ST     W:12ST     W:12ST     W:12ST
```

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.

Chapter 7
Setup Mode and the Setup Editor

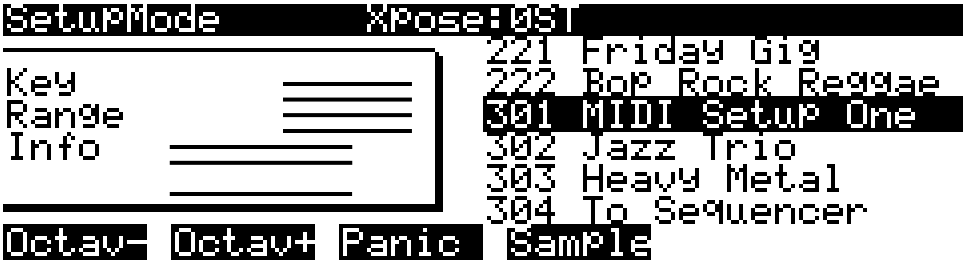
Setup Mode

In Setup mode, the K2600 can take on the identity of eight distinct instruments and eight distinct MIDI transmitters, each of which can use the setup's physical controller assignments (or any subset of those controller assignments). For example, you can create a setup that is split into eight different keyboard regions (called zones). Each zone can play its own program, while also transmitting on its own MIDI channel.

Selecting setups in Setup mode is much like selecting programs in Program mode—just use one of the normal data entry methods to scroll through the list of setups. With the K2600R, standard program changes from your MIDI controller select the correspondingly numbered setups while the K2600 is in Setup mode (if you set the LocalKbdCh parameter on the MIDI-mode RECEIVE page to match the transmit channel of your MIDI controller).

There are some important differences between a program and a setup. A program plays on a single keyboard zone and on a single MIDI channel. A setup enables you to use up to eight keyboard (or MIDI controller) zones, each of which can have its own program, MIDI channel, and control assignments. The parameters you define for each setup affect programs *only while you are in Setup mode*. An exception to this is the control setup, which we discuss on page 7-3.

Press the Setup-mode button to enter Setup mode. You'll see a list of setups, which you can select with any data entry method.

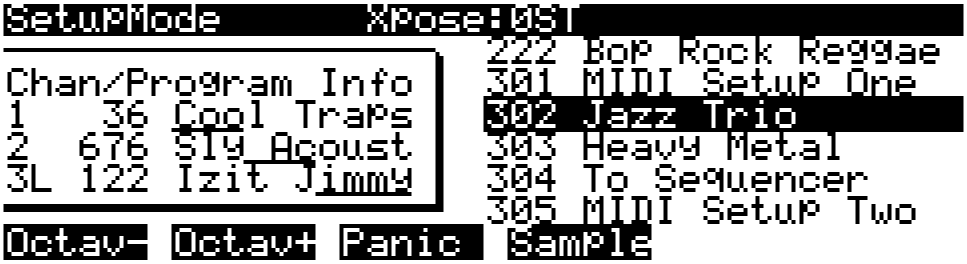


```
SetupMode       Xpose:0ST
                221 Friday Gig
Key             222 Bop Rock Reggae
Range           301 MIDI Setup One
Info            302 Jazz Trio
                303 Heavy Metal
                304 To Sequencer
```

The lines in the info box represent the approximate key range of each zone, and let you know if any zones overlap. In the preceding diagram, the setup has seven active zones (Zone 7 is turned off); Zones 1–4 are at the upper end of the keyboard. Zones 5, 6, and 8, which overlap Zones 1–4, cover the lower two thirds of the keyboard.

For setups containing three or fewer zones, the box displays the MIDI channel and program assignments for each zone, with lines under the Program names to indicate the key range of each zone (as shown in the following diagram). An **L** or an **M** next to the channel number indicates that the zone transmits only locally or via MIDI (the default is Local *and* MIDI).

Off indicates that the zone has been turned off completely (when a zone is turned off, no MIDI, program, or key-range information is visible for the zone).



```
SetupMode       Xpose:0ST
                     222 Bop Rock Reggae
Chan/Program Info    301 MIDI Setup One
1   36 Cool Traps    302 Jazz Trio
2  676 Sly Acoust    303 Heavy Metal
3L 122 Izit Jimmy    304 To Sequencer
                     305 MIDI Setup Two
Octav- Octav+ Panic  Sample
```

You can transpose the entire setup up or down with the two **Octav** soft buttons. Press them simultaneously to set the transposition back to zero. When you transpose a setup, the split points between zones remain in place; each program is transposed within its respective zone.

The **Panic** soft button sends All Notes Off and Reset All Controllers messages to all zones. The **Sample** soft button provides convenient access to the K2600's sampler. Refer to Chapter 14 for complete information on the sampler.

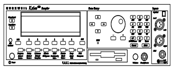
When you select a setup in Setup mode, the K2600 sends a number of MIDI messages, on each of the MIDI channels used by the setup. Some of these include: Program Change commands, MIDI Bank Select messages, Pan and Volume messages, and entry values for physical controllers (entry values are the values that take effect as soon as you select the setup; there are also controller *exit* values, which are the values of the controllers when you leave the setup—either by selecting another setup or by exiting Setup mode). The values of all these messages depend on the parameters you define in the Setup Editor.

Loading Older Setup Versions

You can load setups created on the K2000 or K2500 into the K2600. Setups created on the K2500 are the same as those created on the K2600, so they're completely interchangeable. Since setups created on the K2000 have fewer features (like three zones instead of eight), you have two choices for using K2000 setups on the K26000. You can leave the K2000 setups as they are, and they'll work for either the K2000 or the K2600—but they won't have all the K2600 features like eight zones. Or you can edit the K2000 setups to take advantage of the K2600's expanded features. Once you do this, however, the setups will no longer work on the K2000, so if you want to keep using those setups on a K2000, be sure to keep backup copies of the original setups.

If you edit a K2000 setup to use K2600 features, you may need to reassign the programs it uses. If your K2000 setup uses a K2000 ROM program, the setup won't play the same program when you load it into the K2600, because the two instruments have different lists of ROM programs. In this case, you can either select a similar program for the K2600 setup (if one exists), or you can save the programs used by the K2000 setup into the same ROM IDs in the K2600 (this doesn't really replace the K2600 programs; they reappear when you delete the K2000 programs). If your K2000 setup uses K2000 RAM programs, you won't have any problems as long as you load dependent objects when you load the K2000 setup into the K2600.

K2600 Rack Models Only



The K2600R offers the same splitting and layering capabilities as the keyboard model. Even if your MIDI controller can transmit on only one MIDI channel at a time, you can use the K2600R's setups by setting the Local Keyboard Channel parameter on the MIDI-mode RECEIVE page to match the channel you're using to transmit from your MIDI controller. This will enable you to play the setup's eight zones, and also send the MIDI information from your MIDI controller to the K2600R's MIDI Out port, on the channels used by the current setup.

Once you have properly set the Local Keyboard Channel, the K2600R will remap Controller messages from your MIDI controller.   Some examples of these Controller messages include: Modulation (01), Foot (04), Data (06), Mono Pressure (32), Sustain (64), and Sostenuto (66). By sending preset controller numbers, you can remap them to the controllers assigned within the Setup Editor. For these preset numbers, along with more on using a MIDI controller with the K2600R, see the discussion of the Local Keyboard Channel in Chapter 10.

If you send data on a channel other than the Local Keyboard Channel, no remapping will occur. You will hear only the program assigned to the channel on which your MIDI controller is transmitting.

For the remainder of this chapter, we'll cover topics that apply to both the rack and keyboard versions of the K2600. However, any references to the sliders, ribbons, wheels, buttons, or other physical controllers are intended primarily for K2600 keyboard owners.

The Control Setup

In addition to zone splitting and layering, Setup mode is a powerful way to take advantage of the K2600's programmable sliders, ribbon controllers, and assignable buttons. In order to provide some of the same flexibility for Program mode, we created the control setup, which defines the controller assignments for programs in Program mode.

The default control setup is **97 Control Setup**, but you can choose any control setup you want. To do this, go to the MIDI-mode TRANSMIT page and use any normal data entry method to change the CtlSetup parameter. When you reenter Program mode, all programs will now respond to many of the controller assignments defined in Zone 1 of the control setup (Zones 2–8 are not relevant in Program mode, because a program can occupy only one MIDI channel).

To edit the control setup, press the **Edit** button while the CtlSetup parameter is highlighted on the MIDI-mode TRANSMIT page. This brings you to the Setup Editor, which is described in the following sections. The following table shows which control-setup parameters affect controller assignments in Program mode.

Control Setup–Setup Editor Page, Zone 1	Parameters Affecting Program Mode	Parameters Not Affecting Program Mode
CH/PROG	ZoneArpeg, Destination, MIDIBankMode	LocalPrg, Out, Channel, MIDIBank, MIDIProg, Status, EntryProgChg
KEY/VEL	VelScale, VelOffset, VelCurve	LoKey/HiKey, Transpose, NoteMap, LoVel/HiVel
PAN/VOL	None	
BEND	All	
COMMON	Sync	
ARPEG	All	
RIBCFG	All	
Continuous Controller assignment pages (SLIDER, SLID/2, CPEDAL, RIBBON, WHEEL, PRESS)	Dest, Scale, Add, Curv	Ent and Exit
Switch Controller assignment pages (FOOTSW, SWITCH)	SwType, Dest, On, Off	Ent and Exit
KDFX, FXMOD2, FXMOD3, FXMOD4, FXLFO, FXASR, FXFUN	None	

Table 7-1 Control Setup Parameters Affecting Program Mode

In summary, physical controller destinations, their curves and states, and the Arpeggiator parameters all define controller assignments for programs in Program mode. The other parameters have no effect; this keeps Program mode relatively simple. Program mode lets you change values for transposition, MIDI channels, and programs independently of the control setup.

Once you save changes to the control setup, those changes will affect all programs when you are in Program mode. For example, programming the Large Ribbon in the control setup to have three sections will mean that in every program in Program mode, you will have a three-section Large Ribbon.

You may want to program several different control setups, and switch among them for different applications. Suppose, for example, that you're recording a song, but you don't want to record aftertouch. You can create a setup with pressure turned off in Zone 1 (on the PRESS page in the Setup Editor, set the value of the Press parameter to **Off** for Zone 1). Now whenever you want to record without aftertouch, just select this setup as the control setup.

Zone-status LEDs in Setup Mode (Keyboard Models Only)

Take a minute to scroll through some of the factory setups. As you change setups, you’ll notice that the LEDs in the eight buttons above the programmable sliders go on and off and change color. These LEDs indicate the status of each of the zones in the setup. You may also see the **Solo** button go on. This means that the setup is configured to have only one zone playing when you select it.

While you’re in Setup mode, each of the eight zone-status LEDs will always be in one of four states:

Off Empty zone—that is, a zone that has no program or MIDI channel associated with it. For example, if you select a setup and only four status LEDs light up (regardless of their color), the setup contains just four zones. Whenever you’re in Setup mode, the number of lines in the info box matches the number of zone-status LEDs that are lit.

Red Soloed zone. As you might have guessed, only one zone can be soloed at a time. When a zone is soloed, *only* that zone plays notes and generates controller information. Other zones, if they’re not turned off, still generate program changes and entry/exit controller values.

Green Active zone. As long as no other zone is soloed, an active zone plays notes— and generates controller information, program changes, and entry/exit controller values. If another zone is soloed, an active zone is “backgrounded”—it’s status LED remains green, but it doesn’t play notes or generate controller information.

Orange Muted zone. Muted zones don’t play notes or generate controller information, but they do generate program changes and entry/exit controller values.

In performance situations, the zone-status buttons provide a convenient way to temporarily change the status of one or more zones. This can be very effective for bringing voices and/or controller configurations into and out of your performance. The best way to get familiar with this technique is to play with the buttons, as the next few paragraphs describe.

Select a setup (look for one with lots of active zones that cover the whole keyboard), and play a few bars. You’ll hear sounds corresponding to each of the active zones (green LEDs). If you see any muted zones (orange LEDS), press their zone-status buttons, and they’ll become active. Play around a bit. Try muting all the zones, then bringing them back one by one until all the zones in the setup are active.

Now press the **Solo** button. The **Solo**-button LED lights (it’s always red), and one of the zone-status LEDs (never more than one) turns red. You’ll now hear only that zone as you play. All the active zones are now backgrounded—they’ll still generate program changes and entry/exit controller values, but you won’t hear anything from them.

Now press one of the zone-status buttons. Its LED turns red, and it becomes the soloed zone. The previously-soloed zone returns to its programmed status. Solo each zone in turn, using the info box in the display to find the range covered by the zone. Note that you can solo a zone even if it’s muted.

Press the **Solo** button again, and its LED goes out. The previously-soloed zone returns to its programmed status, and you can hear all the active zones again.

Table 7-2 gives you a quick visual reminder of how zones behave depending on their status.

LED Color	Zone Status	Data Generated by Zone			
		Notes	Controllers	Program Number	Entry and Exit Values
Red	Soloed	✔	✔	✔	✔
Green (no others are red)	Active	✔	✔	✔	✔
Green (another is red)	Backgrounded			✔	✔
Orange	Muted			✔	✔
(Off)	Empty				

Table 7-2 Zone Status in Setup Mode

Remember that any changes you make to zone status in Setup mode are temporary; as soon as you select another setup, that setup’s programmed zone status takes over. To change a setup’s zone status permanently, use the Setup Editor (see *Status* on page 7-9).

The Setup Editor

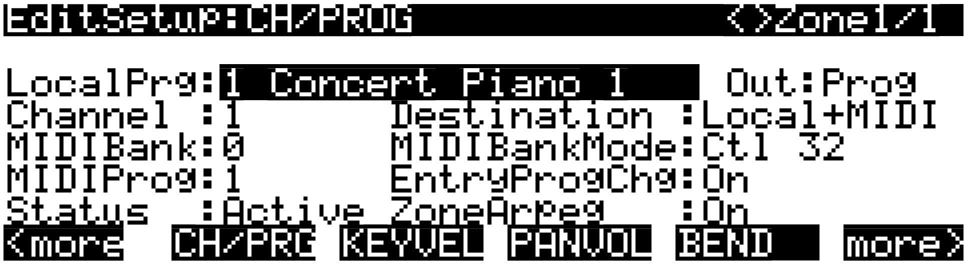
From Setup mode, press the **Edit** button to enter the Setup Editor, where you can make changes to the currently selected setup. Use the soft buttons to select the various Setup-editor pages. The upper line of each page displays the usual mode reminder, as well as the current Setup-editor page, and the current zone. Use the **Chan/Bank** buttons to select one of up to eight different zones, each having its own set of Setup-editor pages.

The parameters on the Setup-editor pages define what each of a setup’s zones sends—both to internal programs and to the MIDI Out port. They also determine how the K2600 responds to MIDI signals received from a MIDI controller connected to the K2600’s MIDI In port (when the Local Keyboard Channel matches the transmit channel of your MIDI controller).

The display diagrams you see in this chapter show the default values for setup **97 ControlSetup**.

The Channel/Program (CH/PROG) Page

This is the first page you see when you enter the Setup Editor. Here, you can select programs, MIDI channels, and MIDI Bank numbers for each of the setup's eight zones. You can also solo or mute each zone, and assign zones to be controlled by the K2600's Arpeggiator.



```
EditSetup:CH/PROG                <>zone1/1

LocalPrg:1 Concert Piano 1     Out:Prog
Channel :1      Destination :Local+MIDI
MIDIBank:0      MIDIBankMode:Ctl 32
MIDIProg:1      EntryProgChg:On
Status  :Active ZoneArpeg   :On
<more   CH/PRG KEYVEL PANVOL BEND    more>
```

Parameter	Range of Values	Default
Local Program	Program list	1 Concert Piano 1
Channel	1-16	1
MIDI Bank	Variable, dependent on MIDI Bank Mode	0
MIDI Program	Variable, dependent on MIDI Bank Mode	1
Status	Status list	Active
Output assignment (Out)	Prog, KDFX-A to KDFX-D	Prog
Destination	Destination list	Local+midi
MIDI Bank Mode	MIDI Bank Mode list	Ctl 32
Entry Program Change	On/Off	On
Zone Arpeggiation (ZoneArpeg)	On/Off	On

Local Program (LocalPrg)

This selects an internal program to play on each zone. As you change the value of Local Program, notice that MIDI Program and MIDI Bank match the local program and bank numbers (Program IDs 0–99 correspond to MIDI Bank 0, 100–199 are MIDI Bank 1, and so on). If you want to transmit different program and bank numbers over MIDI, highlight either MIDI Program or MIDI Bank and select a new value. Note that changing the Local Program parameter again will reset both the MIDI Program and MIDI Bank parameters to match the local program and bank numbers.

Channel

The Channel parameter defines the MIDI transmit channel for the currently selected zone. You can set it to any of the 16 MIDI channels. Normally, you will want each zone on a separate MIDI channel. This is necessary if you want to combine different programs in the setup.

If two zones have the same MIDI channel (and destination), but they have different program settings, there will be conflicts: no MIDI device, including the K2600, can respond correctly to two different simultaneous Program Change commands on one channel. The result will be that only one Program Change will be recognized, and every note played will sound double (if Note Maps are on). This can create odd and unpredictable timing effects, and will reduce your polyphony by 50%.

Nevertheless, there will be occasions when “stacking” zones on the same MIDI channel might come in handy. Suppose you want a physical controller on the K2600 to send data for two *different* numbered MIDI Controllers on the *same channel*. In this case, you must create two zones assigned to the same channel, but with different controller assignments.

Here’s one example: if a receiving synth is using Controller #1 for modulation depth and Controller #13 for modulation speed, you can increase both the depth and the speed with Slider A. Start by assigning Slider A in Zone 1 to **MWheel** and in Zone 2 to **MIDI 13**; then assign both zones to the same MIDI channel. (You may want to make sure you aren’t sending doubled notes. Use the Note Map parameter on the KEY/VEL page to set one zone’s Note Map to **Linear** and the other zone’s Note Map to **Off**.)

Another example: create two or more zones that are identical except for their transposition settings. Now you can play parallel intervals (or chords) with single keystrikes.

MIDI Bank

The K2600’s programs are divided into ten MIDI banks, numbered 0-9. Program 99 in Bank 1, for example, is **199 Default Program**. The MIDI Bank parameter displays which bank the current program is assigned to, and automatically changes to match the Local Program value you set.

You can send Bank Select messages to external MIDI devices as well, by setting the Destination parameter to a value of **MIDI** or **Local + MIDI**, then changing MIDI Bank. Some instruments may have more than ten banks; the MIDI Specification says a device can have up to 16,384 banks, and the K2600 gives you access to all of them. Bank switching via MIDI makes it easy for the K2600 user to select sounds on external instruments, no matter how many banks they might have.

Different MIDI banks are accessible depending on the value of the MIDIBankMode parameter:

Value of MIDIBankMode	Available MIDI Banks
Ctl 0 or Ctl 32	0–127
Ctl 0/32	0–16383
K2000 or K1000	0–9
None	None

When you change the value of the LocalPrg parameter, the value of MIDIBank automatically changes correspondingly. If you want to transmit a MIDI bank number different from the one corresponding to the local program, select the local program first, then change the MIDI bank.

If you select an empty bank (like Bank 16, 383), the zone will still produce sound on the K2600, provided that Destination is set to **Local** or **Local + MIDI**. The LocalPrg parameter will display whatever internal program you set, but the bank number transmitted over the MIDI Out port will be different from the internal program’s bank number.

MIDI Program (MIDIProg)

MIDI Program defines which program number is transmitted out the MIDI Out port on the current zone's MIDI channel.

When you change the value of the LocalPrg parameter, the value of MIDIProg automatically changes correspondingly. If you want to transmit a MIDI program change number different from the one corresponding to the local program, select the local program first, then change the MIDI program.

Different programs are accessible depending on the value of the MIDIBankMode parameter:

Value of MIDIBankMode	Available Programs
Ctl 0 or Ctl 32	0–127
Ctl 0/32	0–16383
K2000 or K1000	0–9
None	None

Status

This parameter determines what the current zone does when you select the setup in Setup mode. If the value is **Active**, the zone sends and receives normally via MIDI. **Muted** means that the zone sends and receives program changes and entry/exit controller values, but doesn't play notes. **Soloed** causes only the current zone to play, "backgrounding" all other zones (backgrounded zones send and receive program changes and entry/exit controller values, but don't play notes). With a value of **Solo/M**, the current zone is both soloed and muted; when the zone isn't soloed, it won't produce sound).

See the section about zone-status LEDs (page 7-5) for more information about muting and soloing zones.

K2600R owners can mute and unmute zones by using the first three mode-selection buttons: **Program** (Mute 1), **Setup** (Mute 2), and **Q Access** (Mute 3). For setups containing up to three zones, pressing each button mutes or unmutes Zone 1, 2, or 3. This also works on the keyboard model.

For setups containing four or more zones, pressing **Mute 1** mutes the current zone; the zone becomes unmuted when you mute another zone. Pressing **Mute 2** or **Mute 3** solos the current zone. You can then press the **Chan/Bank** buttons to solo each zone.

On the K2600R, each button turns only one color (red); a system of lighted/unlighted buttons lets you know that certain zones are soloed or muted. When you use the **Chan/Bank** buttons to scroll to other zones, the buttons will light according to each zone's status. The following illustration explains how to interpret this system.

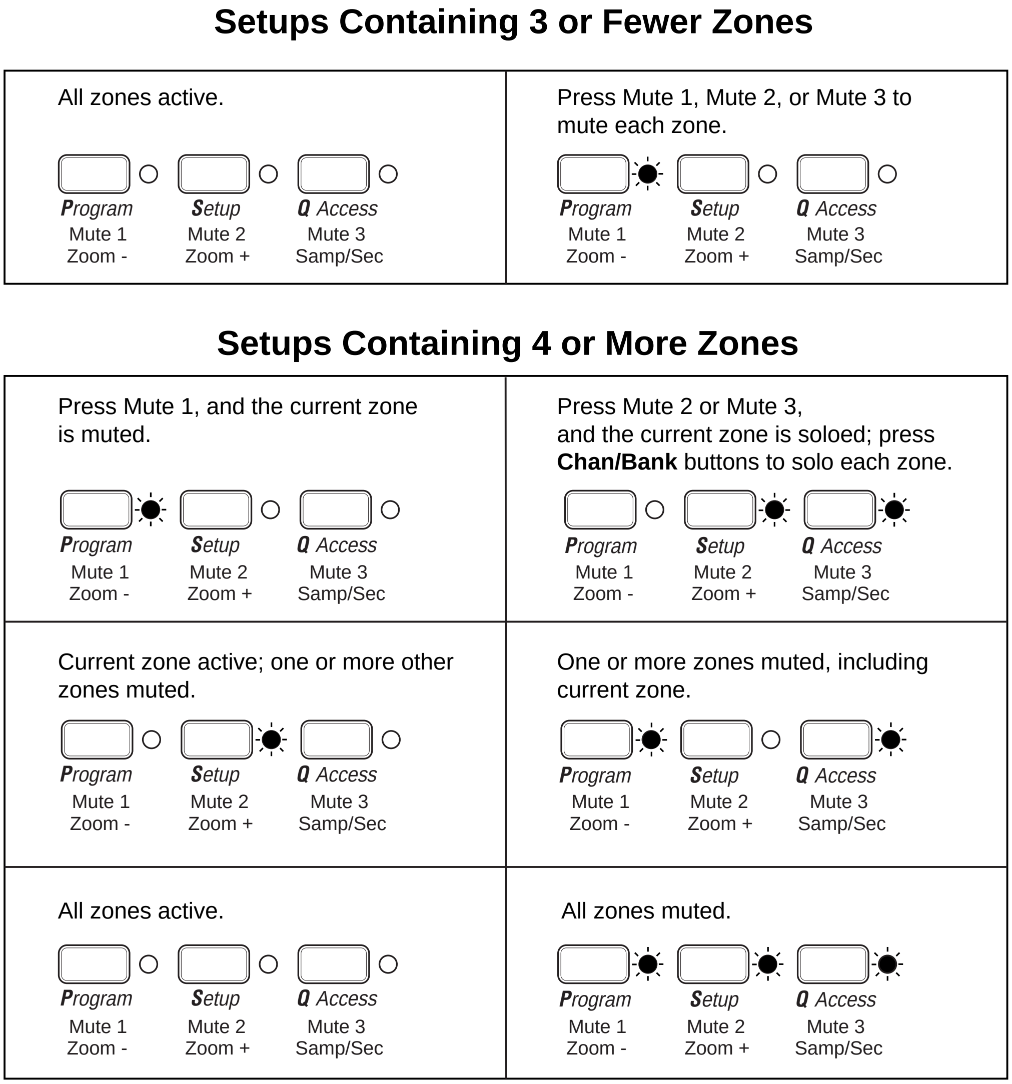

Setups Containing 3 or Fewer Zones
All zones active.
Press Mute 1, Mute 2, or Mute 3 to mute each zone.
Setups Containing 4 or More Zones
Press Mute 1, and the current zone is muted.
Press Mute 2 or Mute 3, and the current zone is soloed; press Chan/Bank buttons to solo each zone.
Current zone active; one or more other zones muted.
One or more zones muted, including current zone.
All zones active.
All zones muted.
Program
Mute 1
Zoom -
Setup
Mute 2
Zoom +
Q Access
Mute 3
Samp/Sec

Figure 7-1 Soloing Zones With the K2600R

The K2600 keyboard's mode-selection buttons also light as shown above. However, buttons **1–8** are more convenient for muting, unmuting, and soloing zones.

Output Assignment (Out)

This determines the signal routing for each zone in the setup—not the final audio output, but the KDFX input to which the zone's signal gets sent. You can route the zone to KDFX A through D, or you can use a value of **Prog**, which uses the output assignments of the program that's assigned to that zone.

Destination

This parameter determines whether the currently selected zone transmits only to the K2600 (**Local**), gets sent only via MIDI (**MIDI**), controls both the K2600 and connected MIDI instruments (**Local+MIDI**), or is disabled (**Off**). Setting Destination to **Off** turns the zone off completely; no MIDI data will be sent and no local program will sound.

MIDI Bank Mode

The MIDI Bank mode you choose determines how bank numbers will be sent over MIDI when the setup is selected, and in what format. It also affects how many MIDI banks and programs you can choose.

None means no bank number is sent, just the program number. **Ctl 0** means that the bank number is sent as a MIDI Controller #0 message. **Ctl 32** means it is sent as MIDI Controller #32. **Ctl 0/32** means it is sent as a dual-controller (two-byte) message, with the most-significant byte (MSB) of the bank number sent as Controller #0 and the least-significant byte (LSB) as Controller #32. Single-byte Bank Select messages (either 0 or 32) allow you to specify banks numbered 0-127. Two-byte messages allow you to specify banks numbered 0-16,383. With 128 programs per bank, this allows you to access 2,097,152 different programs on one instrument.

The MIDI Specification is a little ambiguous when it comes to Bank Select messages, as to whether they should be only Controller 0, only Controller 32, or both Controllers sent as a pair. Different manufacturers design their instruments to respond to different schemes, and if you send Bank Select in a form an instrument doesn't like, it may ignore it or interpret it incorrectly. The MIDI Bank Mode parameter is designed to allow the greatest flexibility in addressing other MIDI instruments. Usually you can look on the MIDI Implementation chart in the user's manual of an instrument to determine how it's designed to receive Bank Select messages, and then set MIDI Bank Mode for each zone to suit the instrument that is receiving data from it. The default setting, which works with the largest number of other instruments, is **Ctl 32**.

There are two other values for this parameter, which will be of special interest to owners of other Kurzweil instruments. A value of **K2000** is intended for use with the K2000, K2500, or K2600. The Bank Select message is sent as Controller 32, with a value between 0 and 9. Remember, the K2000, K2500, and K2600 support only 10 banks, with 99 programs per bank, so Program Changes 100 or higher are sent as Bank Select 1, followed by the last two digits as a Program Change. For example, if Program 124 is assigned to the zone, this will be sent out the MIDI Out port as Bank Select (Controller 32) 1, and then Program Change 24.

K1000 is intended for use with any of the 1200-series keyboards or modules, or any of the 1000-series instruments that have version 5 software installed. Those instruments predate the adoption of standard Bank Select messages; instead, they use Program Changes 100–109 as Bank Selects. If you select Bank 5: Program 42 for a K2600 zone, for example, it will send out Program Change 105 followed by Program Change 42. K2600 program numbers over 99 are not sent.

Entry Program Change (EntryProgChg)

This enables or disables bank and program change commands sent to internal programs or to the MIDI Out jack when you select setups. If it's set to **On**, the program numbers for the programs in the eight zones will be sent via MIDI when a setup is selected. By setting this parameter to **Off**, you can select a setup on the K2600 without changing the internal programs or those on MIDI devices receiving from the K2600. This is useful if you want to send only controller data to the K2600 or to MIDI devices, without changing program assignments.

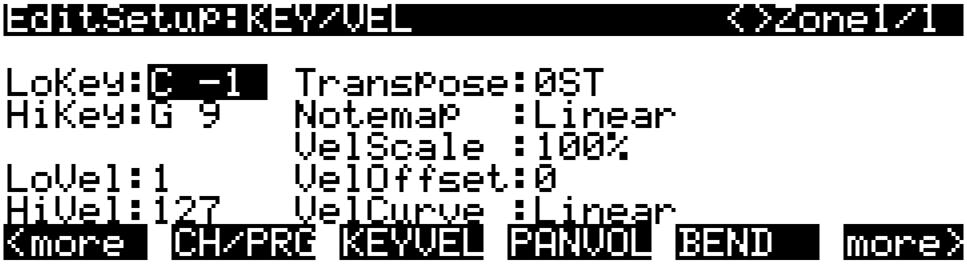
Zone Arpeggiation (ZoneArpeg)

Zone Arpeggiation determines whether the Arpeggiator will affect notes played in the current zone. The Arpeggiator affects only those zones that have this parameter set to a value of **On**.

For any given zone, the Arpeggiator plays notes only within that zone's Key Range. If the Arpeggiator, for example, tries to play a C$^{\#}$4 in a zone, but that zone's Key Range ends at C4, the note will not sound. However, another zone whose Key Range ends at C5 *will* be able to play the C$^{\#}$4 from the Arpeggiator. Therefore, setting a zone's Key Range can be important in deciding how it will respond to the Arpeggiator. (A separate set of range parameters, found on the ARPEG page, determines whether the notes you play get arpeggiated; see *Low Key (LoKey) and High Key (HiKey)* on page 7-38.)

The Key/Velocity (KEY/VEL) Page

The Key/Velocity page allows you to set key range, velocity range, transposition, and Note Maps for each zone.



```
EditSetup:KEY/VEL                <>Zone1/1

LoKey:C -1   Transpose:0ST
HiKey:G 9    Notemap  :Linear
             VelScale :100%
LoVel:1      VelOffset:0
HiVel:127    VelCurve :Linear
```

Parameter	Range of Values	Default
Low Key	C –1 to G9	C –1
High Key	C –1 to G9	G9
Transpose	-128 to +127 Semitones	0 semitones
Note Map	Note Map list	Linear
Low Velocity	1-127	1
High Velocity	1-127	127
Velocity Scale	±300%	100%
Velocity Offset	-128 to +127	0
Velocity Curve	Velocity Curve list	Linear

Low Key (LoKey), High Key (HiKey)

The LoKey and HiKey parameters define the note range of the currently selected zone. The easiest way to change these values is to press the **SetRng** soft button, which you can find by pressing either of the **more** soft buttons. You'll be prompted to trigger the notes you want to be the lowest and highest notes for the zone. When you do, you'll return to the Setup-editor page, and the notes you triggered will be reflected in the values for LoKey and HiKey. They'll also be represented by the lines beneath the program names in the box at the left of the Setup-mode page. You can set these values with normal data entry methods as well.

You can create "negative" ranges as well. To do this, select the HiKey parameter and set its limit *lower* than the LoKey limit. This results in the zone being active at the top and bottom of the keyboard, but being silent in the range between the two limits. This lets you create a layer with a “hole” in the middle, which you can then fill with a different sound on another zone.

Note that using the **SetRng** soft button won’t allow you to create a negative range, since it always defines the higher keystrike as the HiKey value. If you want to set HiKey lower than LoKey (or vice versa), use a normal data entry method. Note: Intuitive Entry doesn’t work for setting values for LoKey and HiKey.

The limits of MIDI are C-1 to G9. The untransposed 88-key range is A0 to C8. The untransposed 76-key range is E1 to G7.

Transpose

This changes the pitch of the zone, without changing its position on the keyboard. It changes the MIDI note numbers generated by the keys in the zone, without physically shifting the zone. The range is ±127 semitones. Since there are 12 semitones (or half steps) to an octave, you can transpose up or down over ten octaves. If you transpose out of the range of the active voice, however, no notes will sound; MIDI note numbers will transmit, but notes will not.

Note Map

Note Map lets you change the way notes are sent from the K2600. The default setting is **Linear**: all notes go out as played. Pressing the **Minus** button takes you to **Off**; no notes are sent, but controllers and other non-note data are.

Next comes **Inverse**, which turns the keyboard upside-down, with the highest key being A 0 and the lowest C 9. If you set Note Map to **Constant**, all of the keys on the keyboard will play the same note. The note defaults to C4, but you can change this with the **Transpose** parameter. This works well when you want the sound from a particular key to play with every note of another layer—for example, playing a ride cymbal with every note in a bass line.

Next are the alternating Note Maps, which let you divide the keyboard in some unique ways. If you are using two or more MIDI devices (including the K2600), you can expand polyphony by assigning each zone to a different alternating Note Map. For example, if you have two K2600s, you can assign two zones to each play the same program on a different K2600, thereby doubling polyphony.

To split a zone into one of two alternating Note Maps, set Note Map to **1 of 2**; now the zone plays on every second key, starting on C, but won’t play on any other keys. Set another zone to **2 of 2**, and this zone will play on every second key, starting on C$^{\#}$, thus covering the remaining keys.

Three- and four-zone alternating Note Maps work the same way, but cause each zone to play only on every third and every fourth key, respectively.

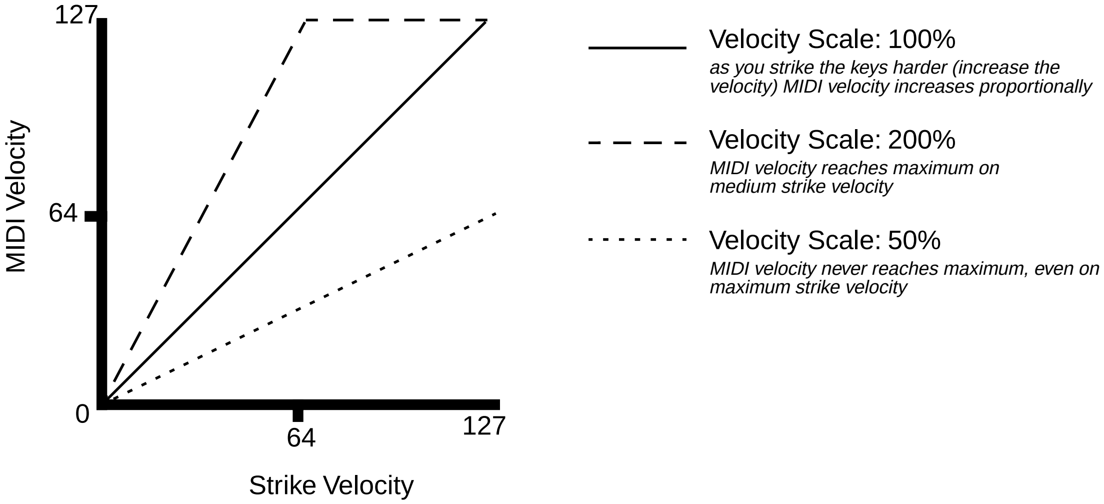
White and **Black** allow you to compensate for the differing velocity responses of white versus black keys on some keyboards. By choosing White for one zone and Black for another, you can modify velocity curves and limits independently for the white and black keys.

Note maps are also used to create drum patterns with the arpeggiator. Several of the ROM setups use this feature.

Velocity Scale (VelScale)

This lets you amplify or diminish velocity response. Normal response is **100%**. Higher values make the keyboard more sensitive (you don't need to play as hard to get higher MIDI velocities) while lower values make it less sensitive (playing harder doesn't change MIDI velocity as much). You can also set the scale to a negative number, in which case the velocity response is turned upside-down: playing harder produces a softer sound and vice versa. This is useful for creating velocity-based crossfades between zones. See the following section on Velocity Offset for ideas about negative scaling.

The following illustration shows what happens when you change Velocity Scale. Note that Velocity Scale is the only parameter changed in this example; the other parameters are set to their defaults (offset = **0**, curve = **linear**, min = **1**, max = **127**).

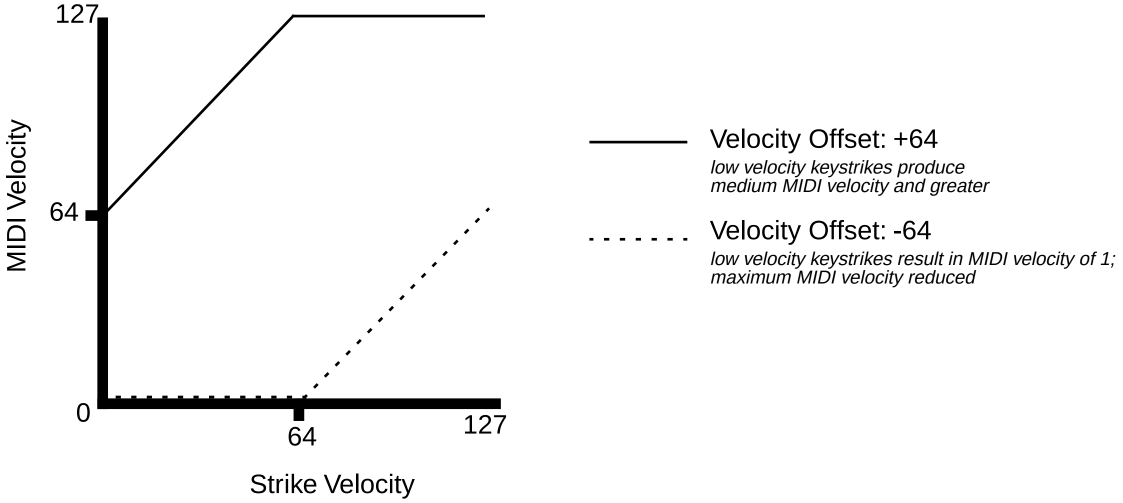


Velocity Offset

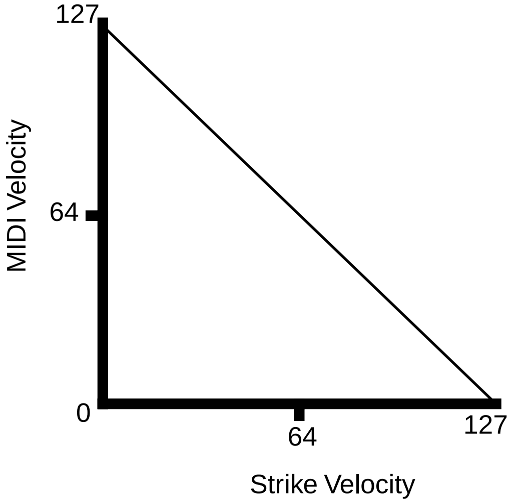
VelOffset also changes the response, but in a more direct way, by adding or subtracting a constant to the key velocity. For example, if this is set to **25** (assuming a scale of 100%), then 25 is added to the velocity of every keystroke, usually making the sound that much louder. The softest possible keystroke will have a value of 25, while a keystroke with velocity of 102 will produce the same sound as a note with velocity 127 (102+25=127). Negative values diminish the response: a setting of **-25** means the loudest velocity available will be 102, while any keystroke 25 or below will produce a velocity of 1 (a velocity value of zero has a special meaning in MIDI and cannot be used for Note Ons).

You can think of Scale as being a proportional change to the velocity, while Offset is a linear change. The maximum values for Offset are **±127**. The following illustration shows the effects of

Velocity Offset. Note that Velocity Offset is the only parameter changed in this example; the other parameters are set to their defaults (scale = **100%**, curve = **linear**, min = **1**, max = **127**).



Offset and Scale work together. If scaling takes the velocity out of the ballpark — for example, you want to set it to **300%** but that puts *all* of your notes at maximum velocity — using a negative offset, say around **-60**, can make it possible to still play at different volumes, although your curve will still be a lot steeper than normal. If you use a negative scaling, then you must use an offset: otherwise all of your velocities will end up as zeroes (well, ones actually, since a MIDI note-on with velocity zero is something else). So to get true inverse scaling (that is, minus 100%), you must set an offset of **127** to get the full range of velocities. Setting the offset to **127** and the scale to **-100%** produces a slope like this (which is the same as the reverse linear curve):



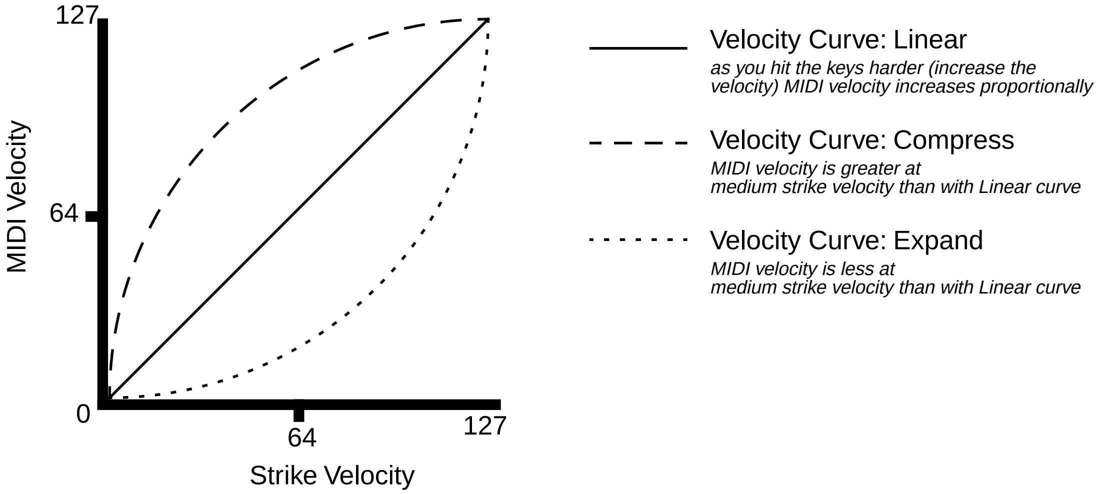
Note that Offset and Scale affect only *MIDI* velocities; that is, these parameters don’t change Velocity Tracking in the programs themselves. Therefore, some programs (such as organ sounds, which often have low VelTrk values) may respond only subtly to Offset and Scale, or not at all.

Velocity Curve (VelCurve)

VelCurve lets you taper the velocity response. The default setting is **Linear**, which means that the output velocity changes directly proportionally to the played velocity.

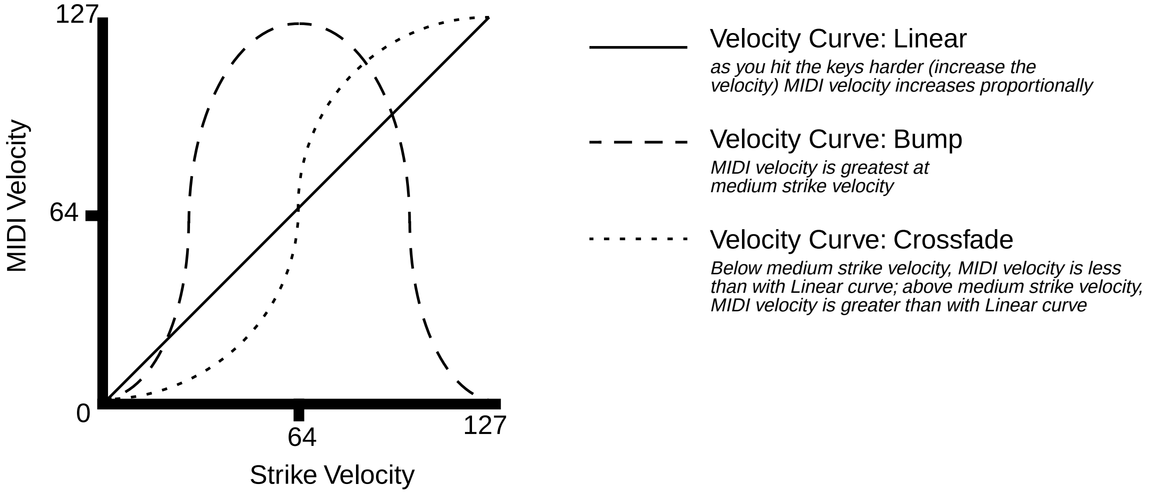
Expand produces a curve that is less steep than the linear curve at keystrike velocities below 64, and steeper than the linear curve at keystrike velocities above 64. In other words, when you're playing softly, you'll notice velocity differences less than with a linear curve, while when you're playing hard, you'll notice velocity differences more.

Compress produces a velocity curve that is the opposite of the expanded curve—that is, you'll notice velocity differences more when you're playing softly than when you're playing hard.

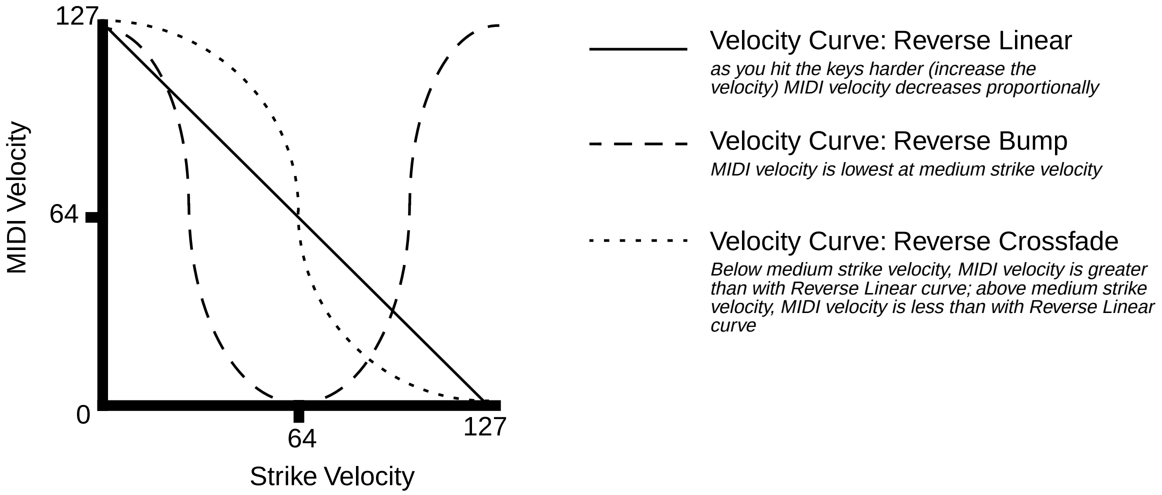


Crossfade (**Xfade**) is designed to be used in tandem with the Reverse Crossfade curve, enabling you to perform smooth crossfades between different programs.

Bump tapers velocity response to resemble a bell curve, so that notes are loudest when your keystrike velocity is 64. Notes get softer as the keystrike velocity approaches 0 or 127.



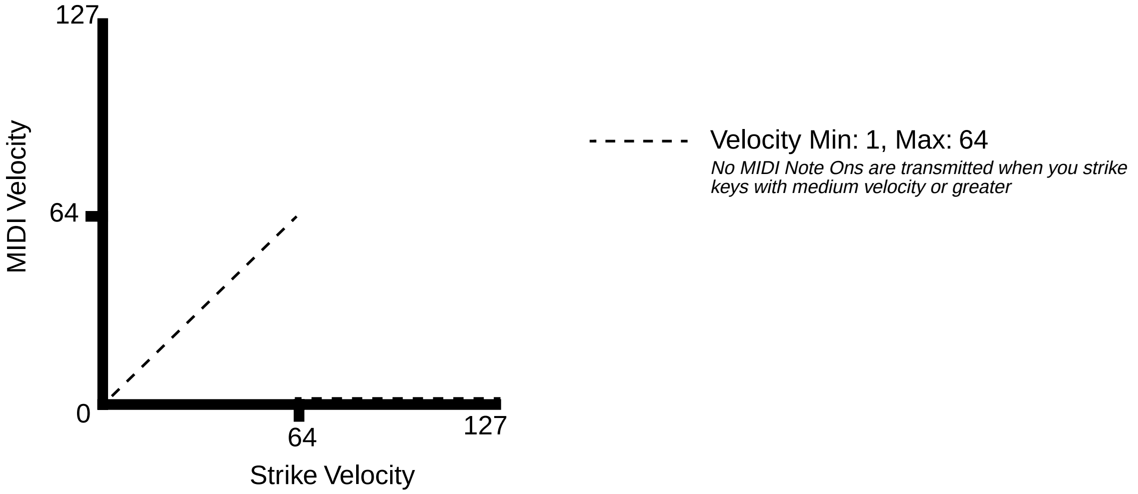
The next five velocity curves are Reverse Linear (**revrsLin**), Reverse Expand (**revrsExp**), Reverse Compress (**revrsCmp**), Reverse Crossfade (**revrsXfd**), and Reverse Bump (**revrsBmp**). These taper velocity in reverse of the five curves we just covered. For example, Reverse Linear’s response is such that striking a key harder will produce a lower volume, striking it softer will produce a higher volume, and so on. This provides a convenient way to achieve negative scaling, by letting you set one parameter instead of two.



Low Velocity (LoVel), HighVelocity (HiVel)

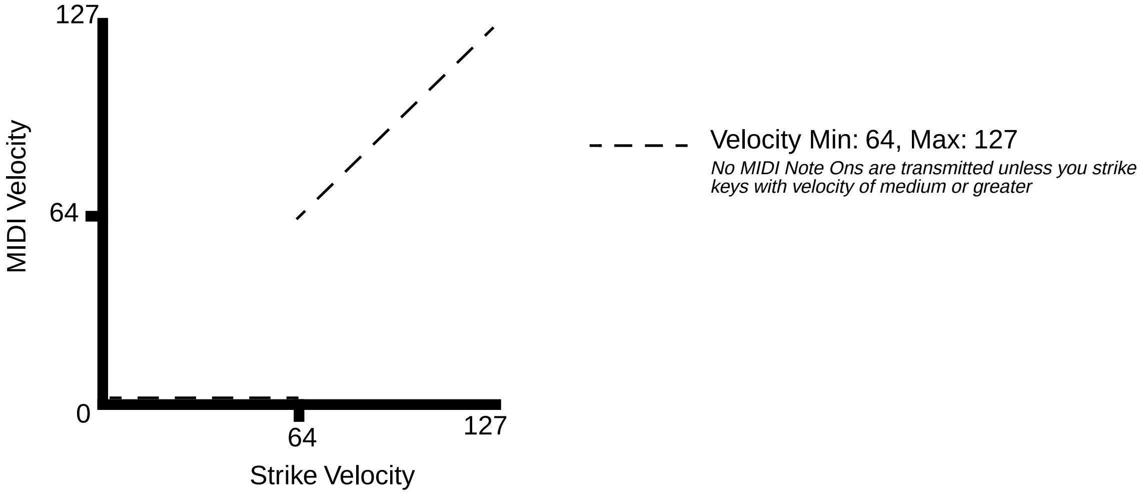
LoVel and HiVel set the minimum and maximum velocity limits that the current zone transmits. A keystroke in the current zone whose velocity — *after* it has been scaled and offset — is below the minimum does not generate a Note On. Neither does a keystroke whose velocity after processing is above the maximum. These parameters are useful for “velocity switching”— having a key play different sounds depending on how hard you strike it.

The values can be anywhere from **1** to **127**. As with other parameters, zones can overlap or be totally discrete, or be identical. Usually, LoVel will have a smaller value than HiVel, but as with LoKey and HiKey, you may also create a gap in velocity response, by setting negative ranges for velocity.



Velocity Min: 1, Max: 64

No MIDI Note Ons are transmitted when you strike keys with medium velocity or greater

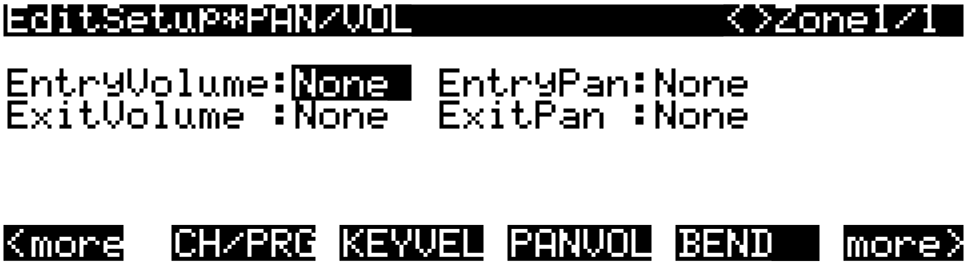


Velocity Min: 64, Max: 127

No MIDI Note Ons are transmitted unless you strike keys with velocity of medium or greater

The Pan/Volume (PAN/VOL) Page

By changing the parameters on this page, you can define how each zone sends MIDI volume and pan messages.



```
EditSetup*PAN/VOL                   <>zone1/1

EntryVolume:None  EntryPan:None
ExitVolume :None  ExitPan :None


<more  CH/PRG KEYVEL PANVOL BEND   more>
```

Parameter	Range of Values	Default
Entry Volume	None, 0-127	None
Exit Volume	None, 0-127	None
Entry Pan	None, 0-127	None
Exit Pan	None, 0-127	None

Entry Volume, Exit Volume

Entry Volume enables you to control the initial MIDI volume setting for each zone of the current setup. When you select a setup in Setup mode, the K2600 sends MIDI volume control (MIDI Controller 07) messages on each of the setup's MIDI channels, according to the value of the Entry Volume parameter. This sets the starting volume level for each zone, for any value other than **None**. Subsequent MIDI volume control signals sent to the setup's MIDI channels affect the volume normally.

When you exit the current setup, Exit Volume sends another MIDI Controller 07 message.

The setting of the Volume Lock parameter on the MIDI-mode CHANNELS page (see Chapter 10) determines whether this parameter has any effect.

Entry Pan, Exit Pan

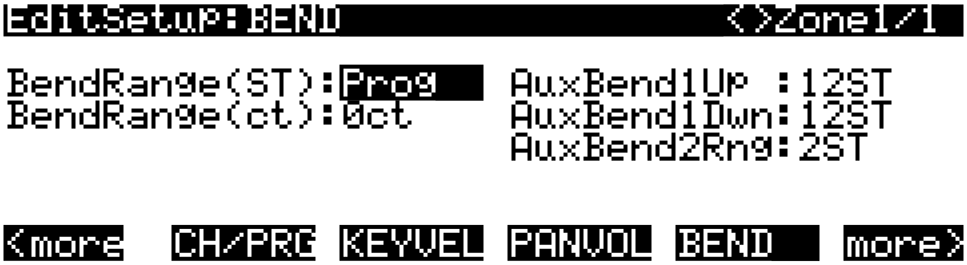
You can set entry and exit values for Pan as well. When you select a setup, the K2600 sends a MIDI pan control (MIDI Controller 10) message on each MIDI Channel in each zone; another MIDI pan control message is sent when you exit the setup. The Entry and Exit values for Pan are the same as those for Volume. There is also a Pan Lock parameter on the MIDI-mode CHANNELS page, which overrides the Setup Editor's Pan settings.

If you are trying to set the Pan and the program doesn't seem to be responding, you should check the Mode parameter on the OUTPUT page in the Program Editor. If it is set to **Fixed**, then the K2600 is ignoring the MIDI Pan message; setting Mode to **+MIDI** allows you to control the program's panning from the Setup Editor.

Most programs respond to pan messages on the next keystrike. This means that if you hold a note and change the pan, the current note will stay at its current position until you strike it again. However, a K2600 program that uses the PANNER algorithm will respond to real-time pan adjustments as well.

The BEND Page

The parameters on the BEND page define the bend ranges for each of the three types of pitch bend messages the K2600 can respond to.



```
EditSetup:BEND                      <>zone1/1

BendRange(ST):Prog   AuxBend1Up :12ST
BendRange(ct):0ct    AuxBend1Dwn:12ST
                     AuxBend2Rng:2ST

<more  CH/PRG KEYVEL PANVOL BEND   more>
```

Parameter	Range of Values	Default
Bend Range (semitones)	Prog, 0–60 semitones	Prog
Bend Range (cents)	±100 cents	0 cents
Aux Bend 1 Up	0–60 semitones	12 semitones
Aux Bend 1 Down	0–60 semitones	12 semitones
Aux Bend 2 Range	0–60 semitones	2 semitones

Bend Range (Semitones) and Bend Range (Cents)

Bend Range (semitones) sends a bend range message to an internal program or a MIDI device, telling it how to define subsequent pitch bend messages. Some programs may behave strangely when you change the Bend Range value, because they use FUNs (see Chapter 17) or DSP Functions (see Chapter 16) to affect the pitch wheel. In that case, you should either set the value of Bend Range to **Prog**, or edit the program itself.

Bend Range (cents) lets you fine tune the value for Bend Range (semitones). **100 cents** equals one semitone, or one half-step; you can set this parameter anywhere between **±100** cents.

Bend Range, in both semitones and cents, affects all controllers that are set to **BendUp** (in the default control setup, the PWhl parameter on the WHEEL page is assigned to **BendUp**). Physical controllers assigned to **BendDwn** are also affected by Bend Range, but they bend notes in the opposite direction from controllers assigned to **BendUp**. Any physical controller that uses the MIDI Control Source list can be assigned to **BendUp** or **BendDwn**. To keep things simple though, you will normally want to use **BendUp** as a Pitch Wheel destination and use **Aux Bend 1** and **Aux Bend 2** for other controllers, such as the Large and Small Ribbons.

Keep in mind that not all MIDI devices respond to Bend Range messages. The K2600, the K2500, and the PC88 support these messages, but with many older MIDI instruments (like the K2000 and K1000), you must set bend ranges on the devices themselves.

Changing programs sends a Bend Range message with the current program's values. So does pressing **Panic**, which is a quick way to reset your K2600 or MIDI slaves if you've used a controller to modulate the bend range.

Aux Bend 1Up and Aux Bend 1 Down

Like Bend Range, Aux Bend 1 defines the range for Pitch Bend messages, but does so for those physical controllers assigned to **MIDI 21**. There are two parameters related to AuxBend 1: an upward value (Aux Bend 1 Up) and a downward value (Aux Bend 1 Down). This means that you can set different values for upward and downward pitch-shifting. For example, you could get the Pitch Wheel to give you both vibrato and whammy-bar effects for the guitar program in a setup. In the zone that contains the guitar program, set AuxBend1Up to **2ST** and AuxBend1Dwn to **-12ST** (on the BEND page), then assign the PWhl parameter on the WHEEL page to a value of **AuxBend 1**. Now moving the Pitch Wheel up gives you a whole tone of upward bend, while moving it down gives you a full octave of downward bend.

In most factory setups, **AuxBend 1** is the assignment for the Large Ribbon.

Aux Bend 2 Range

The K2600 allows you to specify a third pitch bend range; this is called Aux Bend 2, and it defines the range for MIDI Controller 15 messages. For AuxBend 2, you can set only one range for both upward and downward pitch bending.

In most factory setups, **AuxBend 2** is the assignment for the Small Ribbon.

Controllers

Controller editing is one of the strongest aspects of the K2600's usefulness as the main controller for a sophisticated MIDI studio. In this section, we'll talk about two different types of "controllers" as they apply to the K2600. One is the *physical* controllers: the wheels, buttons, pedals, etc. that you move with your fingers or feet. The other is *MIDI* Controllers, which are MIDI commands sent by the K2600. For our purposes, "MIDI Controllers" includes the complete set of Controllers defined by the MIDI Specification, as well as pitchbend, aftertouch, and a few other useful MIDI commands. To fend off confusion, we'll refer to the K2600's physical controllers with a lower-case c, and MIDI Controllers with an upper-case C.

Any *MIDI* Controller can be used as the assignment for any *physical* controller (and for multiple physical controllers, as well). Or in other words, any physical controller like the Mod Wheel can be programmed to send any MIDI control signal. In addition, each controller in each setup zone can be tweaked just like keyboard velocity (or any other Setup-mode parameter). Although controller editing on the K2600 can be somewhat complex, it can also be very rewarding.

To get an idea of the expressive capabilities of Setup mode, explore the factory setups that come with the K2600.

The K2600's physical controllers include the following:

- The eight sliders (A, B, C, D, E, F, G, H) in the Assignable Controllers section
- The two Continuous Control Pedal jacks (CCPedal 1 and CC Pedal 2)
- The Large Ribbon, which you can define as a one- or three-section controller
- The Small Ribbon, which responds to both finger position and pressure
- The Pitch Wheel
- The Modulation Wheel (Mod Wheel)
- Keyboard aftertouch, or Mono Pressure (MPress)
- The Breath Controller jack (which is connected to Continuous Controller Pedal 2)
- The two Panel Switches, located above the Pitch and Mod Wheels
- The four Footswitch pedal jacks (1, 2, 3, and 4)

The following tables and illustration provide an overview of the physical controllers and their parameters.

Continuous Controllers

Physical Controller	Setup Editor Page	Parameter:Values
Sliders A, B, C, D, E, F, G, H	SLID, SLID/2	Destination: MIDI Control Source list
		Scale: -300% to 300%
		Add: -128 to 127
		Curve: Linear, Expand, Compress, Crossfade, Bump, Reverse Linear, Reverse Expand, Reverse Compress, Reverse Crossfade, Reverse Bump
		Entry Value: None, 0 - 127
		Exit Value: None, 0 - 127
CPedals 1 & 2 / Breath	CPEDAL	(same as Sliders)
Small Ribbon and Large Ribbon	RIBBON	(same as Sliders)
Pitch Wheel and Mod Wheel	WHEEL	(same as Sliders)
MPressure	PRESS	(same as Sliders)

Table 7-3 Continuous Controllers

Switch Controllers

Physical Controller	Setup Editor Page	Parameter:Values
Footswitches 1, 2, 3, and 4	FOOTSW	SwType: Toggle, Momentary, Note Toggle, Note Momentary
		Destination: MIDI Control Source list
		On Value: None, 0 - 127
		Off Value: None, 0 - 127
		Entry Value: None, Off, On
		Exit Value: None, Off, On
Panel Switches 1 and 2	SWITCH	(same as Footswitches)

Table 7-4 Switch Controllers

Keyboard Controllers

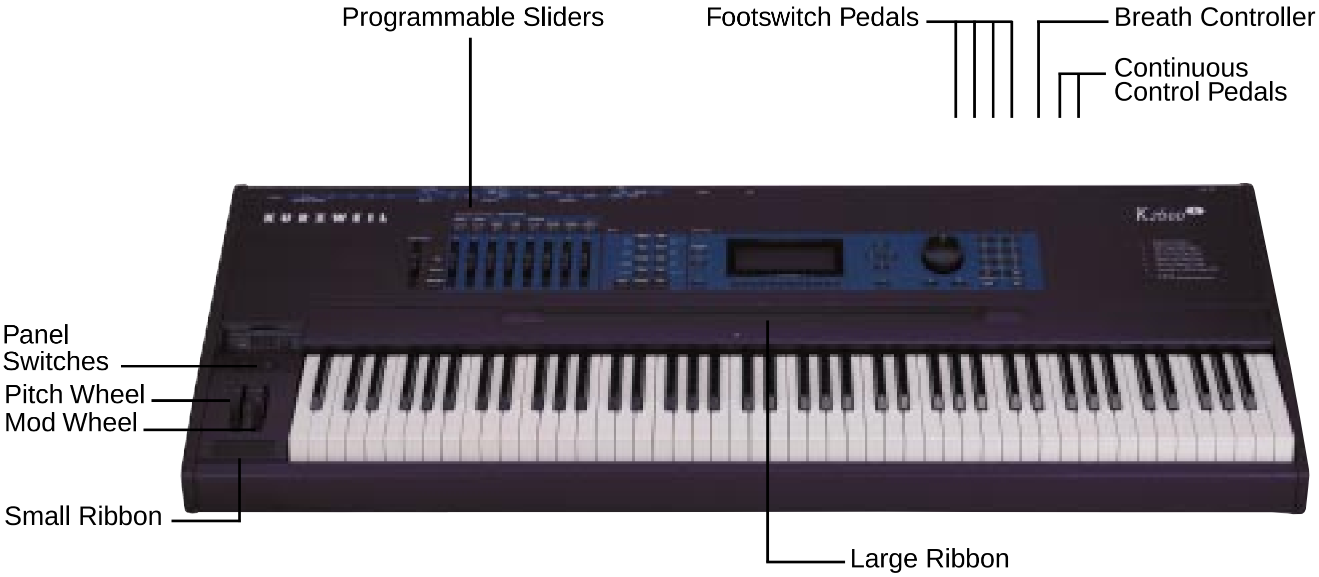


Figure 7-2 Setup-mode Controllers

After you’ve selected which zone and which physical controller to work with, use the Destination parameter to choose what this controller will do. Some of the controllers have default settings that are preprogrammed in all of the factory setups (of course, you can change them). You can select from the entire list of numbered MIDI controllers, as well as other MIDI commands and some “special functions.” To move through the list, you can use the Alpha wheel, or Intuitive Entry with any continuous controller, or call up the Controller’s number with the numeric buttonpad.

The MIDI Control Source List

Here are the available values for the Destination parameter for each controller, in scrolling order:

MIDI Controller Number	Corresponding Destination Name	Description
0	Off	Turns physical controller off
1	MWheel	Default Assignment for Mod Wheel
2	Breath	Default Assignment for CC Pedal 2
3	MIDI 03	
4	Foot	Default Assignment for CC Pedal 1
5	PortTim	Monophonic K2600 programs respond to this controller if portamento is turned on
6	Data	Almost all K2600 programs respond to this controller with DSP sweeps and effects
7	Volume	MIDI Volume
8	Balance	MIDI Balance
9	MIDI 09	
10	Pan	MIDI Pan–programs which use the PANNER algorithm will respond to real-time pan adjustments; all other programs will respond on the next note start
11	Express	MIDI Expression–an attenuator for fading in and out. It scales between minimum (0) and the current value of Volume
12–14	MIDI 12–14	MIDI Controllers 12–14
15	Aux Bend 2	Default destination for Small Ribbon
16	Ctl A	
17	Ctl B	
18	Ctl C	
19	Ctl D	
20	MIDI 20	
21	Aux Bend 1	Default destination for Large Ribbon
22–31	MIDI 22–31	MIDI Controllers 22-31. Almost all K2600 programs respond to MIDI 22–29 in one way or another. In the default control setup (97 ControlSetup) and in most other factory setups, Sliders B–H send MIDI 22–28, and Panel Switch 2 sends MIDI 29.
Special Function Controllers When you enter numbers 32–46 on the numeric buttonpad, you select one of the K2600’s Special Functions. Entering 47–63 sets a physical controller’s Destination to **Off**. Note that these are not MIDI Controller numbers; they’re the K2600’s internal global control numbers		
32	MPress	Default destination for Press and Small Ribbon Pressure (SmPrs)
33	BendUp	Default destination for Pitch Wheel
34	BendDwn	Physical controllers bend in opposite direction of BendUp

Table 7-5 MIDI Control Source List

MIDI Controller Number	Corresponding Destination Name	Description
35	Tempo	MIDI Clock tempo
36	MuteZn	Mute current zone
37	KeyNum	Key Number–sends a MIDI note with a velocity defined by the last value for KeyVel
38	KeyVel	Key Velocity
39	TapTempo	Enables you to tap a switch controller (like FtSw1–4 or PSw1–2) to change the K2600's internal tempo setting
40	ArpOrder	Change the order of arpeggiated notes
41	ArpBeats	Change the number of arpeggiated notes per beat
42	ArpShift	Change the amount of arpeggiator shift
43	ArpLimit	Change the arpeggiator shift limit
44	ArpLmtOp	Change the arpeggiator limit option
45	ArpVel	Change the attack velocity of arpeggiated notes
46	ArpDur	Change the duration of arpeggiated notes
47–63	Off	Reserved for future use
64	Sustain	Default destination for Footswitch 1
65	PortSw	Monophonic K2600 programs respond to this controller if portamento is turned on
66	SostPd	Default destination for Footswitch 2–holds notes that are currently down, but not notes played subsequently
67	SoftPd	Default destination for Footswitch 3– lowers the volume by a preset amount and may soften the timbre as well
68	LegatoSw	Forces mono playback
69	FrezPd	Envelopes freeze at current state
70–79	MIDI 70–79	MIDI Controllers 70–79
80	Ctl E	
81	Ctl F	
82	Ctl G	
83	Ctl H	
84–90	MIDI 84–90	MIDI Controllers 84–90
91	FX Depth	Controls wet/dry mix of effects
92–95	MIDI 92–95	MIDI Controllers 92–95
96	DataInc	Data Increment
97	DataDec	Data Decrement
98–101	MIDI 98–101	MIDI Controllers 98–101
102	Play/Stp	Alternately starts and stops sequencer playback
103	Rec/Stop	Alternately starts and stops sequencer recording
104–115	MIDI 104–115	MIDI Controllers 104–115
116	ArpSw	Arpeggiator On/Off

Table 7-5 MIDI Control Source List (Continued)

MIDI Controller Number	Corresponding Destination Name	Description
117	ArpVel	Arpeggiator Velocity
118	Latch2	Arpeggiator Latch 2
119	ArpLatch	Arpeggiator Latch
120	Panic2	Sends All Notes Off message
121	MIDI 121	Resets all controllers
122	Off	Turns physical controller off
123	Panic	Sends All Notes Off message
124, 125	MIDI 124, 125	MIDI Controllers 124 and 125
126	Mono On	
127	Poly On	

Table 7-5 MIDI Control Source List (Continued)

Continuous Controller Parameters

The continuous (physical) controllers are those that have a *range* of values: the two wheels, two ribbons, eight sliders, two Continuous Control pedals, and mono pressure (aftertouch). As the table on page 7-22 shows, all of them use the same parameters. Each parameter's function is described below.

Destination (Dest)

Use this parameter to select a destination from the MIDI Control Source list (see page 7-24).

Scale

After you've selected a continuous physical controller, you can modify the controller's response similarly to the ways you can modify velocity response. Refer to the graphs beginning on page 7-14 for illustrations of the velocity scaling parameters.

Scale lets you amplify or diminish the action of the controller. Full scale is 100%. Higher values will make the controller more sensitive, and lower values will make it less so. Setting the scale to a negative number makes the controller action work in reverse. As with velocity, you can use a controller to crossfade between two zones by setting the scaling for one zone positive and the other negative. Maximum scale values are **+300**% and **-300**%.

Offset (Add)

This adds or subtracts a constant to the controller, and at the same time sets minimum or maximum values (there's no need for separate Max and Min parameters). If the offset is **25**, the minimum value of the controller will be 25. If it is **-25** (and scale is **100**%) the first one-fifth of the controller's movement (25/127 = about 1/5) will send value of 0, and the maximum value of the controller will be 102 (= 127-25). As with velocity, Scale is a proportional change to the controller, while Offset is a linear change. The maximum values for Offset are **±127**.

Curve (Curv)

This lets you taper the controller response. The default setting is **Linear**, which means that the response follows a straight line as you move the controller.

Expand produces a curve that is less steep than the linear curve at keystrike velocities below 64, and steeper than the linear curve at keystrike velocities above 64. In other words, when you're playing softly, you'll notice velocity differences less than with a linear curve, while when you're playing hard, you'll notice velocity differences more.

Compress produces a velocity curve that is the opposite of the expanded curve—that is, you'll notice velocity differences more when you're playing softly than when you're playing hard.

Crossfade (**Xfade**) is designed to be used in tandem with the Reverse Crossfade curve, enabling you to perform smooth crossfades between different programs.

Bump tapers velocity response to resemble a bell-curve, making its response greatest at medium strike velocity, and progressively weaker as strike velocity increases or diminishes. If you start playing softly, then progressively louder, the response will increase from 1 to 64; if you keep playing harder, the sound will get softer.

The next five curves are Reverse Linear (**rLin**), Reverse Expand (**rExp**), Reverse Compress (**rCmp**), Reverse Crossfade (**rXfd**), and Reverse Bump (**rBmp**). These taper controller response in reverse of the first five curves. For example, Reverse Linear's response is such that moving the pitch wheel forward decrease pitch, moving it back will increase pitch, and so on. As with velocity, the reverse curves offer you a quick way to achieve negative scaling of physical controller response.

To get an idea of how these curves affect controller response, refer to the Velocity Curve charts, which begin on page 7-14.

Entry (Ent) and Exit Values

Entry value allows you to specify an initial value for a controller in a setup that will be sent whenever you select that setup. For example, if you want to make sure that all of the modulation in a zone is turned off when you select a setup, assign a physical controller to a destination of MIDI 01 (MWheel) and set Entry Value to **0**.

Entry values ignore the current position of the physical controller when the setup is selected. In fact, if the physical controller is above or below the entry value when the setup is selected (which it often is), moving the controller will have no effect until it is past its entry value. In the modulation example above, moving the assigned controller won't turn on any modulation until it's pushed all the way *down*, and then up again.

An entry value of **None** is quite different from a value of **0**. **None** means that there will be no initial controller command when the setup is selected, and any subsequent movement of the physical controller will be effective.

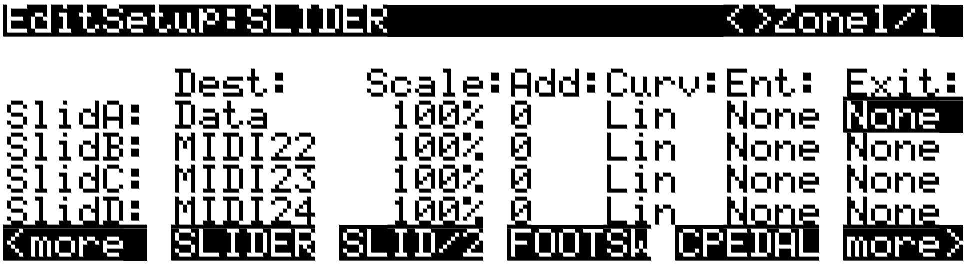
Exit Value tells the K2600 to send a value for that controller whenever you leave the setup, either by selecting another setup or by selecting a different mode altogether. It can be very useful when a controller is doing something to the sound, and you don't want that effect to continue after you leave the setup. For example, if you want to make sure a zone's pitch returns to normal whenever you leave a setup, you would set Exit Value to **64** for any controller whose Destination parameter is set to **BendUp**. Again, **None** means no command is sent.

The SLIDER and SLID/2 Pages

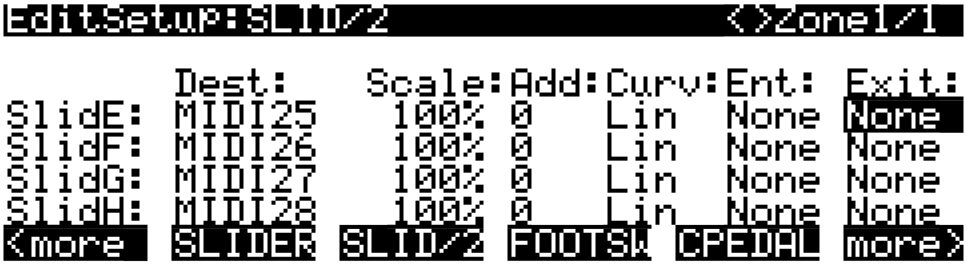
You can assign each of the K2600’s eight programmable sliders to a destination on each of the eight zones. Or, you can assign any combination of sliders to the same zone, allowing you tremendous flexibility. For example, you can assign Sliders A and B to modulate pitch and volume on Zone 1, then assign Slider C to control panning on Zones 2, 3, and 4.

The **SLIDER** button gives you access to Sliders A–D. Press the **SLID/2** button to program Sliders E–H.

Take a look at the parameters. Since all the continuous controllers work in a similar manner, you’ll find these same parameters on the CPEDAL, RIBBON, WHEEL, and PRESS pages.



```
EditSetup:SLIDER                    <>zone1/1

       Dest:    Scale:Add:Curv:Ent: Exit:
SlidA: Data       100% 0   Lin  None None
SlidB: MIDI22     100% 0   Lin  None None
SlidC: MIDI23     100% 0   Lin  None None
SlidD: MIDI24     100% 0   Lin  None None
<more  SLIDER SLID/2 FOOTSW CPEDAL more>
```



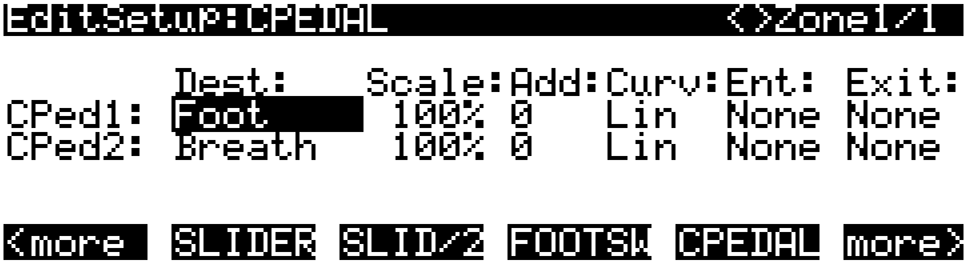
```
EditSetup:SLID/2                    <>zone1/1

       Dest:    Scale:Add:Curv:Ent: Exit:
SlidE: MIDI25     100% 0   Lin  None None
SlidF: MIDI26     100% 0   Lin  None None
SlidG: MIDI27     100% 0   Lin  None None
SlidH: MIDI28     100% 0   Lin  None None
<more  SLIDER SLID/2 FOOTSW CPEDAL more>
```

Parameter	Range of Values	Default
Destination (Slider A)	MIDI Control Source list	Data
Destination (Slider B)	MIDI Control Source list	MIDI 22
Destination (Slider C)	MIDI Control Source list	MIDI 23
Destination (Slider D)	MIDI Control Source list	MIDI 24
Destination (Slider E)	MIDI Control Source list	MIDI 25
Destination (Slider F)	MIDI Control Source list	MIDI 26
Destination (Slider G)	MIDI Control Source list	MIDI 27
Destination (Slider H)	MIDI Control Source list	MIDI 28
Scale	±300%	100%
Add	-128 to +127	0
Curve	Curve list (see text)	Lin
Entry Value	None, 0–127	None
Exit Value	None, 0–127	None

NOTE*: The FOOTSW page, and all the switch-controller pages, are described following the section called Switch Controller Parameters, which begins on page 7-32.*

The Continuous Control Pedal (CPEDAL) Page

If you look at the back of the instrument, you will see that there are jacks for plugging in two CC (Continuous Control) pedals. There is also a jack labeled Breath Controller. This jack is wired in parallel with CC Pedal 2; in other words, you can send MIDI 02 Breath (or any other MIDI or global Controller) with both CC Pedal 2 and an breath controller.



```
EditSetup:CPEDAL                  <>Zone1/1

       Dest:     Scale:Add:Curv:Ent:  Exit:
CPed1: Foot        100% 0   Lin  None None
CPed2: Breath      100% 0   Lin  None None

<more  SLIDER SLID/2 FOOTSW CPEDAL more>
```

Parameter	Range of Values	Default
Destination (CPed1)	MIDI Control Source list	Foot
Destination (CPed2)	MIDI Control Source list	Breath
Scale	±300%	100%
Add	-128 to +127	0
Curve	Curve list (see text)	Lin
Entry Value	None, 0–127	None
Exit Value	None, 0–127	None

The RIBBON Page

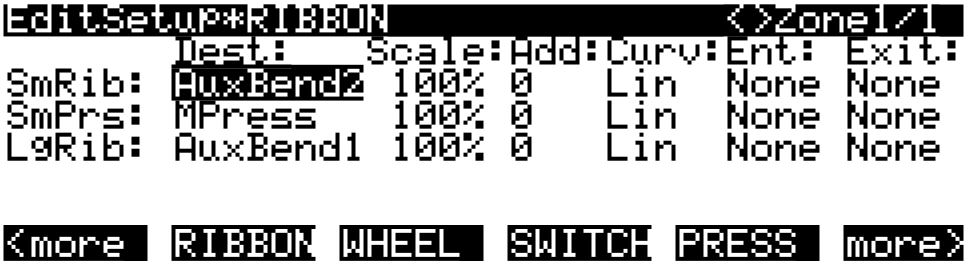
The RIBBON page lets you define controller assignments for the K2600’s two ribbon controllers. Each ribbon senses movement when you press on it and move your finger left or right; this creates numerous possibilities for controlling pitch, volume, panning, crossfades between zones, or any other uses you might imagine.

The Large Ribbon can be used as a single long controller, or it can be divided into three separate sections, each with its own controller assignments (this is done on the RIBCFG page). The two small arrows above the strip indicate the boundaries of the three sections. The large arrow above the ribbon points to the center of the ribbon, for when the ribbon is configured in one section.

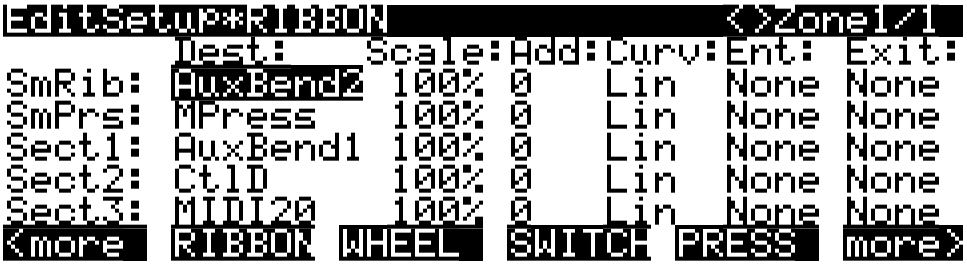
The Small Ribbon, located below the wheels, is both pressure- and motion-sensitive, so it will respond to both finger position and pressure.

To modify other ribbon parameters, go to the RIBCFG page, which is described on page 7-44.

The first of the following diagrams shows how the RIBBON page looks when the Large Ribbon is configured in one sections. The second shows the RIBBON page for a three-section Large Ribbon.



```
EditSetup*RIBBON                           <>Zone1/1
       Dest:     Scale:Add:Curv:Ent:  Exit:
SmRib: AuxBend2  100%  0   Lin  None  None
SmPrs: MPress    100%  0   Lin  None  None
LgRib: AuxBend1  100%  0   Lin  None  None



<more   RIBBON  WHEEL   SWITCH  PRESS   more>
```

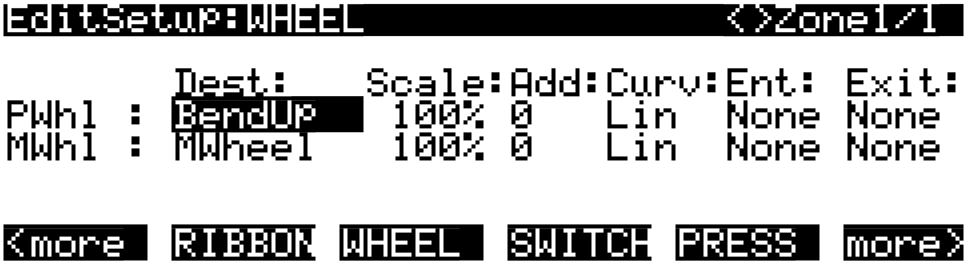


```
EditSetup*RIBBON                           <>Zone1/1
       Dest:     Scale:Add:Curv:Ent:  Exit:
SmRib: AuxBend2  100%  0   Lin  None  None
SmPrs: MPress    100%  0   Lin  None  None
Sect1: AuxBend1  100%  0   Lin  None  None
Sect2: CtlD      100%  0   Lin  None  None
Sect3: MIDI20    100%  0   Lin  None  None
<more   RIBBON  WHEEL   SWITCH  PRESS   more>
```

Parameter	Range of Values	Default
Destination (SmRib)	MIDI Control Source list	AuxBend2
Destination (SmPrs)	MIDI Control Source list	MPress
Destination (LgRib)	MIDI Control Source list	AuxBend1
Scale	±300%	100%
Add	-128 to +127	0
Curve	Curve list (see text)	Lin
Entry Value	None, 0–127	None
Exit Value	None, 0–127	None

The WHEEL Page

The two wheels are typical of what is found on many keyboards. The left one is normally used for pitch bend and springs back to center, while the right wheel is normally used as a standard Mod Wheel.



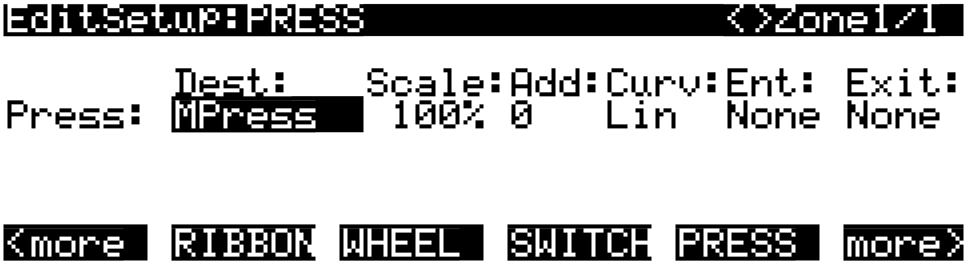
```
EditSetup:WHEEL                   <>zone1/1

       Dest:    Scale:Add:Curv:Ent: Exit:
PWhl : BendUp    100% 0   Lin  None None
MWhl : MWheel    100% 0   Lin  None None

<more  RIBBON  WHEEL  SWITCH  PRESS  more>
```

Parameter	Range of Values	Default
Destination (PWhl)	MIDI Control Source list	BendUp
Destination (MWhl)	MIDI Control Source list	MWheel
Scale	±300%	100%
Add	-128 to +127	0
Curve	Curve list	Lin
Entry Value	None, 0–127	None
Exit Value	None, 0–127	None

The Pressure (PRESS) Page

The K2600 features mono pressure, commonly called aftertouch on other keyboards.

A word about pressure: Key Range in a zone does *not* define which notes will generate pressure in that zone. If pressure is enabled in a zone, playing with aftertouch *anywhere* on the keyboard will produce data. For example, if Zone 1's Key Range is **C3-C5** and you play C2 and push down on the note, pressure messages will be sent from Zone 1. As with any other physical controller, however, you can disable pressure in any zone, or scale it or offset it differently in the various zones. It might help to think of pressure as an extra wheel—wheels operate in a zone regardless of Key Range, and so does pressure.



```
EditSetup:PRESS                   <>zone1/1

       Dest:    Scale:Add:Curv:Ent: Exit:
Press: MPress    100% 0   Lin  None None


<more  RIBBON  WHEEL  SWITCH  PRESS  more>
```

Parameter	Range of Values	Default
Destination	MIDI Control Source list	MPress
Scale	±300%	100%
Add	-128 to +127	0
Curve	Curve list (see text)	Lin
Entry Value	None, 0–127	None
Exit Value	None, 0–127	None

Switch Controller Parameters

Switch (physical) controllers have only two states: on and off. The K2600 switch controllers are:

- Panel Switches 1 and 2 (PSw1 and PSw2)
- Footswitch Pedals 1, 2, 3, and 4 (FtSw1, FtSw2, FtSw3, and FtSw4)

Note that buttons **1–8** above the sliders are dedicated to zone status and muting, as well as sequencer muting, and are *not* assignable controllers.

Switch Type (SwType)

The parameters for Switch controllers are slightly different from those for continuous controllers. The first parameter is Switch Type (SwType). The choices available are **Momentary** in which a switch's action lasts only as long as you are pushing it, and **Toggle**, in which the switch's action lasts until you press it again. You can also assign the switch controller to send a note, using **Note Momentary** and **Note Toggle**.

Momentary mode is used for functions like sustain or portamento, while Toggle mode is used for functions such as arpeggiator on/off. The buttons show which mode they are in by the behavior of their lights: if a button is in Momentary mode, its light glows only as long as you are holding it, while if it is in Toggle mode, the light stays on until you press it again. Bear in mind that button assignments are independent per zone, and since there's just a single light per button, the light shows the state of the button only for the current zone. When you press the button, however, it executes its assignments for all zones that use that button.

Destination

Destination determines what MIDI Controller or other message will be sent when the switch is on—that is, either pressed and held or toggled from the off position. The list of available controllers on page 7-24 is the same as for the continuous controllers, and can be accessed the same way.

If you set SwType to **Note Momentary** or **Note Toggle**, the values for Destination change to display MIDI note values. This can be pretty interesting if you want to do some unusual things with the switch controllers. For example, choose a drum program for the current zone, then set the destination of Footswitch 1 to **Note Momentary**. Now set the value of Destination to a key with a kick drum or a closed hi-hat sound, and impress your drummer friends. Or, you could set SwType to **Note Toggle** and use the Footswitch to start and stop a sampled groove, freeing up your hands to play a different program.

On Value

On Value sets the value of the Controller when the switch is on. In the case of conventionally-switched functions, such as sustain, the On Value will be **127**. (For example, the default for Switch Pedal 1 (FtSw1) is Controller 64 **— Sustain —** with an On Value of **127**.) However, you might want to use a button or pedal as a “soft” switch, in which case you might set Destination to **7** (Volume) and On Value to **50**. Destination can also be set to **Off**, so that turning on the switch has no effect at all in this zone. This can be useful when you are using one switch for multiple functions in different zones.

If Switch Type is set to **Note Momentary** or **Note Toggle**, the On Value will define the velocity of the note message that gets sent.

Off Value

Off Value is the value of the Controller when the switch is off. The default value is **0**. You might want to change this, as in the “soft switch” example above: in order to bring the zone up to full volume when you release the pedal, set Off Value to **127**.

If Switch Type is set to **Note Momentary** or **Note Toggle**, be sure to leave the Off value at **0**; this will send a note off message when you turn the switch off, preventing stuck notes from occurring.

Entry (Ent) and Exit States

Entry State determines whether an initial setting for the switch will be sent when the setup is selected. There are three choices: **None** (no change), **Off** (the Off value), and **On** (the On value). With a Panel Switch button, if the Entry State is **On**, the button will light as soon as you select the setup.

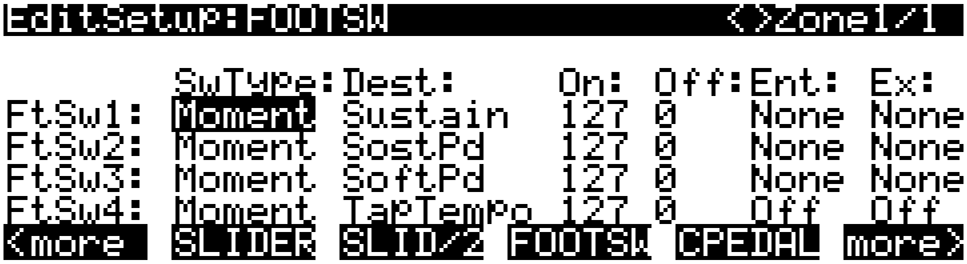
Exit State similarly determines whether a setting for the switch will be sent when you leave the setup, either for another setup or for Program mode. The same three choices (**On**, **Off**, and **None**) are available. This is very useful for turning off sustains when changing setups.

The Footswitch (FOOTSW) Page

On the back of the instrument, there are four jacks for Footswitch pedals.

All the foot switches in the default control setup are set to a switch type that doesn’t generate a note, and consequently the range of values for their respective Destination parameters is the

MIDI Control Source list. If you set a footswitch’s switch type to Note Momentary or Note Toggle, the range of values for its Destination parameter is **C -1–G 9**.

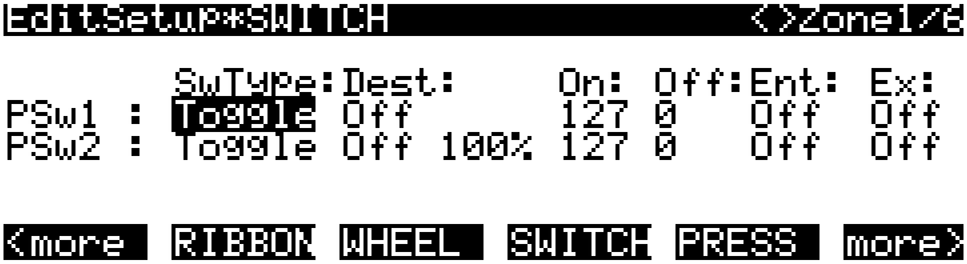


```
EditSetup:FOOTSW                      <>zone1/1

       SwType:Dest:      On: Off:Ent:  Ex:
FtSw1: Moment Sustain    127 0   None  None
FtSw2: Moment SostPd     127 0   None  None
FtSw3: Moment SoftPd     127 0   None  None
FtSw4: Moment TapTempo   127 0   Off   Off
<more  SLIDER SLID/2 FOOTSW CPEDAL more>
```

Parameter	Range of Values	Default
Switch Type	Momentary, Toggle, Note Momentary, Note Toggle	FtSw1: Moment
		FtSw2: Moment
		FtSw3: Moment
		FtSw4: Moment
Destination	MIDI Control Source list when SwType is Momentary and Toggle); C –1 to G9 when SwType is Note Momentary or Note Toggle	FtSw1: Sustain
		FtSw2: SostPd
		FtSw3: SoftPd
		FtSw4: TapTempo
On	0–127	127
Off	0–127	0
Entry Value	None, 0–127	None
Exit Value	None, 0–127	None

The SWITCH Page

The K2600 keyboard offers two Panel switches, located above the pitch and mod wheels. Each of these functions exactly like the Footswitches; you can choose between momentary and toggle switches, or you can use each one to trigger a note.



```
EditSetup*SWITCH                       <>zone1/6

       SwType:Dest:      On: Off:Ent:  Ex:
PSw1 : Toggle Off        127 0   Off   Off
PSw2 : Toggle Off 100%   127 0   Off   Off


<more  RIBBON WHEEL  SWITCH PRESS  more>
```

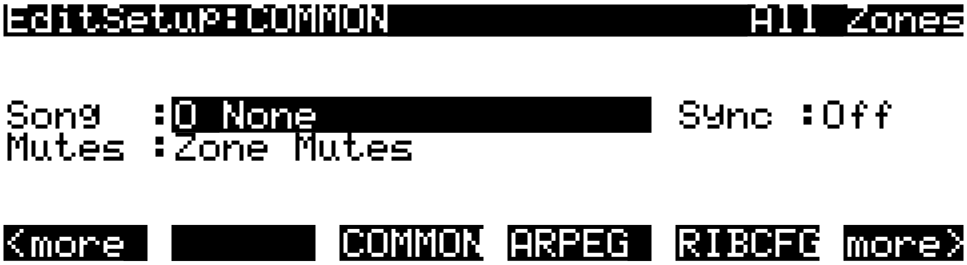
Parameter	Range of Values	Default
Switch Type	Momentary, Toggle, Note Momentary, Note Toggle	PSw1: Toggle
		PSw2: Toggle
Destination	MIDI Control Source list or C –1 to G9	PSw1: ArpSw
		PSw2: MIDI 29
On	0–127	127
Off	0–127	0
Entry Value	None, 0–127	PSw1: None
		PSw2: Off
Exit Value	None, 0–127	Off

The KDFX and FXMOD Pages

These are the same seven effects-control pages that we discussed in Chapter 6 (beginning on page 6-44). They enable you to define the studio and FXMods for all zones in the setup. As long as the FX Mode parameter is set to **Auto** or **Setup**, the values you set on these pages apply to all zones in the setup. If FX Mode is **Master**, all programs and setups use the studio specified on the Effects-mode page.

The COMMON Page

The COMMON page determines what song, if any, becomes current when you select a setup, and how song playback is synchronized. It also sets the operational mode for the buttons above the programmable sliders.



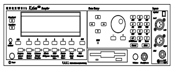
```
EditSetup:COMMON                    All Zones

Song  :0 None                Sync :Off
Mutes :Zone Mutes

<more            COMMON  ARPEG   RIBCFG  more>
```

Parameter	Range of Values	Default
Song	List of existing songs	0 None
Sync:	Off, On	Off
Mutes	Zone Mutes, KB3 Control	Zone Mutes

Song

You can link a song to the current setup, using the Song parameter. Select any existing song (or **0 None** if you don’t want a song associated with the setup). The song **1 New Song** is a blank template that contains no notes (unless you’ve saved **1 New Song** without renaming it).

Press **Edit** while the Song parameter is highlighted to use the Song Editor. See Chapter 12 for more information on the Song Editor's functions.



When the current setup has a song linked to it, the **Stop**, **Play/Pause**, or **Record** buttons enable you to start and stop both Recording and Playback of the current song while you're in Setup mode. For rack-mount model owners, pressing the **Up/Down** cursor buttons toggles between Play and Pause, without resetting the Locate point. Pressing the **Left/Right** buttons toggles between Start and Stop, resetting Locate to 1:1. These double-button presses work in any mode.

If you are planning to use a particular song with a setup, there are some important things to keep in mind. First, every track of your song and every setup zone should be on different MIDI channels. Secondly, set the FX Mode to **Auto** and the FX Channel to **Current** on the Effects-mode page; this way, when you play a song from within Setup mode, you'll hear the effect assigned to that song.

Once you've chosen a song on the COMMON page (and worked out any MIDI conflicts), you can use Setup mode to do some pretty cool things. If you want to start the song with a Footswitch, for example, you can assign the Destination parameter of any Footswitch to **Play/Stp** (MIDI 102). Because of the nature of this special function controller, set SwType to **Momentary** rather than **Toggle**.

Here's another great feature: by linking songs with setups, you can select songs from Quick Access mode. By assigning different songs to different setups, and then assigning those setups in one Quick Access bank, you can select one of up to ten songs with a single button press.

Sync

Setting this parameter to a value of **On** enables you to do two things during song playback:

- When you change the current song, it finishes before the newly-selected song starts
- When using Note Ons (keystrikes or MIDI input) to retrigger and transpose steps in an arrangement, you can trigger a Note On at any time during the current bar, to retrigger the step at the beginning of the next bar

When Sync is **Off**, changing a song during playback immediately starts the new song, and Note Ons immediately retrigger or transpose.

Mutes

The Mutes parameter gives you manual control over the behavior of the buttons above the programmable sliders. There's really only one case in which you'd need to worry about this parameter: when you have a setup that contains both VAST programs and KB3 programs, and you want the buttons to control KB3 features.

By default, the K2600 uses the **Mute** buttons (the buttons above the sliders on keyboard models) to control the muting and unmuting of zones—which means you don't have real-time control over any KB3 features. Changes the Mutes parameter to KB3 Control to use the **Mute** buttons as real-time KB3 controllers. The white labeling above the buttons describes their functions.

The Arpeggiator (ARPEG) Page

The Arpeggiator takes input from the K2600 keyboard (or via MIDI) and turns it into a constant rhythmic pattern. You can control the speed and nature of the pattern in real time. The Arpeggiator resembles what were called “sequencers” on old analog synths—playing a finite series of notes repeatedly, with changes in the series controlled by the notes you play. The Arpeggiator can affect both the K2600 and external MIDI instruments. The notes produced by the Arpeggiator in a given zone go to all of that zone’s destinations: local, MIDI, or both.

The concept behind the K2600’s Arpeggiator is fairly simple, although the options are extensive. You might think of it as a “note processor,” generating complex output from relatively modest input. You can select any number of notes for the input, and tell the Arpeggiator to recognize and remember them. This is called “latching” the notes. The Arpeggiator then processes them by playing them repeatedly, and/or transposing them up and down the keyboard. You have control over several processing parameters: tempo, velocity, order, duration, transposition, orchestration, and whether the intervals between notes are filled chromatically. You can also tell the Arpeggiator how to deal with new information coming from the keyboard. The settings you define on the ARPEG page apply to all zones for which arpeggiation is activated; you can program each zone individually to respond to or ignore the setup’s arpeggiation values.

Figure 7-3 shows the processing order the K2600 uses to create arpeggios from your input:

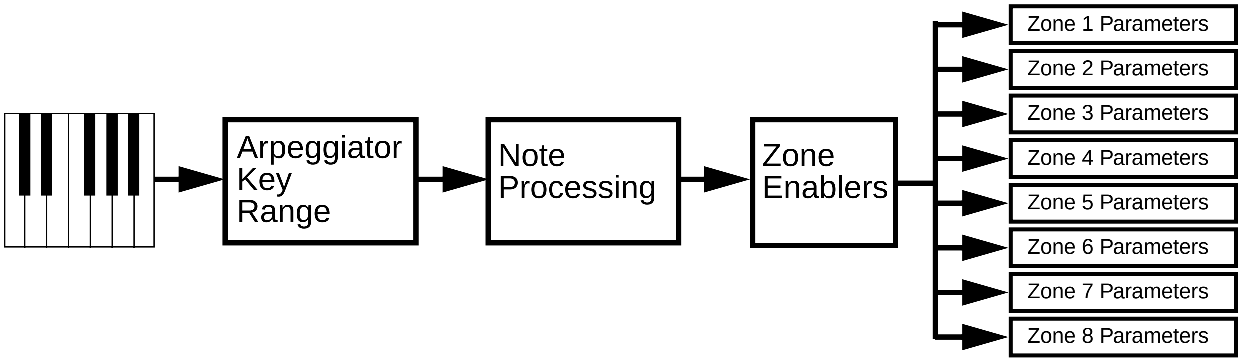
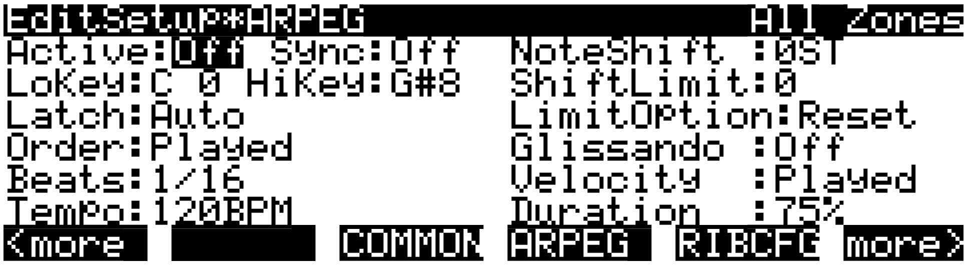


Figure 7-3 Arpeggiator Processing Sequence



```
EditSetup*ARPEG                 All Zones
Active:Off Sync:Off  NoteShift :0ST
LoKey:C 0 HiKey:G#8  ShiftLimit:0
Latch:Auto           LimitOption:Reset
Order:Played         Glissando :Off
Beats:1/16           Velocity  :Played
Tempo:120BPM         Duration  :75%
<more          COMMON ARPEG  RIBCFG more>
```

Parameter	Range of Values	Default
Active	On/Off	Off
Sync	On/Off	Off
Low Key	C –1 to G9	C0
High Key	C –1 to G9	C#8
Latch	Latch list (Auto, Keys, etc.)	Auto
Order	Order list (Played, Up, etc.)	Played
Beats	1/1 to 1/384	1/16
Tempo	1 to 255 beats/minute	120 beats/minute
Note Shift	±88 semitones	0 semitones
Shift Limit	0 to 88	0
Limit Option	Limit List	Reset
Glissando	On/Off	Off
Velocity	List	Played
Duration	1% to 100%	75%

Active

The first parameter on the Arpeggiator menu is Active — is the Arpeggiator on or off for this setup? This can be switched from here, or for convenience in live performance, it can be switched using MIDI Controller number 116. This can either be assigned as the destination of a K2600 physical controller, or it can come from an external MIDI source.

Turning Active on affects zones whose ZoneArpeg values are also set to **On**. By setting the ZoneArpeg parameter (on the CH/PRG page) to **Off** or **On** in the individual zones of a setup, you can choose which zones will be controlled by the Arpeggiator when it is on.

Sync

When Sync is set to **On**, the arpeggiator plays latched notes exactly on the next scheduled beat, which is a function of the Beats parameter and the Song-mode clock. For example, if Beats is 1/8, the arpeggiator plays eighth notes, exactly on each beat and half beat of the current song. When Sync is set to **Off**, latched notes still play at the same *tempo* as the song, but they don't play exactly on the beat.

Low Key (LoKey) and High Key (HiKey)

The Arpeggiator processes notes within the range of these parameters. Notes outside the specified range play normally, and do not become part of the arpeggiation sequence. Set the LoKey and HiKey parameters using the data entry wheel or buttons.

Latch

Latch determines how the Arpeggiator responds to notes when they are triggered.

Keys means that the Arpeggiator plays only while you are holding one or more keys down (or note triggers on). As you play different notes, they get added to the Arpeggiator, and as you release notes, they get taken out. If you play notes faster than the Arpeggiator's current tempo, each subsequent note will be added to the arpeggiation at the next division of a beat. This can cause a lag between the time you play the note and the time you hear it in the arpeggiation.

In the next three modes, the Arpeggiator latches notes only when MIDI Controller 119 (ArpLatch) sends a value of On (64 or higher). An easy way to experiment with these modes is to assign the Mod Wheel to send MIDI 119 (go to the WHEEL page, and set the value of MWhl to **ArpLatch** by pressing **1**, **1**, **9**, **Enter** on the alphanumeric buttonpad).

In **Overplay** mode, the Arpeggiator latches any notes that are being held when ArpLatch goes on, and continues playing them, even after you let them go, until ArpLatch off. Any notes that you play after ArpLatch is already on do not get arpeggiated, even if they're in the arpeggiation range.

Arpeggiation is similar: any notes held when ArpLatch goes on are latched and arpeggiated, and keep going until ArpLatch goes off. Any notes you play outside the arpeggiation range play normally. Notes that you play inside the arpeggiation range do not play normally, but if you hold them on, they become part of the arpeggiation. They drop out of the arpeggiation as soon as you release them.

Like Overplay and Arpeggiation, **Add** means that all notes being held when ArpLatch goes on get latched, and keep playing until ArpLatch goes off (even if you've released the notes). Any notes you play after ArpLatch is already on also get latched.

Auto is independent of ArpLatch; every note you play is automatically latched, and the Arpeggiator runs as long as you hold at least one arpeggiated note. As long as you keep holding on at least one note (it doesn't have to be the same note the whole time), every note you play in the arpeggiation range gets latched.

Pedals is sort of a combination of Keys, Add, and Overplay modes. It relies on both ArpLatch (MIDI 119) and Latch2 (MIDI 118). If neither latch controller is on, notes will arpeggiate only while you are holding down keys (similar to Keys mode). If you activate Controller 119, the keys currently held down will latch, and any additional keys played while Controller 119 is on will also latch (similar to Add mode). When Controller 119 is off, any keys that are not currently held down will be removed from the arpeggiation. If you activate Controller 118, keys currently held down will latch, and any additional keys played while Controller 118 is on will play normally (similar to Overplay mode). This mode is called Pedals mode because you might want to assign Switch Pedal 1 to **Controller 119** (Latch 1) and Switch Pedal 2 to **Controller 118** (Latch 2) to make the pedals function similarly to sustain and sostenuto pedals.

Autohold is similar to Auto. Holding at least one arpeggiated note on and playing other notes latches those notes. Unlike in Auto mode, if you stop holding at least one arpeggiated note on, the arpeggiation continues playing (although you can't latch any more notes). In this case, if you strike another key within the setup's arpeggiation range, you start a new arpeggiation sequence.

Autohold is useful for arpeggiating chords: when you play a chord, it gets latched, and continues arpeggiating after you release the chord. When you play another chord, the previous chord gets unlatched, and the new one gets latched.

You can use the **Panic** soft button or the **Stop** button to stop arpeggiation at any time.

Order

This parameter determines the order in which the K2600 plays arpeggiated notes. **Played** causes them to play back in the chronological order in which you played and latched them. **Up** means that notes play in ascending pitch order, regardless of their chronological order. **Down** means descending pitch order. **Up/Down** causes notes to play from lowest pitch to highest, then from highest pitch to lowest, repeating the cycle until you stop the arpeggiation. The notes at the very top and very bottom only play once. **Up/Dwn Repeat** is similar to **Up/Down**, except that the notes at the top and bottom play *twice* (repeat) when the Arpeggiator reverses direction.

Random plays the currently-latched notes in completely random order. **Shuffle** plays them at random, but keeps track of the notes so that no note repeats until all of the others have played. **Walk** is a “random walk” order: each successive note is either the next or previous note (in chronological order). For example, suppose you’ve latched four notes—G 4, B 4, D 5, and F 5—in that order. The first note the Arpeggiator plays is the G 4. The second note will be either B 4 (the next note chronologically), or F 5 (the “previous” note chronologically—that is, the last latched note). If the second note is B 4, the third note will be either D 5 or G 4. If the second note is F 5, the third note will be either G 4 or D 5.

Simultaneous means that each note you play repeats in time with the Tempo value, sort of like a digital delay with no decay. If you play a C and hold it while you play an E and a G, the Arpeggiator will play all three notes at the same time and at the same tempo. This is the only Order setting that allows two or more notes to sound simultaneously as intervals or chords. Note that **Simultaneous** works well with Note Shift and Shift limit.

Beats

The Beats parameter sets the number of notes per beat. The tempo is based on quarter notes. Therefore, if you set it to **1/4**, you will get one note per beat of the clock. At **1/16**, you will get 4 notes per beat. You can go all the way to 96 notes per beat (**1/384**), but at most tempos, divisions smaller than 1/64 will sound pretty much the same.

Tempo

This sets the tempo, in beats per minute (bpm), at which the Arpeggiator plays when activated. Tempo is tied to the K2600’s internal MIDI clock, as is song playback. Consequently, if a song is playing when you activate the Arpeggiator, the song’s tempo defines the Arpeggiator’s tempo, overriding the Arpeggiator’s settings. Inversely, if you activate the Arpeggiator, then start a song, the Arpeggiator’s Tempo setting defines the song’s tempo.

You can change Tempo in real time by changing the assignment of any of the K2600’s physical controllers in the setup to **Tempo**. The range is **1** to **255 bpm**.

Note Shift

You can tell the Arpeggiator to transpose all of the currently-latched notes each time it plays through them. Note Shift determines how much transposition will occur for each cycle of notes. For example, if you have latched C4 and F4, and you assign a Note Shift of 2, the Arpeggiator will play C4, F4, D4, G4, E4, A4, and so on until it reaches the Shift Limit. The values can range from **-88** to **88**, with **0** (the default) being no transposition.

Shift Limit

Shift Limit determines how far up or down the Arpeggiator shifts from the original note. The minimum value is **0**, and the maximum is **88**. When the Arpeggiator reaches the limit, the Arpeggiator responds according to the setting for the Limit Option parameter.

Limit Option

This parameter determines what the Arpeggiator does when it has shifted the currently latched notes up (or down) to the shift limit. **Stop** causes the Arpeggiator to stop when it reaches the shift limit. **Reset** causes the Arpeggiator to return to its original pitch and repeat the latched cycle of notes, transposing each cycle according to the settings for Note Shift and Shift Limit. If the limit allows the notes to go out of MIDI range (for example, if you set Shift to **12**, set the limit to **80**, and play C4), then those "ghost" notes don't sound, but they take up rhythmic space: the Arpeggiator waits for the cycle to play itself out before starting over.

Unipolar means that after playing up to the shift limit, the Arpeggiator begins shifting notes in the opposite direction, until it reaches the original pitch, where it reverses again. To determine the next note when it reaches the shift limit, the Arpeggiator calculates the interval between the shift limit and what the next note would be if the shift limit weren't there. It then plays the note that is the calculated interval lower than the last note before the shift limit. The same thing happens in reverse when the arpeggiated notes get back down to the original pitch. The following table makes this easier to visualize by showing the result of arpeggiating one note (C4) in Unipolar mode, with Note Shift set to **3 ST** and various values for Shift Limit.

Shift Limit	Resulting Arpeggiation (When LimitOption is Unipolar)			Comment
	Up	Down	Up	
6 ST (F#4)	C4, D#4, F#4,	D#4, C4	D#4, …	Same notes play in both directions when Shift Limit is a multiple of Note Shift
7 ST (G4)	C4, D#4, F#4,	E4, C#4,	D#4, …	Last upward note before shift limit is F#4, next upward note would be A4, which is 2 ST from shift limit (G4); therefore first downward note is E4 (2 ST below last upward note)
8 ST (G#4)	C4, D#4, F#4,	F4, D4,	D#4, …	A4 is 1 ST from shift limit, therefore first downward note is F4 (1 ST lower than last upward note)
9 ST (A4)	C4, D#4, F#4, A4	F#4, D#4, C4,	D#4, …	All symmetrical again; now A4 is within shift limit
10 ST (A#4)	C4, D#4, F#4, A4,	G4, E4, C#4,	D#4, …	Next upward note would be C5, which is 2 ST from shift limit
11 ST (B4)	C4, D#4, F#4, A4,	G#4, F4, D4,	D#4, …	C5 is 1 ST from shift limit
12 ST (C5)	C4, D#4, F#4, A4, C5,	A4, F#4, D#4, C4,	D#4, …	Symmetrical again, including C5

Bipolar starts out the same way as **Unipolar**, but during downward note shifting, it continues past the original pitch until it hits the shift limit in the *opposite* direction, where it reverses again.

Flt Reset adds a bit of apparent randomness to the process. **Flt** stands for "Float," and it means that when the Arpeggiator reaches the shift limit, it resets—but not to its original pitch as with plain Reset. Like Unipolar and Bipolar, it looks at the first note that would exceed the shift limit, and calculates the interval between that note and the shift limit. It then restarts the cycle of latched notes, transposing the entire cycle by the interval it just calculated, then shifting each subsequent cycle by the value of Note Shift, until it reaches the shift limit again.

Here's a very simple example. Suppose that the only note in the Arpeggiator cycle is C4, Note Shift is **4** (a third), and Shift Limit is **7** (so notes won't get shifted above G4). The Arpeggiator plays C4, then E4. The next note should be G#4, but that's above the shift limit—so the K2600 calculates the difference between that G#4 and the shift limit (G4): one semitone. It adds that difference to the original starting note (C4) and plays that note next—C$^{\#}$4. The next note (F4) is within the shift limit, but the next note (A4) isn't, so it gets translated into D4—and so on.

Flt Uni uses the same concept and applies it to Unipolar mode: when the Arpeggiator reaches the shift limit, it calculates the difference between the next note and the limit, and transposes the next cycle of notes down by that interval, then shifts each subsequent cycle down until it reaches the original pitch. **Flt Bipl** is similar to **Flt Uni**, but the downward shift limit isn't the original pitch, it's the negative of the Shift Limit value.

The Arpeggiator can be a lot of fun, even if you don't always understand exactly what it's doing. Keep in mind that the stranger the algorithm you set up, the more unlikely the notes will stay close to one key, so if you want to create something that's going to sound at all diatonic, keep it simple.

Glissando

When the Glissando parameter is **On**, the Arpeggiator chromatically fills between latched notes. When Glissando is on, the Arpeggiator ignores the Note Shift, Shift Limit, and Limit Option parameters.

You must latch at least two notes to get a result. When Glissando is on, all notes played in the arpeggiation range get latched, although you won't necessarily get meaningful results from all latched notes. In general, try to get each subsequent note you latch to be a change in direction. For example, try latching the following sequence of notes: C4, C5, G4, G5, C5, C6, G4, G5. The “glissando” changes direction around each change in direction of the latched notes.

Velocity

Velocity sets the attack velocity of the played notes. **Played** means each note repeats with the same velocity you played it at. **Last** means all notes play at the velocity of the most-recently played note. **Pressure** means the velocities are controlled by keyboard pressure: as you push down on any key, the velocities get higher, and as you ease up they get lower. **Ctrl 117** means the velocity is controlled by MIDI Controller number 117, which can be assigned as the destination of any K2600 physical controller or can come from an external MIDI source.

There's another element that affects arpeggiator velocity: ArpVel, whose global control number is 45 (not MIDI 45). Input from any physical controller assigned to send ArpVel (or any entry value for a controller assigned to send ArpVel) overrides the programmed value of the Velocity parameter, disabling it until you select a setup (or in Program mode, until you select a control setup on the MIDI-mode TRANSMIT page). The override occurs even if the Velocity parameter is set to **Ctrl117**.

There is also a range of fixed values, from **Vel 1** to **Vel 127**. Selecting one of these determines a fixed velocity setting, which you can use to set a constant volume for the arpeggiated portion of all zones. The value you choose from this range defines the velocity level for the Arpeggiator, and will take effect no matter how hard you strike a key.

Duration

Duration determines how long each arpeggiated note plays. 100% means that a note sustains until the next one sounds—very legato. **50%** means that the note fills half the space between itself and the next note. The lowest value is **1%**—*stacattissimo*. This parameter has no effect on percussion sounds or other sounds whose duration is fixed.

Real-time Control of Arpeggiator Parameters

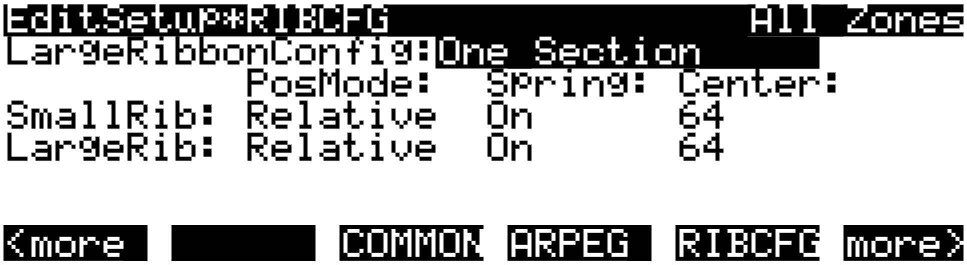
You can have real-time control over several arpeggiator parameters, by assigning physical controllers to special arpeggiator Controller Destinations. Any input (or entry value) from a physical controller assigned to an arpeggiator Controller Destination overrides the programmed values for the parameters. The override remains in effect until you select a setup (or in Program mode, until you select a control setup on the MIDI-mode TRANSMIT page).

There are countless possibilities for configuring physical controllers for real-time control over arpeggiator parameters. The following table covers just one possible configuration: Sliders A–G controlling each of the seven parameters that allow real-time control. We've included some suggested settings for the slider parameters, to make each slider at least reasonably useful for arpeggiator control. These settings are based on an arpeggiator range from C 6 to C 8 (as set with the LoKey and HiKey parameters on the ARPEG page).

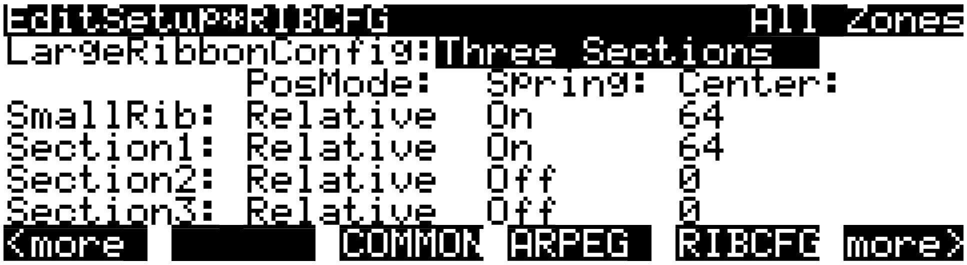
Slider	Assignment	Suggested Slider Settings
A	40 ArpOrder	
B	41 ArpBeats	Scale = **10%**; Add = **5**
C	42 ArpShift	Scale = **7%**
D	43 ArpLimit	Scale = 28%
E	44 ArpLmtOp	
F	45 ArpVel	
G	46 ArpDur	

The Ribbon Configuration (RIBCFG) Page

Once you've selected a destination for the Large and Small Ribbons, you can use the parameters on the RIBCFG page to define how the ribbons respond to finger position and pressure.



```
EditSetup*RIBCFG                 All zones
LargeRibbonConfig:One Section
          PosMode:   Spring:  Center:
SmallRib: Relative   On       64
LargeRib: Relative   On       64

<more          COMMON  ARPEG  RIBCFG  more>
```



```
EditSetup*RIBCFG                 All zones
LargeRibbonConfig:Three Sections
          PosMode:   Spring:  Center:
SmallRib: Relative   On       64
Section1: Relative   On       64
Section2: Relative   Off      0
Section3: Relative   Off      0
<more          COMMON  ARPEG  RIBCFG  more>
```

Parameter	Range of Values	Default
Large Ribbon Configuration	One Section/Three Sections	One Section
Position Mode	Relative/Absolute	Relative
Spring	On/Off	On
Center	0-127	64

Large Ribbon Configuration

You can use the Large Ribbon as one controller, or divide it up into three smaller sections, each with its own controller assignments. Choose a value of **One Section** or **Three Sections** for the LargeRibbonConfig parameter.

Notice the three white arrows directly above the Large Ribbon. The largest arrow (in the middle) points to the Large Ribbon's center. The two smaller arrows on either side are the section boundaries for three-section configurations.

Position Mode (PosMode)

When you touch one of the ribbons, the K2600 responds in one of two ways, depending on the setting of the PosMode parameter. **Relative** means that wherever you touch the ribbon becomes the "zero point" for whatever the ribbon is controlling; you won't notice any change in the sound until you slide your finger. Relative mode tends to be the most natural for performance—just wiggle your finger anywhere on the Large Ribbon to get vibrato in many factory programs and setups. You get the same effect no matter where you do the wiggling.

Absolute means that the zero point for the ribbon is always at exactly the same physical location on the ribbon. By default this is the center of the ribbon, although you can use the Center parameter to put the zero point elsewhere. In Absolute mode, just touching the ribbon affects the sound (unless you touch it at exactly the zero point). Every movement you make along the ribbon sends control values based on how far you are from the zero point.

Spring

When Spring is **On**, the Controller to which you've assigned the ribbon "springs" back to its zero point automatically when you lift your finger off the ribbon. This is generally the behavior you want.

When Spring is **Off**, the Controller to which you've assigned the ribbon sticks at its current value when you lift your finger off the ribbon. While this can be useful, you should be careful with it. When Spring is **Off**, if the ribbon is doing something when you exit the setup, it'll continue doing that when you return to the setup. This can be good or bad. If you want to set Spring to **Off** in a setup, but want to be sure of the initial sound of the setup, go to the RIBBON page in the setup, and for each ribbon for which you've set Spring to **Off**, set the entry value (Ent) parameter to a value of **0** (or whatever you like).

Center

This defines the zero point for each section on the ribbon—the point at which the Controller to which the ribbon is assigned has no effect on the sound. You can choose any point between **0** and **127**. A value of **64** puts the zero point in the physical center of the ribbon. Values of **0** and **127** place the zero point at the section's extreme left and extreme right. Note that setting the Spring parameter to **Off** disables the Center parameter for that ribbon.

The value you choose for this parameter can have considerable effect. In many factory setups, for example, the Large Ribbon affects pitch. In these setups, changing the value of the Center parameter would transpose the setup.

The Utility Soft Buttons

In addition to the Setup Editor's pages, there are basic library and editing soft buttons. Their functions are described below.

Name

This enables you to rename the current setup. Use any data entry method to do this, including the letters on the alphanumeric buttonpad.

Save

Pressing **Save** calls up the standard Save Dialog. Simultaneously pressing the **Plus** and **Minus** data entry buttons toggles between saving the setup to the first available empty location, or replacing a currently existing setup.

Delete

This erases a setup from RAM, freeing up space to store setups in other locations. (You can check the free memory in the K2600 at any time, on the top line of the Master-mode page.) Press **Delete**, and use a data entry method to choose which setup you want deleted. Press **Delete** again, and an "Are You Sure?" message will appear (unless you have set a value of **No** for the Confirm parameter on the Master-mode page). Press **Yes** to delete the setup, or **No** to cancel.

As with programs, setups can be saved to and deleted from RAM only. The names of all setups in RAM have an asterisk (✱) next to them. If you try delete a setup from ROM, the K2600 will ignore the delete command, and the setup will remain in memory.

Dump

This sends a MIDI System Exclusive dump of the current setup’s settings. There is also a function for dumping *all* setups. To dump all setups, go to Master mode. Select the OBJECT page, and press the **Dump** soft button to select which objects to dump via SysEx.

For more on Object Utilities, see Chapter 11. For more about System Exclusive messages, see the *Musician’s Reference*.

New Zone (NewZn)

Press **NewZn** to create a new zone with default parameters. The K2600 imports this zone from Zone 1 of **99 Default Setup**. If there are parameters or entire pages you use often, you can create your own Default Setup and save it to location 99; pressing **NewZn** will then import zones from your custom Default Setup.

Duplicate Zone (DupZn)

This adds a new zone with the same parameters as the current zone.

Import Zone (ImpZn)

You can import, or “bring in,” any zone from any setup in memory. Press **ImpZn**, and use any data entry method to choose a setup to import from. Then use the **Chan/Bank** buttons to select one of that setup’s zones. As you do this, you can audition each zone to hear how it will sound in the current setup. Now press **Import**, and the zone you selected will be added to the current setup.

Note: If you are using all eight zones in a setup and you try to add, duplicate, or import a zone, a “No More Zones” message appears. You must delete an existing zone before you can add, duplicate, or import any new ones.

Delete Zone (DelZn)

This deletes the current zone from the setup. Use **DelZn** to free up zones so you can add or import new ones.

Copy and Paste

These functions are handy if you’ve spent some time tweaking a page or line of parameters, and you want to copy them quickly to other zones. Start by highlighting a parameter’s value and pressing the **Copy** soft button. You can copy more than one value. Then use the **Chan/Bank** buttons to select the zone to which you want to copy those values. Press the **Paste** soft button, and all the values you copied replace the corresponding values in the current zone.

You can also paste the value into any zone of any other setup. Just copy a parameter’s value, press **Exit** to leave the current setup, and select a different setup. Then press **Edit**, select the same page or physical controller type as in the last setup, and paste the new value. The values you copy each remain in one of nine paste buffers, so you can keep pasting these values to different zones in any setup. When you press **Copy** on a new page or line of parameters, you replace the contents of the current paste buffer with the new values.

The table below shows where you can paste copied parameters. Note that the first seven paste buffers each store an entire page, while the last two each store only one line of parameters.

Copied Parameters	Available Destinations in Any Setup Zone
CH/PROG page	Corresponding page
KEY/VEL page	Corresponding page
PAN/VOL page	Corresponding page
BEND page	Corresponding page
COMMON page	Corresponding page
ARPEG page	Corresponding page
RIBCFG page	Corresponding page
Parameters for one continuous controller (on SLIDER, SLID/2, CPEDAL, RIBBON, WHEEL, or PRESS pages)	Assignment for any continuous controller
Parameters for one switch controller (on FOOTSW and SWITCH pages)	Assignment for any switch controller

Clear

Pressing **Clear** resets the selected page or line of parameters to its default value, as defined in **99 Default Setup**. You can edit the Default Setup, and alter the values that are called up for each page when you press **Clear**. The only exceptions to this are on continuous and switch controller assignment pages, where pressing **Clear** resets the current line to a factory-defined set of values.

You can't undo **Clear** without exiting the setup entirely, so be sure you really want to clear the selected parameters.

Set Range (SetRng)

As we mentioned earlier in this chapter, **SetRng** gives you a quick way to set the values for the LoKey and HiKey parameters in the currently selected zone. Press this button, and the K2600 prompts you to strike the low and high keys. When you've done so, the Setup-editor page returns, and the notes you triggered are the new values for the LoKey and HiKey parameters. Notice that the higher of the two notes you played is the HiKey value, regardless of the order in which you triggered the two notes. The ranges you set are now represented by lines in the box at the left on the Setup-mode page.

Editing Hints and Suggestions

Do These Parameters Always Mean Something?

Be careful not to set up parameters that do crazy things. When in doubt, leave things off, or at the factory default settings. On some of the higher-numbered MIDI controllers and Special Function controllers, the **On** and/or **Off** values don't have any meaning, because the controllers have very limited functionality. The following list shows these. **>=64** means greater than or equal to 64, and **<64** means less than 64. **>0** means you can use any value that's greater than zero. **x** means you can use any value.

MIDI Controller Number	MIDI Controller	On Value	Off Value
102	Play/Stop	>=64	<64
103	Record/Stop	>=64	<64
120	All Sound Off	>=64	<64
121	Reset All Controllers	>=64	<64
122	Local Control Off	>=64	<64
123	All Notes Off	>=64	<64
124	Omni mode Off	>=64	<64
125	Omni mode On	>=64	<64
126	Mono mode On (the On value, in some synths, specifies the number of MIDI channels to respond to, with one voice on each)	>=64	<64
127	127 Poly mode On	>0	x

Table 7-6 Little-used MIDI Control Sources

Do I Need All Those Pedals?

It's important to realize that you don't have to have an actual pedal plugged into every Pedal and Switch Pedal jack in order to take advantage of all of them. Entry and exit values can be assigned to a zone and to a physical controller's destination even if the controller isn't there, and they will behave as if the controller were in fact plugged in. So if you want a specific group of Controller commands to be sent out whenever you choose a setup, and you know they're not going to change while you're playing, you can assign them as the destinations of physical controllers that aren't in use.

Multiple Controllers

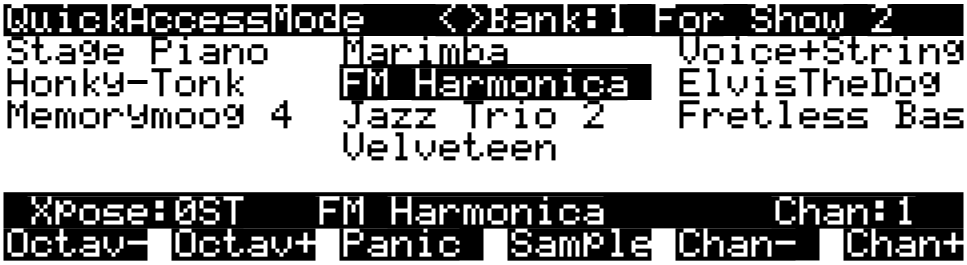
Sometimes you'll want to assign the same Controller number as the destination of more than one physical controller. For example, you might want to be able to control modulation depth both with Wheel 2 and with pressure. The K2600 has no restrictions on assigning multiple physical controllers to a particular MIDI Controller.

Editing Programs

Remember, except for control-setup parameters, everything you define in a setup affects programs *only while you are in Setup mode*. To edit the programs themselves while you are in the Setup Editor, go to the CH/PRG page and press **Edit**. This gives you access to the other editors nested within the Program Editor as well. When you have finished editing your program, press **Exit** to save it and return to the Setup Editor.

Chapter 8
Quick Access Mode and the Quick Access Editor

In Quick Access mode, you can select programs or setups with a single press of an alphanumeric button (or with the other data entry methods). For example, in the illustration below, you would simply press **5** on the alphanumeric pad to choose **FM Harmonica**. Notice that your selection becomes highlighted in the list, as well as appearing on the line just above the soft-button labels.



```
QuickAccessMode    <>Bank:1 For Show 2
Stage Piano    Marimba        Voice+String
Honky-Tonk     FM Harmonica   ElvisTheDog
Memorymoog 4   Jazz Trio 2    Fretless Bas
               Velveteen

 Xpose:0ST    FM Harmonica       Chan:1
Octav- Octav+ Panic  Sample Chan-  Chan+
```

Using Quick Access mode involves selecting Quick Access banks from the list of factory preset or user-programmed banks. You can use the bank selection shortcut to do this: press the **+/–** or **Clear** button on the alphanumeric pad, and you'll be prompted to enter a bank number. Type the desired number on the alphanumeric pad, then press **Enter**. The bank is selected, and you return to the Quick Access-mode page. Or use the **Chan/Bank** buttons to scroll through the QA banks.

Each bank contains ten memory slots, or entries, where you can store programs or setups in any combination. Any program or setup in the currently selected bank can be selected with the numeric buttons **0** through **9**.

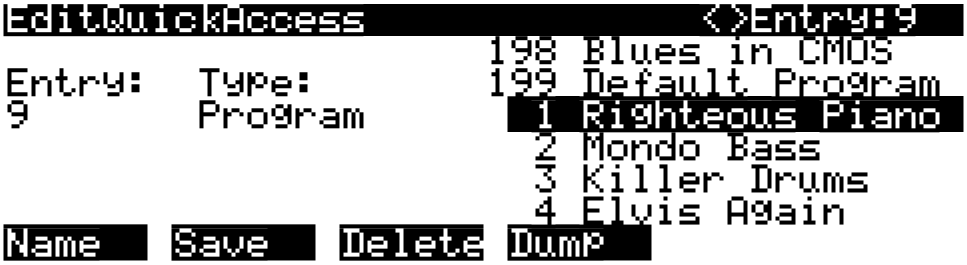
The ROM (factory preset) QA banks are organized into useful groupings of sounds that we think you'll find convenient.

You can store 20 Quick Access banks in each memory bank (except the Zeros bank, which can store 75). See *Storing Objects in the Memory Banks* on page 13-36 for a breakdown of the Quick Access-bank IDs that belong in each memory bank. Press both **Chan/Bank** buttons simultaneously to quickly move between memory banks.

The MIDI Program Change commands that the K2600 sends when in Quick Access mode can differ from those in Program or Setup mode. This depends on the setting you have for the PChgType parameter in MIDI mode. If the setting is **Extended** or **Kurzweil**, the Program Change commands sent are the same as in Program or Setup mode. If the setting is **QA Extended** or **QA Kurzweil**, the K2600 sends Program Change commands that correspond to the current Quick Access bank and the entry you select, not the actual program number of the entry. See *Program Change Formats* on page 10-11.

Everything you need to know about using Quick Access mode for performance is covered in Chapter 2, in the section called *Playing the Presets*, so we'll move on to the Quick Access Editor, which you'll use to create your own Quick Access banks.

The first step in editing Quick Access banks is to select Quick Access mode. Then use the **Chan/ Bank** buttons to select the bank you wish to edit. The currently selected bank is shown in the top line of the Quick Access-mode page. Press the **Edit** button, and you enter the editor, where you can examine each entry in the bank you selected. The Quick Access Editor page looks like this:



```
EditQuickAccess                  <>Entry:9
                           198 Blues in CMOS
Entry:   Type:             199 Default Program
9        Program             1 Righteous Piano
                             2 Mondo Bass
                             3 Killer Drums
                             4 Elvis Again
Name     Save     Delete  Dump
```

The top line gives you the usual mode reminder, and shows you which of the ten entries you're looking at. The cursor is highlighting the object (program or setup) that's stored in that entry.

The easiest way to edit the bank is to use the **Chan/Bank** buttons to scroll through the ten entries. The entry number changes both at the top of the page, and at the left of the page. As the entry number changes, the highlighted objects at the center of the page change as well, showing you what's stored in each entry. On the page above, for example, entry 9 is the current entry. The Type parameter tells you that the object stored at entry 9 is a program. The cursor highlights the program's ID and name.

In this example, you could select a different program with your favorite data entry method. If you wanted to store a setup in that entry instead of a program, you would move the cursor to the Type parameter and change its value to **Setup**. The list of objects would change from the program list to the setup list, and you could move the cursor back to the setup list and select another setup. When you select the Entry or Type parameter, the list of objects at the right disappears, leaving only the currently selected object. This makes it easier to see when it's not highlighted by the cursor.

When you've filled each entry with the object you want, press the **Name** soft button if you want to rename the bank, or press the **Save** soft button to begin the save procedure. Press the **Dump** soft button to dump the bank via MIDI System Exclusive.

Chapter 9
Basic Effects Mode

Introduction

Effects mode puts the power of an entire studio—equalizers, signal processors, and mixers—inside your K2600. It allows you to define processing functions, signal paths, and balances, on sounds created by the K2600 and, in Live mode, sounds from *any* source.

When you're in Program, Setup, or Quick Access mode, you can take advantage of FX Modulation routings (FXMods, which are preset overrides for studio parameters) for real-time control over any aspect of your studio—either from the buttons, sliders, wheels, and pedals on your K2600, or from any MIDI source at all, like a sequencer or a dedicated controller.

Effects mode provides up to five individual effects processors. Four of these are designed to operate on their own individual buses, and the fifth, the "Auxiliary" processor, is designed to be global.

When the K2600 is being used multi-timbrally, Effects mode can provide separate processors for each of several instruments. For example, let's say four K2600 channels are in use, under the control of a sequencer. Each instrument on each channel can have its own processing: flanger for the piano, delay for the sax, compression for the bass, and tight reverb for the drums. Then all of these signals could go through a global Aux processor, where they could be given a room reverb. At the outputs of the K2600, the reverb could show up on one pair, while the other instruments' dry or processed (prereverb) signals could show up at the others, either in mono or stereo. Or the entire mix, carefully balanced and panned, could appear at a single pair, ready to be recorded or played through a PA.

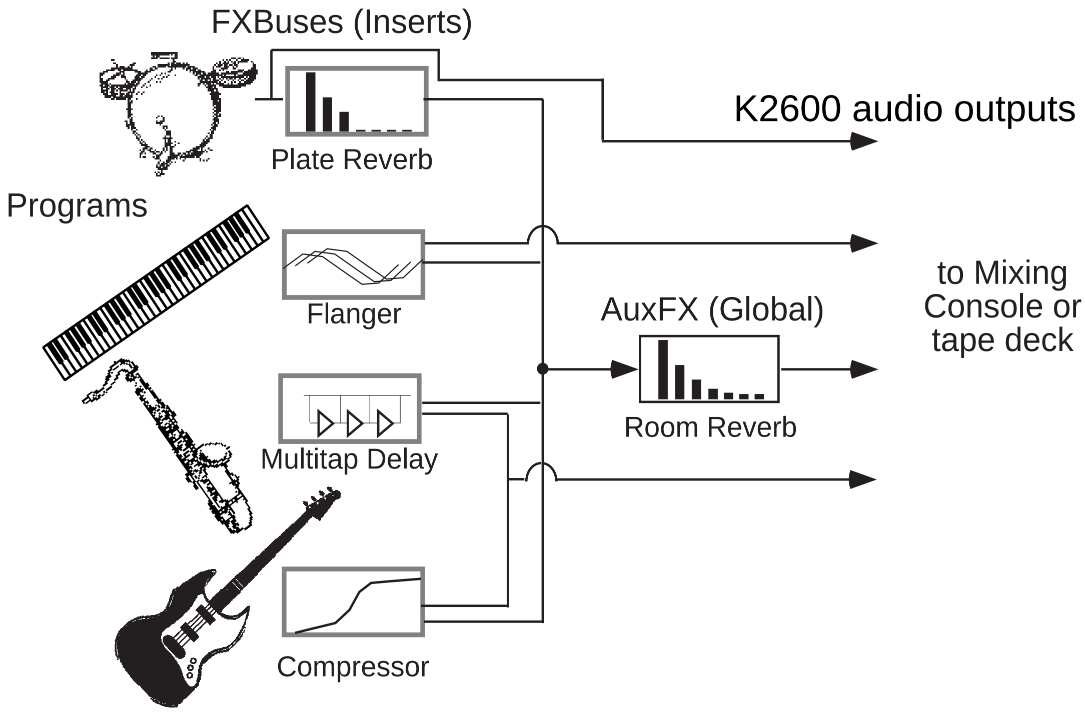


Figure 9-1 A typical KDFX Studio

Alternatively, in a live performance setting, you can use the KDFX in Setup mode as an entire orchestra, with different instruments layered on top of each other and mapped to different parts of the keyboard, all with their own signal processing—and that signal processing can change completely in an instant when you call up a new setup.

These are just two of the huge variety of scenarios possible with KDFX. The structure of KDFX is so flexible that you should never run out of ideas for how it can be used.

Terminology

Before starting to work with Effects mode, take a minute to familiarize yourself with the relevant terms. Understanding the words will help you get the concepts more quickly. Read this chapter first, and refer to it later if something isn't clear.

Studio

A studio is the entire Effects-mode environment, consisting of inputs, FXBuses, an Aux Bus, FX presets, parameter settings, overrides, and outputs. One studio is current at a time in Effects mode. Studios are objects, like K2600 programs, and have a number. They can be linked to programs or setups using the FX Mode setting. The K2600 comes with a large number of factory studios. You can modify them or create new ones and store them in RAM.

FXBus (also called Insert FXBus)

A signal path with some kind of processing —a delay, reverb, flanger, etc.—on it. It has stereo inputs and outputs, and contains an FX preset that determines the kind of processing to be applied. A studio has four insert FXBuses, each with its own FX preset, as well as a fifth FXBus, known as the Aux bus.

Aux FX Bus (Aux Bus)

A separate bus from the insert FXBuses, which is placed in the signal path after the output of the insert FXBuses, so that it can act as a global processor.

FX Preset

Determines the type of signal processing that is present on an FXBus or the Aux Bus. FX presets consist of an algorithm, plus the settings of the parameters associated with that algorithm. Any number of FX presets can use the same algorithm. The K2600 comes with a large number of FX presets, and you can modify them or create new ones and store them in RAM. Algorithms, however, are stored in ROM, so you can't change them. Each FXBus and the Aux Bus gets its own FX preset, so up to five FX presets can be active in a studio at a time.

Algorithm

A specific type of signal processing, like a hall reverb, plate reverb, chorus, flanger, pitcher, compressor, or rotary speaker, which is at the core of an FX preset. KDFX comes with a large variety of algorithms, which are stored in ROM, and are not user-definable. Additional algorithms, supplied by Kurzweil, can be loaded in from disk when they become available. Algorithm parameters (RT_{60}, delay feedback, pitch change, etc.) *are* user-definable; the K2600 stores the values of those parameters in RAM as part of an FX preset.

Size/PAUs

How big an algorithm is, in terms of how much processing power it needs to operate. Size is measured in Processor Allocation Units, or PAUs. The simplest algorithms require only 1 PAU, while more complex ones require 2, 3, or even 4. The total number of PAUs available to the four insert FXBuses is 4. PAUs can be allocated automatically as FX presets are assigned to FXBuses, or manually by the user. The Aux Bus has its own set of 3 PAUs, which are not shared with the insert FXBuses.

Parameters

Refers to the user-controlled settings for the different characteristics of an algorithm. For example, the user can set the reverb time (RT_{60}) of an algorithm to **3.5s**, or the delay feedback of a delay line to **90%**, or the dynamic ratio of a compressor to **10:1**. Parameters are stored, along with the algorithm they modify, as part of an FX preset. In certain cases, the parameter settings within a studio can be overridden, either using FXMods or bus overrides.

FXMods

FXMods give you real-time control over many of the parameters within a studio. Parameters on any of the pages of the Studio Editor can be controlled by MIDI commands—such as physical controller input or sequencer data—or by internal K2600 functions like LFOs, envelopes, clocks, or key states. FXMods are not part a studio or FX preset; they're part of a program or setup.

A program or setup can have up to 18 FXMods, as well as two dedicated LFOs, ASRs, and FUNs for further effects control. The FXMods and the LFOs, ASRs, and FUNs are defined on seven FX pages in the Program and Setup Editors.

Bus Overrides (or BusMods)

Bus overrides allow parameters within an FX preset to be controlled on the FXBus Editor page in the Studio Editor, outside the FX Preset Editor page. Any two parameters in an FX preset can be brought out to its FXBus page. Bus overrides are stored as part of the studio, not as part of the FX preset. They are useful when an existing FX preset is close to what you want, so that you can tweak it without having to create a new FX preset.

FX Mode

The effects mode is determined by the value of the FX Mode parameter, which is found on the Effects-mode page. It determines how the K2600 selects studios when you change programs or setups, and determines whether you have real-time control over studio parameters—in other words, whether FXMods are active.

If the value of FX Mode is **Program** or **Auto**, then as you change programs in Program mode, the K2600 also loads the studio linked with that program. This activates all the FXMods defined within the program.

If the value of FX Mode is **Setup** or **Auto**, then as you change setups in Setup mode, the K2600 also loads the studio linked with that setup. This activates all the FXMods defined within the setup.

If the value of FX Mode is **Auto**, and the value of FX Channel is **Current**, then when you're in Program, Setup, Quick Access, or Song mode, programs, setups and songs automatically use their corresponding studios. In Program, Setup, and Quick Access modes, the studio corresponds to the current program or setup. In Song mode, the studio corresponds to the program on the song's assigned effects channel (which is determined by the value of the EffectChan parameter on the COMMON page in the Song Editor).

If the value of FX Mode is **Master**, changing programs or setups does not load an associated studio; the current studio is defined by the Studio parameter on the Effects-mode page. Any FXMods defined in the current program or setup are inactive.

Studio Editor

To get to the Studio Editor, go to the Effects-mode page, highlight the current studio, and press **Edit**. The Studio Editor contains the following pages:

The INPUT page, where signals coming from the K2600's four stereo output buses are routed to one or more FXBuses.

The FXBUS page, for assigning FX presets to the four FXBuses, and for defining bus overrides.

The AUXFX page, for assigning an FX preset to the Aux Bus, and for defining bus overrides.

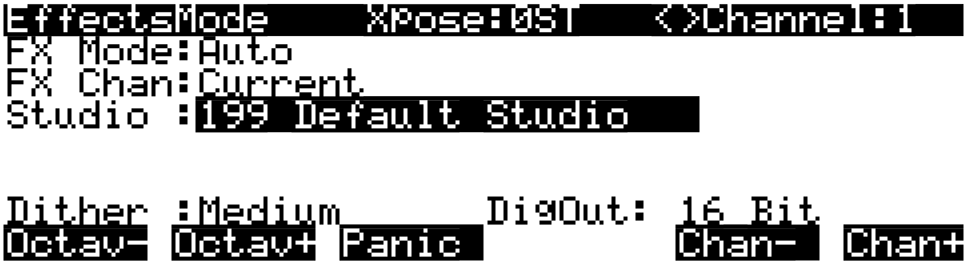
The OUTPUT page, where the signals coming from the FXBuses and Aux bus are routed to the K2600's eight physical outputs (analog and digital).

The FX Preset Editor, which is nested within the Studio Editor, and which contains three pages. The FX Preset Editor is where you select algorithms and set parameters for the FX presets, as well as doing administrative things like naming and saving. You can get to the FX Preset Editor from the either the FXBUS page or the AUXFX page, by highlighting the current FX preset and pressing **Edit**.

Name, Save, Delete, and **Dump**, for doing file management on your collection of studios.

You can also get to the Studio Editor from the Program and Setup Editors, by pressing **Edit** when the Studio parameter is selected (this is the most convenient method for getting into the Studio Editor). In both the Program and Setup Editors, the Studio parameter is on the KDFX page.

The Effects Mode Page



```
EffectsMode     XPose:0ST    <>Channel:1
FX Mode:Auto
FX Chan:Current
Studio :199 Default Studio

Dither :Medium       DigOut: 16 Bit
Octav-  Octav+  Panic          Chan-   Chan+
```

Parameter	Range of Values	Default
FX Mode	Program, Setup, Auto, Master	Auto
FX Chan	None, 1–16, Current (varies according to FX Mode value)	Current
Studio	Studio List	199 Default Studio
Dither	None, Flat, Minimum, Medium, Maximum	Medium
DigOut	16 Bit, 20 Bit	16 Bit

FX Mode

This parameter determines which studio the K2600 selects when you select programs or setups. It also determines whether FXMods are active. There's a complete description on page 9-4. **Auto** is the most commonly-used value for FX Mode, since it's the most flexible.

Effects Channel (FX Chan)

This parameter is closely linked to the FX Mode parameter—the values available for FX Chan depend on the setting for FX Mode. FX Chan affects both studio selection and real-time control of KDFX—again, depending on the value of FX Mode.

When FX Mode is Master

The effects channel is irrelevant when FX Mode is **Master**. KDFX is under the control of the studio you select with the Studio parameter. No FXMods apply.

When FX Mode is Program

The available values for FX Chan are **Current**, and **1–16**. When FX Chan is **Current**, studio selection is determined by whatever program is assigned to the current K2600 channel (as shown in the top line of the display in Program mode or Effects mode). That program's FXMods are also enabled, and they respond to MIDI Controller messages received on the current channel.

When FX Chan is any value from **1** to **16**, studio selection is determined as follows: in Program mode it's determined by the program assigned to that channel; in Setup mode, it's determined by the program in the zone assigned to that channel. FXMods respond to MIDI Controller messages as follows: in Program mode, the FXMods of the program on the channel specified by FX Chan respond to messages on that channel; in Setup mode, the FXMods of the program in the zone using the channel specified by FX Chan respond to messages on that channel.

The Studio parameter disappears, since studio selection is under program control.

When FX Mode is Setup

The only available value for FX Chan is **None**, since studio selection and FXMods are determined by the program in Zone 1 of the current setup. The FXMods of the program in the zone using the channel specified by FX Chan respond to MIDI Controller messages on that channel. The Studio parameter disappears, since studio selection is under setup control.

When FX Mode is Auto

The available values for FX Chan are **Current**, and **1–16**. When FX Chan is **Current**, studio selection and FXMods are determined as follows: in Program mode, it's determined by the program assigned to the current K2600 MIDI channel; in Setup mode, it's determined by the program in Zone 1 of the current setup; in Song mode (or during song playback regardless of mode), it's determined by the program on the channel designated as the song's effects channel—as specified by the EffectChan parameter on the COMMON page in the Song Editor. FXMods respond to MIDI Controller messages as follows: in Program mode, the FXMods of the program on the K2600's current channel respond to messages on that channel; in Setup mode, the FXMods of the program in Zone 1 of the current setup respond to messages on the channel used by Zone 1.

When FX Chan is any value from **1** to **16**, studio selection is determined as follows: in Program mode it's determined by the program assigned to that channel; in Setup mode, it's determined by the program in Zone 1 of the setup (ignoring the FX CHan setting). FXMods respond to MIDI Controller messages as follows: in Program mode, the FXMods of the program on the channel specified by FX Chan respond to messages on that channel; in Setup mode, the FXMods of the program in Zone 1 of the setup respond to messages on that channel (again, ignoring FX Chan).

Studio

When FX Mode is **Master**, the Studio parameter selects the studio for the entire K2600. When FX Mode is **Program** or **Setup**, this parameter is unavailable, since setup selection and FXMods are determined by program or setup selection. When FX Mode is **Auto**, setup selection and FXMods are under program or setup control, but the Studio parameter is still available; its value reflects the studio assignment for the program or setup that was current before you entered Effects mode. You can change the value of the Studio parameter while in Effects mode, so you can hear how different studios affect the current program or setup, without having to enter the Program or Setup Editor. Any changes you make here revert to their previous settings when you exit Effects mode.

Dither

Dithering adds a small bit of noise to a digital signal to lower quantization distortion. Use the Dither parameter to select different digital noise floor characteristics. No setting is superior to another—each one is a tradeoff between perceived noise level and artifacts.

Flat—the digital noise floor is spectrally flat, so it's a good choice for resampling the K2600's own output, as there will be no addition of noise in the process. This is sometimes known as triangular dither.

Minimum, **Medium**, **Maximum**—these utilize noise shaping in varying degrees. The noise floor has been essentially equalized to make it softer at midrange frequencies, and louder at bass and treble frequencies. The overall power level of the noise is actually higher than at the **Flat** setting, but due to the ear's spectral response at low levels, the noise floor appears lower at these settings.

These values are not as useful as **Flat** for resampling, since the boosted noise may become quite noticeable if the sample is played transposed down a couple of octaves. In addition, repeated resampling would cause buildup at the boosted frequencies.

None—no dither. This is provided primarily for completeness. You generally never want to use this setting, unless you are studying dither in a classroom, or you specifically want to hear digital artifacts. Low-level signals are extremely distorted using this setting.

Digital Word Length (DigOut)

This parameter is available only if you have the sampling option in your K2600.

The K2600 is capable of producing 20-bit digital audio signals. Normal 16-bit digital audio provides a maximum dynamic range of about 96 dB; adding the extra four bits adds some 24 dB to the dynamic range. There's an increasing number of products on the market that can handle 20-bit digital signals, including tape and hard-disk recorders, mixers, and signal processors, and the K2600 can talk to these devices with the highest degree of fidelity.

The DigOut parameter toggles between 16-bit and 20-bit operation. The value for DigOut determines the word length used for the KDS eight-channel output, and also for the standard digital output, provided it is in Direct mode (as set on the SampleMode page).

If you are using your K2600 with a Kurzweil DMTi, set DigOut to **16 Bit**, since the DMTi currently operates only in 16-bit mode. If you're connecting the digital output to a 20-bit (or more) capable device, set DigOut to **20 Bit** and you can take advantage of the significantly reduced noise floor.

Note that the setting of DigOut doesn't affect any of the K2600's inputs: those are always operating in 16-bit mode. Likewise, DigOut doesn't affect the K2600's own DACs (digital-analog converters). They always operate in 18-bit mode.

The Structure of Kurzweil Digital Effects (KDFX)

The Studio

All Effects-mode functions are contained in a studio. Only one studio is active at a time.

Each studio has four stereo or eight mono inputs. The sources for these inputs are the four output pairs available in two places: the Pair parameter on the OUTPUT page in the Program Editor, and the Out parameter on the CN/PRG page in the Setup Editor. In both cases, the available values are KDFX-A, KDFX-B, KDFX-C, and KDFX-D. Instead of going directly to the

K2600’s physical outputs, these program outputs go to the inputs of a studio. Only after going through the studio does the audio output appear at the physical outputs. The inputs to the studio can come from one multi-layered program, or from several programs, or from the K2600’s own sample input when the instrument is in Live mode.

The input signals pass through an input routing system, as specified by the settings on the INPUT page in the Studio Editor. Here, the signals go through individual two-band equalizers, or EQs. Then each input, mono or stereo, is routed to any two of the four FXBuses.

Each of the FXBuses contains its own signal-processing program, called an FX preset. Each FX preset has a set of parameters: for example the RT_{60} value on a reverb, or the Feedback level on a delay line. These parameters can be fixed as part of the FX preset, or they can be externally controlled, from one of two different places. The four FXBuses are also called Insert FX, because in a conventional studio, that’s where they would be found: in the insert loop of a mixer, between the channel input and the Mix bus.

There is also a fifth FX preset, which is located on the Auxiliary FX Bus (the Aux bus). The Aux bus follows the four main FX presets in the signal path, and is normally configured as a global processor.

Each FX preset consists of an algorithm, modified by user-definable parameters. An algorithm is a processing function, like a reverb, flanger, or compressor; or a combination of processing functions in a particular order, like a flanger followed by a delay followed by a reverb. The algorithms themselves are fixed in ROM, like Kurzweil ROM samples, but you can change the values of their operating parameters.

Each algorithm requires a certain amount of processing power, which is expressed in Processing Allocation Units (or PAUs). Simple algorithms require 1 PAU, while more complex algorithms require up to 4 PAUs. The amount of processing power available in each FX preset is set by its Allocation parameter. When you are selecting an FX preset for an insert bus, the number of PAUs its algorithm requires appears on the display, so you can keep track of how many PAUs are in use.

PAUs are shared among the four insert buses. There is a limit to the total number of PAUs that the insert buses can use, and that limit is four. PAUs can be manually preassigned to specific FXBuses, or using “Auto” mode they can be assigned automatically as FX presets are assigned to the buses. The Aux bus has a separate set of PAUs—three of them—which are *not* shared with the insert buses.

Finally, the outputs of the FX presets are passed through to an output routing system—as specified by the settings on the OUTPUT page in the Studio Editor—where they are sent to the physical outputs of the K2600.

The following equation summarizes studio structure:

Studio = EQs + Input-page settings + FX presets + Output Editor settings

The next page shows a schematic overview of studio structure.

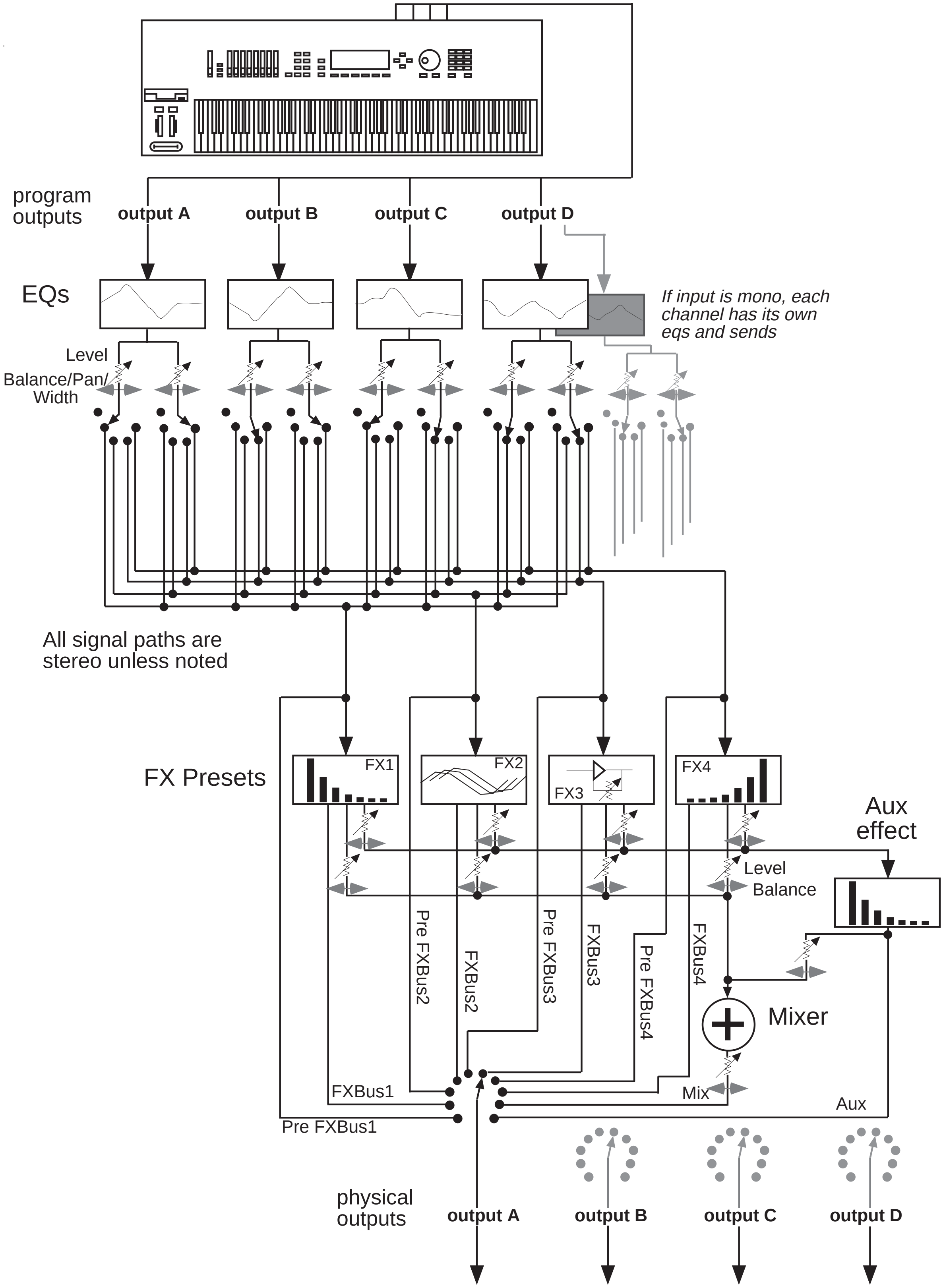


Figure 9-2 KDFX structure

Normal Studio Structure

Many of the studios provided in ROM follow an overall organizational plan, which uses Effects mode's resources efficiently and clearly. While you're by no means required to follow this structure when creating your own studios, it's a good idea to get familiar with it, to see how Kurzweil's own engineers have approached the issue of studio organization.

Inside the ROM programs and setups, the outputs KDFX-A through KDFX-D are assigned based on the type of effects processing that would most likely be appropriate for that sound.

- Input A/FXBus1 contains a relatively simple reverb with a low Size requirement.
- Input B/FXBus2 contains an effect that does not increase the "length" of the sound (that is, no reverb or delay), something like chorus, flange, distortion, pitcher, or EQ.
- Input C/FXBus3 contains effects that take up lots of time, such as delays, and delays with reverb.
- Input D/FXBus4 is dry.
- The Aux bus contains a larger reverb (Size:3), a compressor, or a graphic EQ.

Software Organization

Like K2600 programs, the software user interface of a studio is organized in a "top-down" way: A studio is an "object" in K2600 terms, and the FX presets within a studio are also objects, "dependent" to the studio, the way a keymap is to a program. Studios and FX presets that you modify or create are stored in program RAM. A studio contains up to five FX presets. The same way that one keymap can be used in multiple programs, a particular FX preset can be used in multiple studios.

Like ROM samples in the K2600, effects algorithms are stored in ROM and are unchangeable. Each FX preset contains one algorithm. Just as you can use the same sample in more than one keymap, you can use the same algorithm in as many FX presets as you like. Although you cannot modify algorithms, additional effects algorithms will be available from Kurzweil on disk.

Controlling Effects

Studios can be called up from the front panel, or they can be assigned to specific K2600 programs or setups. If you set the FX Mode parameter on the Effects-mode page to **Program**, **Setup**, or **Auto**, then changing a program (if the current channel agrees with the FX Channel) or setup will select the studio associated with the new program or setup. Of course, just as you can use the same keymap in multiple programs, you can use the same studio in multiple programs or setups. Setting FX Mode to **Master** means that the studio does not change with the program or setup.

Studio parameters are set in the Studio Editor and are initially set up with fixed values, but they can be also be controlled in real time by a wide range of control of sources, including K2600 onboard knobs, sliders, and ribbons; various internal software functions; and external MIDI commands like those coming from a sequencer or a dedicated controller.

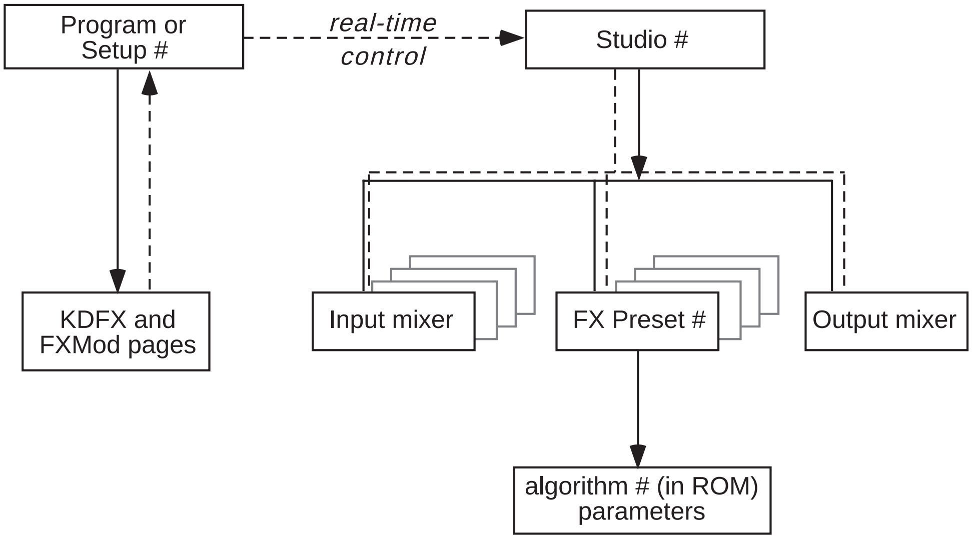


Figure 9-3 KDFX Software organization

Controlling Effects with a Sequencer

A useful technique for sequencer users is to set up the program controlling the studio parameters to be a dummy program with no keymaps but with all of the FX Mods you need in place—this program produces no sound by itself, and exists only to control the studio. That way you can use a dedicated MIDI channel for studio control. See page 15-29 and page 12-21 for more information.

Controlling Effects with SysEx messages

You can also control KDFX by sending MIDI system-exclusive (SysEx) messages to the K2600. See Appendix B in the *Musician's Reference* for specific information.

A Tour of Effects Mode

Let’s take a tour through several studios. After you do this, you should have a pretty good idea of what it’s like to work with Effects mode. You will still, of course, have many questions—answering them is the job of the rest of this manual.

Load the Tutorial files

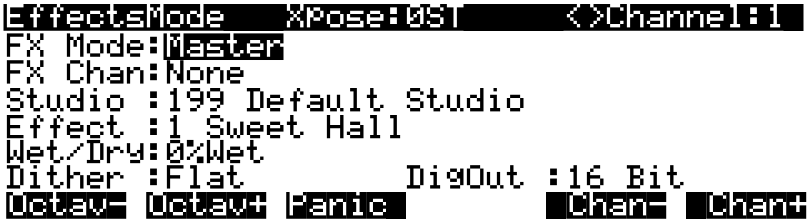
From the K2600 Demos disk that came with your K2600, load the file **KDFXTUTR.K26** into bank 200...299. You can use a different bank if you like, but then your numbers and the ones in this chapter will be different.

1. Insert the disk into the drive
2. Press **Disk** to go into Disk mode
3. Set Current Disk to **Floppy**
4. Press **Load**
5. Cursor down to **KDFXTUTR.K26**
6. Press **OK**
7. In the “Load this file as:” dialog, select **200...299**
8. Press **OK**
9. Press **OverWrt** (this will erase all objects in the 200s bank, so you should save anything you want to keep before doing this; press Cancel if you want to save objects before proceeding).
10. Press **Exit** to get back to Program mode.

A Simple Studio

Call up Program #199 on your K2600. In Program mode, either scroll the Alpha wheel to Program 199 (Default Program), or press **1-9-9-Enter** on the Alphanumeric pad.

Now go into Effects mode by pressing the **Effects** button, and you’ll see this page:

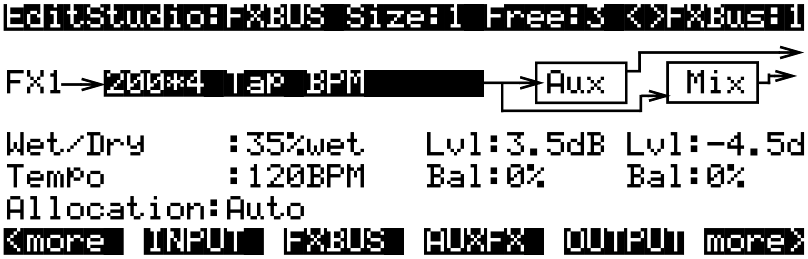


```
EffectsMode    XPose:0ST       <>Channel:1
FX Mode:Master
FX Chan:None
Studio :199 Default Studio
Effect :1 Sweet Hall
Wet/Dry:0%Wet
Dither :Flat         DigOut :16 Bit
Octav- Octav+ Panic          Chan-  Chan+
```

If the page looks very different from this, then put the cursor on the FX Mode parameter and scroll the Alpha wheel until it says **Master**. Now use the **Down** cursor button to highlight the Studio parameter. Scroll the Alpha wheel until the Studio parameter’s value is **200 Simple**.

The FXBus Page

Now press **Edit**, and this page appears:



```
EditStudio:FXBUS Size:1 Free:3 <>FXBus:1
FX1→200*4 Tap BPM        →Aux →Mix
Wet/Dry    :35%wet   Lvl:3.5dB Lvl:-4.5d
Tempo      :120BPM   Bal:0%    Bal:0%
Allocation:Auto
<more  INPUT  FXBUS  AUXFX  OUTPUT more>
```

This is the FXBus page for FXBus1. It is where an FX preset is assigned to the FXBus. Put the cursor on the box containing **200*4 Tap BPM—**this is the current FX preset, a four-tap delay whose speed is expressed in terms of tempo (Beats Per Minute). Use the Alpha wheel to scroll through the many other FX presets that come with KDFX. Like all K2600 objects, those that are in RAM (like this one) will have an asterisk in their name, and those that are in ROM will not. There are three more FXBuses, which you can view by pressing the **Chan/Bank** buttons. In this studio they are all empty (No Effect).

Go back to **4 Tap BPM** on FXBus1, and play the piano sound from your keyboard. The arrow next to FX1 flashes, showing that there is audio passing through this FXBus. The arrow keeps flashing as long as the FXBus is processing audio.

Below the FX preset selector is a Wet/Dry control, which determines how much of the signal will pass through the FX preset. Below that is a Tempo control, which sets the timing of the delays. These parameters are called "bus overrides," because they override parameters which are actually inside the FX preset itself—these parameters can be adjusted from inside the FX preset, or they can be set from out here, where they are much more convenient. If you change FX presets, these values change, because their values inside the various FX presets are all different. More about bus overrides a little later in this chapter.

Levels

To the right are two sets of level controls from the FX preset to the output mixer. The first set, Lvl and Bal, determines how much of the sound will go to the global or Auxiliary effects bus, and how its two channels are balanced. On this FXBus, the signal to the Aux bus is boosted 3.5 dB. The second set determines how much of the sound goes to the Mix bus. On this FXBus, the Mix signal is attenuated -4.5 dB.

Allocation

The Allocation parameter determines how many processor allocation units (PAUs) are reserved for this FXBus. The number of PAUs an FX preset uses is dependent on the algorithm at the core of the FX preset. Algorithms can use anywhere from 1 to 4 PAUs, depending on their complexity. As you scroll through the FX presets, the number of PAUs required by each one is shown on the top line of the display (Size:), along with the number of PAUs that are available (Free:) for other algorithms.

The four insert FXBuses have 4 PAUs to share among them, so if any bus uses more than one PAU, it means that some buses cannot be assigned an FX preset. This is a very common situation, as you shall see. The Auxi bus has its own set of 3 PAUs, which are completely independent and are not shared with the insert FXBus PAUs.

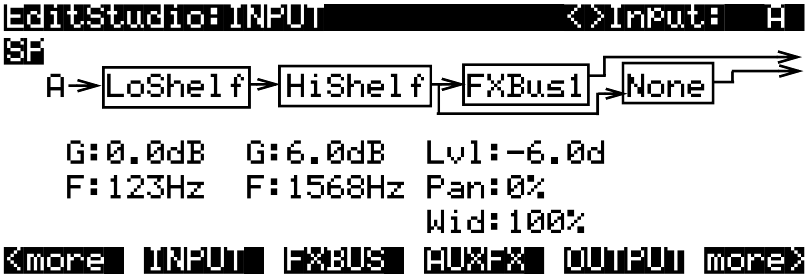
You can preassign a PAU value to an FXBus, in which case any FX presets that require more PAUs than you have given the bus cannot be loaded into the bus. If you try to put an FX preset into a bus that requires more PAUs than are currently available on that bus, the preset's name appears in parentheses—exactly the way a KB3 program appears if you try to select it on a channel that's not the KB3 channel.

In most factory studios, the Allocation parameters are set to Auto, in which case PAUs are assigned dynamically as you assign FX presets to the various buses. As you unassign FX presets from buses, or assign FX presets with smaller PAU requirements, the PAUs freed up are automatically reassigned to other buses where they are needed.

A value of Auto for this parameter allocates PAUs to lower-numbered FXBuses first, as needed.

The INPUT Page

Press the soft button labelled **INPUT**. This page appears:



```
EditStudio:INPUT               <>Input:  A
SP
  A->LoShelf->HiShelf->FXBus1->None

   G:0.0dB  G:6.0dB  Lvl:-6.0d
   F:123Hz  F:1568Hz Pan:0%
                     Wid:100%
```

This is the INPUT page. It is showing Input A, as indicated both on the left side and in the upper right corner—use the **Chan/Bank** buttons to view the other three inputs. Input A is the first stereo signal pair coming from the K2600 Program Editor's Output section. Depending on how the K2600 is set up, this could be a single layer of a single program, or multiple layers, or multiple programs, or one or more zones from a setup, or the output from one or more MIDI channels.

The **S** at the upper left says that the Input A is being handled as a stereo feed; this can be changed to two mono feeds. The P means that the stereo feed has a Pan control; you can choose to make this a Balance control instead, by setting this parameter to **SB**.

If you play on the keyboard, you can see the arrow next to the letter A flashing, as audio is being passed through this part of the studio. The arrow on this page flashes only as long as there is an input signal present.

Equalization

The first two blocks are the low and high EQs on the input. Put the cursor on either box and turn the Alpha wheel, and you will see the options you have available for types of EQ—these include **None**, which bypasses that EQ. The first block has more choices than the second.

The G underneath each block is its Gain; **0.0dB** is unity gain; the signal passes through without change. (There is no Gain parameter when a block is set to LoPass or HiPass.) F is the equalizer's frequency. In this input section, the high frequencies are boosted 6.0 dB above 1568 Hz.

Sends

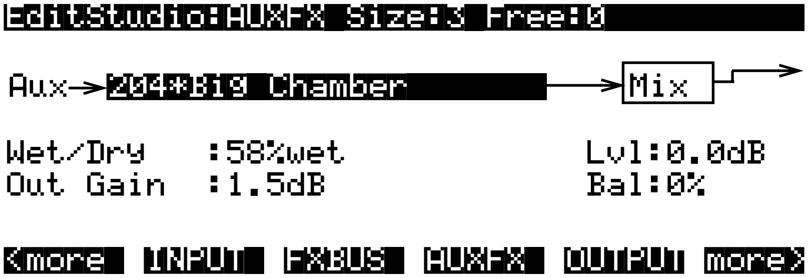
The third and fourth blocks determine the destinations of the Input A signal: each block can be set to route the signal to any of the four insert FXBuses, or to None. You cannot, however, set both blocks to the same destination.

The Lvl control is the FXBus send; it sets the level of the signal to the FXBus above it. In this input section, the level is backed off 6.0 dB, to compensate for the treble boost in the equalizer, so that the signal doesn't overload the FXBus.

Pan determines the position of the signal respective to the left and right sides. Width (which is not shown when you are using mono inputs) determines how much the left and right sides' signals will be separated or blended.

The AUXFX Page

Press the **AUXFX** soft button. This page appears:

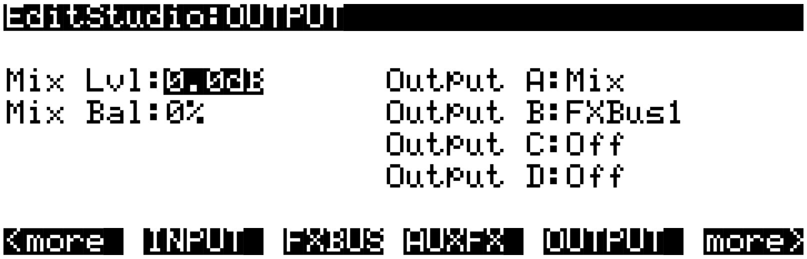


```
EditStudio:AUXFX Size:3 Free:0

Aux→204*Big Chamber        →Mix

Wet/Dry   :58%wet              Lvl:0.0dB
Out Gain  :1.5dB               Bal:0%
```

This is the Global AUXFX page, and shows us what is happening on the Auxiliary effects bus. The Aux bus is a second processor, which follows the four insert FXBuses. It has its own FX preset, with bus overrides, and level and balance controls to feed it into the Mix bus. It doesn't share PAUs with the FXBuses; it has three PAUs of its own, and consequently doesn't have an Allocation parameter. The Aux bus can be routed all by itself to an output, as we'll see. In this studio, the Aux bus contains a chamber reverb.

The OUTPUT Page

Press the **OUTPUT** soft button.



```
EditStudio:OUTPUT

Mix Lvl:0.0dB        Output A:Mix
Mix Bal:0%           Output B:FXBus1
                     Output C:Off
                     Output D:Off
```

The OUTPUT page is the interface to the real world. It determines which of the signals going through the various effects buses show up at the K2600's four sets of *physical* outputs: A, B, C, and D. These four outputs, all stereo, are both analog and digital (through the KDS bus). Output A also goes to the K2600's AES/EBU digital output.

In this studio, Output A is carrying the Mix, that is, the combination of the outputs of the four FXBuses (only one of which is in use) and the Aux bus. Output B is carrying FXBus1, which is the signal after it passes through the delay on FXBus1, but before it gets to the reverb on the Aux bus. The other outputs are carrying no signal.

***Note**: If you're using the Mix audio outputs, keep in mind they carry the summed signals of audio outputs A through D. Normally you would assign each of the audio outputs differently. In the example above, you might set Output B to* **Off**, *or Output A to* **AuxFx**, *to avoid applying the FXBus1 effect to the entire program.*

Here's a diagram of what this studio looks like:

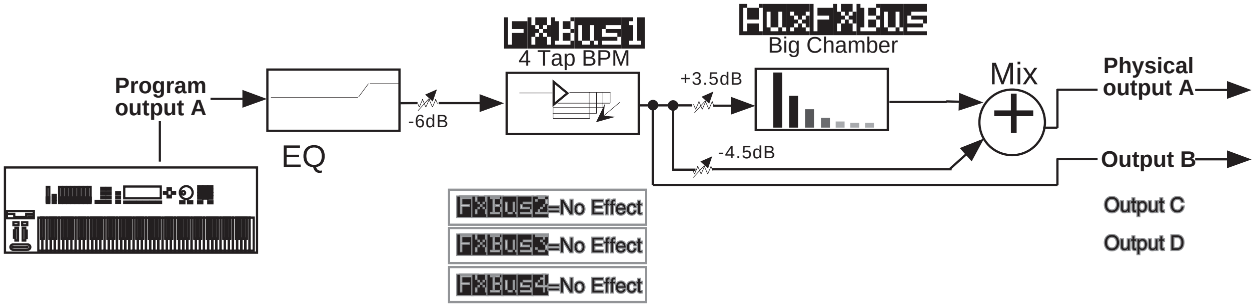


Figure 9-4 Structure of Studio 200*

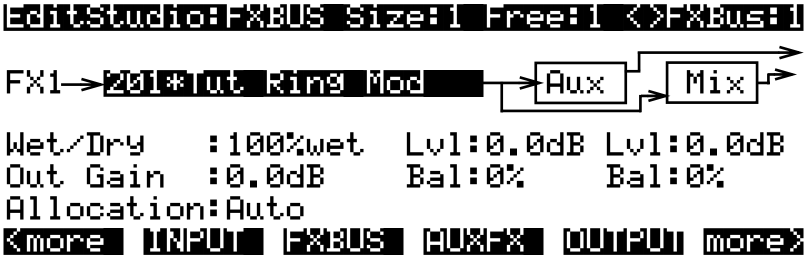
A more complex Studio

Press **Exit** to get to the Effects-mode page, and again to get to Program mode. Call up Program **201 ElecPno/Distgtr**. This is a split keyboard program: On Layer 1, which has been assigned to the KDFX-A outputs, is an electric piano, whose key range goes up to B4. On Layer 2, which goes to the KDFX-B outputs, is a distorted guitar, whose key range starts at C5.

Press the **Effects** mode button and make sure that FX Mode is **Master**, then call up Studio **201 RngMd/PFD/Plt**. Press **Edit** to look inside of this studio.

FXBus1

On the first FXBus is an FX preset called **201*Tut Ring Mod**. The algorithm this uses is a ring modulator, which is a processor that takes the sounds coming into it and combines them with static waveforms by adding and subtracting their frequencies, thereby creating interesting nonharmonic effects. Notice that this FX preset uses 1 PAU.



```
EditStudio:FXBUS Size:1 Free:1 <>FXBus:1
FX1→201*Tut Ring Mod    →Aux   Mix
Wet/Dry   :100%wet  Lvl:0.0dB Lvl:0.0dB
Out Gain  :0.0dB    Bal:0%    Bal:0%
Allocation:Auto
<more  INPUT  FXBUS  AUXFX  OUTPUT more>
```

As you play on the lower part of the keyboard, the arrow next to FX1 flashes, but as you play on the upper part it doesn't. That's because the upper part of the keyboard (the guitar sound) is routed to a different FXBus.

Bus Overrides

There are two FX preset parameters on this page: Wet/Dry mix and Output Gain. These parameters actually exist inside the FX preset, and are placed on this page so you can control them without editing the FX preset itself. These are bus overrides, as described earlier. You can change both the value of the override parameter *and* the name of the parameter that shows up in the bus override: to select a different parameter, simply highlight its name and scroll the Alpha wheel. As you do so, you will see the other parameters inside the FX preset that can be brought out to this page.

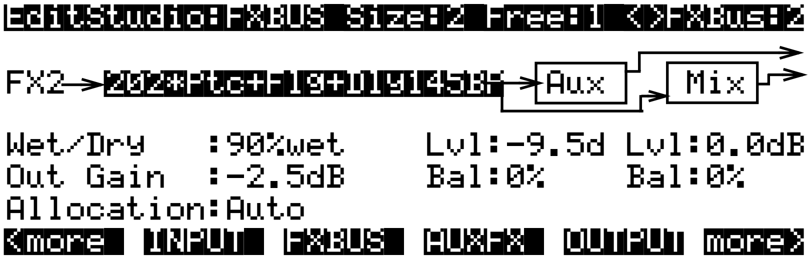
These overrides (that is, which parameters are available, and their values) are stored as part of the studio, not as part of the FX preset, and therefore you don't have to create new FX presets just because you want to change a couple of parameters. There are two bus overrides available for each of the four insert FXBuses and the Aux bus.

If you don't want any parameter control on this page, select the parameter's names, then scroll the Alpha wheel until you see **None**.

The Wet/Dry value is **100%**, and the Output Gain is **0.0dB**—unity. The signal on this FXBus goes both to the Aux bus and the Mix bus at unity gain (**0.0 dB**).

FXBus2

Press the **Chan/Bank Up** button to get to FXBus2.



```
EditStudio:FXBUS Size:2 Free:1 <>FXBus:2
FX2→202*Ptc+Flg+Dly145B→Aux   Mix
Wet/Dry   :90%wet    Lvl:-9.5d Lvl:0.0dB
Out Gain  :-2.5dB    Bal:0%    Bal:0%
Allocation:Auto
<more  INPUT  FXBUS  AUXFX  OUTPUT more>
```

On this bus is an FX preset called **Ptc+Flg+Dly145BP**, which uses a combination algorithm that has a pitch shifter, a flanger, and a delay all rolled into one. The **145BP** part refers to the fact that the delay times are based on a tempo of 145 BPM,

As you play the guitar sound, the arrow next to FX2 flashes, and it keeps on flashing as long as the various feedback delays are sounding. It doesn't flash when you play on the lower part of the keyboard.

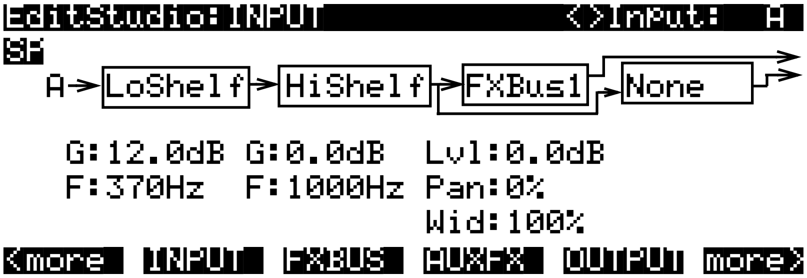
This FX preset uses 2 PAUs. Along with the 1 PAU in use on FXBus 1, this makes 3 of the 4 available PAUs accounted for, so the Free parameter has a value of **1**.

The output configuration of this FXBus has the signal going to the Aux bus attenuated by -9.5 dB, and going to the main Mix bus at unity gain.

The other two FXBuses are empty, which you can confirm by pressing the **Chan/Bank** buttons a few times.

Inputs

Now let's look at the inputs to the FXBuses. Press the **INPUT** soft button, and see this page:

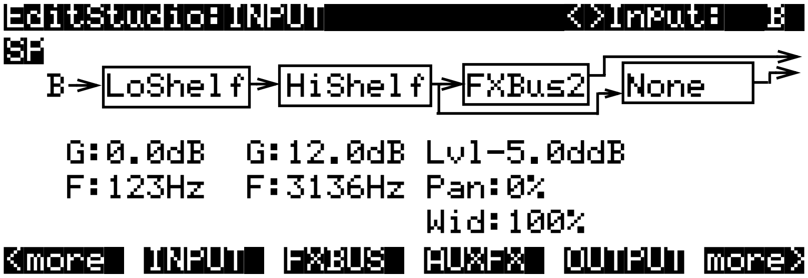


```
EditStudio:INPUT                 <>Input:   A
SF
  A->LoShelf->HiShelf->FXBus1->None

   G:12.0dB G:0.0dB  Lvl:0.0dB
   F:370Hz  F:1000Hz Pan:0%
                     Wid:100%
```

Input A carries the electric piano, coming from the program's KDFX-A outputs. Play on the piano part of the keyboard, and the arrow next to A flashes.

This input is configured to be stereo. It has a large bass boost: 12.0 dB of everything at 370 Hz and below, which adds a strong low-frequency emphasis to the signal being ring-modulated. Its signal is being sent only to the first FXBus. The stereo separation (Width) of the signal is at maximum.

Use the **Chan/Bank Up** button to go to Input B.



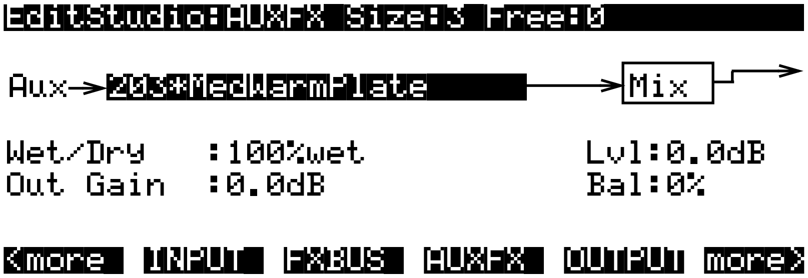
```
EditStudio:INPUT                 <>Input:   B
SF
  B->LoShelf->HiShelf->FXBus2->None

   G:0.0dB  G:12.0dB Lvl-5.0ddB
   F:123Hz  F:3136Hz Pan:0%
                     Wid:100%
```

This is the distorted guitar, coming from the program's KDFX-B outputs. It is also stereo. The incoming signal has a strong treble boost on it. It is sent directly to FXBus2 at -5 dB below unity gain (to compensate for the treble boost), and full width. Play on the guitar part of the keyboard, and the arrow next to B flashes.

The other two inputs, C and D, are not assigned to any FXBus.

AuxFX Bus

Now let's look at the Auxiliary FX Bus (Aux bus). Press the **AUXFX** soft button to look at its page.



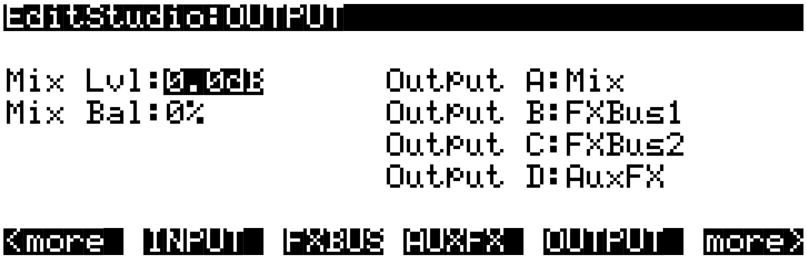
```
EditStudio:AUXFX Size:3 Free:0

Aux→203*MedWarmPlate ——→ Mix ——→

Wet/Dry    :100%wet               Lvl:0.0dB
Out Gain   :0.0dB                 Bal:0%
```

Here is an FX preset called **MedWarmPlate**, which is just what it sounds like: a medium-sized, warm-sounding plate reverb. It has two bus overrides, Wet/Dry mix, and Output Gain. The Aux bus is sent to the main Mix bus at unity gain.

Since both Insert FXBuses have signal going to the Aux Bus, the arrow next to Aux will flash as long as any signal processing is going on in either of the insert FXBuses.

Outputs

Finally, press the **OUTPUT** soft button to get to the OUTPUT page. Here we see that the four physical output pairs all are passing different parts of the studio. If the outputs are connected to an external mixer, you can treat each of them separately: recording them on different tracks of a tape deck, sending them to different outboard processors, or mixing them differently in a monitor mix.



```
EditStudio:OUTPUT

Mix Lvl:0.0dB         Output A:Mix
Mix Bal:0%            Output B:FXBus1
                      Output C:FXBus2
                      Output D:AuxFX
```

Output A has the Mix bus. This is the combined output of the two FXBuses, plus the reverb on the Aux bus. Its gain and balance are at unity.

Output B has the output of FXBus1, that is, the ring-modulated piano, without any reverb.

Output C has the output of FXBus2, the delayed/flanged/distorted guitar, without any reverb.

Output D has the output of the Aux bus, which is *just* the reverb signal, with no dry component, since the value of Wet/Dry on the AUXFX page is **100%**.

***Note**: If you're using the Mix audio outputs, keep in mind they carry the summed signals of audio outputs A through D. Normally you would assign each of the audio outputs differently. In the example above, you might set Outputs B through D to **Off**, or Output A to **Off**, to avoid overlapping assignments.*

Here is the overall structure for this studio:

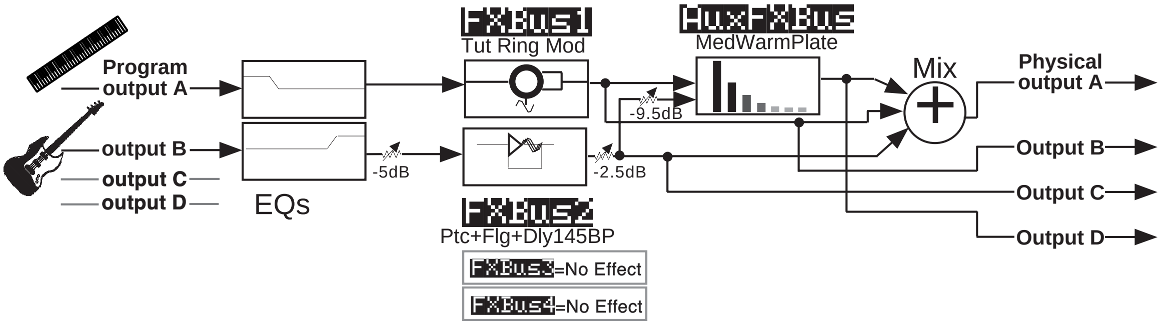


Figure 9-5 Structure of Studio 201

A Complex Studio With Real-time Control

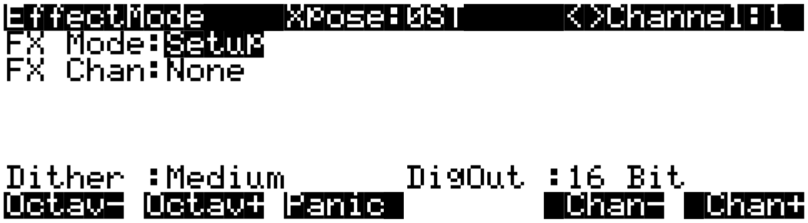
The third studio is a bit more complex, not least because it's under real-time control. Getting real-time control of a studio requires doing some advance work. Most noticeably, you have to define a program or setup that uses a particular studio, then define a set of FXMods in that program or setup—the FXMods provide the settings that link physical controllers like sliders to studio effects like wet/dry mix.

Equally importantly, the FX Mode parameter on the Effects-mode page must be set to a value of **Program** (if you want only programs to affect studio selection and controls) or **Setup** (if you want only setups to affect studio selection and controls)—or **Auto** if you want to affect studio selection and control whenever you select a program *or* a setup.

In this example, we'll work with *setup* control over a studio—the procedure for working with program control is almost exactly the same.

Setting the FX Mode

To enable real-time control of the studio from the setup, we have to put the K2600 in the right Effects mode. Go to the Effects-mode page, and set FX Mode to **Setup**. FX Channel automatically goes to **None**.



```
EffectMode    Xpose:0ST       <>Channel:1
FX Mode:Setup
FX Chan:None


Dither :Medium      DigOut :16 Bit
Octav- Octav+ Panic        Chan-  Chan+
```

If FX Mode was **Master** before you switched it to **Setup**, you'll notice that the Studio parameter disappeared when you switched. That's because when FX Mode is set to **Setup**, you can't view the current studio except from inside the setup. We do this so you can't make any changes in the studio that the setup doesn't know about

The Setup

Now let's look at the setup that we're going to use to control our studio. Press **Setup** to go into Setup mode, and select **200 KDFXCombo**.

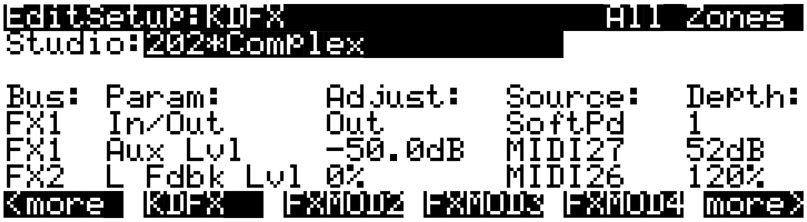
This is a four-zone setup, with bass and drums at the bottom, electric piano in the middle, and a breathy flute-like sound on the top. Each layer goes to a separate KDFX output pair, so they can all get different processing.

If you edit the setup and look at the various zones, you'll see this:

- Zone 1: **Gtr Jazz Band**, a layered bass and drum program, going to KDFX-A.
- Zone 2: **Dual Slap Bass**, also going to KDFX-A. This and the previous zone are active from the bottom of the keyboard up to A3.
- Zone 3: **Pno & Epno & Pad**, an electric piano and pad program, going to KDFX-C, which is active from A$^{\#}$3 to F$^{\#}$5.
- Zone 4: **Hybrid Vox**, going to KDFX-A, and active from G5 and up.

Looking at the Studio

To view the studio, press **Edit**, then **more>** three times, and then the **KDFX** soft button. The name of the studio associated with this setup, **202*Complex**, appears.



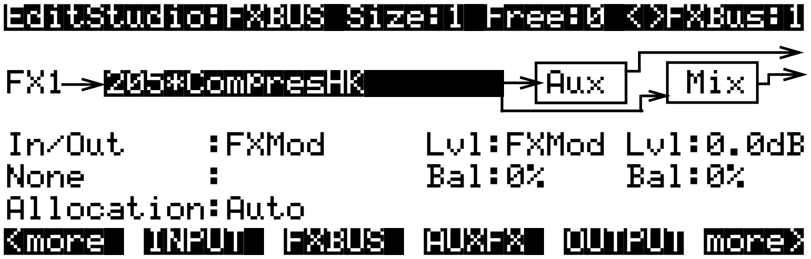
```
EditSetup:KDFX                    All Zones
Studio:202*Complex

Bus: Param:      Adjust:   Source:   Depth:
FX1  In/Out      Out       SoftPd    1
FX1  Aux Lvl     -50.0dB   MIDI27    52dB
FX2  L Fdbk Lvl  0%        MIDI26    120%
<more    KDFX    FXMOD2   FXMOD3   FXMOD4   more>
```

Highlight the studio's name and press **Edit**, and let's dig into this studio.

Press **INPUT** if you'd like to look at the INPUT pages. These are all set up straightforwardly, with Input A going to FXBus1, Input B going to FXBus2, etc.

On the FXBus Pages

Press **FXBUS** to look at the FXBuses. On FXBus1 is **205*CompresHK**, a hard-knee compressor.



```
EditStudio:FXBUS Size:1 Free:0 <>FXBus:1

FX1→205*CompresHK      →Aux    →Mix

In/Out      :FXMod      Lvl:FXMod Lvl:0.0dB
None        :           Bal:0%    Bal:0%
Allocation:Auto
<more    INPUT    FXBUS    AUXFX    OUTPUT    more>
```

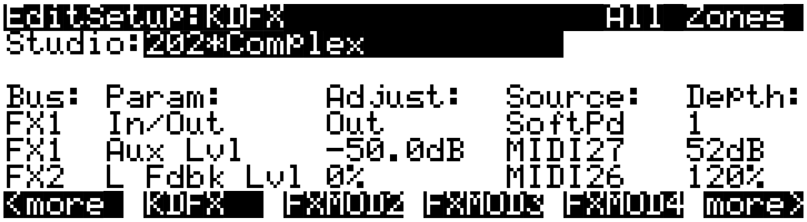
Look at the In/Out parameter. This is a bus override, meaning this parameter is actually inside the FX preset but it can be changed from here. However, instead of saying **In** or **Out** it says **FXMod**. This means that this parameter isn't controlled from inside the studio at all—it's controlled by something outside the studio.

As it happens, it's controlled by the soft pedal, MIDI Controller 67—Switch Pedal 3 for K2600 keyboard users. We'll see how this is done in a moment. Pressing this pedal causes the compressor to kick in, squashing the sound. The MakeUpGain inside the compressor is set to 6.0 dB, so the level doesn't change much when the compressor is engaged.

This type of control is useful on this program, which uses velocity-switching to change drum sounds. As you play harder, the drum sounds change, but they also get louder, which you may not want. With the compressor engaged, the drum sounds can change *without* getting louder.

There's another FXMod, which is assigned to the Aux bus send level. On the Aux bus is a reverb, so this FXMod controls how much of the signal coming through here will get reverb. It's under the control of Slider G, MIDI Controller 27. Play the bass and drums and move that slider, and hear the reverb go in and out.

How are these controls set up? That’s back on the KDFX page in the Setup Editor. Exit the Studio Editor (if you have made any changes, don’t save them—that will only confuse things!), then press the **KDFX** soft button. Take a look at the settings for the first FXMod:



```
EditSetup:KDFX                    All Zones
Studio:202*Complex

Bus: Param:      Adjust:  Source:  Depth:
FX1  In/Out      Out      SoftPd   1
FX1  Aux Lvl     -50.0dB  MIDI27   52dB
FX2  L Fdbk Lvl  0%       MIDI26   120%
```

The first FXMod affects the signal on the FX 1 bus, as shown by the Bus parameter. The Param parameter indicates which parameter is affected on the FX1 bus—in this case, the In/Out parameter. The Adjust parameter defines the initial condition of the parameter controlled on this bus—in this case, the In/Out parameter has an initial value of **Out** (disengaged).

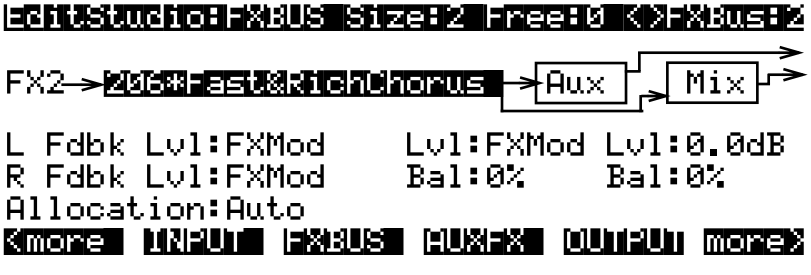
The Source parameter determines what MIDI Controller affects the In/Out parameter. For the first FXMod it’s **SoftPd** (that’s MIDI 67, which is Soft Pedal according to the MIDI specification, and which the K2600 uses as the default destination for Switch Pedal 3). While this setup is current, *any* K2600 controller—be it a wheel, slider, ribbon, or button—that’s programmed to send MIDI 67 will affect the In/Out parameter on the FX1 bus of this studio.

The Depth parameter defines how much the soft pedal affects the In/Out value, and here it is **1**.

For parameters with binary value, likes **In**/**Out** or **On**/**Off**, the Depth parameter can have only three values: **1**, **0**, or **-1**. If it is set to **1** or **-1**, then changing the state of the Source (in this case, depressing the soft pedal), changes the parameter’s state—in this case, going from Out to In. (If it’s set to **0**, then the Source has no effect on the value of the parameter.)

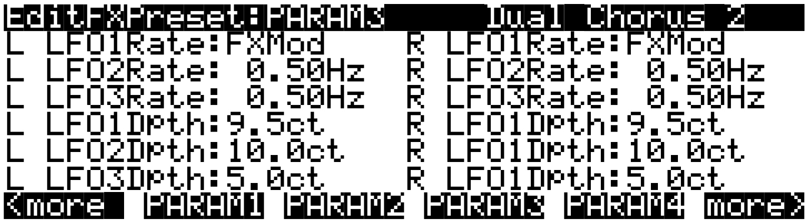
Now look at the second FXMod. It sets the Aux Lvl on the FX1 bus to an initial value **-50 dB**. The Source, **MIDI 27**, can raise that level by as much as 52 dB, to put it at +2 dB—and in fact, that’s what you’re hearing as you move Slider F, which defaults to MIDI 27.

Let’s go back into the Studio Editor and look at FXBus 2. Here’s our piano and string pad, going through an FX preset called **206*Fast&RichChorus**.



```
EditStudio:FXBUS Size:2 Free:0 <>FXBus:2

FX2 206*Fast&RichChorus

L Fdbk Lvl:FXMod     Lvl:FXMod Lvl:0.0dB
R Fdbk Lvl:FXMod     Bal:0%    Bal:0%
Allocation:Auto
```

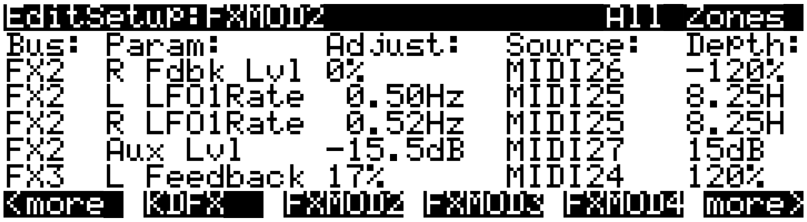
The two bus overrides are L Fdbk Level (Left Feedback Level) and R Fdbk Level, and they are both under FXMod control. The Aux Lvl, as on FXBus1, is also under FXMod control. Press **Edit** to go into the FX Preset Editor, then press PARAM3, you’ll see that the Left and Right LFO1 Rates are *also* under FXMod control.



```
EditFXPreset:PARAM3      Dual Chorus 2
L LFO1Rate:FXMod     R LFO1Rate:FXMod
L LFO2Rate: 0.50Hz   R LFO2Rate: 0.50Hz
L LFO3Rate: 0.50Hz   R LFO3Rate: 0.50Hz
L LFO1Dpth:9.5ct     R LFO1Dpth:9.5ct
L LFO2Dpth:10.0ct    R LFO1Dpth:10.0ct
L LFO3Dpth:5.0ct     R LFO1Dpth:5.0ct
<more  PARAM1 PARAM2 PARAM3 PARAM4 more>
```

Exit out of the FX preset and the studio, to the setup’s KDFX page (see the diagram on page 9-23). Here we see that on FXBus2, L Fdbk Lvl has a starting value of **0**%, and will change, using MIDI Controller 26 (Slider F), up to **120**%.

Press **FXMOD2** to go to the next page of FXMods, and you’ll see the same slider changing the R Fdbk Lvl, only in this case the Depth is **-120**%, meaning the feedback on this channel will be out of phase with the main signal.

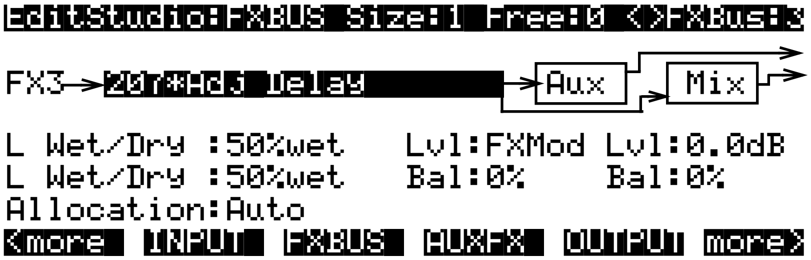


```
EditSetup:FXMOD2                All Zones
Bus: Param:      Adjust:   Source:   Depth:
FX2  R Fdbk Lvl 0%         MIDI26    -120%
FX2  L LFO1Rate  0.50Hz    MIDI25    8.25H
FX2  R LFO1Rate  0.52Hz    MIDI25    8.25H
FX2  Aux Lvl    -15.5dB    MIDI27    15dB
FX3  L Feedback 17%        MIDI24    120%
<more  KDFX   FXMOD2 FXMOD3 FXMOD4 more>
```

Further down the FXMOD2 page are the assignments to the LFO rates: MIDI Controller 25 (Slider E) is assigned to the left and right LFO1 Rates, with minimum values of **0.50** and **0.52 Hz**, respectively, and maximum change of **8.25 Hz**. Set the feedback level high and you can really clearly hear the LFO rate changing.

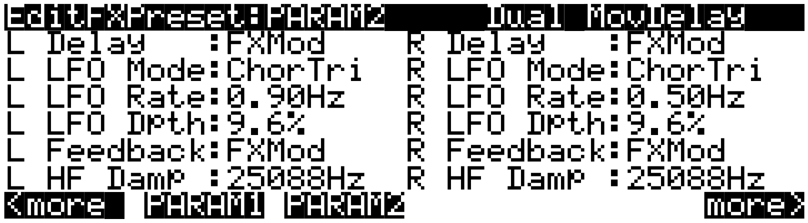
Finally on this bus, MIDI Controller 27, Slider G, controls the Aux Level send, thereby determining how much reverb will appear on the signal. We also used Slider G on the KDFX page to control the wet/dry mix for FXBux1. There are no limits to how many FXMods a single Source can be assigned to. (In fact, there are also no limits on how many different Sources can be assigned to the same parameter, and you may find uses for that feature somewhere—but not here.)

On to FXBus 3. Here’s where our breathy flute sound is, and it’s going through a delay FX preset called **207*Adj Delay**.



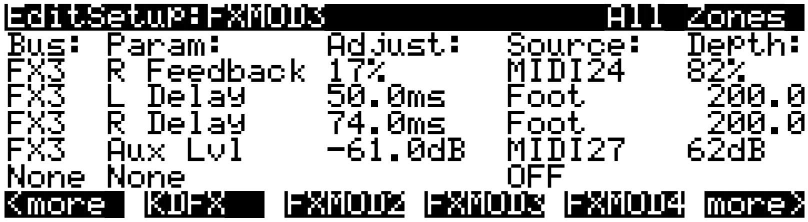
```
EditStudio:FXBUS Size:1 Free:0 <>FXBus:3

FX3→207*Adj Delay      →Aux   Mix

L Wet/Dry :50%wet   Lvl:FXMod Lvl:0.0dB
L Wet/Dry :50%wet   Bal:0%    Bal:0%
Allocation:Auto
<more  INPUT  FXBUS  AUXFX  OUTPUT  more>
```

Go into the FX Preset Editor, and press PARAM2, and on that page, you’ll see that the left and right Delay times and the left and right Feedback settings are all under FXMod control.



```
EditFXPreset:PARAM2       Dual MovDelay
L Delay    :FXMod     R Delay    :FXMod
L LFO Mode:ChorTri    R LFO Mode:ChorTri
L LFO Rate:0.90Hz     R LFO Rate:0.50Hz
L LFO Dpth:9.6%       R LFO Dpth:9.6%
L Feedback:FXMod      R Feedback:FXMod
L HF Damp :25088Hz    R HF Damp :25088Hz
<more  PARAM1 PARAM2                 more>
```

Go back out to the setup, and look at the bottom of the FXMOD2 page. Here we see the L Feedback parameter is under the control of MIDI Controller 24 (Slider D), with a minimum of **17**% and a maximum of 17+82=**99**%. Go to the FXMOD3 page and you’ll see the same assignment is in place for the R Feedback parameter.



```
EditSetup:FXMOD3                All Zones
Bus: Param:      Adjust:   Source:  Depth:
FX3  R Feedback  17%       MIDI24   82%
FX3  L Delay     50.0ms    Foot      200.0
FX3  R Delay     74.0ms    Foot      200.0
FX3  Aux Lvl     -61.0dB   MIDI27   62dB
None None                  OFF
<more  KDFX   FXMOD2 FXMOD3 FXMOD4 more>
```

Also on this page, the Left and Right Delay times are controlled by the Foot Pedal, controller 4—CC Pedal 1 for K2600 keyboard users. Here it varies the left decay time between **50** and **250 ms**, and the right decay time between **74** and **274 ms**.

Finally, the Aux Level from this bus, controlling the reverb send, is once again assigned to MIDI controller 27, Slider G.

So here’s a summary of the FXMods in this studio:

- Soft pedal puts compression on the bass and drums.
- Slider F controls the feedback on the chorus on the piano/string pad. Slider E controls the LFO speed on that chorus.
- Slider D controls the feedback on the delay on the breathy flute.
- Foot Pedal controls the delay time on the breathy flute.
- Slider G controls the reverb send for all three zones.

A Note About Effects in Setups

This applies primarily to K2600R users who are playing their units with a conventional MIDI keyboard. In order to play setups, which normally require a multi-zoned/multichannel keyboard like the K2600 or PC 88, the K2600R includes a parameter called Local Keyboard Channel (LocalKbdCh), which is found on the MIDI-mode RECEIVE page. If you set this parameter to the transmitting channel of the keyboard, then the K2600 takes all incoming MIDI data on that channel and treats it as if it was coming from a local keyboard—that is, it plays all the zones in the setup.

However, when you use this feature, the K2600 will respond to certain MIDI Controller messages coming in on this channel only if the Controller has been specified as a destination on the controller assignment pages in the setup (SLIDER, SLID/2, etc.)—and then, only if the setup assigns the customary physical controller to that MIDI Controller.

For example, Slider A is customarily assigned to Data (MIDI Controller 6), but if on the setup’s SLIDER page, Slider A has been reassigned to something else, then Data messages coming from the external keyboard on the local keyboard channel will not be recognized. Therefore, any MIDI Controller message that is not in its customary place in the setup cannot be used as a KDFX Source, because the setup will not recognize it. See Table 10-1 on page 10-8 for a list of physical controllers affected by using the local keyboard channel.

This is not an issue in Program mode, or when using the K2600 with a sequencer. In those cases the LocalKbdCh should be set to **None**, and there is no restriction on the MIDI data passing through. More on this in Chapter 15.

Building your own

Now that we’ve walked you through a complex studio, here are some hints on how to go about building your own.

First, you want to decide whether you want the studio to be static, or to be dynamic under FXMod control from the K2600 keyboard and/or an external MIDI source. If it’s to be static, the value of FX Mode must be **Master**, while if it’s to be dynamic, the value of FX Mode must be **Program**, **Setup**, or **Auto**, depending on how you want to approach it.

Then you need to look at how your program outputs are arranged, so you can design the studio intelligently. If you’re using internal setups, look at the output assignments on the various zones, and change them so that sounds that need different effects are separated, and those that can use the same effects are grouped. If you’re using a MIDI sequencer, you might want to use the Channel output overrides in MIDI mode, and arrange your sequence so that the assignment of tracks to FXBuses is determined by the tracks’ MIDI channel assignments.

Now set up your studio, assigning FX presets to the program outputs/KDFX inputs. Many of the ROM studios follow a common organizational plan, which might be a good starting point for your studios:

- FXBus 1 contains a relatively simple reverb with a low Size requirement.
- FXBus 2 contains an effect that does not increase the "length" of the sound (that is, no reverb or delay), something like chorus, flange, phaser, distortion, shaper, pitcher, enhancer, EQ, or EQ morpher.
- FXBus 3 contains effects that take up lots of time, such as delays, delays with reverb, or other "Lead" sounds.
- FXBus 4 is dry, since the first three FXBuses have probably used up all the PAUs.
- The AUXFXBus contains a larger reverb (Size: 2 or3), a compressor, or a graphic EQ. It can often be used instead of an FXBus reverb, such as the one on FXBus1. If you use it in this way (set the Aux Lvl on FXBus 1 to 0dB or higher), it frees up FXBus1 for use as an Enhancer, Stereo Image, Flanger, etc.

Finally, set up your FXMods in the setup or program you plan to use. If you're using a sequencer, you might want to dedicate a program *just* to the studio and FXMods, with no sound coming from that program.

Keep in mind that FXMods don't always have to be dynamic—they can be used to assign static (Source: OFF) values to the parameters in a studio that are different from the studio's normal parameters. This lets you create and store multiple variations on a studio without making each one a separate studio—when you want to call up the variation, merely call up the program that contains the correct FXMods.

Don't forget to Save your Studio!

And don't forget to have fun. Check out Chapter 15 for an in-depth look at Effects mode.

Chapter 10
MIDI Mode

Press the MIDI-mode button to enter MIDI mode. There are three pages in MIDI mode:

- TRANSMIT (XMIT soft button)
- RECEIVE (RECV soft button)
- CHANNELS (CHANLS soft button)

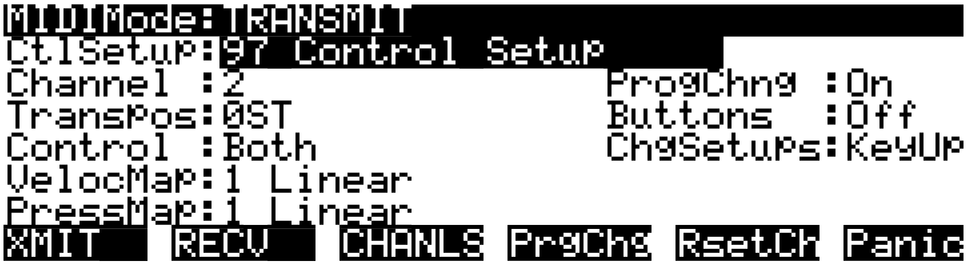
You’ll use these pages to determine what MIDI messages the K2600 transmits, and how it responds to the MIDI messages it receives—as well as how each MIDI channel behaves.

When you enter MIDI mode, you’ll see one of the three available MIDI-mode pages. When you exit MIDI mode, the K2600 remembers which page you were on. The next time you select MIDI mode, that page appears.

The TRANSMIT Page

Press the **XMIT** soft button, and the TRANSMIT page appears. Use these parameters to control how the K2600 sends MIDI information to its MIDI Out port. These settings to some extent affect the K2600’s response to its own keyboard and controllers, but they primarily affect the responses of other MIDI devices that are receiving MIDI from the K2600 on the channel specified with the Channel parameter on this page.

It’s important to remember that many of the settings of the TRANSMIT page are in effect only when a *program* is selected, either in Program mode or in Quick Access mode. If a *setup* is selected, in Setup mode or in Quick Access mode, the setup’s MIDI settings override the corresponding settings on the TRANSMIT page. The TRANSMIT page looks like this:



```
MIDIMode:TRANSMIT
CtlSetup:97 Control Setup
Channel :2                ProgChng :On
TransPos:0ST              Buttons  :Off
Control :Both             ChgSetups:KeyUp
VelocMap:1 Linear
PressMap:1 Linear
```

Parameter	Range of Values	Default
Control Setup	Setup list	97 Control Setup
Channel	1 to 16	1
Transposition	±60 semitones	0
Control	Both, MIDI, Local	Both
Velocity Map	Velocity Map list	1 Linear
Pressure Map	Pressure Map list	1 Linear
Program Change	Off, On	On
Buttons	Off, On	Off
Change Setups	Immed, KeyUp	KeyUp

Rack-model owners, keep in mind that only the Program Change Type, Program Change, and Buttons parameters operate as programmed if the LocalKbdCh parameter on the RECEIVE page does not match the transmit channel of your MIDI controller. If you set them to match, the remaining parameters take effect as well.

Control Setup

This is where you select the current control setup, Zone 1 of which sets the physical controller assignments for all programs while you're in Program mode. Refer to Chapters 6 and 7 for more information on the control setup.

***Note**: The control setup is responsible for many of the settings that each had separate parameters on the K2000 and K2500. If you've worked with one of these instruments, you'll need to relearn how MIDI transmission controls are defined. Instead of setting a number of parameters on the MIDI TRANSMIT page, you'll simply select a control setup. Zone 1 of that setup then determines the controller assignments (as specified on the SLIDER, RIBBON, WHEEL, and other pages in the Setup Editor).*

Channel

This defines which MIDI channel the K2600 uses to transmit MIDI messages. The value for this parameter matches the current MIDI channel displayed on the top line of the Program-mode page. If you change the current MIDI channel while in Program mode, the setting of this parameter changes accordingly, and vice versa.

Transpose

This parameter affects the transposition that's applied to the MIDI data stream. Adjusting this parameter transposes the K2600's notes, as well as notes on slaves receiving from the K2600. This transposition setting is not overridden when you use Setup mode, but is *added* to the transposition settings for the currently selected setup.

Control

Here you determine where the K2600 sends MIDI information. A value of **MIDI** sends the MIDI signal to the K2600's MIDI Out port, but not to the K2600 itself. This is also known as Local Control Off.

If you're using your K2600 with a MIDI sequencer and have a MIDI loop (K2600's Out to sequencer's In, and vice versa), you'll need to select a value of MIDI when your sequencer's

Patch Thru feature—also known as Play Thru, and Soft Thru—is on. This will prevent the K2600's MIDI signal from looping back on itself, which can cause problems. If you deactivate your sequencer's Patch Thru feature, set the Control parameter's value to **Both**, and the K2600 will play normally. Also, you may want to set the value of the LocalKbdCh parameter to **None** when you have a MIDI loop, because you can have problems with doubled notes and MIDI overload. You won't have problems, however, as long as the channels transmitted by the K2600 are all different from the incoming MIDI channel.

A value of **Local** disables the MIDI Out port. Use this setting when you want to play the K2600, but not to send any MIDI information to other MIDI instruments (local control only). A value of **Both** (the default) enables you to play the K2600 and send MIDI information from its MIDI Out port.

Velocity Map—Transmit (VelocMap)

The transmit velocity map affects the way the K2600 sends velocity information to its MIDI Out port. Different maps generate different velocity values for the same attack velocity—that is, they apply different curves to the attack velocities the K2600 receives and remap them to new velocities before transmitting them to the MIDI Out port.

Important: The MIDI velocity maps affect only those MIDI velocity values transmitted via the K2600's MIDI Out port, and are used exclusively to adjust the response of MIDI devices connected to the Out port. If you have a DX7 connected to your K2600, for example, and the DX is distorting, selecting a transmit Velocity Map like **Hard2** should handle the problem. Changing the velocity map on this page does not affect the response of the K2600's sound engine to its own keyboard, or to an external MIDI controller. That's done on the RECEIVE page. See Chapter 18 if you're interested in editing velocity maps.

Also important: Both the transmit and receive velocity maps should be left at values of **Linear** unless you really need to change them. The linear maps give you the most consistent results.

If you have a keyboard model of the K2600, keep in mind that the setting of the Veltouch parameter in Master mode also has an effect on the transmit velocity map.

Pressure Map—Transmit (PressMap)

This is like the VelocMap, but it controls the aftertouch values sent by the K2600 to its MIDI Out port. Use this exclusively to adjust the response of MIDI devices connected to the K2600's MIDI Out port. Changing the pressure map on this page does not affect the response of the K2600's sound engine to its own keyboard, or to an external MIDI controller. That's done on the RECEIVE page. See Chapter 18 for information about editing pressure maps.

Program Change (PChng)

When On, the K2600 sends program change commands to its MIDI Out port when you select programs or setups from the front panel or from your MIDI controller. Select a value of **Off** when you want to change programs on the K2600 but don't want to send program change commands to the MIDI Out port. This parameter doesn't affect the *type* of program change command that's sent; it just determines whether any command is sent at all. (The type of program change command is determined by the settings for three parameters on the CH/PRG page in the Setup Editor.)

Buttons (Bttns)

If you set the value of the Buttons parameter to **On**, the System Exclusive (SysEx) messages generated by your button presses are sent to the MIDI Out port. This enables you to do two things: control a remote K2600 (or earlier model), and record sequences of programming button presses to a sequencer or SysEx software package.

If you have the MIDI In port of another K2600 (or K2500, or K2000) connected to the first one's MIDI Out port, the second instrument will respond to every button press on the first instrument, just as if you were pressing the buttons of the second one. Keep in mind that both devices must be in exactly the same state (the same page in the same mode, with identical lists of RAM objects) when you start. Otherwise the button presses you make on the first instrument may execute other functions on the second instrument.

Much more useful is to send streams of button presses to your sequencer. When you dump them from your sequencer back to the K2600, the K2600 responds as if the buttons were actually pressed. This enables you to set up a variety of "macros," which are strings of commands that can be executed all at once by a single initial command. For example, you can record a sequence of button presses that enters Disk mode, selects a specific SCSI device, and loads one or more banks of samples while you do something more entertaining. Again, it's important to keep in mind that the state of your K2600 must be identical to its state when you recorded the sequence of button presses. If you've added or deleted any objects stored in RAM, for example, the sequence of button presses will select different objects when you play back the button press sequence.

***Note**: Make sure this parameter is set to **Off** before you initiate a SysEx dump of any kind. If this parameter is **On** when you start a dump, the buttons you press to begin the dump will also generate SysEx messages.*

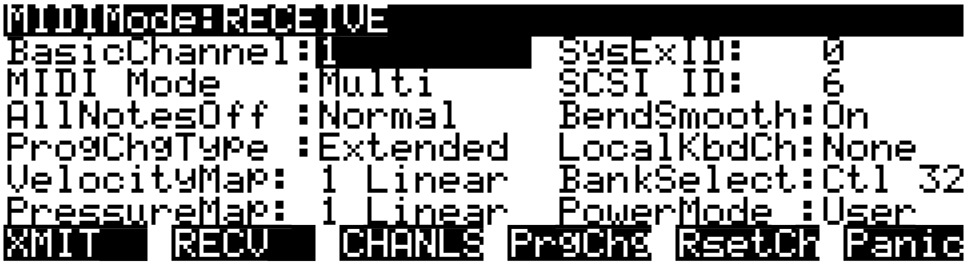
Change Setups (ChgSetups)

This parameter determines the exact timing of setup changes when you select a different setup—either by a normal data entry method or via MIDI program change commands.

Choose **KeyUp** to indicate that you want setup changes to take place only when you've released all currently held notes. Choose **Immed** to indicate that you want such changes to happen immediately when you select the setup.

The RECEIVE Page

Press **RECV** to select the RECEIVE page, where you define the K2600’s response to incoming MIDI signals (with one exception pertaining to Quick Access mode, which we’ll explain later).



```
MIDIMode:RECEIVE
BasicChannel:1        SysExID:    0
MIDI Mode   :Multi    SCSI ID:    6
AllNotesOff :Normal   BendSmooth:On
ProgChgType :Extended LocalKbdCh:None
VelocityMap: 1 Linear BankSelect:Ctl 32
PressureMap: 1 Linear PowerMode :User
XMIT   RECV   CHANLS PrgChg RsetCh Panic
```

Parameter	Range of Values	Default
Basic Channel	1 to 16	1
MIDI Mode	Omni, Poly, Multi	Multi
All Notes Off	Normal, Ignore	Normal
Program Change Type	Program Change Type List	Extended
Velocity Map	Velocity Map list	1 Linear
Pressure Map	Pressure Map list	1 Linear
System Exclusive ID	0 to 127	0
SCSI ID	0 to 7	6
Bend Smooth	On, Off	On
Local Keyboard Channel	None, 1 to 16	None
Bank Select	0 only, 32 only, Ctl 0, Ctl 32	Ctl 32
Power Mode	User, Demo	User

Basic Channel

The basic channel determines which channel will always be available to receive MIDI information. Depending on the MIDI receive mode (below), the Basic channel may be the only receiving channel, or one of several.

MIDI Receive Mode (MIDI Mode)

The MIDI Mode parameter determines the MIDI receiving capabilities of the K2600. When set to **Omni**, the K2600 responds to incoming MIDI events on all MIDI channels, and plays them on the current channel. This is normally used for diagnostic purposes only.

At a setting of **Poly**, the K2600 responds only to events that are sent on the same channel as the K2600’s current MIDI channel (the one displayed on the top line of the Program-mode page). In Poly mode, the currently selected channel is always the basic channel, so if you change channels, the basic channel changes accordingly.

With a value of **Multi** (the default), the K2600 responds to events on all active channels. This is the mode you’ll use when you’re driving the K2600 with a sequencer, since you can play a different program on each channel. At this setting, you can turn individual channels on and off (on the CHANNELS page, described later in this chapter).

All Notes Off

If this parameter's value is set to **Normal**, the K2600 responds to All Notes Off messages received over MIDI. **Ignore** causes these messages to be ignored. If you're using a Roland product as a MIDI controller for your K2600, you'll want to set the value of this parameter to **Ignore**. This is because some older Roland products occasionally send an All Notes Off message when no keys are held down—even if you're sustaining notes with a pedal. You might find all your sustains missing from your sequence, for example, if you're driving your K2600 from one of Roland's hardware sequencers. Setting this parameter to **Ignore** takes care of this problem.

Regardless of the setting for this parameter, the K2600 always responds to its own **Panic** button by shutting off all active notes and controllers.

Program Change Type (ProgChgType)

This determines how the K2600 responds to program change commands received via MIDI. See *Program Change Formats* on page 10-11 for an explanation of the various values available for this parameter.

Velocity Map—Receive

The velocity map applies a preset curve to incoming velocity messages. It maps incoming velocity levels to new levels that correspond to the eight dynamic levels used by the VTRIGs and keymaps for velocity level selection. See Chapter 18. Normally you'll leave this set to **1 Linear**. Adjust this parameter's value only when you need to alter the K2600's response to the velocity messages from a MIDI controller, for example, if you're getting too much or too little volume when you play, or when you're using a sequencer to drive the K2600.

Pressure Map—Receive

Like the velocity map, this determines how the K2600 responds to incoming pressure (aftertouch) messages.

System Exclusive ID (SysExID)

The SysExID parameter differentiates between more than one MIDI device of the same model. You won't need to change the default setting of 0 unless you have multiple K2600s (or K2500s or K2000s) receiving SysEx messages from a single source. In that case, make sure each instrument has a different SysExID. Then you can direct SysEx messages to the appropriate K2600 with the SysExID byte that's included with every SysEx message. A value of **127** specifies "Omni Receive." That is, at this value, a K2600 responds to a SysEx message regardless of the SysEx ID of the message (as long as the manufacturer and device IDs match—see Chapter 7 of the *Musician's Reference* for more information about System Exclusive messages).

SCSI ID

Use this parameter to change the SCSI ID of your K2600. You can ignore this parameter unless you've connected a SCSI device (external SCSI disk or CD-ROM drive) to the K2600's SCSI port. You can use either or both SCSI ports to chain up to seven SCSI devices to the K2600 (a total of eight devices can be chained together); just be sure to set each one to a different SCSI ID. Most SCSI devices available today make it easy to change their SCSI IDs, so you may not have to adjust this parameter even if you have several SCSI devices connected. See Chapter 13 for more information about using SCSI devices.

Bend Smooth

This parameter can improve your K2600's performance when you're driving it from a MIDI guitar controller. Its default value is **On**.

You may find that pitch bending seems to carry over from the previous note to the next note, causing it to start on the wrong pitch. This is probably due to the automatic pitch smoothing provided by the K2600. If this is happening, try setting the BendSmooth parameter to a value of **Off**.

Local Keyboard Channel (LocalKbdCh)

Note: *Changing the setting of the Local Keyboard Channel parameter is useful only when your K2600 is receiving MIDI information from an external source—maybe you have a rack model 2600, or you have a favorite MIDI keyboard that you use to control all the gear in your studio, or you use a lot of outboard sequencing. If you're using the K2600 as a standalone music workstation or performance keyboard, you can ignore this parameter.*

The available values for this parameter are **None**, and **1–16**. The default is **None**, which disables the local keyboard feature, since you may not want to send your MIDI controller's MIDI information to devices connected to the K2600R's MIDI Out port. If you want to use Setup mode on the K2600R, however, you'll probably want to change the setting of this parameter. This is because LocalKbdCh enables you to take advantage of the K2600's eight setup zones, even if your MIDI controller transmits on only one MIDI channel at a time.

It's important to understand that a setup is a control-oriented object, designed to transmit note and control information. On the keyboard version of the K2600, the keyboard itself transmits on up to eight channels when in Setup mode. But with the rack, if your keyboard sends information on only one MIDI channel, you need a way to distribute that information from one channel to eight channels. This is what the LocalKbdCh parameter does. It takes the signal coming in via one channel and turns it into different information, depending on where you are in the K2600.

The local keyboard channel changes the way the K2600 performs in other modes as well. It changes the incoming information depending on what you have displayed in the K2600. For example, if LocalKbdCh is **1**, and you're in Program mode with Channel 5 in the display, then the information coming in on Channel 1 gets remapped to Channel 5, and you hear the program assigned to Channel 5. But if you set LocalKbdCh to **None**, then if you send on Channel 1, you hear the program that is assigned to Channel 1, even if you're looking at Channel 5.

Local Keyboard does more than just change the MIDI channel. When Local Keyboard Channel is set to **None**, you will notice that the **Octav–** and **Octav+** soft buttons found in Program, Setup, Quick Access, and Effects modes do not function. If you use the Local Keyboard Channel parameter, however, they function correctly. And you can even use it to change one type of MIDI Controller number to another.

Here's how it works. The K2600 receives MIDI information on the channel that corresponds to the value you set for this parameter, and relays it to its MIDI Out port, using the MIDI channels currently shown in the display. If you're in Program mode (or in Quick Access mode with a program selected), the K2600 relays the LocalKbdCh MIDI information to the channel to which the program is assigned. If you're in Setup mode (or in Quick Access mode with a setup selected), the K2600 relays the LocalKbdCh MIDI information to all the channels currently used by the setup.

The K2600 also remaps certain MIDI Controller messages so that they correspond (in most cases) to the default assignments for the K2600's physical controllers (as listed in Table 10-1).

This enables you to receive the messages listed in the first column of Table 10-1 from a MIDI source, and remap them to whatever values are available for the parameters listed in the second column. The third column lists the Studio-editor page where each parameter appears. Note that when the Large Ribbon is configured in one section, you cannot remap incoming MIDI Controller messages through it.

The MIDI Controller number received from your MIDI source…	…gets sent to the MIDI Controller number assigned as the value for this parameter…	…which is on this page in the Studio Editor
Mono Pressure	Press	PRESS
Modulation (01)	ModWhl	WHEEL
Breath (02)	CPedal2	CPEDAL
Foot (04)	CPedal1	CPEDAL
Data (06)	Slider A	SLIDER
Ctl A (16)	Small Ribbon Position	RIBBON
Ctl B (17)	Small Ribbon Pressure	RIBBON
Ctl C (18)	Large Ribbon Section 1	RIBBON
Ctl D (19)	Large Ribbon Section 2	RIBBON
MIDI 20	Large Ribbon Section 3	RIBBON
MIDI 22	Slider B	SLIDER
MIDI 23	Slider C	SLIDER
MIDI 24	Slider D	SLIDER
MIDI 25	Slider E	SLID/2
MIDI 26	Slider F	SLID/2
MIDI 27	Slider G	SLID/2
MIDI 28	Slider H	SLID/2
Sustain (64)	FtSw1	FOOTSW
Sostenuto (66)	FtSw2	FOOTSW
Soft Pedal (67)	FtSw3	FOOTSW
Legato Switch (68)	FtSw4	FOOTSW
Freeze Pedal (69)	Switch 1	SWITCH
MIDI 70 (70)	Switch 2	SWITCH

Table 10-1 Controller Remapping Via Local Keyboard Channel

Here’s a more specific example. Suppose your MIDI controller transmits on MIDI Channel 1, and you’ve set the LocalKbdCh parameter to a value of **1**. You’ve also set the MWhl parameter on the WHEEL page in the Setup Editor to a value of **Volume** (MIDI 07) for each setup zone. Then you’ve selected a setup that uses MIDI Channels 1, 2, and 3. When you send a Modulation message (MIDI 01) from the MIDI controller, you’ll affect the K2600’s volume (unless the VolLock parameter for Channel 1 is on), and the K2600 will send a Volume message to its MIDI Out port, on Channels 1, 2, and 3.

Bank Select

BankSelect allows you to choose between having the K2600 respond to Controller 0 or Controller 32 or both. The reason for this is that various manufacturers have chosen one method or the other. The four possible values for this parameter are:

0 only	Responds to controller 0 only.
32 only	Responds to controller 32 only.
Ctl 0	Responds to 0 or 32.
Ctl 32	Responds to 0 or 32.

Power Mode

Power Mode has two possible values: **User** and **Demo**. When set to **User**, the user's parameter settings are retained. When set to **Demo** several parameters are returned to default values when the unit is powered up. The default value for the Power Mode parameter is **User**.

The following parameters are reset when Power Mode is set to Demo.

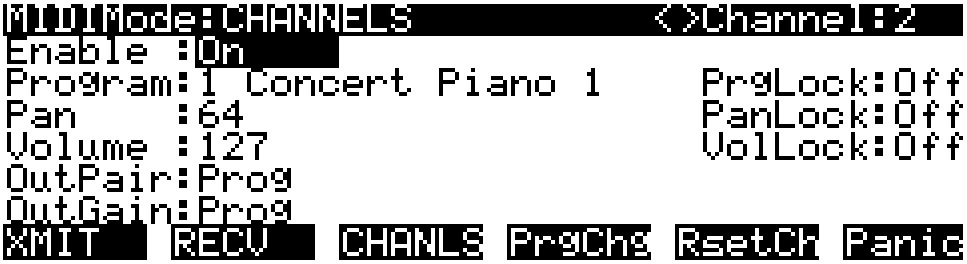
Master mode:	KB3 Channel resets to **1**
MIDI Transmit:	Control resets to **Both**; Channel resets to **1**; Transpose resets to **0 ST**
Effects mode:	FX Mode resets to **Auto**, and FX Chan resets to **Current**
Disk mode:	Current Disk resets to **Floppy**

The Channels Page

Press the **CHANLS** soft button to select the CHANNELS page, where you can define numerous parameters for each MIDI channel independently. Use the **Chan/Bank** buttons to select the MIDI channel you wish to work on.

The CHANNELS page is very useful when you're doing multi-timbral sequencing, with programs assigned to numerous MIDI channels. The CHANNELS page lets you set several control characteristics for each MIDI channel. This makes it easy to adjust the playback of the sequence without editing the sequence itself. For example, you might turn off the Enable parameter for one or more channels to mute the tracks on those channels. You could also set the

VolLock parameter to **On**, to ignore any MIDI volume messages the K2600 receives on a given MIDI channel.



```
MIDIMode:CHANNELS                 <>Channel:2
Enable :On
Program:1 Concert Piano 1         PrgLock:Off
Pan    :64                        PanLock:Off
Volume :127                       VolLock:Off
OutPair:Prog
OutGain:Prog
XMIT   RECV   CHANLS PrgChg RsetCh Panic
```

Parameter	Range of Values	Default
Enable	Off, On	On
Program	Program list	Program ID 1
Pan	0 to 127	64 (centered)
Volume	0 to 127	127 (maximum)
Output Pair	Prog, KDFX-A to KDFX-D	Prog
Output Gain	Prog, –12 to 30 dB in 6 dB increments	Prog
Program Lock	Off, On	Off
Pan Lock	Off, On	Off
Volume Lock	Off, On	Off

Enable

Use this parameter to turn the currently selected channel on or off. When on, the channel will receive MIDI information, and the settings of the parameters on the MIDI CHANNELS page will be in effect. When off, the channel will ignore all MIDI information.

Program

Use this parameter to assign a program to the currently selected channel. The channel will still respond to program change commands received via MIDI, unless the PrgLock parameter (described below) is set to **On**.

Pan

This offsets the pan position of the current program as set on the OUTPUT page in the Program Editor. A value of **0** is maximum offset to the left, **64** is no offset, and **127** is maximum offset to the right. Changing the value of this parameter is like inserting a MIDI pan message. MIDI Pan (MIDI 10) messages will change the value of this parameter, unless the PanLock parameter (described below) is set to **On**.

If the Mode parameter on the OUTPUT page in the Program Editor is set to Fixed, changing the value of Pan on the CHANNELS page in MIDI mode has no effect.

Volume

This sets the volume for any program assigned to the currently selected channel. A value of **0** is silence, and a value of **127** is full volume. The value of this parameter will change in response to MIDI Volume (MIDI 07) messages, unless the VolLock parameter (described below) is set to **On**.

Output Pair (OutPair)

This parameter sets the audio output group for the program assigned to the currently selected channel. The default value of **Prog** means that the output group is determined by the program's value for the Pair parameter on the OUTPUT page in the Program Editor. In this case, the channel's output group changes depending on the program assigned to it, with the output group being routed on a per layer basis within the program. Values of **KDFX-A**, **KDFX-B**, **KDFX-C**, or **KDFX-D** fix the output group regardless of the program that's assigned to the channel.

Output Gain (OutGain)

OutGain boosts or cuts the level at the audio outputs for any program assigned to the currently selected channel. This allows you to make a program louder or softer without having to edit the program.

Program Lock, Pan Lock, Volume Lock

When the parameter locks are set to **On**, the three parameters they control do not respond to their respective MIDI controller messages. In that case, you could change the Program, Pan, and Volume settings from the front panel, but not via MIDI.

Program Change Formats

The K2600 can store more programs than the MIDI program change specification can handle (MIDI lets you send program change numbers from 0 to 127 or 1 to 128 only). So we've designed a system that makes program selection more flexible. This is true whether you're selecting programs from the K2600's front panel, or via MIDI.

Program Change Type	For Use With:
Extended	Other K2600s (or K2500s or K2000s) similarly set, plus all other instruments that use the Bank Change controller
Kurzweil	K1200s, and 1000s with version 5 software
0–127	Older MIDI devices that transmit program change commands in the range from 0–127 only.
QA Bank E	Other K2600s (or K2500s or K2000s) similarly set, when in Quick Access mode
QA Bank K	K1200s and v5 1000s, when in Quick Access mode
QA 0–127	With the K2600 in Quick Access mode, when using it with older MIDI devices

First of all, the K2600's programs (and all of its objects) are numbered and grouped according to a decimal system, that is, in multiples of ten. This is much easier to keep track of than the binary-oriented groupings of many synths, which feature banks of 8, 16, or 64 programs.

Next, the K2600 gives you 999 program change numbers to work with. These are organized into ten banks of 100 each (the memory banks). A program's object ID is its program change number, as discussed on page 5-2. This makes it easy to keep track of your programs. The K2600 can use several different formats for interpreting program change commands. The value for the ProgChgType parameter on the RECEIVE page determines which format is used, and the one you should select depends on your MIDI system.

If you expect you'll always change programs from your K2600's front panel, you can finish this paragraph and skip the next few sections. In this case, selecting programs is as simple as entering the program change number (the program's object ID) on the alphanumeric pad, and pressing **Enter**. Even program numbers above the usual MIDI limit of 127 can be selected this way.

Extended and Kurzweil Program Change Types

In the early days of MIDI, most instruments had small numbers of memory locations, usually 32, 64, or 128. As instruments began to have more memory locations, however, users ran against the limitation of only 128 values for program changes in the MIDI spec. Because of this, Bank Change Controller was added, allowing users to switch between banks of up to 128 programs per bank.

Previous to the addition of the Bank Change Controller, Kurzweil had developed their own method of switching banks by using two program changes, one to switch the bank, the second to call up the program within the bank (as described below). The K2600 can respond to either the Bank Change controller or the double-program-change method. In a nutshell, the difference between the **Extended** setting and the **Kurzweil** setting is this: In **Extended**, the K2600 will receive and respond to the Bank Change controller. When set to **Kurzweil**, the K2600 will receive only the double-program-change method.

Extended Program Changes

If you're controlling your K2600 from a MIDI device that can handle the MIDI Controller 0 or 32 program-change format, you'll have the greatest flexibility if you set the ProgChgType parameter to a value of **Extended** (or **QA Bank E**, but that explanation comes later).

When you're using the extended program change format, then depending on the value of the BankSelect parameter on the RECEIVE page in MIDI mode, the K2600 will respond to either MIDI Controller 0 or 32 program change commands for bank selection (Zeros through 900s), and standard program change commands for program changes within the current bank. Different values have different results, as shown in the following table:

Program Change Command Type	Value of Message	Result
MIDI controller 0 or 32 (MC 0 or MC 32)	0 to 9	Selects memory bank zeros–900s
	10 to 127	Ignored
Standard (PCH)	0 to 99	Selects correspondingly numbered program in current memory bank
	100 to 127	Selects correspondingly numbered program in next-highest bank

If your K2600 is already in the memory bank you want to use, you can send it single PCHs from 0 to 99, to select programs within that memory bank. The K2600's response depends on the setting for the MIDIBankMode parameter on the CH/PRG page in the Setup Editor. If you want to change the memory bank, the K2600 must receive either an MC 0 or 32 message with value

0–9. The next PCH in the range 0–99 will select the correspondingly numbered program in the newly selected bank. The following table of examples should help make it clear.

Bank Change Command Received	Program Change Command Received	Result
MC 0 or 32: value 0	PCH: value 99	Program 99 (0s bank, 99th program)
MC 0 or 32: value 1	PCH: value 42	Program 142 (100s bank, 42nd program)
MC 0 or 32: value 1	PCH: value 120	Program 220 (200s bank, 20th program)
MC 0 or 322: value 9	PCH: value 0	Program 900 (900s bank, 0th program)
MC 0 or 32: value 9	None	900s bank selected, no change in current program (bank selection is pending for next PCH)
MC 0 or 32: value 10	PCH: value 99	MC 32 message ignored; 99th program in current bank selected (for example program 199 if in 100s bank)

Kurzweil Program Changes

When you use the Kurzweil program change format, you'll use PCH messages to select different memory banks, followed by a second PCH command to change the program within the current bank, as the following examples demonstrate. You'll want to use this format if you're controlling your K2600 from a Kurzweil 1000- or 1200-series instrument.

1st Program Change Command Received	2nd Program Change Command Received	Result
PCH: value 39	None	39th program in current bank selected
PCH: value 99	PCH: value 27	27th program in current bank selected (99 is selected, then overridden by 27)
PCH: value 102	PCH: value 16	Program 216 (200s bank, 16th program
PCH: value 105	PCH: value 44	Program 544
PCH: value 109	PCH: value 0	Program 900
PCH: value 127	PCH: value 99	99th program in current bank (1st PCH is ignored, since it's above 109)
PCH: value 127	PCH: value 104	No change in current program; 400s bank is selected pending next PCH

0-127 Program Change Type

You may be controlling your K2600 from an "old" MIDI device—one that was built before the MIDI Controller 0 program change format was developed. If your MIDI controller is one of these (if its manual doesn't mention MIDI Controller 0 program changes, it's an "old style" machine), you might want to set the ProgChgType parameter to a value of **0–127**. This will enable you to select programs 0–127 from the controller. This limits your range of program selection, but it configures the K2600 to respond predictably to the controller. (You'll have to select higher-numbered programs from the K2600's front panel) Of course, you could use the Kurzweil format, but in many cases you'll have to send two program change commands to get the program you want.

Quick Access Banks—Extended (QA Ext)

Using this setting is similar to using the Extended program change format, but it goes one step further. Incoming program change commands are interpreted just as they are in the normal Extended format. But the resulting program change number, instead of selecting a program, selects a Quick Access bank entry (you must be in Quick Access mode for this to work). There are two advantages to using this format. First, it allows you to select both programs and setups using program change commands, without having to switch between Program and Setup modes. Second, you can remap incoming program change commands to select programs or setups with different IDs. This is handy if the sending unit can't send program change commands higher than 127.

First, a brief review of Quick Access bank structure. Each Quick Access bank can store ten entries, each of which can be a program or a setup. Each of the K2600's 10 memory banks can store 20 Quick Access banks (except the Zeros bank, which can store 75). Therefore when you're in Quick Access mode, you have access to 200 (or 750 in the Zeros bank) programs or setups without leaving the currently selected memory bank. The QA Ext program change format lets you select any one of those programs or setups via MIDI. If you select another memory bank, you have a different set of 200 programs and setups at your disposal.

When you're using this format, the K2600 will respond to MC 0 or 32 messages for selecting QA banks, and to PCHs for selecting entries within the current bank. PCHs select entries according to their "chronological" listing within the QA bank (not according to their IDs).

Command Type	Range of Values	Result
MIDI controller 0 or 32 (MC 0 or MC 32)	0 to 7	Selects QA bank 0n, 1n, 2n, 3n, 4n, 5n, 6n, 7n in current memory bank
	8–127	Ignored
Standard (PCH)	0–99	Selects last digit (n above) of QA bank, and entry within that bank
	100–127	Ignored

Depending on the QA bank entry you want to select, you'll send the K2600 either a PCH (value 0 to 99), or a MIDI Controller 0 or 32 message (value 0 to 7) followed by a PCH. Sending a single command will let you select from a range of 10 QA banks and select an entry within that bank (see the table below). To select a different range of QA banks, send an MC 0 or 32 message followed by a PCH.

The MC 0 or 32 messages selects the range of QA banks (10s through 70s), while the PCH selects the bank within that range, as well as the entry within that bank. Neither the MC 0 or 32 nor the PCH selects a different *memory* bank (Zeros through 900s). In fact, you can't change the memory bank via MIDI when using this format. All program and setup selections are made within the currently selected memory bank. You'll know which memory bank is selected by looking at the ID of the currently selected Quick Access bank in the top line of the Quick Access-mode page. Several examples follow.

If the Zeros Memory Bank is Currently Selected

Bank Range Command	Bank / Bank Entry Command	Resulting Selection
MC 0 or 32: value 0	PCH: value 6	No change (K2600 interprets this as QA bank 0, entry 6. There is no QA bank 0. Lowest valid PCH value in this case is 10, which would select QA bank 1, entry 0)
None	PCH: value 9	Entry 9 in current QA bank
MC 0 or 32: value 0	PCH: value 32	QA bank 3, entry 2
MC 0 or 32: value 1	PCH: value 4	QA bank 10, entry 4
MC 0 or 32: value 1	PCH: value 28	QA bank 12, entry 8
MC 0 or 32: value 2	PCH: value 44	QA bank 24, entry 4

Remember that in the Zeros memory bank, the Quick Access bank IDs go through 75. So if the Zeros memory bank is the current memory bank, you can send MC 0 or 32 values as high as 7 for the bank range command. And you can send PCH values as high as 99 for the bank/bank entry command. (When you're in the other memory banks, you can send MC 0 or 32 values of 0 or 1, and PCH values of 0 to 99.)

If the 200s Memory Bank is Currently Selected

Bank Range Command	Bank / Bank Entry Command	Resulting Selection
None	PCH: value 44	QA bank 204 or 214; entry 4
MC 0 or 32: value 0	PCH: value 6	QA bank 200, entry 6
MC 0 or 32: value 0	PCH: value 32	QA bank 203, entry 2
MC 0 or 32: value 0	PCH: value 99	QA bank 209, entry 9
MC 0 or 32: value 1	PCH: value 4	QA bank 210, entry 4
MC 0 or 32: value 1	PCH: value 28	QA bank 212, entry 8
MC 0 or 32: value 2	PCH: value 44	No change; MC 0 or 32 value 2 is invalid in 200s bank.

Quick Access Banks—Kurzweil (QA Kurz)

This works almost exactly like the QA Ext format. The only exception is that within the QA Kurz format, the K2600 expects the bank range command to be a PCH, and not MC 0 or 32. MIDI Controller 0 or 32 messages are not recognized. The K2600 expects to receive PCHs of value 0–99 to select a bank and entry, or a pair of PCHs, the first having a value of 100–107 to select a different 10-bank range.

QA 0–127

Finally, there's the QA Bank format for use with older MIDI devices (program change commands 0–127 only). It works similarly to the other QA formats, but the allowable range of values is limited to 0–107.

QA Banks and General MIDI Files

The K2600 isn't a General MIDI (GM) instrument, but if you load the GM files from the accessory disks, the K2600 automatically configures itself to make it easy for you to call up the

GM programs. When you load the files (actually one file split over two floppies), the K2600 loads the programs into the memory bank you select, and also loads them into Quick-Access banks, with the programs ordered so that they respond correctly to Program Change commands. Loading the file also changes the value of the ProgChgType parameter from its default of **Extended** to **QA 0–127**, so that Program Change commands select programs in the QA banks.

This allows you to load the GM programs into an empty bank, so you don't have to overwrite any existing objects that have the same IDs as the objects in the GM file. Keep in mind, however, that after loading the GM file, non-GM programs will be available only from the K2600. When sending program changes from your sequencer, you'll call up the 128 General MIDI programs, and nothing else. If you want to have sequencer access to both the GM programs and the non-GM programs, you can change the ProgChgType parameter (found on the MIDI-mode RECEIVE page) back to **Extended**. Once you do this, however, any program changes in your GM sequences will no longer call up the right GM programs, and you will have to insert new Program and Bank Change commands into the sequence to call up the appropriate programs.

QA Formats and MIDI Transmission

If you're in Quick Access mode and you're using one of the QA formats for the program change type, selecting QA banks or bank entries from the K2600 (with the alphanumeric buttonpad, the cursor buttons, the Alpha Wheel, the **Plus/Minus** buttons, or the **Chan/Bank** buttons) also sends corresponding program change commands to the K2600's MIDI Out port.

The nature of these commands depends on the value of the ProgChgType parameter. The K2600 sends either an MC 0 or 32 message followed by a PCH (when ProgChgType is **QA Ext**), or a pair of PCHs (when ProgChgType is **QA Kurz**) or a single Program Change command (when ProgChgType is **QA 0–127**). The following tables give specific examples.

Current QA Bank	Entry From Alphanumeric Pad	Commands Sent	
		MC 0 or 32	PCH
1	0	0	10
1	9	0	19
2	0	0	20
2	9	0	29
9	9	0	99
10	0	1	0
19	9	1	99
20	0	2	0
29	9	2	99
75	9	7	59
100	0	0	0
105	9	0	59
110	9	1	99
117	7	1	77
119	9	1	99

Table 10-2 QA Extended Program Change Examples

Current QA Bank	Entry From Alphanumeric Pad	Commands Sent	
1	0	100	10
1	9	100	19
2	0	100	20
2	9	100	29
9	9	100	99
10	0	101	0
19	9	101	99
20	0	102	0
29	9	102	99
75	9	107	59
100	0	100	0
105	9	100	59
110	9	101	99
117	7	101	77
119	9	101	99

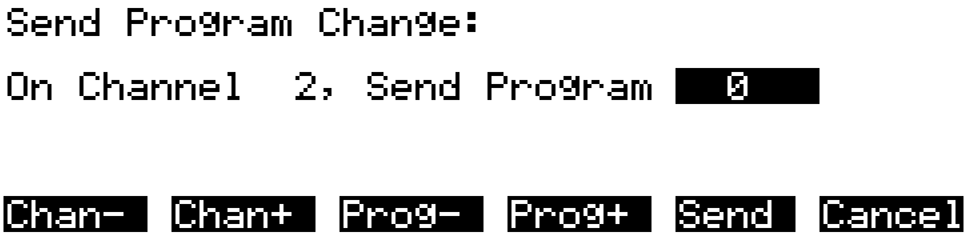
Table 10-3 QA Kurz Program Change Examples

The Soft Buttons in MIDI Mode

The first three soft buttons select the three MIDI-mode pages. The **PrgChg** soft button lets you send a program change command on any MIDI channel. The **RsetCh** soft button lets you return all channel parameters to their default values. The **Panic** soft button sends an All Notes Off and an All Controllers Off messages to the K2600 and on all 16 MIDI channels.

Program Change (PrgChg)

When you press this soft button, a dialog appears:



```
Send Program Change:
On Channel  2, Send Program    0
```

This dialog lets you send program changes out the MIDI Out port, but does not change internal programs.

The **Chan/Bank** buttons, the **Up/Down** cursor buttons, and the **Chan–** and **Chan+** soft buttons can all be used to change the channel on which the program change command will be sent. The **Left/Right** cursor buttons, the **Plus/Minus** buttons, the Alpha Wheel and the **Prog–** and **Prog+** soft buttons can all be used to change the program change number that will be sent. When you've set the channel and the program change number, press the **Send** soft button to send the program change command. Or press the **Cancel** soft button if you don't want to send it.

You can change the channel and the program number as many times as you want before you press **Send**. You also can use the alphanumeric pad to select a program number directly.

Reset Channels (RsetCh)

When you press this soft button, the K2600 will ask you if you want to reset all channels, and a pair of **Yes/No** soft buttons will appear. If you press the **Yes** soft button, all settings on the CHANNELS page will return to their default values. For example, you may have set several MIDI channels to route their audio to Output Group B for a special project. When the project's over, you can reset the Channels to restore the audio routing to each individual program (a value of **Prog**), rather than selecting each channel's page and setting the Pair parameter back to a value of **Prog**. Press the **No** soft button if you decide not to reset the channels.

Panic

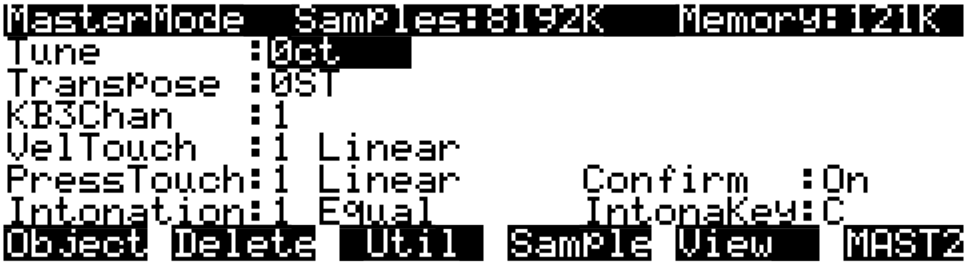
This soft button sends an All Notes Off and All Controllers Off message both to the K2600 and over all MIDI channels.

Chapter 11
Master Mode

Press the **Master** mode button to enter Master mode, which contains parameters affecting the K2600’s overall performance.

The Master Mode Page

On the Master-mode page you’ll find parameters for setting the overall tuning and transposition of the K2600, the MIDI channel to be used for KB3 programs (explained below), and for several keyboard and programming adjustments. You can also enter the sampler from the Master-mode page.



```
MasterMode  Samples:8192K    Memory:121K
Tune      :0ct
Transpose :0ST
KB3Chan   :1
VelTouch  :1 Linear
PressTouch:1 Linear      Confirm  :On
Intonation:1 Equal       IntonaKey:C
Object  Delete   Util   Sample  View    MAST2
```

Parameter	Range of Values	Default
Tune	± 100 cents	0
Transpose	± 60 semitones	0
KB3 Channel	1 to 16	1
Velocity Touch	Velocity Map list	1 Linear
Pressure Touch	Pressure Map list	1 Linear
Intonation	Intonation Table list	1 Equal
Confirm	On, Off	On
IntonaKey	C–B	C

Tune

Adjusting the value of this parameter tunes every program in the K2600 by the amount you specify. Tuning can be adjusted up or down 100 cents (one semitone) in one-cent increments. This parameter is useful for getting in tune with recordings and acoustic instruments. Adjusting the tuning in Master mode does not change the settings on the PITCH page of individual programs, but will be added to any adjustments you make there. Master-mode tuning adjustments affect only the K2600’s notes, and not notes sent via MIDI.

Transpose

Like the Tune parameter above, Transpose affects every K2600 program, but not those notes sent to the MIDI Out port. You can adjust the MIDI transposition on the TRANSMIT page in MIDI mode.

KB3 Channel

The architecture of KB3 programs is completely different from the architecture of regular VAST programs. KB3 don't have layers containing keymaps and samples that get played with each keystrike. Instead, they have an array of oscillators that are constantly running, which requires a different kind of processing (and more of it). Consequently, a KB3 program won't run on a "normal" MIDI channel. It has to have a special channel that processes note information differently. And that's what the KB3 Channel parameter does.

There can be only one KB3 channel, although you can set it to be any of the 16 MIDI channels. It's like saying "OK, I want Channel 1 to be the channel that plays KB3 programs. All the other channels are for regular programs and other stuff."

Any program can play on the KB3 channel, but *KB3* programs can play *only* on the KB3 channel. If you're on a keyboard channel that's not the KB3 channel, and you select a KB3 program, the program's name appears in parentheses in the display, and you won't be able to trigger any notes on that channel. In this case the box at the left of the display reminds you which channel is the KB3 channel.

Velocity and Pressure Touch (VelTouch and PressTouch)

These two parameters are used by the keyboard model only, and have no function in the K2600R. If you are using a K2600R, leave these parameters set to their default values.

If you change the setting of the VelTouch parameter, remember that it also has an effect on the transmit velocity map (which is on the MIDI-mode TRANSMIT page).

Intonation

Most modern western music uses what is known as equal temperament. This means that the interval between each semitone of the 12-tone octave is precisely the same as every other interval. Many different intonation intervals have evolved over the centuries, however, and the K2600 supplies you with 17 different intonation "tables" to choose from. (There are also a few extra "tables" listed, which we'll describe in a moment.) By changing the value for this parameter, you select from among the intonation tables stored in the K2600's memory. Each of these tables defines different intervals between each of the semitones in a single octave.

Scroll through the list of Intonation tables, and listen for the differences between semitones. Some of the intervals between semitones may be quite different from equal intonation, but you'll notice that all notes are precisely tuned with notes that are an octave apart. This is because the intonation tables set the intervals within a single octave, and apply those intervals to each octave. If this doesn't make sense, the explanation of the Intonation Table Editor, in Chapter 18, will help clarify things. If you're hoping to create fully microtonal tunings by editing intonation tables—sorry, that's not possible. But you *can* create microtonal tunings using the Keymap Editor; see Chapter 14.

Determining the Version Number of Your ROM Objects (Intonation Tables 18–22)

As you're scrolling through the list of intonation tables, you may notice a listing for an eighteenth intonation table with a name such as **18 Obj v*n.nn***. This isn't really another intonation table. Rather, this is where the K2600 stores the version number of some of your ROM objects. If you ever need to find out what version of ROM objects you've got loaded, this is where you look. Simply go to the Master page, then scroll the Intonation parameter until **18** is displayed. If you have more than one block of ROM objects installed, you'll see additional "tables," up to and including 22. And don't forget to return to your correct intonation table when you've checked the version numbers of your ROM objects.

List and Description of Intonation Tables

1	Equal	No detuning of any intervals. The standard for modern western music.
2	Classic Just	Tunings are defined based on the ratios of the frequencies between intervals. The original tuning of Classical European music.
3	Just ♭7th	Similar to classic Just, but with the Dominant 7th flatted an additional 15 cents.
4	Harmonic	The perfect 4th, Tritone, and Dominant 7th are heavily flatted.
5	Just Harmonic	
6	Werkmeister	Named for its inventor, Andreas Werkmeister. It's fairly close to equal temperament, and was developed to enable transposition with less dissonance.
7	1/5th Comma	
8	1/4th Comma	
9	Indian Raga	Based on the tunings for traditional Indian music.
10	Arabic	Oriented toward the tunings of Mid-Eastern music.
11	BaliJava1	Based on the pentatonic scale of Balinese and Javanese music.
12	BaliJava2	A variation on 1Bali/Java, slightly more subtle overall.
13	BaliJava3	A more extreme variation.
14	Tibetan	Based on the Chinese pentatonic scale.
15	CarlosAlpha	Developed by Wendy Carlos, an innovator in microtonal tunings, this intonation table flats each interval increasingly, resulting in an octave with quarter-tone intervals.
16	Pyth/aug4	This is a Pythagorean tuning, based on the Greek pentatonic scale. The tritone is 12 cents sharp.
17	Pyth/dim5	This is a Pythagorean tuning, based on the Greek pentatonic scale. The tritone is 12 cents flat.
18–22	Obj v*n.n*	Not an intonation table; indicates version number of K2500 ROM objects.

In general, you should select a nonstandard intonation table when you're playing simple melodies (as opposed to chords) in a particular musical style. When you use intonation tables based on pentatonic scales, you'll normally play pentatonic scales to most accurately reproduce those styles. An excellent reference source for further study of alternative tunings is *Tuning In: Microtonality in Electronic Music*, by Scott R. Wilkinson.

Confirm

Confirmations are special displays that the K2600 shows you when you are about to alter memory permanently. The confirmations ask if you really want to do what you're about to do, and give you another chance to cancel the operation you're about to execute. With the Confirm parameter set to **Off**, these prompts do not appear. You'll still be alerted before doing something that might cause you to lose your work, but your margin of error is slimmer with confirmations turned off.

Intonation Key (IntonaKey)

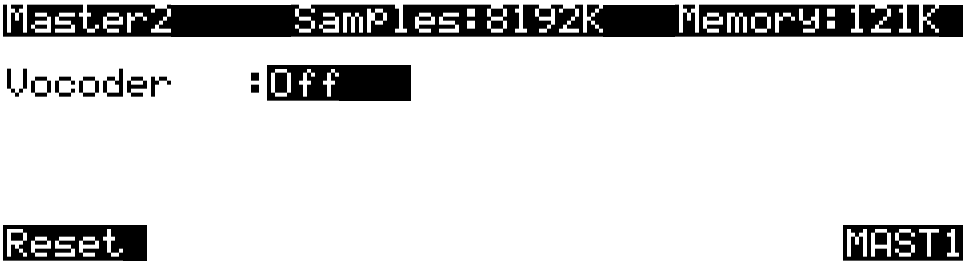
This sets the tonic, or base note from which the currently selected intonation table calculates its intervals. If you select **G** as the intonation key, for example, and the intonation table you select tunes the minor 2nd down by 50 cents, then G$^{\#}$ will be a quartertone flat relative to equal intonation. If you change the intonation key to **D**, then D$^{\#}$ will be a quartertone flat. If you use nonstandard intonations, you'll want to change the intonation key as you change the key you're playing in. If the Intonation parameter is set to **Equal**, changing IntonaKey has no effect.

You can also set the intonation key from an external MIDI device. Note On events at C -1 through B -1 (MIDI note numbers 0 through 11) will set the intonation key at **C** through **B**, respectively.

To trigger notes in the range required to set the Intonation key, you can transpose the K2600 temporarily from its front panel, or from your MIDI controller if it has the ability. Alternatively, you could create a setup with just the lowest octave transposed down two octaves, then select it when you want to change the Intonation key. If you're driving your K2600 from a sequencer, you could simply insert the appropriate note events anywhere in the sequence to change the intonation key.

The MAST2 Page

Press the MAST2 soft button to reach the MAST2 page:



```
Master2       Samples:8192K    Memory:121K
Vocoder    :Off



Reset                                MAST1
```

The MAST2 page enables you to perform a hard reset of your instrument (this deletes everything in RAM, so be careful!), and to turn the vocoder feature on and off.

The Vocoder

Vocoding is a special feature that allows you to use an input signal to control another audio (slave) signal. Typically you would use a synthesizer for the input signal, although in fact you can use any sound source. You must have the sampling option to be able to use the vocoder.

Cables and Connections

Using the K2600 for Both Input and Slave Signals

You'll need an insert cable (Y cord) with a 1/4-inch stereo (Tip/Ring/Sleeve) plug (male) on one end and 2 mono jacks (female) on the other end. The right side mono jack should be 1/4-inch. The left side can be either 1/4-inch or XLR. (You will be plugging a Mic into the left side, so if the insert cable has a 1/4-inch jack, you'll need an adapter from XLR to 1/4-inch.)

1. Plug the stereo side of the insert cable into the 1/4-inch stereo Sample Input.
2. Connect a microphone to the left mono jack of the insert cable. It *must* be the *left* input.
3. Connect the B Right output of the K2600 to the right mono jack of the Insert cable.

Using the K2600 for Input Signal and External Source for Slave Signal

There are two setup methods for this configuration. Here's the first:

You'll need an insert cable (Y cord) with a 1/4-inch stereo (Tip/Ring/Sleeve) plug (male) on one end and 2 mono jacks (female) on the other end. The right side mono jack should be 1/4-inch. The left side can be either 1/4-inch or XLR. (You will be plugging a Mic into the left side, so if the insert cable has a 1/4-inch jack, you'll need an adapter from XLR to 1/4-inch.)

1. Plug the stereo side of the insert cable into the Stereo Analog Input of the sampler.
2. Connect a microphone to the left mono jack of the insert cable. It *must* be the *left* input.
3. Connect the output of your external sound source to the right mono jack of the insert cable.

The second method:

1. Connect a microphone into the left low impedance input (XLR) of the sampler.
2. Connect your external sound source to the right low impedance input (XLR) of the sampler.

Final Audio Output

You must have audio cables connected from the A outputs on the K2600 to your mixer or amp. Don't use the Mix outputs.

MIDI

If you have a keyboard model, and your external slave is a rack (or it is a keyboard but you want to use the K2600's keyboard to control the slave), connect a MIDI cable from the MIDI Out port of the K2600 to the MIDI In port of the slave.

Setting Up the K2600

1. Go to Sample mode (press the **Sample** soft button in Program, Setup, or Quick Access mode).
2. Set the Input parameter to a value of **Analog**.
3. Set the value of the Source (Src) parameter to **External** (**Ext**).
4. Set the value of the Mode parameter to **LiveIn**.

5. Verify that mic signal is on the left side only. Adjust the Gain parameter as needed, to get a good signal level.

6. Verify that your sound source (either the K2600 or external source) is on the right side only.

7. Go the Effects-mode page and make sure that the FX Mode parameter is set to **Auto** and the FX Chan parameter is set to **Current**.

Enabling Vocoder Mode

1. Load the file VOCODER.K26 into any bank. It's on the disk called *K2600 Farm & Extras.* See Chapter 13 if you need help loading a file.

2. Go to Master mode.

3. Press the **MAST2** soft button.

4. Set the value of the Vocoder parameter to **On**.

5. Exit from Master mode (press any of the other Mode buttons).

Note that enabling the vocoder activates special software, which replaces the software used for the SHAPE2 and AMP MOD OSC functions in the F3 block of an algorithm. Therefore any programs that use SHAPE2 and AMP MOD OSC in the F3 block will sound different while the vocoder is active. Turning the Vocoder parameter Off will restore those DSP functions and disable vocoding.

Using the Vocoder

Go to Setup Mode and select one of the setups in the memory bank where you just loaded the vocoder file. If you are using an external sound source for your slave, choose the setup **Vocoder-ExtSlave**. If you are using the K2600 as the input source for the slave, then you can choose either **Vocoder-22 Band** or **Vocoder-20 Band**. The 22-band vocoder will allow you to play up to 4 voices of polyphony on the slave program; the 20-band vocoder will allow you to play up to 8 voices of polyphony on the slave program.

Play a note or chord on your keyboard and speak into the microphone. You should be able to hear what you are speaking, but the sound will be a string sound (assuming you are using the K2600 as the slave source), pitched to the note or chord you are playing.

If you have a keyboard model, try moving Sliders A, B, and C, and listen for changes in the sound. If you have a rack-mount model, you can send MIDI Controller numbers 6, 12, & 13 from your controller. Since the setups contain entry values for these sliders, you may have to move the slider across its full range before it begins to take effect.

Effects Issues and Output Issues

The studio assigned to the vocoder setups is configured in the following manner: If you are using the K2600 for the slave signal, the slave program (in zone 3) has its output assigned to KDFX-B, which is being routed to the FXBus2, with no effect. On the OUTPUT page in the Setup Editor, Output B is set to **FXBus2**, thereby sending the signal from the slave program to the B outputs and from B Right into the right side of the sample input.

The slave program has its output panned hard right within the program, so if you decide to try using a different slave program, you will probably want to edit the program itself to pan its output hard right, so you get 100% of the signal. You don't need to worry about setting the output pair within the program, because the Out parameter on the CH/PRG page of the Setup Editor is set to KDFX-B in zone 3, thereby overriding any settings from within the program.

The vocoder programs themselves are assigned to KDFX-A, which is being routed to FXBus1. On the OUTPUT page in the Setup Editor, Output A is set to **Mix**. So the final output of the vocoder programs is run through the effect and then comes out the A Outs and the Mix Outs. Don't use the Mix audio outputs, however, or you'll hear the slave program along with the vocoder.

If you choose to change the effects, you may find it easier to edit the vocoder studio, and try changing the effects assigned to FXBus1, FXBus2, and AuxFX. But if you want to change to a different studio, you will need to make sure the following parameters are set correctly: on the FXBUS page, for FXBus2, set the Level parameters for both Aux and Mix to **Off**, and on the OUTPUT page, set Output B to **FXBus2**.

How Vocoding Works

A vocoder is a device that analyzes the time-varying audio spectrum of one signal (the master) and imposes that spectrum as a filter on a second signal (the slave.) The method we use is an emulation of the traditional analog technique involving banks of bandpass filters and envelope followers.

The master signal is what you send from the microphone, and the slave signal is what you send from an external synthesizer or other sound source, or a program from the K2600.

The master signal is sent to a number of bandpass filters in parallel. The center frequencies are spaced to cover the most useful frequencies. The lowest frequency filter is a low pass rather than a bandpass, which groups all low-frequency components together. Likewise, the highest filter is a high pass. The outputs of all these bandpass filters go into individual envelope followers, which detect the level of signal present in each band. The output of the envelope follower is then used as a control for the slave signal.

The slave signal is also sent to the same number of bandpass filters. These generally have the same center frequencies as the master bandpasses. The output signals from the slave bandpasses are multiplied, one by one, by the outputs of the envelope followers (from the master signal). The resulting products are all added together for the final output.

Since each band requires two layers (one for master and one for slave), the largest number of bands you can have for vocoding is 24. (24*2=48, which is your maximum polyphony.) The programs in the Setup called **Vocoder-ExtSlave** use 24 bands. If you want to use the K2600 to generate your slave signal, then you have to use either the 22- or 20-band vocoder setups, which have fewer bands, and therefore leave 4 or 8 voices of polyphony available for the slave signal program.

Since 48 (or 44 or 40) layers are used, and a drum program has a maximum of 32 layers, we use two 24 (or 22 or 20) layer programs, on different MIDI channels, that are combined in a setup.

Each of the setups has 3 zones. In the 22- and 20-band vocoder setups, the first two zones are used for the vocoding programs and the third zone plays the internal program that is used for the slave signal. In **Vocoder-ExtSlave**, the third zone is set to transmit via MIDI only, on Channel 1. (This allows you to play your external sound source, if it is a rack, but won't play a K2600 internal program.)

Layers are grouped in pairs, with the master signal going to the first layer, and the slave to the second. All odd numbered layers are master and all even numbered layers are slave. If you look at the algorithms in the vocoding programs, you will see that the first two DSP blocks (after PITCH) of each layer are a bandpass filter (or low pass or hi pass filters for the first and last bands). The first layer then has a DSP called MASTER, while the second layer has a DSP called

SLAVE. These stages are then followed by an AMP stage. These DSP blocks perform the function of an envelope follower and gain multiplication.

The signal flows from the odd numbered (master) layer to its associated even numbered (Slave) layer (for example, from layer 1 to 2), which is something that does not happen in other algorithms. The low pass frequencies controlled by the third time slot for each layer set the response speed of the envelope follower. They are normally set to the same frequency. The master layer controls the frequency of one pole of low pass filtering, and the slave layer controls two more poles.

The AMP page on the master layer does nothing. There is no output from this layer, so any settings on the OUTPUT page don't matter. The slave layer's AMP page does do an actual amplitude control. The output pages for slave layers are active, and can be used to choose the output group and set the step panning.

All of the master layers use the LiveIn Left keymap and all of the slave layers use the LiveIn Right keymap. That is why you must plug the microphone into the left side of the sample input and the slave source into the right side.

As is always the case with Live mode, a note message is required in order for an incoming signal to be processed through VAST. Therefore, the two layers in the setup assigned to the vocoding programs have Pswitch2 set to generate a C4 with a velocity of 127, as soon as the setup is selected. That note remains on until you select a different setup. The setups are edited so that none of the notes on an 88 note keyboard are assigned to either of the two vocoding programs

Real-time Control of the Vocoding Programs

The most important control parameter is the envelope follower speed, set by the third time slot low pass parameters. These are set to C 6 on all the layers for the initial level. Slider A (MIDI 6) lowers the cutoff up to 8 octaves (9600 cents). Therefore, the higher you raise the slider, the slower the envelope follower speed. C 6, as a filter cutoff, has a time constant on the order of one millisecond. This is generally too fast. For best results, this should be lowered about 4 octaves to C 2 (half the range of the Data Slider), to a time constant of 16 milliseconds. Too slow and the vocoder will not respond to quick transients, like consonants, and too fast will result in a jittery sort of sound, as the envelopes follow every little fluctuation. At the fastest possible setting, the envelopes follow the master audio signal itself, and an extremely harsh intermodulation is heard between master and slave. The vocoder setups have an entry value of 64 for this slider, so when the setup is selected it is the equivalent of having the slider halfway up.

Slider B (MIDI 12) is used to control the width of the band pass filters (for all bands except the lowest and highest). The vocoder setups have an entry value of 10 for this slider, the equivalent of having the slider at the first dot above the bottom.

Slider C (MIDI 13) transposes the center frequencies of all the slave bandpasses upward together. It gives you the same result as pitch shifting the master signal up. Vocal formants will be munchkinized as you bring the slider up. The vocoder setups have an entry value of 0 for this slider, the equivalent of having the slider at the bottom.

Additional Notes and Programming Suggestions

The classic application of a vocoder is to make instrumental sounds talk/sing. The slave signal has to have a lot of high frequency content, or the consonants will not be heard clearly. However, there is no rule set in stone that you must speak words into the microphone. Using the vocoder just as a timbral control can be just as interesting. You can get very expressive results by using your voice to control a lead line, doing the articulation and filter control by talking, singing, or just making various vocal sounds. You can get some of the same types of results you would by using a breath controller. It’s a little like having a 24-band graphic equalizer, but instead of controlling it with your hands, you use your voice.

Furthermore, you don't even have to use a microphone as the master. You can send a signal from anything else that has varied timbral content and get interesting results. For example, the master signal could be a drum loop or some other recorded sound that changes timbres regularly.

The analog sample inputs on the K2500 are line level, not mic level. This means you have to boost the gain on the sample page to get a good signal. But this also increases the general noise level of the input signal. If you have a mic preamp, or plug the mic into a mixing board before sending the signal to the K2600, you can lower the Gain parameter and start with a much cleaner signal. This is highly recommended.

In addition, you will find you get better results if you run the preamped mic signal into a compressor before sending it to the K2600. This can also be done for the slave signal. Using compressors will give you a much more even dynamic result, making it easier to play and control your sound. This is because the dynamic range of the master and slave signals is added together. For example, let's say both the master and slave signals have a dynamic range of 20 dB. The resulting signal will have a dynamic range of 40 dB, giving you a very wide range between the softest and loudest signals you can produce.

One way to improve intelligibility is to mix in a little of the master signal into the final audio output. This can be done in a couple of ways. If you run the mic into a mixer, you can split the signal, sending it both to the K2600 as well as to your final mix.

A second way is to include it in the vocoder program. You can do this by editing one of the programs in the 22- or 20-band vocoder setups. You would want to add a layer to the program (it doesn't matter which one of the two programs you edit). Set the Keymap for the layer to LiveIn L and choose Algorithm 1 with the DSP function set to **NONE**. You could then control the amount of the signal by editing the Adjust parameter on the F4 AMP page (or even assign a control source to vary the amount).

You could then try various algorithms and DSP functions to further modify the signal. Running the signal through a high pass DSP to emphasize vocal articulations is one obvious example. Just make sure that you don't use the SHAPE 2 or AMP MOD OSC DSP functions. In that case, the master signal won't be output.

If you are using the K2600 for the slave signal, try editing the slave vocoder program. A simple thing to try is to choose a different keymap. The AMPENV in this program has been set to User, with a lengthy decay, so you can even choose decaying sounds such as guitar, and get interesting results. And of course, you can choose other programs as the slave.

And of course, you should try making some of your own programs to use as a source. Just edit the setup and change the program in zone 3 to your new program. For example:

- Use an LFO to modulate the center frequencies of the slave bandpasses, or the master bandpasses.
- Try panning alternate bands of the slave layers to L and R to create a "fake stereo" program.
- Try different center frequencies from the ones used in the preset programs.
- Currently the center frequencies of the slave layers match the master layers. Try scrambling the slave frequencies relative to the master frequencies.
- If you are using the K2600 for the slave signal and need more polyphony, you can delete some of the layers in the vocoding programs. Make sure to delete matching sets of master and slave layers. You will probably want to readjust the frequencies and widths of the remaining layers accordingly.

More applications

Instead of using a microphone or other external source for your master, you could use the K2600 to generate *both* the master and slave signals. There are two ways you could set this up. You can either edit the setup to add another program on a 4th zone, or you could edit the slave source program to add more layers. Then split the keyboard so that one side plays the master zone/layers and the other side plays the slave zone/layers. On the OUTPUT page, make sure all the master layers are assigned to B and panned hard left and the slave layers assigned to B and panned hard right. You will then have to alter the wiring setup described at the beginning of this document so that the B Left jack is going to the left side of the stereo sample input.

If you edit width of the master layers so that they are extremely narrow, and set the frequencies to a specific scale pattern, then if you sing into the microphone, you will only hear sound as you sing the specific pitches in that scale.

If you edit the width of the slave layers so that they are extremely narrow, then you will get a very pure tonal sound, hearing only very specific pitches depending on the harmonic content of the master.

Another possibility for using very narrow width master layers: Edit the slave layers so that instead of using a series of bandpass filters, each slave layer uses different DSP functions in the F1 and F2 slots (remember that the F3 slot still needs to be set to LPCLIP in order for the vocoding function to work—you can change algorithms as long as the algorithm allows LPCLIP to be selected for the F3 slot). Now, if you sing various pitches, the slave signal will be played through the various corresponding VAST algorithms.

It is actually possible to use samples in RAM (or ROM) instead of the Live Mode In for either the master or slave signals (or even both of them). Just change the Keymap parameter on the KEYMAP Page. (Remember that you need to edit the Keymap parameter on all master and/or slave layers.) In this case, the keymap would be playing a single held sample, so you will want to use a looped sample. Loops with changing harmonic content will work best. The note used in the setups is C 4, so you would want the sample root at C 4 to hear it back without transposition. You will need to edit the layers, save the programs, and reselect the setup before you will hear the change. If both the master and slave layers call up samples in the unit, then as soon as you select the setup, you will hear sound without even touching the keyboard! You might want to assign a slider to the F4 AMP page on the slave layers to control the amount of output. If the master and slave layers are loops of slightly different lengths, then you will hear a continually changing sound that could appear to go in indefinitely without changing.

Continuing with the previous suggestion, you could set the slave layers to different keymaps, each layer assigned to a different sample loop. Edit the DSP functions on the slave layers so that F1 and F2 are set to **NONE**, or some other DSP function. Set the master layers to very narrow widths. Now, as your master signal changes frequencies you will hear different sample loops fading in and out.

The Soft Buttons in Master Mode

Object

This soft button brings up the object utilities. They're described beginning on page 11-13.

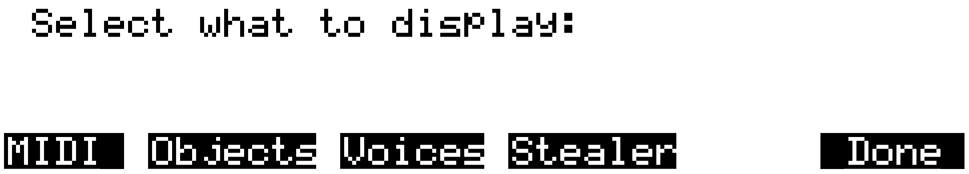
Delete

This soft button brings up the delete bank dialog, enabling you to erase sets of objects, either entire banks or all objects, from RAM. If the Confirm parameter on the Master-mode page is set to a value of **On**, you'll be given an extra chance to cancel before the set of objects is actually deleted. Once the deletion is complete, the objects are irretrievable, so you may want to save objects to disk before deleting them from RAM.

To delete individual objects, use the functions that are available when you press the **Object** soft button.

Util

With this button you call up the Utility page, which gives you access to four analytic and diagnostic tools. Double pressing the two center soft buttons from any editor is another way to get to the Utility page. The Utility page looks like this:



Select what to display:

The **MIDI** soft button launches MIDIScope™, a useful subprogram that lets you monitor the MIDI messages from the K2600's keyboard and those received via MIDI. This is a good way to make sure you're receiving MIDI from MIDI masters. It's also good for making sure your controls are assigned where you want them, checking your attack velocities, etc.

The **Objects** soft button displays the entire list of objects stored in RAM. This is an easy way to check the object ID of any object you've created. You cannot manipulate objects, though, as you can with the Objects Utility (see page 11-13).

When you press the **Voices** soft button, the display shows the K2600's active voice channels as you play. Blocks of capital Xs in six columns of eight represent the 48 notes that the K2600 can play simultaneously. The Xs change to lower case xs, then to commas and periods, then finally drop out as each voice releases or decays to silence.

This feature gives you an indication of the envelope level of each voice, though not necessarily the volume level. Nonetheless, this can give you a valuable indication of how your voices are being used. For example, if all or most of the voices are being represented by capital Xs, then there's a good chance that when voice stealing takes place an audible voice will be reallocated.

The Voices utility works a bit differently for KB3 programs. The K2600 uses one voice of polyphony for every two tone wheels in a KB3 program. In the Voices utility, the voices used by the tone wheels appear as Xs, meaning that the voices are used for the KB3 program. They don't get reallocated at any time, since they're always on, even if you're not playing any notes. Any voices not dedicated to a KB3 program behave normally. So if you have a setup that contains a KB3 program in one zone, and VAST programs in one or more other zones, you can monitor the voice allocation of the non-KB3 voices in the section of the display that isn't constantly filled with Xs.

Use the **Stealer** soft button to select a display that will show how the K2600 is allocating its 48 voice channels. When you trigger a note, the note number will appear in one of the display's three columns, and will remain visible while the note is sustained. The four-digit numeral you see is an internal value that has no direct significance.

As long as fewer than 48 voice channels are being used, new note numbers will appear as you play additional notes, and the note numbers for notes that have decayed or have been released will disappear. When all 48 voices have been activated, the display will show which voice channels are shut off (“stolen”) to enable new notes to play.

Press the **Done** soft button when you are finished with the Utility page. This is the same as pressing the Exit button.

View

Press the **View** soft button to set the view mode to large type. Press it again to return to the normal view. The large-type view affects Program, Setup, and Quick Access modes. When you’re in these modes, program names, setup names, and Quick Access-bank entry names appear in large, easy-to-read type.

Sample

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

Reset

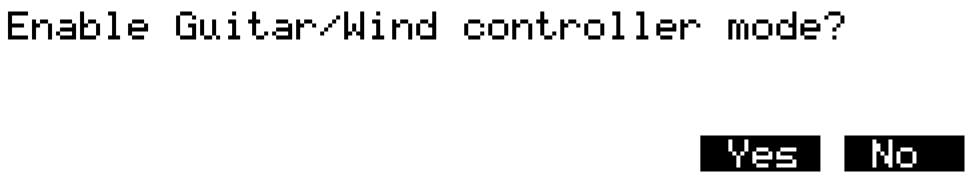
This button is on the MAST2 page. Press the **Reset** soft button if you want to return your K2600’s memory to the state it was in when you bought it.

Caution! *The K2600 will ask you if you want to delete everything (meaning all RAM objects), and a pair of* ***Yes/No*** *soft buttons will appear. Press* ***No*** *if you want to keep any objects you may not have saved. Press* ***Yes****, and everything stored in RAM will be erased. All parameters will be restored to default values. After a few seconds, the K2600 will return to the Program-mode page.*

Guitar/Wind Controller Mode

If you are using a wind controller or guitar controller with your K2600, you may not always get the sound you expect. Since these controllers will sometimes send a MIDI Note On command before sending Breath or Volume data, the attack transients that characterize each instrument may not get generated properly.

Consequently, the K2600 provides a special mode that may improve its response to your guitar or wind controller. To enter Guitar/Wind Controller mode, press both **Chan/Bank** buttons while in Master mode, then confirm with the **Yes** soft button.



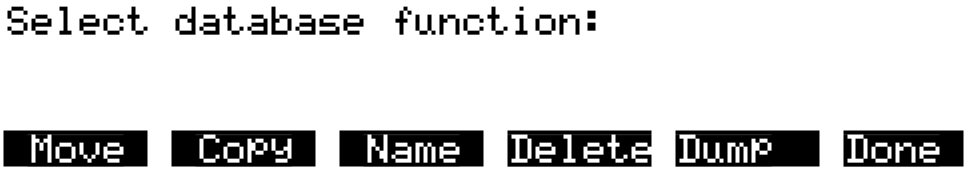
Enable Guitar/Wind controller mode?

Guitar/Wind Controller mode slightly delays MIDI Note On and Note Off commands, so that response to pitch bend and other expressive components of a note will be more accurate. If you’re hearing a glitch in the attack of notes from your guitar or wind controller, you should try setting your K2600 to this mode. Keep in mind, however, that since this mode slightly changes the order in which MIDI commands are sent, it may affect the performance of the K2600 under some circumstances.

Restarting the K2600 or performing a soft reset disables Guitar/Wind Controller mode.

Object Utilities

Object Utility functions are useful for moving or copying objects into various banks, naming or renaming objects, deleting objects, and dumping objects over MIDI. To access these functions, press the **Object** soft button while in Master mode. You will see the following dialog:



```
Select database function:


 Move    Copy    Name   Delete  Dump    Done
```

The soft buttons are used to choose the various object utility functions.

Move Move selected objects to a new bank or a specific starting ID.

Copy Copy selected objects to a new bank or a specific starting ID.

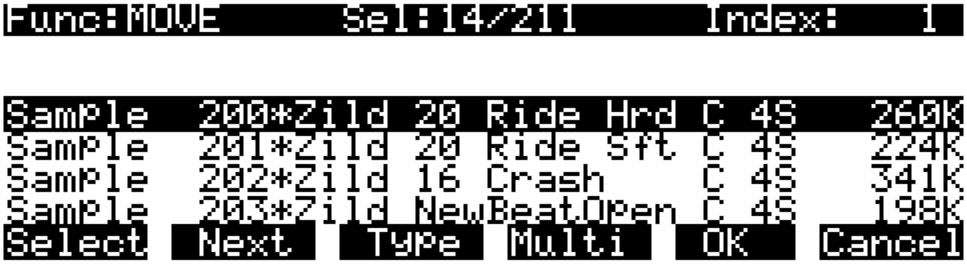
Name Name selected objects.

Delete Delete selected objects.

Dump Dump selected objects over MIDI.

Done Exit from the object utilities.

Each function's multiple object selection interface is identical to the one used in the Save Objects dialog. For a complete description of this interface, see *Saving Individual Objects* on page 13-26. Here is what the **Move** page would look like (with several objects selected):



```
Func:MOVE       Sel:14/211      Index:    1

Sample  200*Zild 20 Ride Hrd C 4S    260K
Sample  201*Zild 20 Ride Sft C 4S    224K
Sample  202*Zild 16 Crash    C 4S    341K
Sample  203*Zild NewBeatOpen C 4S    198K
Select  Next    Type    Multi   OK   Cancel
```

The name of the function is displayed on the top line:

```
Func:MOVE
```

If you press **Cancel** while in one of the object utilities, you return to the Object Utilities page (the “Select database function:” dialog pictured above). Any objects that were selected when you pressed **Cancel** will still be selected if you subsequently enter a different object utility (by pressing a different soft button such as **Name**, for example). The selections are reset when you exit the Object Utilities page (by pressing the **Done** button).

All of the features of the Save Object dialog are accessible here:

> You can use the Multiple Object Selector (described on page 13-36) to select ranges of objects according to object types, IDs, strings in the object names, or dependent relationships.
>
> You can quickly select or deselect all objects using the **Left/Right** cursor and **Up/Down** cursor double-presses.
>
> You can audition any of the program, keymap, sample, or song objects by pressing either the **Left** or **Right** cursor button, when the desired object is highlighted. Songs will play until either cursor button is pressed again. To audition a program, keymap, or sample object, play a note on your MIDI controller (after pressing the **Left** or **Right** cursor button). K2600R owners also have the shortcut of holding the **Cancel** button, then pressing the alphanumeric buttons to play notes.

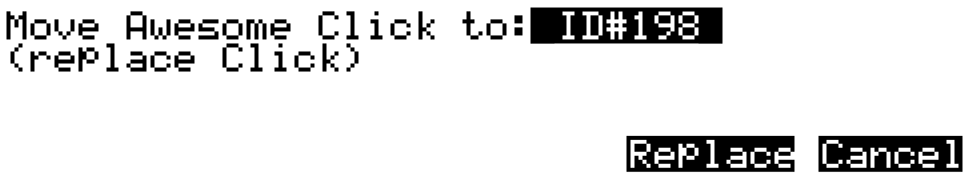
Move

Pressing **Move** from the Object Utility page takes you to the Move utility. The Move utility allows you to select any group of objects and move them to a different bank. If you select several objects of a single type, then you are allowed to set a specific starting ID for the objects, of any number (0-999). For example, you could move a group of samples from scattered IDs to a continuous range of IDs starting from ID# 354.

If you move objects that are dependent objects of other objects (such as samples that are dependents of a particular keymaps,) the parent objects are automatically relinked to their dependents that have moved. What this means is that you can move any objects to any new ID numbers, without having to worry if your programs, keymaps or songs will still play correctly.

Moving a Single Object

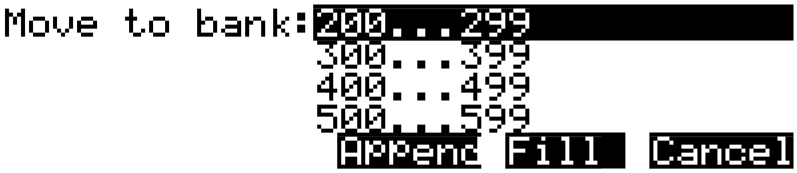
If you select a single object to move, you will see a dialog similar to the Save/Replace dialog you see when saving any object.



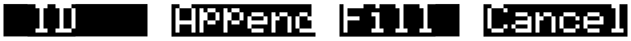
```
Move Awesome Click to: ID#198
(rePlace Click)
```

Moving Multiple Objects

Select the objects that you want to move, then press **OK**. You will see the following dialog:


Move to bank: 200...299

This is similar to the Load function, where you are asked to choose a bank and mode for loading. If you have selected more than one object, and all of the selected objects have the same type, then there will be an additional button available, **ID**:

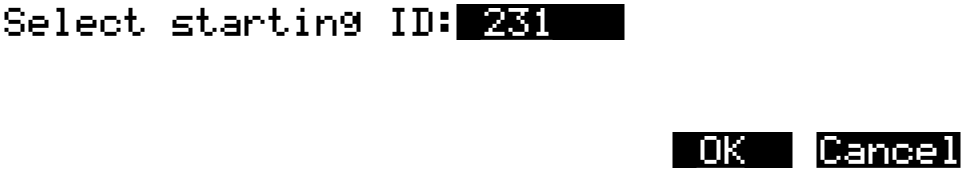


Here is a description of the above soft buttons:

Append: Try to use each object's ID offset within its current bank as the ID offset within the specified destination bank. If there is already an object at this offset in the new bank, increment the ID until a free ID slot is found. For example, if you were moving Programs 202, 209, 217, and 230 to the 400s bank, which already has a program at 409, the resulting IDs for the moved programs would be 402, 410, 417, and 430.

Fill: Use consecutive numbering for each object that is moved, starting from the beginning of the specified destination bank. Any object IDs that are already being used in the specified bank will be skipped over. For example, if you were moving Songs 300, 315, 489, and 841 to the 200s bank, which already contains Songs 200 and 203, then the moved songs' ID numbers would be 201, 202, 204, and 205.

ID: Use consecutive numbering for each object that is moved, skipping over IDs that are in use (like Fill mode), starting from the ID that you specify. You will see the following dialog when you press the **ID** soft button:


Select starting ID: 231

Cancel: Return to the Move object dialog.

The objects are moved as soon as **Append** or **Fill** is pressed, or when a starting ID is selected. After the Move function completes, you will still be in the Move object dialog, and you will still be scrolled to the previously highlighted object, even if it has moved to a new ID.

Keep in mind that when you only select one object to move, you can replace another object. However, when you select multiple objects for moving you cannot overwrite any objects.

Copy

Pressing **Copy** from the Object Utility page takes you to the Copy utility. The Copy utility allows you to select any group of objects and copy them to a different bank. Only object data is copied, and not sample data. If you copy a sample object, you will end up with a “copy sample” that points to the same region of sample RAM as the original.

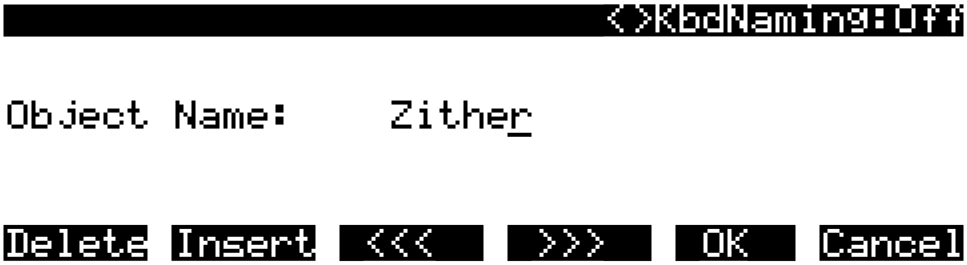
The operation of the Copy utility is identical to the Move utility just described.

If the objects to be copied in a single operation include any objects grouped together with any of their dependents, the new copies of the parent objects will reference the new copies of the dependent objects. As an example, suppose you select Song 400 and its three dependent programs, Programs 200, 210, and 303. If you copy all of these objects at once into the 700s bank, using Fill mode, you will see the copies at Song 700 and at Programs 700, 701, and 702. Song 700 will reference the copies of the programs (at 700, 701, and 702). In contrast, if you had only made a copy of Song 400 as Song 700, the song would reference the old programs (at 200, 210, and 303).

Name

The Name utility allows you to rename an object without entering an editor. You can also use this utility to rename one or more objects with the same name. This is much faster than renaming each object individually. A lot of times you might want several objects to have the same name except for a unique identifier at the end of the name. Using this utility function, you could assign a common name to multiple objects at once, and then quickly name each one a little differently.

When you press **OK** after selecting objects, you will see the following dialog prompting you for an object name, with a suggested default:



<>KbdNaming:Off

Object Name: Zither

The default name that you see comes from the highlighted object in the object list, regardless of whether the highlighted object is actually selected. This makes it easy to copy the name of one object on to another.

As in all naming dialogs on the K2600, you can do a double-press of the **Left/Right** cursor buttons to put the naming cursor on the last character of the string. This is helpful when putting unique characters at the ends of names.

Left/Right *cursor button double-press -> Move cursor to the end of the name*

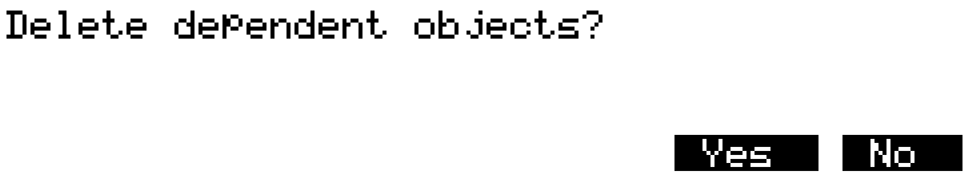
The Relink-by-Name feature (described earlier) relies on there being unique names for dependent objects of the same type, so it is a good practice to make object names unique, particularly samples.

You can use the keyboard naming feature when naming objects. See page 5-5 for details. You can also use the string-replacement feature to make multiple changes to object names. See *Renaming Multiple Objects* on page 3-8 for details.

Delete

The Delete Objects utility is very useful for reclaiming unused object and sample RAM in your K2600. This utility allows you to select any arbitrary group of objects for deleting, and audition them if necessary before getting rid of them. This can be a convenient way to delete individual or selected groups of objects. However, if you want to delete an entire bank or everything in RAM, the **Delete** soft button on the Master-mode page is quicker.

If any of the selected objects have dependents that were not selected, you will see the question:



Delete dependent objects?

If you answer **Yes** to this question, all dependent objects of the selected objects are deleted, unless they are being used as dependents of other objects that are to remain in memory.

Answering **No** will delete only those objects that were selected.

Dump

This utility is for dumping selected objects over MIDI. If any of the selected objects have dependents that were not selected, you will be asked the question “Dump dependent objects?”

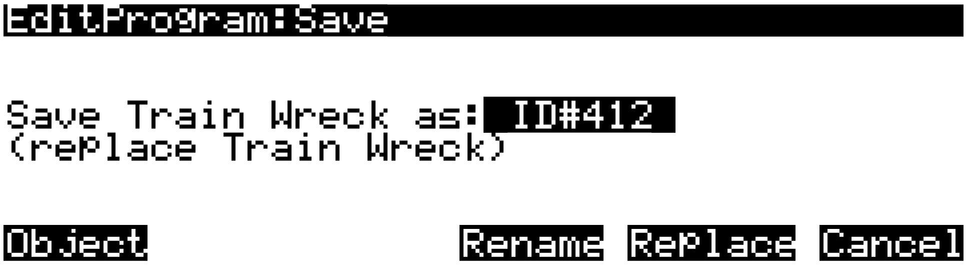
Press **OK** to initiate a MIDI System Exclusive dump of the selected set of objects, one by one out the MIDI Out port of the K2600. Dumping everything can generate massive dumps, so you should know the limits of the device you’re dumping to.You can cancel the dump at any time with the **Cancel** soft button.

Note that only sample objects (which contain the Start, Alt, Loop, and End points, as well as the values of all parameters found on the MISC page in the Sample Editor) are dumped by this utility, and not RAM sample data. (RAM sample data can be dumped via the MIDI Sample Dump Standard from within the EditSample page. See Chapter 6 of the *Musician’s Reference* for more information on the MIDI Sample Dump Standard.) Dumping the sample object of a RAM sample is not very useful because the sample memory address ranges are fixed in the object. This means that if you load the sample object back into the K2600 via MIDI, there is virtually no way it will point to and play back the same area of sample memory as when it was dumped, let alone the same sample data. However, this can lead to some interesting results.

Sample objects that reference the K2600’s ROM sample area will reference the same area when you load them back in via MIDI.

Using the Object Utilities from the Editor

You can get to the object utilities while editing any object. This is provided as a convenience, for example to be able to do certain housekeeping work such as deleting samples to free up room in your sample RAM, or making copies of objects. Access to the utilities can be done by pressing the **Object** soft button from any Save/Replace dialog in the editor:



EditProgram:Save

Save Train Wreck as: ID#412
(replace Train Wreck)

You can get to this Save/Replace dialog when editing an object by either exiting after you have modified the object, or pressing the **Save** soft button.

If you try to use the Copy utility to copy the exact object you are in the process of editing, you will make a copy of the edited version. With sample objects this would be one way to save off a copy sample that references a small part of a much larger sample. You could remain in the Sample Editor, and continue to edit the larger sample, by pressing **Done** followed by **Cancel** after making the copy. This may be a faster way to save many “snippets” out of a sample than continually reentering the Sample Editor after saving copy samples to different IDs.

When using the Object Utilities from within the editor, you must be careful not to delete any of the objects you are currently editing. This could have unpredictable results.

Chapter 12
Song Mode and the Song Editor

Getting Started with the Sequencer

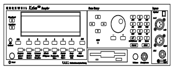
The K2600's sequencer is a powerful and versatile tool for songwriters, composers, and anyone else who wants to record and play back songs. As with any tool, however, it's best to start with the basics. This section begins with a tutorial where you will record a song, then shows some of the mixing capabilities of the sequencer. If you are familiar with other sequencers, you will have no problem using Song mode in the K2600. Read through this section, however, to learn about the features that make the K2600's sequencer unique.

What is a Sequencer?

A sequencer is similar in some ways to a multi-track tape recorder: you can record and play back all sorts of music and sounds, layer sounds on top of other sounds, and change or manipulate things that you've previously recorded. Unlike a tape recorder, however, you do not actually record sounds with a sequencer. Rather, you are recording commands that cause sounds to be played. Nonetheless, we will sometimes explain sequencer features by drawing analogies to familiar tape recording techniques such as splicing and overdubbing.

There are several advantages to recording a song by sequencing. For one thing, sequencer commands take up much less disk space than digitally recorded music would, so you can get a lot of information (that is, music) on a single floppy disk. Furthermore, you can easily make changes to your sequences. For example, you can change individual notes, transpose parts, or change instrumentation. Lastly, you can share the sequences you create with other musicians.

A Word about the Local Keyboard Channel



Before you being sequencing, we'd like to remind you about the Local Keyboard Channel parameter on the MIDI-mode RECEIVE page (described on page 10-7). Local Keyboard Channel is especially important for sequencing with the K2600R, since it enables you to record on different tracks without constantly switching transmit channels on your controller. Regardless of which K2600 model you own, you should do the following whenever the K2600 is going to receive MIDI information from an external source:

- Set the Local Keyboard Channel on the MIDI-mode RECEIVE page to a specific channel (1–16).

- Set your MIDI controller (keyboard, percussion controller, etc.) to transmit on the same channel.

Performing the above two steps means that you'll be able to hear the individual channels (each of which is assigned by default to a separate recording track) as you scroll through the different recording tracks in the K2600's Song mode. Local Keyboard Channel performs a *rechannelizing* function that makes this happen.

Patch Through

There’s one more use for Local Keyboard Channel. With any model of the K2600, the Local Keyboard Channel parameter lets you patch through (also known as soft through) to external sound modules. When Local Keyboard Channel is enabled, the K2600 takes the rechannelized information and sends it out the MIDI port. This lets you hear an external module while you are recording a track assigned to that module.

Tutorial: Recording a song

In this tutorial, we’ll record a song by using the steps described below. Bear in mind, though, that this is just one approach to sequencing a song. This example includes:

- Assigning programs to channels
- Recording a drum loop
- “Unlooping” the drum track and adding the remaining instruments
- Mixing the song

Assign Programs to Channels

Start by deciding what instruments you want to use in the song. Suppose you want to record a bass /drums / organ rhythm track with a lead instrument on top. You’ve decided to use the following programs:

- **54 Jazz Kit II**
- **30 Warm Bass 1^2**
- **22 Gospel Organ**
- **79 Modulead**

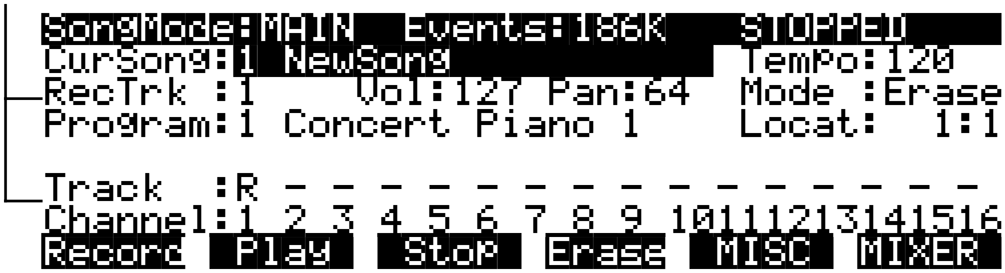
Set up your K2600 so that each of these instruments is on a separate MIDI channel. Since Song mode automatically assigns each channel to a separate sequencer track (1-16, consecutively), you’ll then be all set when you start laying down tracks, and won’t have to go scrolling through the program list. Don’t worry about changing your mind later, though, since you can always make changes after you’ve recorded your initial tracks.

If you use a KB3 program in a song, make sure that the channel to which you assign it is the KB3 channel; otherwise it won’t play. You could always change the KB3 channel to match the channel you want to use for recording the KB3 program, but we recommend deciding on a channel that will always be the KB3 channel, and keeping it that way. Also keep in mind that KB3 programs require one voice of polyphony for every two tone wheels in the program. Since most KB3 programs use at least 79 tone wheels, that leaves only eight voices free for other programs.

Aside from the KB3-channel issue, it doesn’t really matter which channel you use for the programs you want to record. In this example, we’re going to put the drums on Channel 1 and the bass, organ, and lead on Channels 2, 3, and 4.

Follow these steps to assign the programs to separate channels:

1. Press the **Song** mode button to enter Song mode. The display will look something like this:



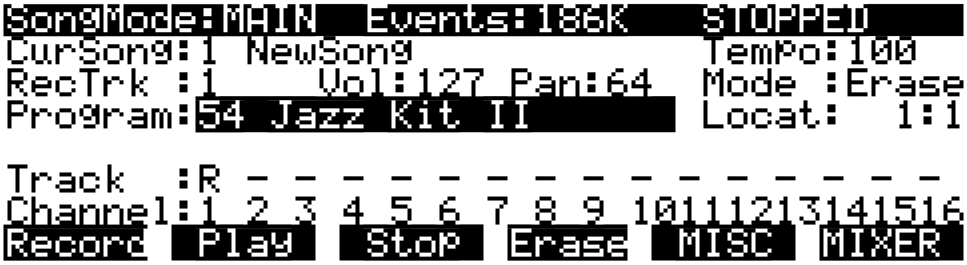
Recording track is set to Track 1

```
SongMode:MAIN  Events:186K    STOPPED
CurSong:1 NewSong               Tempo:120
RecTrk :1    Vol:127 Pan:64  Mode :Erase
Program:1 Concert Piano 1    Locat:   1:1

Track  :R - - - - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record  Play   Stop   Erase  MISC   MIXER
```

   Notice that the sequencer is ready for you to record a new song, and the recording track (RecTrk) is set to track 1. If **1 NewSong** doesn't appear as the value for the CurSong parameter, press the **MISC** soft button, then press the **New** soft button on the MISC page. You'll return to the MAIN page, and CurSong will say **1 NewSong**.

2. Use the **Down** cursor button to move the cursor to the Program parameter.

3. When this parameter is highlighted, type **54** then press the **Enter** button. You've now assigned **Jazz Kit** to channel 1.

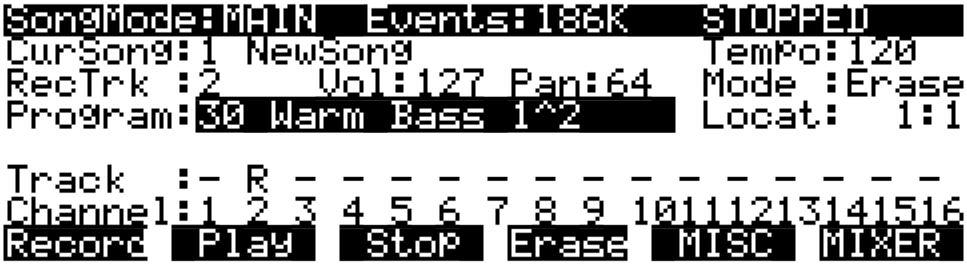


```
SongMode:MAIN  Events:186K    STOPPED
CurSong:1 NewSong               Tempo:100
RecTrk :1    Vol:127 Pan:64  Mode :Erase
Program:54 Jazz Kit II        Locat:   1:1

Track  :R - - - - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record  Play   Stop   Erase  MISC   MIXER
```

4. Press the **Up Chan/Bank** button. This changes the RecTrk parameter to 2, but leaves the Program parameter highlighted. Notice how the R in the Track region moves from Track 1 to Track 2. Also notice how each of the sixteen tracks has a default channel associated with it. You could change this if you wanted to, but most people find it easiest to associate track 1 with channel 1, track 2 with channel 2, and so on.

   Incidentally, when you're assigning programs on this page, you could use the **Up/Down** cursor buttons to highlight RecTrk, then change the recording track and select the Program parameter again, but the **Chan/Bank** button method is more convenient.

5. On Recording Track 2, select Program **30 Warm Bass 1^2**.



```
SongMode:MAIN  Events:186K   STOPPED
CurSong:1 NewSong            Tempo:120
RecTrk :2    Vol:127 Pan:64  Mode :Erase
Program:30 Warm Bass 1^2     Locat:   1:1

Track  :- R - - - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record  Play  Stop  Erase  MISC  MIXER
```

6. Repeat the above two steps to assign **22 Gospel Organ** to Channel 3 and **79 Modulead** to Channel 4.

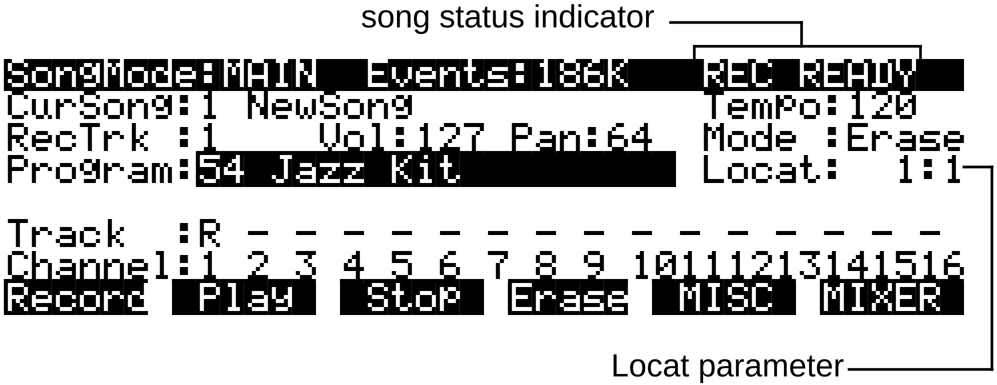
You’ve now chosen the programs for your first sequence. It’s important to realize, though, that you have not recorded anything yet. The programs will be there when you need them, but they have not yet been included in a song. Also, don’t forget that you can change the program assignments any time before or after you record the song.

Record a Drum Loop

Our song will be based around a four measure drum loop that we’ll record now. Later on, we’ll “unloop” the drum track for additional recording.

The length of the drum loop is determined by the current *endpoint*, so we’ll start by recording four measures of silence to set the endpoint.

1. Set RecTrk to **1** then press the **Record** soft button. The Song Status indicator (top right-hand corner of the display) flashes REC READY.



song status indicator

```
SongMode:MAIN  Events:186K   REC READY
CurSong:1 NewSong            Tempo:120
RecTrk :1    Vol:127 Pan:64  Mode :Erase
Program:54 Jazz Kit          Locat:   1:1

Track  :R - - - - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record  Play  Stop  Erase  MISC  MIXER
```

Locat parameter shows current measure and beat number

2. Press the **Play** soft button. The Song Status indicator now reads RECORDING.

   The K2600’s built-in metronome begins clicking, and the Song-mode LED blinks in time with the current tempo.

   Notice the Locat parameter on the right side of the display, which shows the current Bar and Beat number. When you begin recording, the K2600 provides you with a four-beat countoff, during which time Locat’s Bar and Beat number are preceded by a minus sign.

You can change the length of the countoff by changing the value of the Countoff parameter on the MISC page.

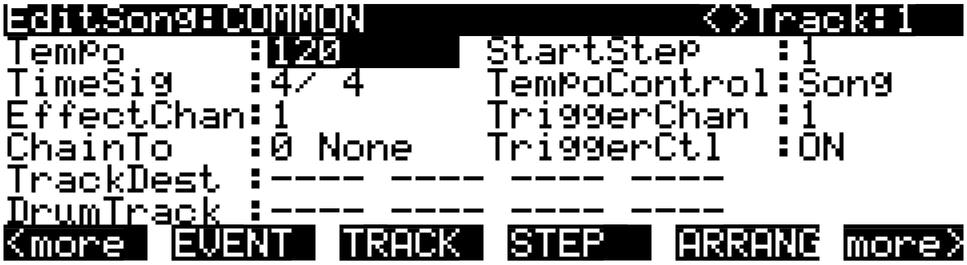
Since we are recording four measures of silence, press the **Stop** soft button as soon as Locat reads **5:1**.

NOTE*: The sequencer will truncate to the nearest downbeat, so as long as you press Stop before Locat reads* ***5:2*** *(but after it reads* ***4:4****) you'll be OK. Don't worry about this too much, though, since in the next step we'll show you how to check (and change, if necessary) the endpoint.*

When you press **Stop**, you'll be asked to if you want to save this song. Even though you've just recorded four measures of silence, go ahead and save it; this makes it an official song object. By the way, it's pretty easy to rename the song at this stage. Just press the Rename soft button during the Save Song dialog

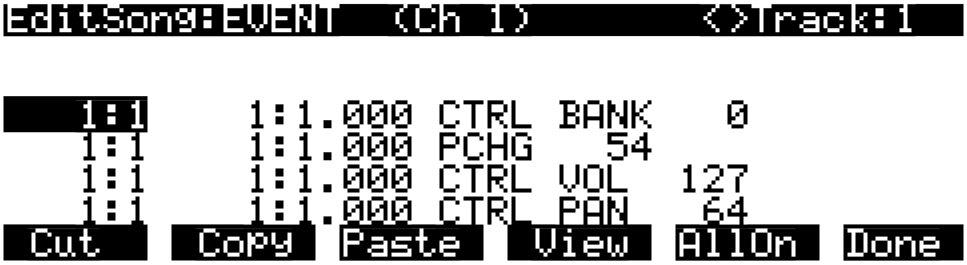
For the purposes of this example, we'll assume that you pressed **Stop** a few beats too late. You'll see how easy it is to correct this sort of thing in the Event Editor.

3. On the Song-mode MAIN page, make sure that the cursor is highlighting any parameter other than the Program parameter, then press the **Edit** button. The COMMON page appears. (If the Program parameter had been highlighted when you pressed **Edit**, you would have entered the Program Editor, which is not what you want to do right now.)



```
EditSong:COMMON                <>Track:1
Tempo     :120        StartStep   :1
TimeSig   :4/ 4       TempoControl:Song
EffectChan:1          TriggerChan :1
ChainTo   :0 None     TriggerCtl  :ON
TrackDest :---- ---- ---- ----
DrumTrack :---- ---- ---- ----
<more   EVENT   TRACK   STEP   ARRANG  more>
```

4. Now press the **EVENT** soft button to bring up the Event Editor, which looks something like this:

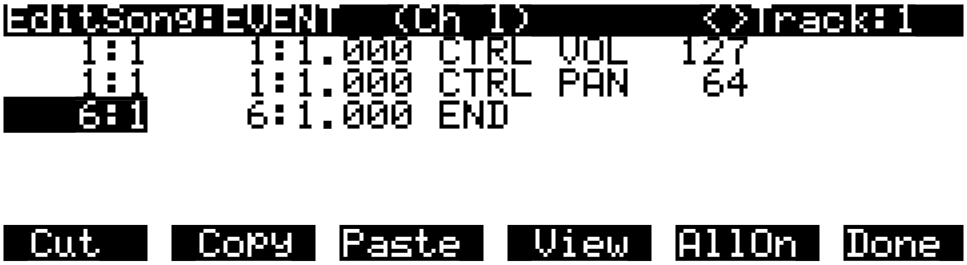


```
EditSong:EVENT   (Ch 1)        <>Track:1


     1:1     1:1.000 CTRL BANK    0
     1:1     1:1.000 PCHG    54
     1:1     1:1.000 CTRL VOL   127
     1:1     1:1.000 CTRL PAN    64
 Cut     Copy    Paste    View   AllOn   Done
```

The Event Editor gives you access to an editable list of all note events, controller events, and other MIDI events that Song mode uses to describe your sequence. While you're looking at the Event Editor, notice the data that the sequencer records, even when no notes are played.

5. Use the Alpha Wheel to scroll to the bottom of the event display.

The last event listed is the endpoint, which should correspond to the first beat of the Bar following the last Bar in the song. For our four measures of silence, then, the endpoint should be **5:1**. The display below, however, shows that five measures have been recorded:



```
EditSong:EVENT   (Ch 1)        <>Track:1
   1:1     1:1.000 CTRL VOL   127
   1:1     1:1.000 CTRL PAN    64
   6:1     6:1.000 END




 Cut    Copy   Paste   View  AllOn  Done
```

Fortunately, it’s quite easy to change the endpoint from **6:1** to **5:1** to remove the extra measure that’s been accidentally recorded. (If your endpoint is at **5:1**, you won’t need to change it now. If, however, your endpoint is less than or greater than **5:1**, you should proceed with the next step.)

6. Check the endpoint, and change it if necessary.

   To change the endpoint from **6:1** (or any other incorrect value) to **5:1**, press the **Right** cursor button to position the cursor in the Bar:Beat:Tick column (**6:1.000** in the above example). Type **51000** then press the **Enter** button. The endpoint is changed, and the song is now four measures long.

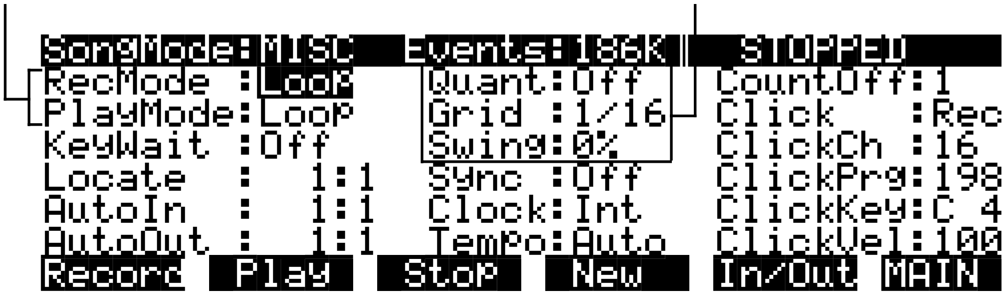
   Press the **Done** soft button to return to the COMMON page in the Song Editor. If you changed the endpoint while in the Event Editor, save the song now (you may have to press one of the **more** soft buttons to see the **Save** soft button).

7. Press **Exit** to return to the MAIN page in Song mode.

8. Make sure that the RecMode and PlayMode parameters are set to **Loop**.

   From the MAIN page, press the **MISC** button. Set the RecMode parameter on the MISC page to **Loop**.

   The PlayMode parameter should already be set to **Loop**. If it isn’t, turn the Alpha Wheel until **Loop** is highlighted.



RecMode & PlayMode parameters

Quantize parameters

```
SongMode:MISC   Events:186K    STOPPED
RecMode :Loop   Quant:Off     CountOff:1
PlayMode:Loop   Grid :1/16    Click   :Rec
KeyWait :Off    Swing:0%      ClickCh :16
Locate  :   1:1 Sync :Off     ClickPrg:198
AutoIn  :   1:1 Clock:Int     ClickKey:C 4
AutoOut :   1:1 Tempo:Auto    ClickVel:100
Record  Play   Stop   New   In/Out  MAIN
```

9. Turn on input quantization.

While you're on the MISC page, take a look at the quantize parameters (Quant, Grid, and Swing) in the middle column of the page. Quantization is a very useful feature, especially if you're having a bad rhythm day. When you quantize a track, the sequencer moves the elements of that track closer to a grid based on the time signature of the song. You can use quantization to tighten up a rhythm track subtly, or to create a precise, unwavering mechanical rhythm.

For our drum loop, we'll try the total quantization experience, so position the cursor over the Quant parameter, and turn the Alpha Wheel until the value is set to **100%**. Move the cursor down to the Grid parameter. The default value of **1/16** indicates that quantization will move the notes you play to the closest 16th-note division in the Bar. Try double-pressing the **Plus/Minus** buttons below the Alpha Wheel to move through a range of useful grid values. Note that some of the values have **tr** or **t** appended to them. These are grid settings that allow you to maintain a triplet feel. We'll use a setting of **1/16**, so return to this value if you've changed it, then press **Exit** or the **MAIN** soft button to return to the MAIN page in Song mode.

***NOTE**: The K2600's sequencer also provides a full range of advanced quantization features that you can apply to previously recorded tracks. To learn about these, check out the Quantize and Reference Quantize functions on the TRACK page in the Song Editor See page 12-47.*

10. Make sure you are in Merge mode (it's the default, so you'll be in Merge mode unless you've changed the value of the Mode parameter). This is important, because you want to be able to overdub on the track as it loops. (In Erase mode, you would erase all existing notes every time the loop came around.) To activate Merge mode, go to the MAIN page and set the Mode parameter to a value of **Merge**.

11. Begin recording drums.

    Press the **Record** soft button (observe the REC READY indicator on the top line) then press the **Play** soft button when you're ready to begin. Remember to wait for the four beat countoff before you start to play.

    Since you are in Merge mode, you don't need to do everything at once. A common approach to making drum loops is to record a different voice each time the loop comes around. For example, on the first loop you could record snare hits on the back beats (1:2, 1:4, 2:2, 2:4, etc.). Then you could add kick drum to the snare when the loop comes around again; you'll be able to hear the previously recorded part, as well as the new part. On the third pass you might record ride cymbal, followed by hi-hat or other percussive accents. Keep it simple at first, because you can always save the part while it's basic (but correct), then make additions later. To keep track of where you are, watch the flashing Song-mode LED or the Locat parameter on the MAIN page.

12. Press the **Stop** soft button when you've finished recording the drums.

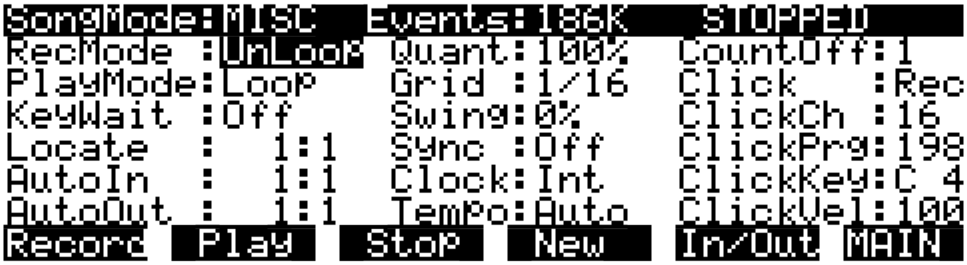
    Save the changes to your song by pressing the **Yes** soft button followed by the **Replace** soft button.

Record a Bass Line

When you are satisfied with your drum loop, you can begin using it as the foundation for a song. What we'll do here is set RecMode to **Unloop** while leaving PlayMode set to **Loop**. This means that the drum loop will keep playing while we record new unlooped material of any length. The endpoint of the song will change to reflect the length of the newly recorded material.

1. Press the **MISC** soft button to bring up the MISC page.
2. Set the RecMode parameter to **Unloop**. Leave the PlayMode parameter set to **Loop**.

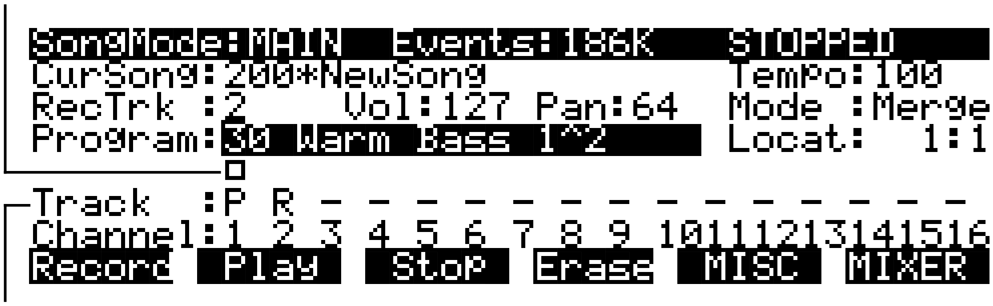
   Depending on the type of song you are recording, you may also want to turn quantization off before you record your bass part.



```
SongMode:MISC   Events:186K    STOPPED
RecMode :UnLoop Quant:100%   CountOff:1
PlayMode:Loop   Grid :1/16   Click   :Rec
KeyWait :Off    Swing:0%     ClickCh :16
Locate  :   1:1 Sync :Off    ClickPrg:198
AutoIn  :   1:1 Clock:Int    ClickKey:C 4
AutoOut :   1:1 Tempo:Auto   ClickVel:100
Record  Play   Stop   New    In/Out  MAIN
```

3. Press the **MAIN** soft button to return to the MAIN page.
4. Set the recording track (RecTrk) to Track 2.

This track already has
material recorded on it.



```
SongMode:MAIN   Events:186K    STOPPED
CurSong:200*NewSong            Tempo:100
RecTrk :2    Vol:127 Pan:64    Mode :Merge
Program:30 Warm Bass 1^2       Locat:   1:1
        □
Track  :P R - - - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record  Play   Stop   Erase  MISC   MIXER
```

Track status indicators: track 1 is set to Play, track 2 is set to Record.
Tracks 3 through 16 are empty.

   Since you previously assigned **Warm Bass 1^2** to channel 2, it should appear in the Program parameter when you set Track 2 as the recording track. Note, too, that the track status indicator for Track 1 changes to **P** (for **Play**) when you select Track 2 for recording. The small square above the track status indicator tells us that material is contained on that track.

5. Press the **Record** soft button to enter REC READY mode.
6. Press the **Play** soft button, then begin laying down a bass track.

Remember that by default there is a four-beat count off, during which time the Locat value is preceded by a minus sign (-). No material is recorded during the count off, though anything you play during the countoff gets quantized to the first Beat of the song. As you are recording the bass track, your drum loop will keep playing. Play for as long as you want; the sequencer will lengthen the song as needed.

7. Press the **Stop** soft button when you are done recording the bass.

   You will be given the usual save options. To keep what you've just recorded, press the **Yes** soft button followed by the **Replace** soft button.

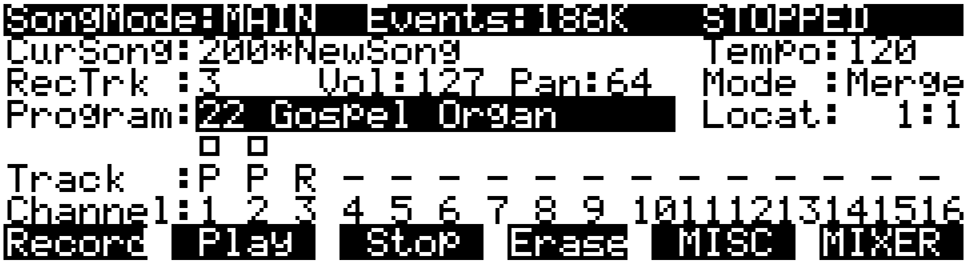
   Since you unlooped the drum track when you recorded the bass, you've changed the endpoint of the song to be wherever you stopped the bass track. You can check the endpoint (and change it, too, if you want) using the Event Editor, as described earlier.

Record the Remaining Instruments in Your Song

Now that you've defined your song with the bass and drum tracks, you can put the organ and lead (or whatever instruments you've chosen) into your song.

1. Set the recording track (RecTrk) on the MAIN page to Track 3.

   Notice the small squares above the track status indicators for Tracks 1 and 2, reminding you that you've now got material on two tracks.



```
SongMode:MAIN   Events:186K    STOPPED
CurSong:200*NewSong            Tempo:120
RecTrk :3    Vol:127 Pan:64  Mode :Merge
Program:22 Gospel Organ        Locat:   1:1
        □  □
Track  :P  P  R  -  -  -  -  -  -  -  -  -  -  -  -  -
Channel:1  2  3  4  5  6  7  8  9  10111213141516
Record   Play    Stop   Erase   MISC   MIXER
```

2. Press the **MISC** soft button to bring up the MISC page.

3. Set RecMode to **FixLen**.

   Since you've defined the length of your song with the bass track, setting RecMode to **FixLen** means that the song will play through only once each time you record a new part.

4. Record the organ in the same way that you recorded the bass track in the previous section.

   Notice that you can do this from the MISC page, without returning to the MAIN page.

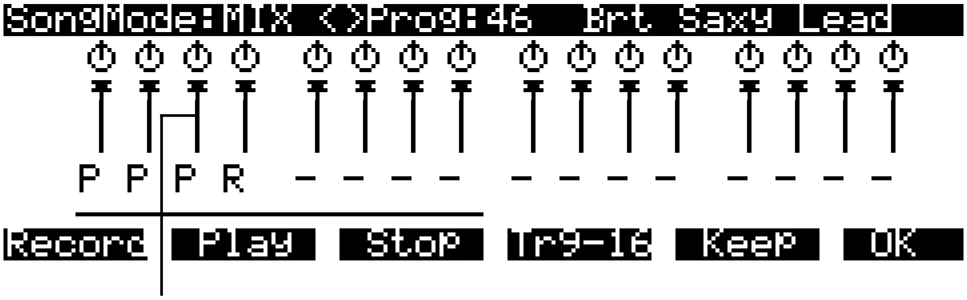
5. Continue recording instruments until you have played all the parts of your song.

Mix Your Song

The MIX page lets you change the panning and volume levels for the tracks in your song. Needless to say, this is one of the most important steps in the completion of your song production, and potentially one of the most creative. This example will keep things simple by showing you how to change the volume level of one of your instruments. We'll also take a quick look at the track mute feature.

1. Press the **MIXER** soft button to bring up the MIX page.

   The icons that represent pan-position knobs and volume-level faders resemble the controls on a traditional mixing board. Manipulating them should be quite intuitive. Simply position the cursor over a pan position knob or volume level fader, then turn the Alpha Wheel to set the level you want. On keyboard models, the sliders control the volume of the current bank of channels (as indicated by the line near the bottom of the display). In the diagram below, the sliders would control the volume on Channels 1–8.



SongMode:MIX <>Prog:46 Brt Saxy Lead

P P P R - - - - - - - - - - - -

Track 3 Volume Level Fader

   For example, suppose you want to turn down the organ on Track 3:

2. Use the **Right** or **Left** cursor button to position the cursor over Track 3's volume level fader on the MIX page.

   Although the tracks aren't numbered on the MIX page, they're laid out logically: left-to-right, from 1 through 16 consecutively. Track 3, then, is the third track from the left.

3. Use the Alpha Wheel to turn down the volume of the track by changing the position of the volume slider.

4. Press the **Keep** soft button and save the change.

   Now when you play back the song, Track 3's volume starts playing at the newly set level.

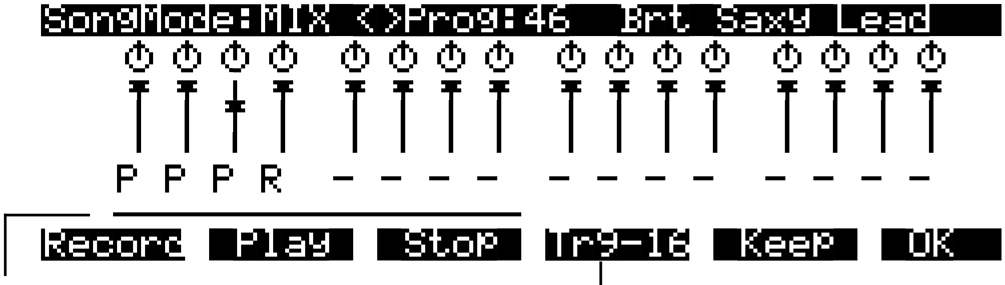
   You can also record real-time volume and pan changes from the MIX page or enter numeric values for these parameters on the MAIN page.

Using the Mode Buttons to Mute a Track

Finally, we'll take a quick look at the track mute feature, which lets you use the K2600's mode buttons to mute individual tracks. This can be invaluable during mixdown.

You may have noticed a horizontal line underneath the sliders for Tracks 1 through 8 on the MIX page. As the illustration below shows, this is the "Quick Mute Track Bank Indicator," showing which bank of eight tracks will respond to the eight track-mute buttons. The eight mode buttons

(Program, Setup, etc.) double as track-mute buttons while you're on the MIX page; press one or more of them and the corresponding tracks are muted Press them again to bring the track back into the mix.



SongMode:MIX <>Prog:46 Brt Saxy Lead

Quick Mute Track Bank Indicator

Toggles track muting between tracks 1-8 and tracks 9-16.

Use the **Tr 1-8** / **Tr 9-16** soft button to toggle between banks of eight tracks, either 1–8 or 9–16. When you press this soft button, the horizontal bar repositions itself below the affected tracks. Table 12-1 shows the K2600's mode buttons, and which tracks they mute when you're on the MIX page:

Available Buttons	**1-8**	**9-16**
Program	1	9
Setup	2	10
Quick Access	3	11
Effects	4	12
MIDI	5	13
Master	6	14
Song	7	15
Disk	8	16

Table 12-1 Track Muting in Song Mode

Let's listen to our song with and without drums:

1. On the MIX page, press the **Play** soft button. The song, with all of its instruments, begins to play.
2. Press the **Program** mode button on the front panel. The Program-mode LED lights, and Track 1 (the drum track) is muted.
3. Press the **Program** mode button again to unmute the drum track.

   Using your K2600's front panel buttons, you can mute one or more tracks at once, or even mute eight tracks at a time.

Tutorial: Arrangements

The Arrangement Editor gives you a variety of ways to record and play songs. The following tutorial is designed to walk you through the steps of creating songs using the Arrangement Editor. Here are some typical tasks:

- Record two or more songs, then string them together in any order, as steps in an arrangement.
- Record additional tracks over the steps in an existing arrangement, saving the new tracks in the arrangement song.
- Use up to 32 tracks for recording and playback.
- Trigger songs or arrangements by striking keys (or triggering notes from any MIDI controller)—an excellent feature for live situations.

Creating an Arrangement

This involves recording and saving each section of your final piece of music as a separate song, then using an arrangement to string together the sections in any order you choose. There are a couple reasons you might want to do this.

First of all, many pieces of music tend to be composed in sections. If your music tends to feature various sections repeated in varying order, using the Arrangement Editor can be easier than copying those sections and pasting them into a single song.

The second reason has to do with memory requirements. No objects (aside from samples) can be larger than 64k (10,000 to 16,000 notes, depending on the amount of controller information you record). You may need to use arrangements to accommodate the size of long musical pieces.

When you're recording songs that you intend to combine into arrangements, it can be helpful to name the songs Part 1, Part 2… or Chorus, Verse… That way, when you are putting together the arrangement, you can quickly identify each section. You might also want to organize the IDs of the songs. For example, you could start numbering the songs at the second ID in a memory bank (**301**, **302**…), then save the arrangement song at the first ID in the bank (**300**, in this example). That way, you can create a file containing the arrangement song and its constituent songs, and when you load it into the K2600, the arrangement (the one you're likely to want to play) will be the first song in the bank.

Once you have your songs, you're ready to create another song to use as the arrangement.

1. In Song mode, select **1 NewSong** as the value for the CurSong parameter. This is important, because the arrangement song should not contain any note or controller information.
2. Press **Edit**, and set the Tempo parameter to match the tempos of the songs you'll be adding to the arrangement. (This assumes they are all the same tempo. We'll cover how to deal with different tempos later on.)
3. Press either **more** soft button, then press **Save**. The save dialog gives you the opportunity to rename the song and give it the ID of your choice. Press **Save** again when you've made the changes you want.
4. Press the **ARRANG** button (you're still in the Song Editor). The top line of the display tells you that you're looking at Step 1 of a song that contains one step. Cursor down to the Song parameter and select the song you want to use for the first step in the arrangement.

5. Press **Add** to add another step to the arrangement. Select the song you want for the second step. Continue adding steps and selecting songs as needed. If you need to get rid of a step, select the step and press the **Delete** soft button.

   If you press the **Play** soft button while on this page, you'll hear the currently selected step. (If the step's Mode parameter is set to Next, the K2600 will play the next step when the current step finishes.) Once you have more than one step in an arrangement, you can use the **Chan/Bank** buttons to scroll through the steps.

6. Save, then press **Exit** to return to the MAIN page in Song mode. Notice that there are now three dashes below the Program parameter. Press **Play**, and the dashes change to indicate the current song, and the current step. The Locate parameter shows the bar number of the arrangement, not the bar number of the current step. In other words, if the first step in an arrangement has 16 bars, then when Bar 1 of Step 2 is playing, Locate will show **17:1**, not **1:1**.

Arrangement Parameters: the ARRANGE page

Mutes

The Mutes parameter has 16 values, in groups of 4. By default, tracks are unmuted (active), as represented by the dashes. To mute a track, move the cursor to highlight the dash corresponding to the track you want to mute, and press either **Plus/Minus** button to change the dash to **M**. This mutes the track.

Muting tracks is a good way to make several different-sounding steps out of the same song. If you create a multi-step arrangement in which each step uses the same song—with different tracks muted in each step—you can bring different parts in and out in each step.

Xpose

You can transpose any step up or down. This allows you to transpose the song without having to edit the actual song data. Every track in the step gets transposed.

This is great for many sounds, but not so useful for steps containing programs that have different timbres assigned to different keys—like drum programs. Fortunately, you can designate any track as a drum track, which prevents it from getting transposed when you transpose the step. You must designate drum tracks in the individual songs that are used as steps in the arrangement, not in the arrangement song.

To designate a drum track, go to the COMMON page for the song containing the track in question. Find the DrumTrack parameter, and move the cursor to highlight the dash corresponding to the track in question. Press either **Plus/Minus** button to change the dash to a **D**. Don't forget to save.

Times

You can set any step to play from 1 to 120 times before stopping or continuing to the next step in the arrangement.

Mode

This is normally set to **Next**. In this case, the arrangement will play the next-highest-numbered step once the current step is finished. If the last step is set to **Next**, the arrangement will repeat Step 1. To make the song stop after the last step, set the last step's Mode parameter to **Stop**.

Arrangement Parameters: the COMMON page

Start Step

This determines the step number where the song will start. Normally this is **1**, but it can be any step in the arrangement.

Tempo Control

You can choose whether the tempo is controlled by the arrangement song itself or by each song in the arrangement. When set to **Song** it will use the tempo and time signature set in the arrangement song itself. When set to **Arrange**, it will use the various tempos and time signatures used in the arrangement's constituent songs.

Timing Issues

You might encounter timing problems when using the Arrangement Editor; notes can be delayed when the arrangement switches from step to step. Here's how to avoid the problem.

Each time you record a track for the first time, the K2600 places four events right at the beginning of the track: Bank Change, Program Change, Volume, and Pan. If you go into the Event Editor, you'll see these four messages appearing at 1:1:000. Normally they'll be the first four events you will see.

By the time you have recorded several tracks, these events start to become a large number of events all occurring at the exact same point in time. The K2600 processes these events sequentially, and if enough events happen at the same time, some of them get delayed. For example, if you have 10 tracks, then you will have 40 of those initial events, in addition to any note and controller info that also exist at 1:1:000 in a track.

To prevent the delays caused by too many events, you can delete unneeded events. Typically, you are probably not changing the Program, Volume, and Pan settings for each track when you switch from step to step. For example, quite often you might have the same program on a MIDI channel for all the steps. In this case, all of the program and bank change messages after the first step are not needed, and can be deleted. You can edit each step in the arrangement this way (don't remove these events from the first step, however).

There are two different methods you can use to get rid of these messages in a track. The simplest way is to go into the Event Editor. Use the **Chan/Bank** buttons to select the track whose events you want to edit—you can select each track individually, or select All to view the events from every track in the step. To remove an event, highlight it and press **Cut**.

The second way is to use the Erase function in the Track Editor. Again, choose the track with the **Chan/Bank** buttons. Set the From parameter to **1:1** and the To parameter also to **1:1**. Of course, you won't want to have Events set to All, or you'll eras any note events that occur at 1:1, as well as the unwanted events. If you set Events to **Program Change** and press **Go**, you'll erase the Program and Bank change events. You can then set it to **Controllers**. At this point, if you leave Ctl set to **All**, you can delete both the Pan and Volume events with one operation. But if you have other controller info that occurs at 1:1 (such as Mod Wheel or sustain pedal) then you would also be erasing those events. So you can use the Ctl parameter to select just Volume and just Pan, pressing **Go** after each selection.

In general, the Event-editor method is quicker if you are working on one track at a time. But if you have many tracks and know that you want to erase these events from all of them, using the Track Editor with all tracks selected is faster.

Of course, if you are changing the program changes, pan, or volume in a track when the song changes from step to step, you need to leave those events in, but typically you might be doing that in only one track, while six or seven other tracks stay the same.

Timing problems can also occur between steps due to improperly-located endpoints. If you have a timing problem, you should check in the Event Editor to make sure the end point of each step falls on the first beat of the bar *after* the last bar of the step. For example, if your step is 8 bars long, the end point should be at 9:1:000.

Removing Initial Events from Step One

Since you almost always want to have initial Program, Bank, Volume, and Pan events in an arrangement to make sure it plays properly, it makes sense to have those events in each track of the song used for the first step on the arrangement. But what if that same song is used in a later step, or Step 1 plays a number of times? In this case, as soon the step restarts, you have unneeded events that could contribute to delays.

In this case, the solution is to delete those initial events from the Step 1 song, and record them into the arrangement song. In the arrangement song, select a recording track, press **Record**, then highlight the Program parameter and select the program you want for the track. You can also select the Pan and Volume parameters if you want to set them to a specific volume. Once you have these parameters set, press **Stop**. You need to do this for each track that you are using in the song.

Recording Additional Tracks

So far, all of the recording we've described has been done in the individual songs used as steps. The arrangement song we created has no data in it.

But you can also record tracks in the arrangement song. For example, you might want to record a series of rhythm section grooves: just bass, drums, and maybe some comping parts. Now you can use those grooves as step in an arrangement, then record lead lines through the entire arrangement.

1. Follow Steps 1 through 6 of *Creating an Arrangement* on page 12-12 to create an arrangement song, using some different grooves you have created.
2. Start recording new tracks in the arrangement song. Remember that each MIDI channel can have only one program assigned to it. Therefore you may want to select the tracks you record in the arrangement song so that their MIDI channel assignments are different from those of the tracks in the step songs.

Maximizing Track Use

If you extend the previous example, you'll realize that you can actually make use of 32 MIDI channels—by creating an arrangement containing steps that use all 16 channels, then recording 16 channels of music in the arrangement song itself. Both the step songs and the arrangement song can play back through the K2600, through another instrument connected to the K2600's MIDI Out port, or through both.

It is important to remember that there are still only 16 MIDI channels, and any one MIDI channel can play only one program. Therefore there is no way to have more than 16 different programs playing at the same time on the K2600. But there are two reasons why you would want to use more than 16 tracks.

First, you can have two or more tracks assigned to the same MIDI channel. For instance, if you were recording drums, you might want to put different drums from the same program on different tracks, to make recording and editing easier. On the bottom line of the display on the MAIN page in Song mode, there are 16 channel parameters, one for each track. The numbers don't represent *tracks*; they represent the track's MIDI channel assignment. (The dashes—or other characters—above the numbers represent the tracks. The dashes and characters are the values for the Track parameter.) You can assign any track to any MIDI channel.

The second way you can use more than 16 tracks is if you have an external sound module in addition to the K2600. Each track can be assigned to play only the K2600's internal sounds (local), or to be sent only to the MIDI Out, to play the external instrument. To assign the track for local or MIDI playback, go into the Song Editor and on the COMMON page you'll see 16 Track Destination parameters. If the value is a dash, the track is going to both the K2600 and its MIDI Out port. **L** means local, and plays the K2600 only. **M** means that the track goes only to the K2600's MIDI Out port. A value of **x** means that the track is muted.

In the following example, all the tracks in the step song play the K2600, while all the tracks in the arrangement song go to the MIDI Out port. You can also have the step songs and arrangement song set to the same track destinations, as long as it's OK for them to play the same sounds.

1. Create a song with multiple tracks and save it.
2. On the COMMON page in the Song Editor, set the TrackDest value for each track of this song to **L**. Save, then press **Exit** to return to the MAIN page in Song mode.
3. Using the Cursong parameter, call up **1 NewSong**, and press **Edit**, then **ARRANG**.
4. Using the Song parameter, call up the song you just recorded. This makes your song a step in the arrangement song, which you're currently editing. Press **Done** to return to the COMMON page.
5. Set the TrackDest parameter for each track to M, so that the arrangement song won't also play the K2600.
6. Save, then press **Exit** to return to the MAIN page in Song mode.
7. Record additional tracks as part of the arrangement song. You might have to record programs changes in each track of the arrangement song to set up the external instrument properly.

Triggering Arrangement Steps From the Keyboard

Each step in an arrangement can be triggered by playing a key (or triggering a note from any MIDI controller). This can be great for live performance, because you can repeat each step as many times as you like.

1. Go to the TRANSMIT page in MIDI mode, and assign a control setup that has a value of **On** for the Sync parameter on the COMMON page of the Setup Editor. Create one if you need to, and name it **SongSetup**. We'll explain why shortly why you need to use this control setup.
2. Go to Song mode, and create an arrangement, following Steps 1 through 6 of *Creating an Arrangement* on page 12-12.
3. Go to the COMMON page in the Song Editor. There are two parameters on this page for use with key triggering of steps:

   TriggerChan: Notes on this MIDI channel can trigger the current step. Notes on any other channels will not trigger the step. Set this to match the MIDI channel of the K2600 or whatever controller you're using to trigger the steps.

TriggerCtl: This determine if the keys will trigger the steps. Set this parameter to **ON**, so when you hit the appropriate note on the trigger channel, the step will start playing. If you set it to a MIDI Controller number that has a physical controller assigned to it, then the keys will trigger the step only when the controller is on (for switch controllers) or above its halfway point (for continuous controllers).

4. Press **ARRANG**, then press the **Chan/Bank Down** button until you see Step 1 in the top line of the display. Set the Mode parameter to a value of **Stop**.

5. Note the values for the LoKey and HiKey parameters, then set them to **E 1** and **E 2** by doing the following:

   - Press the **SetRng** soft button

   - Strike E 1 on the K2600 or your MIDI controller

   - Strike E 2

   Note the new values for LoKey and HiKey. Now the current step starts playing when you strike E 1. Strike another note (say E 2) while the step is playing, and on the first beat of the next bar, the step repeats, and all the tracks that aren’t drum tracks get transposed up a corresponding number of semitones (in this case, an octave). Generally, when you’re triggering steps using keystrikes, you’ll hit the triggering key somewhere in the last bar of the current step. This causes the new step to start right after the end of the current step.

   You can also cause steps to start as soon as you hit the triggering note. To do this, go to the TRANSMIT page in MIDI mode, and assign a control setup that has a value of **Off** for the Sync parameter on the COMMON page of the Setup Editor.

6. Set the Latch parameter to a value of **On**. Now the steps will continue playing after you’ve released the triggering note.

7. Set the VelTrk parameter to a value of **Off** if you want the step to play back at the level at which you recorded it. Set it to **On** to vary the playback level according to the velocity of the keystrikes that trigger the step.

RAM Tracks

If you have the K2600 sampling option, you can create RAM Tracks, which combines Song mode with the sampler. The RAM Tracks feature enables you to create a sample during song playback, then have the K2600 do the work of building a program out of the sample. The K2600 also inserts the sample into the song so that it plays back in sync with the song.

Possibly the best feature of RAM Tracks is how it affects polyphony. By sampling the K2600’s audio output, you can condense an entire song into a single track that uses only two voices of polyphony.

Two Important Concepts

RAM Tracks and Song Playback

You don’t have to be *recording* a song to create a RAM track. The idea behind RAM tracks is that you can make a quick sample during playback of a song, then integrate that sample into the song. Of course, you *can* create a RAM track while recording a song, but it’s often best to take one step at a time.

A Matter of Timing

The song must *already be playing* when you start sampling. There are several ways you can do this:

- Start the song, then at the appropriate location in the song, press **Record** on the SampleMode page, and start the sample input.
- Set the Thresh parameter on the SampleMode page to a dB value—one you know you'll exceed with your sample input signal. Start the song, then start your sample input at the appropriate location in the song. This method makes it easy to sync your sample with the song.
- Set Thresh to **Key**. Start the song, then at the appropriate location, then trigger the sampler by striking a key either on the K2600 keyboard (or on a MIDI source that's controlling the K2600), and start the sample input. You must send the trigger on the K2600's current MIDI channel.

Creating RAM TRacks

1. Start by configuring the sampler: go to Program mode, and press the **Sample** soft button to go to the SampleMode page. Set the Input parameter to **Analog**, and set the Source parameter: **Ext** for an external source, **Int** if you want to sample the K2600's output. Set the Time parameter to give you enough time to record the sample you want. Set the Mon parameter to **On** if you want the K2600 to play the sample input through its audio output. Set the Mode parameter to Mono (L), Mono (R), or Stereo, depending on your input signal. Adjust the Gain parameter to bring the signal level as close to 0 dB as possible.
2. Go to Song mode, and call up a song that has at least one empty track. If you plan to start sampling right at the top of the song, you might want to set the Click parameter (on the MISC page) to a value of **Cnt**, which gives you a countoff before the song starts playing (the value of the CountOff parameter—also on the MISC page— sets the number of bars of countoff).
3. Set the RecTrk parameter to an empty track.
4. Start the playback of the song. On keyboard models, press the **Play/Pause** button. On rack models, press the **Left** and **Right** cursor buttons at the same time.
5. Press **Record** to start sampling. When you have the sample, press **Stop** (if you run out of time or sample memory before you press **Stop**, the K2600 stops sampling automatically). Either way, the song stops its playback as well.
6. The K2600 prompts you to strike a key to set the sample root. Strike any key (or trigger any note) that you want, or press **Default** to set the root at C 4. The K2600 shows you the maximum signal level, or if the sample clipped, it shows you the number of clips. It also prompts you to save the sample.
7. Press **Yes** if you like the sample, and the save dialog appears. We recommend naming the sample now, to make it easy to keep track of it. The name you give the sample will also be the default name for the song, when the time comes to save the song.
8. Once you've saved the sample, the K2600 asks you if you want to place the sample into the current song. When you press **Yes**, the K2600 asks you to pick the track you want to use for the sample's playback. Pick any empty track: the track must be empty because the

K2600 will eventually create a new program for the sample, and that program must be on its own MIDI channel. Press **OK** when you've selected a track.

9. The K2600 creates a program, and shows you the new program's ID. It then prompts you to strike a key to specify a note for triggering the sample during the playback of the song. It can be any key; the K2600 automatically handles the transposition required to ensure that the sample plays back at the right pitch.
10. Go to the MAIN page in Song mode, and on the recording track you set in Step 3, you'll see the new program. The program assigned to this track has the same name you gave the sample when you saved it. If you look at the events on this track (on the EVENT page in the Song Editor), you'll see a PCHG event that selects your new program, and a note event corresponding to the key you struck to set the sample trigger.
11. Repeat Steps 5 through 9 to create additional samples and insert them into the song. You can use the same track, or a different, empty track. For this example, we'll use the same track. In this case, when you press **OK**, the K2600 asks you if you want to add the sample to the program that's already being used for the track (the program you created when you pressed **OK** in Step 8. Press **Yes**. The K2600 prompts you to strike a key, as in Step 9. If you strike a key that's already being used by one of the samples in the program, the K2600 alerts you, and lets you to assign a different key (press **No**) or the same key (press **Yes—** although if you do this, then the song will trigger all samples that share the same trigger key, every time the trigger key gets played).
12. If you use the same track, the K2600 creates another layer in the program on that track, and assigns the sample to it. You can record up to 32 samples on this track, since a VAST program can contain up to 32 layers. If you use a different track, the K2600 creates another new program when you place the sample into the song.

When you've recorded all the tracks you want, you can edit the programs assigned to those tracks, using all the features available for VAST synthesis.

***Note**: There's no way to synchronize sample playback exactly to MIDI. While the K2600 handles the synchronization extremely accurately, it's possible for the sample to drift away from the song playback, at a rate of up to .5 milliseconds per minute of playback. That's a minuscule amount, but it might matter to you. If you need to be more precise, we recommend that you keep your sample time short, and record a relatively large number of short samples, as opposed to a small number of long samples. This will virtually eliminate the risk of your samples drifting out of sync with the song playback.*

Using Song Mode

Selecting a Song for Playback

Select the Song parameter with the cursor buttons, then use any data entry method to scroll through the list of songs. Press the **Play** soft button, and the song will begin playing. Press the **Stop** soft button, and the song will stop and "rewind" to the beginning. If you press **Play** while the song is playing, the song will stop and the play pointer will revert to the Locate parameter, and will show your current location in the song. At this point you have two options. If you press the **Play** button again, the song will continue from its current location. If you press the **Stop** soft button, the song will return to 0:0.

The K2600 automatically selects programs for playback based on the MIDI channel(s) and the programs assigned to them at the time the song was recorded. When you start playback, the

K2600 sends program changes, on all relevant channels, to its sound engine and to the MIDI Out port if the PChng parameter on the MIDI-mode TRANSMIT page is turned on.

If you want to use a different program for playback than the one originally recorded, you have two alternatives. First, you can edit the song, changing each individual Program Change event (PCHG) to reflect the desired program IDs. This is done on the EVENT page in the Song Editor. Or you can set the RecTrk parameter to the track on which you want the program change, highlight the program parameter, press **Record**, select the desired, program, and press **Stop**.

Another alternative is to defeat the song's recorded program changes, and manually set each MIDI channel used by the song to play the desired program. Start by setting the ProgLock parameter to a value of **On**. The ProgLock parameter is found on the MIDI-mode CHANNELS page; you can set it independently for each of the 16 MIDI channels. When a channel's ProgLock parameter is set to a value of **On**, the K2600's sound engine will ignore all program changes it receives on that channel, whether it's via MIDI or from the K2600's front panel, or from within the song.

When you've set the ProgLock for each of the channels used in the song, go to Program mode, select the MIDI channels used by the song, and assign the programs you want to use. When you return to Song mode and play the song, the automatic program changes are defeated, and the song plays the programs you assigned. When you set ProgLock back to Off, the originally-recorded program changes take effect again.

Effect Selection During Recording and Playback

When you're recording or playing back a song, the setting for the FX Chan parameter (on the Effects-mode page) determines which studio (plus FXMods) gets applied to the song. Only one studio can be applied at a time, even for multi-part songs using more than one program. Depending on the combination of values for the FX Mode and FX Chan parameters in Effects mode, the FX channel may automatically track the current MIDI channel. In this case, if you change the current MIDI channel during playback (or during a MIDI recording) the current FX channel (and consequently the current effect) will change also, which might not suit your needs. Consequently, there's a way to force the FX channel to remain constant during playback or MIDI recording, even if you move to another mode during the playback or MIDI recording and change the current MIDI channel.

This is done by setting the FX Mode parameter to a value of **Auto**, and the FX Chan parameter to a value of **Current—**which is the most generally useful combination of settings for these parameters. In this case, while you're in Song mode—even if you move to another mode during playback or MIDI recording—the value of the FX Chan parameter automatically changes to match the song's effect channel (the value of the EffectChan parameter on the COMMON page in the Song Editor). Therefore the effects applied to the song are determined by the program assigned to the song's effect channel, and will not change, even if you change the current MIDI channel during playback or MIDI recording.

Of course, changing MIDI channels during playback or MIDI recording will not change the FX channel if the FX Channel parameter is set to a value from **1** to **16**, or if the FX Mode parameter is set to a value of **Master**.

Synchronizing Songs

The K2600 has an internal MIDI clock, which is always running at a speed set by the Tempo parameter (on the MAIN page). When you're in Song mode and the Clock parameter (on the MISC page) is set to a value of Internal (**Int**)—and the Sync parameter (also on the MISC page) is set to **Xmit** or **Both—** songs will sync to the K2600's internal clock. At this setting, the clock signal is sent to the K2600's MIDI Out port. This is standard MIDI Sync, and any device that accepts MIDI Sync will play in sync with the K2600.

If you set the Clock parameter to external (**Ext**), the K2600 expects to receive MIDI clock at its MIDI In port. Make sure that the Sync parameter on the MISC page in Song mode is set to **Recv** or **Both**, to enable the K2600 to receive sync messages as well as MIDI clock.

To play back a song, press **Play**, and the K2600 starts as soon as it receives both MIDI clock and a Song Start message. Or if you want to record, press **Record**, and the K2600 starts recording as soon as it receives both MIDI clock and a Song Start message.

If Sync has a value of **Off** or **Xmit**, the K2600 still uses the external clock, but you can't trigger recording or playback remotely; you have to use the K2600's soft buttons.

If you're using Song mode to capture a sequence that you've recorded on an external sequencer, you'll want to consider the Clock parameter's setting before you record via MIDI. If you have the Clock parameter set to **Ext**, the K2600 will follow the clock of your external sequencer. As a result, the notes in the song you create will fall regularly on the beats (unless your externally recorded sequence uses a time signature other than 4/4). This will make it easier for you to find the notes in the Song Editor. If you set the Clock parameter to **Int**, the notes in the song will not necessarily align with the beats of the measures in the song, but the song will play back exactly as you recorded it on the external sequencer—including tempo changes you may have incorporated into the externally recorded sequence. If the Clock parameter is set to **Ext**, tempo changes will not carry over to the K2600, and will not be heard when you play back the song.

Finally, keep in mind that when the Clock parameter is set to **Ext**, programs that use one or more of the Clock control sources (see Chapter 6 of the *Reference Guide*) will sync to the external MIDI signals. If no external clock signal is received, the Clock control sources are disabled.

Songs and Effects: A Brief Tutorial

On page 12-20 we discussed how to keep the effects constant during song playback and MIDI recording. For more complete control over the effects used in a song—including real-time control—we recommend dedicating one track of the song to effects control. Here's how to do it.

1. Go to Effects mode, and make sure that the value of FX Mode is **Auto**, and the value of FX Channel is **Current**.
2. Go to the MAIN page in Song mode, and call up a song.
3. Press **Edit**, which takes you to the COMMON page in the Song Editor.
4. Set the EffectChan parameter to the channel you want to use for effects control. Choose a channel that isn't being used for any of the existing tracks in the song. Channel 16 is the default click-track channel, so you probably don't want to use Channel 16. For this tutorial, we'll use Channel 15.
5. Save the song and exit the Song Editor. You're back on the MAIN page in Song mode.
6. Set the recording track (the RecTrk parameter) to match the effects channel that you set in Step 4—in this case, Channel 15. Note the **R** in the Track status indicator line.
7. Move the cursor to highlight the current program. This is where you select the program to be used for the current recording track. Since you aren't going to record any notes on this track, the program assignment doesn't matter. On the other hand, if you already have a program that uses a studio and FXMods that you like, use it here.
8. Press **Edit**, and since the cursor was highlighting an editable object (the program), you'll enter the Program Editor.
9. Go to the KDFX page, and choose a studio. Assign any FXMods that you want to use.

10. Save the program (we recommend renaming it as well), then press **Exit** to return to the MAIN page in Song mode.

So far, so good. Your song uses Channel 15 for effects control, and the effects for the song are determined by the studio used in the program on Channel 15. That is, every program in the song directs its output to this studio, according to the value of the Pair parameter on the OUTPUT page in the Program Editor.

You'll recall that a studio has four inputs, each of which can be responsible for a different effect (or no effect at all). You may want to edit each program in the song, to send its output to the desired studio input. There's a quicker way, however, that doesn't involve editing programs: you can determine program output (and consequently studio input) based on MIDI channels. For example, you can send Channel 1 to KDFX-A, in which case any program on Channel 1 will send its output to KDFX-A, regardless of the program's output settings.

1. Press the **MIDI** mode button to enter MIDI mode, and press the **CHANLS** soft button. This takes you to the CHANNELS page.

2. Press the **Chan/Bank Up** or **Down** button to display the settings for one of the channels you've used in your song. (the top line of the display shows you the current channel).

3. Change the value of the OutPair parameter. Any value other than **Prog** means that the output settings are determined by the *channel*, not by the program assigned to that channel.

4. Change the value of the OutPair for the other channels that you've used in the song.

Now you have all the outputs directed to the right studio inputs. Keep in mind, though, that the output settings are customized for the current song. You'll need to repeat this process for each song—unless you set OutPair back to **Prog** for every MIDI channel.

Next you'll need to make sure that the song always calls up the right studio.

1. Go to the MAIN page in Song mode, and make sure that the recording track is still the one that uses Channel 15. Also make sure that the program on that track is the one containing the studio you want to use for the song.

2. Record a couple of bars. This automatically puts a Program Change command at the beginning of the track. Each time you play the song, the K2600 selects that program—and consequently the studio and FXMods associated with that program.

That's it. If you want the song's effects to change during playback, use this track to record movements of the controllers that are designated as FXMods.

With one small difference, this approach to effects control also works if you're using an external sequencer. When you're recording in Song mode, the EffectChan parameter (on the COMMON page in the Song Editor) determines the channel used for effects control. When you're using an external sequencer, it works a bit differently.

1. Press the **Effects** mode button to enter Effects mode.

2. Set the value of the FX Mode parameter to **Program**.

3. Set the value of FX Channel to whatever channel you want to use for effects control.

4. Create a program that uses the studio and FXMods you want, and use that program on the channel you chose as the FX Channel.

Memory Limits

While there’s no actual time limit to the length of the songs you record, their size is limited to 64K (or to the maximum amount of available free RAM space you have, if it’s less than 64K). However, you can create longer songs by recording each section as a separate song, then putting it together with the Arrange feature. If you run out of RAM space while recording a song, the recorder stops and prompts you to save the song. It’s a good idea to check your free RAM space before you begin recording a song, and to check the “Used” field as you record. If you’ve used all the available RAM for recording, you may find that when you go to the Song Editor to delete a song or edit its tempo, the K2600 tells you that there’s not enough memory to edit. In this case you won’t be able to edit any object greater than 4K in size. Objects smaller than 4K can still be edited, because the K2600 always reserves a minimum of 4K of RAM.

If you want to delete a song and the K2600 won’t let you enter the Song Editor, select the default song (**1 NewSong**). Since it’s smaller than 4K (as long as you haven’t saved any changes to it), you’ll be able to enter the Song Editor. Press the **Delete** soft button, then use the Alpha Wheel to select the program you want to delete. Press the **Delete** button again, and the song will be erased, freeing up enough RAM to edit other songs. (You could also delete the song in Master mode—by pressing Object, then Delete, to get to the file deletion dialog.)

Loading MIDI Files From Disk

If you have a Type 0 or Type 1 MIDI sequence file stored on an MS-DOS disk (720K or 1.4 M) or a SCSI device, you can load it into one of the RAM banks, and the K2600 will be able to play it from Song mode.

Recording Multi-timbral Sequences via MIDI

You can record sequences from an external MIDI device using Song mode. Program numbers and MIDI channel assignments of multi-timbral sequences are recorded with the notes. To record via MIDI, connect the MIDI Out port of your sequencer to the K2600’s MIDI In port. Select Song mode, and set the Clock parameter to External. This will sync the K2600 with the MIDI clock of the external sequencer.

You will probably want to set the Local Keyboard Channel parameter to **None** when recording from an external sequencer, since the rechannelizing effect of that parameter could have unintended results.

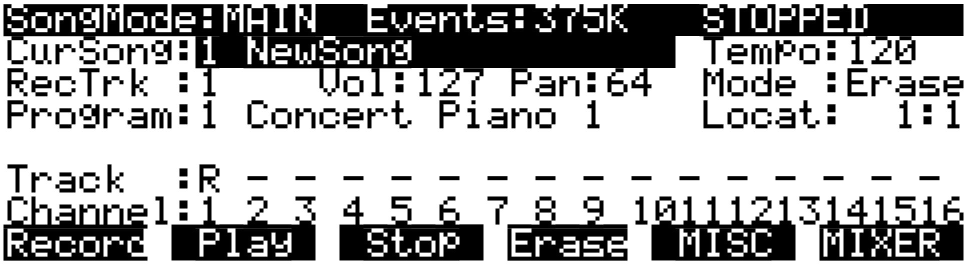
To record all your tracks in one pass, set the RecTrk parameter to **Mult** and make sure that for each channel of information on your source sequence, you have a track enabled to record and a unique channel assigned to that track. (The default setting of all tracks enabled to record on channels 1–16 will always work.)

You can also record individual tracks from your source sequence by setting the RecTrk parameter to a specific track. The K2600 will record only information coming in on the channel that the RecTrk parameter is set to.

Press the **Record** button, and the K2600 will wait for the first clock start from the sequencer. Make sure that your sequencer is set to send MIDI clock signals, and start the sequence. The K2600 will begin recording when it receives the first MIDI clock start from the sequencer. When the sequencer has finished its playback, press the **Stop** soft button, and the K2600 will stop recording and ask if you want to save the song.

Song Mode: The MAIN Page

The Song-mode MAIN Page allows real time recording and playback, song and track selection. From this page you can view and edit the tracks' channel, program, volume and pan settings, as well as other useful items.

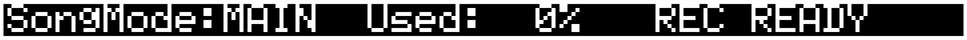


```
SongMode:MAIN  Events:375K   STOPPED
CurSong:1 NewSong            Tempo:120
RecTrk :1    Vol:127 Pan:64  Mode :Erase
Program:1 Concert Piano 1    Locat:   1:1

Track  :R - - - - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record  Play   Stop   Erase  MISC   MIXER
```

Parameter	Range of Values	Default
Current Song (CurSong)	Song ID & Name	1 NewSong
Recording Track (RecTrk)	1 – 16, None, Mult	1
Program	Program ID & Name	Current Program
Setup	Setup ID & Name	Current Setup
Track Status	- (Empty), R, M, P	- (Empty)
Channel	1 – 16	1 – 16
Volume	0 – 127	127
Pan	0 – 127	64
Tempo	1 – 255 BPM	120 BPM
Mode	Merge, Erase	Merge
Location (Locat)	1:1 – 9999:9	1:1

The Events field on the top line displays the number of events that you can store in RAM. 375 K in the above example represents maximum available memory with P/RAM installed. The maximum is 121 K without P/RAM. Note that this figure shows the number of free *events*, each of which takes up about four bytes of RAM. That's why the number you see here is typically about 25% of the free RAM (in *kilobytes*) you see in the Samples field in the top line of the Disk- and Master-mode pages.

When the Song Status is REC READY or RECORDING, the Events field changes to Used, and indicates the percentage of the recording buffer that you have filled, instead of the free event space.



```
SongMode:MAIN  Used:   0%   REC READY
```

Song Status, also on the top line of the display, is always one of the following:

STOPPED The default sequencer status; also appears when you press the **Stop** or **Pause** button.

PLAYING Appears when the **Play** button is pressed, but only if the following conditions are true: the **Record** was not pressed prior to pressing Play, the Key Wait parameter is set to **Off**, and the Clock parameter is set to **Int** (or Clock is **Ext** and MIDI clock is detected).

REC READY Appears when the **Record** button is pressed while Song Status is STOPPED. REC READY flashes, indicating that the sequencer is waiting to start recording.

RECORDING Appears when the **Play** button is pressed while REC READY is flashing (unless Key Wait is **On** or the Clock is set to **Ext**). RECORDING also appears if the **Record** button is pressed while Song Status is PLAYING.

KEY WAIT Appears when the **Play** button is pressed, while Song Status is STOPPED or REC READY, if the KeyWait parameter on the MISC page is set to **On**.
KEY WAIT flashes, indicating that recording or playing will begin when you strike a key.

EXT. CLOCK Appears when the **Play** button is pressed, while Song Status is STOPPED or REC READY, if the Clock parameter on the MISC page is set to **Ext**.
EXT. CLOCK flashes to show that the K2600 is waiting for an external MIDI clock message to start recording or playing.

Current Song (CurSong)

This shows the ID and 16-character name of the song currently selected for recording, playback, or editing. When a song is selected, Program Change, Volume, and Pan information is sent to all MIDI channels assigned to tracks that have data on them, and the internal clock is set to match the setting of the Tempo parameter.

Tempo

Controls tempo for the selected song. You can make temporary changes, record real time tempo changes, or set an initial tempo for the current song.

Whatever the tempo is set to when you record your first track will be the song’s initial tempo. Temporary changes may be made during playback, but the tempo will reset to the initial tempo when the sequencer is STOPPED.

To change a song’s initial tempo, press **Record** (the Song Status will change to REC READY), set the tempo desired, then press **Stop**. The initial tempo can also be changed with the Tempo parameter on the COMMON page in the Song Editor. The song will always start playback at the initial tempo, even though this tempo marker does not get recorded as a tempo event on any track.

If the sequencer is RECORDING, any tempo value changes will be recorded in real time. Unlike the special case of setting the initial tempo, any tempo changes recorded in real time are recorded as tempo events.

Fractional Tempos

You can use fractional tempos (120.5, etc.) in your sequence. However, the initial tempo can not be fractional, and you cannot enter a fractional number in the tempo parameter on the MAIN or COMMON pages. You must first record a real time tempo event, then go to the Event Editor and change it to a fractional amount.

To do this, press **Record**, then **Play**. The sequencer starts recording. Use any data entry method to choose a tempo. The value is unimportant since you will be changing it in the EVENT Editor. Next press **Stop** and save the song. Now when you go to the EVENT Editor (see page 12-44), you will see a tempo event. You can now edit the value to a fractional amount. To have the song start immediately with the fractional tempo, edit its location to 1:1:000.

Recording Track (RecTrk)

Determines which track is record enabled. Set the record enabled track to **Multi** to record more than one channel simultaneously or to use a setup in your song.

When RecTrk is set to a single track (**1–16**), Record (R) is displayed for that track in the Track Status Indicator region (above the Track and Channels region). Conversely, with one exception, when any track's Status Indicator is changed to Record (R), that track is shown as the value for the RecTrk parameter.

The exception is when RecTrk is already set to **Mult**, you can select the record enabled tracks by toggling the Track Status Indicator to Record (R), and the RecTrk will remain set to **Mult**.

When **Mult** is initially selected, all of the empty tracks will be record enabled. Tracks containing data will remain set to play (P), but you can manually set them to record (R).

The parameter(s) below RecTrk change according to the value of RecTrk and in one case, the mode from which you enter Song mode. If RecTrk is set to a single track (**1-16**), Program is displayed and you can select the program to be assigned to that track.

If you change RecTrk to **None**, the display changes to show the Channel parameter followed by the Program parameter (although the Program parameter's *name* doesn't appear, just its *value*). If you switch through the channels, the program also changes, showing the program currently assigned to that channel.

A setting of **Mult** makes the parameters below RecTrk dependent on the mode from which you entered Song mode. If you enter Song mode from Program mode, the Channel and Program parameters appear below RecTrk. If you enter Song mode from Setup mode, only one parameter, Setup, appears.

Program

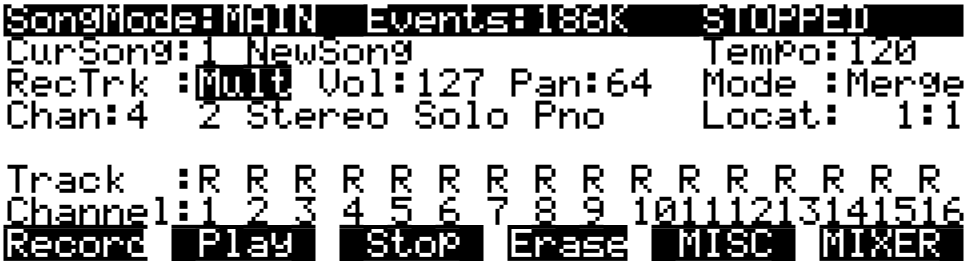
Scroll through the programs in memory to select the program before initially recording each track of your song. Any MIDI program changes on the current RecTrk or Chan cause the ID and name of the track's program to change during playback.

This parameter's name is not visible when RecTrk is set to **None** or **Mult** (to make room for the Chan parameter); just its value appears.

You'll see the Setup parameter instead of the Program parameter when you've entered Song mode from Setup mode. The Setup parameter functions similarly to Program.

Programs selected in Program mode or from a Quick Access bank are selected as the program on the current RecTrk when you return to Song mode.

To change a track’s program quickly, press **Record**, select the program, then press **Stop**. Or you could press **MIXER** to go to the MIX page, change the program as desired, then press **Keep**. This preserves all changes you have made to any other tracks: volume, pan, tempo, etc.



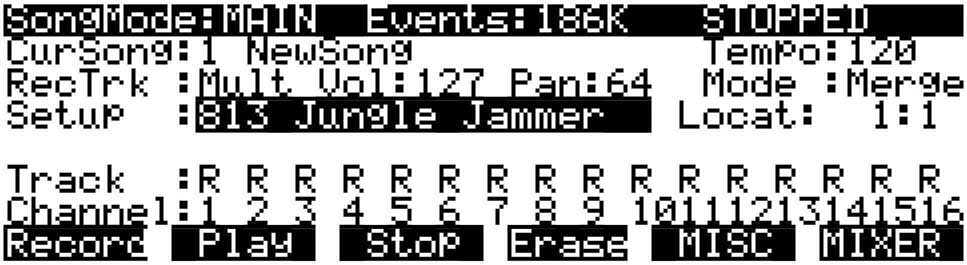
```
SongMode:MAIN   Events:186K    STOPPED
CurSong:1 NewSong              Tempo:120
RecTrk :Mult Vol:127 Pan:64   Mode :Merge
Chan:4  2 Stereo Solo Pno     Locat:   1:1

Track  :R R R R R R R R R R R R R R R R
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record  Play   Stop   Erase  MISC   MIXER
```

Channel (Chan)

This parameter determines the control channel and is available only when RecTrk is set to **None** or **Mult**. When RecTrk is Mult, this parameter appears only if you have entered Song mode from Program mode. In this case, the Channel parameter gets squeezed onto the same line as the Program parameter, which is why you don’t see the Program parameter’s name, just its value.

Setup

Displays the ID and name of the setup to be recorded. This parameter is available when RecTrk is **Mult**, and you enter Song mode from Setup mode. The display diagram below shows an example of the Song-mode page with the Setup parameter replacing the Channel and Program parameters.



```
SongMode:MAIN   Events:186K    STOPPED
CurSong:1 NewSong              Tempo:120
RecTrk :Mult Vol:127 Pan:64   Mode :Merge
Setup  :813 Jungle Jammer    Locat:   1:1

Track  :R R R R R R R R R R R R R R R R
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record  Play   Stop   Erase  MISC   MIXER
```

Using setups in Song mode takes a bit of extra planning. Since each zone in a setup uses a separate MIDI channel, you need to make sure that each setup zone has a corresponding track and channel allocation. It’s important to know how many channels, and consequently how many tracks, are needed for recording a particular setup. Each setup can have up to eight zones that can respond to your playing differently, depending on what range of the keyboard is being played, or if certain velocity and/or controller values determines when a particular zone will respond. Be aware of the behavior of each setup you intend to record so that you can allocate the proper tracks and channels needed in your song.

Volume (Vol)

You can set an initial volume level for the playback and recording of each track as a value between **0** and **127**. If the channel of the RecTrk (or the control channel, if RecTrk is set to **Multi** or **None**) contains any recorded volume change (controller code 7), the change will be reflected as the Vol parameter's value in real time.

To change a track's initial volume quickly, press **Record**, change the value of Vol, then press **Stop**.

Pan

You can set an initial pan position (the balance between the Left and Right audio channels) for the playback and recording of each track as a value between **0** and **127**. A value of **64** is center. If the channel of the RecTrk or the control channel contains any panning data (controller code 10), the Pan parameter's initial value for the current track is modified in real time.

To change a track's initial pan position quickly, press **Record**, change the value of Pan, then press **Stop**.

Mode

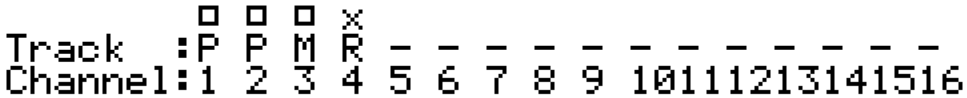
If Mode is set to **Merge** you will be able to overdub when recording on a track containing previously recorded data. You'll usually want to set Mode to **Merge** when RecMode (on the MISC page) is set to **Loop**. Otherwise, each time through the loop, the previously recorded information will be erased.

If you set Mode to **Erase**, the previously recorded data on the record enabled track will be replaced with the new data only during the Bars and Beats you are actually recording, and the previously recorded data before and after the newly recorded Bars and Beats will be preserved.

Location (Locat)

The Bar and Beat displayed as the Locate value changes relative to current location of the song during playback and recording. You can set this to a negative Bar and Beat location to start playback a set length of time before the beginning of the song.

Whenever you set the Locate point, that location will be used as the return point when **Stop** is pressed. Simply press **Stop** again to reset the song to the top (**1 : 1**).



```
          □ □ □ x
Track    :P P M R - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
```

Mode Indicators (+ and x):

Mode Indicators appear only for tracks that already contain data.

A plus sign (+) appears above the Track Status Indicator of a track set to record (R) when the Mode parameter is set to **Merge**.

An (x) appears above the Track Status Indicator of a track set to Record (R) when the Mode parameter is set to **Erase**.

Activity Indicators (□)

A small square (□) above the Track Status Indicator of a track set to **Play** (P) or **Mute** (M) means the track contains data.

During playback and recording, the indicators above tracks containing any MIDI data will flash a small, filled-in square when any MIDI activity is detected. The filled-in square also flashes over a track any time that there is incoming MIDI data on that track's channel, even while the sequencer is STOPPED.

Track Status Indicators

Using the **Up**, **Down**, **Left**, and **Right** cursor buttons to position the cursor onto a Track Status Indicator, you can toggle an empty track (-) into Record (R) with the Alpha Wheel or **Plus/ Minus** buttons.

Once a track contains data, it will have a (P) as a Track Status Indicator, and it will be played during playback. You now will be able to toggle between Play (P), Mute (M), and Record (R).

The track selected as the RecTrk will display an (R), designating it as the recording track. If the RecTrk is set to **Mult**, initially all empty tracks will have Record (R) as a Track Status Indicator, any of which can be switched back to empty (-) if at any time recording on specific tracks is not desired.

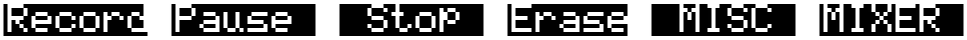
If there isn't a track with an (R), the RecTrk parameter's value will be **None**. (The exception is when the RecTrk is set to **Mult** and you have switched all of the tracks out of record enable.)

Track Channels

Each track has a MIDI Channel that it uses to receive and transmit data. By default, tracks 1–16 of a new song are assigned to Channels 1–16 respectively, although a track can play or record on any channel and the same channel can be used for more than one track. Keep in mind, however, that only one program can be assigned to a channel at a time, so if you have more than one track assigned to the same channel, they'll play the same program—the one on the higher-numbered track, since that's the most recent Program Change command received on that channel.

Soft Buttons on the MAIN Page

These buttons are similar to the transport controls on a tape deck. Some of those decks require you to press Play and Record simultaneously to begin recording. The K2600's transport buttons aren't like that, however. It's important that you press only one of these soft buttons at a time to insure proper recording start points, and to always be sure of the current sequencer status.



Record changes the Song Status to REC READY if the current Song Status is STOPPED. If the current Song Status is PLAYING, it will be switched to RECORDING when you press **Record**.

If the RecTrk is set to None, pressing **Record** will select the first available empty track for recording, thus setting the RecTrk to the newly record-enabled track number and placing an (R) in that track's status display. Song Status will change to REC READY or RECORDING, depending on the previous Song Status as described above.

Play plays back any recorded data when pressed while the song status is STOPPED. Playback will begin from the bar and beat specified in the Locate parameter.

When the Song Status is REC READY, pressing the **Play** soft button will begin recording.

Pause and **Play** share the same soft button. **Pause** appears only when the Song Status is PLAYING or RECORDING. Pressing **Pause** while the song is playing will stop the playback (soft button switches to **Play**), and the location remains at the current bar and beat, allowing you to continue from that location by pressing **Play** again.

Pressing **Pause** while recording will stop the recording process as if you had pressed **Stop**.

Stop halts the playback or recording, and resets the song's location to either the default Bar 1, Beat 1 value, or to whatever location you defined with the Locate parameter. If the location is defined as something other than Bar 1, Beat 1, press **Stop** twice to return to **1:1**.

Pressing **Stop** when the Song Status is RECORDING will always prompt the "Save changes to this song?" dialog (described below), and provides you with the opportunity to listen to the **new** song and compare it with the **old**, previously saved, song before answering **Yes** or **No**.

Here are some useful alternatives to using these button presses:

- Keyboard models have dedicated front panel buttons for **Record**, **Play**/**Pause**, and **Stop**. You'll find them just below the eight mode buttons.
- Footswitches can be assigned to **Play**/**Stop** and **Record**/**Stop**. On the MIDI-mode TRANSMIT page, assign either footswitch to controller number 102 or 103. Using controller 103, you can even record from within the Song Editor.
- Double button presses allow Song **Play**, **Pause**, and **Stop** soft button functions from most places where these buttons are not available. Use the cursor **Left**+**Right** double-press for **Play**/**Stop**, and use the cursor **Up**+**Down** double-press for **Play**/**Pause**.

Erase removes all channelized data from the track on which the cursor is currently positioned (on either the Track or Channel parameters) or from the record-enabled track if the cursor is positioned elsewhere. As described on page 12-32, a dialog appears that allows you to verify your intentions before permanently erasing any data. Once you've confirmed the erasure, it takes place immediately, so be sure you really want to erase the track before pressing the **OK** soft button. You will not be able to revert to a version of the song that includes the erased tracks—unless you've saved the earlier version to another ID. So before you start removing data that you may not want to lose permanently, you should also check out *Delete* on page 12-53, and *Song Editor: Track Functions* on page 12-51.

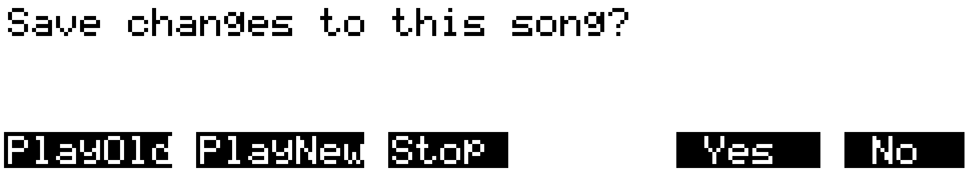
The Erase function doesn't show a dialog or erase any data if an empty track is selected.

MISC accesses more sequencer control parameters found on the Song-mode MISC page. These miscellaneous controls include record- and play-mode settings, auto punch-in points, quantization on input settings, tempo, and click attributes, as documented later in this chapter, starting on page 12-32.

MIXER lets you view a graphic representation of knobs and level faders for each track on the MIX page. You can modify program changes, volume and pan position on this page. Record status can also be selected here. The section on the Mixer begins on page 12-39.

Save this song? Dialog

The following dialog appears after you have recorded a track and pressed **Stop**, or if you have entered the Song Editor and made changes, then pressed Exit, or if you press **Save** in the Song Editor.



Save changes to this song?

PlayOld appears along with the **PlayNew** soft button in the “Save this song?” dialog after the recording process has been stopped. Pressing **PlayOld** will play the current song, minus the last, but not yet saved, recorded data. The exception is when there is no previously recorded data (that is, recording the first track of a NewSong), then you will only have a **Play** soft button to audition the recorded data just entered.

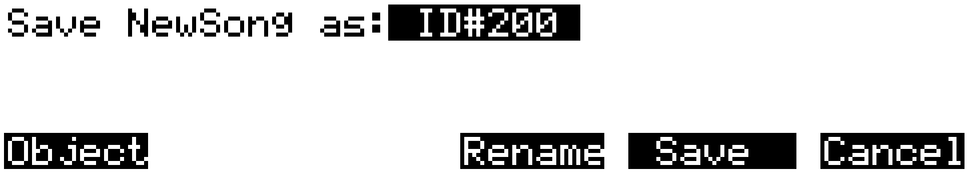
PlayNew/**Play** soft buttons allow you to play all of the recorded data, including data on the track(s) you have just recorded.

Stop halts the playback of either the Old or the New version of the song you are currently auditioning. Press the **Stop** soft button to stop the playback or recording, and reset the song’s location to either the default Bar 1, Beat 1 value, or to whatever location you defined in the Locat parameter.

Yes saves the data on the track(s) you just recorded. Whatever was played back when you pressed **PlayNew** will be the version of the song to be saved when you press Yes. The “Save New Song?” Dialog (shown below) will be displayed.

No returns you to the Song-mode page in which you were last recording, without saving any unsaved changes to the current song.

Save New song? Dialog



Save NewSong as: ID#200

Saving a song requires that you assign it an ID and this is where you do so. If you select an ID of an existing song object in RAM, the **Save** soft button will become **Replace** and **Replace existing Song** will appear on the display as a warning and a chance to change your mind.

There are only 20 IDs available for song objects per bank in banks 100 - 900. In the zeros bank, there are 75, IDs 1–75. In the 200s bank there are IDs 200–219, in the 300s bank, IDs 300–319, and so on.

Object accesses some useful database functions while still in Song mode, before you actually save the current song. Pressing **Object** jumps you directly to the Object Utility, described on page 11-13. When you press **Done** on the Utility page, you will be returned to the “Save New Song” dialog.

The Rename, Save, and Cancel soft buttons function the same as they do in all other editors.

Erase Track Dialog

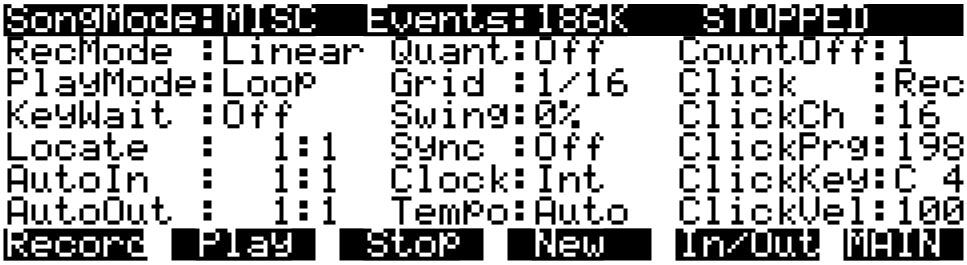
This dialog asks you if you are sure you want to erase a specific track. The track number will correspond to the track currently selected by the cursor position. When RecTrk is set to Multi or None, the Erase track function looks to the cursor position to determine the track to erase. If the cursor is positioned somewhere other than the Track or Channel parameters, Erase will have no effect. Erasing a track will not alter the song’s End point, nor will it remove any tempo events, since these elements are common to all tracks in the song.

Yes erases all data on the selected track and returns to the MAIN Song mode. The song will keep the erased track record enabled, but it will be an empty track.

No aborts the erasing of the selected track and returns to the MAIN Song mode with all previously recorded tracks intact.

Song Mode: The MISC Page

This page contains parameters that determine how the sequencer behaves during playback and recording, and when the K2600 is connected to another MIDI sequencer. Most of these settings are stored in the Master Object, (all except the Auto punch-in points) and none of these are saved with the song.



```
SongMode:MISC  Events:186K    STOPPED
RecMode :Linear Quant:Off   CountOff:1
PlayMode:Loop   Grid :1/16  Click   :Rec
KeyWait :Off    Swing:0%    ClickCh :16
Locate  :   1:1 Sync :Off   ClickPrg:198
AutoIn  :   1:1 Clock:Int   ClickKey:C 4
AutoOut :   1:1 Tempo:Auto  ClickVel:100
Record   Play    Stop    New    In/Out  MAIN
```

Parameter	Range of Values	Default
Record Mode	Linear, Fixlen, Loop, Unloop, Auto	Linear
Play Mode	Linear, Loop, List, Chain	Loop
Key Wait	On, Off	Off
Locate	-25:2 – 999:4	1:1
Auto In	1:1 – 999:4	1:1
Auto Out	1:1 – 2500:	1:1
Input Quantize	Off, 1% – 100%	Off
Grid	1/1 – 1/384	1/8
Swing	-99% – 125%	0%
Sync	Both, Xmit, Recv, Off	Off
Clock	Int, Ext	Int
Tempo	Auto, Fixed	Auto
Countoff	Off, 1, 2, 3	1
Click	Off, Rec, On, Cnt	Rec
Click Channel	1 – 16	16
Click Program	1 – 999	198
Click Key	C-1 – G9	C4
Click Velocity	1 – 127	100

The top line displays the amount of free event space and the current sequencer state, which is one of the following: STOPPED, PLAYING, REC READY, RECORDING, KEY WAIT, or EXT. CLOCK.

RecMode

When recording the first track of a new song, you will be able to record as if you had an endless length of “tape” no matter what the recording mode is set to.

When you record the first track, all recording modes operate the same way. This is because until you define the length of a new song, its End point is the default setting of Bar 8001, the maximum amount of Bars in a song. The End point is referenced and modified in different ways depending on the RecMode.

The End point of a song is defined when:

1. The **Stop** button is pressed to end the recording of the first track. The new End point is aligned to the nearest downbeat of the (empty) Bar immediately following the last Bar you were recording when **Stop** was pressed.

2. **Stop** is pressed while recording any track past the previously set End point in Linear or UnLoop recording mode. Again, the new End point is aligned to the downbeat of the (empty) Bar immediately following the last Bar you were recording when **Stop** was pressed.

3. The AutoOut Bar and Beat is set past current End point, after recording in Auto RecMode, and when the first track is recorded in Auto RecMode, the AutoOut Bar and Beat becomes the End point.

4. A new End point is entered on the EVENTS page.
5. Using the Track-edit functions Copy, Insert, and Delete to alter the song's length.

The End point of a song is used as a loop point in Loop and UnLoop modes, and it defines the Fixed Length of a song when you record in FixLen mode.

Linear: Record as if you had a nearly endless length of "tape."

FixLen: The song will not continue recording past the End point of the song when the RecMode is set to **Fixed Length**. Recording will automatically stop at the End point.

Loop: While RECORDING, the song will play the data between 1 : 1 and the End point over and over, allowing you to overdub in each pass if the Mode parameter on the MAIN Song-mode page is set to **Merge**. Make sure the Mode parameter is set to **Merge** if you intend to overdub in Loop Record, or else each consecutive pass in Loop RecMode will erase the data recorded on the previous pass. While you are still recording in Loop mode, you may selectively erase individual note events by pressing and holding the **Enter** button and depressing the desired notes on the keyboard during the times you would like them erased. Once the song is saved, you can not use this feature to erase individual notes.

UnLoop: When recording in UnLoop RecMode, any existing tracks will be played back as if they were looping from Bar 1 : Beat 1 to the End point, but they are actually being re-recorded linearly over absolute Bars and Beats until you press **Stop**. UnLoop allows you to record a linear track over a short looping section without first having to copy the section over and over again to achieve a new desired Song length. The End point of the Song is extended to the downbeat of the (empty) Bar immediately following the last Bar you were recording when **Stop** was pressed.

For example, let's say you've a recorded a four-bar drum loop and now want to record an eight-bar bass line. This would be a situation where UnLoop would come in handy. While the drum track keeps looping, the bass track will record in linear fashion, and the end point will be moved to the point at which you press **Stop**. Actually, the drum track will also change. It will play through its loop twice, but while the information is repeating in the loop, it will be recorded to the track. So now if you look at the drum track, you will see information in bars 5-8 (a duplicate of the information in bars 1-4).

Auto: Set RecMode to **Auto** to punch-in record, (either in merge or erase mode), on a track between the Bars and Beats Defined in AutoIn and AutoOut. To punch in and continue recording until you press **Stop,** set the AutoOut point before the AutoIn point.

PlayMode

The PlayMode setting, along with the End point, determine how the song or songs are played back. The song's PlayMode setting will be ignored when the song is included as a step in an arrangement; See *Song Editor: The ARRANGE Page* on page 12-63.

Linear: Set the PlayMode to **Linear** to hear the current song played only once, from the song position set in the Locate parameter to the End point. The song will be returned to the Locate Bar and Beat when it reaches the End point.

Loop: The Loop PlayMode will loop the current song from the End point back to Bar 1, Beat 1 continually during playback until **Stop** is pressed.

List: song objects in memory can be played back to back in numerical ID order starting from the current song followed by the song with the next highest ID. Once the current song reaches its End point, immediately the CurSong parameter is updated to the next highest song object ID and it will play from Bar 1, Beat 1 to its End point. When the song with the highest ID in memory has played through to its End point, the sequencer will stop and that song will be the new **CurSong**.

Chain: For every song, there is a parameter called ChainTo found on the COMMON page that determines what other song, if any, will immediately follow this song's playback when PlayMode is set to **Chain**. When the current song ends, the ChainTo song will replace the previous CurSong, and if its ChainTo parameter is set to any value other than **0 None**, then the Chain PlayMode will continue playback with the next song being chained. When the last song in the chain has played through to its End point, the sequencer will stop and that song will be the new CurSong.

The most common use for the Chain PlayMode is to construct a set list of different songs. Since the songs chained together are played immediately after one another, we recommend that you include a few measures of silence either at the end or the beginning of each song in the Chain.

When you want to string songs together more seamlessly, you can create an arrangement with the Arrangement Editor. See page 12-63 and page 12-12.

KeyWait

KeyWait specifies whether the sequencer will wait for a Note event before going into PLAYING or RECORDING status. With the KeyWait On, press the **Play** soft button while the sequencer is STOPPED or REC READY and the new status, KEY WAIT, will flash in the Song Status Field until a key is played. You can override the KEY WAIT status by pressing the **Play** soft button twice.

Locate

The Locate Bar and Beat will change in real time during play back and recording to reflect the song's current position. It can be set to a Bar and Beat before (negative values) or during a song. Once a song's length is defined, the End point is the maximum value for the Locate parameter. If Locate is not set to 1 : 1, the count off, if any, is disabled during play back or recording.

This parameter is identical to the Locate parameter on the MAIN page.

AutoIn

When you are in Auto record mode, AutoIn is the Bar and Beat when the recording will begin. If AutoIn is not set to 1 : 1, the count off, if any, is disabled. The AutoIn setting will not have an effect on recording unless the RecMode is set to **Auto**. When this parameter's value is highlighted, pressing the **In/Out** soft button updates only the AutoIn parameter; it doesn't update the AutoOut parameter.

AutoOut

When you are in Auto record mode, AutoOut is the Bar and Beat when the recording will stop. Set the AutoOut location earlier than the Bar and Beat defined as the AutoIn point in order to record to the very end of a song. The AutoOut setting will not have an effect on recording unless the RecMode is set to **Auto**.

Input Quantize (Quant)

This parameter determines how much Note events are moved towards grid locations upon the initial input of the events. If set to **Off**, no Quantizing will occur while you record, and the exact timing of your performance will be preserved during play back. If set to **100%**, every recorded Note event will be aligned to the closest grid location, defined by the Grid setting. Input

Quantize is used to quantize your performance as you record it in. However, you may wish to record without quantization and go back and quantize at a later point. To do this, use the Quantize function in the Track Editor, as described on page 12-54.

Grid

This setting determines the size of the Input Quantize grid expressed as a fraction of a Bar with a 4/4 meter. Set Grid to **1/1** for whole note grid, **1/16** for sixteenth notes. All of the standard note durations and every fractional Bar divisions in between (including triplets, for example, 1/12 = 1/8tr, 1/24 = 1/16tr, and so on) are available as the size of the Input Quantize grid. You can select commonly used Grid values by double-pressing the **Plus/Minus** buttons.

Swing

The Swing percentage is applied to the quantize grid. Zero percent swing is straight time, **100%** produces a swing (triplet) feel. A positive Swing value determines how close every other grid location is moved to a point 1/3 of the way towards the next grid point. Negative Swing moves every other grid location closer to a point 1/3 of the way towards the previous grid point.

Sync

The Sync parameter is used in conjunction with Clock. It controls transmission and reception of MIDI sync messages except actual clock. These are the messages it controls: Song Start, Song Stop, Song Continue, Song Select, and Song Position Pointer.

Song Select

Song Select is a command similar to a Program Change command; it enables you to select songs via MIDI. Like other MIDI messages, Song Select has 128 values. That's about half the number of songs you can store in the K2600, so if you have a lot of songs, you can't use Song Select to select all of them.

The K2600 maximizes the number of songs you *can* select by remapping Song Select values of 0–127 so they match the way songs are stored in the K2600's RAM banks (75 in the Zeros bank and 20 in each of the other nine banks). If you want to use Song Select extensively, you should number your songs according to the song IDs listed in the following table. Songs in the Zeros, 100s, and 900s banks are not available for selection with Song Select.

Song Select Value	Song ID
0–19	200–219
20–39	300–319
40–59	400–419
60–79	500–519
80–99	600–619
100–119	700–719
120–127	800–807

Song Position Pointer

The K2600 sends a Song Position Pointer message via MIDI when you start a song. The message indicates the location (Bar and Beat) at which the song starts playing—often that's the start of the song (1:1), but you can set the Locate parameter anywhere you want, and start the song from there when you press **Play**. The K2600 also responds to Song Position Pointer messages received from an external source.

When you have the K2600 synced to another sequencer, the Song Position Pointer message shifts the auxiliary sequencer's start point correspondingly. This is normally an extremely convenient feature, but there's one thing to avoid.

The K2600 lets you set a negative value for Locate, which gives you a countoff before the song starts (if you do this, set the Countoff parameter on the MISC page to **Off**, so you don't repeat the countoff). Keep in mind that the Song Position Pointer message doesn't support negative values, so your auxiliary sequencer might lose sync if you use a countoff. The safest approach is never to start a song with a Locate value less than **1:1**.

Clock

Specifies the source clock as being internal or external. When the Clock is set to external (**Ext**), the K2600 will wait to receive MIDI clock data, via its MIDI In port, from another device capable of generating MIDI clock data before playback and real-time recording can begin.

Tempo

The Tempo parameter, when set to **Fixed**, provides a tempo lock feature to override any real-time tempo changes recorded into a sequence. If set to Auto, tempo changes will be respected.

CountOff

Selects the number of bars of countdown, if any, before playback or recording starts. This works in conjunction with the click, so if the Click parameter is set to **Off**, the CountOff setting will have no effect. If the click is set to record only, then the CountOff will happen only when RECORDING.

Click

The Click parameter controls the click behavior. Set to **Off**, there is no click, and consequently no CountOff. When it is set to **On**, a click is present during playback and recording. To have a click only while RECORDING, set the Click to **Rec**. A value of **Cnt** generates a click only during RECORDING CountOff, if any.

ClickCh

Specifies which MIDI channel will be used for the metronome click.

ClickPrg

If click is in use, ClickPrg specifies which program will be used as the metronome click's sound. The click channel will be locked on to this program internally, and this program number will be transmitted via MIDI on the click channel to external any device(s) when playback or record is started. The default click program is **198 Click**, which uses the Clave keymap.

ClickKey

The ClickKey is the note to be used for the metronome click.

ClickVel

ClickVel determines the attack velocity to be used by the metronome click. The first beat of each measure gets played at exactly this velocity level while the other beats scaled to about 90% of this value as a way to provide an accent.

If there's not enough of a distinction between Beat 1 and the other beats, you can do the following to extend the dynamic range of the click program:

1. Go to Program mode, and select Program **198 Click**, then press **Edit**.
2. Press the **more>** soft button, then the **F4AMP** soft button.
3. Select the Velocity Tracking (VelTrk) parameter, and increase its value. The more you increase it, the more difference you hear between Beat 1 and the other beats. Don't go too high, though. This may cause Beats 2, 3, and 4 to drop too low. When you save, we recommend using a different program name and ID, or at least a different ID.
4. Go back to Song mode, select the new program's ID as the value for the ClickPrg parameter, and start recording.

Soft Buttons on the MISC Page

Record, **Play(Pause)**, and **Stop** work the same as they do on the MAIN page, described on page 12-24.

New selects **1 NewSong** as the current song and jumps back to the MAIN page. The tracks in the new song will be empty, but all initial program, volume and pan settings, and all parameters in the MISC page remain set the same way they were in the previous song.

In/Out provides a quick way to enter the Auto punch-in points in real time, based on your current location in the song. When you press **In/Out**, the K2600 updates either the AutoIn or AutoOut parameter (or both), depending on their current values and the value of the Locate parameter.

Typically, you'll start song playback, press **In/Out** when you reach the desired punch-in point, then press it again when the song reaches the desired punch-out point.

More specifically, the first time you press **In/Out**, the K2600 sets AutoOut to match the current value of Locate. Press **In/Out** again, and the K2600 shifts the current value of Auto Out to the value of Auto In, and updates AutoOut to match the current value of Locate.

If the value of AutoIn or AutoOut is highlighted when you press **In/Out**, the K2600 updates only the selected value.

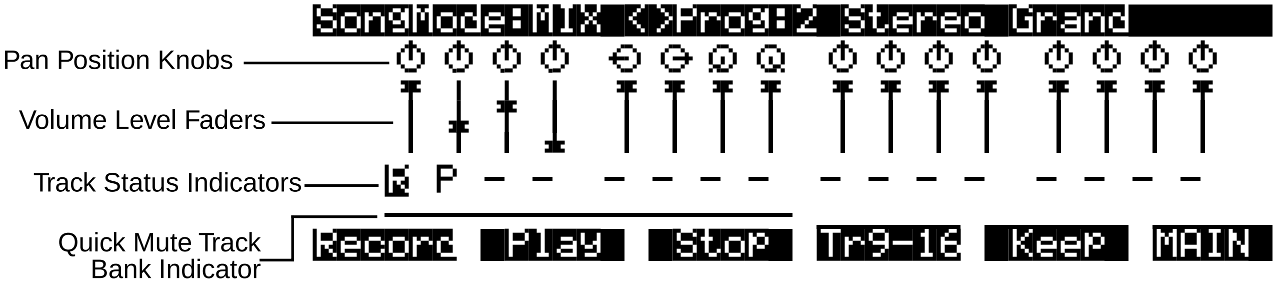
The AutoIn and AutoOut parameters display only Bars and Beats. However, the actual In and Out points will be precise to the Tick occurring at the time the **In/Out** button was pressed.

MAIN returns you to the MAIN page.

Song Mode: The MIX Page

The MIX page allows you to set and keep new initial settings for all sixteen tracks' program changes, volumes, and pan positions. You can set these three parameters for each track, then press the **Keep** button, prompting a dialog that asks if you really want to update these settings. Press **Yes** to make the changes. Another way to set these initial settings is to press **Record** to put the sequencer into REC READY status, make any desired changes, then press the **Stop** soft button.

You can also record real-time changes by changing the value of the highlighted fader or knob while recording on that track. If you do make recording changes in real time, though, make sure that RecMode is set to **Merge**, or else the data previously recorded on the track will be erased.


SongMode:MIX <>Prog:2 Stereo Grand
Pan Position Knobs
Volume Level Faders
Track Status Indicators
R P - - - - - - - - - - - - - -
Quick Mute Track Bank Indicator
Record
Play
Stop
Tr9-16
Keep
MAIN

The top line of this page displays the program number and name for the currently highlighted track. The different tracks are selected by moving the cursor with the **Left** and **Right** cursor buttons. The program for each track can be changed in the MIX page with the **Chan/Bank** buttons. Pressing the **Chan/Bank** buttons at the same time will jump to the next bank of 100 programs.

Pan Position

Position the cursor over any one of the sixteen Pan Position "knobs" on the display and turn the Alpha Wheel to change the panning for the selected track. The graphic display will move smoothly between the left and right settings and these changes can be recorded in real time. Use the alphanumeric buttonpad to enter in a value between **0** (hard left) and **127** (hard right) if you want to have a track jump immediately to a new pan position. The default pan position is **64** (straight up).

Volume Level

Position the cursor over any one of the sixteen Volume Level "faders" on the display and turn the Alpha Wheel to change the MIDI Volume for the selected track. The graphic display will move smoothly, setting volume changes that can be recorded in real time. Use the alphanumeric buttonpad to enter in a value between **0** (no volume) and **127** (maximum volume) if you want to have a track jump immediately to a new volume level. The default value is **127** (maximum volume).

On keyboard models, you can use the sliders to adjust the volume of the currently selected bank of eight tracks.

Track Status Indicators

Using the **Up**, **Down**, **Left**, and **Right** cursor buttons to position the cursor onto a Track Status Indicator, you can toggle an empty track (-) into Record (R) with the Alpha Wheel or **Plus** or **Minus** buttons.

Once a track contains data, it will have a (P) as a Track Status Indicator, and it will be played during playback. You now will be able to toggle between Play (P), Mute (M), and Record (R).

The track selected as the RecTrk will display an (R), designating it as the record-enabled track. If the RecTrk is set to **Mult**, then all of the empty tracks will have Record (R) as their Track Status Indicator. If RecMode is not set to **Mult** and there isn't a track with an (R), the RecTrk parameter's value will be **None**. If you are in Multi record mode, and have turned all record-enabled tracks (R) back to empty (-) so that there isn't a track with an (R), the RecTrk remains set to **Mult**.

Quick Mute Track Bank Indicator

This parameter appears as an underscore directly below the graphics for either the bank of Tracks 1–8 or Tracks 9–16. It indicates which bank of eight tracks will respond to the Quick Mute feature, described with the **Tr 1-8**/**Tr9-16** soft buttons, below. It also indicates which tracks respond to the keyboard model's sliders for adjusting track volume.

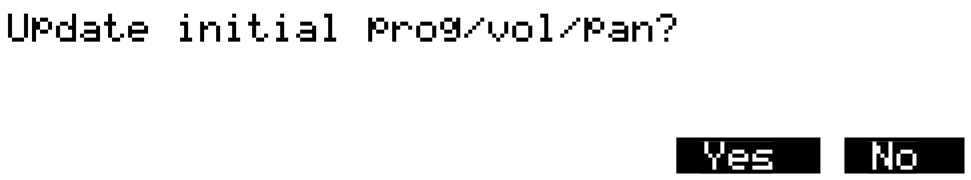
Soft Buttons on the MIX Page

Record, **Play(Pause)**, and **Stop** work the same as they do on the MAIN page, described earlier.

Tr 1-8 / (Tr 9-16): The eight mode select buttons to the left of the display are used as Track Mutes when on the MIX page. For example, press the **Setup** button to mute Track 2; notice that its track status indicator changes to **M**. Each of these buttons has an LED to indicate that the corresponding track is being muted, but since there are sixteen tracks and only eight buttons, this soft button will select, and display, the bank of eight tracks that can be muted in this fashion.

Press the **Tr 1-8** soft button to toggle the Quick Mute Track Bank Indicator under tracks 1 -8 or tracks 9 - 16, selecting which bank of eight tracks will respond to the Quick Mute feature.

Keep: If you have made any changes to the initial program, volume or panning of a track on the MIX page, press **Keep** to prompt this dialog:



Update initial Prog/vol/Pan?

Press **Yes** if you are sure you want to update the initial program change, MIDI volume, and pan settings for tracks already containing data in the current song, to the new values you just made on the MIX page. The settings will be modified and you are returned to the MIX page. Press **No** to abort the updating of the initial program change, MIDI volume and pan settings for the current song.

MAIN returns you to the MAIN page.

The Song Editor

In general, you'll get to the Song-editor pages by pressing the **Edit** button any time you're in Song mode. There's one exception: if the Program parameter is currently highlighted on the display, you'll enter the Program Editor when you press **Edit**.

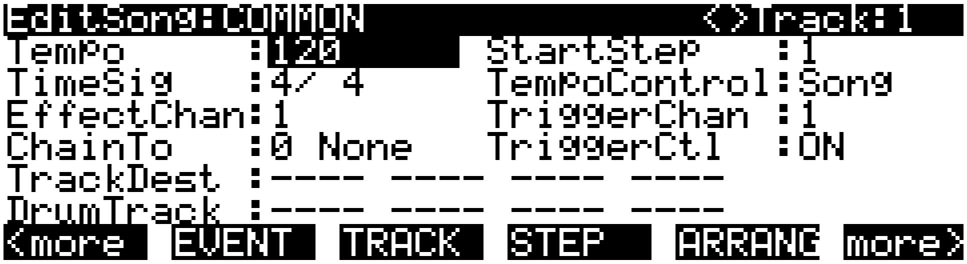
There are a few conventions shared by all of the Song-editor pages (except the ARRANGE page). Displayed at the top of each Song-editor page is the name of the page and the currently selected track (1–16, or all tracks). All of the values for the parameters found in any of the Song-editor pages are saved in the song object.

Soloing the Current Track

On all of the Song-editor pages, you can solo the current track by pressing either the **Setup** or **Quick Access** mode buttons (you can press the **Program** mode button to mute the current track). This in effect mutes all other tracks in the current song. Using the **Chan/Bank** buttons allows you to select the current track for editing and/or soloing. You can select the track to solo even if you are already in solo mode.

Song Editor: The COMMON Page

Press the **Edit** button on the front panel of the K2600 to display the COMMON page and begin editing a song. This is where you will find parameters common to all tracks, such as tempo and time signature, control parameters for effects and arrangements, and soft buttons for switching to other Song-editor pages.



```
EditSong:COMMON              <>Track:1
Tempo      :120       StartStep   :1
TimeSig    :4/ 4      TempoControl:Song
EffectChan:1          TriggerChan :1
ChainTo    :0 None    TriggerCtl  :ON
TrackDest :---- ---- ---- ----
DrumTrack :---- ---- ---- ----
<more  EVENT  TRACK  STEP  ARRANG  more>
```

Parameter	Range of Values	Default
Tempo	1 – 255 BPM	120
Time Signature	1 – 99 / 1, 2, 4, 8, 16	4/4
Effects Channel	1 – 16, None	1
Chain To	0 None, Song ID	0 None
Track Destination	-, L, M, X	-
Drum Track	-, D	-
Start Step	None, 1 – 255	1
Tempo Control	Song, Arrangement	Song
Trigger Channel	1 – 16, None	1
Trigger Control	Control Source list	On

The top line of this page displays the selected track or tracks. Select the available current track for editing by using the **Chan/Bank** buttons. Press both of the **Chan/Bank** buttons together to select all tracks.

Although the current track information is displayed on the top line, the COMMON page's parameters are global settings for the song and do not directly affect individual tracks.

Tempo

This is another place where the song's initial tempo can be set or modified.

TimeSig

Affects the click, playback looping, and locate function as well as some editing operations. Does not change the recorded data.

EffectChan

If the FX Mode and FX Chan parameters in Effects mode are set to **Auto** and **Current**, this parameter specifies which channel will control effects (studio and FXMods) while in Song mode and during playback of a song. In this situation, you'll hear the studio assigned to the program that's on the channel designated as the EffectChan will be heard.

Every song can have its own setting for EffectChan. This makes it convenient to change studios every time you change a song, but you'll need to pay more attention to which programs are controlling the studios—especially if you're creating arrangements with lots of steps. If you want to limit changes in studios as you change songs, you can do it easily. See *Effect Selection During Recording and Playback* on page 12-20 for more information.

ChainTo

The ChainTo setting is only used when the PlayMode parameter on the MISC page is set to **Chain**. Another song object is selected as the ChainTo value and will playback immediately after the current song has played for its entire length.

Parameters Used with the Arrangement Feature

The remaining parameters, as well as the Drum Track parameter, are used in conjunction with the Arrangement feature (via the **ARRANG** soft button). This feature allows you to create a song by arranging it in a series of sections, called steps. You can save the various sections of your sequence as separate songs, then assemble them using the Arrangement Editor. This method allows you to save memory, since you can repeat steps without having to duplicate the actual song data.

Another great feature within the Arrangement Editor gives you the ability to trigger steps by striking a specific note or group of notes on your controller. See page 12-63 for more information on arrangement features.

StartStep

The value for this parameter determines which step in this song's arrangement is played first. If set to **None**, the playback of the arrangement, using the transport buttons, will be disabled, though steps can still be triggered from Note events.

TempoControl

When the current song is composed of other songs entered as steps in the Arrangement Editor, the TempoControl parameter determines whether the current song's tempo setting or the arranged songs' tempo settings will be the tempo(s) used for the playback and recording of the arrangement.

Set TempoControl to **Song** if you want to use the current song's tempo as the master tempo. All of the steps' songs will playback at the same tempo, ignoring the tempos originally set for each song when you recorded or edited them.

Use the **Arrangement** setting for TempoControl if you would like the current song (containing arrangement data) to playback and record at the tempos originally defined in each arrangement step's song, ignoring its own tempo setting. The master tempo will change to the tempo of the current step's song.

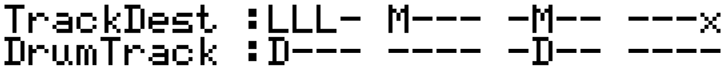
TriggerChan

This parameter controls the MIDI channel used to trigger arrangement steps via key presses as defined in the Arrangement Editor.

TriggerCtl

This parameter specifies the global control source used to enable the triggering of arrangement steps via key presses as defined in the Arrangement Editor. For example, suppose you set this parameter to a value of **MIDI 29**, which by default is the destination of Panel Switch 2 (Button 10). When Panel Switch 2 is on, striking any key in the trigger range (specified by the LoKey and HiKey parameters on the ARRANGE page). If you set this parameter to **ON**, striking a key in the trigger range always triggers a step.

TrackDest



```
TrackDest :LLL- M--- -M-- ---x
DrumTrack :D--- ---- -D-- ----
```

The MIDI data on each track has a destination assignment selectable with the TrackDest parameter. There are four possible indicators:

–: This is the default setting for all tracks in a new song. It means that MIDI data on the track are transmitted both locally to the K2600, and out the MIDI Out Port on the track's channel to any external devices.

L: The track's MIDI data will be transmitted locally only, to the K2600's internal sound generator. None of the track's MIDI data will be sent to the MIDI Out Port.

M: The track's MIDI data will be transmitted only to MIDI Out.

x: Disable the transmission of the track's MIDI data either locally or to MIDI. This is a quick way to mute a track more permanently without erasing it.

DrumTrack

Any of the song's tracks can be defined as Drum Tracks so that their Note events do not get transposed when a transposition is applied in the Arrangement Editor.

This feature is particularly useful when a drum kit program (or any other nonpitched program) is used in a song, assigned to a step of an arrangement, which is being triggered over a range of keys, and you want the sounds produced by each note number in that program to be preserved in each transposition. If in the song being used as a step in an arrangement (NOT the arrangement song itself), there is a (D) designating the track playing the drum program as a Drum Track, the originally recorded Note events on that track will remain unchanged.

The Drum Tracks' settings do not have any effect on edits made on the TRACK page in the Song Editor. Any tracks defined as Drum Tracks are transposed when a transposition is applied to these tracks from the TRACK page.

Soft Buttons on the COMMON Page

EVENT: Accesses an Event-list style editor. On the EVENT page in the Song Editor, you can scroll through, modify, add, or delete any or all of the tracks' MIDI events. The EVENT page is described on page 12-44.

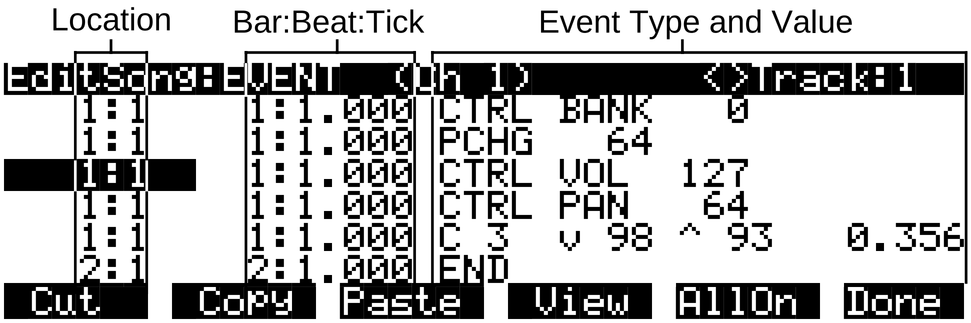
TRACK: Brings you to the TRACK page. This page accesses useful track based edit functions. There is a selectable edit function that can be applied to the selected track or all tracks in your song. The TRACK page is described on page 12-47.

STEP: Brings up the STEP page, from which nonreal-time note/rest entry is performed. The term STEP is used here to refer to step recording, and does not refer to the steps on the ARRANGE page. The STEP page is described on page 12-60.

ARRANG: Enters the ARRANGE page, gaining access to a group of arrangement specific parameters. The ARRANGE page is described on page 12-63.

Song Editor: The EVENT Page

Every type of recorded MIDI event is visible from this page. You can view and change these events if necessary.



```
EditSong:EVENT   (Ch 1)         <>Track:1
    1:1    1:1.000 CTRL BANK   0
    1:1    1:1.000 PCHG   64
    1:1    1:1.000 CTRL VOL  127
    1:1    1:1.000 CTRL PAN   64
    1:1    1:1.000 C 3  v 98 ^ 93   0.356
    2:1    2:1.000 END
 Cut    Copy   Paste   View   AllOn   Done
```

To scroll through the events, make sure the location (Bar and Beat, in the first column) is highlighted. Use the Alpha Wheel, the **Up** and **Down** cursor buttons, or the **Plus/Minus** buttons. As you scroll through the events, each event is executed by the sequencer. In the case of note events, you will hear the note played, although the duration will be short. If you have scrolled through a Sustain (MIDI Controller 64) message with an On value then you will hear the note sustain as if the sustain pedal was depressed. The note will continue to sustain until you scroll through a Sustain message with a value of Off.

You can also jump directly to a specific bar and beat by typing the bar number and beat number, then pressing **Enter**. Keep in mind if you have controller or program data previous to the point that you jump to, those events may not have been executed and you may hear unexpected results. For example, if you have program changes at bar 1 and bar 8, then if you jump from bar 1 to bar 9 any notes you scroll through will be played with the program change from bar 1.

The channel of the selected event is displayed on the top line of the page. When all of the tracks are being viewed in the Event list, the track of the selected event is displayed along with its channel. Use the **Chan/Bank** buttons to select an active track to view and edit the MIDI events recorded on it. You can view the data on all tracks by pressing both **Chan/Bank** buttons at the same time (or press **Chan/Bank Up** or **Down** until you scroll past the highest- or lowest-numbered track in the song). As you scroll through each event, the track and channel for the selected event appear in the top line of the display.

Location

The first column represents the Bar and Beat Locations of the different events in a song. Scroll through the events on the selected track(s) with the Alpha Wheel or enter in a specific Bar and Beat on the alphanumeric buttonpad to jump to events occurring on that Beat. A quick way to jump to the End point in a track is to press **9999** and then **Enter** on the alphanumeric buttonpad.

Bar, Beat, and Tick

Bar, Beat, and Tick are editable parameters for each event. They determine when an event happens relative to the other events within the song. To enter values with the alphanumeric buttonpad, first position the cursor in this column and then simply enter up to three digits if you only want to adjust the Tick value, four or five digits in order to change the location to a new Beat and Tick within the current Bar, or enter in the complete Bar, Beat, and Tick value to relocate the event to another Bar. No punctuation is necessary when entering any of these values.

Event Type and Value

The Event Type and Value region displays the MIDI event type (and related information) at each Event-list location in the song. Different event types display different kinds of information, and have different editable values. You can’t change the type of an event, but you can change the values associated with that event type.

For example, you can’t change a CTRL message to a PCHG message, but for any control message, you can change both its destination and value. A Note On is always a Note On—but you can change the values for its note number, attack velocity, release velocity, and duration. Incidentally, you won’t see the event type for Note Ons, just the editable values.

Table 12-2 lists the ranges of the editable event values.

General MIDI Events	The PCHG, BEND, MPRS, SYSX, TMPO, and END events will show their values in the display region directly to the right of the Event Type region. All of their values can be edited. Here are the ranges of values for each event type:	
	Program Change (PCHG)	0–127
	Pitch Bend (BEND)	-8192–8191
	Mono Pressure (MPRS)	0–127
	System Exclusive (SYSX)	The message in hexadecimal form
	Tempo (TMPO)	1.00–255.00 bpm
	End Point (END)	Any Bar: Beat: Tick setting
MIDI Note Events	Note events have four editable values: Note Number, Attack Velocity (indicated by a “v”), Release Velocity (indicated by a “^”), and Note Duration.	
	Note Number	C-1 – G9
	Attack Velocity	v1 - v127
	Release Velocity	^1 - ^127
	Note Duration	Beats: Ticks
MIDI Controller Events	Controller events have two editable values: Controller Type and Controller Value. Defined controllers are referred to by their names.	
	Controller Type	0 - 127
	Controller Value	0 - 127

Table 12-2 MIDI-event Value Ranges

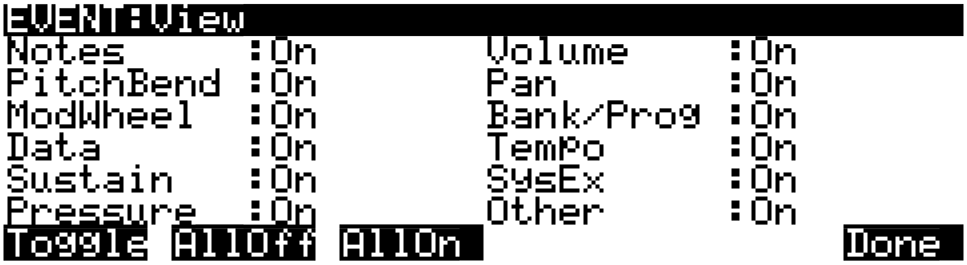
Soft Buttons on the EVENT Page

Cut: Removes the currently selected event from the Event list and temporarily stores it in a memory buffer so that you can immediately paste it into a new location.

Copy: Makes a duplicate of the currently selected event and temporarily stores it in a memory buffer so that you can immediately paste it into a new location.

Paste: Inserts the most recent cut or copied event into the Event list at the currently selected Bar: Beat: Tick location. The pasted event will share the same location with the event that already existed at that location in the Event list, but it will appear before the pre-existing event.

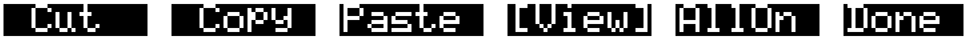
View: Brings you to the View page where you can set the view filter parameters for the Event list.



```
EVENT:View
Notes      :On        Volume     :On
PitchBend  :On        Pan        :On
ModWheel   :On        Bank/Prog  :On
Data       :On        Tempo      :On
Sustain    :On        SysEx      :On
Pressure   :On        Other      :On
Toggle  AllOff  AllOn                Done
```

The View page determines which events you see when you're on the EVENT page. If you set a parameter on the View page to **Off**, you won't see any of the corresponding events on the EVENT page, and the events won't be executed as they normally are when you scroll through the Event list. For example, if you turn Sustain off, then as you scroll through the list none of the notes normally affected by sustain will be sustained. However, if you exit the editor and play the song, all events will be executed regardless of the View filter settings.

When all of the parameters in the View page are set to **On**, the Event list on the EVENT page will display all of the recorded MIDI data on the selected track(s). If at least one of these view parameters is set to **Off**, the **View** soft button on the EVENT page will be in brackets (as shown below) indicating that you might not be seeing all of the recorded MIDI data on the selected track(s). If you press **AllOn**, the brackets disappear, and you'll see all event types.



Cut Copy Paste [View] AllOn Done

Press the **Toggle** soft button to switch all View filter parameters to the opposite of each one's currently set value.

AllOff: Available only on the View page. Sets all View filter parameters to Off. The only item viewable on the EVENT page will be an END point.

AllOn: Available on both the EVENT page and the View page. Sets all View filter parameters to On. This enables you to view all of the MIDI events that are on any or all tracks. If you're on the View page, the **Done** soft button returns you to the EVENT page. The view filter settings will be applied to the Event list.

Done: On the View page, returns to the EVENT page. On the EVENT page, returns to the COMMON page.

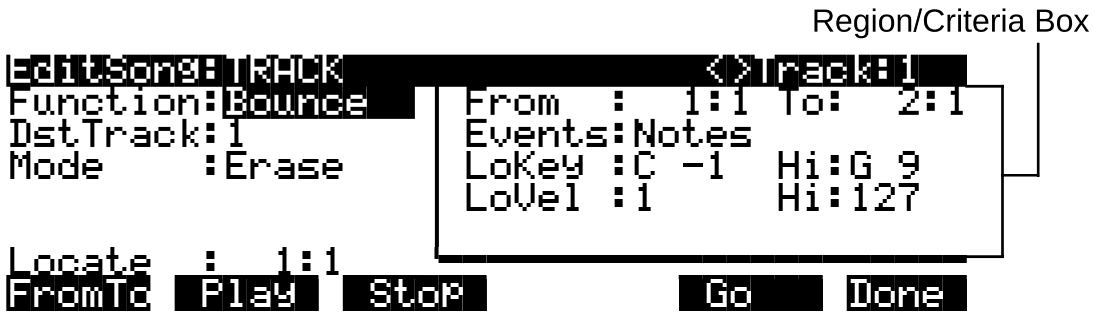
Song Editor: The TRACK Page

This page allows you access to useful track-based edit functions. These functions are:

Erase	Shift
Copy	Transpose
Bounce	Change
Insert	Thin
Delete	Remap
Quantize	Grab
Reference Quantize	

For each function, there is a set of parameters to control how the function operates, and on what region of the selected track(s). As usual, the top line of this page displays the selected track or tracks. Select the available current track(s) for editing by using the **Chan/Bank** buttons. Press both of the **Chan/Bank** buttons together to select All tracks.

Below is an example of the TRACK page for the Bounce function.



Region/Criteria Box

```
EditSong:TRACK                 <>Track:1
Function:Bounce    From   :   1:1  To:   2:1
DstTrack:1         Events:Notes
Mode    :Erase     LoKey :C -1   Hi:G 9
                   LoVel :1      Hi:127

Locate   :   1:1
FromTo  Play   Stop            Go     Done
```

You will notice that the page is divided into two halves, with the right half being a separate box This is called the Region/Criteria box. The parameters in this box are used to select the range of events (from a start Bar and Beat to and end Bar and Beat) for modification, as well as which types of events function will affect.

The parameters in this box will generally be the same for most functions. For some functions, however, some parameters may not apply. For example, Quantize and Transpose apply only to notes, while Remap applies only to Controllers. In addition to the Region/Criteria box parameters, the Locate parameter is also found on each function.

Since these parameters are common to most Track functions, we will define them first. Then we'll describe the individual functions along with the parameters specific to each, which are normally found on the left side of the page. Two functions, (Quantize and Reference Quantize) have a unique parameters in their Region/Criteria boxes. We'll describe those parameters along with the functions.

Once you've chosen a function and set the parameters to your liking, press **Go**. This executes the editing function. You can then play the sequence to hear the results of your edit. If you don't like your edit, simply exit the editor and press **No** when you are asked if you want to save. If you do like your edit, you can press **Done** and use the **more** buttons to get to **Save**, or just exit the editor and save the changes. Or, you can go to another edit function. Keep in mind though, that if you choose to perform more than one edit without saving, and you are not satisfied with one of the changes you make, you will have to exit the editor without saving and then redo each of the changes you made. That's why it's usually best to save after each successful edit.

Common Parameters for Edit Song: Track Functions

Locate

This parameter is available for every function on the TRACK page.

The Locate Bar and Beat will change in real time during play back and recording to reflect the song's current position. It can be set to any Bar and Beat, including negative values. Play back begins at, and Stop resets the song to the Locate Bar and Beat.

Region / Criteria Box Parameters

From and To

From and To are available in most TRACK edit functions to define a range of time on the selected track(s).

The From value defines the first Bar and Beat in a range of time selected for editing. Although From is always expressed in Bars and Beats, and will be aligned exactly on the Beat when adjusted manually, it is possible for From to represent a value with finer resolution when a range of time is defined in real time using the **FromTo** soft button.

The To value defines the Final Bar and Beat in a range of time selected for editing. Although To is always express in Bars and Beats, and will be aligned exactly on the Beat when adjust manually, it is possible for the To value to represent a value with finer resolution when a range of time is defined in real time using the **FromTo** soft button.

Events

Any and all types of MIDI events are available for editing, selectable in this parameter. Some events will provide you with settings for a range of values, or other MIDI event specific criteria. Available Values are: All, Notes, Controllers, MonoPress, PitchBend, ProgChange, PolyPress, SysEx, and Tempo.

When Events is Set to All

All MIDI events on the track(s) you are editing, that occur in the region of time between the From and To settings, will be affected by the edit function.

When Events is Set to Notes

Note Number and Key Velocity ranges can be set for Note events.

LoKey

Determines the lowest note in a range of notes to be affected. This can be set to any MIDI note value; the default is C-1.

High Key (Hi)

Determines the highest note in a range of notes to be affected. This can be set to any MIDI note value; the default is G9.

LoVel

An attack velocity range can be specified as criteria for selecting Note events for editing. The LoVel parameter sets the lowest velocity a Note needs to have in order to be edited. Notes on the selected track(s) with a attack velocities lower than the LoVel will not be affected by the edit. The available values are 1–127; the default is 1.

High Velocity (Hi)

The Hi parameter sets the highest attack velocity a Note needs to have in order to be edited. Notes on the selected track(s) with attack velocities higher than the value of Hi are not affected by the edit. The available values are **1–127**; the default is **127**.

When Events is Set to Controller

If your song contains any MIDI Controller data, the Ctl parameter is used to select particular controller data to be edited. Refer to the Control Source list in Chapter 6 of the *Reference Guide.*

LoVal

You may further specify a particular range of values to edit by setting a high and low value. LoVal will define the lowest modifiable value in the selected controller's recorded data. Value ranges are not definable when Ctl is set to All. Available Values are **0–127**.

High Value (Hi)

Hi defines the highest modifiable value in the selected controller's recorded data. Value ranges are not definable when Ctl is set to All. Available Values are **0–127**.

Soft Buttons on the TRACK Page

FromTo is a quick way to define the region of time you intend to edit. There are a couple of ways to use this feature when the sequence is playing back in real time, and both ways will set the temporal boundaries of the region to a finer resolution than Bars and Beats.

One way is to first position the cursor over the From parameter in the Region/Criteria box and then press the **Play** soft button. During playback, every time you press **FromTo**, the K2600 updates the value of From to match the current playback location. Position the cursor over the To parameter to change the value of To in a similar fashion.

If you haven't selected either the From or To parameter, pressing **FromTo** during playback updates From or To—or both—depending on the current playback location (the value of the Locate parameter) at the time you press **FromTo**. If you press it while the Locate value is earlier in the song than the current To value, the K2600 updates the From value. If you press FromTo again (without stopping playback) while the Locate value is later than the current From value, the K2600 updates the To value.

Play will start the playback of the song from the Bar and Beat set in the Locate parameter. When the song is playing, this soft button becomes **Pause**.

Stop stops the playback of the song and return to the Bar and Beat set as the Locate value.

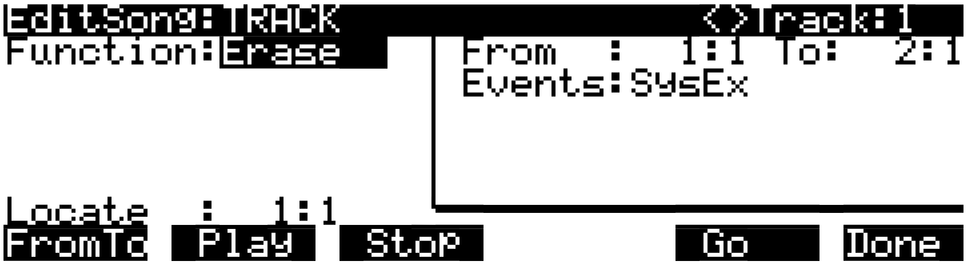
Go performs any of the Track-based edit functions described above.

Done will return you to the EditSong : COMMON page.

Song Editor: Track Functions

Erase

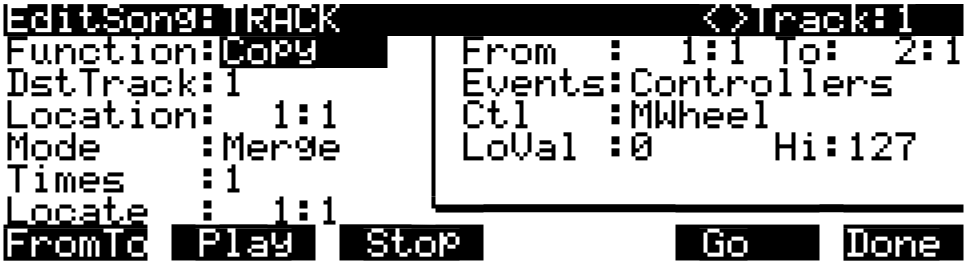
This function erases specified events from a region of time, but it doesn't delete the region of time. The result is like erasing a section of recording tape. If you want to completely remove a segment and shorten the length of the track, you can do it with the Delete function.



```
EditSong:TRACK                  <>Track:1
Function:Erase      From  :  1:1 To:  2:1
                    Events:SysEx



Locate  :  1:1
```

Copy

Use the Copy function to duplicate the selected events from the current track and place them in the same track or on another track, either merging with or overwriting existing data.



```
EditSong:TRACK                  <>Track:1
Function:Copy       From  :  1:1 To:  2:1
DstTrack:1          Events:Controllers
Location:  1:1      Ctl   :MWheel
Mode    :Merge      LoVal :0     Hi:127
Times   :1
Locate  :  1:1
```

If you do not want to copy all of the MIDI events in the defined range of time on the current track, use the Events parameter in the Region/Criteria box to select a specific MIDI event type you would like the edit function to affect. Some event types provide you more criteria selection parameters. It is often a good idea to set Events to **Notes** when copying, and then add any necessary controller or other data to the track at a later time.

DstTrack: 1 – 16 / All

Select a destination track for the copied events with the DstTrack parameter. All selected events described in the Region / Criteria box will be placed in the destination track(s) at any Bar and Beat you specify.

If the currently selected track is All tracks then the destination track will be All tracks as well.

No matter what channel the current track (source track) is set to when you use the copy function, the events will be played on the destination track's channel.

Location: 1 : 1

Specify a Bar and Beat location in the destination track where the copied data will be placed with the Location parameter. If the length of the copied region extends from the Location point beyond the song's existing End point, a new End point is defined.

Mode: Merge / Erase

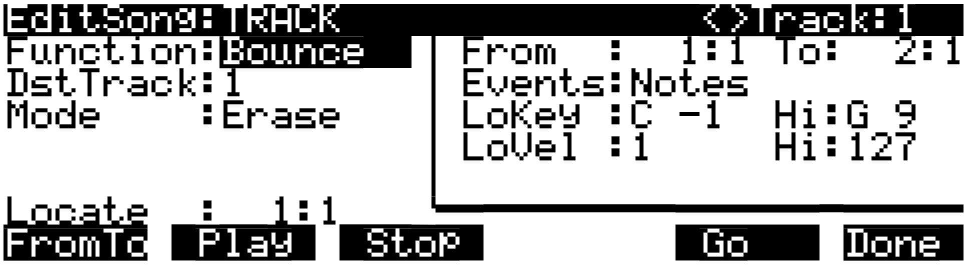
The Mode setting determines whether the copied events merge with, or erase existing events on the destination track from the location point to the end of the copied region.

Times: 1 – 127

The value selected for the Times parameter determines how many copies of the selected region are placed, one after another, in the destination track.

Bounce

Use the Bounce function to move the selected events from the current track to another track, either merging with or overwriting existing data on the destination track. The Bounce function differs from the Copy function in that the original data is not preserved in the original track. As on a multi-track tape recorder, Bounce will always put the data in the same timeline on the new track that it was on the old track.



```
EditSong:TRACK                  <>Track:1
Function:Bounce     From  :  1:1 To:  2:1
DstTrack:1          Events:Notes
Mode    :Erase      LoKey :C -1  Hi:G 9
                    LoVel :1     Hi:127

Locate  :   1:1
FromTo  Play  Stop        Go      Done
```

DstTrack: 1 - 16

Select a destination track for the events to be moved to with the DstTrack parameter. All selected events described in the Region / Criteria box will be placed in the destination track at the data’s original location.

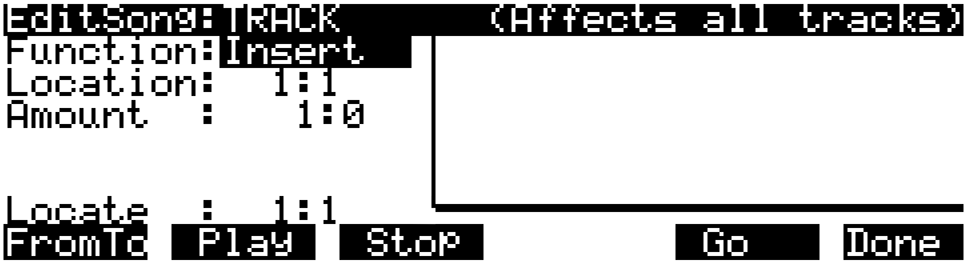
No matter what channel the current track (source track) is set to when you use the bounce function, the events will be played on the destination track’s channel.

Mode: Merge/ Erase

The Mode setting determines whether the bounced events merge with, or erase existing events on the destination track from the location point to the end of the copied region.

Insert

The Insert function is used to add blank time to the current song, modifying the song's End point appropriately. The Insert function will affect all tracks. This is similar to splicing a piece of blank tape to an existing segment of recording tape.



```
EditSong:TRACK        (Affects all tracks)
Function:Insert
Location:   1:1
Amount  :    1:0

Locate  :   1:1
```

Location: 1 : 1

The insertion point for the blank time being added is selected as a Bar and Beat Location value. Events that occurred at or after this Bar and Beat, before you insert time, are not erased when you perform this function, rather they are offset by the length of the blank time being added to a Bar and Beat later in the song.

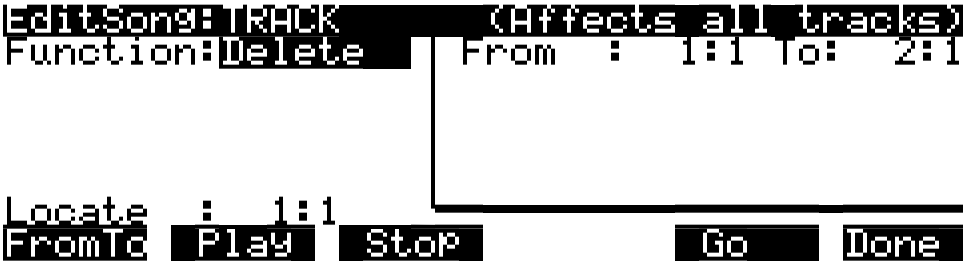
Amount: 1 : 0

The length of the blank time being added is defined as a number of Bars and Beats in the Amount parameter.

There are no Region / Criteria parameters available for the Insert function.

Delete

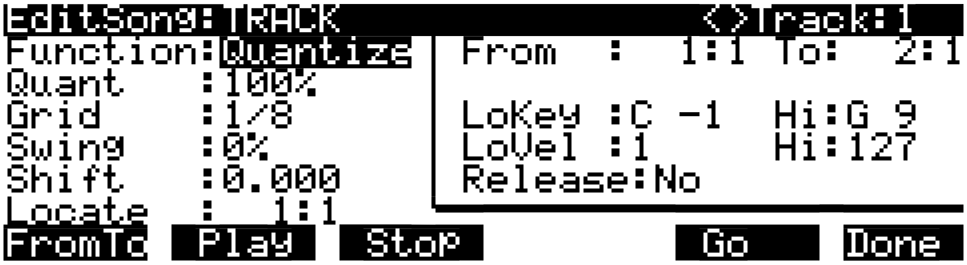
The Delete function is used to remove a region of time from the current song. This function is different from the erase function because not only does it remove the events from the selected time, it will delete the entire selected range of time from the song, modifying the song's End point appropriately (on all tracks). This is similar to cutting a section out of a tape and splicing the ends.



```
EditSong:TRACK        (Affects all tracks)
Function:Delete    From  :  1:1 To:  2:1

Locate  :   1:1
```

Quantize

Use the Quantize function to adjust the timing of Note events. Keep in mind that only Note events are quantized; other types of events, such as controllers, are not quantized.



```
EditSong:TRACK                  <>Track:1
Function:Quantize | From  :  1:1  To:  2:1
Quant    :100%    |
Grid     :1/8     | LoKey :C -1   Hi:G 9
Swing    :0%      | LoVel :1      Hi:127
Shift    :0.000   | Release:No
Locate   :  1:1
```

Quant: Off / 1% – 100%

The Quantize parameter determines how much the selected Note events are moved towards grid locations. If set to **Off**, no aligning of previously recorded notes to grid locations will occur. If set to **100%**, every recorded Note event will be aligned to the closest grid location, defined by the Grid setting. Notes will be moved to a position half way between the grid location and the original Note-event location if Quant is set to **50%**.

Grid: 1/1 – 1/384

This setting determines the size of the Quantize grid, expressed as a fraction of a Bar with a 4/4 meter. Set Grid to **1/1** for whole note grid, **1/16** for sixteenth notes. All of the standard note durations and every fractional Bar divisions in between are available as the size of the Input Quantize grid. Press the **Plus/Minus** buttons simultaneously for quick selection of grid values.

Swing: -99% – 125% (defaults as 0%)

The Swing percentage is applied to the quantize grid. **0%** swing is straight time, **100%** produces a swing feel (triplet feel). A positive Swing value determines how close every other grid location is moved to a point 1/3 of the way towards the next grid point. Negative Swing moves every other grid location closer to a point 1/3 of the way towards the previous grid point.

Shift: -26.020 – 26.020 (Beats and Ticks, 480 Ticks = 1 Beat)

In addition to quantizing the selected note events to specified grid locations in varying amounts, the Quantizing function allows you to offset the original note locations forward and backward in time any number of ticks (1/480th of a Beat) up to 26 Beats, before aligning them to grid locations. Shift is used to compensate for any notes played too early or late.

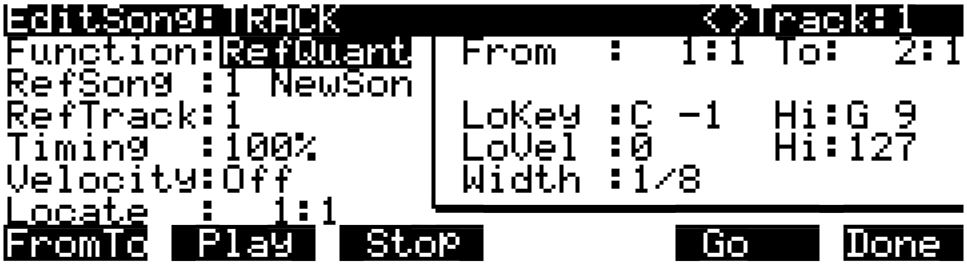
Region / Criteria Box

Release: Yes / No

Set the Release parameter to **Yes** if you would like each quantized note event's Note Off message to be aligned to the grid location nearest to the time the key was originally released.

Reference Quantize

The Reference Quantize function is similar to the Quantize function in that it aligns Note events to a grid. The difference is that the grid locations are not mathematically perfect divisions of a Bar. Instead, Reference Quantize defines the grid based on the timing of note events from a previously recorded reference track.



```
EditSong:TRACK                  <>Track:1
Function:RefQuant  From  :  1:1 To:  2:1
RefSong :1 NewSon
RefTrack:1         LoKey :C -1  Hi:G 9
Timing  :100%      LoVel :0     Hi:127
Velocity:Off       Width :1/8
Locate  :   1:1
```

RefSong: 1   NewSong

Use this parameter to select a song from which you'll then select a reference track. Select the song containing the desired reference track as the RefSong value.

RefTrack: 1 – 16

Use RefTrack to select the track to be used as the reference track. The timing of this track's note events is used as the quantization grid that's applied to the current track when you use Reference Quantize.

Timing: Off / 1% – 100%

The Timing parameter determines how much the selected note events are moved toward grid locations. If set to **Off**, no aligning of previously recorded Notes to grid locations occurs. If set to **100%**, every recorded note event gets aligned to the closest grid location, defined by the timing of note events from the reference track. Notes are moved up to the limit set by the value of the Quant parameter—for example, a value of **50%** for Quant means that notes get moved half the distance from their current location to the nearest grid point.

Velocity: Off / 1% – 100%

In addition to referencing the timing of note events on the reference track, you may also scale the attack velocities of the note events being quantized to velocity values closer or identical to the velocities played on the reference track.

Leave the Velocity parameter set to **Off** if you want to reference only the note event timing when using Reference Quantize. Set Velocity to **100%** to have the velocity values of the note events being quantized exactly match the velocity values of the note events on the reference track. A setting of **50%** changes the velocities to values halfway between the velocity values originally recorded and the values on the reference track.

Region / Criteria Box

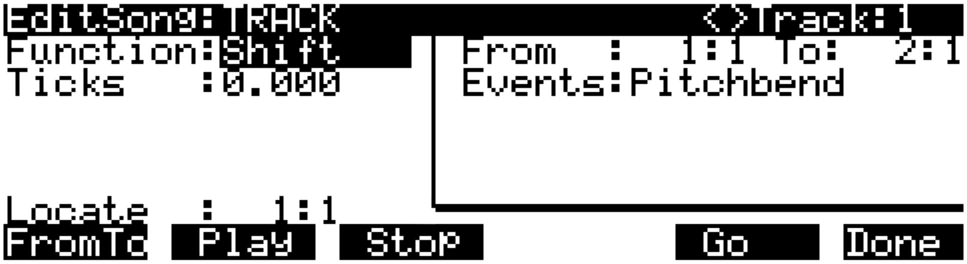
Width: 1/1 – 1/384

The Width setting determines the duration of a window of time centered around each of the referenced grid locations. If a note event on the track being quantized happens during this window of time, it will be moved closer to the referenced grid location according to the Timing percentage. Note events occurring outside this window of time remain unquantized.

Shift

The Shift function allows you to offset the existing MIDI events forward or backward in time any number of ticks (1/480th of a Beat) up to approximately 26 Beats. This function does not affect the End point.

Events can not be shifted beyond the End point or before Bar 1 : Beat 1. The events can be shifted only as far as these temporal boundaries. All events that can’t be shifted the full Ticks amount will be placed at the boundary location.



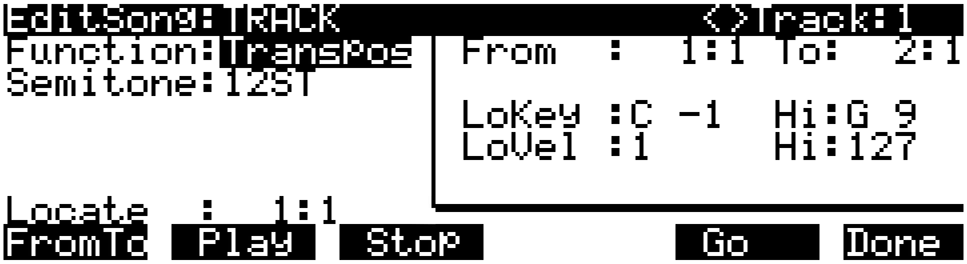
```
EditSong:TRACK                  <>Track:1
Function:Shift     From   :  1:1 To:   2:1
Ticks   :0.000     Events:Pitchbend

Locate  :   1:1
```

Ticks: -26.020 – 26.020 (Beats and Ticks, 480 Ticks = 1 Beat)

The Ticks parameter specifies the number of Beats and Ticks that the MIDI events, from within the selected region, are moved forward or backward in time relative to their original locations.

Transpose

Use the Transpose function to change the MIDI Note numbers of the selected Note events.



```
EditSong:TRACK                  <>Track:1
Function:TransPos  From   :  1:1 To:   2:1
Semitone:12ST
                   LoKey :C -1   Hi:G 9
                   LoVel :1      Hi:127
Locate  :   1:1
```

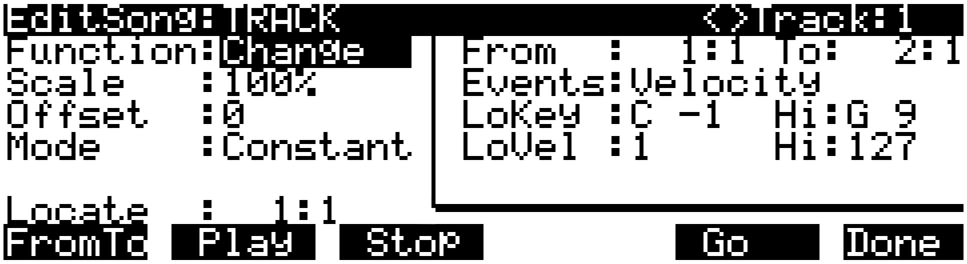
Semitone: -128ST – 127ST

An increment of one semitone represents a change of one MIDI Note number. You can transpose Note events only within the range of MIDI Note numbers 0 to 127.

Change

The Change function is used to modify attack velocities, release velocities, or the values of any existing controller data on the current track. A static change of values can be made as well as having the change take place over a region of time.

Change can not modify or add data that doesn’t exist on the current track. If you hear note events played back on a track, then you know there is an attack and release velocity value for each one, and the effect of the Change function can usually be easily detected. Controller values are sometimes more difficult to change since there can be inconsistent gaps of time between each controller event.



```
EditSong:TRACK                  <>Track:1
Function:Change   | From  :   1:1 To:   2:1
Scale   :100%     | Events:Velocity
Offset  :0        | LoKey :C -1   Hi:G 9
Mode    :Constant | LoVel :1      Hi:127

Locate  :   1:1
FromTo  Play  Stop          Go    Done
```

Scale: 0% – 20000%

The selected velocity or controller events' values can be changed to a percentage of the original values determined by the Scale parameter. A setting of **100%** has no affect. Values are scaled lower with a Scale percentage set from **0%** to **99%**. Low values can be set higher using a Scale percentage above **100%** on up to **20,000%**, although the maximum value of 127 can not be exceeded for any velocity or controller type.

Offset: -128 – 127

Offset can be used alone or in conjunction with Scale to add or subtract a set amount to or from the original (or scaled) values. Values for velocities can not be less than 1 or greater than 127. Values for controllers can not be less than 0 or greater than 127.

As an example, to set all Velocities to a value of 55, you would set Scale to **0%** (multiplies all original values by zero) and set Offset to **55** (adds 55 to the product of the Scale parameter).

Mode: Constant / PosRamp / NegRamp

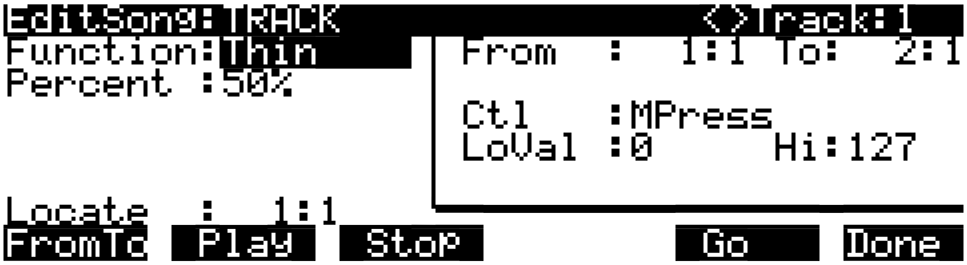
Set Mode to **Constant** to have values modified in a uniform fashion, as determined by the Scale and Offset settings, for the entire selected region of time and range of values.

When the Change function is applied with Mode set to **Positive Ramp**, the selected velocity or controller values will gradually change over the region of time, defined by the locations set for the From and To parameters, from the original value to the new value determined by the Scale and Offset settings. The first events being modified within the region will have little or no change from their original values. The amount of Scale and Offset applied will increase as the song approaches the Bar and Beat defined in the To parameter, where the full amount of described change will occur.

You can set Mode to **Negative Ramp** to achieve the opposite dynamic effect of **Positive Ramp**. **Negative Ramp** works in the same way, but the amount of Scale and Offset applied will decrease from the full amount of change described by Scale and Offset to little or no change as the song approaches the Bar and Beat defined in the To parameter.

Thin

Use the Thin function to reduce the number of actual controller events used for any controller type on the current track. Sometimes a controller can produce the same noticeable effect by using fewer events to describe its change, which in turn will save memory and possibly prevent timing delays caused by an excess of MIDI data.



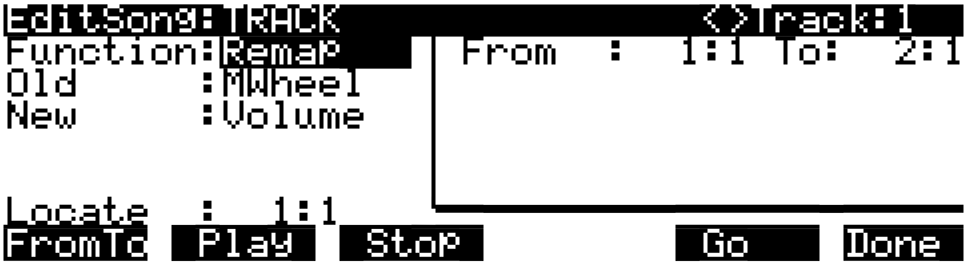
```
EditSong:TRACK                    <>Track:1
Function:Thin      From  :  1:1 To:   2:1
Percent :50%
                   Ctl   :MPress
                   LoVal :0      Hi:127

Locate   :  1:1
```

Percent: 0% – 100% (defaults as 50%)

The Percent value determines how much effect the Thin function will have on a stream of controller data. With this value set to **0%**, there will not be any reduction of data when Thin is applied. Set Percent to **100%** to eliminate most of the specified controller's events from the current track. Even at **100%**, certain controller messages will not be eliminated; these include all pitch bend messages with a value of 0 and the initial settings of some controllers, such as volume, pan, etc.

Remap

Use the Remap function to apply the values of any one type of controller data, already recorded on a track, to another controller type. The effect the real time changes of the "Old" controller had will be replaced by the effect the "New" controller has by using the exact same controller values.



```
EditSong:TRACK                    <>Track:1
Function:Remap     From  :  1:1 To:   2:1
Old     :MWheel
New     :Volume


Locate   :  1:1
```

Old: Controller Codes (0 – 120)

Defined controllers are referred to by their names.

The "Old" controller is set to the controller type that you wish to remap. This controller data must already exist on the current track in order to apply it to the "New" controller type.

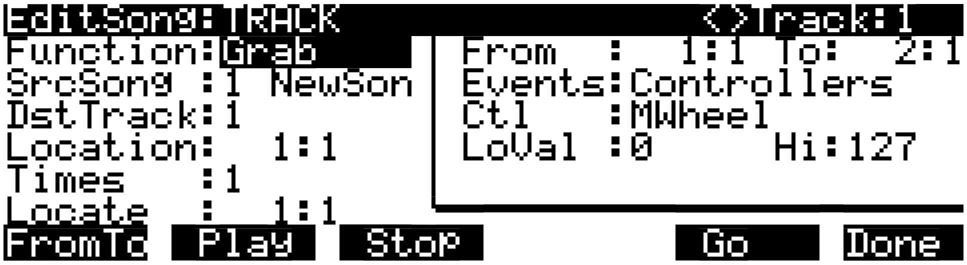
New: Controller Codes (0 – 120)

Defined controllers are referred to by their names.

The "New" parameter is set to the controller code you wish to have use the existing values, once used by the "Old" controller, to produce a different effect.

Grab

Grab is similar to the Copy function, except that the Grab function allows you to copy selected data from tracks that exist in other songs in memory.



```
EditSong:TRACK                 <>Track:1
Function:Grab        From   :   1:1 To:   2:1
SrcSong :1 NewSon    Events:Controllers
DstTrack:1           Ctl    :MWheel
Location:   1:1      LoVal :0       Hi:127
Times    :1
Locate   :   1:1
FromTc  Play  Stop         Go    Done
```

SrcSong: 1   NewSong

The Source Song parameter is set to the ID and name of the song in RAM that contains the desired track data you wish to grab in order to use it in the current song. The source track is determined by the Track parameter displayed on upper right hand side of the page, selectable with the **Chan/Bank** buttons.

DstTrack: 1 – 16 / All

Select a destination track for the grabbed events with the DstTrack parameter. All selected events from the source song and track described in the Region / Criteria box will be placed in the destination track(s) at any Bar and Beat you specify.

If the currently selected track is All tracks then the destination track will be All tracks as well.

No matter what channel the current track (source track in the source song) is set to when you use the grab function, the events will be played on the destination track’s channel.

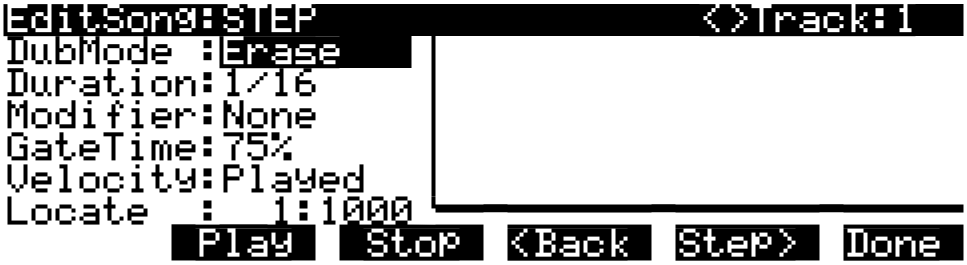
Location: 1 : 1

Specify a Bar and Beat location in the destination track where the grabbed data will be placed with the Location parameter. If the length of the grabbed region extends from the Location point beyond the song’s existing End point, a new End point is defined.

Times: 1 – 127

The value selected for the Times parameter determines how many copies of the selected region are placed, one after another, in the destination track.

Song Editor: The STEP Page



```
EditSong:STEP                    <>Track:1
DubMode  :Erase
Duration:1/16
Modifier:None
GateTime:75%
Velocity:Played
Locate   :   1:1000
```

Parameter	Range of Values	Default
Dubmode	Merge, Erase	Merge
Duration	1/1 – 1/64	1/8
Modifier	None, Triplet, Dotted, DotDot	None
Gatetime	0 – 250%	75%
Velocity	Played, 1 – 127	Played

The Step Editor allows for nonreal-time entry of note events and rests of varying durations and attack velocities at any location within a song. The Step Editor can be used to enter the first note events into a new song, but it will not set the song's End point. Initial program changes, volume and pan settings are remembered as if you recorded the first track using one of the real-time recording modes. The End point will be modified if tracks already containing data are step-edited beyond the current End point.

The top line displays the currently selected RecTrk and can then be switched to any other track. When All tracks are selected, the step edits are performed on the currently selected RecTrk.

Recording With the Step Editor

Recording using the Step Editor is easy. Set the parameters to your desired settings and make sure the Locate parameter is set to the bar and beat at which you wish to start recording. Then just strike a key. The note you play will be displayed in the box on the right along with its velocity or duration. If the Velocity parameter is set to **Played**, then it will reflect the actual velocity you played. Otherwise, it will be specific amount determined by that parameter. The duration of the note is determined by a combination of the Duration and Gate Time parameters.

Once you release the key, the Locate parameter will advance in time, by an amount determined by the Duration parameter. If you want to enter a rest, press the **Step>** soft button without striking a key, and once again the Locate parameter will advance the selected amount.

You can enter note events longer than the selected duration by holding a key and pressing the **Step>** button. You will see the duration of the current note change in the box on the right. This can be a quicker method of entry than changing the Duration parameter if you need a note that is twice or three times as long as the duration. For example, if the Duration is set to **1/8**, you can quickly enter a quarter note by holding the note and pressing **Step>** once. This adds an extra eighth-note duration to the eighth-note duration created by triggering the note. Enter a half note by pressing the **Step>** button three times (adding three eighth-noted durations). Note that if you are using this method and the Gate Time parameter is set to less than **100**%, the truncated duration is applied to the step created by triggering the note. All other steps get the full duration. For example, if you have a Duration of **1/8** and a Gate Time of **75%**, then striking a note will produce a duration of 180 (75% of 240). But if you hold the note and press **Step>**, then the duration will be 420 (180 + 240), instead of 360 (75% of 480).

You can strike more than one note at a time to enter chords. You can also strike several notes one at a time (holding on to the previous notes) to create a chord. As long as you do not release all of the notes, the Locate parameter will stay at the same point and the notes will be entered as a chord. If you want to create a chord where certain notes have longer durations than others, strike the notes of the chord, then release the notes you wish to be shorter and press the **Step>** button. The box on the right will change to show only the notes you are still holding with their longer durations.

If you make a mistake while entering notes, you have two ways to fix it. With Dub Mode set to **Merge**, if you're still holding the note, you can press the **<Back** soft button. If you set the Dub Mode to **Erase**, you can press the **<Back** button to erase the previous event, then strike the correct note. Or you can set the Locate parameter to the proper spot and strike the correct note.

DubMode

DubMode determines if the note events entered in the STEP page will merge with, or erase all other existing MIDI events located on the selected track, and in the Bars and Beats being step-edited.

You would set DubMode to **Erase** to replace all existing track data just in the Bars and Beats you edit. Data before and after the edited locations will be preserved. Set to **Merge** to add note events to existing track data.

The DubMode value will reflect the last value given to the Mode parameter on the Song-mode MAIN page, and if the DubMode in the Step Editor is changed, the Mode parameter on the MAIN page will be changed to the same value.

Duration

The actual duration of a note event entered in step time is determined by the values given for the Duration, Modifier, and GateTime parameters.

The Duration parameter sets the base note duration where **1/1** is a whole note, **1/2** is a half note, **1/32** is a thirtysecond note, and so on. The setting for this parameter, along with its Modifier, determine the size of the jump from the current song location to the next step entry location. The double button press of the **Plus/Minus** buttons is one method of selecting a step's duration.

Use the alphanumeric buttonpad to quickly select a note duration and modifier whenever the cursor is positioned over the Duration parameter. These are the buttonpad duration settings:

1	whole note
2	half note
3	thirtysecond note
4	quarter note
5	sixtyfourth note
6	sixteenth note
7	applies a dotted modifier to current duration
8	eighth note
9	applies a triplet modifier to current duration
0	removes any modifier

Modifier

The Modifier parameter allows you to change the Duration value with three standard modifiers. Set to **None** if you do not chose to use a modifier.

Use the triplet modifier for a resulting duration 2/3 of the value set in the Duration parameter. (For example, an eighth note = 240 ticks, an eighth note triplet = 160 ticks.)

A Dotted modifier adds half of the current Duration value to that duration, and a DotDot modifier will add 3/4 of the current Duration value to itself. (For example, an eight note = 240 ticks, a dotted eighth note = 360 ticks, and a double dotted eighth note = 420 ticks.)

GateTime

GateTime determines the percentage of the note duration, (set with the Duration and Modifier), that is actually played. The value given for the GateTime parameter does not affect the size of the jump from one step to another, but only the length of time that the note(s) entered for each step are sustained. You can use a value greater than 100 for a legato feel, but keep in mind that when two identical notes overlap you may get unexpected results. This is because the Note Off command from the first (overlapping) note will also cut off the second note (when it is identical).

Double-press the **Plus/Minus** buttons to increment this value in 20% intervals.

Velocity

Each note event entered in step-edit mode is assigned an attack velocity either by setting the Velocity parameter to a value between **1** and **127**, or by setting the value to **Played**, where the velocity at which you actually enter notes is used for each step.

Locate

The Bar, Beat and Tick displayed as the Locate value is the location where the next step entry will placed in the song, and once the step is entered, Locate will advance an amount of time determined by the Duration and Modifier settings. The Bar and Beat, but not the Tick, will update during playback. It can be set to any Bar and Beat, including negative values, so that playback begins at, and Stop resets the song to the Locate Bar and Beat.

If the Bar and Beat value is set beyond the current End point, and no notes are entered before you press Play, then that location will be ignored when you begin playback and the song will start from Bar 1 : Beat 1. If set to a negative Bar and Beat, no step entry is possible.

Soft Buttons on the STEP Page

Play starts the playback of the song from the Bar and Beat set in the Locate parameter. When the song is playing, this soft button becomes **Pause**.

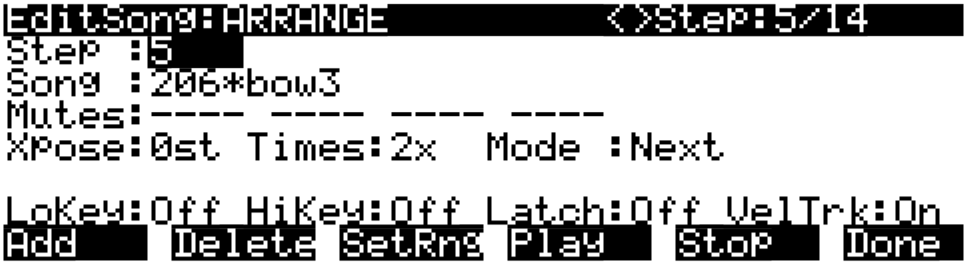
Stop halts the playback of the song and return to the Bar and Beat set as the Locate value.

<Back / Step> moves the current location of the song backward and forward in time, as reflected in the Locate Bar, Beat, and Tick. How far the location is changed from its current setting is determined by the Duration and Modifier settings.

Done returns to the COMMON page when you are finished step editing.

Song Editor: The ARRANGE Page

This page allows you to create a song by arranging other songs together in the order you specify. The other songs become sections (called steps) of the current song, which can be repeated, transposed, etc. You can even trigger steps by striking specific keys on the controller.

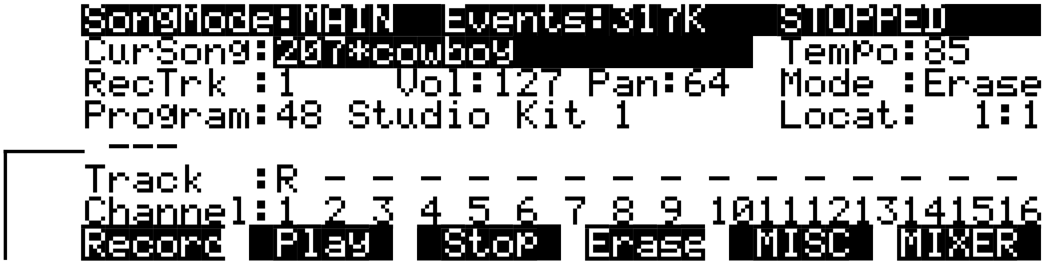


```
EditSong:ARRANGE           <>Step:5/14
Step :5
Song :206*bow3
Mutes:---- ---- ---- ----
Xpose:0st Times:2x  Mode :Next

LoKey:Off HiKey:Off Latch:Off VelTrk:On
Add    Delete SetRng Play   Stop   Done
```

You can also create an arrangement of other songs to play back simultaneously with the current song. This gives you up to 32 tracks: 16 tracks in the current song plus 16 in the song or songs called by the arrangement (which are specified by the ARRANGE page’s Song parameter). Alternatively, the current song might not contain any normal song data, but could include only the arrangement data, by which the pieces of a composition are assembled. One thing you can’t do, however, is “nest” arrangement songs. In other words, if you include a song with arrangement data in another song with arrangement data, it won’t work.

More often than not, you would want to start using the Arrange feature from a new song that doesn’t have any track data recorded yet. Press the **ARRANG** soft button on the COMMON page to enter the ARRANGE page, gaining access to a group of arrangement specific parameters. All songs have these parameters for creating complex arrangements of existing song objects, using ranges of the keyboard and realtime performance controls for triggering steps, but not every song will use them.

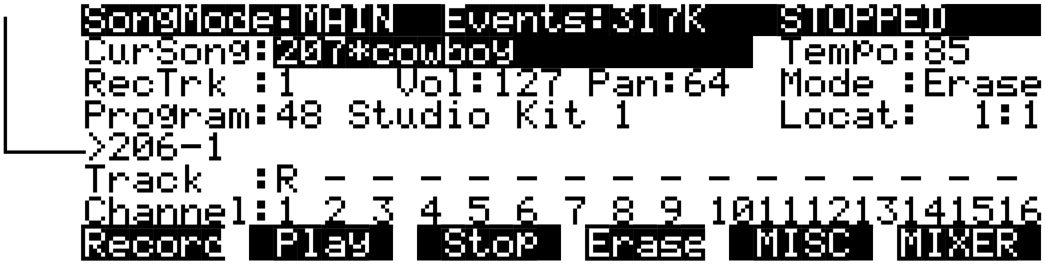
When a song containing an arrangement is selected as the CurSong on the MAIN page, three dashes (---) appear above the Track parameter name to indicate that the current song has been saved with arrangement information.



```
SongMode:MAIN  Events:317K    STOPPED
CurSong:207*cowboy            Tempo:85
RecTrk :1    Vol:127 Pan:64   Mode :Erase
Program:48 Studio Kit 1       Locat:   1:1
 ---
Track  :R - - - - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record   Play    Stop    Erase   MISC    MIXER
```

These three dashes show that the song includes arrangement information.

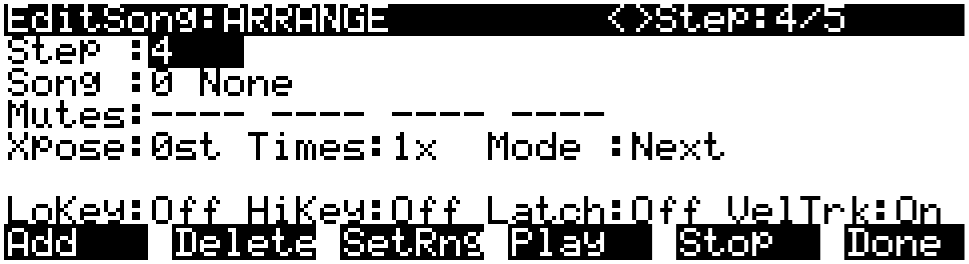
When you play a song that includes arrangement information, dashes are replaced by the play indicator (>), song ID number, and step number.



```
SongMode:MAIN  Events:317K    STOPPED
CurSong:207*cowboy            Tempo:85
RecTrk :1    Vol:127 Pan:64   Mode :Erase
Program:48 Studio Kit 1       Locat:   1:1
>206-1
Track  :R - - - - - - - - - - - - - - -
Channel:1 2 3 4 5 6 7 8 9 10111213141516
Record   Play    Stop    Erase   MISC    MIXER
```

When an arrangement is PLAYING, the three dashes on the MAIN page are replaced with the ID of the song currently being played (or triggered) in the arrangement, and the number of the current step.

The square Track Status Indicators will flash when MIDI data is present on the tracks during play back of an arrangement. The actual track data playing back is supplied by the songs entered into the steps of an arrangement.



```
EditSong:ARRANGE            <>Step:4/5
Step :4
Song :0 None
Mutes:---- ---- ---- ----
Xpose:0st Times:1x  Mode :Next

LoKey:Off HiKey:Off Latch:Off VelTrk:On
Add     Delete  SetRng  Play    Stop    Done
```

Parameter	Range of Values	Default
Step	1 – 99	1
Song	Any song In Memory	0 None
Mutes	– , M	– (Not Muted)
Xpose	-128ST – 127ST	0ST
Times	1x – 120x, Infinite	1x
Mode	Next, Stop	Next
LoKey	C-1 – G9, Off	Off
HiKey	C-1 – G9, Off	Off
Latch	On, Off	Off
VelTrk	On, Off	On

The first number displayed in the **Step** field on the top line is the current step. The second number is the total amount of steps in the arrangement. The current step in the display above, for example, is the fourth step in an arrangement consisting of five steps. Use the **Chan/Bank** buttons to select the current step for editing.

Step

The number of steps in an arrangement defaults to 1. All other steps are then added when needed. The step of the arrangement selected for editing is displayed as the step. Use the **Chan/Bank** buttons to select the different steps in your arrangement. StartStep on the COMMON page (see page 12-41), as well as the key ranges of the steps, control what steps are played. The behavior and control of each step is defined with the parameters described below.

Song

The Song is the ID and name of a song object in memory that will play back in proper numerical order with the other steps in an arrangement or will respond to any triggering from a defined keyboard range in that step. This song's (up to) 16 tracks will play in addition to any tracks in the current song.

Mutes

The possible sixteen tracks of the current step's song can be individually muted. Tracks displaying an (M) for the Mutes parameter are muted during playback of the song in the arrangement.

Xpose

Each step in an arrangement can impose a transposition on the song selected for its Song parameter during playback. Xpose determines how many semitones a step's song is transposed above or below the song's original key during playback. All tracks of the step's song will be transposed, except for those tracks defined as drum tracks in the COMMON page of the step's song. See page 12-43 for a discussion of drum tracks.

Times

Times sets the number of times that an arrangement step plays before the arrangement moves on to its next step. You can set this value to play the step up to 120 times before the next step is played, or you can specify an infinite loop (**Inf**), in which case the arrangement will play that step until **Stop** is pressed.

Mode

The Mode setting found in the Arrangement Editor determines whether an arrangement plays the next step or if the sequencer will stop after playing the current step. Unless you want the arrangement to loop (that is, return to step one after playing the song's final step) set the final step's mode to stop.

Triggering Steps from a Key

You can play and transpose the steps in your arrangement with the keys on your keyboard. This is a good way to try out arrangement or composition ideas, and also has some exciting live performance applications. If you were suddenly struck by inspiration at a gig, for example, you could spontaneously repeat or transpose sections of an arrangement. See page 12-63 for more information.

The four parameters along the bottom of the ARRANGE page let you set up real-time control of the steps in an arrangement. Make sure that the TriggerCtl parameter, found on the COMMON page (see page 12-43), is set to **ON**, which is the default, or to the global control source that you are using to enable this feature. TriggerChan, also on the COMMON page will need to be set to the current MIDI channel; the default is channel **1**.

The control setup's Sync parameter (on the COMMON page in the Setup Editor) affects how steps begin—immediately, or at the beginning of the next measure—when you trigger the steps from a key. See page 7-36 for a description of the Sync parameter.

LoKey

The first parameter, LoKey, is the setting for the lowest note in a keyboard range, that when played by the TriggerChan, will trigger the start of the current step. Triggering a step from the LoKey will play back its song in its original key transposed the number of semitones set in the Xpose setting. As you play up the keyboard chromatically, the step will transpose its song in semitone increments. All tracks of a song being transposed by keyboard triggering will be transposed accordingly, except for tracks defined as Drum Tracks on each song's COMMON page.

HiKey

The highest note of the keyboard range designed to trigger the current step is set in the HiKey parameter. HiKey defines the largest transposition of a step from the key of the song triggered by the LoKey. The HiKey must be a higher note value than the LoKey in order to trigger any step from the keyboard.

Latch

If Latch is set to **Off**, any step you trigger with keystrikes continue to play only as long as you hold down the key that triggered it. Set Latch to **On** if you want to trigger the step with a keystrike, and have it continue playing after you release the key. In this case, the step plays for its entire length, or until you either press **Stop** or retrigger the step.

VelTrk

Turn the VelTrk parameter **On** to make your real-time arrangements more dynamically controllable. The original attack velocity of every Note event in the triggered step's song will be scaled to values determined by the attack velocity of the note you play when triggering.

Soft Buttons on the ARRANGE Page

Add adds another step to the arrangement. The new step will be inserted as the step number directly after the current step. You would add a step to place a new section into a song arrangement or to define another keyboard range for triggering.

Delete removes the current step.

SetRng lets you quickly define the keyboard range for triggering the current step from the MIDI keyboard. This dialog will appear:

Strike low key...

Strike low key…

Play the lowest note of the range you wish to define, then:

Strike high key…

Play the highest note of the range you wish to define. At any time you may abort the range setting procedure by pressing the **Cancel** soft button, which returns to the ARRANGE page.

Play lets you audition your arrangement from the ARRANGE page, starting playback from the currently selected step.

Stop halts the playback of an arrangement from the ARRANGE page.

Done exits the ARRANGE page and returns to the COMMON page.

Chapter 13
Disk Mode

Disk mode lets you load, save, back up, and copy files of objects between the K2600 and the outside world, through the K2600's floppy drive or SCSI ports. The floppy drive accepts MS-DOS® format double-sided double-density (DSDD—720K) and high-density (HD—1.4 M) floppy disks. To save data from the K2600 to a floppy disk, the disk must be formatted (initialized) by the K2600 or by a computer running MS-DOS-compatible formatting software.

Most SCSI (Small Computer System Interface) devices will operate with the K2600 via its 25-pin SCSI ports. The most common use for these ports is to connect one or more hard disks (or removable drives like Zip or Syquest) for storing samples and other objects. You can also connect a CD-ROM drive for reading files to the K2600. The K2600 will treat a CD-ROM drive like any other SCSI device (except that you can't save files to it). The K2600 can read writable CDs (CD-Rs), although it can't write to them.

The K2600 can address up to 2 gigabytes of file space on any one hard disk, so you can connect a maximum of 14 G of storage—up to seven hard disks. The K2600 can use all of the space on disks of 2 G and smaller. On larger disks, formatting with the K2600 creates a 2 G partition on the disk; the K2600 can read only that partition. If you format a hard disk on a computer, make the first partition just under 2 G. You can create other partitions, of any size, but the K2600 can read only the first partition.

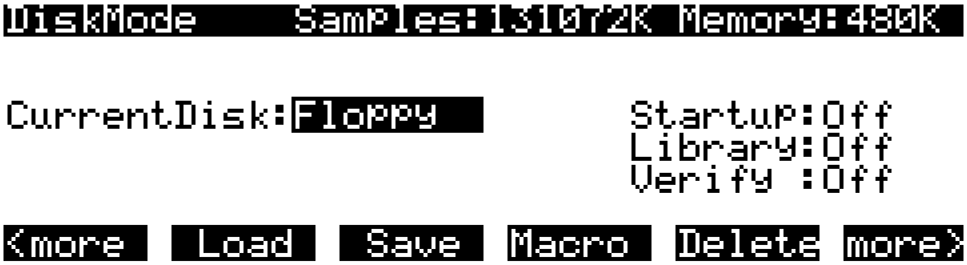
Disk mode in the K2600 allows flexibility to organize disk files and their contents. Many powerful operations are included that can save a lot of time by allowing you to easily specify exactly what you want to load or save. Examples of this range from organizing related files into directories, to loading macros (lists of files or selected objects) from multiple SCSI disks, to setting up programs to automatically link with samples off of a CD-ROM.

Here's a summary of Disk-mode functionality:

- One floppy drive, two SCSI ports
- MS-DOS file system compatibility
- Sample transfer using standard audio file formats AIFF and WAV (including support for looped and tuned WAV files)
- Support for Roland,™ Akai,™ and Ensoniq™ sample files
- SMDI sample transfers
- Support for song files (sequences) in MIDI Type 0 and Type 1 format
- Support for ISO 9660-format CDs: reading, copying, and backing up
- Support for writable CDs (CD-Rs):

Disk Mode Page

To enter Disk mode, press the **Disk** button, and the Disk-mode page will appear:



```
DiskMode     Samples:131072K Memory:480K

CurrentDisk:Floppy          Startup:Off
                            Library:Off
                            Verify :Off

<more   Load   Save  Macro  Delete more>
```

As usual, the current mode is displayed on the top line. At the middle of this line, the amount of available sample RAM is shown (if you have added optional sample RAM; if you haven't, this part of the top line will be blank). To the right of the top line you see the amount of memory available for storing all other RAM objects. Straight from the box, this number should be around 500 K. If you have added the optional P/RAM extension kit, the number will be about 1.5 M. The P/RAM kit enables you to store hundreds of additional programs, songs, and any other objects. If you do a lot of programming or sequencing, the P/RAM option is a good idea.

In the center of the page is a line indicating the currently selected disk. Select different disks using any data-entry method. You can select a floppy disk, or SCSI IDs 0–7. (If you connect an Apple Macintosh® personal computer, don't select SCSI ID 7, since that's the SCSI ID of the Mac, which can't be changed. All SCSI devices connected in a network must have different SCSI IDs in order for the network to function.) When you want to communicate with any of the SCSI storage devices in your network, set the Current disk parameter to the value that matches the SCSI ID of the device you want to address. That is, the K2600 will interact with the SCSI storage device whose SCSI ID matches the value of the Current Disk parameter. If you choose a value of **Floppy**, the K2600 will communicate with its floppy disk drive.

The manual for your SCSI disk should tell you its SCSI ID. Most newer SCSI disks show their SCSI IDs on their rear panels, and many have adjustable SCSI IDs.

The currently selected device will be read from or written to when you load, save, rename, or delete files. Use the soft buttons to start any of these operations. Refer to*Disk Mode Soft Buttons* on page 13-6 for complete information.

Formatting a Floppy Disk

Floppy disks must be formatted (initialized) to the Kurzweil format before they can be used with the K2600. The K2600 uses the MS-DOS format, so floppies formatted on DOS computers (or any computer that can format disks in DOS format) should work with the K2600. To format a disk on the K2600, insert a floppy disk in the K2600's floppy drive, or in the floppy drive of any computer with MS-DOS formatting capability. Make sure the disk is unlocked (the sliding switch on the back of the disk is set so you can't see through the hole on that side of the disk).

Press the **Disk** button to enter Disk mode. Make sure the Current disk parameter says **Floppy**, so you don't accidentally format any SCSI devices you might have connected! Press the soft button labeled **Format**. The K2600 will ask you if you want to format, and a pair of **Yes/No** soft buttons will appear. Press the **Yes** soft button, and three soft buttons will appear, letting you choose between formatting the floppy as a 720K (double-density) or 1.4M (high-density) disk. Press the appropriate soft button, or press the **Cancel** soft button.

If the floppy you inserted is double-density and you press the **1.4M** soft button, the format procedure will fail. This is also true if the disk is high-density and you press the **720K** soft button. If the floppy is single-sided (SSDD), you can press the **720K** soft button, and the floppy will be formatted as a double-sided disk. (The only difference between single- and double-sided floppies is that both sides of a double-sided floppy have been tested by the manufacturer.)

Once you select between the 720K or 1.4M format, the K2600 will remind you that formatting will erase the floppy, and will give you two more chances to cancel the formatting procedure—we want to make sure you don't accidentally erase any disks. Press the **Yes** soft button to continue formatting. When formatting begins, the display will tell you that the disk is being formatted. You'll hear the disk drive turning, and the disk drive LED will light.

Formatting a 1.4M floppy disk takes just under three minutes, including the automatic verification. When formatting is finished, the K2600 asks you if you want to format another disk.

Connecting a SCSI Device

It's easy to connect SCSI devices to the K2600's SCSI ports. Using a SCSI device will give you off-line storage, and can speed up your loading and saving operations considerably.

You'll need a SCSI cable with a 25-pin SCSI connector on the end to be connected to the K2600. If your SCSI device does not have a 25-pin connector at one end, you can find SCSI cables like these at any personal computer store. Connect the 25-pin end of the cable to either of the K2600's SCSI ports, and the other end to your SCSI device. Two SCSI ports are provided to enable you to chain SCSI devices together. Before you start connecting cables, however, please read the next section carefully. We've also included important information about SCSI in Chapter 6 of the *Musician's Reference*.

SCSI Termination

Simply put, SCSI termination prevents the electrical signals used by SCSI devices from being reflected from unconnected SCSI ports, and possibly disrupting the data stream. The rule for SCSI termination is that the two SCSI devices on the ends of a chain of SCSI devices must be terminated, and all devices in between, however many, must be unterminated. Newer SCSI devices usually make it easy to enable or disable their termination settings. Older SCSI devices may require an external terminator to be installed. These are available at all personal computer stores. Make sure you get the right size for your device (25-pin or 50-pin).

It's impossible to describe all the possible configurations of SCSI devices, so we'll provide you with a few general guidelines that will cover the requirements for most SCSI systems. If you're chaining large numbers of SCSI devices together, you may have to do a little juggling, but chances are you'll already have some experience with SCSI termination.

First of all, it's *very* important that you terminate your SCSI system properly. Improper termination can result in lost data, can interfere with the operation of your SCSI devices, and over the long term, can damage them.

If your SCSI system includes a personal computer, you'll need to be sure that it is internally terminated. If you're not sure whether it's internally terminated, you should call your local computer dealer for confirmation.

If your SCSI system includes only the K2600 and an internally terminated computer, you're probably all set. First, check the SCSI Termination switch on the K2600's back panel (next to the SCSI ports), and make sure that it's set to Auto. Then connect the computer's SCSI port to either of the K2600's SCSI ports, and you're ready to roll. With the K2600's termination enabled, you've satisfied the requirement of terminating both ends of the chain.

If you have an internally terminated computer, a K2600 and an external hard disk with *two* SCSI ports, setting up is also painless. Connect the computer's SCSI port to one of the hard disk's SCSI ports, and the K2600's SCSI port to the hard disk's other SCSI port. Make sure the hard disk is not terminated, since it's in the middle of the chain. In this configuration (with a terminated computer at one end and the K2600 at the other), you can chain up to six hard disks between them. Make sure they're all unterminated, and don't forget to set each disk's SCSI ID to a different value.

Disabling the K2600's SCSI Termination

If you have a computer, a K2600 and an external hard disk with only *one* SCSI port, there's only one way to set them up: put the K2600 in the middle (with one SCSI port connected to the computer, and the other to the hard disk). In this case, you can either change the setting of the SCSI Termination switch to Disable (which means that termination is always off), or you can leave it set to Auto, which means that if you put the K2600 in a configuration that requires it to be terminated, it will enable its internal termination automatically (and disable it when appropriate, as well).

If you install an internal hard disk in your K2600, we recommend that you leave the K2600's SCSI Termination switch set to Auto. If your K2600 is going to be at the end of a SCSI chain (which is what we recommend), then the internal hard disk must be terminated. If the K2600 is going to be in the middle of a chain, the internal hard disk must be unterminated (we don't recommend this).

If you're planning to buy an external SCSI hard disk to use with your K2600, it's a good idea to buy one with two SCSI ports. Most new hard disks have two ports, and can be terminated or unterminated relatively easily. This gives you added flexibility, since you can install it at the end of a chain, leaving its termination in place, or in the middle of a chain, using both its SCSI ports, and removing its termination.

When your SCSI device is connected, you can select it with the Current disk parameter on the Disk-mode page. Use any data-entry method to select the SCSI ID that matches the SCSI ID of your SCSI device. If you're using the alphanumeric buttonpad to select the device, enter **9** to select the floppy. Newer SCSI devices usually have an external switch for setting their IDs. Older units may not have these; check your device's owner's manual for its SCSI ID.

Using your K2600 in a SCSI System

SCSI IDs

All devices in a chain of SCSI devices must have different SCSI IDs, including the K2600. The K2600's SCSI ID is set at 6 by default, and can be changed on the RECEIVE page in MIDI mode. If your SCSI system includes an Apple® Macintosh,® be sure not to use SCSI ID 7 for any of your other devices, since the Mac's SCSI ID is 7, and can't be changed.

Once you've made sure that all connected devices are set to different SCSI IDs, you should be able to select the devices, format them, and start loading and saving files.

Formatting a SCSI Device

The procedure for formatting hard disks is essentially the same as with floppy disks, once the SCSI device is selected with the CurrentDisk parameter. The K2600 will recognize the disk as a SCSI disk, and will warn you that formatting will erase the contents of the disk. Compared with personal computers, the K2600's formatting time for SCSI disks is surprisingly short.

Directories

A directory is a file on the disk that lets you group other files together as you might separate documents using folders in a file cabinet. You can create directories on K2600 Format SCSI and floppy disks.   You can even create directories within directories; these are called subdirectories.

Directories are very useful for organizing your sample, song, and program files. The K2600 provides many operations for setting up and managing the directories on your disks and the files within them.

Path

The Path field shows the current directory on the current disk if it is a K2600 format disk. This field is displayed upon returning to the Disk-mode page after you have pressed one of the disk function soft buttons and viewed the file contents of a specific disk. It stays visible on the Disk-mode page until you power down or do a soft reset.

The K2600 always starts at the root (top-level) directory when you power it up, or when you change the value of the CurrentDisk parameter. When you use the disk functions to view other directories, the Path field updates the current directory value to track your movements.

The root directory is displayed as a backslash:

```
Path = \
```

If you press the **Load** button and load a file from a subdirectory called SOUNDS, the Path field will appear as

```
Path = \SOUNDS\
```

The backslash character is a directory separator, as in the following Path:

```
Path = \NEWTUNE\SAMPLES\DOGS\
```

This represents the directory DOGS, which is a subdirectory of the SAMPLES directory, which is a subdirectory of the NEWTUNE directory in the root directory. If the path is too long to fit on the top line of the display, it gets abbreviated. The maximum length of a path in the K2600 is 64 characters (including the backslash characters).

Startup

The Startup parameter determines what disk will be used for loading the power-up macro file **BOOT.MAC** (see *Creating a Startup File* on page 13-69). If this is set to **None**, then the K2600 will power-up in a normal fashion. If this is set to a SCSI device or **Floppy**, when the K2600 is next powered on it will look for the **BOOT.MAC** file in the root directory of the specified disk, and load each of the entries in the macro specified within.

This feature provides a very flexible way to automatically configure your K2600’s memory contents whenever you turn the power on.

Library

This feature works in conjunction with the macro feature to provide a way to distribute macro files that load data from removable media without having to know in advance the SCSI ID of the removable-media drive. A macro file stores its references to disks by DISK ID (SCSI ID or Floppy), or by either a “Library” or “Unspecified” designation (see *Macros* on page 13-43). Typically, you would set the Library parameter to be the same as the SCSI ID of your CD-ROM drive, if you were loading macro files from a Floppy disk or another SCSI disk that referenced CD-ROM files containing samples or keymaps.

Verify

Set Verify to **On** when you want the K2600 to verify saves, copies, and backups (the K2600 can’t verify loads). The operations take longer, but it provides insurance against corrupted files.

Disk Drive Information

Below the device type and disk capacity you’ll see specific information about the current disk’s manufacturer, model number and internal mechanism (this is true for SCSI disks only; no information appears about floppy disks). The K2600 requests this information from a SCSI disk when you select that disk with the Current Disk parameter. This information may be needed when determining if a given disk is compatible for SCSI operation with the K2600.

Macro On Indicator

When (Macro on) is visible, the K2600 records all file-loading operations in its macro table. See *Macros* on page 13-43.

Disk Mode Soft Buttons

Here is a brief description of each of Disk mode’s soft button:

Load Load selected file(s) or object(s) from the current disk into K2600 memory.

Save Save banks of objects, selected objects, or a macro as a K2600 file on the current disk.

Macro Display the macro function page, where you can create and edit macros.

Delete Delete files from the current disk if it is a K2600 disk. See *Deleting Files and Directories* on page 13-66.

Rename Change the filename of a file on a K2600 disk. See *Renaming Files* on page 13-65.

Move Change the location of a file from one directory to another (on the same disk).

Util Check the free space, find files, and view directory organization and sizes on the current disk.

NewDir Create a new directory on K2600 disks.

Backup Hierarchical file backup between disks.

Copy Single or multiple file copy between disks.

Sleep Send SCSI sleep command to the current disk. See the discussion below.

Format Format the current disk as a K2600 disk.

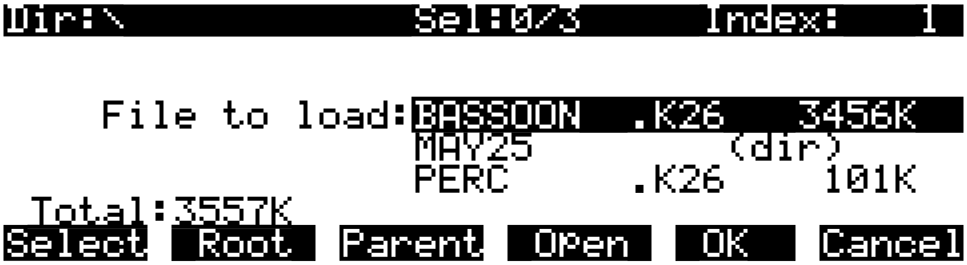
The Sleep Soft Button

Many SCSI devices will “sleep” when they’ve been idle for a few minutes. In other words, the disk will stop spinning, in order to save power and reduce wear. The K2600 lets you tell your SCSI devices to sleep. Just press the **Sleep** soft button, and if your devices have this feature, they will sleep. This is particularly useful in a quiet studio situation.

Any Disk-mode operation will “wake” the device again. The K2600 will ask you to wait while the device’s disk starts spinning. As soon as the disk is spinning at full speed, the K2600 will execute the operation you selected. Some SCSI devices automatically sleep when they power up. (A device of this type usually provides a way to override this feature; check its manual.) Any Disk-mode operation will wake a disk in this case, as well.

File List Dialog

The file list dialog appears when you select a disk function (such as Load or Rename) to operate on one or more files on a disk. Here is a typical file list dialog, for the Load function:



```
Dir:\                     Sel:0/3      Index:    1

    File to load:BASSOON   .K26    3456K
                 MAY25          (dir)
                 PERC     .K26    101K
 Total:3557K
```

When you enter this dialog, the K2600 displays the contents of the current directory, in an alphabetized scrolling list. If the current directory cannot be located (for example, if you’ve changed floppies or removable hard disks), the K2600 displays the current disk’s root directory. The root directory will also be selected if the disk was just chosen by the CurrentDisk parameter on the Disk-mode page (remember that the current directory is always set to the top level when the CurrentDisk parameter is changed, or if the K2600 has just been powered on).

The display for all disks (including floppy) shows the 3-character extension of all files in the directory (except directories themselves). Extensions are created when the file is saved by the K2600. You cannot modify the extensions on the K2600. This is because the K2600 uses the extensions to tell it what kind of data the files contain.

Directories created by the K2600 have up to 8-character names, with no extension. A directory can have an extension if it is created on an external computer (more on this later).

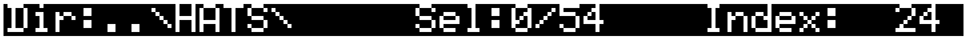
Here is a list of extensions used by or accepted by the K2600:

.AIF	Audio Interchange File Format (AIFF)
.KOS	Kurzweil K2500 or K2600 operating system file
.KRZ	Kurzweil K2000 format file
.K25	Kurzweil K2500 format file containing objects and/or sample data
.K26	Kurzweil K2600 format file containing objects and/or sample data
.MAC	Kurzweil K2500 or K2600 disk macro file
.MID	MIDI Type 0 or Type 1 sequence file
.WAV	Microsoft RIFF WAVE format

***Note**: In most cases, when we refer to **.K26** files, we're including the older-format **.K25** and **.KRZ** files as well, since the K2600 can read these file formats.*

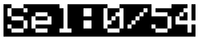
When loading files, the K2600 will try to find out the type of file if the extension is not the same as is suggested above (with one exception: **.MAC** files). The K2600 can create files with almost all of the above extensions; the exceptions are the older-format **.KRZ**, **.K25**, and **.KOS** files.

The top line of the file list contains several items of information pertaining to the currently displayed directory contents. A typical information line looks like this:



```
Dir:..\HATS\       Sel:0/54     Index:   24
```

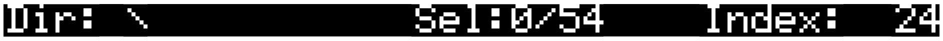
In the center of this line is an indicator of the number of files in the currently displayed directory. This number is grouped together with the number of selected files, for example:



```
Sel:0/54
```

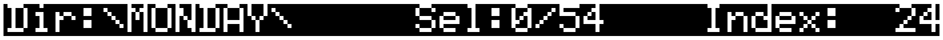
This example indicates that you have selected none of the 54 files in the current directory,. File selection is possible in several of the disk functions (more on this below). The total number of files also includes any subdirectories of the current directory, but not the files within the subdirectories.

On the left end of the top line of the file list page is the current directory, sometimes in an abbreviated form. If you are in the root directory, the display will read:



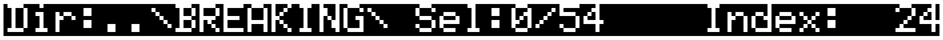
```
Dir: \             Sel:0/54     Index:   24
```

If you are in the directory \MONDAY, the display will read:



```
Dir:\MONDAY\       Sel:0/54     Index:   24
```

If you are in a directory that is more than one level down from the root directory, such as \FX2\GLASS\BREAKING, the display will read:

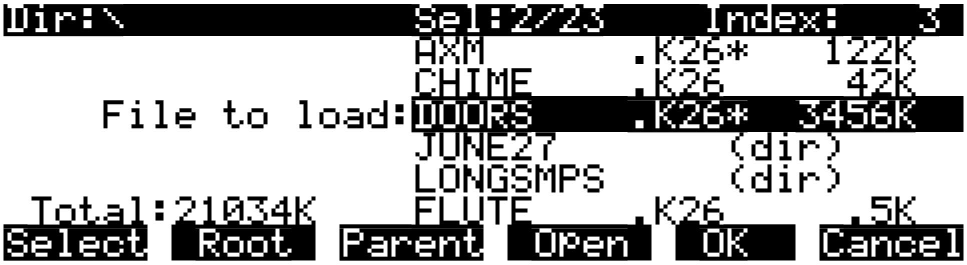


```
Dir:..\BREAKING\ Sel:0/54     Index:   24
```

The "..\" indicator tells you that you are more than one level down from the root directory.

The File Index

On the right side of the top line is the Index field. This tells the position of the highlighted file relative from the beginning of the file list. The first entry in a file list is index 1.



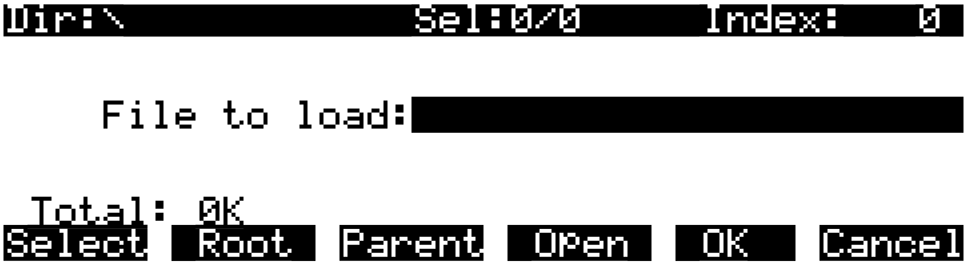
```
Dir:\                   Sel:2/23     Index:    3
                        AXM      .K26*   122K
                        CHIME    .K26     42K
    File to load:DOORS    .K26*  3456K
                        JUNE27       (dir)
                        LONGSMPS     (dir)
 Total:21034K           FLUTE    .K26     .5K
```

Typing a number on the alphanumeric buttonpad will automatically scroll the display to the corresponding entry in the file list. Typing an out-of-range value such as 999 is a quick shortcut to get to the end of the file list.

In addition to remembering the current directory on the most recently used disk, the K2600 also remembers the index within the file list for the current directory. For example, if you were to hit **Cancel** on the above page, go to Setup mode to check the current setup, then return to Disk mode to load a file, the file index would still show **3 DOORS.K26** after you pressed **Load**. This index is remembered until a new disk is selected by changing the value of the Current Disk parameter on the Disk-mode page.

There are exceptions to this however. For example, when a file is written to the disk using the Save function, the index will subsequently be set to the file that was just saved. The index can also be explicitly set using the List and Find utilities (see *Disk Utilities* on page 13-60).

If there are no files in the current directory, then the index is 0, and no value appears for the File to load parameter:



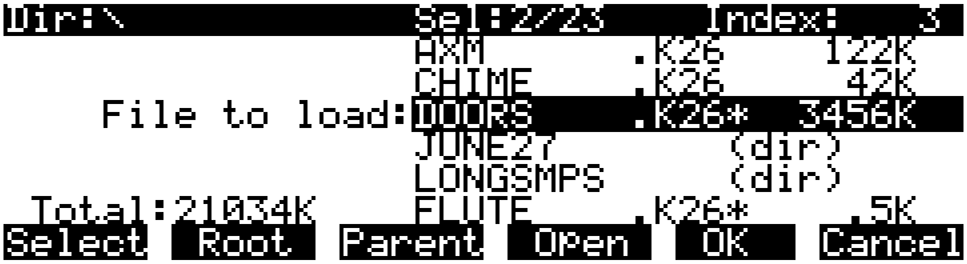
```
Dir:\                   Sel:0/0      Index:    0

    File to load:

 Total: 0K
```

The maximum number of files that can be accessed within a single directory is 360. If you have more files than this amount in a single directory, then you will not be able to view the entries past index 360.

While in this dialog, pressing the **Chan/Bank** buttons will scroll the file list either forward or backward by “pages” of 5 entries. It is often easier to scroll the list this way when looking to see if a particular file is present in a directory.

Soft Buttons in the File List Dialog

Use the **Select** soft button for multiple file selection in the Load, Delete, and Move functions. In the display below, there are two files selected (**DOORS.K26** and **FLUTE.K26**), as indicated by the asterisk (*****) following their filenames. If you pressed **OK** in the following display:



```
Dir:\                Sel:2/23     Index:    3
                     AXM       .K26     122K
                     CHIME     .K26      42K
    File to load:DOORS     .K26*   3456K
                     JUNE27        (dir)
                     LONGSMPS      (dir)
 Total:21034K        FLUTE     .K26*     .5K
```

the files **DOORS.K26** and **FLUTE.K26** would be loaded.

The **Select** button will toggle the selection, meaning that if you press **Select** on a given file, the asterisk will go on if it is currently off, and vice-versa. Selecting can be done for files only, not for directories. You can select as many files as you wish using the **Select** button. There is also a way to select all files at once, or clear all file selections at once, using a double-press of the cursor buttons:

- **Left/Right** cursor double-press: Select All Files
- **Up/Down** cursor double-press: Clear All Selections

Pressing either the **Left** or **Right** cursor individually performs a separate function for finding directories, described below.

You can select multiple files only within a single directory. Changing directories clears any selections.

Once you have selected one or more files, press **OK** to perform the disk function (in this example, Load) on all files marked with an asterisk, regardless of whether they’re visible in the display. If there are no files marked with an asterisk, the function operates only on the highlighted file.

The **Root** soft button returns you to the top-level directory. If the display is already at the root directory (as indicated by the Dir field on the top line of the display) the only effect of pressing Root will be to reset the file index to1 if there are files in the directory.

The **Parent** soft button moves you up one level in the directory hierarchy. If the display is already at the root directory, this button has no effect.

The **Open** soft button performs a different operation depending on the disk function and the type of the currently highlighted file or directory (or selected files). In all disk functions, pressing **Open** on a directory—indicated by **(dir)** after the filename—will open that directory and display its file list.

When you first open a directory for viewing, the index is 1 (the first file in the list). The K2600 remembers the index of the previous directory you were in before you pressed **Open**, so if you return to that directory by pressing **Parent**, the index changes accordingly. This index is remembered for one level down, and therefore is useful when stepping through a list of subdirectories from a single directory level.

In the Load function, pressing **Open** for a standard **.K26** file will start the Load Object feature. This allows selected individual objects from the file to be loaded into the K2600. If **Open** is pressed on a macro file (**.MAC** extension), then individual file entries within a macro file can be selected for loading.

For all other functions, if **Open** is pressed when a **.K26** or a **.MAC** file is highlighted, the object file or the macro file will be opened for viewing. For example, pressing Open on a **.K26** file while in the Delete function will display the objects within the file in a scrollable list, however no delete action will be possible on the individual objects.

Pressing the **OK** soft button will cause the K2600 to proceed with the selected function. After pressing **OK**, there may be further dialogs such as bank specification (for the Load function), confirmation (for Delete), or name entry (for Rename). One exception to this is in the Load function; when a directory is highlighted, pressing **OK** is the same as pressing **Open** (it displays the contents of the highlighted directory).

The **Cancel** soft button exits the file list dialog, completing the disk function with or without any operation taking place. The K2600 returns to the Disk-mode page. Pressing the **Exit** button will do the same thing as **Cancel**.

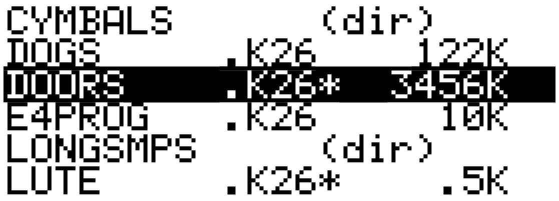
Total

The total size of all the files in the directory is indicated at the bottom left of the file display above the soft buttons. This total represents only the disk space used by the files in the directory being viewed. The K2600 includes a free space utility that indicates how much space is being used on the current disk. Also, there is a List utility that can be used to calculate the size of all files within a selected directory subtree. These functions are described in the section called *Disk Utilities* on page 13-60.

Quick Scrolling to Subdirectories

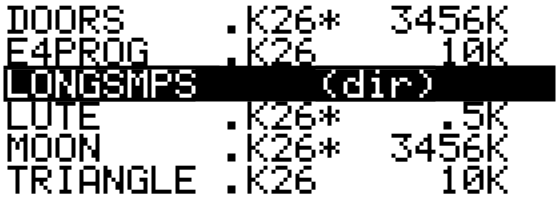
It is sometimes difficult to locate a subdirectory entry in the file list for the current directory, if there are many files in the current directory. To make this easier, individually pressing either the **Left** or **Right** cursor buttons will set the file index to the previous or next directory (respectively) in the current directory list. The index will wrap around the beginning or end of the list, so that repeated presses of either cursor button will cycle through all of the subdirectories. If you have many subdirectories, you can scroll through them all very quickly using this method.

For example, given the following file list display:



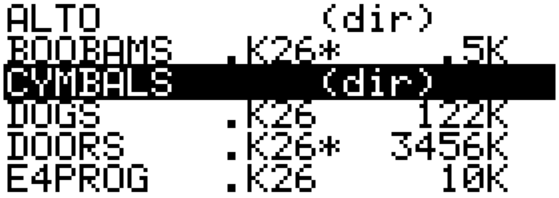
```
CYMBALS       (dir)
DOGS     .K26    122K
DOORS    .K26*  3456K
E4PROG   .K26     10K
LONGSMPS      (dir)
LUTE     .K26*    .5K
```

Pressing the **Right** cursor takes you two entries further to the next directory:



```
DOORS    .K26*  3456K
E4PROG   .K26     10K
LONGSMPS     (dir)
LUTE     .K26*    .5K
MOON     .K26*  3456K
TRIANGLE .K26     10K
```

or, pressing the **Left** cursor takes you two entries back to the previous directory.



```
ALTO         (dir)
BOOBAMS  .K26*    .5K
CYMBALS      (dir)
DOGS     .K26    122K
DOORS    .K26*  3456K
E4PROG   .K26     10K
```

Creating Directories

As stated above, you can create directories for organizing your K2600 files, whether you are using SCSI or floppy disks. You can create directories on any disk formatted by a K2600, K2500, or K2000.

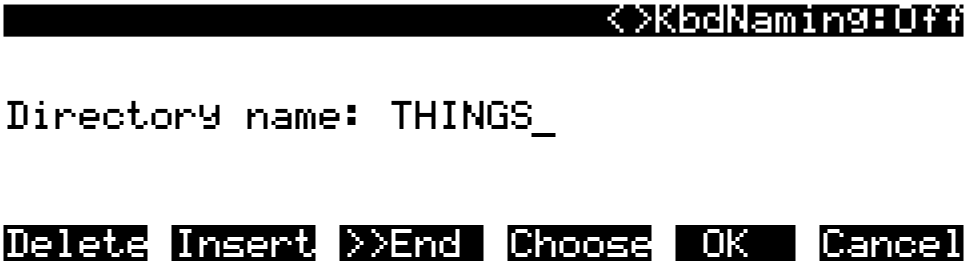
Directories appear in the normal file list with the indicator (dir) to the right of the directory name.

There are two ways to create new directories.

- Press the **NewDir** button while on the Disk-mode page
- Press the **NewDir** button during the Save dialog.

Creating a Directory From the Disk Mode Page

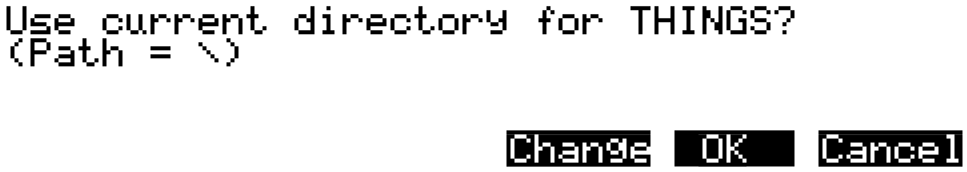
When you press **NewDir**, the K2600 prompts you for the directory name:



```
                                <>KbdNaming:Off

Directory name: THINGS_

Delete  Insert  >>End  Choose  OK  Cancel
```

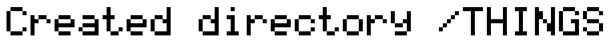
Pressing **>>End** will take the cursor to the last character in the name. The **Choose** button allows you to grab a filename from the current disk (see the discussion of file-name grabbing, in *More Features of the Save Dialog* on page 13-28). Otherwise, the name will default to either **NEWFILE** after a powerup, or the name will be that of the most recent file saved or loaded. Once you choose a name to start with (or the default), you can edit the name using the **Left** and **Right** cursor buttons, the **Delete** and **Insert** soft buttons, and the **>>End** soft button. You can also use keyboard naming, as described on page 5-5.

After you have chosen the directory name and pressed **OK**, you have the choice of where (in what directory) to put the new directory you are creating.



```
Use current directory for THINGS?
(Path = \)

                      Change   OK    Cancel
```

Pressing **OK** will select the default path, which is the current directory. Pressing **Change** will allow you to view the disk, traversing its directories, until you find the one in which you want to create the new directory. In this case, pressing **OK** creates a directory called **THINGS** in the root directory.



```
Created directory /THINGS
```

The display shows that the K2600 has created the directory, then the Disk-mode page reappears.

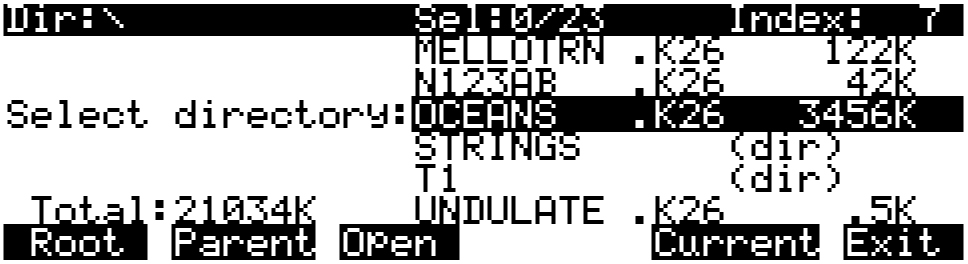
Creating a Directory in the Save Dialog

As a convenience when saving files to a directory, you can press **Save** from the Disk-mode page and then press **NewDir** in the Save dialog. You'll get the same prompts as when you create a directory from the Disk-mode page. When you press OK, the display shows that the K2600 has created the directory, then the Save dialog reappears.

When you create a directory from within the Save dialog, the K2600 resets the current directory to the directory you just created.

The Directory Selection Dialog

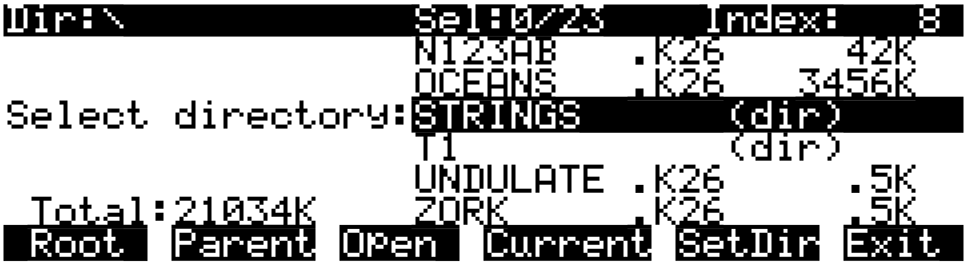
When making a new directory, as well as in many of the disk functions, you will be presented with the opportunity to change the current directory, or the default directory for a disk operation. A good example is the "Use current directory? "prompt that you see when you create a directory. If you press **Change**, you will see a slightly modified file list dialog, through which you can select any directory on the disk. The display looks like this:



```
Dir:\                Sel:0/23     Index:   7
                     MELLOTRN .K26     122K
                     N123AB   .K26      42K
Select directory:OCEANS   .K26    3456K
                     STRINGS        (dir)
                     T1             (dir)
 Total:21034K        UNDULATE .K26      .5K
 Root   Parent  Open       Current  Exit
```

When you enter this dialog, you will be in whatever directory was displayed as the default. From here you can go into other directories by using the soft buttons Root, Parent, and Open. Notice that there is no **Select** button. This is because the purpose of this dialog is to choose a single directory as opposed to selecting multiple files. However, the Root, Parent, and Open buttons function exactly as described above (for the file list dialog). The Sel field (on the top line) shows you how many files/directories you have selected out of the total number of files/directories in the current directory.

If you've highlighted a directory, there is one additional soft button displayed, **SetDir**. Notice the **Current** button moves over one button to the left:



```
Dir:\                  Sel:0/23     Index:    8
                       N123AB   .K26      42K
                       OCEANS   .K26    3456K
Select directory:STRINGS         (dir)
                       T1              (dir)
                       UNDULATE .K26      .5K
 Total:21034K          ZORK     .K26      .5K
 Root   Parent  Open   Current  SetDir  Exit
```

You can use either of two soft buttons to select a directory in this dialog.

Current This selects the directory you are currently in (whose file list you are viewing), as specified in the Dir parameter on the top line of the display. For example, if you wished to select the directory STRINGS using the **Current** button, you would first press **Open** to display the contents of that directory, and then press **Current**. If you instead wanted to choose the root directory, you would simply press **Current**, since that is the directory you are viewing (notice the **Dir:** \ at the top).

SetDir This selects the directory you are scrolled to, such as STRINGS in the display above. This method is often quicker and more convenient than pressing **Open** followed by **Current**, which does the same thing. The **SetDir** soft button is present in the display only when the scrollbar highlights a directory entry.

Disk Mode Functions

Now that you are familiar with the basics of creating directories and moving around in the K2600 file system, it is time to discuss some of the features provided in the disk functions themselves.

Loading Files

The **Load** button instructs the K2600 to copy a file from the current disk to the K2600's RAM. Press the **Load** button, and a list of files stored in the currently selected device will appear. Scroll through the list of files with the Alpha Wheel or **Plus/Minus** buttons, then press **OK—**or press **Cancel** to return to the Disk-mode page.

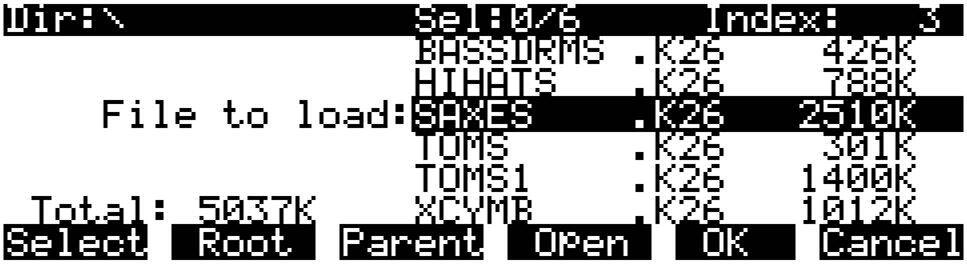
When you press **OK**, the Bank dialog will appear (as described in *Load Function Dialog* on page 13-19) and you'll be asked to select the memory bank to load the file into. Scroll through the list of banks with the Alpha Wheel or **Plus/Minus** buttons until the desired memory bank is highlighted, then press **OK**. Or press **Cancel** to back up a page and select another file to load. Once you have selected a bank to which to load, you will be asked to choose a method for loading. The method you choose determines how the objects in the file will be ordered when loaded into the bank.

Loading Individual Objects

Since files can contain over 3000 objects, it is often useful to load only a subset of the information contained in a K2600 file. Sometimes, this capability is necessary even to be able to load certain files, if the size of the file's samples or data is greater than the K2600's internal RAM size.

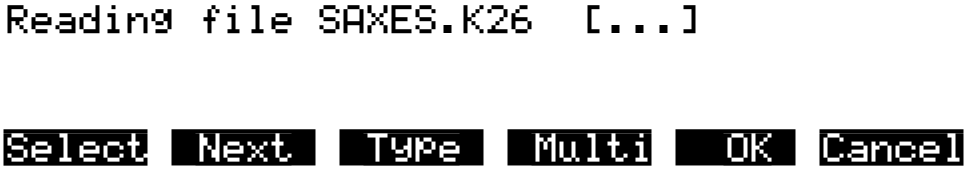
You can select individual objects or groups of objects (samples, programs, keymaps, effects, songs) for loading from within a single K2600 file.

The Load Object feature is accessible from within the Load File dialog. To activate it, scroll the file list until you have highlighted the file that you wish to load objects from:



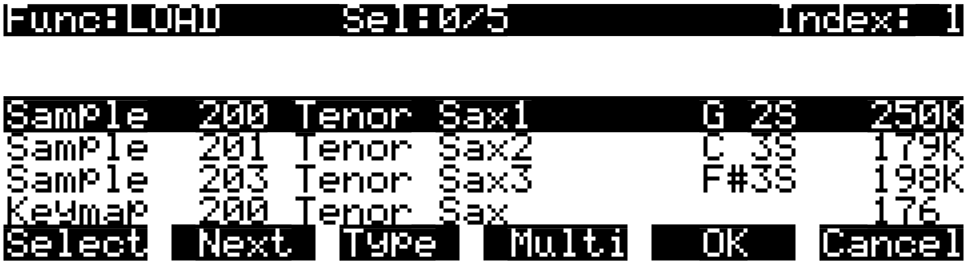
```
Dir:\                  Sel:0/6       Index:    3
                       BASSDRMS .K26     426K
                       HIHATS   .K26     788K
     File to load:     SAXES    .K26    2510K
                       TOMS     .K26     301K
                       TOMS1    .K26    1400K
  Total: 5037K         XCYMB    .K26    1012K
Select  Root    Parent  Open    OK      Cancel
```

Press **Open** to begin the Load Object dialog. (Note: The file must be in **.KRZ**, **.K25**, or **.K26** format in order to load individual objects from it.) The K2600 then scans the file contents in order to present a list of all of the objects in the file. Sometimes this procedure can take a few moments, depending on how many objects are in the file. During this time, you will see the following display:



```
Reading file SAXES.K26  [...]



Select  Next    Type    Multi   OK      Cancel
```

The soft buttons in the above display do not become active until the process of scanning is finished. When this happens, the K2600 will display a list of the file's objects, in the exact order that they are stored in the file:

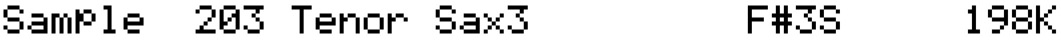


```
Func:LOAD       Sel:0/5                 Index: 1


Sample   200 Tenor Sax1          G 2S     250K
Sample   201 Tenor Sax2          C 3S     179K
Sample   203 Tenor Sax3          F#3S     198K
Keymap   200 Tenor Sax                    176
Select  Next    Type    Multi   OK      Cancel
```

The objects in the list are usually grouped by type (sample, program, keymap, etc.). The list can be scrolled using the Alpha wheel or the **Up** or **Down** cursors. The **Chan/Bank** buttons on the front panel can be used for fast scrolling. The list will jump by five entries at a time, moving the entry on the bottom line to the top line.

Note: *When scrolling through large numbers of objects (more than 100), the K2600 may sometimes pause for a few seconds if it needs to get more information from the disk file. When this happens, some gyrating dots will briefly appear in place of the Index value on the top line of the display.*

Each line in the scrollable list represents one object, and displays the object's type, ID, name, and size. Samples have additional information: the sample's root key and a stereo sample indicator:



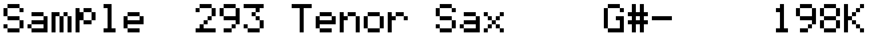
Object Type	ID	Name	Sample Info	Size
Sample	203	Tenor Sax3	F#3S	198K

The ID numbers are the same numbers that were used to reference the objects when the file was last saved by the K2600. These numbers will usually be different after the objects are loaded, depending upon the bank (for example, 200...299) and mode that is specified for loading. There is more information on these modes in the section called *Load Function Dialog* on page 13-19.

The Size field is interpreted differently for samples and nonsamples. For nonsamples (songs, programs, etc.), it shows the number of bytes used by the object in the file, and hence the amount of program memory that the object will occupy in the K2600. For samples, the size field shows the size of the all sample data associated with the object, and is displayed in kilobytes (K).

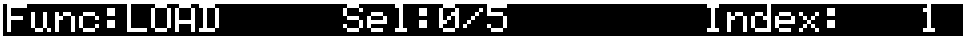
For samples, the letter S after the root key indicates a stereo sample.

Due to display space constraints, if the sample's root key happens to be in the lowest MIDI octave range (that is, C -1 through B -1), it will be displayed in a truncated form. For example, if a sample's root key was set to G$^{\#}$-1, the display would read:



```
Sample  293 Tenor Sax    G#-    198K
```

The status line at the top of the display specifies the function being performed, the number of selected objects in the list followed by the total number of objects in the file, and the current list index:



```
Func:LOAD       Sel:0/5          Index:    1
```

As with the file list, entering in a number from the alphanumeric buttonpad will jump to the indexed entry, and typing in a large number like 9999 will go to the end of the list.

The soft buttons on this page are used for multiple selection of the objects in the list as well as for moving around the list when there are many items selected or listed. This same dialog is also used for many other functions in the K2600, namely for saving selected objects to disk and for several object utility functions that are described later.

Here is a brief description of each button's function, followed by a detailed explanation of its operation.

Select Select or deselect an object.

Next Jump to the next selected object.

Type Jump to the next object of a different type.

Multi Go to the Multiple Object Selector page.

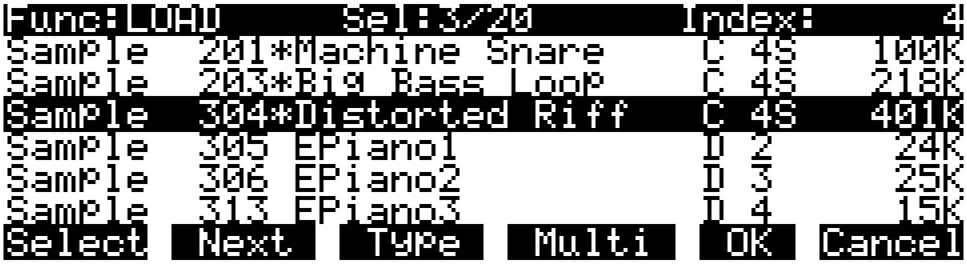
OK Tell K2600 to proceed to load the selected objects.

Cancel Exit back to the File List Dialog.

Press the **Select** button to choose the highlighted object for loading. An asterisk (*) is placed in between the object name and the object ID for any items that are selected. Deselect a selected object by pressing **Select** again. The asterisk will disappear. The easiest way to choose objects for loading is to scroll the list and individually press **Select** on each object you want to load.

If you only want to select one object for loading, you need not select it with the **Select** button. Instead, pressing **OK** implicitly selects the highlighted object if there are no other objects selected. If there are objects selected, however, then the highlighted object will not be loaded unless it is selected.

This page shows three samples selected for loading (IDs 201, 203, and 304):



```
Func:LOAD       Sel:3/20       Index:      4
Sample  201*Machine Snare     C 4S     100K
Sample  203*Big Bass Loop     C 4S     218K
Sample  304*Distorted Riff    C 4S     401K
Sample  305 EPiano1           D 2       24K
Sample  306 EPiano2           D 3       25K
Sample  313 EPiano3           D 4       15K
```

The **Next** button will cause the index into the list to jump to the next selected object, forward in the list. When the end of the list is reached, the search will wrap around from the beginning. If you have more than one object selected, then if you repeatedly press Next you can easily cycle through all selected items. If there are no items selected, then this button doesn't do anything.

The **Type** button jumps to the next object of a different type from the one that is currently highlighted. This is a convenient way to find a particular type of object in the list. If you want to skip over the samples and the keymaps in an object list and jump right to the programs, press **Type** about two or three times, stopping when you notice that a program is highlighted.

Press the **Multi** button to enter the Multiple Object Selector (described on page 13-36). This powerful utility can be used to control the selection or deselection of many objects, cross-referenced by object types and ranges of ID numbers.

When you are all done selecting objects to load, press **OK**. As stated above, if only one object is to be loaded, it is implicitly selected if it is the currently highlighted object and there are no other selected objects in the list.

Cancel returns to the file list dialog, highlighting the file you just opened. You can load the entire file after pressing **Cancel** by pressing **OK** when you return to the file list.

Shortcuts when Loading Objects

Select All/Deselect All

Selecting or deselecting all of the objects at once can be done with the same double-presses as described for the file list dialog, namely:

- **Left/Righ**t cursor double-press: Select All Objects
- **Up/Down** cursor double-press: Clear All Selections

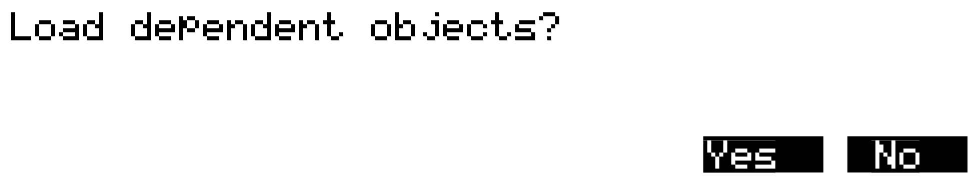
If you want to load most but not all of the items from a file (for example, if there happens to be a Master table in the file that you don't want to load), it may be fastest to first select all objects using the **Left/Right** double-press, and then manually deselect any unwanted items.

Viewing the Name Table

The name table is an object that appears in files that were created using the **Names** button in the "Save dependent objects?" dialog (see *The Name Table* on page 13-31). This object contains a list of dependent objects needed by the other objects in the file at the time the file was saved. There is more information about this later on, however it is worth mentioning here that a highlighted name table object's contents can be viewed by pressing either one of the **Left** or **Right** cursor buttons.

Loading Dependent Objects

When you press **OK** after selecting one or more objects, the K2600 will ask the following question:



Load dependent objects?

This dialog appears because one or more of the selected objects might have dependents associated with them in the file. (Remember, dependents are those objects needed by other objects; samples are dependents of keymaps, effects and keymaps are dependents of programs, and so on.) When this dialog appears, it does not necessarily mean that there really are dependents of the selected objects. The K2600 will not know whether there are dependent objects in the file until it begins to read in the selected objects, and determines what their dependents are.

Answering **Yes** to the question tells the K2600 to also load the dependents. You may wish to answer **No** if, for example, you are simply loading a program or a keymap as a template for use with other objects. You can also manually select only some of an object's dependents, and then answer **No** to "Load dependent objects?" to prevent other unwanted dependents from being loaded.

To summarize, it is not necessary to select any of the dependents of an object if you plan on loading all of the dependents. As an example, for a file containing dozens of programs, keymaps, and samples, you may choose to highlight a certain program and press **OK**, and answer **Yes** to the “Load dependent objects?” question. The K2600 will do the rest, by only loading the samples and keymaps that are needed by the selected program.

Similarly, if you selected certain *keymaps* from a file, and then answered **Yes** to “Load dependent objects,” the K2600 would figure out exactly what samples need to be loaded as dependents of the selected keymaps.

Auditioning Samples from a Disk File

Often when working with files that contain samples it is helpful to be able to hear what the samples sound like before loading all or part of the file. It is possible to audition samples in the file, from within the Load Object dialog.

To audition a sample, first scroll to the sample that you wish to hear. Then, press either the **Left** or **Right** cursor. The K2600 will load the sample (or 1 second of it if it’s longer than a second). The audition starts from the very beginning of the sample data (note that if the first second of data is silence then you won’t hear very much when the sample is auditioned). If the loop points fall within the first second of the stored sample data, they will be loaded as well. The K2600 display will blink after the completion of loading the sample audition data. When the sample segment has been loaded, it can then be played back at its root key as well as transposed up and down the keyboard.

Once a sample has been auditioned, it remains active across the keyboard until another sample is auditioned. The audition function ends when either **OK** or **Cancel** are pressed.

There must be sufficient sample RAM in the K2600 to load one second of the sound for auditioning. This amount varies according to the sample rate of the sample, but for most samples this will be less than 100K bytes. If the auditioned sample does not play, check that there is enough free sample memory in the K2600. It is also possible to see the following error if the K2600 object RAM is full or very near full:

```
Not enough memory to audition
```

Loading Objects from Floppy Disk Files

Individual objects can be loaded from K2600 floppy disk files, with the caveat that some samples in a multiple-floppy disk file cannot be auditioned because they reside on more than one disk. These samples will still appear in the object list, but their size will be in parentheses meaning they cannot be loaded (either explicitly or as dependent objects) and cannot be auditioned.

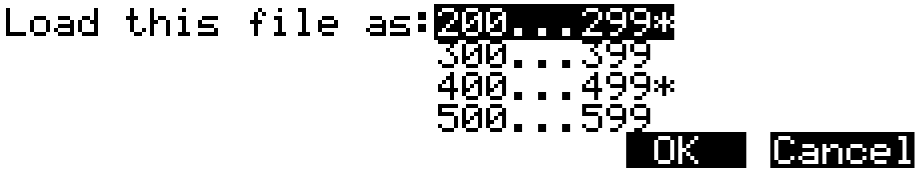
Load Function Dialog

Bank Status Indicator

After you have chosen what you wish to load, you are presented with a dialog allowing you to determine what bank will be used to load the file’s data. The bank-status indicator (an asterisk) indicates whether a bank contains objects.

If an asterisk is present after the bank number (for example, 400...499*), it means that there are objects in the bank, whether they are RAM or ROM objects. ROM objects are in the Zeros, 100s, and 700s bank, unless you have an optional ROM block installed. In that case ROM objects are also stored in the 800s, and/or 900s bank. If there is no asterisk on the line for a bank, it means the bank is empty.

In the following dialog, there are user objects in the 200s and 400s bank, and possibly also in other banks that become visible when the selection is scrolled.

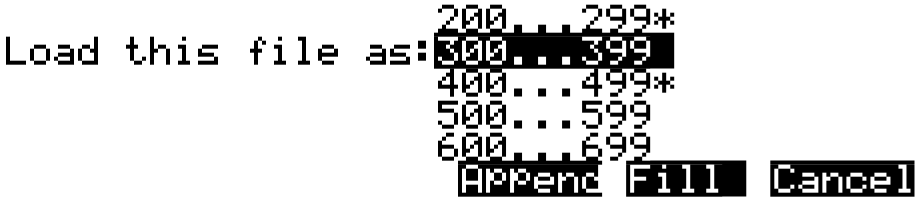


```
Load this file as:200...299*
                  300...399
                  400...499*
                  500...599
                         OK    Cancel
```

This indicator makes it easier to find an empty bank to use for loading, if needed.

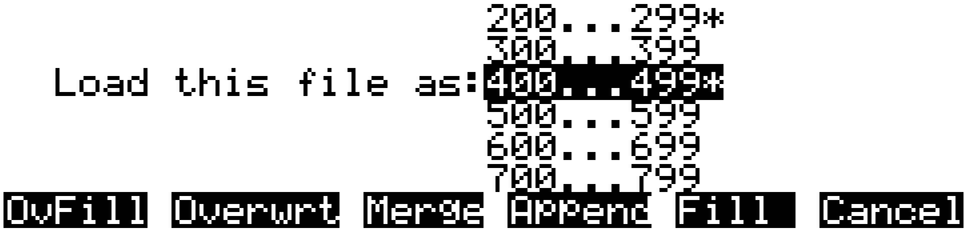
Loading Methods

Once you have pressed **OK** to decide on what bank to use, you will see this dialog if the bank is empty:



```
                  200...299*
Load this file as:300...399
                  400...499*
                  500...599
                  600...699
                   Append Fill   Cancel
```

You will see the following dialog if the bank contains any objects (in RAM or ROM):



```
                  200...299*
                  300...399
Load this file as:400...499*
                  500...599
                  600...699
                  700...799
OvFill Overwrt Merge Append Fill   Cancel
```

The soft buttons control the mode for loading and renumbering of objects from the file. Here's how they work:

OvFill First deletes all RAM objects in the selected bank, and then loads objects using consecutive numbering.

Overwrt First deletes all RAM objects in the selected bank, and then loads objects using the object ID numbers stored in the file.

Merge Preserve the object ID numbers stored in the file for the objects to be loaded, overwrite objects already in memory if necessary.

Append Try to use the object ID numbers stored in the file for the objects to be loaded. If an ID number is already in use, increment the ID number until a free slot is found.

Fill Ignore the object ID numbers stored in the file. Try to use consecutive numbering from the beginning of the selected bank. If an ID number is already in use, increment the ID number until a free slot is found.

Cancel Cancel the mode selection, and go back to choosing a bank. Scrolling to a different bank value will have the same effect as Cancel.

Typically, you will just want to use the **Fill** method. **Append**, **Merge**, and **Overwrt** try to preserve the numbers stored with the objects in the file, but this should only really be necessary if you depend on program numbers or effect numbers to be at a certain MIDI program change number. **OvFill** is like **Fill** except the selected bank (or Everything) is cleared out before loading.

Overwrt and **OvFill** operate in different ways after a selected bank has been filled up for a given object type (for example, after you have loaded more than 100 programs into a bank). **Overwrt** will continue to preserve the objectIDs stored in the file, and will individually overwrite objects in the bank following the just filled bank. **OvFill** does not overwrite past the end of the selected bank; it instead skips over object IDs that are in use, loading only into unused IDs. Because of this difference, it can sometimes be faster to load a file using **OvFill** rather than **Overwrt**. However, this applies only if the objects to be loaded would extend past the end of a selected bank.

Note that when loading into a specific bank (as opposed to loading as "Everything"), the object IDs in the file are used as follows: The "bank" digit is ignored, and the remainder of the number is used when the K2600 rebanks the object ID into the bank that you specify. For example, if you save Program 453 into a file, and load it back into the 300s bank, the K2600 will use the number **53** when deciding upon a new object ID. If the 300s bank was previously empty, and the load mode is Append, then the program will end up with ID 353.

For loading as "Everything," the ID number for an object stored in a file is taken literally, and not rebanked (except if **Fill** or **OvFill** mode is chosen, in which case the K2600 will use ID numbers starting from 200).

The following example shows how each different loading methods affect how four programs load into a bank that already contains programs.

Example: Starting with the following objects already stored in the K2600 internal RAM:

Program ID	Program Name
200	Acoustic Piano 2
204	Bright Piano
205	Tin Ear Piano
210	Chorused Piano
211	Electric Piano 2

Suppose you were to load a file containing the following objects into the 200s bank:

Program ID	Program Name
405	Blues Organ
406	Gospel Organ
409	Cheezoid Organ
410	Internal Organ

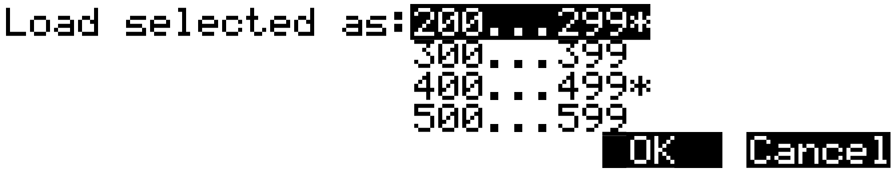
The following table shows the IDs that each program end up with when you load the organs (with IDs in the 400s) into the 200s bank, which contains the pianos. Note that in Merge mode, Organs 405 and 410 replace Pianos 205 and 210.

Original Program ID	Program Name	Program IDs After Loading				
		OvFill	Overwrt	Merge	Append	Fill
200	Acoustic Piano 2	Deleted	Deleted	200	200	200
204	Bright Piano	Deleted	Deleted	204	204	204
205	Tin Ear Piano	Deleted	Deleted	Deleted	205	205
210	Chorused Piano	Deleted	Deleted	Deleted	210	210
211	Electric Piano 2	Deleted	Deleted	211	211	211
405	Blues Organ	200	205	205	206	201
406	Gospel Organ	201	206	206	207	202
409	Cheezoid Organ	202	209	209	209	203
410	Internal Organ	203	210	210	212	206

Selecting Multiple Files to Load

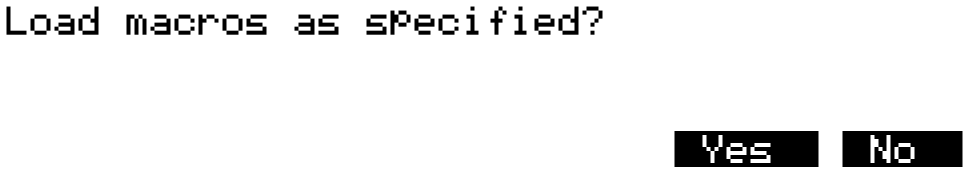
As stated previously, you can select multiple files for loading into the K2600 from within a single directory, in one operation. This is done from the file list dialog with the **Select** button.

After you have selected one or more files in this way, you will still choose a bank and mode to be used for the load process, just as with loading a single file. However, the dialog prompt will say Load selected as:



```
Load selected as:200...299*
                 300...399
                 400...499*
                 500...599
                         OK    Cancel
```

If you selected any macro files (**.MAC** extension) from the directory, then once you have select the mode for loading, you will see the question:



```
Load macros as specified?



                         Yes    No
```

The answer to this question instructs the K2600 that any macro files will have their macro entries loaded according to the bank and mode:

Yes specified in the macro entry.

No currently specified for this multiple file load. In other words, whatever you select for Bank and Mode will override the instructions for each entry in the macro.

At this point the files will begin to load. When all the files have been successfully loaded or the load process has been aborted, the K2600 returns to the Disk-mode page.

If there are any errors encountered during a multiple file load, such as running out of object RAM, you will be asked once if you wish to abort the load. In some cases, you may wish to continue loading. If you continue (and don’t abort), the only way to abort will be to use a special procedure described in the next paragraph.

Aborting a Multiple File Load

There is a way to abort the process of loading multiple files. Aborting can only be done “in between” files that are being loaded, and not during the load of any one file (short of powering off or soft-resetting the machine by pressing **+/-**, **0**, and **Clear** simultaneously, but this is not recommended!).

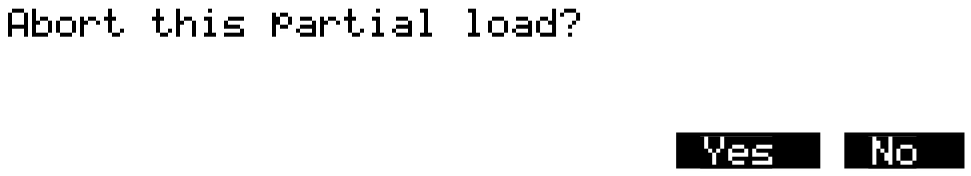
Aborting a multiple file load is done by pressing and holding down either of the Plus (+) or the Minus (-) buttons that are located just below the Alpha wheel. This should be done at least one-half second before you anticipate the current file to finish loading, or else the K2600 will not sense that you wish to abort the load.

You will see the following question after the current file being loaded is completed:

Abort the load?

It may be a good idea to practice using this method of aborting a multiple file load, so that when the time comes that you accidentally select 100 files, you will remember how to abort the process. This same method (of holding the **Plus** or **Minus** buttons down) is also used to abort the Backup feature and the macro file load feature.

If you run out of object or sample RAM, you will have one opportunity to abort the load as explained above. However, if you continue from that point you may end up seeing the same error message “Memory is full” for each file that you had selected. This can be a rather tedious process, however it is still possible to abort out of this by holding down the **Plus** or **Minus** button simultaneously while pressing **Yes** when you see the following question:



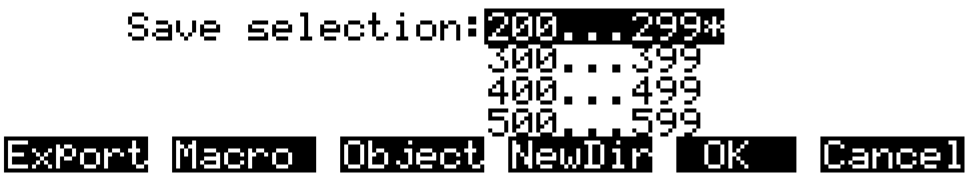
Abort this partial load?

More Load Function Enhancements

There are more features having to do with the Load function that are described later on in this guide, such as loading macro files and loading AIFF files.

Saving Files

The **Save** button starts the process of saving from the K2600 to the currently selected device. When you press the **Save** soft button on the Disk-mode page you will see the bank dialog:

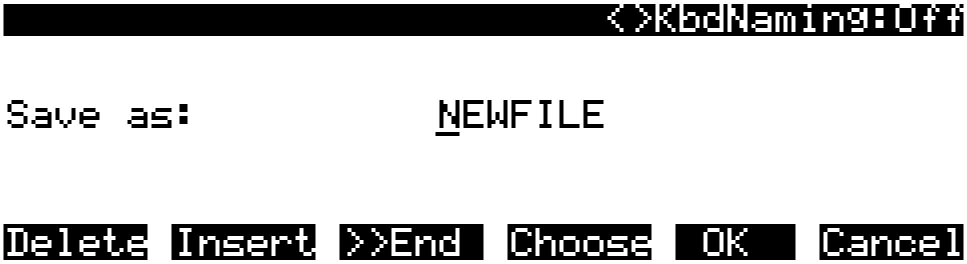


```
Save selection:200...299*
               300...399
               400...499
               500...599
```

The **Macro** soft button will be present only if macro recording is turned on. (See page 13-43 for more information on macros.)

You can save an entire bank of objects, or by pressing the **Object** soft button, select individual objects to be saved. If you choose to save using the bank method, all RAM objects within that bank will be saved. (You cannot save ROM objects. If you wish to save a ROM object, such as a program, you must first save it internally as a RAM program.) If any objects within the selected bank have dependent RAM objects that exist in a different bank, you will be asked if you want to save dependent objects. See page 13-29 for more on saving dependent objects.

Use one of the data-entry methods to select a bank to be saved. If you press the **Cancel** soft button, you'll return to the Disk-mode page. After you've selected the bank, press **OK**. The following page will appear:



<>KbdNaming:Off

Save as: NEWFILE

You can now name the file according to the naming procedures outlined in Chapter 5. You can enter up to eight characters. When you've entered a name, press **OK** to save the file as shown in the display, or press **Cancel** to return to the file dialog. When the file is saved, the K2600 adds an extension (**.K26**) to the filename. This enables the K2600 to recognize it as a Kurzweil file when it examines the disk's directory.

Saving Master and Everything Files

Among your choices in the Bank dialog are Master files and Everything files. Master files consist primarily of the items on the Master-mode page and the three MIDI-mode pages. They also include information like marked pages, view settings, and MIDI channel and program assignment. In fact, saving Master files (or dumping them via SysEx) is a good way to configure your K2600 (or another K2600) to your performance or sequencing needs. For example, you might save different Master files with every sequence you create using an external sequencer. Then, when you load the Master file, you would have all the correct programs assigned to the appropriate MIDI channels.

Everything files consist of the Master file parameters and every other RAM object. Saving an Everything file will literally save everything in RAM, including samples, into a single file. These can be quite large, so if you don't have a hard disk, be sure to have a few preformatted floppy disks handy.

Split Files

When you're saving memory banks, it's not uncommon to create files larger than 720K or even 1.4M—especially when you're saving RAM samples. If you're saving to floppy disks, the K2600 will create split files stored on multiple disks. A little advance preparation is necessary for this.

If the K2600 fills a floppy disk before it finishes saving a file, it will prompt you to insert a second disk in the floppy drive. This disk must be preformatted; the K2600 will not format disks in the middle of a save operation. When the second disk is inserted, the K2600 will continue to save. This process will be repeated until the entire file is saved. The K2600 marks these disks internally with a number that indicates the sequence in which they were saved.

When you’re loading these split files, they must be inserted in the same sequence as they were saved. When the K2600 has loaded the contents of the first disk, it will prompt you to insert the second disk, and so on. Make sure to label your disks as soon as you save to them, so you’ll keep them in the correct order.

Soft Buttons in the Save Selection Dialog

The meaning of the soft buttons in the “Save selection” dialog is as follows:

Export Save a sample or a song in an exported file format (that is, AIFF, WAVE, MIDI Type 0 or Type 1). This feature is described in *Importing and Exporting Data Using Standard File Formats* on page 13-71

Macro Save entries from the current macro table as a macro file (**.MAC**). This soft button is displayed only if macro file recording is on.

Object Save selected objects from the K2600’s RAM.

NewDir Create a new directory on the current disk, and return to this dialog afterwards. This is described previously in *Creating Directories* on page 13-12.

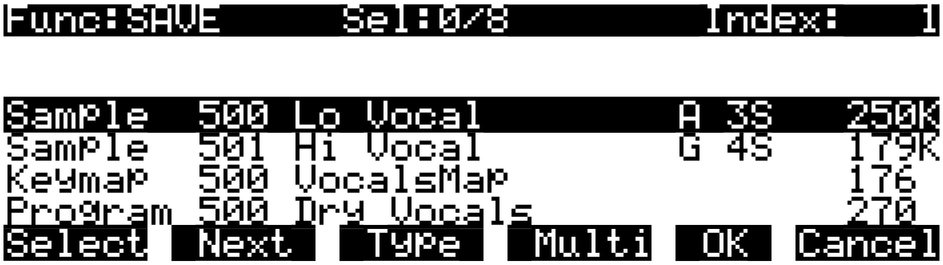
OK Save all the objects from the highlighted bank (for example, 200...299), and optionally also save dependent objects.

Cancel Exit from the Save function.

Export (page 13-71**)**, **Macro (**page 13-43**)**, and **NewDir (**page 13-12) are all explained elsewhere in this guide. This section will describe the process of saving K2600 objects into K2600 format disk files.

Saving Individual Objects

You can select any group of objects in the K2600’s RAM for saving into a single file.To save individual objects, from the above dialog, press **Object**. The K2600 will display a scrollable list of all the objects in RAM, very similar to the display for the Load Object feature (described previously):



```
Func:SAVE      Sel:0/8          Index:    1

Sample  500 Lo Vocal            A 3S   250K
Sample  501 Hi Vocal            G 4S   179K
Keymap  500 VocalsMap                  176
Program 500 Dry Vocals                 270
Select  Next  Type  Multi  OK  Cancel
```

The procedures for saving objects are essentially the same as the procedures described on page 13-15 for loading objects.

Shortcuts when Saving Objects

Select All/Deselect All

Selecting or deselecting all of the objects at once can be done with the following double-presses (two front-panel buttons simultaneously pressed):

- **Left/Righ**t cursor double-press: Select All Objects
- **Up/Down** cursor double-press: Clear All Selections

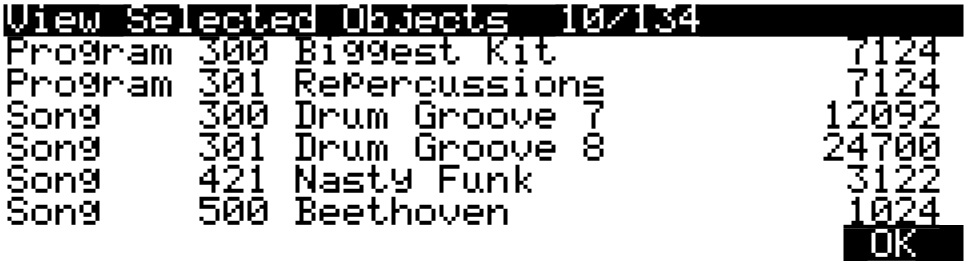
If you want to save most but not all of the items from a file (for example, if there are some songs in RAM that you don't want to be saved in the file), it may be fastest to first select all objects using the **Left**/**Right** double-press, and then manually deselect any unwanted items.

Viewing Selected Objects

When there are lots of objects selected, but they are scattered in the objects list, it can be helpful to be able to view a list of only the currently selected objects. Do this by double-pressing the **Chan/Bank** buttons.

- Double-press of **Chan/Bank** buttons: View Selected Objects

For example, if there were 10 objects selected, and you pressed both **Chan/Bank** buttons simultaneously, the K2600 would show a list similar to this:



```
View Selected Objects  10/134
Program 300 Biggest Kit           7124
Program 301 Repercussions         7124
Song    300 Drum Groove 7        12092
Song    301 Drum Groove 8        24700
Song    421 Nasty Funk            3122
Song    500 Beethoven             1024
                                   OK
```

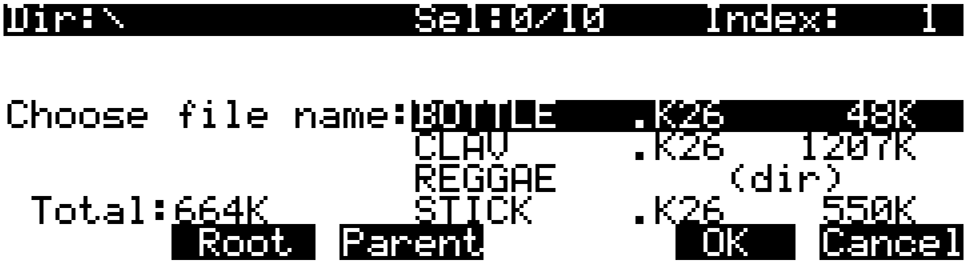
The top line shows 10 objects selected out of the 134 that are currently in RAM. If the number of selected objects is larger than the 6 objects that fit on one page (as in this example), the list can be scrolled to view all of the information.

Note that this feature is not available in the Load Object dialog.

More Features of the Save Dialog

The Choose File Name Function

When entering in a filename for saving, there is a **Choose** soft button. When **Choose** is pressed from the file naming dialog, the K2600 will access the current disk directory and display the following:



```
Dir:\                    Sel:0/10     Index:    1

Choose file name:BOTTLE     .K26      48K
                 CLAV       .K26   1207K
                 REGGAE        (dir)
 Total:664K      STICK      .K26     550K
       Root    Parent           OK     Cancel
```

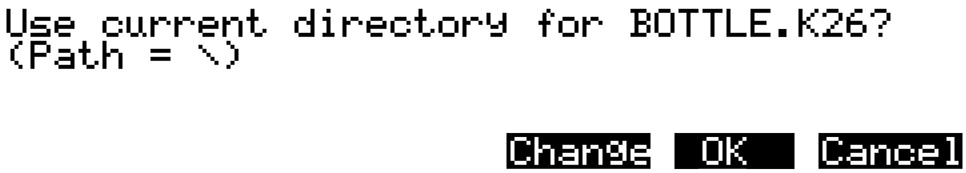
The function of this dialog is to grab the text of any filename on the current disk, and either use it as a starting point in the file naming dialog, or else use the chosen filename exactly. This helps when replacing files on the disk (where the name must exactly match the file being replaced), or adding files to the disk that have similar names or appended revision numbers. You can save time by not having to enter the entire filename on the K2600's alphanumeric buttonpad.

The **Open** soft button is visible in the "Choose file name" dialog only when a subdirectory is highlighted.

Traversing directories from the Choose function does not change the current default directory.

Selecting the Directory to use for Saving a File

After you specify the filename when saving any file, select where to put it: by default it goes in the current directory, but you can specify any other directory on the current disk:



```
Use current directory for BOTTLE.K26?
(Path = \)


                     Change     OK     Cancel
```

Pressing OK will accept the default path (the current directory), which in this example is the root directory (represented by the backslash character). Pressing **Change** will allow you to view the disk, traversing its directories, until you find the one in which you want to save the file. If you choose a different directory from the default, it will become the new default directory. For more information on selecting a directory, see *The Directory Selection Dialog* on page 13-13.

Saving Any File sets the File Index

After saving a file, you can go to any disk function (such as Load), and the just saved file will be automatically highlighted. This makes it easy to find a file that you have just saved, in case you want to delete it, add it to a macro, move it to a different directory, open it (if it is a **.K26** or a **.MAC** file), etc.

Auditioning Objects in RAM

When deciding which individual objects to save, it can be difficult to know if you are selecting the correct ones. This is especially true if many objects have similar or identical names, or if the names of the objects are not descriptive enough to know what they are. The K2600 has a feature that allows auditioning of samples, keymaps, programs, and songs right from the Save Object dialog (as well as all of the other object utility dialogs that are discussed later). To activate this feature, scroll to an object of an appropriate type to be auditioned, and press either the **Left** or **Right** cursor button. The display will blink, and the objects can now be heard as follows:

Samples play at their root key, as well as transposed across the keyboard. Stereo samples will play in stereo. Auditioning samples in this way is similar to listening to samples from the SampleMode page in Master mode. The samples are auditioned using a “hidden” program set up according to the parameters in Program **199 Default Program**. This default program can be customized if needed by editing and saving a new program 199.

If you audition any sample objects, the last one that you audition will become the “preview” sample the next time you go to the SampleMode page in Master mode. This can be a quick way to edit the sample without having to edit a program and a keymap.

Keymaps are reproduced accurately, and are played according to the parameters in Program **199 Default Program**. This default program in ROM is set up to have a 0% effects level (dry). Therefore, auditioning keymaps can be a very convenient way to hear them isolated from the effects.

Programs play exactly as they would if they were selected from the Program-mode page.

Songs start playing when either the **Left** or **Right** cursor button is pressed, and stop playing when either cursor is pressed while the song is playing. The most recent song that is auditioned from this page become the current song (as seen on the Song-mode page).

Setups play exactly as they would if they were selected from the Setup-mode page.

Once auditioned, the above object types remain active on the keyboard until another object is auditioned, or until **Cancel** is pressed. If a song is being auditioned, no other objects are auditioned until the song audition is stopped (by pressing one of the **Left** or **Right** cursor buttons).

Saving Dependent Objects

When you save a file, you may see a prompt as part of the Save dialog that asks you whether you want to save dependent objects. A dependent object is simply an object that’s associated with another object. The dependent object can be stored in a different memory bank—for example, a RAM sample with ID 301 that’s used in a program with ID 402, or in the same bank as the file being saved. Rather than forcing you to save dependent objects separately and to keep track of them yourself, the K2600 gives you the option of automatically saving the dependent objects as part of the file you save. When you load the file again, the dependent objects will be loaded along with the objects to which they’re attached.

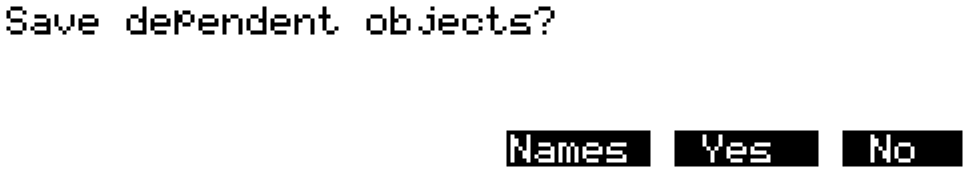
There are a few things to keep in mind regarding dependent objects. First, it’s not uncommon for RAM samples to be dependent objects, and they can take up quite a bit of memory. If you save a RAM sample as a dependent object when you save a bank of programs to disk, you may create files that won’t fit onto a 720K or 1.4M floppy disk. Since the K2600 can’t format floppy disks in the middle of a save operation, you should have spare formatted disks ready to go before you start saving. See the section called *Split Files* on page 13-25.

While the K2600 makes it easy for you to keep track of your dependent objects, you need to keep aware of what happens with dependent objects when saving to disk and reloading. Consider this example. Suppose you create 30 new programs, each of which uses a keymap containing four different RAM samples. If you save these programs to a disk file, and save dependent objects with them, you’ve created a file containing 30 programs and 120 dependent RAM samples. So far, so good. Suppose you then load that file into the 300s bank. The K2600 will load the 30 programs into the 300s bank just fine, but it will be able to load (at most) only the first 100 dependent objects to the 300s bank (each memory bank can hold a maximum of 100 objects of a given type). The remaining 20 dependent objects will be loaded into the 400s bank. If there are no objects of the same type in the 400s bank, there’s no problem. But if there are objects of the same type in the 400s bank, some or all of them will be replaced by the newly loaded dependent objects.

The easiest way to prevent this is to make sure that you don’t create more than 100 dependent objects attached to the other objects in a given memory bank. The easiest way to do *this* is to avoid creating dependent objects when possible, by saving objects with IDs in the same memory bank as the objects to which they’re related. For example, if you create a program that uses RAM samples, and you save the program with ID 201, resaving the RAM samples used by that program with IDs in the 200s will prevent dependent objects from being created for that program. If you do this, you’ll minimize the number of dependent objects you create, and you’ll be unlikely to force dependent objects to be loaded into a higher-numbered memory bank when you load files.

Once you have selected objects for saving (either individually as just described or by bank selection), the K2600 will determine if any of the items chosen to save have any dependent objects in RAM that were not chosen. For example, if you select a program to be saved and nothing else (using the Save Object feature), the program may have dependent effects, keymaps, and samples that are in RAM.   Dependent objects that are in ROM (for example, ROM samples or keymaps) do not get saved to disk.

You will see the following dialog displayed if there are any dependent objects in RAM of any objects that were selected for saving:



Save dependent objects?

Choosing **Yes** will cause any dependent objects to be saved in the file together with the selected objects. Choosing **No** means that unselected dependents will not be saved.   The **Names** button creates a new kind of object to be stored in the file, called the name table.

The Name Table

A file's name table is a list of any dependent objects that were not explicitly selected for saving in the file. Each entry in the name table contains the object type, object ID, and the name of a dependent object.

A file's name table is used by the K2600 at only one time: when the file is loaded. At that time, the K2600 will search for dependent objects that were not saved in the file originally. The search matches dependent objects by name with objects that are already in RAM, and links them to the "parent" object. The name-table data are then discarded when the file load is finished. This search feature is referred to as **Relink-by-Name**.

Relink-by-name can help you work efficiently with K2600 objects and disk files. Careful use of this feature can save you many megabytes of disk storage. It can also free up time for working on music and production instead of waiting for sample data to be resaved.

Relink-by-Name allows you to save objects and their dependent objects separately (in multiple files) and be able to link them up later on by loading the files in the correct order. This can be a very efficient way of working with the K2600's many levels of dependent objects. The most common way in which Relink-by-Name speeds up development of sounds is when making small adjustments to a program that has as its dependents a large amount of sample data. You can separate the program and sample data, so that after changing a program parameter, only a file containing the program and a name table need be resaved.

When loading a file that contains a name table, the following rules should be observed in order for correct relinking to occur.

1. *Use unique names for dependent objects at every level***.** For example, if you were going to be relinking several samples from one file with a program and a keymap from another file, each sample should have a different name. Otherwise, the dependent objects (the samples) will not get relinked properly. This will create problems such as keymap ranges that don't play as they are supposed to.

2. *The dependents to be relinked must already be loaded*. Otherwise they will not be found and relinked when the file containing the parent objects is loaded. This constraint on the order of file loading can be made easier to work with by using the macro file feature (described later). You can construct a macro file to automatically load the dependents files and the parent files in the correct order, making sure that any files containing dependents are loaded first. An alternative to loading the files with a macro would be to save the dependent and parent files in the same disk directory with similar filenames such that they will appear consecutively in the alphabetized file list. Once you have done this, it is easy to select both files for loading in the correct order.

These rules may appear complicated at first, but they will seem natural once you have worked out a few examples with your own files.

The search algorithm used for relinking dependent objects to their parent objects during loading is as follows:

The search for a dependent object (whose name matches that of an entry in the name table) begins at the beginning of the bank that is specified for loading the parent file. All possible IDs are then consecutively searched. When the last ID of the 900s bank has been searched (typically 999), the search will wrap around to ID 1 up until the end of the bank just before the specified bank. The search stops once a dependent with a matching name has been found and relinked.

For example, if a file containing a one-layer program is loaded into the 400s bank, and the file includes a name table that lists the layer's keymap by name, then the K2600 will begin to look through all possible keymap IDs starting at 400, until ID 999. The search then continues from

ID 1, stopping at ID 399. If the search does not successfully find a match, the dependent will be unresolved, and in this example the program would show a value of “Object id not found” for its Keymap parameter, where the object id is the value that was stored in the file.

The search is done in this “circular” manner so as to allow you to direct which dependent objects get relinked. This may be necessary if you end up with multiple copies of dependent objects with the same name; you can differentiate between them by loading the parent file into a specific bank that is the same bank or “before” the bank containing the objects you wish to relink to. Note that this can only be taken so far, since it would be impossible for the K2600 to differentiate between objects with the same name within the *same* bank.

The relinking process happens in the background, without any notification or error messages if items cannot be relinked.

Working with Relink-by-Name

Here are a couple of more in-depth examples that can show how Relink-by-Name works in a practical situation.

Consider that your K2600’s RAM contains the following one-layer program and also its dependent keymap and samples (the technique used in this example could well apply to any programs with any number of layers):

Program: Program 317 Steinwave Piano

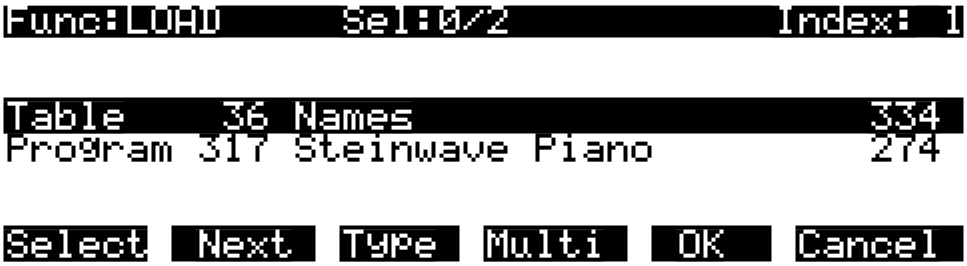
Keymap: Keymap 300 Steinwave Piano

Samples: Sample 300 StwaveG1 .... Sample 310 StwaveC7

In this case you might wish to save the samples and the keymap in one file, and the program in another file. So, from the Save Object dialog you could first select all the samples from 300-310, and Keymap 300, for saving into a file, let’s say **STWAVE1.K26**.

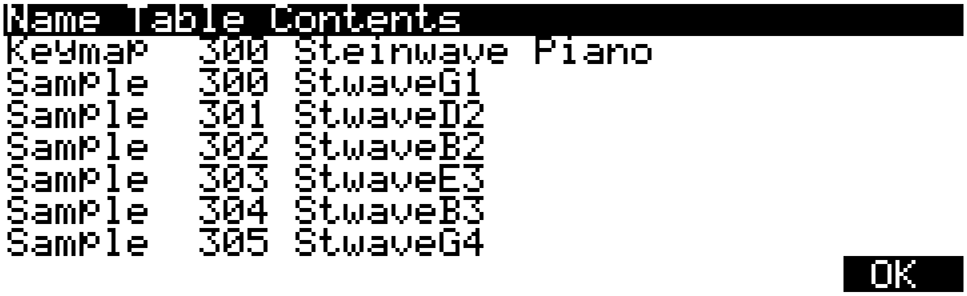
You would then return to the Save Object dialog and save just Program 317 in a separate file in the same directory, let’s say **STWAVE2.K26**…only this time, you will be asked the “Save dependent objects” question pictured above. Answer this by pressing **Names**.

After saving, the file **STWAVE2.K26** will contain two objects in it, Program 317 and a name table. You can easily verify this by going to the Load function (or any other disk function) and pressing **Open** on the file just saved (which should come up already highlighted). The display of objects for the file will look like this:



```
Func:LOAD      Sel:0/2              Index: 1

Table     36 Names                      334
Program 317 Steinwave Piano             274
```

The name table will always be the first object in the list. You can verify the exact contents of the name table by using the “View Name Table” shortcut (as described on page 13-18); make sure the name table is highlighted, and press either the **Left** or **Right** cursor button (as if you were “auditioning” the name table). You would then see the following:



```
Name Table Contents
Keymap  300 Steinwave Piano
Sample  300 StwaveG1
Sample  301 StwaveD2
Sample  302 StwaveB2
Sample  303 StwaveE3
Sample  304 StwaveB3
Sample  305 StwaveG4
```

The Name Table Contents list shows what would have been saved in the file had you answered **Yes** to “Save dependent objects?” instead of answering by pressing **Names**. More importantly, it allows you to see what objects need to be in the K2600’s RAM *before* loading this file.

The object IDs shown in the table are the same numbers that those dependent objects used at the time this file was saved. (The ID numbers are necessary in order for Relink-by-Name to function, since they are the “link” between the higher level objects and the names of the dependents.)

An important thing to notice about this particular name table is that the sample names are not needed by the K2600 for relinking purposes. In fact, the only information necessary for relinking the dependent objects of this file is the keymap object. The reason for this is that when this file containing the program is loaded, all of these dependent objects should already have been loaded, and the keymap should already be correctly linked to the samples. Although the samples’ names are redundant from the K2600’s point of view, they are included for free, so to speak, and you may find them very helpful if you ever need to know exactly what the dependents of this file were intended to be.

The Name Table Contents list is scrollable if there are more than seven objects in the name table.

Now that the two files **STWAVE1.K26** and **STWAVE2.K26** have been created using the name table, they can be reloaded and correctly relinked. The files can be loaded into any bank— they do not need to go back into the bank they were originally in—since the **STWAVE2.K26** file will search through all the banks to find the objects by name in order to relink them. In fact, if you were to immediately reload just the file containing the program (**STWAVE2.K26**), into any bank, you would find that it was automatically relinked to the correct keymap, since the keymaps and samples are currently in memory.

Furthermore, you could edit the program and create more variations of it that reference the **Steinwave Piano** keymap, add ROM layers, and/or effects if desired, and resave all of the programs (and any effects) to the same or a new file (remember to press **Names** when you are asked “Save dependent objects?”) You never have to resave the file **STWAVE1.K26** that contains the keymap and samples, if all you have done is edited the programs or added more of them. This can be a tremendous time-saver.

If the keymap and sample files are found on a CD-ROM disk, then using Relink-by-Name is not only a time-saver, but a disk-space saver as well. If you like the samples and keymaps from a CD-ROM file, there is no need to duplicate the sample data on your own writable hard disk. Instead, all you have to do is save a program file in the above manner, and then make sure the CD-ROM file is loaded first before you load the program file.

If you needed to add some sample data to the file (for example, you want to add a root to the keymap or process and reloop a sample from the CD-ROM), you can do this by explicitly selecting the new sample data and the keymap for saving along with the program and the name table. Then, the new sample would not be listed in the name table (it would be in the same file as the name table), and the keymap would be relinked to all of the samples by name instead of the program being relinked to the keymap (as before). What you put in the different files is up to you, and there is no limit to where you can break up the objects in one file or another. The main thing to be aware of are the two rules for Relink-by-name mentioned above:

1. *Files containing dependent objects must be loaded first*.
2. *Always use unique names for like objects types*. (NOTE: In cases where duplicate names exist in different banks, load the file(s) containing dependent objects, then load the file that contains the name table into the same bank or to the one just before it. This will prevent relinking conflicts.

As you will see later, you can create a macro file that will automatically load both of the files in the correct order, no matter what disks they are on or what disk directories they are in. By using macro files in this way, you can avoid having to explicitly load multiple files and remember the correct order each time.

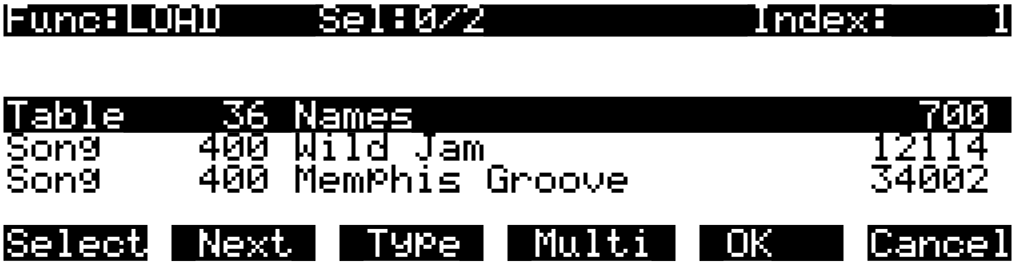
You can also use the Multiple Object Selector (see page 13-36) to help in the process of identifying dependent objects and parent objects that you want to place into separate files. For example, you could easily select all dependent keymaps and samples of any group of programs, to create a “dependents” file. Then, you could quickly select the programs and any other objects that you wanted to be relinked later on, and save them in another file.

Here is another practical example using songs (sequences). Suppose you have loaded several files into your K2600, such that you now have all your favorite instruments in RAM. Then, you make a bunch of songs using a combination of ROM programs and the RAM programs you loaded.

The dependent object structure of the songs would look something like this:

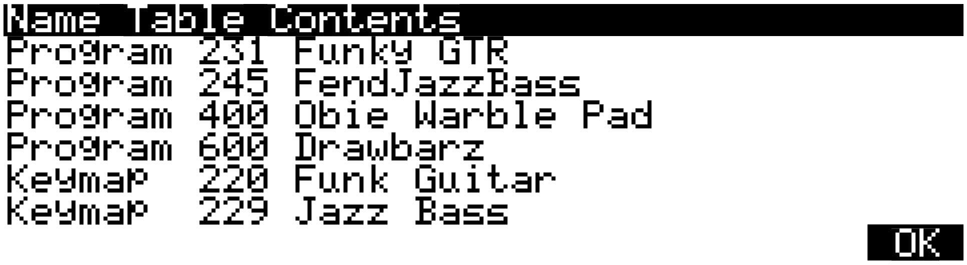
Songs	**400 Wild Jam**	**401 Memphis Groove**
Programs	**600 Drawbarz** **231 Funky GTR** **50 Studio Kit 1** (from ROM)	**245 FendJazzBass** **400 ObieWarble Pad**
Effects	ROM Effects	
Keymaps, Samples	Lots of ‘em…	

In this case you might want to save all of the songs in one file, and be able to automatically relink the dependent programs used by the song tracks. All of the programs are presumably already saved in separate files. The only file that needs to be created is one that contains all of the song objects, plus a name table. Once again, this is done by selecting the songs from the Save Object dialog, and answering Names to “Save dependent objects?” The contents of this file can then be displayed by pressing Open (as was done for the previous example).



```
Func:LOAD     Sel:0/2              Index:     1


Table      36 Names                         700
Song      400 Wild Jam                    12114
Song      400 Memphis Groove              34002

Select   Next    Type    Multi    OK    Cancel
```

Also as shown in the previous example, you can display the contents of the name table:



```
Name Table Contents
Program 231 Funky GTR
Program 245 FendJazzBass
Program 400 Obie Warble Pad
Program 600 Drawbarz
Keymap  220 Funk Guitar
Keymap  229 Jazz Bass
```

Notice that the ROM program **50 Studio Kit 1** will not be listed in the name table. Any dependent objects that are in ROM do not need to be relinked by name. ROM objects are always directly referenced by their object ID number, since they don’t get saved in any files.

Once the song file has been saved, it can be loaded at any time and correctly relinked, as long as the other files containing the necessary programs have already been loaded.

For this type of situation, where you may be working on songs always using a consistent set of programs, it is beneficial to make a macro file that can be loaded in one step to direct all of the various program files to be loaded. After that, any time you load a song file containing a name table referencing these programs, the songs should get relinked to the correct programs.

If you happen to have multiple copies of the necessary programs already loaded into different banks, you can control which bank of programs will be linked to the songs by choosing a certain bank to load the song file into. The relinked programs will be the first set encountered according to the Relink-by-Name search algorithm defined above.

Not Loading the Name Table

There may be a time that you wish to load objects from a file containing a name table, but you don’t want the K2600 to relink any dependent objects according to the name table. This can be accomplished by “Opening” the file from the Load function, and selecting any desired objects from within the file, *except* the name table. The selected objects will be loaded into the bank you specify, however the Relink-by-Name mechanism will not function.

Relink-by-Name Processing Time

Normally, the time taken to relink several dependent objects using the name search will be insignificant, relative to the time it takes to load the data from the file. However, if you are attempting to relink a very large amount of dependents by loading one file (say, 200 samples or so), there may be a noticeable wait while the K2600 searches its object database for the dependents. If this happens, it’s best to be patient.

Storing Objects in the Memory Banks

There is a separate bank of Object IDs for each object type. That is, you can store 999 programs, 999 samples, 255 songs, and so on. There are two groups of object types, based upon the number of available Object IDs. Table 13-1 shows the number of IDs and ID ranges—in ROM and in RAM—for both groups of object types.

Object Type	Total Available Object IDs	ROM ID Ranges	RAM ID Ranges
Samples Keymaps Programs Setups	999	1–99 100–199	200–299 300–399 400–499 500–599 600–699 700–799 800–899 900–999
Quick Access Banks Songs Velocity Maps Pressure Maps Intonation Tables	255	1–75	100–119 200–219 300–319 400–419 500–519 600–619 700–719 800–819 900–919

Table 13-1 Memory Banks: Object IDs Available for Different Object Types

The Multiple Object Selector Page

The Multiple Object Selector gives you several ways to select multiple objects for various operations—for example, to load all setups with IDs between 250 and 299, to save all programs in the 400s bank, including their dependent RAM keymaps (but not their dependent RAM samples), or to delete all samples whose name includes “Gazonk.”

The Multiple Object Selector is available in two places:

- In Disk mode, in the Load and Save dialogs
- In Master mode, on the Object Utility pages—Move, Copy, Name, Delete, and Dump (see page 11-13)

Each of these dialogs and utility pages has a **Multi** soft button. Pressing it takes you to the Multiple Object Selector.

Using the Multiple Object Selector: An Overview

1. In Disk mode, press **Load** or **Save**, or in Master mode, press **Object**, then press **Move**, **Copy**, **Name**, **Delete**, or **Dump**. You’ll see a list of objects that you can scroll through with the Alpha Wheel. (If you’re in Disk mode and loading objects, you’ll need to navigate through the directories and open a file before you’ll see the list and the **Multi** button.) This list of objects—conveniently called the *object list*—is what the Multiple Object Selector searches through.

2. Instead of scrolling through the object list manually and pressing **Select** for each object you want to select, simply press **Multi**. The Multiple Object Selector appears.

3. Set the value of the Select parameter, which determines the operating mode for the Multiple Object Selector.

4. Set the values of any other parameters that are visible. Different parameters are visible depending on the value of the Select parameter. This step is called setting the *selection range*. The selection range determines which objects get selected when you execute the next step.

5. Press **Set**. In most modes, this selects every object in the selection range, and returns you to the page you were on before you pressed **Multi**. Notice the asterisks between the IDs and names of the selected objects.

6. Complete the operation you started in Step 1.

Operating Modes: The Select Parameter

The Multiple Object Selector has four operating modes, which determine how the Multiple Object Selector defines the selection range within the object list. Use the Select parameter to set the operating mode. There are four values:

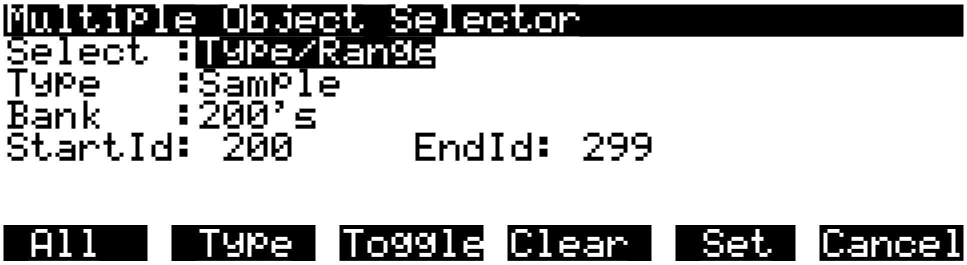
Type/Range Restricts the selection range to a particular object type (like programs or samples), and lets you specify a range of IDs (like 1–100).

Dependents Restricts the selection range to objects that are dependents of whatever object(s) you specify.

Everything No restrictions; the entire object list becomes the selection range.

Search String (**SearchStrg**) Restricts the selection range to objects whose names contain a string of characters that you specify (for example, all objects whose names include “clav”).

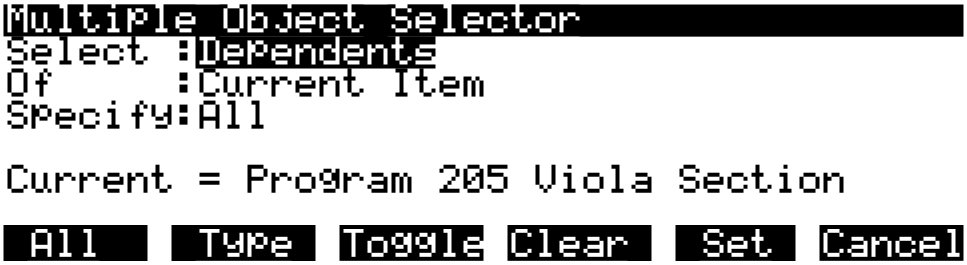
The first two operating modes in the Multiple Object Selector have other parameters associated with them. The following diagram shows what Type / Range mode looks like.



```
Multiple Object Selector
Select :Type/Range
Type   :Sample
Bank   :200's
StartId: 200     EndId: 299
```

Use this mode for operations on a particular type of object (like loading all setups, or just Setups 250–299). The Type, Bank, StartId and EndId parameters let you specify which objects to work with. See *Type/Range Mode* on page 13-41 for more information.

Change the value of the Select parameter to **Dependents** if you want to select objects based on their dependencies (for example, when you want to save 20 programs and their dependent keymaps). A different set of parameters appears.



```
Multiple Object Selector
Select :Dependents
Of     :Current Item
Specify:All

Current = Program 205 Viola Section
```

You can't use this mode with the Load function, since the K2600 can't calculate dependencies on objects that aren't already in RAM. You can use this mode with all the other functions mentioned at the beginning of the Multiple Objects Selector section.

Use the Of parameter to specify whether you want to select dependents of the current object, or dependents of previously-selected objects. In the former case (with Of set to **Current Item**), pressing **Set** selects the dependents of the object showing in the Current field (Program **205 Viola Section** in the display above—it's always the object that was highlighted on the previous page). In the latter case (with Of set to **Selected Objects**), pressing **Set** selects all the objects that you marked for selection on the previous page (all objects with asterisks between their IDs and names).

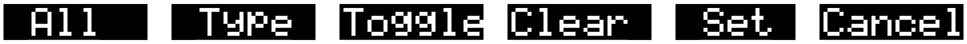
The Specify parameter determines what types of dependent objects get selected when you press **Set**. This is handy when you want to save one type of dependent object, but not another. See *Dependents Mode* on page 13-42 for more information.

If you set the Select parameter to a value of Everything or SearchStrg, all other parameters disappear. In Everything mode, the K2600 selects every item in the list on the previous page. When you press **Set**, you'll return to that page, and see every object selected.

In SearchStrg mode, the K2600 selects every object whose name contains a user-defined string of characters. In this case, when you press **Set**, the K2600 prompts you to enter a string of characters using the alphanumeric buttonpad. Enter the characters, and press **OK**. The K2600 returns you to the page you were on before you pressed **Multi**, selecting the objects whose names contain your string.

Multiple Object Selector Soft Buttons

The Multiple Object Selector has six soft buttons:



Cancel probably doesn't need explanation; it takes you back to the previous page without changing the current selection of objects. The other buttons fall into two groups.

All and Type

The first two are short-cut buttons—one for selecting all objects (just like Everything mode), and one for selecting or deselecting all objects of a particular type.

All Returns the Select parameter to **Type/Range**, if it was not already set that way. Sets Type to **All Types** and Bank to **All Banks**, and also sets StartId to **0** and EndId to **999**. This is equivalent to using Everything mode. The advantage to using the **All** button is that you can select all objects, but still be in Type/Range mode, where you can refine the selection range (for example, all objects in the 400s bank, or all programs).

Type Returns the Select parameter to **Type/Range**, if it was not already set that way. Sets the Bank parameter to **All Banks**, and also the StartId to **0** and the EndId to **999**. The Type parameter's value matches the type of the object currently indexed from the object list. For example, if you scrolled to a setup object then pressed **Multi**, pressing the **Type** soft button would set up the Type parameter to **Setup**. This is usually used to quickly select or deselect all objects of a particular type by scrolling to the first object of that type, and then pressing **Multi-> Type-> Set or Multi-> Type-> Clear**. If you don't want to include all banks in the selection range, it is easy to adjust the Bank or ID parameters to narrow the range.

Toggle, Clear, and Set

In most cases, these soft buttons select or deselect the objects in the selection range, then return you to the previous page (the page you were on when you pressed **Multi**). The exception is SearchStrg mode, in which case pressing any of these three buttons prompts you to specify the string that determines the selection range.

Toggle For each of the objects in the specified range, toggle the selection status of the object. If an object is not already selected, this selects it (an asterisk will appear between its ID and name when you return to the previous page). If an object is already selected, this deselects it (asterisk disappears).

Clear Deselects all objects in the selection range.

Set Selects all objects in the selection range.

Example: Toggle

Toggle is useful when you want to select all objects in the list *except* those that meet certain conditions. For example, you may want to free up some RAM by deleting all objects that are not being used by a song that you're working on.

1. Go to Master mode, and press the **Object** soft button, then the **Delete** soft button. You'll see a list of RAM objects.
2. Highlight the song whose dependent objects you want to keep, then press **Multi**. The Multiple Object Selector appears.
3. Set the value of the Select parameter to **Dependents**, the value of the Of parameter to **Current Item**, and the value of the Specify parameter to **All**. This specifies that you want to select all dependents of the highlighted song.
4. Press **Set**. This selects all of the song's dependent objects, and returns you to the DELETE page, showing the list of RAM objects. Note the asterisks between the IDs and names of the selected objects.
5. Press **Multi** again, and set the value of Select to **Everything** (or press **All**).
6. Press **Toggle**. This selects everything that wasn't selected, and deselects everything that was. The result is that everything *not* used by your song is selected.
7. Press **OK**. If the K2600 asks whether you're sure, press **Yes**.

Example: Clear

Suppose you're in Disk mode, and you want to save everything in RAM except programs.

1. Press the Save soft button to call up the **Save** dialog, then press the **Object** soft button.

2. Select the entire object list by pressing the **Left/Right** cursor buttons together.

3. Press **Multi**. Set the Select parameter to a value of **Type/Range**.

4. Set the value of Type to Program, and the value of Bank to All Banks.

5. Press **Clear**. The K2600 returns to the Save dialog. As you scroll through the object list, you'll notice that no programs are selected, and all objects that aren't programs *are* selected.

Example: Set

Suppose you wanted to save all keymaps and samples in the 300s bank to a single file.

1. In Disk mode, press Save, then press Object.

2. Set the Select parameter to **Type/Range**, the Type parameter to **Keymap**, and the Bank parameter to **300's**.

3. Press **Set**. This selects all the keymaps in the 300s bank.

4. Press **Multi** again, change the Type parameter to **Sample**, and press **Set** again. Now all keymaps *and* samples in the 300s bank are selected.

5. Press **OK** and continue with the Save operation.

Entering Selection Criteria in the Multiple Object Selector

This section describes the operation of the selection modes provided on the Multiple Object Selector page. These are accessed by scrolling the Select: parameter to different values, as pictured above.

Type/Range Mode

This mode lets you select objects based on their type, and on a particular range of object IDs.

Parameter	Possible Values	Function
Type	Sample, Keymap, Effect, Program, Setup, QABank, VelMap, PrsMap, IntTbl, Song, Table, All Types	Sets the desired object type. The value **All Types** will select all of the other possible types.
Bank	000s, 100s, 200s, 300s, 400s, 500s, 600s, 700s, 800s, 900s, All Banks	Sets the desired bank. Changing this parameter causes the StartId and the EndId to be set to the limits of the chosen bank (for example, a value of 300s sets the StartId to 300 and the EndId to 399). A value of All Banks sets the StartId to 0 and the EndId to 999. *The actual range used for selections when Toggle, Set, or Clear is pressed is taken from the setting of the StartId and EndId parameters*. For example, if you set the Bank to 200s and then change the StartID to 300 and the EndID to 399, the 300s bank will be selected, not the 200s. The Bank parameter is used as a quick way to set up the ID range for an entire bank, or all banks.
StartID	0–999	Sets the specific starting ID of the selection range.
EndId	0–999	Sets the specific ending ID of the selection range.

Table 13-2 Object Selection by Type / Range

It is possible to set the EndId before the StartId. If this is the case, the selection range is empty.

Dependents Mode

This mode is used to select a group of objects that are dependents of other objects. This is not available when loading objects in Disk mode.

Parameter	Possible Values	Function
Of	Current Item, Selected Items	If set to **Current Item**, selection range is confined to those objects in the object list that are dependents of the currently indexed item (**Current =**), including the currently indexed item itself. If set to **Selected Items**, then the selection range includes any objects in the object list that are dependents of any currently selected objects (those with asterisks between their IDs and names). The currently indexed item is ignored unless it is already explicitly selected.
Specify	All, All->Keymap, All->Program, Keymap->Sample, Samples Only	This parameter is used to limit which dependent objects are included in the selection range for the appropriate objects included via the Of parameter. The normal setting is **All**, which means all dependents are included. The other settings are useful primarily when separating objects into different files for reloading later using macros and Relink-by-Name. If set to **All->Keymap**, then the selection range includes all dependent objects down to the level of keymaps. That is, samples will be excluded from the selection range. If set to **All->Program**, then the selection range includes any dependent objects down to the level of programs and effects (keymaps and samples are excluded from the selection range). **Keymap->Sample** includes all keymaps and samples that are dependent objects, and nothing else. **Samples Only** includes all samples that are dependent objects, and nothing else.
Current	Type, ID, and name of the currently indexed object	Displays the object that will be used if **Current Items** is the value of the Of parameter.

Table 13-3 Object Selection by Dependents

Everything Mode

Everything includes all objects in the list. You may prefer to use the **All** button for this purpose.

Search String (SearchStrg) Mode

This selection mode will ask for a search string to be entered, as soon as you press either the **Toggle**, **Clear**, or **Set** button. The range for the selection/deselection will be any objects whose names contain the search string, ignoring upper/lower case. As soon as you press the OK button after entering a search string, the K2600 executes the toggle, clear, or set command that you specified at the beginning of the search operation. SearchStrg mode is not available when loading objects.

Working with the Multiple Object Selector

The Multiple Object Selector minimizes button presses and quickly allows you to select whatever group of items you want from the K2600's RAM. It's available for all of the related object management functions.

You may notice that the cursor positions and parameter settings are remembered whenever you exit the Multi Selector dialog, even if you exit the dialog and choose a different function. For example, if you end up doing a lot of selecting of samples, or of dependents at various levels, the parameters will stay set up the way you left them as you move from function to function (for example, from Copy to Delete to Save, etc.).

“Select Dependents” mode is very useful not just for saving dependents, but also for splitting up groups of objects for placing in different files. By using the optional settings for the Specify parameter (**All-> Keymap, All-> Program, Samples Only** etc.), you can separate the group of objects that you want to save at any level of the object tree that is necessary.

Examples of possible operations using Multiple Object Selector:

- Select all the keymaps that are dependents of a block of programs.
- Select all the samples starting from ID 398.
- Select all the objects that have “piano” in their object name.
- Select the programs, setups, and effects that are dependents of song 200.
- Select all of the keymaps and samples that are dependents of songs 400-410.

Macros

The K2600 lets you create lists of disk files called macros. The files can be located on any disk, whether SCSI-based or in the internal Floppy drive. Files from SCSI disks in Roland and Akai format can also appear in macros. Ensoniq files are not currently supported by K2600 macros.

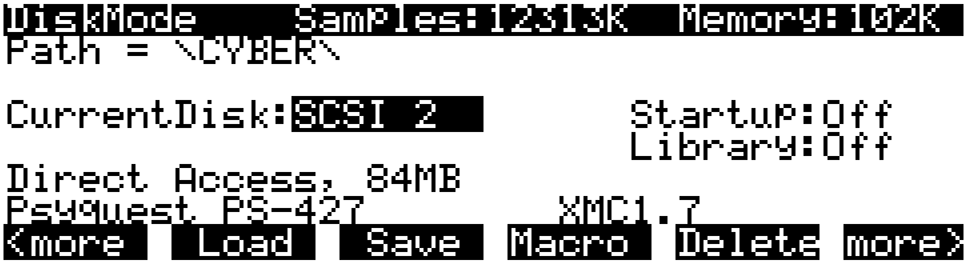
Macros are stored in a data object called a Macro table, and these can exist in two forms:

- A Macro table object in the K2600’s nonvolatile RAM. We call this the RAM Macro table.
- A disk file, containing one Macro table object. This disk file is called a macro file, and it has a **.MAC** extension (visible in the directory listing).

Macros are used primarily to load a K2600 with sound and sequence data from several files, or with selected objects within files. When a macro file is loaded, every selected object in every selected file in that macro file’s Macro table will be loaded, according to the order of the entries in the Macro table.

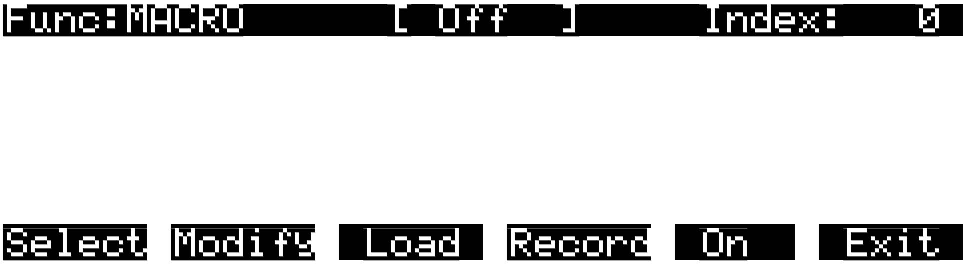
The Macro Page

There can be only one Macro table in the K2600’s memory at any time. This object is created for the first time by turning on Macro Record mode, from the MACRO page, which you reach by pressing the **Macro** soft button on the Disk-mode page, as shown below.



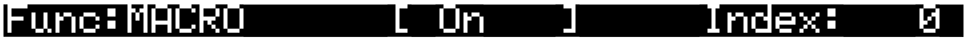
```
DiskMode     Samples:12313K   Memory:102K
Path = \CYBER\

CurrentDisk:SCSI 2         Startup:Off
                           Library:Off
Direct Access, 84MB
Psyquest PS-427         XMC1.7
```

The following page is what you will see if macro recording is Off:



```
Func:MACRO        [ Off  ]        Index:    0



Select  Modify   Load   Record   On     Exit
```

The top line displays the disk function, the current macro mode, and an index value into the Macro table.



```
Func:MACRO        [ On   ]        Index:    0
```

Macro Modes

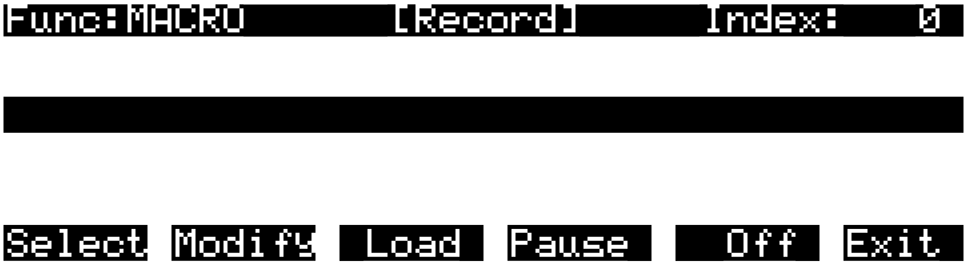
The K2600 has three macro modes: Record, Pause, and Off.

Off There is no Macro table in the K2600.

Record A Macro table exists, and the K2600 adds all file-loading operations to the Macro table.

Pause A Macro table exists, but the K2600 *does not* add file-loading operations to it.

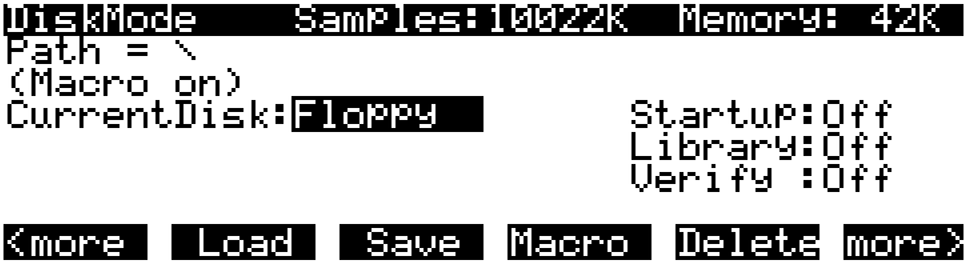
Note that whenever macro mode is Off, there are two soft buttons labeled **Record** and **On**. Pressing **On** will enable Macro Record mode, and then will return to the Disk-mode page. As an alternative, pressing **Record** will also enable Macro Record mode, but the display will remain on the MACRO page. Once you press **Record**, the soft buttons and the top line of the display will change. The display will look like this:



```
Func:MACRO        [Record]        Index:    0



Select  Modify   Load   Pause    Off    Exit
```

The new macro mode is displayed (**[Record]**), and the soft button that used to say **Record** now says **Pause**. The soft button that used to say **On** now says **Off**. Pressing **Pause** will cause the macro mode to read **[Pause]** and the **Record** soft button will reappear. You can switch between Record and Pause by pressing this button repeatedly.

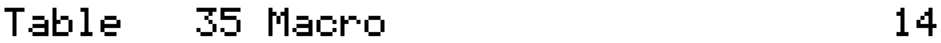
Whenever Macro Record mode is enabled, you will see the indicator **(Macro on)** near the top left of the display on the Disk-mode page:



```
DiskMode      Samples:10022K  Memory: 42K
Path = \
(Macro on)
CurrentDisk:Floppy           Startup:Off
                             Library:Off
                             Verify :Off

<more   Load    Save   Macro  Delete  more>
```

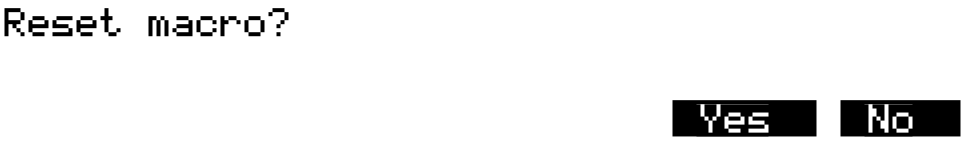
The Macro Table

When Macro Record mode is enabled after being in the Macro Off state, a new object called a Macro table gets created in the K2600's memory. In the object list for the Save dialog, the Macro table would appear as:



```
Table   35 Macro                          14
```

A Macro table can be deleted from memory only by pressing the **Off** soft button, or by performing a hard reset of the K2600. Pressing the **Off** button will display the following question:



```
Reset macro?

                               Yes    No
```

Pressing **Yes** will delete the Macro table from memory, and then will return to the Disk-mode page. The Macro mode is set to Off, and the **(Macro on)** indicator is no longer displayed on the Disk-mode page.

Pressing **No** will return to the MACRO page with no action taken. The "Reset macro?" question is displayed to allow you to change your mind about deleting the Macro table, in case you have accidentally pressed the **Off** button.

When the Macro table is first created it takes up a minimal size (14 bytes) in your nonvolatile RAM. With each new entry that is added, the Macro table will increase in size by approximately 40 to 100 bytes (or possibly more if the entry specifies an individual object list). In Macro Pause mode, the Macro table remains in memory but does not change size since file operations are not recorded. This is useful if you need to load files into the K2600 but you don't want them to be entered into the Macro table.

In Macro Record mode, the Macro table gets progressively larger with every file-loading operation. Consequently we recommend that you leave the Macro mode set to Off unless you are recording and saving macros. This will prevent the Macro table from taking up RAM.

On the other hand, if you have lots of RAM (here's another plug for the P/RAM option) you may wish to leave Macro Record mode enabled all the time. This can be useful for viewing a history of files you have recently loaded. Both the Macro mode and the Macro table are remembered between power-cycles of the K2600 via the battery-backed memory.

A macro can hold as many entries as there is space for in your K2600's nonvolatile RAM.

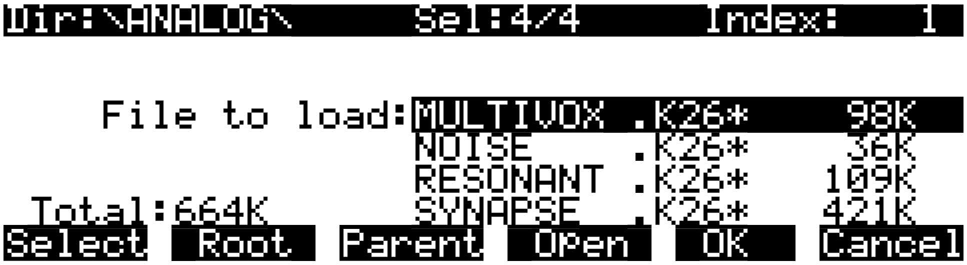
How to Make a Macro File

This section will take you through creating, saving, and loading a macro file. A simple example will be used. Afterward, you will be able to apply the example and create your own macro files.

The first step in making a macro file is to turn on Macro Record mode (from the MACRO Page, press **On**, if you have not already done so).

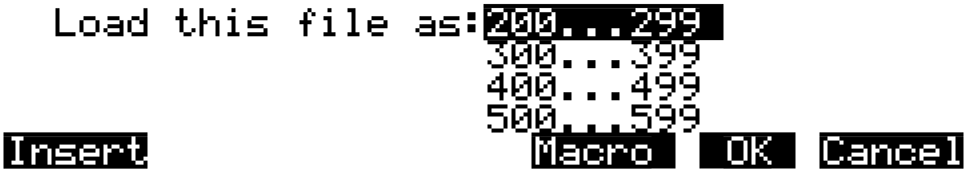
Creating the Macro

Suppose you have the following four files on your disk (on SCSI ID 5, in the directory \ANALOG\\) that contain analog-style synthesizer programs, and you would like to have one macro file that will load them all:



```
Dir:\ANALOG\        Sel:4/4      Index:    1

    File to load:MULTIVOX .K26*    98K
                 NOISE    .K26*    36K
                 RESONANT .K26*   109K
  Total:664K     SYNAPSE  .K26*   421K
Select   Root   Parent   Open   OK   Cancel
```

Using multiple selection, you can select all four files, as shown (you can also open each file and select one or more objects in that file; when you load the macro, only those objects get loaded). When you press **OK** you will see the usual Load dialog, but with the extra soft buttons **Macro** and **Insert**:



```
  Load this file as:200...299
                    300...399
                    400...499
                    500...599
Insert                 Macro   OK   Cancel
```

The extra soft buttons are available only in Macro Record mode. First, select the bank that you want, as usual. Press **OK** means to load all of the selected files into the K2600, *and* add all of the files to the Macro table. If you're just creating a macro file, and don't need to load any files at the moment, press **Macro**, which adds the files to the Macro table, but doesn't load them into the K2600.

When you add files to a Macro table, they get added at the *end* of the Macro table by default. **Insert** is for inserting file entries at any point in a Macro table. See *Macro Insert* on page 13-57 for more information.

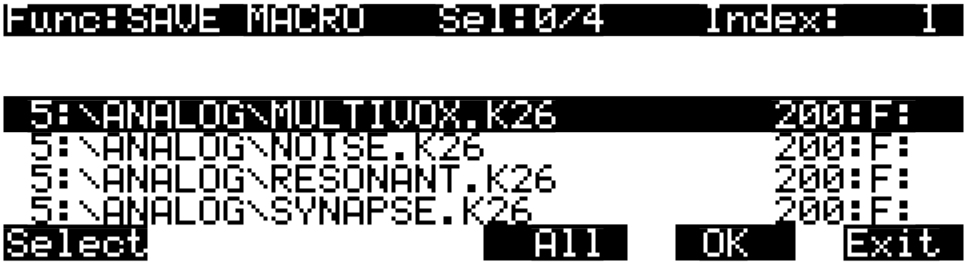
Once you have pressed either **Macro** or **OK**, the loading-mode buttons appear (**OvFill**, **Overwrt**, **Merge**, **Append**, and **Fill**). Choose a mode based on what you want to happen *when you load the macro*, because the mode gets saved as part of the Macro table. You should do this because the bank you select for the Macro table may be empty now, but it might not be when you load the macro. You need to set the mode accordingly. If preserving the IDs of the loaded objects is important, you should use Merge or Overwrt. If IDs aren't as important as preserving the objects already in RAM, use Fill or Append.

For the sake of this example, let's choose the 200s bank and Fill mode. When you press **Fill**, the K2600 executes the **OK** or **Macro** command you entered earlier. If you had pressed **Macro**, the K2600 would add the selected objects to the Macro table. If you had pressed **OK**, the K2600 would add the selected objects to the Macro table, and load them into the selected bank as well.

You have now created a macro. If you go to the MACRO page (from the Disk-mode page, press **Macro**), you'll see the files listed in the K2600's current Macro table.

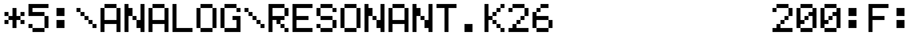
Saving the Macro File

At this point you have a Macro table with several entries in it, but you don't have a macro file until you save the current Macro table. From the Disk-mode page, press **Save**, then press **Macro**. You'll see the following dialog:



```
Func:SAVE MACRO   Sel:0/4     Index:    1

 5:\ANALOG\MULTIVOX.K26         200:F:
 5:\ANALOG\NOISE.K26            200:F:
 5:\ANALOG\RESONANT.K26         200:F:
 5:\ANALOG\SYNAPSE.K26          200:F:
Select             All    OK     Exit
```

This is called the Save Macro page. The soft buttons on this page control which Macro table entries (macro entries) will get saved to the Macro table in the macro file. You can select multiple entries using the cursor buttons and the **Select** soft button. Selected entries have an asterisk on the first character of the display line, such as this:



```
*5:\ANALOG\RESONANT.K26         200:F:
```

You can use the following double-presses to select and deselect all entries in the list:

- **Left/Right** cursor double-press: Select all macro entries
- **Up/Down** cursor double-press: Clear (deselect) all currently-selected macro entries

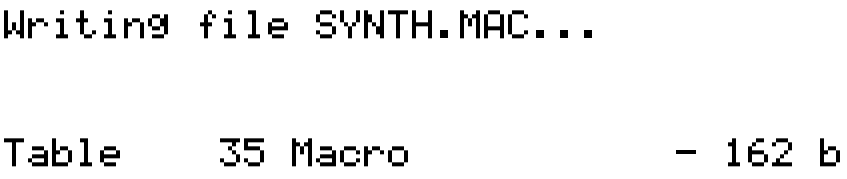
The top line indicates how many total macro entries are in the current Macro table, and how many are selected.

Pressing **OK** saves the selected macro entries to be saved in the file. If there are no entries selected when you press **OK**, the K2600 saves only the highlighted entry.

You might think that there isn't much use for a macro file with only one entry in it, however it can be a convenient link to an often-used file. For example, you could create a macro file called **\PERC.MAC** in the root directory on the disk where you store your programs. This macro file could load a single object, namely the file **\MYSOUNDS\PERC\TECHNO\PERC.K26**. When you wanted to load **PERC.K26**, you could simply load the macro PERC.MAC, instead of having to open three directories to select the file for loading. This gives you quick access to the file while preserving the organization of your program files.

If you know that you want to save all of the entries into the macro file (as we do for this example,) just press **All**. The K2600 will go through the standard file saving dialog in which you choose a filename and select a directory to save the file in.

Let's save the file as **\ANALOG\SYNTH.MAC**. Macro files are automatically saved with the **.MAC** extension. While the file is being saved, you'll see something like this:



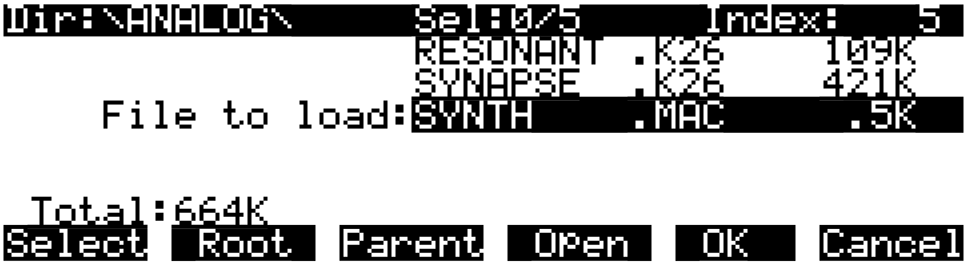
```
Writing file SYNTH.MAC...


Table    35 Macro           - 162 b
```

Loading the Macro File

So far, so good. We have created a macro in memory and saved it to the disk, in the same directory as the files that are listed in the macro.

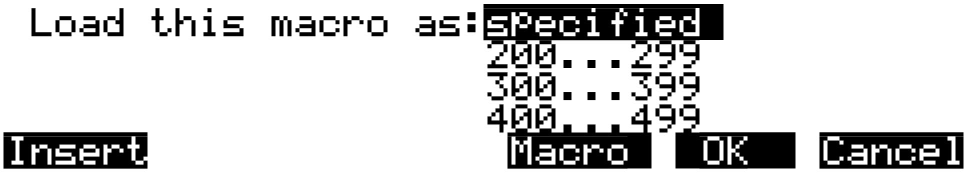
This example loads files from within a single directory on a single disk, to keep things simple. But you can create macros that load files from any number of directories on any number of disks.

Now, let's go to the Load page and try to load the macro file, which will load all the files in the macro file's Macro table. When you return to the Load page, the file list highlights the macro file that was just saved (as it would after any type of file that you save):



```
Dir:\ANALOG\      Sel:0/5      Index:    5
                  RESONANT .K26    109K
                  SYNAPSE  .K26    421K
    File to load:SYNTH    .MAC      .5K

 Total:664K
Select  Root  Parent  Open   OK   Cancel
```

Press **OK** to load **SYNTH.MAC**. Now the display reads:



```
 Load this macro as:specified
                    200...299
                    300...399
                    400...499
Insert                   Macro   OK   Cancel
```

There are a couple of things to notice here. The first is a new choice in the bank list: **specified**. “Load this macro as specified” means load all the files in the macro following the exact instructions for the bank and load mode for each file. In our example, all the files were specified to be loaded into the 200s bank using Fill mode. If this is acceptable at the time you want to load the macro, you can just press **OK**. Otherwise, you can override the bank and mode settings for the entire macro by choosing a different bank and mode before pressing **OK** (this is called *rebanking* the macro).

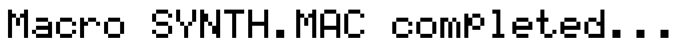
The other thing noticeable about the above display is that the **Macro** and **Insert** buttons are still available, because the Macro mode is still Record. This means that when you load the macro, it gets added to the RAM Macro table. But it’s not the *filename* (**SYNTH.MAC)** that gets added, as is the case with **.K26** files. Instead, *every macro entry in the file’s Macro table* gets loaded into the RAM Macro table. This is a convenient way to edit a macro file or combine elements of several macro files into one macro. See *Editing Macros* on page 13-54.

In our example, since we’re in Macro Record mode, pressing **Macro** would add the macro file’s entries into the RAM Macro table. Pressing **OK** would add the macro file’s entries into the RAM Macro table, *and* would load the corresponding files into RAM. Since we added the files listed in the macro to the RAM Macro table when we recorded the macro, the RAM Macro table now includes a duplicate set of entries.

Whatever method of loading you choose (that is, specified in the macro or overriding the macro), the K2600 locates each file in the macro in the exact order in which the entries are listed. If the files are on different disks in your disk system, you can observe your various disks as they’re selected in turn and files are loaded from them.

If the K2600 cannot locate one of the files, you’ll see a “Not Found” error message. If a disk cannot be accessed (for example, if the SCSI ID stored with the macro entries in this example is no longer the current SCSI ID of the disk), you will see the message “Problem mounting disk,” to which you must press **OK**. If Confirm is set to **On** on the Master-mode page, loading will stop on the first error message, giving you a chance to cancel the operation or keep going. If you answer **Yes** at this point, the operation will continue, even if the K2600 encounters subsequent errors. If you run into a lot of errors due to loading an out-of-date macro file, the macro process can be discontinued using a special procedure described later in the section called *Aborting a Macro Load*. (page 13-59).

When the macro is done loading, you’ll see this display:

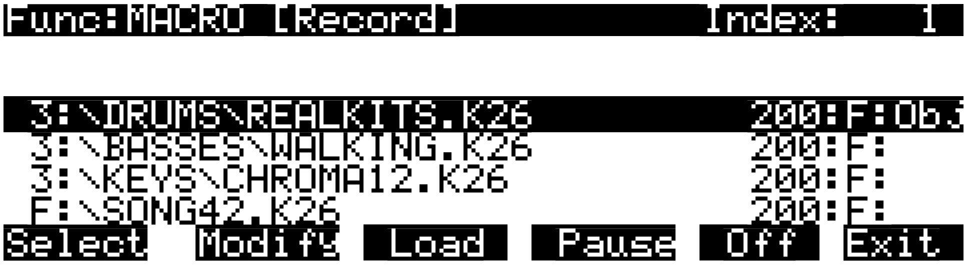


Macro SYNTH.MAC completed...

The K2600 returns to the Disk-mode page. You should now be able to go to the Save dialog, or Program mode, and verify that all of the objects from the files of the macro are now in the K2600’s memory.

Macro Entries

Each file-loading operation that is recorded into the Macro table is called a macro entry.Each macro entry stores information about how a disk file should be loaded. Each entry is displayed as a single item in a scrollable list on the MACRO page, with various fields indicating the parameters of the entry. The following diagram shows how the MACRO page might look once four files have been recorded into the RAM Macro table:



```
Func:MACRO [Record]               Index:    1

 3:\DRUMS\REALKITS.K26            200:F:Obj
 3:\BASSES\WALKING.K26            200:F:
 3:\KEYS\CHROMA12.K26             200:F:
 F:\SONG42.K26                    200:F:
```

Table 13-4 describes the information stored in a macro entry. The highlighted entry in the diagram above indicates a file on a disk with a SCSI ID of 3. This file is stored in a directory called **DRUMS**. The filename is **REALKITS.K26**. The K2600 will load it into the 200s bank, using Fill mode. "Obj" means that individual objects within the file are selected for loading, and *only* those objects will be loaded from this file.

Disk ID	Specifies the disk from which to load. There are ten possible values: The numbers **0–7** represent SCSI 0 through SCSI 7. The letter **F** represents the Floppy drive. The letter **U** means Unspecified disk ID (see page 13-51). The letter **L** means the Library disk (see page 13-52).
Directory path/ filename	This is the directory path and the filename of the file on the disk to be loaded by this macro entry. The display can show up to 28 characters of this name, although the RAM Macro table stores the entire path and filename.
Bank	The bank where you want to load the file. This will have a value from 0 through 900 (by 100s), or the letter E for Everything (all banks).
Mode	The mode specified for loading the file. The following one-letter codes are shown in the display: **O** means Overwrite mode (Overwrt) **V** means OverFill mode (OvFill) **M** means Merge mode **A** means Append mode **F** means Fill mode This field is to the right of the bank field, after a colon.
Object indicator	When this field is empty, the entire file gets loaded. If "Obj" appears in this field, the K2600 loads only those objects that were selected for loading during the recording of the macro. If the entry represents a file on a third-party SCSI-format disk such as Akai or Roland, this field indicates the manufacturer: **Aka** means Akai format **Rol** means Roland format Ensoniq format disks are not currently supported in macros.

Table 13-4 Information Stored in Macro Entries

Using the Bank and Mode Fields

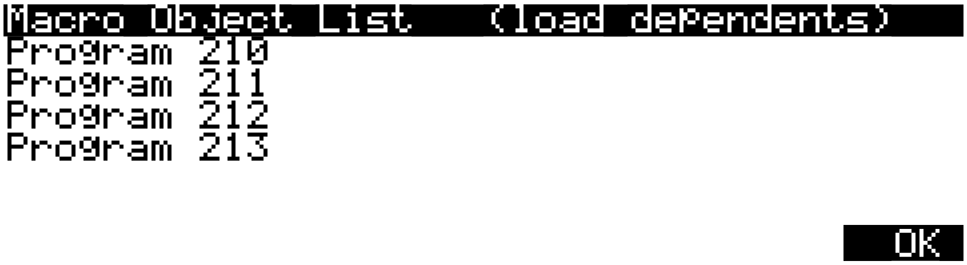
The bank and mode fields in a macro entry are relevant only if a macro file is loaded as specified. This means that each file listed in the macro will be loaded exactly as the bank and mode fields of the macro entry dictate. You can override the macro entry's settings during the Load operation, and specify a different bank and mode for the entire macro (you can't specify overrides for individual entries).

Depending on you working style, you may not have much use for the settings of the bank and mode fields. If you're always loading things into different banks depending on the situation, you'll probably change the bank and mode each time you load anyway, so the bank and mode fields in the macro entries won't matter much to you.

The bank and mode fields are more important when you want to use macros to fill the K2600's memory banks a particular way, and you want to be able to do it automatically and consistently.

Viewing the Object List for a Macro Entry

If a macro entry contains an object list, it can be examined by scrolling the Macro table display until the item with the Obj indicator is highlighted, and then by pressing either the **Left** or **Right** cursor button on the front panel. You will see a display that looks like this:



```
Macro Object List   (load dependents)
Program 210
Program 211
Program 212
Program 213



                                   OK
```

The Macro Object list, a scrollable list, shows what objects are to be loaded from the file specified in the currently indexed macro entry. You will not see the names of the objects in this display, because they are not stored in the Macro table. The objects are referenced only by object type and object ID. The (load dependents) indicator in the top line means that the macro process should also load all dependents of the objects in this list.

If you need to know the names of objects in a macro entry object list, it is possible to begin a disk function (such as Load), find the file specified in the macro entry, press **Open** to display the file's objects, and look up what the objects are, using the information in this display.

Unspecified Disk ID

When you record a macro entry to the RAM Macro table during a load operation, that entry's disk ID matches the ID of the disk from which you loaded the file. So, for example, if your hard disk has a SCSI ID of 5, all the files you load from that disk will show 5 in the Disk ID field. This information gets saved when you save the RAM Macro table to a macro file. The next time you load that macro file, the K2600 looks for a disk with SCSI ID 5 and expects to load files from that disk.

That's good, because you probably don't change the SCSI ID on your hard disk very often. But what if you have a removable-media drive (like a Zip drive or Jaz drive) with SCSI ID 5, and you pack a disk full of programs and samples to give to another band member who has a removable-media drive with a different SCSI ID? Does one of you have to keep changing SCSI IDs to exchange files?

Fortunately, no, because the K2600 lets you create macro files with entries that don’t specify a Disk ID. When the K2600 is loading a macro file, and encounters an entry with an unspecified Disk ID, it expects to find the files on the same disk as the macro file. So you can create a macro file by loading a bunch of files from your Zip drive, editing every entry so that it has an unspecified Disk ID, and saving the RAM Macro table. Then you can put the macro file and all the files in its Macro table on a Zip disk, and hand the disk to a friend. Your friend can then load the macro into her K2600, and it will load the files no matter where she has her Zip drive’s SCSI ID set.

This feature is likely to be most useful for people who distribute K2600 sound files and macro files on removable media. By leaving the Disk ID unspecified, they can be sure that anyone can load the files without regard for SCSI ID.

Of course, there are other uses. If you use a single hard disk with your K2600, or if you regularly work with macro files that load files from the same disk, then if you set all the macro file’s Disk IDs to unspecified you won’t have to edit your macro files if you happen to change your disk’s SCSI ID.

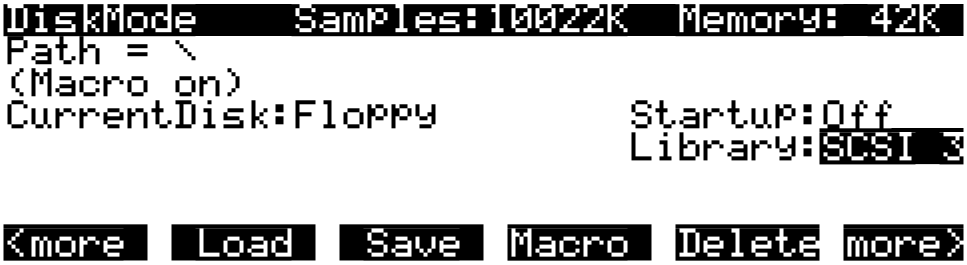
The Modify Macro page is where you can set unspecified Disk IDs (in Disk mode, press the **Macro** soft button, then the **Modify** soft button). See *Editing Macros* on page 13-54 for more information.

The Library Disk

If a macro entry is set to the library Disk ID, it means that the file to be loaded should be found on the disk at the SCSI ID set by the Library parameter on the Disk-mode page. This designation is similar in purpose to the unspecified Disk ID, because it is a way to avoid needing to hard-wire the SCSI ID in advance. The library disk ID is intended to be used in macro files that reside on removable media such as a floppy disk, whose macro entries are supposed to load particular files on a specific sample CD.

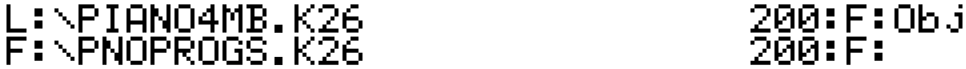
The main purpose of this feature is so that macro files can be distributed on floppy disks containing programs that link up with sample files from CD-ROMs. You don’t want to have to copy a CD-ROM file to one of your hard disks in order to make new programs that use that CD-ROM file’s sample and keymap data. If you set up your CD-ROM drive as the library disk, then create a macro file with entries that specify Library (L) for the Disk ID, the K2600 will be able to load files directly from the CD-ROM, then load the programs from the floppy disk, link the programs with the sample and keymaps using the Relink-by-Name feature.

Here’s an example. In the following diagram, the Library parameter has been set to **SCSI 3**.



```
DiskMode      Samples:10022K   Memory: 42K
Path = \
(Macro on)
CurrentDisk:Floppy             Startup:Off
                               Library:SCSI 3
```

We’ll assume that SCSI 3 is your CD-ROM drive, and that you have a floppy disk that contains a macro file with the following entries:



```
L:\PIANO4MB.K26                  200:F:Obj
F:\PNOPROGS.K26                  200:F:
```

When you load this macro file, the K2600 first looks for the file **\PIANO4MB.K26** on your CD-ROM drive, because your CD-ROM's SCSI ID is 3 and you've set the Library parameter on the Disk-mode page to **3**. As long as you have the right disk in the CD-ROM drive, the K2600 loads the file, and then loads **\PNOPROGS.K26** from the floppy disk.

If the K2600 executes the above macro, but the Library parameter has not been set (is set to **Off**) an error message

```
Library has not been set
```

will be displayed when the macro process attempts to load **\PIANO4MB.K26**.

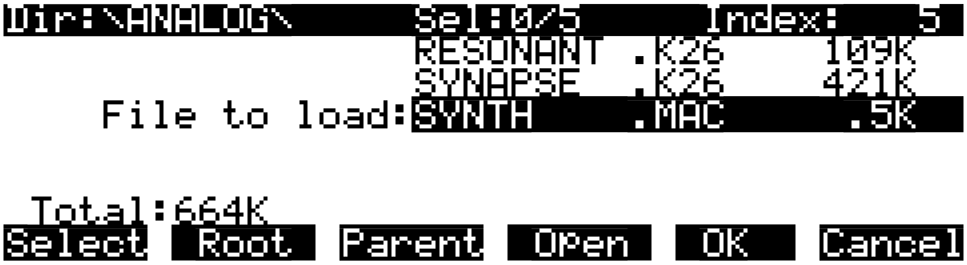
The K2600 remembers the setting of the Library parameter across power-cycles, via the battery backed memory. Set up the Library disk once, and it stays that way until you change it.

The Modify Macro page is where you can edit macro entries to use the Library disk (in Disk mode, press the **Macro** soft button, then the **Modify** soft button). See *Editing Macros* on page 13-54 for more information.

Although the Unspecified and Library Disk IDs are meant to be used with distributable media such as CD-ROMs and floppy disks, these features will work with any supported disks.

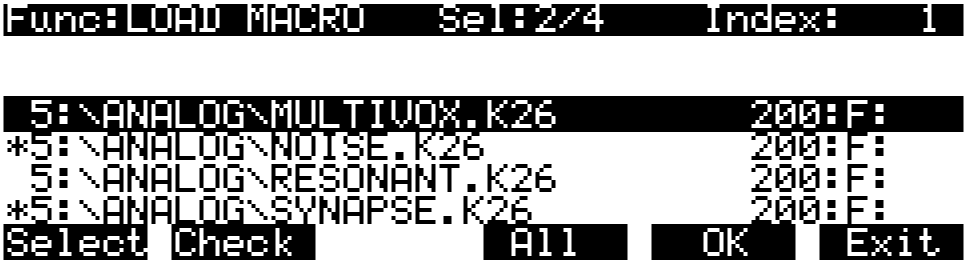
Loading Selected Entries from a Macro File

It is possible to examine the contents of a macro file from any disk function page—the same way you would open a **.K26** file to check out what objects are stored in it—by highlighting the **.MAC** file and pressing **Open**:



```
Dir:\ANALOG\       Sel:0/5      Index:    5
                   RESONANT .K26    109K
                   SYNAPSE  .K26    421K
    File to load:SYNTH    .MAC      .5K


 Total:664K
```

The K2600 will need to read the macro file into a temporary area of internal memory, which means there needs to be enough free RAM to accommodate it. Press **Open** during the Load function to enter a dialog similar to the MACRO page and the Save Macro page. This is called LOAD MACRO page:



```
Func:LOAD MACRO   Sel:2/4     Index:    1


 5:\ANALOG\MULTIVOX.K26          200:F:
*5:\ANALOG\NOISE.K26             200:F:
 5:\ANALOG\RESONANT.K26          200:F:
*5:\ANALOG\SYNAPSE.K26           200:F:
```

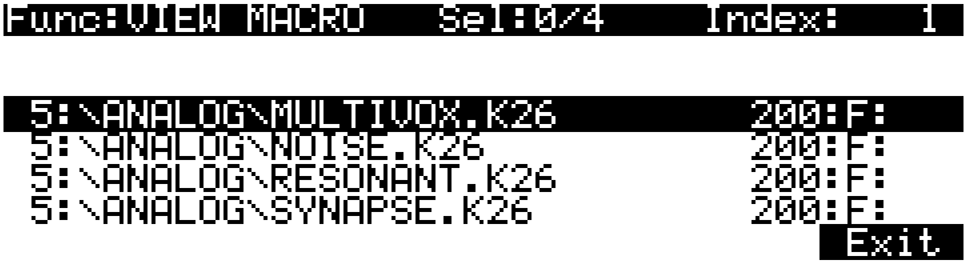
From the LOAD MACRO page, you can select one or more individual macro entries for loading, instead of having to load the entire macro. This is done using the **Select** soft button, identical to the method of saving macro entries. In fact, this dialog operates identically to the Save Macro dialog, with one exception, the **Check** soft button.

The **Check** button will cross-check all of the macro entries in this opened macro file against the current RAM Macro table, if there is one. Any entries in the opened macro file that are not in the RAM Macro table will be selected when you press the **Check** button. The selected macro entries can then be loaded by pressing **OK**.   This can be helpful to avoid loading in duplicate files if you...

1. Use Macro Record to keep a running history of files that you have already loaded into the K2600.

2. Have a lot of macro files that load similar lists of files.

If the **Open** button is pressed from a disk function other than Load, you will see the VIEW MACRO page:



```
Func:VIEW MACRO   Sel:0/4     Index:    1


 5:\ANALOG\MULTIVOX.K26          200:F:
 5:\ANALOG\NOISE.K26             200:F:
 5:\ANALOG\RESONANT.K26          200:F:
 5:\ANALOG\SYNAPSE.K26           200:F:
                                  Exit
```

The only function of this dialog is to view the Macro table entries stored in a macro file. This feature is useful when, for example, you are about to delete a macro file and want to know what information is contained in the file before you remove it from the disk.

Editing Macros

The RAM Macro table can be edited from the MACRO page. You can select one or more macro entries and execute any of the following operations on them:

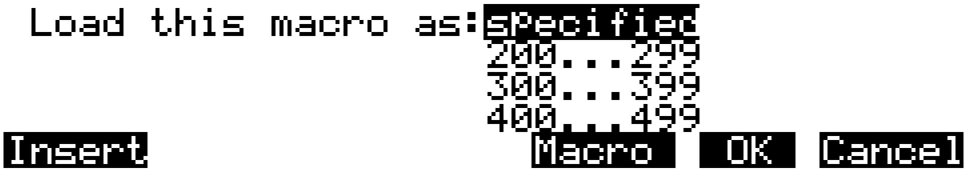
1. Change the Disk ID.

2. Change the bank and mode settings.

3. Delete the selected macro entries.

To edit a macro file already saved on your disk, it is necessary to first load the macro file into the RAM Macro table:

Making sure that Macro Record mode is enabled, go to the Load function, highlight the macro file you wish to edit, and then either select certain entries from the macro file (by pressing **Open** to get to the LOAD MACRO page), or just press **OK** to load the entire macro file. When you see the following display, press the **Macro** soft button, so that the K2600 will not load the files listed in the macro.



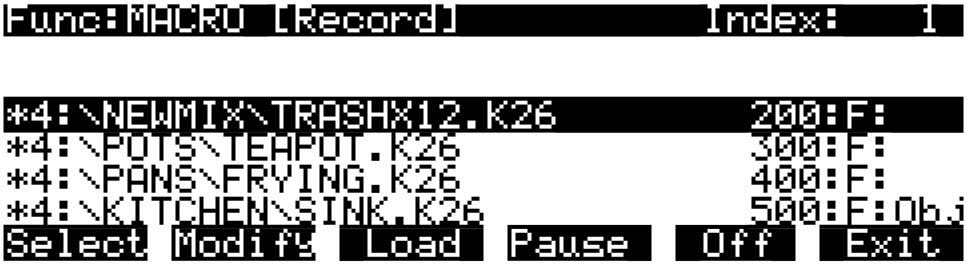
```
 Load this macro as:specified
                    200...299
                    300...399
                    400...499
Insert                Macro   OK   Cancel
```

If you want, you can rebank the macro by scrolling the bank list to something other than **specified**. Similarly, if you override the load mode, it also will be reflected in the RAM Macro table.

To edit entries from the Macro table, return to the MACRO page. For this example we will edit all the entries at once (like other similar dialogs, if you are concerned with only one list entry, it does not need to be explicitly selected with the **Select** soft button). You can use the following double-presses to select and deselect all entries in the list:

- **Left/Right** cursor double-press: Select all macro entries
- **Up/Down** cursor double-press: Clear all selections

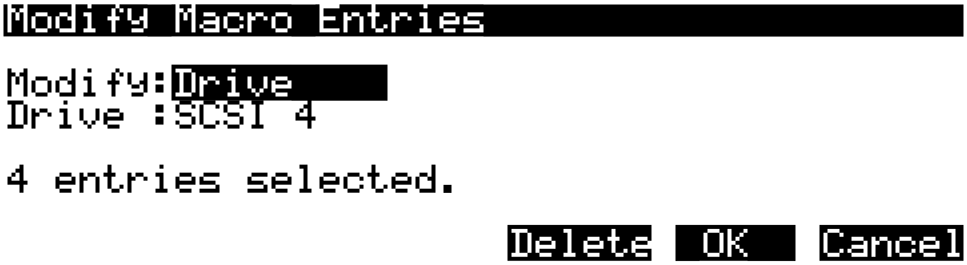
With all the entries selected, our display looks like this:



```
Func:MACRO [Record]                Index:    1

*4:\NEWMIX\TRASHX12.K26             200:F:
*4:\POTS\TEAPOT.K26                 300:F:
*4:\PANS\FRYING.K26                 400:F:
*4:\KITCHEN\SINK.K26                500:F:Obj
Select  Modify  Load   Pause   Off   Exit
```

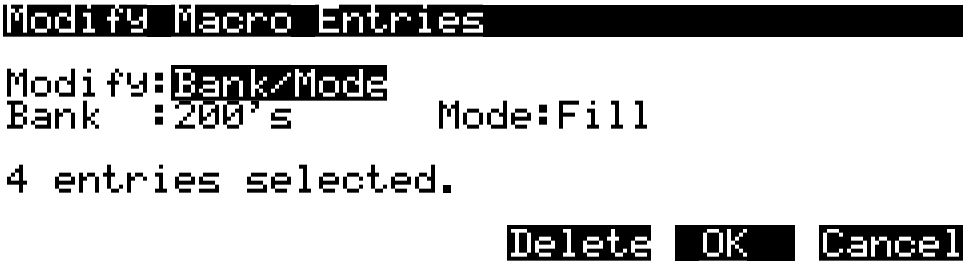
The two remaining soft buttons are **Modify** and **Load**.

Press the **Modify** soft button to change any items mentioned at the top of this section. You will see the following display:



```
Modify Macro Entries

Modify:Drive
Drive :SCSI 4

4 entries selected.

                    Delete   OK   Cancel
```

Use the Drive parameter to change the Disk ID for the selected macro entries. This is where you would set the entries' Disk IDs to be **Unspecified** or **Library**. If you increment the Modify parameter, the display switches to let you modify bank and mode information:

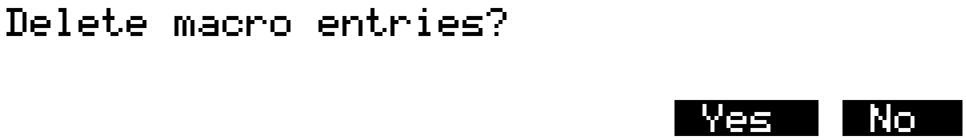


```
Modify Macro Entries

Modify:Bank/Mode
Bank  :200's      Mode:Fill

4 entries selected.
```

The initial settings of the parameters on these pages are always taken from the lowest-indexed macro entry that is selected on the MACRO page. In addition, every time you return to the "Modify Macro Entries" page, both the Modify parameter and the highlighted value will be the same.

Pressing **OK** will set all of the selected macro entries to have Disk ID or bank and mode settings according to the parameters set up on this page. The display will return to the MACRO page with the same entries still selected. Any modifications to the parameters will be visibly apparent. Selecting multiple entries for editing allows you to change those entries in a uniform way. In our example above, you could change the macro so that all the files were loaded into a single bank, instead of the separate banks they had previously been loaded to.

Press **Delete** to remove the macro entries from the Macro table. You will see the display:



```
Delete macro entries?
```

If you answer **Yes**, the display returns to the MACRO page and all of the previously selected entries will be gone from the list. If you answer **No**, the display will return to the Modify Macro Entries dialog.

Pressing **Cancel** in the Modify Macro Entries dialog will return to the MACRO page with everything that was selected still selected, but with no parameter changes made to any macro entries.

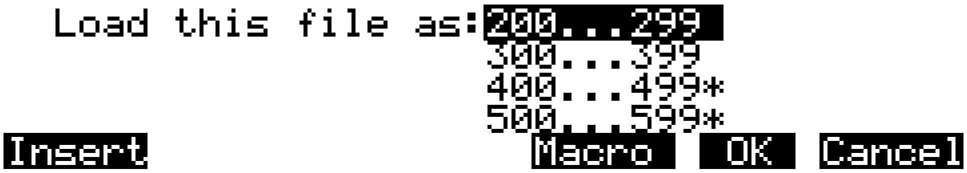
Here are the parameter values for Modify Macro Entries:

Parameter	Values
Modify	Drive, Bank/Mode
Drive	Floppy, SCSI 0–SCSI 7, Unspecified, Library
Bank	000s–900s, Everything
Mode	Append, Merge, Fill, Overwrite, OvFill

Once you have made the necessary changes to the RAM Macro table, you can go to the Save Macro page to write selected (or all) entries to a new disk file (or replace an original macro file that you loaded from a disk).

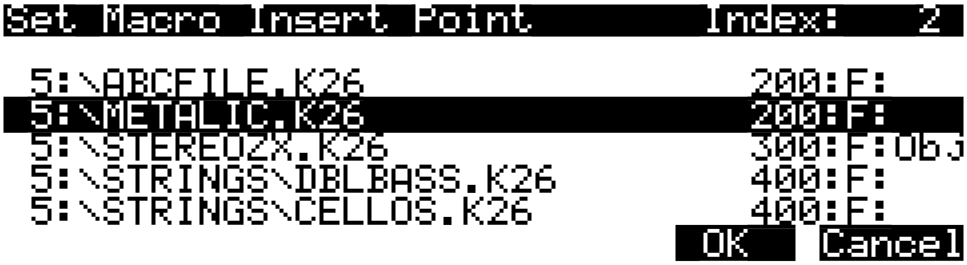
Macro Insert

You can insert new macro entries into the middle of the RAM Macro table if necessary. This is done by pressing the **Insert** soft button at the "Load this file as:" prompt, when loading a file (if Macro Record is enabled):



```
  Load this file as:200...299
                    300...399
                    400...499*
                    500...599*
Insert                Macro   OK   Cancel
```

When you press **Insert**, you will see a dialog displaying the current RAM Macro table:

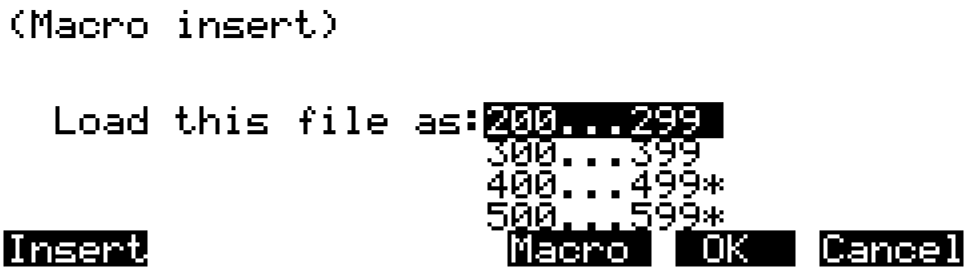


```
Set Macro Insert Point        Index:    2

 5:\ABCFILE.K26                  200:F:
 5:\METALIC.K26                  200:F:
 5:\STEREOZX.K26                 300:F:Obj
 5:\STRINGS\DBLBASS.K26          400:F:
 5:\STRINGS\CELLOS.K26           400:F:
                               OK   Cancel
```

Scroll the Macro table until the entry before which you want to insert is highlighted. In the above display, any new macro entries added by this load operation will be inserted in the Macro table just before the entry for **METALIC.K26**. That is, the new entry would have index 2, the index for **METALIC.K26** would shift from 2 to 3, and the indices for all the entries after **METALIC.K26** would increase by 1.

Press **OK** to enable the insert point. Press **Cancel** to disable the insert point.

The display will return to the Load dialog. If a macro insert point has been set, an indicator will appear at the top left of the display:

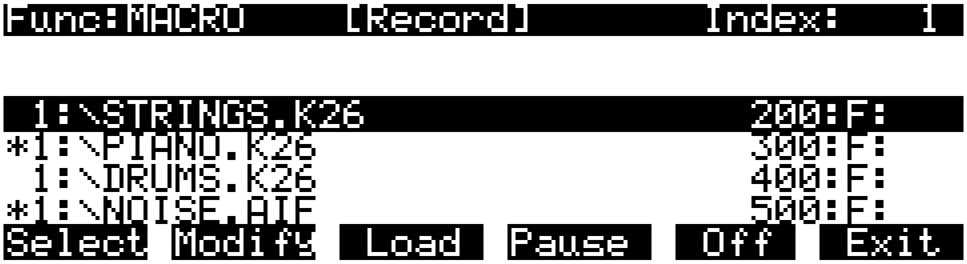


```
(Macro insert)

  Load this file as:200...299
                    300...399
                    400...499*
                    500...599*
Insert               Macro   OK   Cancel
```

The insert point can be disabled before loading, by pressing **Insert** again and pressing **Cancel** from within the Set Macro Insert Point dialog. The (Macro insert) indicator will disappear. Similarly, the insert point can be changed before loading by pressing **Insert** again (the display will highlight the current insert point,) scrolling to a different insert point, and pressing **OK**.

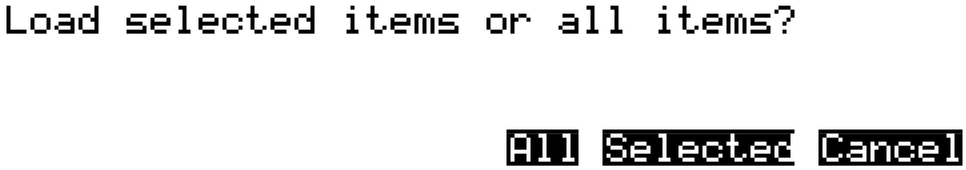
Executing the RAM Macro Table

You can load any group of files listed in the RAM Macro table. This is done using the **Load** soft button on the MACRO page:



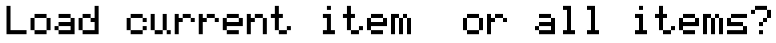
```
Func:MACRO     [Record]        Index:    1

 1:\STRINGS.K26                     200:F:
*1:\PIANO.K26                       300:F:
 1:\DRUMS.K26                       400:F:
*1:\NOISE.AIF                       500:F:
Select Modify  Load   Pause   Off    Exit
```

Pressing the **Load** button gives you the choice of loading either all of the files in the Macro table or loading only the files that are selected:



```
Load selected items or all items?

                     All Selected Cancel
```

If you don’t have any items explicitly selected (with an asterisk), the message you see when you press **Load** is slightly different:



```
Load current item  or all items?
```

The current item is the file that was highlighted on the MACRO page.

The files that have been selected for loading will be loaded in their respective order in the Macro table, using the bank and mode parameters that are specified in the list. In the example diagram above, if you were to load selected items, first **\PIANO.K26** would be loaded into the 300s bank, and then **\NOISE.AIF** would be loaded into the 500s bank.

Saving and Loading a Macro Table in a .K26 file

Macro table objects can be explicitly saved or loaded (without being “executed”) using Save Individual Object and Load Individual Object. If for some reason you wanted to save a Macro table that you’ve been working on, and then be able to load it again later on to be worked on some more, you would use this method. The Macro table can be selected for saving and loading just like any other object. When you load a Macro table using Load Individual Object, it will overwrite any Macro table already in memory. Once it is loaded, you may have to go to the

MACRO page and enable Macro Record mode to continue to record further load operations into the macro.

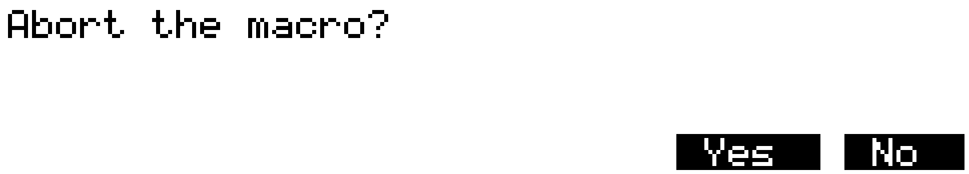
The “Save Everything” feature of the Save dialog does not include the Macro table. This is done to prevent inadvertent distribution of what would most likely be a meaningless Macro table to other users.

Aborting a Macro Load

You can abort the process of loading a macro file. Aborting can only be done “in between” files that are being loaded, and not during the load of any one file (short of powering off or doing a soft reset—but we don’t recommend this).

Aborting a macro load is done by pressing and holding down either of the **Plus** or **Minus** buttons. This should be done at least one-half second before you anticipate the current file to finish loading, or else the K2600 will not sense that you wish to abort the load.

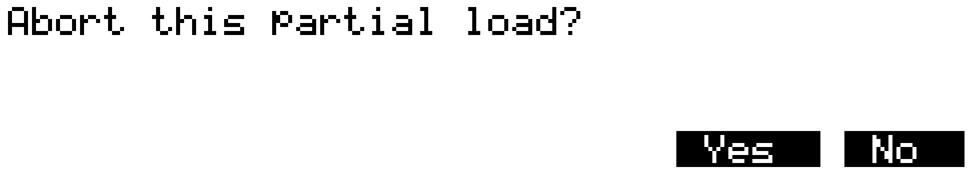
You will see the following question after the current file finishes loading:



Abort the macro?

It may be a good idea to practice using this method of aborting a macro file load, so that when you accidentally load an out-of-date macro file with 25 entries all at the wrong SCSI ID, you’ll remember how to abort the process. This same method (of holding the **Plus** or **Minus** buttons down) is also used to abort the Backup feature and the multiple file load feature.

If you run out of object or sample RAM, you will have one opportunity to abort the macro as explained above. However, if you continue from that point you may end up seeing the same error message “Memory is full” for each file to be loaded. This can be a rather tedious process, however it is still possible to abort out of this by holding down the **Plus** or **Minus** button simultaneously while pressing **Yes** when you see the following question:



Abort this partial load?

If the macro that you abort was loaded by multiple selection together with other files, you will have to abort twice, once to get out of the current macro file load, and a second time to get out of the multiple file load process.

If you are aborting a macro because the Disk ID is incorrectly specified (as evidenced by lots of "Problem mounting disk" errors) you will need to hold down either the **Plus** or **Minus** button while pressing **OK** to satisfy the error prompt. The display may blink while holding down the **Plus** or **Minus** button, but as soon as you have pressed **OK** you will see the "Abort the macro?" question.

Remote Macro Load

You can remotely load a macro into the K2600 from a sequencer. This can be useful, for example, if the K2600 is inaccessible or inconveniently situated.

Once you've created the macro that you will be remotely loading from the sequencer, set the sequencer to record, then dump the macro object to the sequencer using the **Dump** soft button on the Master-mode Object page. Then, add the SysEx LOADMACRO (10h) command to the sequence, following the macro object. Although some sequencers allow you to record a SysEx command directly into a sequence, the K2600 provides a convenient shortcut, described in the next paragraph.

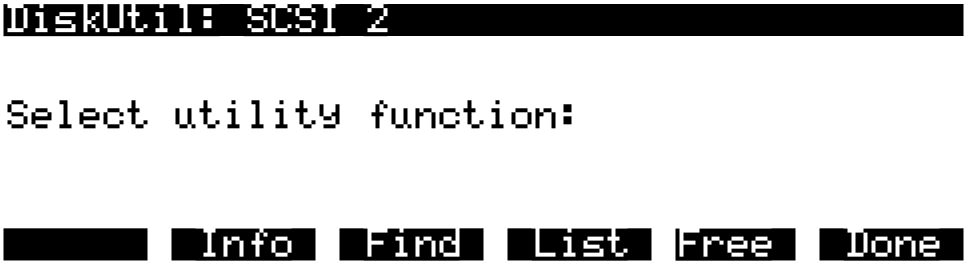
To add the LOADMACRO command to a sequence (after dumping the macro object to the sequencer), leave the macro object highlighted on the Master-mode Object page, then press the **Left** or **Right** cursor button. The K2600 will display: "Send SysEx Load Macro command?" Press the **Yes** soft button and the K2600 will add the LOADMACRO SysEx command to the sequence.

***Note**: You cannot remotely load a macro to the K2600 while the K2600 is on the Disk-mode page or in an edit mode.*

Disk Utilities

The Disk Utility functions provide certain necessary information about disks and their files and directories. These functions are useful when you want to know how your disks are organized and how much disk space you have available. They also help you to locate files and directories.

To access the Disk Utilities page, press the **Util** button from the Disk-mode page. The Disk Utilities page looks like this:



```
DiskUtil: SCSI 2

Select utility function:
```

The functions on the Disk Utilities page are used for finding out information about the Current Disk. The Current Disk is always indicated on the top line of this page. If you want to use the utility functions for a different disk, you must first set it to be the Current Disk on the Disk-mode page.

Here is a brief description of each soft button:

Info Provides general information about the current disk, such as the Disk ID and formatting specifications.

Find Enables you to search for files that match a certain character string in their filenames.

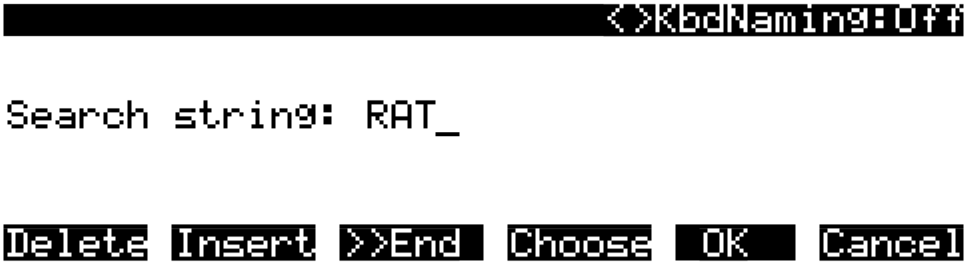
List Lists an expanded directory tree from any level of the hierarchy, showing the current directory's contents, and the contents of all of the current directory's subdirectories. This function can be used to determine the total size of files within any tree of subdirectory. It is also helpful for finding files on the disk.

Free Calculates the available space on the disk.

Done Exits to the Disk-mode page.

Find Files

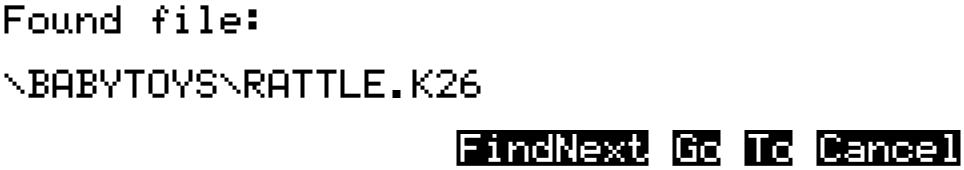
The Find files utility first prompts you to enter a character string to be searched for:


<>KbdNaming:Off

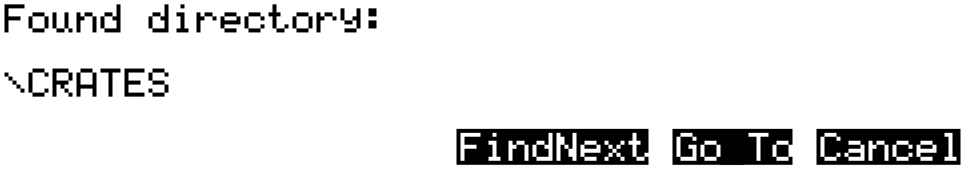
Search string: RAT_

You can use the **Choose** button to grab the text of a filename from the current disk, as described previously.

If you press **OK**, the K2600 will begin to search the disk for any files or directories that contain the search string in their names. The search starts in the root directory and searches the entire disk. When a matching file or directory is found, you'll see one of the following:


Found file:

\BABYTOYS\RATTLE.K26


Found directory:

\CRATES

If the search string is found anywhere within a filename it will be matched. The search algorithm independently checks both the filename and the extension. For example, if you wanted to find any macro file on the disk, you could enter in **MAC** for the search string. This would find any macro files as well as any files or directories that have **MAC** in their filename.

When a match is found, there are three choices displayed:

FindNext Continue searching the disk for another file or directory that matches the search string.

Go To Exit to the Disk-mode page, setting the current directory and file index of the K2600 to the location of the found file or directory.

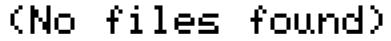
The next disk function you choose will display the current directory with the found file already highlighted. If a directory was found, then the first file in the directory list will be highlighted.

Cancel Exit to the Disk-mode page.

When the search has checked all of the items on the disk, you'll see this dialog:

```
Search completed.
```

If no matching files were found, you will also see

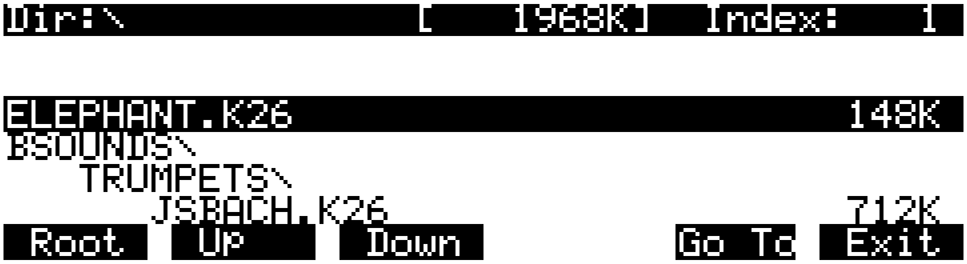


```
(No files found)
```

The K2600 will remember the last search string that you entered. This makes it easy to repeat the same search. If you press **Util-> Find** again, the "Search string:" dialog will contain the previously used string.

List

The List utility allows you to view directories on a disk with the expanded contents of all subdirectories. Each line is indented according to its directory level, so that you can get sense of how your directory tree is organized.



```
Dir:\                [    1968K]  Index:    1

ELEPHANT.K26                              148K
BSOUNDS\
   TRUMPETS\
      JSBACH.K26                          712K
```

The Dir field shows the directory that is being listed. The size value displayed on the top line of the display represents the total size of the directory subtree that is currently being viewed. The **Root**, **Up**, and **Down** soft buttons navigate through the directory hierarchy:

Root Display the disk from the top level, meaning that all of the files on the disk will appear in the scrollable list.

Down Set the display to the level of the highlighted file or directory.

For example, scrolling to **TRUMPETS** in the above list, and pressing **Down**, would focus the list on the contents of the **TRUMPETS** directory, starting with **JSBACH.K26**.

Up Set the display up one directory level.

Go To Exit to the Disk-mode page, setting the current directory and file index to the location of the highlighted file or directory.

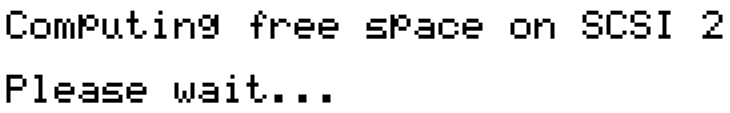
The next disk function you choose will display the current directory with the found file already highlighted. If a directory was found, then the first file in the directory list will be highlighted.

Cancel Exit to the Disk-mode page. The current directory is unchanged.

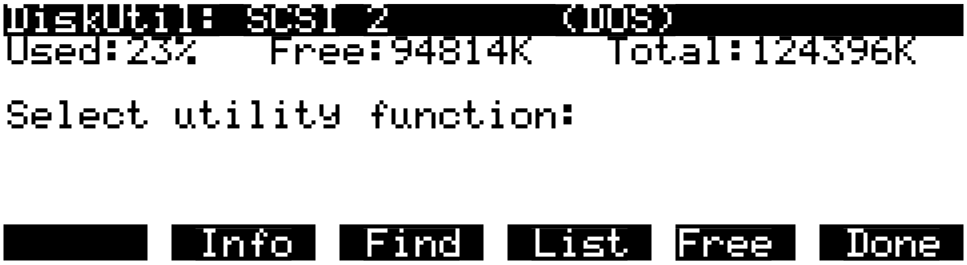
The files are listed in the order that they appear on the disk, unalphabetized. The traversal order of the directories is the same one that is also used for the Backup function.

Free

The Free utility calculates the amount of free space on the current disk and displays the result on the Disk Utilities page. This may take a small amount of time to calculate, depending on the disk.



```
Computing free space on SCSI 2
Please wait...
```



```
DiskUtil: SCSI 2         (DOS)
Used:23%   Free:94814K   Total:124396K
Select utility function:
```

The parameters tell you the following:

Used The percentage of the disk that is taken up by the existing information stored on it.

Free The amount of disk space available for new files.

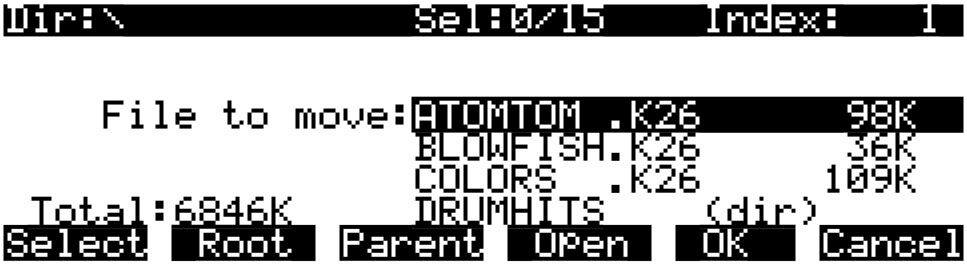
Total The size of the disk. This will be the size of the usable partition if MS-DOS format.

For SCSI disks, if the current disk was formatted on a PC or a Mac in MS-DOS format and contains at least one primary partition, you will see the **(DOS)** indicator on the top line. Using this format is described in *MS-DOS File System Compatibility* on page 13-71.

For floppy disks, the density (HD or DD) will be displayed. All K2600 floppy disks are MS-DOS format compatible.

Moving Files Between Directories

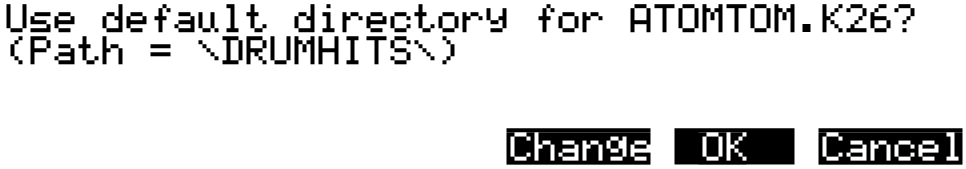
Files and directories can be moved from one directory to another on a given disk. You can either choose multiple files to move using the **Select** soft button, or just move the single highlighted file or directory (if no other files are selected). As you would probably expect, moving a directory also moves all the files within the directory. To use this function, press the **Move** soft button from the Disk-mode page. Then choose the file or files that you want to move:



```
Dir:\                    Sel:0/15    Index:    1


    File to move:ATOMTOM .K26        98K
                 BLOWFISH.K26        36K
                 COLORS  .K26       109K
  Total:6846K    DRUMHITS     (dir)
```

Press **OK** when you have made your selection. Press **Cancel** to return to the Disk-mode page.

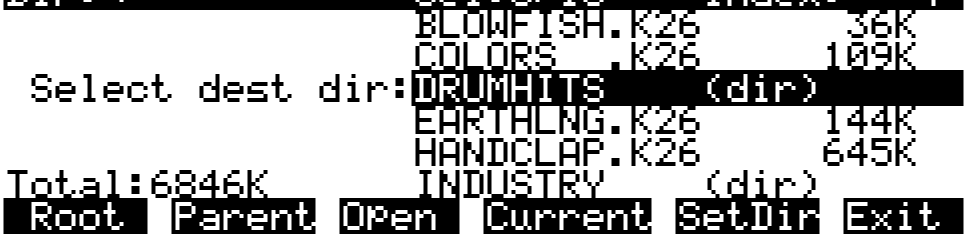
The K2600 remembers the most recent destination directory that a file was moved to. If the current directory is different from the most recent destination directory, you will see the question:



```
Use default directory for ATOMTOM.K26?
(Path = \DRUMHITS\)
```

Press **OK** to use the default.

Press **Change** if you want to select a different destination from the default shown. The K2600 will then display a directory selection dialog (see page 13-13), and you can select the move destination directory from there:



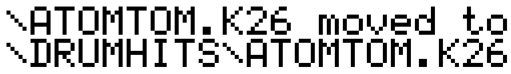
```
Dir:\                    Sel:0/15    Index:    4
                 BLOWFISH.K26        36K
                 COLORS  .K26       109K
  Select dest dir:DRUMHITS     (dir)
                 EARTHLNG.K26       144K
                 HANDCLAP.K26       645K
Total:6846K      INDUSTRY     (dir)
```

The move operation begins when you press either **Current** or **SetDir**.

If the default destination directory is the same as the source directory, the K2600 will skip the Use default directory? question and instead go right to the Select dest dir dialog.

A good way to organize files into subdirectories is by using the Move command. First, create the subdirectories you need, using the **NewDir** function. Then, use multiple file selection to select the files to be grouped into a particular subdirectory. The files can be moved to their new location in one operation.

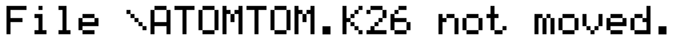
For each file that is moved, you will see a confirmation message:



```
\ATOMTOM.K26 moved to
\DRUMHITS\ATOMTOM.K26
```

Note the following:

- You can select multiple files for moving within a directory. However, you cannot move files from more than one directory at a time. If you select any files and then switch to another directory, the files you had chosen will be deselected.
- If you are moving a directory, you cannot move it in to a subdirectory of itself.
- If the source and destination directories are the same, the file will not be moved, and an error message such as the one below is displayed. This would happen if you pressed **Current** above.
- The same message will be displayed if there is a file in the destination directory with the same name as the file to be moved.

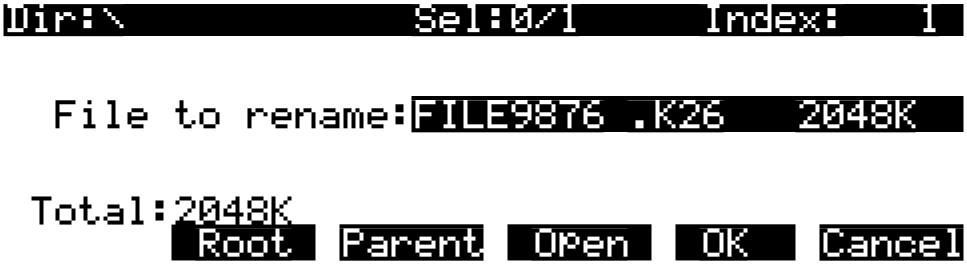


```
File \ATOMTOM.K26 not moved.
```

Renaming Files

Press the **Rename** soft button (from Disk mode) to rename files or directories without loading them. When you press **Rename**, the K2600 will prompt you to select the file to be changed, by showing you a list of the files found on the current disk.

When you've selected the file to be renamed, press **OK**, and the K2600 will ask you to enter the new filename. When you've done this, press **OK**, and the filename will be changed.

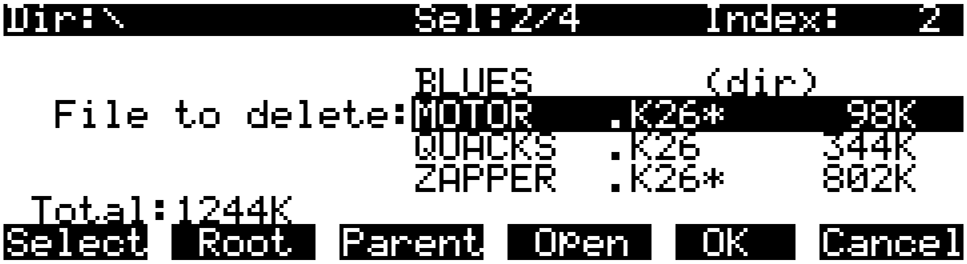


```
Dir:\                Sel:0/1      Index:    1

   File to rename:FILE9876 .K26   2048K

  Total:2048K
```

This function can be used to change only the 8-character filename, not the extension. When you press **Rename**, the File List dialog is displayed and you can navigate through the directories to choose the file or directory you wish to rename. Unlike the other disk functions that use the File List dialog, you will not see the **Select** soft button. This is because you can rename only one file at a time. Therefore you simply choose the file you want and press **OK**. The K2600 will then ask you to enter a new name, which you can do a number of ways: Alpha Wheel, **Up/Down** cursor buttons, alphanumeric buttonpad, or keyboard naming (see page 5-5 for a description of keyboard naming). Once you've done this, press **OK** again, and the filename will be changed.

Deleting Files and Directories

Press the **Delete** soft button (from Disk mode) to delete files and directories. The Delete function supports multiple selection of files for deletion. Select one or more files and/or directories to be deleted, and press **OK** (or **Cancel** to abort). Be careful! You don't get a second chance to change your mind once you've pressed **OK**. Once a file is deleted, it's gone. Remember the fundamental directive of computer users: Save early, save often; make backups.



```
Dir:\                Sel:2/4     Index:    2

                     BLUES        (dir)
  File to delete:MOTOR     .K26*      98K
                     QUACKS    .K26      344K
                     ZAPPER    .K26*     802K
 Total:1244K
```

When you press Delete, the File List dialog appears, and you can navigate through the directories to choose the file or directory you want to delete.

Within the current directory, you can select multiple files for deletion. You can't, however, delete files from more than one directory at a time. If you move to a different directory in the middle of a deletion, any files you had selected up to that point get deselected.

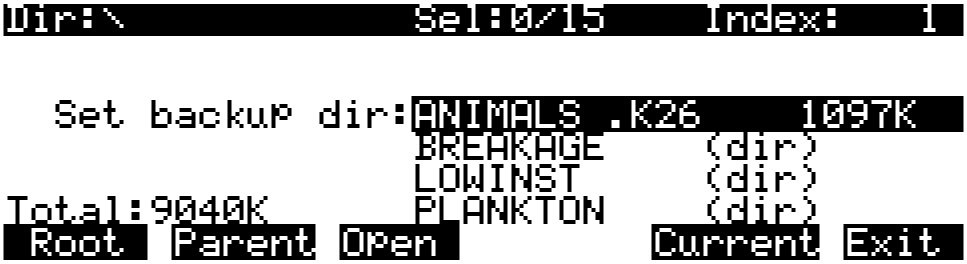
You cannot delete directory if it has any files in it. To delete a directory, you must first delete its contents. Also, you can't use the **Select** soft button to select a directory for deletion. To delete a directory (once it's empty, of course), highlight the directory and press **OK**. If the Confirm parameter on the Master-mode page is set to **On**, the K2600 will ask you if you're sure. Press Yes and the K2600 begins deleting the selected objects. If one of the selected objects is a directory that contains files, the K2600 will tell you that it can't delete the directory.

When selecting files and directories for deletion, you can use the **Open** soft button to open directories and files. Opening a directory at this point is useful for selecting files within the directory. Opening a file is less useful, since you can't delete individual objects from files. You can view the file's contents, but you can't select any of them for deletion.

Backup and Copy Functions

File Backup

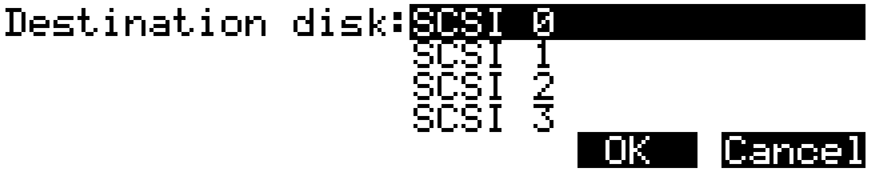
To access the Backup function from the Disk-mode page, first make sure that the current disk is set to be the disk that you want to back up. Next, press the **Backup** soft button:



```
Dir:\                  Sel:0/15    Index:    1


   Set backup dir:ANIMALS .K26     1097K
                  BREAKAGE      (dir)
                  LOWINST       (dir)
Total:9040K       PLANKTON      (dir)
 Root   Parent  Open             Current  Exit
```

Select a directory to be backed up (see *The Directory Selection Dialog* on page 13-13). Backup allows you to copy all of the files within a directory from one disk to another. All of the files within the directory that you choose (plus all of its subdirectories and the files within them) will be copied to the new disk. If you want to backup the entire disk, then make sure the current directory is the root directory (as in the picture), and press **Current**.

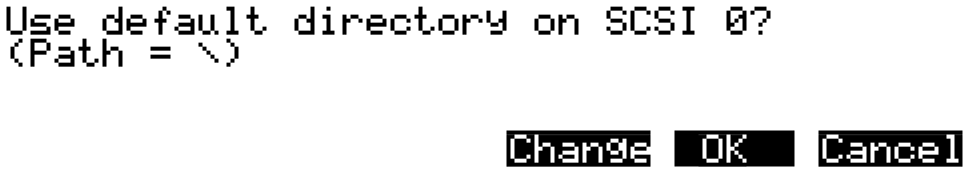
Next, you will see a dialog for choosing the destination disk:



```
Destination disk:SCSI 0
                 SCSI 1
                 SCSI 2
                 SCSI 3
                           OK     Cancel
```

Select the disk you wish to transfer files to. It must be different from the current disk.

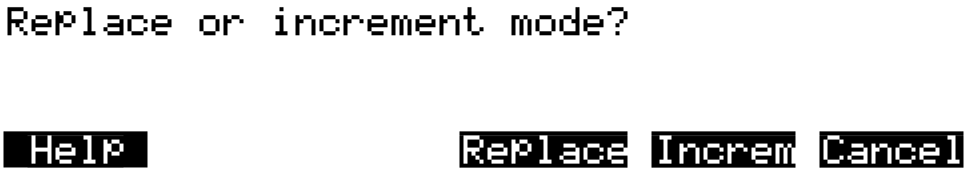
Next, you can select a directory on the destination disk that will receive the transferred files.



```
Use default directory on SCSI 0?
(Path = \)


                      Change   OK    Cancel
```

The default is always the root directory on the destination disk. Press **OK** to select the default. To select a different directory, press **Change**.

Next, select the Backup mode when you see the following question:



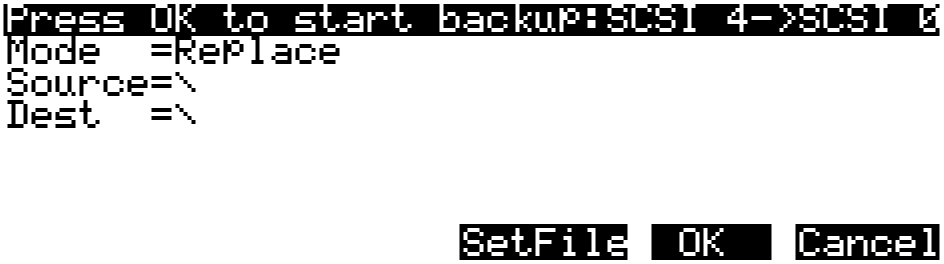
```
Replace or increment mode?
```

Replace Any files to be transferred that already exist in the destination directory will be replaced (overwritten).

Increm Any files to be transferred that already exist in the destination directory will be skipped (not transferred).

Help Displays a reminder about the meaning of Replace and Increment modes.

After you select the Backup mode, you will see a confirmation dialog with all of your selections so far:



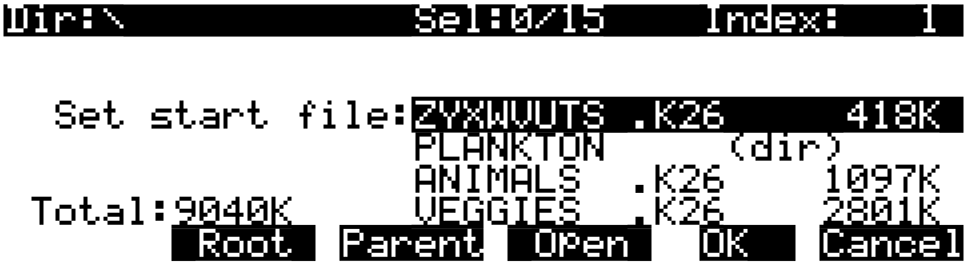
```
Press OK to start backup:SCSI 4->SCSI 6
Mode  =Replace
Source=\
Dest  =\
```

OK Begin the Backup function according to the parameters on this page.

Cancel Exit to Disk mode.

SetFile Set Backup starting file. This is mainly used when you are backing up files from a larger SCSI disk to several smaller removable media disks. Once one of the removable destination disks is full, note the last file that was successfully copied. Enter the Backup function again, setting up the source and destination path parameters exactly as before. But this time use the **SetFile** button to start from the file after the last file copied.

You will notice that the file list displayed when **SetFile** is pressed is not in alphabetical order, but in the actual order the files occur in the directory on the disk:



```
Dir:\                   Sel:0/15     Index:    1


  Set start file:ZYXWVUTS .K26      418K
                 PLANKTON     (dir)
                 ANIMALS  .K26     1097K
 Total:9040K     VEGGIES  .K26     2801K
```

Find the last file copied from the previous partial backup. Set the list index to one entry past the location of this file. If this file is already the last entry in the file list, the “next” file to continue the multi-part backup would be the file or directory that comes after the file list entry for the currently viewed directory. To find this file or directory, you will need to look in the Parent directory.

A large SCSI hard disk backed up in this way can be restored by individually backing up the removable media onto the SCSI hard disk.

Backup and Copy will transfer files much faster if you have cleared your object RAM first using the Delete Everything command (save any objects in RAM to disk first).

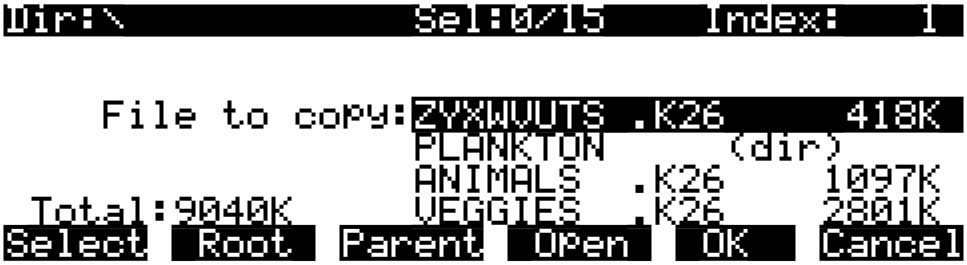
When using the Backup function to transfer files from SCSI to floppy, directories are ignored (that is, not created on the floppy). This “flattens” out any directory structure so that multi-disk files can be created on the floppy disks.

File Copy

The Copy function lets you copy one or more files from one disk to a specified directory on the destination disk.

Copy is similar to Backup, with a few differences:

- You can’t copy directories.
- The Replace or increment mode dialog does not appear. Instead, if you’re copying a file that already exists in the destination directory, the K2600 asks “Replace existing *filename* on *destination disk*?



```
Dir:\                Sel:0/15    Index:    1

    File to copy:ZYXWVUTS .K26      418K
                 PLANKTON     (dir)
                 ANIMALS  .K26     1097K
 Total:9040K     VEGGIES  .K26     2801K
Select  Root  Parent  Open   OK   Cancel
```

Creating a Startup File

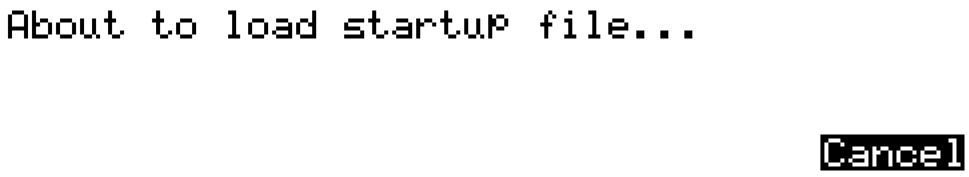
You can create a macro file that will be automatically loaded when you power up your K2600. This file, called the startup file, or boot macro, can be on a floppy disk or on a disk at any SCSI ID. See the section on macros for background information.

The steps needed to create a Startup file are:

First, create a macro file called **BOOT.MAC** in the root directory of the disk that you will use as the Startup disk. Specify in the macro the exact ordering of files that you would like to have loaded into the K2600 when powered on. When you save the macro file, just name the file **BOOT**, and the K2600 will add the **.MAC** extension.

Second, set the Startup parameter on the Disk-mode page to be the Disk ID of the Startup disk. So, if your **BOOT.MAC** file was on a floppy disk, set the Startup parameter to **Floppy** and make sure that you have the correct floppy in the drive when you next turn the K2600 on.

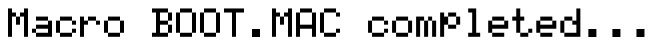
When the K2600 is powered on, it will display the following message (after the introductory VAST logo):



```
About to load startup file...

                                    Cancel
```

The K2600 looks for a file **BOOT.MAC** in the ***root directory*** on the disk specified by the Startup parameter. If the file is not found, or the disk cannot be accessed, you will get an error message. The Startup load can be bypassed in the first few seconds after the K2600 is turned on, by pressing the **Cancel** button.

If **BOOT.MAC** is found by the K2600, it will begin to load the macro file as if you had loaded it explicitly from the Load function in Disk mode. When the macro has completed, you will see the following:



```
Macro BOOT.MAC completed...
```

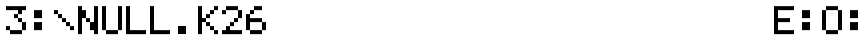
The K2600 will go directly to Program mode afterwards.

Deleting Banks in a Startup File

You may want the Startup file to clear out one or all banks in the K2600 before loading files. This could help overcome the problem of having "silent" copies of programs in your RAM that depend on samples that are no longer there (because they disappeared the last time the K2600's power was turned off). The following trick will allow a macro entry to essentially function as a Delete Bank or Delete Everything command:

Create a file somewhere on (preferably) your Startup disk, by saving an empty bank from the K2600. Call the file **NULL.K26**. Now, insert this file at the beginning of a boot macro you are creating: load the file, specify the bank you want to delete in the Startup file (or specify **Everything** if you want to clear RAM completely), and specify **Overwrite** for the load mode. Make sure you press **Macro** and not **OK**, so that the overwrite doesn't take place until you use the Startup file.

Here is what that macro entry for this file might look like on the MACRO page, if you were doing a "Delete Everything"



```
 3:\NULL.K26                              E:0:
```

The E:0: stands for "Load to all banks, using Overwrite mode."

MS-DOS File System Compatibility

The K2600 is compatible with fixed and removable disk drives that use the MS-DOS hard disk and floppy disk formats. If you want to use this feature, *you must first format the disk media on a computer* such as a PC compatible or a Mac running appropriate MS-DOS conversion software.

The MS-DOS hard disk format is structured so that the disk can be split up into multiple partitions. *The K2600 uses only the first partition that it finds on the disk*. Therefore, it is usually best to format the media with only one partition taking up all usable space on the disk. Working from the K2600 front panel with an MS-DOS formatted disk will appear the same as working with a disk that has been formatted with the K2600's own Format function (on the Disk-mode page). The Free utility (Disk mode->**Util**->**Free**) can be used to identify whether a disk is in DOS format or in standard K2600 format. If the **(DOS)** indicator is displayed, it means the K2600 has determined that the disk is a DOS-format hard disk with at least one primary DOS partition. Floppy disks do not display the DOS message.

Some advantages of working with an MS-DOS compatible disk format over the standard K2600 format are:

- Easier sharing of K2600 files with other users over computer communications lines
- Ability to use graphical file management interfaces for organizing files and directories
- Ability to back up K2600 data using a PC compatible or Mac with commercially available software
- Easier transfer of data using standard file formats such as AIFF, WAVE, and MIDI Type 0, for importing and exporting samples and sequences

Filename Compatibility

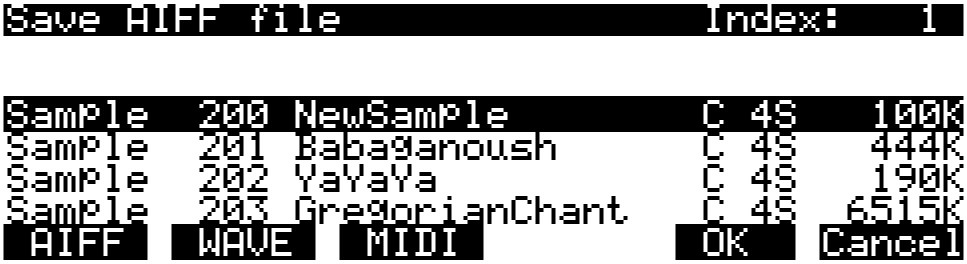
DOS format does not support space characters in filenames. The K2600, though, allows spaces to be used within filenames. If you plan to transfer files between the K2600 and a DOS compatible computer, it is recommended that you use only filenames without space characters in them. Otherwise, a computer may have trouble identifying the files.

Importing and Exporting Data Using Standard File Formats

The K2600 supports three common data interchange file formats, Apple Interchange File Format (AIFF), Microsoft RIFF WAVE, and MIDI Type 0 and Type 1. The first two are used to transfer sample data, and the latter is used for sequences.

The K2600 can recognize these file types automatically on loading, regardless of the file extension. You can load these files as you would any standard K2600 file, and also as part of a macro file load. The most recent sample file loaded will become the "preview" sample, which means you can quickly access it for playing or editing on the SampleMode page (press the **Master** mode button, then the **Sample** soft button). Similarly, the most recently-loaded MIDI Type 0 file will become the current song on the Song-mode page.

You can save files in these formats on the Export page. This page is accessible from the Disk-mode page by pressing **Save-> Export**.



```
Save AIFF file                        Index:    1


Sample   200 NewSample                C 4S    100K
Sample   201 Babaganoush              C 4S    444K
Sample   202 YaYaYa                   C 4S    190K
Sample   203 GregorianChant           C 4S   6515K
  AIFF     WAVE     MIDI               OK    Cancel
```

The Export page allows you to save one sample or song object per file. Choose the format you wish to save in, and press the corresponding soft button. For AIFF and WAVE, only sample objects are listed. For MIDI Type 0 and Type 1, only song objects are listed. Scroll to the object that you wish to save, and press **OK**. The dialog will proceed the same as if you were saving a K2600 file. You will be prompted for a filename, and will have the option to select a different default directory to save the file in.

The K2600 will automatically place a standard extension on the file when it is created on the disk. These extensions are sometimes necessary when transferring files to external programs that can recognize the file format based on the extension. They also help you to recognize the format of each file when looking at a directory listing. You can use the Find Files utility (Disk mode-> **Util**-> **Find**) to search for files that match a certain extension. The standard extensions used on the files are:

.AIF	AIFF
.WAV	WAVE
.MID	MIDI Type 0, Type 1

The first time you enter the Export page after powering on (or after a soft-reset), the format defaults to AIFF. After that, the K2600 remembers the most recent format that you used. For example if you save a MIDI Type 0 file, and then go back to the Song-mode page to record more sequences, the next time you return to the Export page, the file format will still be set to **MIDI**, and all of the song objects will be listed. You can audition the samples and songs the same way as you would on the Save Object or Object Utility pages (by pressing the **Left** or **Right** cursor buttons).

AIFF and AIFF-C Files

The K2600 can read 8 or 16 bit AIFF files, mono or stereo. The sample rate, sustain loop, loop mode, base note, sample name, and sample detuning are supported. AIFF-C files that do not use compression can also be read by the K2600. The K2600 will save 16-bit AIFF files, either mono or stereo.

WAVE Files

The K2600 can read 8 or 16 bit WAVE files, mono or stereo. It can also save 16-bit WAVE files, either mono or stereo.

Standard MIDI Files (MIDI Type 0 and Type 1 Files)

The K2600 reads and writes MIDI Type 0 (single-track) and Type 1 (multi-track) files. The K2600 supports all musical timestamp resolutions, and automatically scales imported information to the K2600's internal sequencer resolution of 768 ticks per beat, if necessary.

Tempo information is supported, defaulting to 120 beats per minute if no tempo is specified.

SysEx data are supported.

If there is a time-signature event in a file, the first one becomes the song's time signature. Otherwise, the time signature defaults to 4/4. Time signature changes are not supported.

Apart from SysEx, tempo and time-signature events, all other meta events are skipped to minimize RAM usage.

Importing Samples from Other Manufacturers

The K2600 will load samples from Akai, Roland, and Ensoniq EPS and EPS-16 Plus SCSI disks and floppies (including ASR-10 "Ensoniq" format), using the Load operation. (Roland samples can be loaded only via SCSI, while all other samples can be loaded from floppy or via SCSI.) The displays you see will vary depending on the samples you're loading, but several features are the same. We'll describe the similarities first, then elaborate on the differences.

If you're working with sample files in one of these formats, you may notice that once you select the disk that contains the samples (this is done with the Current Disk parameter on the Disk-mode page), the soft buttons change to accommodate the structure and content of the disk. The K2600 automatically recognizes the type of disk when you select it.

Press the **Load** soft button, and you'll see a page prompting you to select something to load (we'll call them objects, since different manufacturers give them different names). The top line of the display will tell you the number of objects available of the currently selected type, as well as the index number of the currently selected object. You can select any object in the list by typing its index number on the alphanumeric buttonpad and pressing **Enter**. The next step is to use the soft buttons to select the type of object to be loaded.

Once you've selected the type of object to load, press the **OK** soft button, and the bank dialog will appear, enabling you to select the bank into which the object(s) will be loaded. When you've selected a bank, press the **OK** soft button, and the loading process will begin. At the center of the display you'll see the object currently being loaded. The top line of the display will fill with asterisks to indicate the status of the current object. The bottom line will tell you the total number of kilobytes to be loaded.

The K2600 will create layers as necessary when you load objects. These layers have the same settings as Layer 1 of Program 199.

When the load is complete, the Disk-mode page will reappear. You can now proceed with another load, or go to any other mode. If you exit Disk mode, the K2600 will remember the file that you selected most recently. When you return to Disk mode, this file will be highlighted.

Once you've loaded a sample or program file, you can save it as a Kurzweil object. You'll find it can be loaded and backed up much faster as a Kurzweil object than in its original format.

Akai

The first page to appear is the page for loading files. The soft buttons name the operations: **HDrive**, **Volume**, and **File** on the left, and **OK** and **Cancel** on the right.

The hierarchy of objects is shown by the three soft buttons on the left. The display prompts you: "File to load:" The **HDrive** button selects the partition on the currently selected disk. The **Volume** button selects volumes within the currently selected partition. The **File** button selects an individual sample file from within a volume. The **OK** button, toward the right, executes the displayed function: partition selection, or loading a volume or file. The **Cancel** button returns you to the Disk-mode page.

When you press the **HDrive** button, the center of the display's top line shows the currently selected volume in the currently selected partition. The prompt at the center of the display will read: "HD Partition." The list of available partitions will appear following the prompt. They're usually named A through F. Use the cursor buttons or numeric entry to highlight a different partition. Pressing the **OK** soft button will select the highlighted partition.

Pressing the **Volume** button will change the prompt to "Volume to load:" The list of available volumes in the current partition will appear. The center of the top line will show the current partition. The Layer buttons will scroll through the list of available partitions. Use the cursor buttons or numeric entry to select a different volume. Pressing the **OK** button will load the entire highlighted volume, unless the volume is larger than your available sample RAM, in which case, the K2600 will load as many files as will fit.

The Bank dialog will appear, enabling you to select the bank that will receive the volume. Press OK again, and you'll be prompted to press either the **Progs** soft button, which will load program information in addition to the samples, or the **Samps** soft button, which will load only the sample information. Programs are identified by the extension **.p**, and are stored in program RAM. Samples have the extension **.s**, and are stored in sample RAM. You can press **Cancel** to return to the Disk-mode page without loading the volume.

If you load sample objects, you'll see the following prompt: Create preview program/keymap? If you answer **Yes**, the K2600 will load the samples into a program that it creates based on Layer 1 of Program 199. Loading program objects will load multi-layer samples and keymaps, and sometimes velocity switches. The K2600 will create layers as necessary when you load program objects. These layers have the same settings as Layer 1 of Program 199. In some cases, the K2600 will also create stereo keymaps to preserve the separation of stereo samples.

If you press the **File** button, the prompt will change to the Load dialog. You can view the list of files with the cursor buttons, or use numeric entry. The top line of the display will show the currently selected volume. Select different volumes with the Layer buttons. The size of the currently selected file, in kilobytes, is shown just above the soft buttons on the left. Press the **OK** button to load the highlighted file.

Press **OK**, and the Bank dialog will appear. Press **OK** again, and the file will be loaded into the highlighted bank.

Sometimes you may find an Akai floppy disk that your K2600 can't read. In this case, check the format of the disk. Akai allows you to format a double sided-double density (DD) disk for high density (HD). This is not standard in the computer industry, since DD disks are not verified for use as HD disks. If this is the case, you will need to load the disk into an Akai, then save it out to a properly formatted disk.

Roland

For Roland disks, the hierarchy is a bit different; the objects that can be loaded are called volumes, performances, patches, and samples. The page that was selected last time a SCSI load was executed will appear when you initiate the load operation. Following the prompt is the list of available objects, with the size of the object in kilobytes displayed as well. The top line of the display will show the number of available objects of the selected type.

Use the soft buttons to highlight the object to be loaded. The layer buttons will take you through the current object list in increments of 100. Press **OK** to execute the load. The Bank dialog will appear. Press **OK** again, and the object will be loaded. The display will update you on the progress of the load.

EPS

For EPS disks, the hierarchy consists of files and directories. Directories can be nested several layers deep. When you press the **Load** soft button, you'll be prompted to select a file or directory to load from the list of available files and directories. The currently highlighted object will be either a file or a directory. If it's a file, its name and size will be shown following the prompt. If it's a directory, its name appears, followed by **(dir)** to indicate its type. The Layer buttons will take you to the first and last files of the currently selected directory. Note that if you're loading from floppy disks, you won't see any of the displays or soft buttons relating to directories, since the directory architecture is not supported for floppies.

When a file is highlighted and you press **OK**, the Bank dialog appears; press **OK** again to load the file. When a directory is highlighted and you press **OK**, you enter that directory, and the list of files and subdirectories in that directory appears, each file followed by its size, and each subdirectory, if any, followed by **(dir)**. The top item in every list you select is always the parent directory of the files below it. Select the top item in a list to go up one directory level.

Pressing the **Exit** soft button will take you one level back up the hierarchy. Pressing it repeatedly will take you to the root directory—the directory at the top of the hierarchy. The quickest way to the root directory is to press the **Root** soft button. The top line of the display shows you the name of the currently selected directory (or subdirectory). Pressing the **All** soft button will load all files in the current directory (but not any subdirectories). The Bank dialog will appear, and when you press **OK**, you'll be prompted to press the **Progs** button to load program information in addition to the samples, or the **Samps** button to load only the samples.

Chapter 14 Sampling and Sample Editing

Setting Up For Sampling

Before you begin sampling, you'll need to connect the proper cables from your sample source to your K2600. The cables and input jacks you use depend on the sample format you choose, and the output configuration of your sample source.

Note that sampling requires the K2600 sampling option. Even without the sampling option, however, you still have access to all of the sample editing features covered later in this chapter, provided you have some sample RAM in your K2600. Samples can be loaded from disk, or dumped into the K2600 via MIDI Sample Dump Standard (SDS) or over SCSI using the SMDI protocol. See the *Musician's Reference* for information on the MIDI Sample Dump Standard and SMDI. Also see *SIMM Specifications* on page 9-2 of the *Musician's Reference* for information about sample RAM requirements.

Cables and Input Jacks

If you're going to be sampling from an analog source, you have two options:

- For unbalanced signals, use a 1/4-inch mono or stereo cable connected to the 1/4-inch (HiZ) stereo analog input jack
- For balanced signals, use balanced XLR (cannon) cables connected to one or both of the XLR mono analog inputs

Although it's possible to send a balanced signal on a 1/4-inch cable, avoid sending a balanced signal to the 1/4-inch jack when you're making stereo samples, since doing so can cause phase cancellation in your signals.

Using a mono cable sends the signal to the K2600's left channel. If you use a mono cable, be sure to set the Mode parameter on the SampleMode page to a value of **Mono(L)**.

If you're using a digital sample source, you can use either a coaxial cable or an optical cable, depending on the output format of your sample source. The coaxial input on the K2600 accepts a standard male XLR fitting. The K2600's XLR input is configured as follows: Pin 1 is Common, Pin 2 is High, and Pin 3 is Low. Depending on your sample source (a commercial DAT deck, for example), you may need to use a cable with an RCA connector on one end and an XLR connector on the other. In this case, you should tie the Common wire to the Low wire at the RCA end. Glyph Technologies (telephone 1-800-335-0345) carries an XLR-to-RCA cable that works with the K2600.

If your digital sample source has an optical output, connect your cable to the optical input jack above the coaxial jack on the K2600's rear panel. This jack is covered by a small plug which is easily removed. This plug should be kept in place whenever the optical input is not in use, since dust or dirt can cause the optical input to malfunction.

Entering The Sampler

There are two different ways to get to the SampleMode page. The method you choose depends on the type of sampling you are doing—how many samples you are making and whether you need custom keymaps.

The difference between the two methods is primarily a matter of ease of access to the Keymap Editor. Once you have made your samples, you need to assign to a keymap and assign that keymap to a layer in a program. Refer to the section entitled *Building a Keymap* on page 14-39 for a step-by-step explanation of how to create keymaps.

From Program, Setup, Master, or Quick Access Mode

The simplest way to get to the SampleMode page is from Program, Setup, Master, or Quick Access Mode. Press the soft button labelled **Sample** on any of these pages. This is a good method to use if you are making only a couple of samples, or if you want to assign each sample to its own keymap and program. Once you have created and saved your sample, you can press the **Preview** soft button. This button provides a quick way to create a program and keymap, with your sample assigned across the entire range of the keyboard. The program is a one-layer program that uses the settings from the Program **199 Default Program**.

From the Keymap Editor

This is a better method to use if you're going to be doing lots of multi-sampling, or if you need to create custom keymaps in which you have your new samples assigned across the keyboard in one keymap. Call up Program **199 Default Program**. Press **Edit**, then **Keymap**. Select Keymap **168 Silence**, then press **Edit** again. This brings you to the Keymap Editor. (In fact you can choose any program and keymap you want to start with, but by choosing these, you are starting with a "blank slate.") Now from the Keymap Editor, press the **MIDI** mode button. This takes you to the SampleMode page. Once you have created and saved your samples, press **Exit**. You will now return to the Keymap-editor page, where you can immediately assign those samples across the keyboard. Once you have created and saved your keymap, you can either exit the Keymap Editor and create a program that uses your new keymap, or you can return to the SampleMode page for another round of sampling.

Sampling Analog Signals

The K2600's analog sampling input is optimized for a low-impedance line level signal (-10 dBm). With a line-level signal, an input gain setting of 0 dB should prevent any clipping of the sample even at maximum output from the source. You can compensate for lower input levels with the Gain parameter on the SampleMode page.

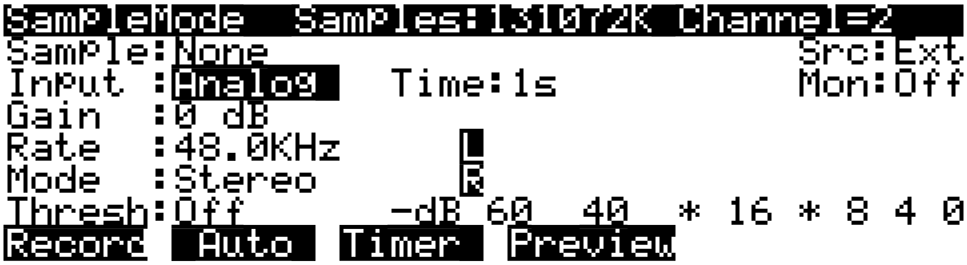
If you're sampling through a microphone, you'll probably want to use a preamp to optimize your signal-to-noise ratio. If you don't have a preamp, you can adjust the Gain parameter on the SampleMode page. A setting of **21 dB** will give you reasonable results for many applications. This will increase the noise level as well, however.

Running your sample signal through a mixer before sending it to the K2600 will give you the most flexibility in controlling your signal level, since you can use its gain or pad if needed. This may add noise to the signal, however. For the cleanest possible signal, you'll want to connect your sample source directly to the K2600. The best results will be achieved by sampling from a digital source, using one of the K2600's digital sample inputs.

Assuming your connections are made, you're ready to set up your first sample recording. Select the SampleMode page (refer to *Entering The Sampler* above). The top line of the SampleMode page gives you the amount of free sample memory, and the amount of free program memory.

Input

On the SampleMode page, you’ll set the conditions for your sample recording. Depending on the input type you select, a different set of parameters will appear on this page. When you’ve selected analog input, the page appears as in the diagram below. The differences between analog and digital sampling are discussed in the section called *Sampling Digital Signals* on page 14-8.



```
SampleMode  Samples:131072K Channel=2
Sample:None                      Src:Ext
Input :Analog   Time:1s          Mon:Off
Gain  :0 dB
Rate  :48.0KHz       L
Mode  :Stereo        R
Thresh:Off      -dB 60  40  * 16 * 8 4 0
Record   Auto   Timer   Preview
```

The digital meters at the lower right of the display give a good indication of your sample level. When you send a signal from your sample source, you should see the meters respond.

Src

The possible values for the Src parameter are Internal (**Int**) or External (**Ext**). Choose a value of **Ext** when you want to sample the signal from an external source that’s connected to one of the K2600’s sampling inputs. Use a value of **Int** if you want to sample the K2600’s own output.

Gain

The meters are calibrated in -dB units. A level of **0 dB** indicates the maximum signal without clipping. The sample will be free of clipping as long as the meter levels don’t exceed 0 dB. For optimum results, you should adjust the K2600’s Gain parameter (or the gain from your sample source) so that the signal stays below 0 dB. Otherwise, the signal will be clipped, causing the loss of sample data, and usually resulting in audible distortion of the resulting sample. A few clips (fewer than 100) may not cause any appreciable distortion. You’ll get the best signal-to-noise ratio with meter levels as close to 0 dB as possible, although you’ll find that samples with maximum meter readings as low as -12 dB can sound remarkably noise-free.

The relatively slow LCD output of the meter levels cannot show every peak in the incoming signal. Therefore, you won’t necessarily see every transient in every sample you take. You will be able to see any transient that is clipped, however, since whenever a clip occurs, the K2600 will display the word “CLIP” above the meters, and will flash the Master-mode LED. It will also give you the number of clips in each sample before you save it.

Rate

After you’ve set your levels, you need to select the sample rate. You have four rates to choose from. The tradeoffs that determine your best sampling rate are frequency response and storage requirements. Higher sample rates capture more frequency content from your samples, but take up more memory. Lower rates give you more sample time, but don’t give the same frequency response as higher rates. Rates of 29.4 or 32 KHz yield a flat response up to about 14 and 15 KHz, respectively. 44.1 and 48 KHz yield a flat response up to 20 KHz, which is the upper limit of audibility for most humans. The lower rates may be adequate for most sounds, since many sounds have little content above 15 KHz. Sounds with a great deal of high-frequency content, such as cymbals, should probably be sampled at the higher rates. You can save memory by using lower sample rates for sounds without much high-frequency content—acoustic or electric bass, for example.

Another consideration in selecting sample rate is the K2600's transposition range during sample playback. The K2600 transposes samples by changing the sample playback rate; the higher the playback rate, the higher the pitch of the sample. The K2600 can achieve a maximum sample playback rate of 96 KHz. Normally, a sample made at 48 KHz can be transposed up a maximum of one octave, since the playback rate doubles for every octave of upward transposition. If you set the SmpSkp (sample skipping) parameter (on the KEYMAP page in the Program Editor) to **Auto** or **On**, you can transpose up two octaves at 48 KHz. A sample made at 29.4 KHz can be transposed up approximately 21 semitones (an octave and a sixth)—or 42 semitones with SmpSkp set to **Auto** or **On**. There is no limit on downward transposition, regardless of the sample rate. See page 6-24 for more information about sample skipping.

Each portion of a sample (each individual sample element made by the K2600 during the sampling process) takes up two bytes of sample memory. A one-second stereo sample at 48 KHz consists of 96,000 individual samples (48,000 x 2), taking up 192,000 bytes (about 188K) of sample memory. The same sample taken at 32 KHz takes up about 125K. A one-second mono sample taken at 32 KHz takes up about 63K.

If you plan to do a lot of sampling, you may want to consider adding sample memory to your K2600. SIMMS (Single In-line Memory Modules) are available at your dealer, or at most computer stores or mail-order houses. Be sure to read *Choosing and Installing SIMMs for K2600 Sample Memory* on page 9-2 of the *Musician's Reference* before you go SIMM shopping, though.

At a sampling rate of 44.1 KHz, each megabyte of sample RAM that you add increases your sample time by about 11.5 seconds (5.5 seconds for stereo samples). At 48KHz, each megabyte gives you about 10 seconds of mono sampling, and about 5 seconds of stereo sampling. Table 14-1 lists a few standard sample RAM configurations and their total sample time capacity (in seconds) at various sample rates.

Total RAM	Sampling Mode	Sampling Rate in KHz				
		29.4	32.0	44.1	48.0	
16M	Mono	4:20	4:16	3:04	2:48	Total Sampling Time (min:sec)
	Stereo	2:16	2:08	1:28	1:20	
32M	Mono	9:20	8:32	6:08	5:36	
	Stereo	4:32	4:16	2:56	2:40	
64M	Mono	18:40	17:04	12:16	11:12	
	Stereo	9:04	8:32	5:52	5:20	
128M	Mono	37:20	34:08	24:32	22:24	
	Stereo	18:08	17:04	11:44	10:40	

Table 14-1 RAM and Sampling Capacity

Mode

Use the Mode parameter to select mono or stereo sampling. (Keep in mind that stereo samples take up twice as much memory as mono samples.) Use a value of **Mono** for a mono signal. You can use either **Mono(L)** or **Mono(R)** to isolate either the left or right side of a stereo signal.

Audio sampling input doubles as a two channel "drum" trigger, allowing audio signals to trigger samples. On the SampleMode page, set Mode to **Trigger**. Adjust Thresh to control triggering sensitivity. This triggers the currently assigned click program. The left input will trigger click key note number +1, right input will trigger click key +2. The click key and click program can be accessed on the Song-mode MISC page.

There’s also Live mode, which lets you connect any audio source to any of the K2600’s sampling inputs (assuming you have the Sampling Option), and use that input just like a regular VAST program (the input goes through a DDP algorithm, then through KDFX, then to the audio outputs). Set Mode to LiveIn to use Live mode. See page 14-41 for more information about Live mode.

Threshold (Thresh)

The Thresh parameter controls when the K2600 actually begins sampling incoming signals. If you set it to a value of **Off**, sampling begins immediately when you press the **Record** soft button. Otherwise the K2600 waits for the incoming signal to exceed a specified threshold before beginning to record. You can set the threshold from **-90** to **0 dB**, in 6 dB increments.

Sampler recording can also be triggered via the keyboard. Set Thresh to **Key**, then press **Auto**. Striking a MIDI note event now will trigger the sampler and assign the sample root to the key you struck, all in one easy step—making sample mapping easy and intuitive.

Time

The Time parameter lets you determine how long the sample will be. The available sample time is a function of the sample rate and the amount of available sample memory. The K2600 calculates this automatically, and limits the maximum value of the Time parameter accordingly. At a value of **0** for this parameter, the K2600 will not record. (Of course, you can always stop sampling before the specified time by pressing the **Stop** soft button.)

Sample

The Sample parameter lets you select any sample in memory for auditioning. This is a convenient way to listen to the samples you’ve made without having to create keymaps and programs for them manually. With a value of **None** for this parameter, the K2600 plays the last program or setup you selected before entering Sample mode. The list of values includes all ROM and RAM samples.

When you select a sample for auditioning, the K2600 automatically creates a temporary keymap and program, based on the settings for Program 199—which is a simple single-keymap program with few controller assignments—and the effects set to **0% wet** (**100% dry**). Any edits you’ve made to Program 199 are reflected in the sample you audition. When you exit the SampleMode page, the temporary keymap and program disappear until the next time you audition a sample. You can create regular RAM keymaps and programs using the **Preview** soft button; see the discussion of the **Preview** button in the section called *Recording Samples* on page 14-6.

If you don’t have enough free program RAM, you may be unable to audition samples, since the K2600 won’t have enough RAM to create the temporary keymap and program. In this case, deleting a few objects from RAM will restore the audition feature.

Monitor (Mon)

The Monitor parameter provides a convenient way to listen to what you’re recording. When set to a value of **On**, any signal received at the analog sample input will appear at the K2600’s Mix outputs and the headphone jack. Adjusting the input gain will affect the monitor gain as well. A clean monitor signal, however, does not guarantee a distortion-free sample. Always check the meters on the SampleMode page and look for the CLIP indicators to ensure that your sample is free of clipping. Note that the Mon parameter is not available when the Input parameter is set to a value of **Digital**. The Monitor feature applies only to the analog sampling inputs. You should monitor digital sources from the sources themselves.

Recording Samples

Press the **Record** soft button to begin the sample recording process. If the Thresh parameter is set to a value of **Off**, recording will begin immediately, and will continue for the number of seconds indicated by the Time parameter. The display will indicate that recording is in process. Any other value for the Thresh parameter will cause the K2600 to wait until the specified threshold is exceeded, then recording will proceed normally. The display will indicate that you're making a threshold recording, but won't actually begin recording until the threshold is exceeded.

End the sampling process (either to save what you've done, or to abort) by pressing the **Stop** soft button.

When recording is complete, and you've pressed the **Stop** button, the K2600 will prompt you to strike a root key. The sample is assigned to the key you strike. This "root" is the key at which the sample will be played back without transposition. When sampling pitched sounds, it generally makes sense to assign a root key that matches the pitch of the original sample, although you can set the root key anywhere you like. If you press the **Default** soft button, the K2600 uses C 4. You can change the root key at any time on the MISC page in the Sample Editor.

When the root key has been assigned, the K2600 asks you if you want to save the sample. At this point the display will show one of two things—the number of clips, or if no clips occurred, the maximum level (in dB) of the sample signal.

You can listen to the sample before deciding whether to save it. If you decide not to keep the sample, press the **No** soft button, and you'll return to the SampleMode page. If you press **Yes**, you'll see the normal Save dialog. When you've saved the sample, you'll return to the SampleMode page. You'll also have the opportunity to name the sample. A recommended convention for naming samples is to include the root key as part of the name. This is particularly useful for pitched samples. Including the root key in the sample name helps when you are creating a keymap, because it tells you how much transposition of the sample you will hear depending on its key assignment.

Once the sample is recorded and saved, you may want to edit it, using the TRIM page, LOOP page, or any of the sample DSP functions.

The Auto Soft Button

To save time when sampling with either the analog or digital inputs, you can use the **Auto** soft button. If the Thresh parameter is set to **Off**, sampling begins immediately. Once sampling is complete and you've pressed Save, the K2600 automatically assigns a root key of C 4, and saves the sample to the first available ID above 199.

If Thresh is set to a dB value, sampling begins when the incoming signal exceeds the Thresh level. If Thresh is set to **Key**, sampling begins when you strike a key.

Auto sampling is useful when you're making a series of samples that you expect to have the same approximate signal level. Since auto sampling doesn't show you the maximum signal level or the number of clips in the sample, it's a good idea to make your first sample in the series using the **Record** button. Once you have the input signal at the right level, you can make the rest of the samples in the series with fewer button presses.

The Timer Soft Button

If you need to delay the beginning of your sample recording, you can press the **Timer** soft button instead of the **Record** or **Auto** soft buttons. This will begin a ten-second countdown before sample recording actually starts. The display will show the countdown. When the countdown reaches zero, The Program, Setup, MIDI, and Master-mode LEDs will flash three times.

If you have the Thresh parameter set to a value of **Off**, sample recording begins immediately after the LEDs flash. If you have the Thresh parameter set to a dB value, sampling begins when the incoming signal exceeds the Thresh level. If Thresh is set to **Key**, sampling begins when you strike a key.

The Preview Soft Button

When you've finished taking a sample, you can press the **Preview** soft button to automatically create a keymap and program using the new sample. It uses the settings for the Program **199 Default Program** as a template. Unlike the temporary keymap that's created when you audition a sample (and disappears when you select another sample), the preview keymap and program are stored in RAM and can be selected at a later time. The program and keymap will have the same name as the sample.

When you press the **Preview** soft button, the Bank dialog appears, prompting you to select a bank where the preview program will be stored. Select a bank, then press the **OK** soft button. The K2600 creates a keymap and a program, using the lowest available ID numbers in that bank for both the keymap and the program. The display tells you the ID of the new program.

Multiple Sample Previews

The **Multi** soft button starts a process that lets you automatically build a program for previewing just about as many samples as you want (104 to be exact).

1. Press **Multi**. You'll see a list of available samples (keep in mind that these sample objects may consist of multiple sample roots). This list of sample objects is another version of the multiple object selector described on page 13-36.

2. Use the **Up/Down** cursor buttons and the **Select** soft button to highlight and select sample objects. The asterisk that appears indicates that the sample is selected. If you don't select any, the K2600 assumes you want to preview them all.

3. Press **OK**. If you've selected more than one sample, the K2600 asks you if you want to combine the sample objects into a single keymap and program. (If you've selected only one sample —one that doesn't consist of multiple sample roots—the K2600 returns to the Bank dialog, where you can select a different bank if you want, then press **OK**. The K2600 creates a keymap and program, tells you what the ID of the program is, and returns to the SampleMode page.)

4. At the "Combine into…" prompt, press **Yes** The K2600 asks you if you want a tuned layout. (If you press **No**, at the "Combine into…" prompt, you return to the Bank dialog, where you can select a different bank if you want, then press **OK**. The K2600 creates a keymap and program for *each sample root*. If you're previewing a number of sample objects that consist of multiple sample roots, the list of programs can get quite long. In most cases it's much more convenient to combine the samples into one program.

5. At the "Tuned Keymap Layout" prompt…decide how you want the sample objects to be laid out in the preview program. Pressing **Yes** maximizes the use of the keyboard. First you'll see the Bank dialog again. Press **OK**, and the K2600 takes the list of sample objects you selected in Step 2, and in order of their IDs, starts assigning them to their normal root keys. If two or more samples use the same root key(s), the most-recently assigned sample gets assigned to the next highest available key, and its coarse tune is adjusted so it plays at its root pitch. When all the roots are assigned to keys, the K2600 fills in between the roots,

so all keys play one of the samples. The number of keys playing each sample depends on the total number of sample roots you’re previewing. Tuned layouts are useful for previewing pitched samples.

Pressing **No** at the “Tuned Keymap Layout” prompt is useful for previewing large numbers of samples, or percussion samples. When you press **No**, you’ll see the Bank dialog. Press **OK**, and the K2600 takes the list of sample objects you selected in Step 2, and in order of their IDs, starts assigning them to keys, beginning at C 2, one root per key. The coarse tune gets adjusted so they all play at their root pitches.Keys below C 2 play the sample assigned to C 2, transposed accordingly. Above the highest key used, you’ll hear the sample with the highest key assignment, transposed accordingly up to the upper transposition limit.

In either case, after the K2600 finishes processing the samples, it tells you the ID of the preview program (or the lowest ID if it created more than one program), then returns to the SampleMode page, with the preview program as the current program.

Sampling the K2600’s Output

You can sample the K2600’s own sounds when in Analog sampling mode. To do so, set the Src parameter on the SampleMode page to a value of **Int**. Then, just press the **Record** soft button and start playing.

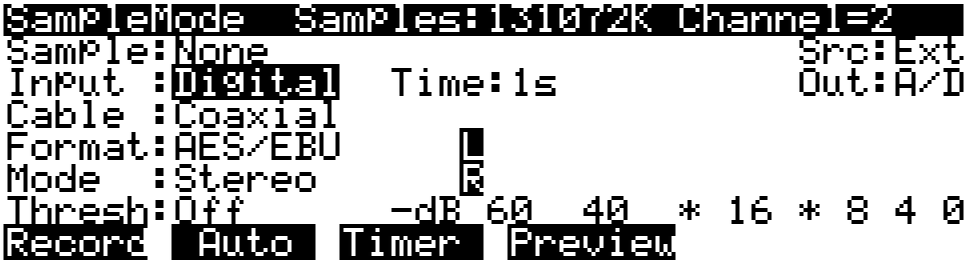
The K2600’s “sample-while-play” capabilities offer a number of useful possibilities. It allows you, for example, to create composite sounds made up of several K2600 sounds or even sequences. This can help you make efficient use of the K2600’s polyphony. By building composite sounds from other composite sounds, you could actually cause a frighteningly large number of K2600 sounds to become a single sample. The only constraints are your imagination—and the amount of sample RAM installed in your K2600.

You can also sample the K2600 directly into songs, using the RAM Tracks feature. See page 12-17.

Sampling Digital Signals

The process for sampling through either of the digital inputs is essentially the same as that for sampling analog signals, although there are a few additional parameters associated with digital sampling formats.

You’ll notice that the SampleMode page changes considerably when you change the value of the Input parameter from **Analog** to **Digital**. There are a few more settings to be made before you start recording. Note that the Src and Out parameters appear only if you have the both the sampling option and the digital I/O option. We’ll discuss the Out parameter in more detail on page 19-3.



```
SampleMode  Samples:131072K Channel=2
Sample:None                       Src:Ext
Input :Digital  Time:1s           Out:A/D
Cable :Coaxial
Format:AES/EBU       L
Mode  :Stereo        R
Thresh:Off      -dB 60  40  * 16 * 8 4 0
Record  Auto  Timer  Preview
```

The first difference is the fact that there are no parameters for gain and sample rate. There's no need for a gain parameter because with digital sampling, since you're making an exact digital copy of the source signal. The Rate parameter is excluded because the K2600 automatically recognizes the source sample's rate and sets its own rate accordingly. Also, the Mon parameter does not appear when sampling digitally. Any monitoring you wish to do must be done from the sample source.

Cable

Set the Cable parameter to a value of **Coaxial** or **Optical**, depending on the type of cable you're using. Many consumer products with digital output provide an optical jack. There is a small plug covering the K2600's optical input; this plug must be removed before you can connect the optical cable to the K2600. The plug should be replaced whenever the optical input is not in use.

If you plan to use a coaxial cable, select a value of **Coaxial** for the Cable parameter. If you're sampling from a consumer product like a CD player, you'll probably need to get an adaptor, or a cable with an RCA fitting on one end and an XLR (male) fitting on the K2600 end. Some professional products have an XLR fitting for their outputs, so an ordinary microphone cable will be suitable in most cases. Longer cables can cause signal loss, however, so if you need a long cable, you may need to get a special cable designed for digital information transfers. There's detailed information about cables for sampling on page 14-1.

Format

Use the Format parameter to tell the K2600 the format of the incoming sample. Most consumer products use SPDIF (Sony/Philips Digital Interface Format), while most professional machines use the AES/EBU (Audio Engineering Society/European Broadcast Union) format. Refer to the owner's manual of your sample source for information regarding its digital format.

The Mode, Time, and Thresh parameters function for digital sampling just as they do for analog sampling.

If the K2600 detects an incoming clock signal, the display shows LOCK, and the sample rate of the signal. If you don't see LOCK, you're not getting signal, and you won't be able to sample. The K2600 automatically sets itself to the clock rate it detects. (For sample rates other than 48, 44.1 and 32, the rate doesn't show in the display, but the K2600 still samples the input correctly. You might need to adjust the coarse tune to get the proper root pitch.)

Src

When Input is set to **Digital** on the SampleMode page, the Src parameter appears only if you also have the digital I/O option. It's the same parameter that appears when Input is **Analog**. Use a value of **Int** for Src when you want to sample the K2600's Mix output. Use **Ext** when you want to sample an external source via the coaxial or optical stereo digital input jacks.

If you have the digital I/O option, Src has another value: Return (**Rtn**). This enables you to sample a two-channel return from KDS Out A (this requires a Kurzweil DMTi or other device that's compatible with KDS digital format).

Out

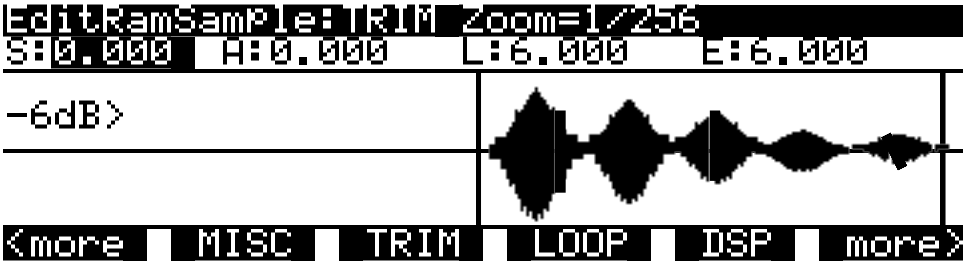
This parameter appears only if you also have the digital I/O option. It doesn't really have anything to do with sampling; it determines how the K2600 handles its digital audio output. See Chapter 19 for information about using the digital audio outputs.

Editing Samples

Most of the functions within the Sample Editor follow a general pattern. There are two ways to enter the Sample Editor. If you start from Master mode and press the **Sample** soft button, then select a sample and press **Edit**, you can hear the isolated sample. If you want to hear the sample in the context of a program, enter the Sample Editor through the Program Editor: start by selecting a program in Program mode—usually the program containing the sample you want to edit. Press the **Edit** button to enter the Program Editor. Press the **KEYMAP** soft button to view the KEYMAP page. The KeyMap parameter is selected (highlighted) when the page appears. Press the **Edit** button to enter the Keymap Editor. The KeyRange parameter is selected when the page appears. The notes within the currently selected key range are the only ones that will be affected by your edits. You can hold the **Enter** button and trigger notes to select different key ranges.

If you want to select a different sample, use the cursor buttons to select the Sample parameter. Use the Alpha Wheel to select a sample. Press the **Edit** button once more, and you'll enter the Sample Editor. (Pressing the **Edit** button while in the Keymap Editor will enter the Sample Editor regardless of which parameter is selected.) The effects of the current program will be applied to the sample.

The TRIM page appears when you enter the Sample Editor. A representative TRIM page is shown below.



EditRamSample:TRIM zoom=1/256
S:0.000 A:0.000 L:6.000 E:6.000
-6dB>

There are three basic sample editing pages—TRIM, LOOP, and MISC (Miscellaneous). The soft buttons for these pages are visible when you enter the Sample Editor. The **DSP** soft button is visible as well if you're editing a RAM sample; pressing it will take you to the DSP function page, where you can select a DSP function with the Alpha Wheel or **Plus/Minus** buttons.

The **DSP** soft button does not appear if you're viewing a ROM sample. Instead you'll see a **Link** soft button.

The **<more>** soft buttons will take you to the soft buttons for the other functions. You can trigger notes at any time while you're editing, to hear your changes as you make them.

The Function Soft Buttons in the Sample Editor

In addition to the **MISC**, **TRIM**, and **LOOP** soft buttons, which select Sample-editor pages, there are several function soft buttons. As with other K2600 editors, the function soft buttons are labeled with upper and lower case letters, to distinguish them from the page selection soft buttons, which are labeled in all capital letters. The **<more>** soft buttons give you access to the other soft buttons that are available.

Zoom- and Zoom+

These buttons are active only when you're viewing the TRIM and LOOP pages. They increase or decrease the resolution of the waveform display, enabling you to see a larger or smaller segment of the waveform of the currently selected sample. The top line of the display indicates the zoom position in terms of a fraction—for example, 1/256—which indicates the number of individual sample elements represented by each display pixel. A value of 1/256 means that each pixel represents 256 individual sample elements. The maximum zoom setting of 1 shows you a very small segment of the sample. The minimum setting of 1/16384 shows you the largest possible segment of the sample. Each press of a **Zoom** soft button increases or decreases the zoom by a factor of 4.

As a convenience, the **Program** and **Setup** mode buttons also serve as zoom buttons while in the Sample Editor. You can press the two left soft buttons together to toggle between the default zoom setting and your current zoom setting.

Gain- and Gain+

Also active only for the TRIM and LOOP pages, these buttons increase or decrease the magnification of the currently displayed sample waveform, enabling you to see the waveform in greater or lesser detail. At the left of the display, you'll see the magnification setting, which is expressed in dB units. You can adjust the magnification from -72 dB (maximum magnification) to 0 dB. This doesn't affect the actual amplitude of the sample, only the magnification of its display.

As a convenience, the **MIDI** and **Master** mode buttons also serve as gain adjustment buttons while in the Sample Editor.

The simplest way to think of the **Zoom** and **Gain** buttons is to remember that the **Zoom** buttons control the left/right magnification of the waveform, while the **Gain** buttons control the up/down magnification. Neither button has any affect on the sound of the sample. You'll often use the **Zoom** and **Gain** soft buttons together to focus in on a particular sample segment, then magnify it to see it in close detail.

For example, you might want to zoom out to view an entire sample waveform, to decide which segment you want to edit. You could then zoom in to focus on a particular segment. Once you've zoomed in, you may want to boost the Gain to enable you to set a new Start (S) point with greater precision, or ensure that you get a smooth loop transition.

Abort

Use the **Abort** soft button to cancel a sample dump before it's complete. You'll be prompted to verify whether you really want to cancel the dump.

Split

The **Split** soft button enables you to create two mono samples from a single stereo sample, or to split up a multi-root block of samples. When you press this button, the K2600 will prompt you: Split this sample? When you press the **Yes** soft button, you'll be prompted to enter the ID for the first sample. Select an ID with the Alpha Wheel or **Plus/Minus** buttons, then press the **OK** soft button. If you've selected a stereo sample, the K2600 splits it into left and right sides. If it's a block of samples, the K2600 splits it into individual sample roots. In either case, the split samples will automatically be assigned IDs, starting with the ID you select.

Splitting stereo samples enables you to use the separate sides individually, or to phase the samples by assigning each side to a separate keymap, then delaying one of the layers slightly.

Join

You can create sample objects that contain multiple sample roots. Many of the ROM samples are like this—multiple samples stored in memory as a single sample object. Joining samples is a great way to cut down on the number of IDs that you use for your samples, since joined samples all use the same object ID.

1. Press the **Split** soft button, then press the **Join** soft button. A list of RAM samples appears. This is another version of the multiple object selector, as described on page 13-36.
2. Use the **Up/Down** cursor buttons and the **Select** soft button to highlight and select samples (don’t mix mono and stereo samples). The asterisk that appears indicates that the sample is selected.
3. Press **OK**. If the K2600 asks you “Are you sure?” press **Yes**. You’ll return to the Sample Editor.
4. Save the sample. We recommend using a new ID. Every sample root in the joined sample uses this ID. The names and note numbers are different. When you save the sample, the K2600 will ask you if you want to copy the sample data. There’s no need to do this; it will only take up additional memory. Everything works properly if you don’t copy the data, and when you save your new sample to disk, everything gets saved accordingly.

When you return to the SampleMode page (or if you’re looking at the Sample parameter in the Keymap Editor), you can scroll through the list of samples, and see your newly-created sample block. Notice the names and/or note numbers changing, while the ID remains the same throughout each sample block.

A Few Notes About Joined Samples

Don't try to join stereo and mono samples; the K2600 can join samples only if they’re the same type.

You can't use any of the Sample Editor’s DSP functions on joined samples; make sure to process each sample before joining.

Units

With the **Units** soft button you can change the units used to display the locations of the current sample’s Start, Alt, Loop and End points. The default setting displays these points in seconds, that is, the number of seconds from the physical start of the sample. Pressing the **Units** soft button will change the units to samples—that is, the number of individual sample elements from the physical start of the sample. Press it again to return to a view of the sample in seconds.

As a convenience, the **Quick Access** mode button also serves as a units button while in the Sample Editor.

Link

The **Link** soft button lets you fix the interval between the Start, Alt, Loop and end points of the current sample, so it remains constant when you move one or more of the points. This is done by selecting the desired parameter with the cursor buttons, then pressing the **Link** soft button. The colon (:) following the parameter’s name will change to an arrow (>) to indicate that it is linked. You can link any or all of the four sample points. When sample points are linked, moving one of them will move the linked points correspondingly. For example, suppose the current sample’s Start (S) point is 0.0 seconds, and its Alt (A) point is 0.5 seconds. The interval between the sample’s Start and Alt points is exactly half a second. If you select the Start parameter, then press the **Link** soft button, the Start point will be linked. This won’t have any effect until you link at least one more point. If you select the Alt parameter and press the **Link** soft button, the Start and Alt points will be linked. Now if you move the Start point to 1.0 seconds, the Alt point will automatically move to 1.5 seconds, preserving the half-second interval between Start and Alt. To remove the link on any of the points, select the point again, and press the **Link** soft button again. The arrow will change to a colon, indicating that the link has been removed.

When editing RAM samples, you can't assign Start, Alt, Loop, or End points that extend beyond the boundaries of the actual sample data (you *can* do this with ROM samples). Consequently, if you attempt to move any of the linked points of a RAM sample beyond the physical start or end of the sample, the K2600 changes the interval between the points accordingly. For example, suppose you have a sample with Start and Alt linked. Start is at 0 seconds, and Alt is at 1 second. If you move Alt to a later point, Start tracks with it, so the interval between Start and Alt remains at 1 second. If you move Alt to .5 seconds, the K2600 tries to move Start accordingly, but can't move it earlier than 0 seconds, so the interval between Start and Alt shrinks to .5 seconds. Even if you move Alt back to 1 second, the interval remains at .5 seconds. You'd have to reset Start manually to restore the original 1-second interval.

As a convenience, the **Song** mode button also serves as a link button while in the Sample Editor.

Name, Save, Delete, and Dump

These soft buttons are similar to the **Name**, **Save**, **Delete**, and **Dump** soft buttons in the other editors, initiating the corresponding dialogs to name, save, delete, or dump the currently selected sample.

When you press the **Save** soft button, and choose to save the sample to a new ID, the K2600 will ask you if you want to copy the sample data. If you answer Yes, the K2600 will make a separate copy of the sample. If you answer No, the K2600 will simply mark the location of the original sample data and share the sample between the original and the edited sample. This can save a great deal of memory space. If you delete a sample that's partially or completely shared with another, the K2600 deletes only the portions that are unused by the shared sample, always optimizing its memory for maximum storage capacity.

Note that if you use the Utility function to view the objects currently stored in the K2600, you'll see each object listed separately, including shared samples. The shared samples will each indicate their size, even though they're referring to the same memory location. This might lead you to believe that you're using more memory for samples than you actually are. If you use the Utility function to calculate your total sample memory usage, remember not to include any shared samples in the total.

Dump lets you transfer (dump) samples using the MIDI sample dump protocol. See Chapter 6 of the *Musician's Reference* for more information.

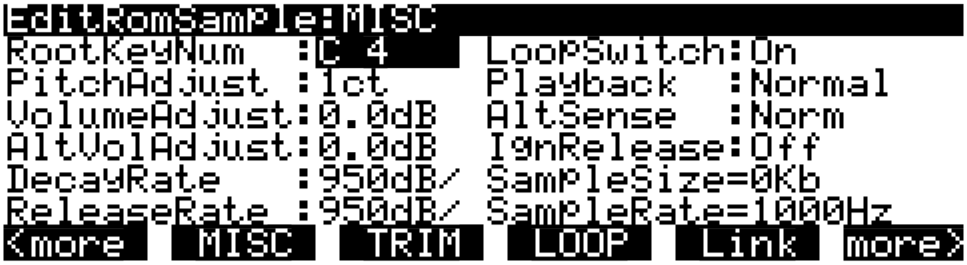
The Page Buttons

The soft buttons labeled in all capital letters select the various pages in the Sample Editor: **MISC**, **TRIM**, and **LOOP** (as well as **DSP** when you're editing RAM samples).

The Miscellaneous (MISC) Page

On the MISC page, you'll set several parameters that affect the behavior of the current sample. These parameters affect the entire sample. The diagram of the MISC page shows a ROM sample, and since the DSP functions cannot be applied to ROM samples, the **DSP** soft button is not available. In its place is the **Link** soft button, which enables you to maintain equal times between various points in the sample—Start and Alt, for example.

The default values shown in this diagram reflect the settings for the Default program 199.



```
EditRomSample:MISC
RootKeyNum  :C 4     LoopSwitch:On
PitchAdjust :1ct     Playback  :Normal
VolumeAdjust:0.0dB   AltSense  :Norm
AltVolAdjust:0.0dB   IgnRelease:Off
DecayRate   :950dB/  SampleSize=0Kb
ReleaseRate :950dB/  SampleRate=1000Hz
<more  MISC  TRIM  LOOP  Link  more>
```

Parameter	Range of Values
Root Key Number	C -1 to G 9
Pitch Adjust	Variable (depends on sample rate)
Volume Adjust	-64.0 to 63.5 dB
Alternative Volume Adjust	-64.0 to 63.5 dB
Decay Rate	0 to 5000 dB per second
Release Rate	0 to 5000 dB per second
Loop Switch	Off, On
Playback Mode	Normal, Reverse, Bidirectional
Alternative Sample Sense	Normal, Reverse
Ignore Release	Off, On

Root Key Number

The root key number represents the key at which the sample will play back without transposition (that is, at the same pitch as the pitch of the original sample). When you're creating your own samples, the key you strike as the root key will be the key you see as the value for this parameter.

Pitch Adjust

Use this parameter to change the pitch of the sample relative to the key from which it's played. Setting a value of **100cts**, for example, will cause the sample to play back one semitone higher than normal. This parameter is handy for fine tuning samples to each other if they're slightly out of tune.

Volume Adjust

Uniformly boost or cut the amplitude of the entire sample. Compare this to the DSP Volume Adjust parameter, which lets you boost or cut the amplitude of a specified segment of a RAM sample.

Alternative Start Volume Adjust (AltVolAdjust)

This parameter sets the amplitude of the sample when the alternative start is used. See page 6-26 for a discussion of AltSwitch.

Decay Rate

This parameter defines how long the sample takes to decay (fade) to zero amplitude (silence). Decay Rate affects each sample individually, and is in effect only when the amplitude envelope for the program (the Mode parameter on the AMPENV page in the Program Editor) is set to **Natural**. If Mode is **User**, the settings on the AMPENV page override the setting for DecayRate.

DecayRate operates in two different ways, depending on the nature of the sample. If it's a ROM sample or a Kurzweil-format RAM sample, DecayRate takes effect in the loop portion of the sample, after all the attack stages of the amplitude envelope are complete. For non-Kurzweil RAM samples, DecayRate takes effect as soon as the sample starts, regardless of the length of the attack stages.

Release Rate

The release rate determines how long the sample will take to decay to zero amplitude when the note trigger is released. The higher the value, the faster the release rate.This release affects each sample individually, and is in effect only when the amplitude envelope for the program (the Mode parameter on the AMPENV page in the Program Editor) is set to **Natural**. In this case, the release begins as soon as the note is released.If Mode is **User**, the settings on the AMPENV page override the setting for ReleaseRate.

To create an extended sample loop that will play data after the sample's loop on key-up, set the Alt sample pointer after the sample end pointer, then set a relatively low value for the release rate.

Loop Switch

This parameter activates or deactivates the looping of the currently selected sample. When set to **On**, the sample will loop according to the settings on the LOOP page. When set to **Off**, the sample will play through to its End point and stop.

Playback Mode (Playback)

This parameter lets you modify the direction in which the sample is played. Set it to a value of **Reverse** if you want the sample to play from its End (E) point to its Start (S) point. Choose a value of **Bidirectional** to cause the sample to play from Start to End, then reverse direction and play again from End to Loop and back, repeating until the note trigger is released (this works only when the Loop Switch parameter is set to **On**).

Alternative Sample Sense (AltSense)

This provides a convenient way to activate the alternative start of a sample. When set to **Normal**, the alternative start will be used when the Alt Switch control is **On** (this is set on the KEYMAP page), or when the control source assigned to it is above its midpoint. When set to **Reverse**, the alternative start will be used when the Alt Switch control is **Off**, or when the control source assigned to it is below its midpoint.

Ignore Release (IgnRelease)

When set to a value of **Off**, the sample will release normally when the note trigger is released. When set to **On**, the note will not release, even when the note trigger is released. This setting should be used only with samples that normally decay to silence; nondecaying samples will play forever at this setting. This parameter is equivalent to the IgnRelease parameter on the LAYER page, but affects only the currently selected sample.

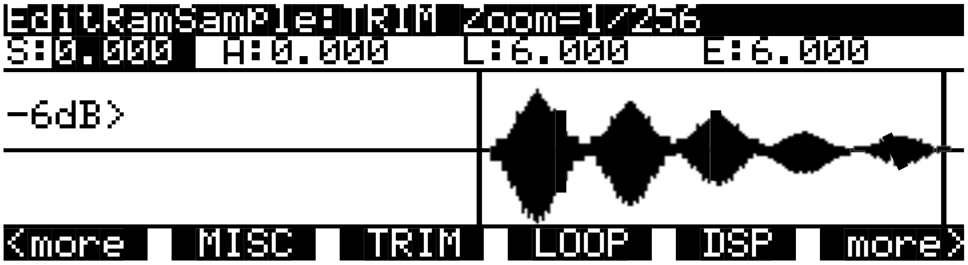
Sample Size and Sample Rate

These are not editing parameters. They're on this page to give you a convenient place to check the size in kilobytes of the current sample, and the rate at which it was sampled.

The TRIM Page

The TRIM page lets you set the Start, Alt, Loop, and End points of the current sample. The top line tells you whether you're editing a ROM or RAM sample, and indicates the zoom setting. At the left of the display is the Gain (display magnification) setting. This Gain setting doesn't affect the amplitude of the sample, just the view in the display.

In the diagram below, the sample points are expressed in individual sample elements. Pressing the **Units** soft button (or the **Quick Access** mode button) will display them in seconds.



EditRamSample:TRIM zoom=1/256

S:0.000 A:0.000 L:6.000 E:6.000

The four parameters on this page are Start (S), Alternative Start (A), Loop (L), and End (E). Selecting these parameters and adjusting their values enables you to modify how the sample plays back when notes are triggered.

There are four vertical lines that indicate the settings of the four parameters. You'll see all four lines only if the values for each of the four parameters are different; otherwise, the lines overlap. Selecting one of the parameters moves the line corresponding to that parameter to the center of the display, where it flashes to indicate the parameter's position. Turning the Alpha Wheel will move the sample waveform relative to the line. The line remains in the center of the display, and the waveform shifts to indicate the new position of the point. You can also use the alphanumeric buttonpad to enter new values directly. Press the **Enter** button to register the values you enter.

The Start (S) point determines the beginning of the current sample. You can truncate the beginning of the sample by increasing the value of the Start (S) parameter. You might do this to remove silence at the beginning of a sample, or to remove some or all of the attack. You can't decrease the Start point of RAM samples below zero, but if you want to add silence at the beginning of a RAM sample to create a delay, you can use the Insert Zero DSP function to add as much silence as you like. The start points of ROM samples can be set lower than zero (you can set negative numbers for the Start (S) and Alt (A) parameters). Doing this will cause portions of other samples to be played, which can create interesting effects.

Note that for RAM samples, you won't see any waveforms displayed to the left of the Start point, but you will for ROM samples.

The Alt (A) parameter lets you set a second, optional start or end point for the current sample. The Alt will be used when the Alt Switch parameter on the KEYMAP page is set to **On**, or when it's set to a specific control source and that control source is generating a value of more than +.5. (For example, if you assign **MWheel** as the control source for the Alt Switch parameter, the Alt will be used when the Mod Wheel—or whatever control source you have set to send MWheel—

is above its halfway point.) The Alt can be set before, after, or at the same point as the Start or End.

If you set the Alt after the End, you can extend the play of looped samples. Normally, looped samples will play through to the End, then will loop back to the Loop point, and continue looping like this until the note is released, when they go into their normal release. If the Alt is set after the End, looped samples will loop in the same way while notes are sustained. As soon as you release the notes, however, the samples will play through to the Alt point before going into release.

The Loop (L) parameter sets the beginning of the looped portion of the current sample. Although you can adjust this parameter while you're on the TRIM page, you'll normally want to adjust it while viewing the LOOP page, so you can see the loop transition points, enabling you to create as smooth a loop as possible. The Loop can be set at any point before the End, including before the Start and Alt. If you try to move it after the End, the End will move with it.

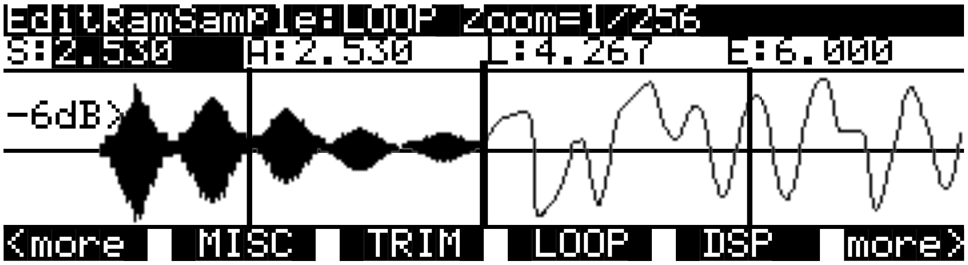
The End (E) parameter sets the point at which the current sample will stop playback. Typically you'll use this parameter to trim unwanted silence off the end of a sample, although you can use it to shorten a sample as much as you want.

If you want to truncate a sample to save memory, there are two points to keep in mind. First, if the Alt parameter is set before the Start, you won't save any memory by truncating the Start. Likewise, you won't save memory by truncating the End if the Alt is set after the End. You won't reclaim memory by truncating a sample until you save the sample and exit the Sample Editor.

You can also use the Truncate DSP function to automatically truncate your samples at a specified noise floor. As with the TRIM page, you'll recover memory after saving the sample and exiting the Sample Editor.

The LOOP Page

The LOOP page features the same four parameters as the TRIM page, but the waveform display is quite different. The best way to understand what you see on the LOOP page is to switch back and forth between the TRIM and LOOP pages and study the waveform displays.



EditRamSample:LOOP zoom=1/256
S:2.530 A:2.530 L:4.267 E:6.000

On the TRIM page you see the entire waveform—or as much of the waveform as your current zoom setting allows. When you move to the LOOP page, you'll notice that the page is split into two sections, left and right, divided by a vertical bar in the center. This bar is thicker than the vertical lines representing the Start, Alt, Loop, and End points, and does not move when you adjust any of these points.

To the left of the dividing bar you see the same segment of the current sample that you see on the TRIM page. The four vertical lines representing the Start (S), Alt (A), Loop (L), and End (E) points are visible. (Remember, you'll see all four vertical lines only if the values for the Start, Alt, Loop, and End parameters are different.) To the right of the dividing bar you see the entire loop segment of the sample.

In the center of the loop segment is a dotted vertical bar that represents the loop transition point—that is, the point at which the sample reaches its End point and loops back to the Loop point. You can visualize the loop segment by starting at the vertical transition point; this is the beginning of the loop, as defined by the setting for the Loop parameter. The waveform progresses to the right, representing the initial portion of the loop segment. The waveform "disappears" off the far right side of the display, and "reappears" at the thick dividing bar at the center of the display. The waveform again progresses to the right, representing the final portion of the loop segment. It reaches the dotted vertical transition line, representing the End point of the sample, where it jumps once again to the loop point and repeats the cycle.

If you select the Loop (L) parameter and change its value, you'll see the segment of the waveform to the right of the transition point shift its position. If you select the End parameter and change its value, you'll see the segment of the waveform to the left of the transition point shift its position.

When you're setting a loop segment for a sample, you'll want to adjust both the Loop and End parameters so the two ends of the waveform meet (or come as close as possible) at the transition point. You'll notice an audible click in your sample loop if the ends of the waveform do not meet at the transition point. The closer you can get the two ends of the waveform, the better the sound quality of your loop will be. With a bit of experimentation, you'll develop the ability to create smooth loop transitions.

You'll also want to try to set the loop point at a zero-crossing—a point where the sample waveform is neither positive or negative. Pressing the **Plus/Minus** buttons together will search (from left to right) for the sample's next zero-crossing. You can usually press these buttons several times for any given sample without noticeably affecting the sound of the sample. If you press the **Minus** button, you'll reverse the direction of the search, and the next time you press the **Plus/Minus** buttons together, the K2600 will search for the next zero-crossing to the left. Press the **Plus** button again to search toward the right.

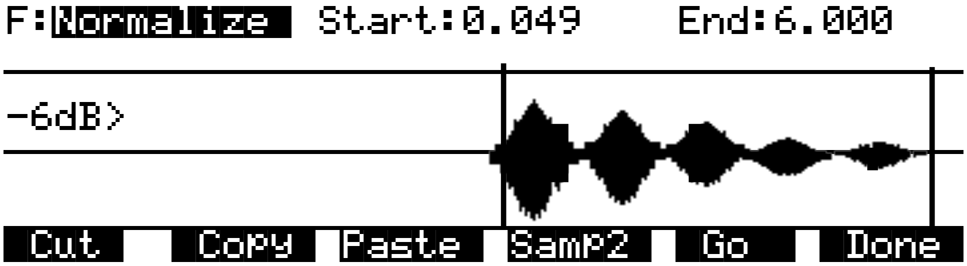
***Note**: The Gain affects the sensitivity of the zero-crossing search algorithm. The higher the gain, the more zero crossings it will find. At low gain settings, the algorithm will fail to find many or even all zero crossings.*

If you adjust the display Gain and Zoom of the sample while on the LOOP page, you'll notice that the Gain affects the waveform on both sides of the loop point, while the Zoom affects only the left side of the page. You can't zoom in on the loop transition point in the right half of the display.

You can also use the crossfade loop (XfadeLoop) DSP function to get a smooth transition between loop points. As with the TRIM page, you'll recover memory after saving the sample and exiting the Sample Editor.

The DSP Page (RAM samples only)

Select the DSP page with the **DSP** soft button. This gives you access to a long list of nonreal-time DSP functions, with which you can modify your RAM samples. The first time you select a DSP function, you'll see the Normalize function, shown below. Afterward, the most recently selected DSP function will appear when you select the DSP page.



F:Normalize Start:0.049 End:6.000

-6dB>

All of the DSP functions operate on a segment of the current sample that you select before executing the function. In most cases, you'll use the Start and End parameters to define the start position and end position of the segment you want to modify. There are a few exceptions to this rule, which will be explained as applicable.

Please keep in mind that the Start and End parameters on the DSP pages are not the same as the sample Start (S) and End (E) parameters that you set on the TRIM and LOOP pages. When you're working on one of the sample DSP functions, Start and End position refer to the range of the sample that you want to process. Adjusting these parameters does not affect the overall start and end of the sample. It affects only the portion of the sample that you want to process. When you audition the sample by triggering a note, you'll hear only the range of the sample within the Start and End parameters on the current DSP page. To hear the entire sample, press the **Done** soft button to return to the Sample-editor page. Similarly, moving the S and E parameters on the TRIM or LOOP pages have no effect on the Start and End parameters on the DSP pages.

Use the **Plus/Minus** buttons or the Alpha Wheel to select the starting and ending positions of the selected sample segment (the Start and End parameters). You can audition the selected sample segment by triggering any note within the current key range.

The actual processing of the sample begins when you press the **Go** button. The K2600 will display a row of scrolling dots while it's processing. When it's finished, it will prompt you: Keep these changes? Trigger a note to hear the result. Press the **No** soft button to return to the page for the currently selected DSP function. This will undo any changes you've made. Press the **Yes** soft button to make the change. The Save dialog will appear (except for the Truncate function). You can save to a different ID than the one displayed, if you want to preserve the original sample. Pressing the **Cancel** soft button will return you to the current DSP page without saving the sample (the sample reverts to its original condition).

Six of the DSP functions (Mix, Mix Beat, Mix Echo, Insert, Replicate, and Beat Volume Adjust) involve selecting a second sample segment (Sample 2) to be processed with the currently selected sample. In these cases, use the **Samp2** soft button. When you press it, another page appears, enabling you to select a second RAM sample using the Alpha Wheel or the alphanumeric buttonpad. You can audition Sample 2 by triggering a note.

Once you've selected Sample 2, use the Start and End parameters to define the start and end positions of the segment of Sample 2 that you want to process. While you're on the Samp2 page, you'll hear the currently selected Samp2 when you trigger a note. When these positions are defined, press the **OK** soft button to return to the DSP page where you can continue your editing.

Once you've selected a sample on the Samp2 page, it remains selected until you return to the Samp2 page and select another sample. You can always use the Samp2 page to audition a second sample.

You can use the **Copy** soft button on the Samp2 page to copy the selected segment to a buffer, then paste the segment into Sample 1 when you return to the DSP page.

The Soft Buttons on the DSP Page

Cut

This button will cut (remove) the currently selected sample segment from the currently selected sample, and store it in a buffer. This is equivalent to cutting a section out of a piece of audio tape and splicing the remaining ends together. The cut segment is then available for pasting elsewhere in the sample. The segment you cut will remain in memory until you replace it with another cut or copy command, or until the K2600 is shut off. If you accidently cut a sample segment, you can restore it by immediately pressing the **Paste** soft button.

Compare this function to the Delete DSP function, which removes the selected sample segment, but doesn't store it in a buffer. If you delete a sample segment, it's *gone*.

Copy

Use this button to store the selected sample segment in a buffer without altering the current sample. The copied segment will remain in memory until you replace it with another cut or copy command, or until the K2600 is shut off.

Paste

This button has an effect only after you've cut or copied a sample segment using the Cut or **Copy** soft buttons. The **Paste** soft button inserts the contents of the Cut/Copy buffer after the start position of the currently selected sample segment. This is like splicing a section of audio tape into another section; it extends the length of the sample. You can undo a paste by pressing the **Cut** soft button immediately after pasting.

When you cut or copy a sample, it's stored in a buffer. Whatever is stored in the buffer will remain available for repeated paste commands until replaced by another cut or copy command, or until the K2600 is shut off or reset (soft or hard). The buffer is also cleared if you execute a master delete, and select "Delete everything."

***Note**: Cutting and pasting always "save" the resulting sample, so you can't use the exit-without-saving short cut to undo a cut or paste operation.*

Samp2

The **Samp2** button, as described earlier in the DSP page section, enables you to select a segment of a second RAM sample to be processed with the currently selected sample.

Go

Press the **Go** soft button when you want to execute the currently selected DSP function on the currently selected sample. When processing is completed, the K2600 will prompt you to keep the changes. You can audition your changes and decide whether to keep them, then press **Yes** or **No**.

Done

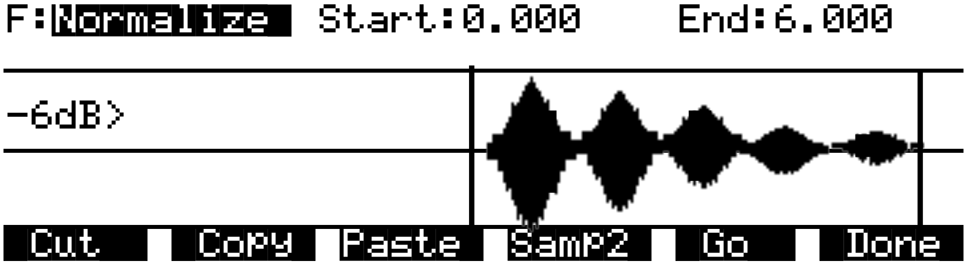
Press the **Done** soft button to return to the previously selected Sample-editor page when you're finished with the DSP functions.

DSP Functions

Once you've entered the Sample Editor, press the **DSP** soft button to gain access to the DSP functions. The DSP function parameter will be highlighted, allowing you to scroll through the list of functions with the Alpha Wheel or **Plus/Minus** buttons.

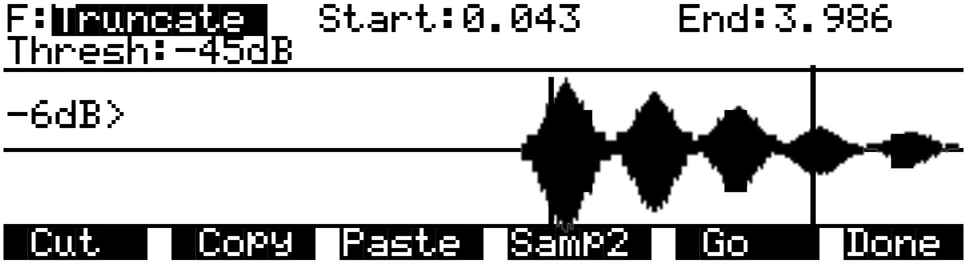
Remember that the DSP functions operate only on RAM samples. You'll notice that if you're editing a ROM sample, there is no **DSP** soft button; instead, there's an extra **Link** soft button for your convenience.

1 Normalize


F:Normalize Start:0.000 End:6.000
-6dB>

With the Normalize function, you can rescale the amplitude of the selected sample segment to optimize its level relative to other samples. The Normalize function will uniformly boost the amplitude of the current sample range as high as possible without clipping, stopping just before the loudest element of the sample would be clipped. You might want to use the Volume Adjust function to boost the current segment manually, but the Normalize function does it automatically, and prevents you from boosting the amplitude too much.

2 Truncate

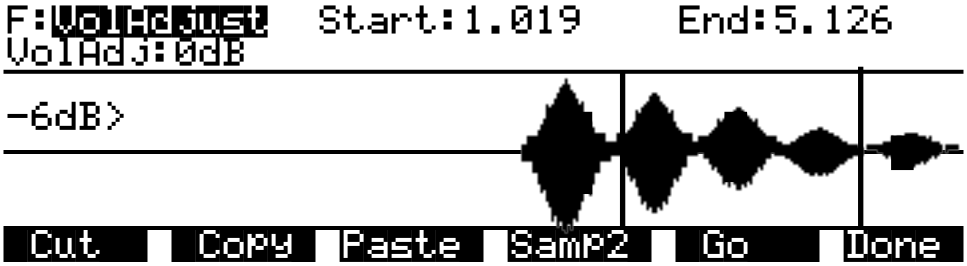

F:Truncate Start:0.043 End:3.986
Thresh:-45dB
-6dB>

The Truncate function will automatically reset the Start (S) and End (E) points of the sample. This can be quicker than trimming the sample manually on the TRIM page.

Use the Start and End parameters to select a specific sample range. Set the Thresh parameter from **-96** to **0dB** to set the noise floor. When you press the **Go** soft button, the K2600 will search inward from the start and end points you set, until it finds the first sample that exceeds the noise floor. Everything outside the range will be left out. You'll want to experiment with different thresholds to find the noise floor that suits each sample you truncate. If the new End point is inside the current loop point, the loop will be disabled.

When you press the **Go** soft button, you'll be prompted "Keep this change?" Unlike the other DSP functions, answering **Yes** does not bring up the Save dialog. It registers the new Start and End positions as the Start (S) and End (E) points (as set on the TRIM or LOOP page). In fact, you can go to either of these pages to readjust the S and E points if you want to include more of the sample. To save the new S and E points, press the **Done** soft button (if necessary) to return to the SampleMode page, then press either the **Save** soft button, or **Exit**.

3 Volume Adjust

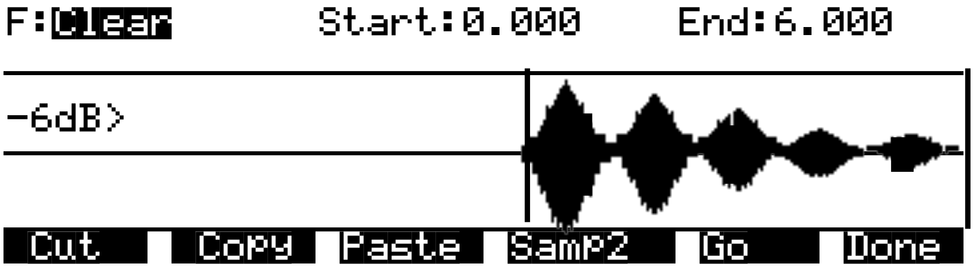


F:VolAdjust Start:1.019 End:5.126
VolAdj:0dB

Use this function for a uniform cut or boost in the amplitude of the selected sample segment. This function will clip samples if you adjust the volume too high. This won't hurt the K2600, and you may find it useful in some applications. In any case, you'll need to choose your start and end points carefully, if you want to avoid abrupt changes in volume.

When you've selected the range to be adjusted, select the VolAdjust parameter with the cursor buttons, and use the Alpha Wheel to adjust the volume of the selected range. You can cut/boost the volume from -96 to 96 dB.

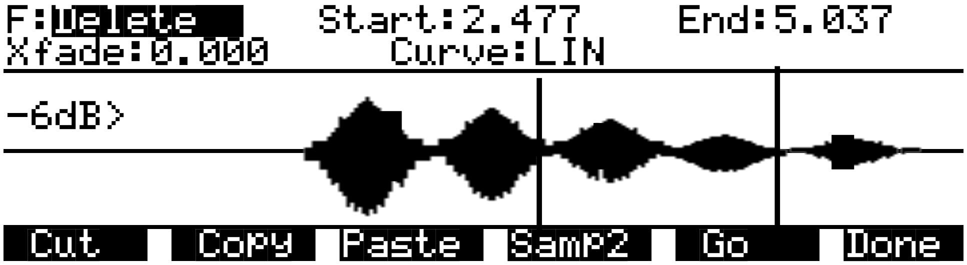
4 Clear



F:Clear Start:0.000 End:6.000

The result of this function is like erasing a section of recording tape. Use it to create sections of silence without changing the overall length of the sample. If you want to completely remove a segment and shorten the sample, you can do it with the Delete function.

5 Delete



F:Delete Start:2.477 End:5.037
Xfade:0.000 Curve:LIN

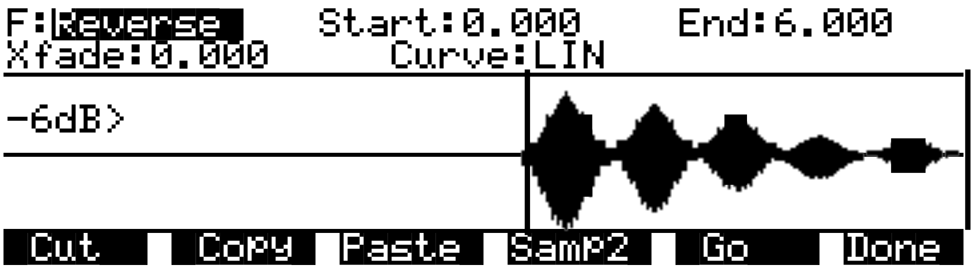
Unlike the Clear function, this will erase the samples within the selected range and shorten the sample, like cutting a section out of a tape and splicing the ends. If you want to silence a segment of the sample without shortening it, use the Clear function.

The Crossfade (Xfade) parameter enables you to smooth the transition from the deleted sample segment to the remaining sample segments, and can create overdub effects. The value of the Xfade parameter defines the amount of time the sample will take to fade to silence at the beginning of the deleted segment, and to ramp back up to the volume of the remaining portion of the sample.

The Curve parameter lets you choose from a variety of crossfade curves that affect the nature of the crossfade. A value of **LIN** gives a straight linear curve that fades uniformly from start to finish. A value of **EXP** sets an exponential curve that starts gradually and steepens toward the finish. A value of **COS** sets a Cosine curve. A value of **EQL** applies an equal crossfade that is the same at both ends of the deleted segment. A value of **MIX** assigns a curve that will start the crossfade before the start point of the deleted segment, and will end after the end point of the deleted segment. This differs from the other curves, which apply crossfade only to the deleted segment. See *Crossfade and Volume Adjust Curves* on page 14-34 for a diagram that helps to explain these curves.

Crossfading a sample will shorten it by half the time you specify. The greatest length of crossfade you can get is the entire length of the shorter crossfaded segment (even though you can enter greater amounts with the Alpha Wheel or alphanumeric buttonpad). A crossfade of this length would crossfade the two samples over the entire length of the shorter segment.

6 Reverse

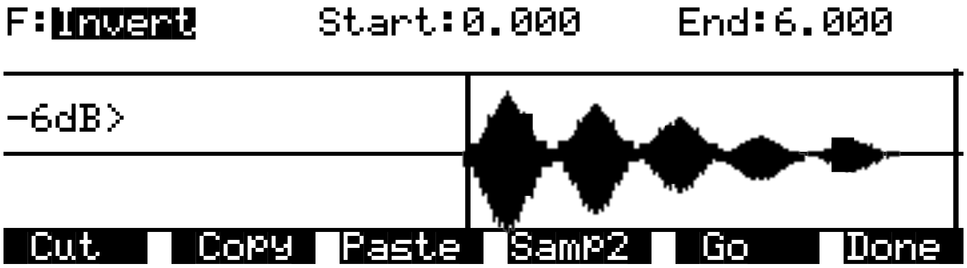


```
F:Reverse     Start:0.000    End:6.000
Xfade:0.000      Curve:LIN
```

With this function you can reverse the order of the individual samples between the start and end positions you set. The Xfade parameter lets you apply a crossfade to the start and end of the reversed segment. The Curve parameter lets you select a crossfade curve. The available values are **LIN**, **EXP**, **COS**, **EQL**, and **MIX**. These curves are described on page 14-34.

Like the crossfade parameter in the Delete function, this crossfade will also shorten the sample. The maximum crossfade length is half the length of the reversed segment.

7 Invert

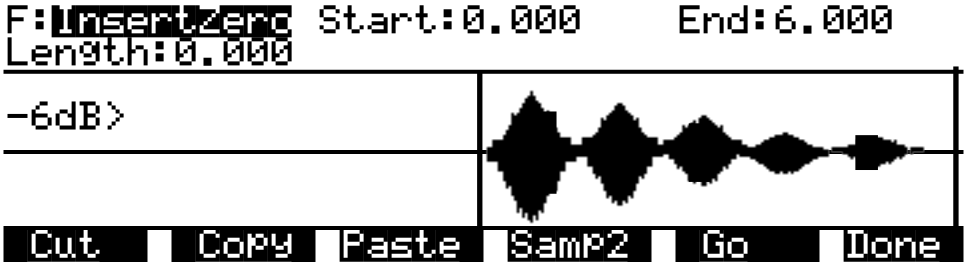


```
F:Invert      Start:0.000    End:6.000
```

Use this function to invert the waveform of the selected sample range. This reverses the phase of the sample. You won't hear any difference unless it is played in reference with the original sample.

The results will vary depending on the type of sample. For an interesting phase effect, make a copy of a sample, then invert the copy, then assign it to the keymap of a second layer in a program that uses the original sample in the keymap of the first layer (or set them as the Left and Right samples in a stereo keymap). Add a delay to one of the layers or keymaps to create the phase effect.

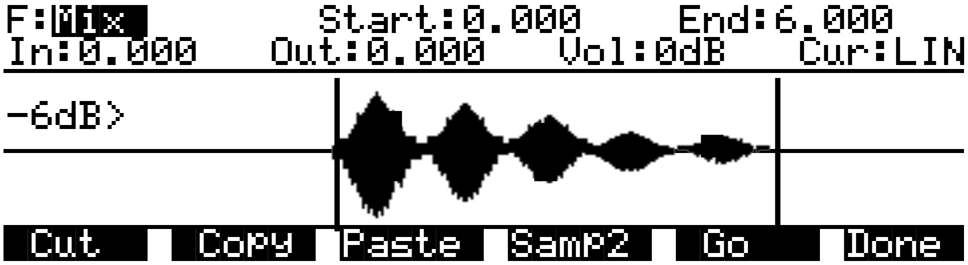
8 Insert Zero



```
F:InsertZero Start:0.000    End:6.000
Length:0.000
```

This function will insert a period of silence of any length into the selected sample range. This function is equivalent to splicing a section of blank tape into an existing segment of recorded tape. It's useful for creating a delay on the start of a sample, or creating a silent spot in the middle or at the end of the sample. Adjust the Start parameter to determine the point at which the period of silence will begin. Adjust the Length parameter to specify how long the silence will be. The End parameter has no effect for this function.

9 Mix



```
F:Mix        Start:0.000    End:6.000
In:0.000   Out:0.000   Vol:0dB   Cur:LIN
```

With this function you select a segment from Sample 2, and merge it with the selected segment from Sample 1, beginning at the point you set with the Start parameter. This is equivalent to mixing two audio signals through a mixing board. If the Sample 2 segment is longer than the segment from the Start (S) to the End(E) of Sample 1, the resulting sample will be longer than the original Sample 1.

Use the In and Out parameters to specify the length of time it takes Sample 2 to reach full amplitude and to fade to silence. The Curve parameter selects the curve of the fades. The available values are **LIN**, **EXP**, **COS**, **EQL**, and **MIX**. These curves are described on page 14-34.

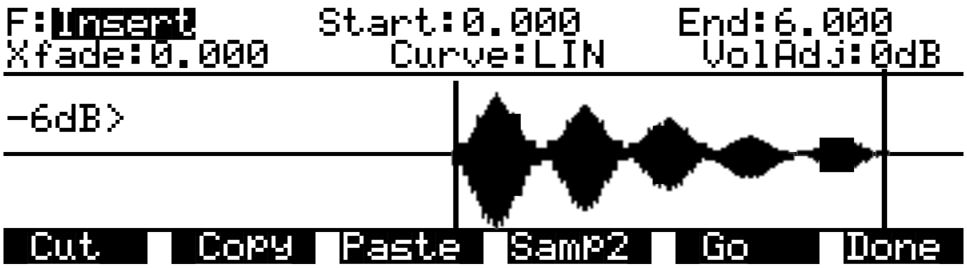
The Volume Adjust parameter will cut or boost the amplitude of Sample 2 from -96 to 96 dB before merging.

If the sample rate of Sample 2 is different from that of Sample 1, the K2600 will alert you that the sample rates differ. If you mix samples with different rates, you'll hear a pitch shift in the mixed sample. If you don't want this pitch shift, use the Resample function to match the sample rates of the two samples before mixing.

Mixing Samples, Step-by-step

First, use the Start parameter of Sample 1 to set the point at which the mix will begin. Then press the **Samp2** soft button, and a page will appear enabling you to view and audition the list of RAM samples. Select a sample with the Sample parameter, then use the cursor buttons to select the Start parameter. Turn the Alpha Wheel to set the beginning of the segment to be mixed in. Repeat this process for the end position of the sample. You can audition the sample as you're setting the range. Once the start and end points are set, press **OK** to return to the DSP page. Then press **Go** to initiate the mix.

10 Insert



```
F:Insert      Start:0.000    End:6.000
Xfade:0.000      Curve:LIN    VolAdj:0dB
-6dB>
```

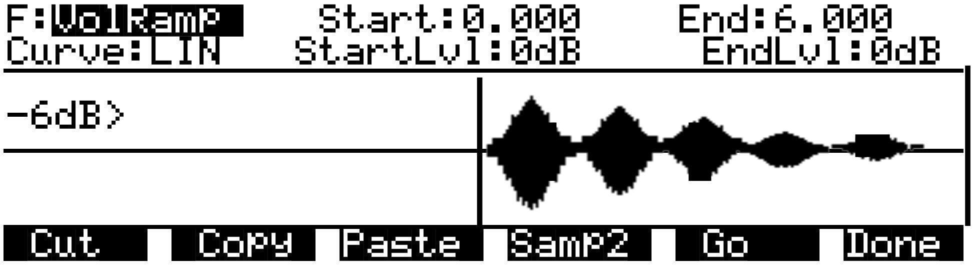
Use this function to insert the selected segment from Sample 2 into Sample 1. This is like splicing a section of tape into an existing tape. This differs from the Mix function, which merges the two samples into one.

Use the Crossfade parameter to control the crossfades at the start and end of the inserted sample. The Curve parameter selects the curve of the crossfade. The available values are **LIN**, **EXP**, **COS**, **EQL**, and **MIX**. These curves are described on page 14-34.

The Volume Adjust parameter will cut or boost the amplitude of Sample 2 from -96 to 96 dB before inserting.

If the sample rate of Sample 2 is different from that of Sample 1, the K2600 will alert you that the sample rates differ. If you insert a sample with a different rate, you'll hear a pitch shift in the inserted sample. If you don't want this pitch shift, use the Resample function to match the sample rates of the two samples before inserting.

11 Volume Ramp

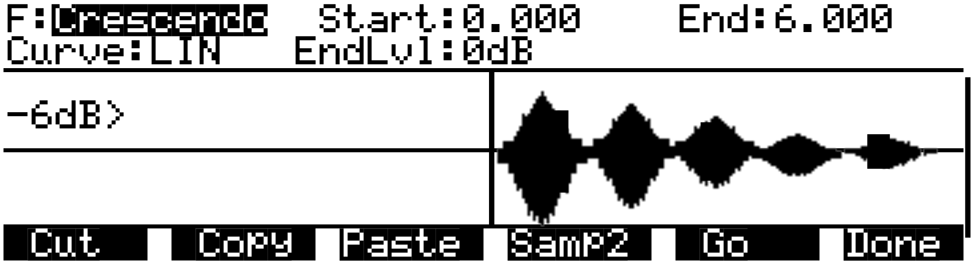


```
F:VolRamp    Start:0.000    End:6.000
Curve:LIN   StartLvl:0dB     EndLvl:0dB
-6dB>
```

This function lets you apply a ramp to the volume of the selected sample range. The Start Level and End Level parameters let you set the amount of cut (negative value) or boost (positive value) at the start and end points of the segment. The Curve parameter determines the shape of the ramp that scales the amplitude of the sample between the start and end amplitudes. The available values are **LIN**, **EXP**, **COS**, **EQL**, and **MIX**. These curves are described on page 14-34.

The VolRamp function affects only those samples within the start and end points. The sample will clip if you apply large amounts of volume ramp. If you want to ramp the volume of a sample segment up or down, then keep the volume at that level, use the Crescendo function.

12 Crescendo/Decrescendo (Crescendo)



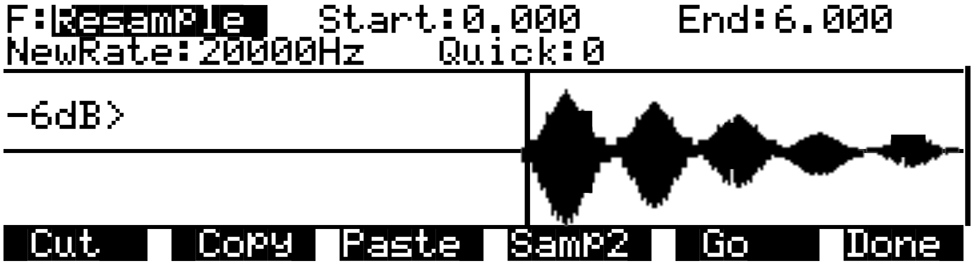
```
F:Crescendo  Start:0.000    End:6.000
Curve:LIN   EndLvl:0dB
-6dB>
```

Similar to Volume Ramp, this function applies a curve that scales the amplitude of the selected sample segment. Unlike Volume Ramp, however, you simply select a start and end point, and a single level. The amount of cut or boost starts at 0dB at the start point of the ramp, and reaches the level you specify when it reaches the end point of the ramp. The samples after the end point are scaled to the amplitude level of the end point.

When you set a negative end level (decrescendo), the sample's volume is cut by the amount you specify. If you set a positive end level (crescendo), the sample segment is first cut by the amount you specify, then is boosted back to its original level. This enables you to add large crescendos without clipping the sample. You may want to adjust the gain of the layer using the sample to match its level with other sounds.

Use the Curve parameter to select the shape of the curve within the selected segment. The available values are **LIN**, **EXP**, **COS**, **EQL**, and **MIX**. These curves are described on page 14-34.

13 Resample



```
F:Resample   Start:0.000    End:6.000
NewRate:20000Hz   Quick:0
```

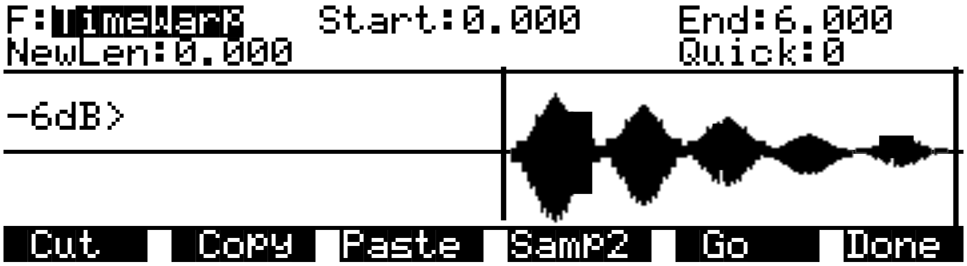
Use this function to change the sample rate of the samples in the selected segment. This is convenient for converting samples to new rates matching those of other samples to be mixed with or inserted into. It's also useful for saving memory and altering the timbre of a sample.

If you include the entire sample in the segment, the new rate will be applied to the entire sample, and will be saved with the sample. If you select a shorter segment, only that segment will be modified, and it will sound pitch shifted relative to the remainder of the sample. To resample so that you'll hear a higher pitch for the selected segment, select a lower sample rate. For a lower pitch at the selected segment, choose a higher sample rate. This is because the K2600 applies the same playback rate to the entire sample, and doesn't compensate for the differing sample rates of the sample segments.

If the loop points of a looped sample are included in the segment to be converted, the K2600 will ask you if you want to adjust the rate slightly to optimize the loop. Press the **No** soft button if you don't want the rate adjusted.

You can use the Quick parameter to select from two resampling routines. Use **Quick 0** to get an idea of the sound, then use **Quick 1** for your final take.

14 Time Warp



```
F:TimeWarp   Start:0.000    End:6.000
NewLen:0.000                Quick:0
```

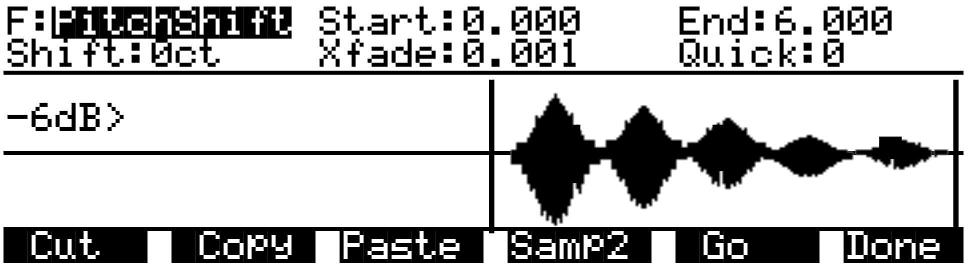
With this function you can change the length of the selected sample segment without affecting the pitch. This function applies sophisticated routines that lengthen or shorten the selected sample segment to play it back over a different time period, modifying the playback rate so the pitch remains unchanged.

The Start and End parameters define the segment to be processed. Use the New Length (NewLen) parameter to specify how long you want the resulting sample to be. While the function is in progress, the display will indicate the percentage of individual sample segments that have been processed.

The Quick parameter lets you select one of three warping routines. Use **Quick 0** to audition your new sample, then use **Quick 1** or **2** for your final take. **Quick 2** takes more time to process, but gives you a better final result.

This function is extremely useful for fine adjustments in the length of a sample. You can also apply greater amounts of warp for a wide range of effects. Experimentation will give you an idea of the amount of alteration you want to apply.

15 Pitch Shift



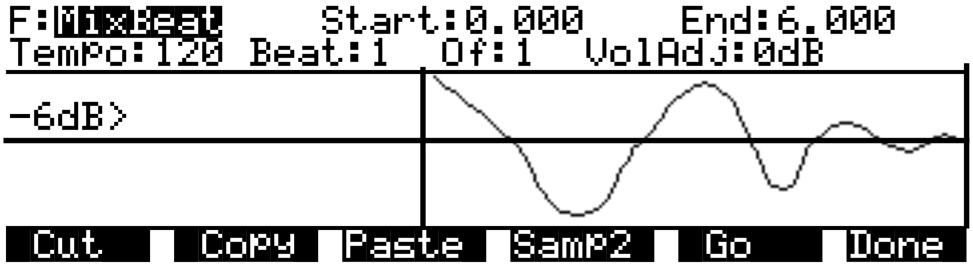
```
F:PitchShift  Start:0.000    End:6.000
Shift:0ct     Xfade:0.001    Quick:0
-6dB>
```

The PitchShift function is the counterpart of the TimeWarp function; it shifts the pitch of the selected sample segment without changing the playback time—very useful for tuning samples when the playback time is crucial. Use the Start and End parameters to define the segment to be shifted. The Shift parameter determines the amount of pitch shifting, up to ± 30000 cents.

Like the crossfade parameter in the Delete function, this crossfade will also shorten the sample. The maximum crossfade length is half the length of the reversed segment.

The Quick parameter lets you select one of three shift routines. Use **Quick 0** to audition your sample, then use **Quick 1** or **2** for the final take. **Quick 2** takes longer to process, but gives you better results.

16 Mix Beat



```
F:MixBeat     Start:0.000    End:6.000
Tempo:120 Beat:1  Of:1  VolAdj:0dB
-6dB>
```

With this function you can mix the selected range of Sample 2 into Sample 1 at regular intervals. One natural application of Mix Beat is to mix various percussion samples (Sample2) on different beats of Sample 1. You could then loop Sample 1, creating a drum-loop sample out of individual percussion samples. You don't have to use percussion samples, however. You can create many interesting and unusual effects by mixing different types of samples, especially if you set the Tempo parameter to a high value.

The Tempo parameter sets the interval between repetitions in beats per minute. The Of parameter establishes the number of beats per measure, and the Beat parameter sets which beat gets mixed. For example, if you set the Tempo parameter to 120, you'll have two beats per second. If you set the Beat parameter to **1**, and the Of parameter to **1**, Sample 2 will be mixed in twice per second, for the duration of the selected range of Sample 1. The first mix of Sample 2 occurs at the Start (S) point of Sample 1. If you change the Of parameter to **4**, Sample 2 will be mixed into Sample 1 on the downbeat of every measure of 4.

The length of the ranges you set for Sample 1 and Sample 2 affects the results of the mix. In the above example, if Sample 1 is two seconds in length, and the mix of Sample 2 is on Beat 1 of 1, you'll mix four segments of Sample 2, at half-second intervals. On beat 1 of 4, you'd hear just one mix of Sample 2, right at the top.

To set up a MixBeat, first select the desired segment of Sample 1 using the Start and End parameters. Then press the **Samp2** soft button to select the sample to be mixed in, and the selected range of that sample. Note that when you go to the Samp2 page from the Mix Beat page (and from the Replicate and Mix Echo pages), an extra parameter—Increment (Incr)—appears. This parameter isn't available for the other DSP functions.

You can use the Incr parameter to shift the starting position of Sample 2. If you set a positive value, the start and end points of Sample 2 will shift later each time it's mixed. A negative value will shift the start and end points of Sample 2 earlier with each mix.

To optimize processing time, keep the range of Sample 2 shorter than the interval between mixed-in segments. In the example above, if the mix were on beat 1 of 1, you'd want to keep Sample 2 at a range of a half second or less. If it were on beat 1 of 4, the range of Sample 2 could be as much as two seconds.

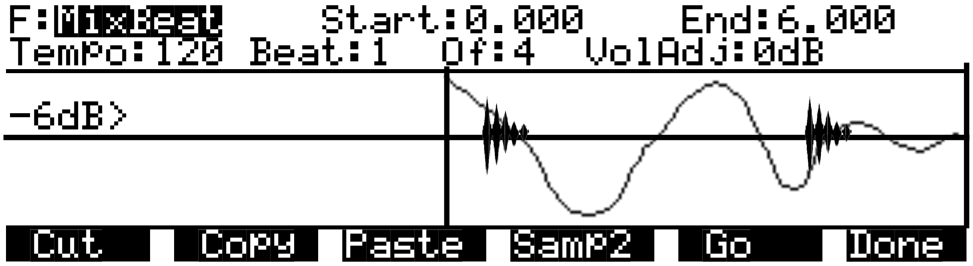
Next, press the **OK** soft button to return to the Mix Beat page, then use the Tempo parameter to select the rate at which Sample 2 will be mixed in. You can choose a tempo from 1 to 9999 beats per minute. Try setting the tempo at 9000 or more, and mix a very small range of Sample 2 into Sample 1 to create a wide variety of periodic waveforms.

Now set the values of the Beat and Of parameters to determine how the mixed sample will repeat. The Beat parameter determines the beat (s) on which Sample 2 will be mixed—from 1 to 9999. The Of parameter determines the measure length—also from 1 to 9999.

Finally, use the VolAdjust parameter to set the volume of the mixed sample segment—from -96 to 96 dB.

If the Of parameter is set to a value of **0**, the Sample 2 segment will be mixed in on every beat, regardless of the setting for the Beat parameter. If the Beat parameter is set to a negative value, the segment of Sample 2 that's mixed in will be moved forward in time by the length of one beat each time it's mixed in; that is, you'll hear a later portion of the sample. Another way to accomplish this is to use the Increment (Incr) parameter on the Samp2 page. Set it to a positive value to use a later portion of Sample 2 with each repetition, or a negative value to use an earlier portion. The Beat parameter must be set to a value of **0** or higher for this to work.

For example, suppose you've chosen a six-second sample as Sample 1, and you use the entire sample as the selected segment. You also select a half-second segment of Sample 2 to be mixed in. If you choose a Tempo value of 120 beats per minute (2 beats per second), there will be 12 beats within your six-second Sample 1 segment. If you set the Beat parameter to **1** and the Of parameter to **4**, then Sample 2 will play on the first, fifth, and ninth beat of your six-second Sample 1 segment. The result will look something like the diagram below.

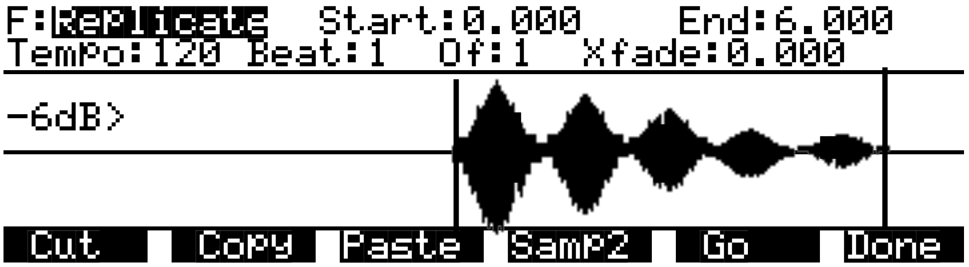


```
F:MixBeat     Start:0.000    End:6.000
Tempo:120 Beat:1  Of:4  VolAdj:0dB
-6dB>
```

If you change Beat to **3**, Sample 2 will play on beats 3, 7, and 11. If you set Beat to **1** and Of to **6**, Sample 2 will play on beats 1 and 7. If you set Beat to **-1**, Sample 2 will still play on beats 1 and 7, but the sound of the sample will change each time, as the start point of the Sample 2 segment moves forward (later) by one beat each time it is mixed in.

Mix Beat (as well as Replicate and Mix Echo) may seem a bit complicated at first, but if you experiment with different settings—especially for Beat and Of—you'll quickly begin to get a feel for what these functions do.

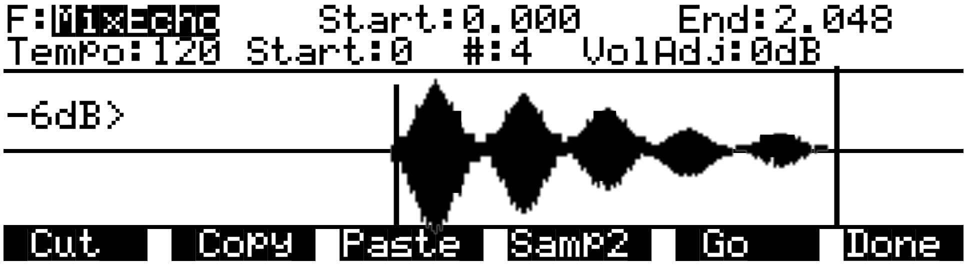
17 Replicate



```
F:Replicate   Start:0.000    End:6.000
Tempo:120 Beat:1  Of:1  Xfade:0.000
-6dB>
```

With the exception of two differences, the replicate function is similar to the Mix Beat function. The primary difference is that the replicate function uses the selected Sample 2 segment to overwrite (replace) the selected Sample 1 segment, instead of merging the two segments like the Mix Beat function. This makes it run faster. The other difference is that there is a crossfade parameter instead of a Volume Adjust parameter. The crossfade parameter lets you smooth the transition points from Sample 1 to Sample 2.

18 Mix Echo



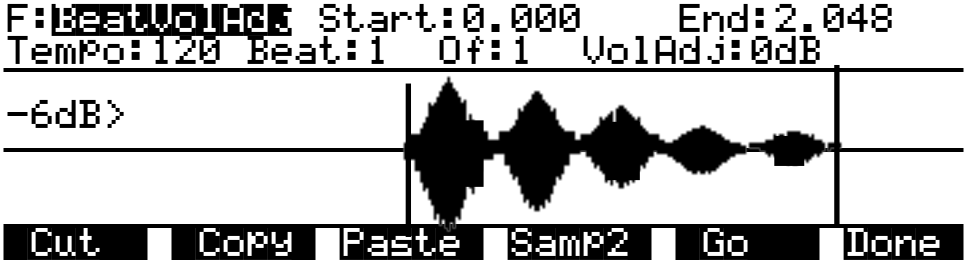
F:MixEcho Start:0.000 End:2.048
Tempo:120 Start:0 #:4 VolAdj:0dB
-6dB>

This function operates much like Mix Beat, but instead of the Beat and Of parameters, you have Start and # parameters. The Start parameter sets the beat at which the selected Sample 2 segment begins being mixed with the selected Sample 1 segment. Sample 2 is repeated on every beat after this starting point. The # parameter determines how many times the Sample 2 segment is repeated.

The Volume Adjust parameter will affect the volume of each repetition of Sample 2. It sets the relative level of the last beat, and boosts or cuts each repetition by a proportionate amount from the starting level. This is a linear adjustment, measured in decibels, evenly spaced over the total number of mixed-in beats.

If you set the # parameter to a positive value, the adjustment in volume begins on the second mixed-in beat (the first beat plays at the initial level). If you set a negative value, the first mix of Sample 2 will be cut or boosted by the Vol adjust amount.

19 Beat Volume Adjust

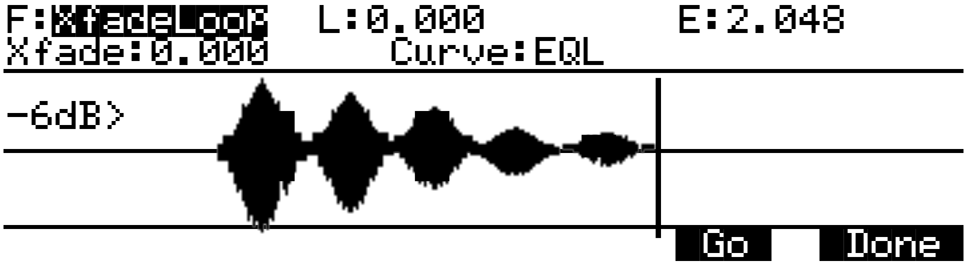


F:BeatVolAdj Start:0.000 End:2.048
Tempo:120 Beat:1 Of:1 VolAdj:0dB
-6dB>

This function works much like Mix Beat. The difference is that you're not mixing in any samples, you're just volume-adjusting Sample 1 at regular intervals. Each beat is adjusted in volume by the amount specified for the VolAdj parameter. This is useful for mixing part of a sample within itself.

Set the Start and End positions of Sample 2 to define the length of time over which the volume will be adjusted. For example, if you had a six-second sample, and you set the Tempo to 120, the Beat parameter to **1**, and the Of parameter to **1**, you'd have twelve beats defined. If you then set the range of Sample 2 at a half-second, and set the VolAdj parameter to -3 dB, then you'll insert 12 volume adjustments into Sample 1. Each adjustment lasts a half second, and brings the level down 3 dB.

20 Crossfade Loop (XfadeLoop)

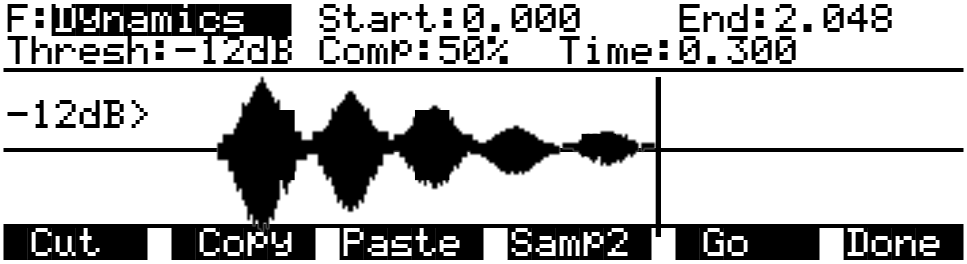


```
F:XfadeLoop  L:0.000       E:2.048
Xfade:0.000     Curve:EQL
```

The Crossfade Loop function lets you create smoother loops by crossfading the beginning segment of the loop with a segment of equal length at the end of the loop. These segments can be defined by the Loop and End parameters as set on the TRIM or LOOP page for the current sample, or with the Loop and End points on the XfadeLoop page. Changing the Loop and End parameters on the XfadeLoop page will change them on the TRIM and LOOP pages, and vice versa. Using this function is equivalent to setting the loop on the LOOP page, but with the added feature of a crossfade at the loop transition point.

The Xfade parameter determines the length of the crossfade, while the Curve parameter sets the shape of the crossfade curve. The available values are **LIN**, **EXP**, **COS**, **EQL**, and **MIX**. These curves are described on page 14-34.

21 Dynamics



```
F:Dynamics   Start:0.000    End:2.048
Thresh:-12dB Comp:50%  Time:0.300
```

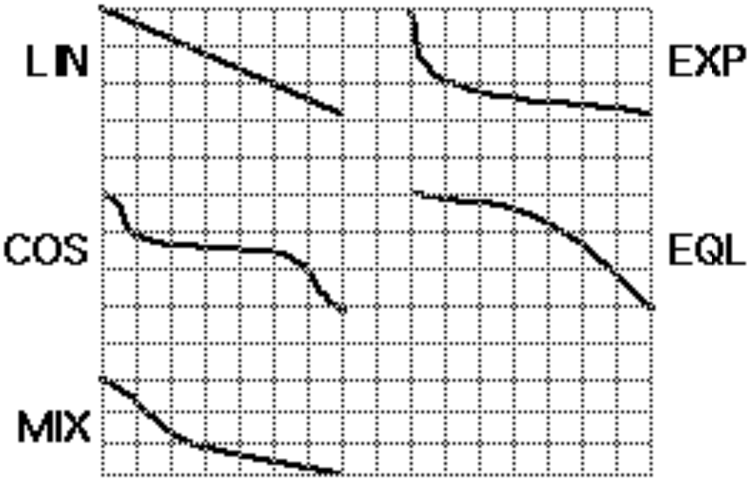
The Dynamics function is a general-purpose compressor with a few features that make it suitable for creating smooth sample loops. First, while most compressors set a threshold relative to 0 dB, the Dynamics function sets its threshold relative to the peak of the signal you're compressing. Secondly, after compressing the signal, it adjust the volume of the compressed segments (there may be more than one) so that they match the uncompressed segments as closely as possible.

Set the Start and End to select a segment of the signal for compression. The Thresh parameter sets a threshold in dB, relative to the peak amplitude of the selected segment. Any portion of the segment between Start and End that exceeds that threshold gets compressed. The Comp parameter determines how much compression gets applied to sample points that exceed the threshold, expressed as a percentage of the amount by which the sample points exceed the threshold. The Time parameter controls the length of time that the compressor ramps back down to zero compression when it reaches the end of a compressed segment. This helps assure a smooth transition between compressed and uncompressed segment.

Here’s an example. Suppose you set Thresh, Comp, and Time as shown in the preceding diagram. Beginning from the Start point, when the signal first reaches the threshold, the Dynamics function begins compressing the signal, attenuating every point above the threshold by 50%. Compression continues until the signal goes back below the threshold. At that point, the compression starts diminishing, and ramps back to zero in 0.3 seconds. If the signal goes above the threshold again within the selected segment, this process repeats.

Crossfade and Volume Adjust Curves

There are five curves that can be applied to a number of DSP functions: LIN, EXP, COS, EQL, and MIX. The LIN curve is a straight linear curve, which will create an even cut or boost. The EXP curve is exponential, that is, gradual at one end and steep at the other. The COS curve is a segment of a cosine curve, which is relatively steep at both ends and flatter in the middle. The EQL curve steepens at an even rate, approximating an equal-power fade curve. The MIX curve is a gradual curve that approximates manually dropping or raising the faders on a mix board. The diagram below shows each of the curves as it would be applied to a cut in amplitude.



The Keymap Editor

The Keymap Editor lets you customize the K2600’s factory preset keymaps and save them to RAM. You can also build your own keymaps from scratch.

Keymaps are an integral part of every layer of a program. Each keymap contains a set of parameters determining which sample(s) the K2600 will play when you trigger a note. Each layer has at least one keymap, but it can have two keymaps when you’re working with stereo samples. Each of these stereo keymaps uses two of the 48 available voices.

Each keymap consists of a set of key (note) ranges—C 4 to G 4, for example. The entire span of each keymap is from C 0 to G 10. Each range has a sample root assigned within the range. Each sample root is a distinct ROM or RAM sample. Within each key range, the sample root is transposed up and down to play on each of the range’s notes. You can view each range by changing the value of the Key Range parameter on the Keymap-editor page. You can mix samples of different timbres within a single keymap, and even tune individual keys to any pitch by defining key ranges to single notes and assigning samples to each of those notes.

When you trigger a note, the K2600 identifies the key range where the Note On event occurred. It also checks the attack velocity value of the note. It then addresses its memory, and retrieves the sample root that's assigned to that key range and attack velocity value. If the note that's triggered is not the note where the sample root is assigned, the sample is transposed to play at the correct pitch. The K2600 then generates the digital signal that represents the sound of the note. At this point the keymap's job is done, and the signal proceeds through the layer's algorithm and on to the audio outputs.

You can assign as many key ranges to a keymap as you like, even creating a separate range for each note. This would allow you to tune each key independently, to create microtonal tunings. For keymaps that use a single timbre, like the Grand Piano, there's a key range for each sample root stored in memory. For acoustic instrumental sounds, the more key ranges you have for a keymap, the more realistic the sound will be, since there will be less pitch shifting of the sample root within the key range.

Of course, you can assign sample roots with different timbres within the same keymap. Many of the drum kit keymaps in ROM, for example, have about 20 key ranges, with several different timbres assigned as the sample roots. You can also create a keymap with a single key range that spans from C 0 to G 10, if you want to stretch a single sample root from C 0 to G 10. Keep in mind, however, that samples can be transposed upward only to a limit of 96 KHz for the playback rate. For 48K samples, that's an octave of upward transposition—or two octaves if you set the SmpSkp parameter (on the KEYMAP page in the Program Editor) to **On** or **Auto**. Samples can be transposed downward without limit.

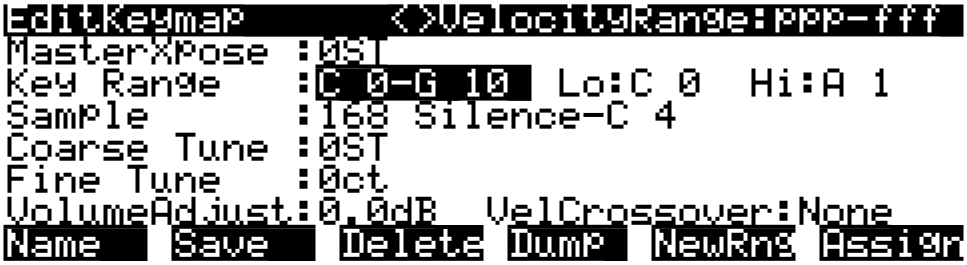
Think of a keymap as if it were a single piece of string, divided into different sections that adjoin one another. Sections cannot overlap. If you have one range that goes from C4 to F4 and another that goes from F#4 to C5, then if you change the first range to be C4 to G4, the second one will change to be G#4 to C5.

Also, you can't have "nothing" assigned to a key range. Even if it is Silence (#168), there will always be a sample assigned to every range in the keymap. This is something to watch out for when creating drum programs. For example, let's say you are creating a program with 20 layers. Each layer has its own keymap, which has just one sample assigned to part of the keyboard with the rest of the key range assigned to Silence. Make sure that you limit the note range of each layer using the LoKey and HiKey parameters on the LAYER page in the Program Editor. If each layer covers the entire range, then each note you played would trigger 20 voices (one for each layer). You would only hear one drum per note because all the other layers are triggering "Silence." Because of the voice-stealing algorithms in the K2600, the voices would almost immediately become available again, since they have no amplitude. But for one brief instant, the voice would be triggered, which could cause other voices to be cut off.

You can also create multi-velocity keymaps—that is, keymaps that will play different timbres depending on the attack velocities of your Note On events. Program **12 DynEPiano^EPnoPF**, for example, uses a keymap with two velocity ranges. Each key range in a multi-velocity keymap contains two or more distinct sample roots that the K2600 chooses between, according to the attack velocity of the note. When in the Keymap Editor, you can select the different velocity ranges within a keymap using the **Chan/Bank** buttons. To create your own multi-velocity keymaps, Select any program (like 199 Default Program), and press **Edit** to enter the Program Editor. Then select the KEYMAP page, and select one of the multi-velocity keymaps as the value for the Keymap parameter (the multi-velocity keymaps are IDs **31–38**). The description of the Velocity Crossover parameter on page 14-39 has more information.

The Keymap Editor is nested within the Program Editor. The first step in using the Keymap Editor is to select the keymap you want to edit. This is done on the KEYMAP page in the Program Editor, using the Keymap parameter. Once you've done this, just press the **Edit** button, and you'll enter the Keymap Editor. If you want to edit a different keymap, return to the KEYMAP page in the Program Editor and select the desired keymap. If you want to build a keymap from scratch, start with the keymap **168 Silence** (see *Building a Keymap* on page 14-39).

This keymap template contains one key range from C 0 to G 10, and is a convenient starting point for adding key ranges and assigning sample roots. The Keymap-editor page looks like this:



```
EditKeymap       <>VelocityRange:ppp-fff
MasterXpose :0ST
Key Range   :C 0-G 10  Lo:C 0  Hi:A 1
Sample      :168 Silence-C 4
Coarse Tune :0ST
Fine Tune   :0ct
VolumeAdjust:0.0dB  VelCrossover:None
Name    Save    Delete  Dump    NewRng  Assign
```

Parameter	Range of Values
Master Transpose	-128 ST to127 ST
Key Range	Variable
Low Key	C 0 to G 10
High Key	C 0 to G 10
Sample	Sample Root list
Coarse Tune	–120 to 60 semitones
Fine Tune	-49 to 50 cents
Volume Adjust	± 48 dB
Velocity Crossover	None, ppp to fff

The top line of this page tells you which velocity range you're currently looking at. If the current keymap is a multi-velocity keymap, the **Chan/Bank** buttons let you select between the velocity ranges. The velocity range is set with the Velocity Crossover parameter(s), described on page 14-39.

The Soft Buttons in the Keymap Editor

The first four soft buttons execute the basic library functions, enabling you to name, save, or delete the current keymap, or dump it via a MIDI SysEx message.

New Range (NewRng)

The **NewRng** button lets you define a range to edit, whether it's to assign a different sample, or to adjust the pitch or volume. Just press **NewRng**, then trigger the note you want as the low note, then the high note. The K2600 will prompt you for each note. When you trigger the high note, you'll return to the Keymap-editor page, and the edit range you defined will be selected. The next change you make will affect only that edit range.

There's more than one way to use this function. If you set an edit range that's completely within an existing key range, you can modify the edit range without affecting the rest of the key range or the adjacent key ranges. If you set an edit range that overlaps part or all of another key range, the sample assigned to the lower key range will be applied to the entire edit range. This is an easy way to define a new key range that replaces one or more existing key ranges.

Assign

The **Assign** soft button lets you select a sample, then specify the key range to which it's assigned. This enables you to insert a new key range within the current keymap. When you press the **Assign** soft button, a dialog appears that prompts you to select a keymap from the Keymap list. Scroll through the list, then press the **OK** soft button. You'll then be prompted to define the new key range by triggering the notes you want to be the lowest and highest notes of the range. (Press the **Cancel** soft button if you change your mind.) When you trigger the low and high notes, the new key range is inserted. If the new key range partially overlaps an adjacent key range, the existing key range will be adjusted to accommodate the new range. If the new key range completely overlaps an existing key range, the original key range will be replaced.

Special Double Button Presses in the Keymap Editor

Suppose you have a sample whose root key is C 4, and you want to assign it to A 0, because you don't expect to play it often. If you want it to play back without transposition, you'll have to adjust the Coarse Tune parameter. Calculating the right value for Coarse Tune can get tedious if you're assigning a large number of samples. Fortunately, there's a short cut.

1. Assign a sample root to a key range, either using the Lo, Hi, and Sample parameters or using the **Assign** soft button.

2. Highlight the value of the Coarse Tune parameter.

3. Press the **Plus/Minus** buttons at the same time. The value of Coarse Tune changes automatically. If the sample is assigned to one note, the K2600 sets Coarse Tune so that the note plays the sample without transposition. If the sample is assigned to a range of notes, the K2600 sets Coarse Tune so that the middle note of the range plays the sample without transposition.

Keymap Editor Parameters

Master Transpose

This parameter does not really pertain to the keymap itself. Instead it is identical to the Transpose parameter found on the MIDI-mode TRANSMIT page. If you change the value here, the same value will be reflected on the MIDI-mode TRANSMIT page, and vice versa. It transposes the entire instrument globally. The reason it is placed on this page is that it will allow you to assign samples across the entire keyboard easily, when you are using a keyboard that has fewer than 88 notes.

As with the Transpose parameter on the MIDI-mode TRANSMIT page, if you have the rack-model K2600, this parameter will not transpose without using the Local Keyboard Channel parameter (refer to page 10-7 for more on the Local Keyboard Channel).

Key Range

This parameter shows you which key range you're currently viewing or editing. Changing the value of this parameter selects a different key range, and shows you the sample assignment of that key range, as well as other parameters related to that key range. Use this parameter to move from one key range to another within the keymap, and assign different samples without adding new key ranges, as well as editing the transposition, tuning, volume and velocity crossover (if any) of the sample root assigned to the key range.

Low Key (Lo), High Key (Hi)

With these parameters you can use any of the data entry methods to change the low and high notes of the current range. These parameters let you extend or shorten the width of a key range. You can extend a key range to the full capacity of the K2600 (C 0 to G 10).

The setting for the low key cannot be higher than the setting for the high key. Similarly, the setting for the high key cannot be lower than the setting for the low key.

Sample

This is where you assign a sample root to the current key range. Depending on the nature of the sample root—an individual sample or a block of sample roots—the sample's name looks a bit different in the display. Each sample's name consists of three parts: a numeral, a name, and a note number—for example, **1 Grand Piano G#1**.

The numeral is the sample block ID. If the sample object is an individual sample, the sample block ID is the same as the sample's object ID. If the sample object is a group of sample roots, the object ID of the first root in the group determines the sample block ID. The remaining roots in the block have the same ID, and differ only in their note numbers.

Next comes the name of the sample, which typically describes the sample's timbre. The final part of the sample's name refers to the pitch at which it was originally sampled. For many timbres, multiple samples are made at various pitches. As you scroll through the Sample list, you'll see only the pitch of the sample change until you reach the next sample block.

Highlight the Sample parameter, hold the **Enter** button, then play a note to see the complete name of the sample played by that note. For example, when you play middle C on a piano program you'll see a name such as **1 Grand Piano C4**. Move down the keyboard an octave and the sample will be **1 Grand Piano G$^{\#}$2**.

Coarse Tune

Coarse Tune allows you to transpose a sample for a given range. This is extremely useful when you have set the Root key of the sample for one note but want to assign the sample to a different part of the keyboard and still be able to play it without transposition. For example, if you originally set the Root key at C4 but want the sample assigned to C3, you would set Coarse Tune to 12ST, transposing it up one octave. Now the original pitch will play at C3, one octave down. If you examine the drum and percussion kit keymaps in ROM, you will see that we have done this. Most of our ROM drum samples have the Root key set at C4.

There's a short cut for adjusting the Coarse Tune automatically so that the sample plays with minimal transposition in the assigned key range. See *Special Double Button Presses in the Keymap Editor* on page 14-37.

Fine Tune

This gives you further pitch control. Once the sample's pitch is close to the desired level, use the Fine tune to sharpen or flatten it as much as a half-semitone.

Volume Adjust

Here you can adjust the volume of the notes in the current key range. This enables you to make each key range play at the same volume even if the samples in the various ranges were recorded at different volumes.

Velocity Crossover (VelCrossover)

This parameter applies only when the keymap assigned to the currently selected program is a multi-velocity keymap. The name of the keymap usually indicates whether it's of the multi-velocity variety (**29 Bass^Slap Bass**, for example). Multi-velocity keymaps have a predetermined number of velocity levels, each of which can be assigned a different sample.

The K2600 supports keymaps with up to eight velocity levels. You can't add velocity levels to existing keymaps; if you want to create your own multi-velocity keymaps, select an existing multi-velocity keymap in the Program Editor before entering the Keymap Editor. Then you can select the different velocity levels with the **Chan/Bank** buttons, and assign samples to the different levels. The currently selected velocity range is shown in the top line of the display.

When the current keymap is a single-velocity keymap, the VelCrossover parameter does not appear on the Keymap-editor page. When the current keymap is dual-velocity, the value for the Crossover parameter will be one of the eight dynamic markings from ppp to fff. The K2600 translates each of your Note Ons into one of these dynamic values, using the settings for the VelTouch or VelocMap parameters. When this translated value exceeds the setting for the VelCrossover parameter, the K2600 plays the sample assigned to the upper velocity range.

When the currently selected keymap has three velocity ranges, the VelCrossover parameter becomes two parameters: LowCrossover and HiCrossover. The K2600 plays the sample assigned to one of these ranges depending on the translated value of each note's attack velocity.

When you want to use a keymap with more than three velocity levels, pick one with velocity crossover levels that you like, since you can't change the crossover velocity in keymaps with more than three levels.

Building a Keymap

If you used the Keymap Editor to enter the SampleMode page, then just press **Exit** from the SampleMode page and you are ready to begin creating a keymap. If you entered the SampleMode page from Master mode, do the following. Start in Program mode, and select Program 199, the Default program. Press the **Edit** button, and you'll enter the Program Editor. Press the **KEYMAP** soft button, and the KEYMAP page will appear. The Keymap parameter will be automatically selected. Press **1**, **6**, **8**, **Enter** on the alphanumeric pad to assign the keymap **Silence**. This isn't absolutely necessary, but it makes it easier to recognize the key ranges that have samples assigned to them when you start assigning samples. You can actually choose any program you want to start with, but by choosing these, you are starting with a "blank slate."

With the Keymap parameter still selected, press the **Edit** button, and you'll enter the Keymap Editor. The Key Range parameter will be automatically selected, and you see its values: C 0 to G 10 (the entire MIDI keyboard range). The Sample parameter will have a value of **168 Silence C 4**.

Now you're ready to start assigning samples to key ranges within the keymap. We'll assume that you've loaded samples with roots at C 1, C 2, C 3, etc. and that you plan to assign a root to each octave. To begin, press the **Assign** soft button. The display will prompt you to select a sample. Use the Alpha Wheel to scroll to one of your samples, or type its ID on the alphanumeric pad and press **Enter**. When you've found the sample you want to use, press the **OK** soft button. The display will say "Strike low key…" Trigger A 0 (MIDI note number 21, the lowest A on a standard 88-note keyboard). The display will change to say "Strike High Key…"Now trigger F 1 (MIDI note number 29). The display will return to the Keymap-editor page. The Key Range parameter will show A 0–F 1, and the Sample parameter will show the sample you selected when you started the range assignment.

One more time…Press the **Assign** soft button. Select another sample root at the prompt, and press the **OK** soft button. Now trigger F# 1 for the Low Key prompt, and F 2 for the High Key prompt. At this point you've defined two key ranges, the first from A 0 to F 1, and the second from F# 1 to F 2. You can repeat the process as many times as you want, creating a new key range each time.

Once you have your samples assigned, you may need to transpose them so that they play back at the correct pitch within the range you have chosen. To do this, highlight the Key Range parameter, scroll to the range you need, then highlight the Coarse Tune parameter. Adjust Coarse Tune to bring the sample to the proper pitch within that key range. Then scroll back up to the Key Range parameter, select the next range, and continue as needed.

Here's a fairly important point that may or may not affect your keymap construction. Suppose you want to build a keymap that uses the same sample in several adjacent key ranges, and you plan to add a bit of detuning to the samples in each range. You might think that you could build the keymap first, then go into the Sample Editor and tweak the samples when the keymap is finished. Yes, but…

Suppose you used the technique we described above to assign a vocal sample whose root was C 4 to a key range from A 3 to E 4. Then you assigned the same sample to a key range from F 4 to B 4. You might be surprised to find that when you finished the F 4–B 4 key range and the Keymap-editor page reappeared, the current key range would not be F 4 to B 4, but A 3 to B 4! This is because the K2600 automatically merges adjacent key ranges that are identical (this is done to save memory). Therefore, some parameter must be different in each adjacent key range you create if you want to build keymaps using the technique we just described. So if you want to use the same samples in adjacent key ranges with, for example, minor pitch or volume modification, you should make those changes to the current sample on the Keymap-editor page *before* assigning the next range.

Using the Analog Inputs to Trigger Samples

The analog sampling inputs double as one- or two-channel trigger inputs. This allows audio signals from external sources (such as microphones and tape recorders) to trigger internal samples. The following steps explain how to use the trigger inputs.

1. Connect cables from the outputs of your external audio source to the analog sampling inputs on the K2600. For details on which cables to use with the analog sampling inputs, read *Setting Up For Sampling* on page 14-1.

2. Enter Program mode, and select the program containing the sample(s) you want to trigger.

3. Set the Song-mode Click parameters to match the program you chose:

   Enter Song mode, select the MISC page, and set the Click Channel and Click Program parameters to the MIDI channel and program number of the program you just chose. Change the value of the Click Key parameter to define the key number played by the sequencer's metronome. This indirectly determines what samples are triggered by the inputs, since the inputs always trigger the samples that are assigned to the first two keys above the Click Key.

   The default Click parameters are Click Channel **16**, Click Program **198 Click**, and Click Key **C4**. (Therefore the default keys triggered by the analog inputs are C$^{\#}$ 4 and D 4.)

4. Enter Sample mode, set Src to **External**, and set Mode to **Trigger**.

5. Set the trigger threshold:

Highlight the Thresh parameter on the SampleMode page, and adjust the input sensitivity so that your input signal causes smooth triggering. Threshold values approaching -90db let you trigger the sample using a lower input level, while values approaching -6db require a greater input level.

Now analog input signals will trigger samples in the click program. The sampler's analog inputs are stereo, so you can trigger samples on two different keys. The left channel will trigger samples that are one key above the Click Key; the right channel will trigger samples that are two keys above the Click Key.

One great use for the trigger inputs is to replace the kick and snare from a multi-track recording with samples in your K2600. Assuming the kick and snare are recorded on separate tracks, and the Click Key is C4, the recorded drums will trigger samples assigned to C#4 and D4 in the click program.

Keep in mind that the Click Key number is separate from the trigger key numbers. This is because you may want to use the click program as the metronome sound, and trigger samples other than the Click. To do this, go to Program mode, select the click program, and press **Edit**. Then press the **KEYMAP** button and press **Edit** again. Now assign your samples to the two keys above the Click Key defined on the Song-mode MISC page. Before you leave the Program Editor, be sure the KeyTrk parameter on the KEYMAP page is set to **100ct/key**.

When you save the edited click program, remember to replace it in the same location. If you save to a new location, set the new program number in the Click Program parameter on the Song-mode MISC page.

Live Mode

If you have the sampling option, you can use what we call Live mode. In Live mode, the K2600 takes any input signal and routes it through the VAST DSP algorithms and KDFX. You can connect any audio source—synths, mics, CD players, anything—to any of the K2600's sampling inputs, and treat that input as if it were a regular VAST program.

The easiest way to use Live mode is to use one of the factory programs (**740–749**).Some of the programs are optimized for certain applications (for example, guitar cabinet simulations), while others are meant to be used as templates.

You can't use Live mode and make samples at the same time, since both features use thesame internal components.

Creating a Live Mode Program

1. Press the **Sample** soft button to bring up the SampleMode page.

2. Set the Src parameter for the source you are using.

   For example, if you have plugged a microphone into the K2600's HiZ sampling input, choose **Ext**. Be careful if you choose **Int**, since you can inadvertently create a feedback loop.

3. Set the Mode parameter to **LiveIn**.

   Two samples will automatically be created: **197 Live Input L** at C 4 and **198 Live Input R**, also at C 4. The soft buttons on this page are disabled when you set Mode to **LiveIn**.

4. Use one or both of the live-input keymaps (**197** and **198**) in an existing LM program, or in one you create.

For a stereo program, set Stereo to **On** on the KEYMAP page in the Program Editor.

5. Edit the program's parameters for the effect(s) you want to use.
6. Play C 4, then input the audio source that you want to run through Live mode.

   Hint: Set VelTrk on the EditProg F4 AMP page to **0 dB**; otherwise the velocity with which you strike C 4 will affect your output. For alternative ways of triggering the sound (for example, with assignable controller buttons or pedals), edit the control setup.

   You should now hear your VASTed audio source through the K2600's Mix outputs.

Live Mode Programs

ID	Program Name
740	LM VirtualDesk 1
741	LM VirtualDesk 2
742	LM EQ Room Hall
743	LM TubeAmp_ Gtr
744	LM Synth Sliders
745	LM EQ StIm Hall
746	LM ParaFlange
747	LM EQ Overload
748	LM Filters
749	LiveMode Default

Live mode also includes two Live mode keymaps at 197 and 198 (Left and Right respectively).

Usage Notes

To use the programs, you must hold down a key (C 4, unless you're going for a special effect) for the inputs to run through VAST. An alternative way to trigger the sound is to edit the control setup found in the MIDI-mode TRANSMIT page. For example, on the SWITCH page in the Setup Editor, you could set the switch type (SwType) to note toggle (Note T), and set the destination (Dest) to C 4. This allows you to turn the program on and off via a button press, and keeps sound sustaining while the button is on. Rack-model owners can get the same results with any programmable switch controller. Keep in mind that if you change the Live mode program, you need to restrike a key (or button) for the signal to go through that program.

You can also edit the Live mode keymap to ignore release if you want to use the keyboard to activate Live mode.

You cannot sample and use Live mode together, the two functions use the same components.

Some Ideas for Using Live Mode

If you've ever used an old-fashioned mono analog synthesizer with an audio input (anything from a Moog Rogue to an ARP 2500 or Serge Modular), you know how much fun it can be to pass a musical signal through the synth and modify it in real time with the filters, envelopes, modulators, etc. Live mode brings that concept to digital synthesis, and lets you use all of the power of the K2600 on any kind of input signal.

For starters, you can simply hook up a CD player to one of the K2600’s sampling inputs, get a bunch of your favorite CDs, and start fooling around. (A turntable works well too.) Here are some ideas for going further:

Pitch Changing

Unlike an analog synthesizer, the K2600 makes it possible to alter the pitch of the incoming signal in real time. But the K2600 is not a conventional pitch shifter, so if you are used to working with such a device you will have to alter your thinking a little.

For example, when you bend the pitch down from the unity pitch (C 4), using a VAST function, it slows the playback of the incoming signal, but it doesn’t change the rate at which the signal is coming in—your CD is still spinning, and putting out a constant audio signal. So as you lower the pitch, the playback lags behind, and when you return the pitch to normal, the playback snaps back to the present—which means some of your audio literally disappears into the ether. If you bend the pitch down and hold it there for a while, eventually the buffer fills up and updates itself, and you will hear it snap forward in time, although the data playing will continue to be slowed down. Again, some of the audio disappears.

When you bend pitch upward, the K2600 plays buffered data from the input source, which enables the K2600 to “play ahead” of the input. You may hear some of the input data repeat when you release the pitch bend.

These details aside, all kinds of wonderful pitch effects are achievable. Here’s an example.

1. Start with Program **749 LiveMode Default**.
2. Go to the PITCH page.
3. Assign LFO1 to **Src1**, with a depth of **-200ct**.
4. In order to keep the playback from constantly crossing above unity, set the Coarse parameter to **-2ST**.

Or try these settings:

Src1	**MWheel**
Depth	**-1200c**
Src2	**LFO1** (On the LFO page, set LFO1’s MnRate to **.50Hz**, MxRate to **20.00**, and RateCt to **Data**.)
DptCtl	**MWheel**
MinDpt	**0ct**
MaxDpt	**1200ct**

Sometimes the LIve-mode audio will sound discontinuous as LFOs and the buffers get out of sync. You might be able to smooth out the rough spots by making another layer with no pitch alterations, and crossfading between the layers:

1. Duplicate the layer.
2. Clear all the settings on the PITCH page.
3. Go to the OUTPUT page and set Crossfade to **MWhl** on both layers.
4. On layer 1, set XFadeSense to **Rvrs**; on layer 2, set XFadeSense to **Norm**.

Now at the Mod Wheel extremes, you will hear only one layer or the other, while in the middle, you will hear a combination of the pitch-modulated signal and the unmodulated signal. By experimenting with FUNs, you can get more precise crossfades.

The program **744 LM Synth Sliders** includes this kind of crossfade, tied to the Pitch Wheel, to implement a 3-layer crossfade. Moving the Pitch Wheel up fades to a layer which is bending the pitch up. The surprise is that moving the Pitch Wheel down bends the pitch down, then up again, crossfading to a layer that is playing back in reverse! Yes, reverse playback works with Live mode: on the KEYMAP page, set PlayBackMode to **Rvrs**.

Arpeggiator

You can also do controlled pitch shifting on incoming audio using the arpeggiator. By constantly sending new note starts, it is possible to bend the pitch without losing the tempo of the incoming signal.

It can work in both directions, although when you are shifting signals up in pitch, you're "borrowing" the audio from a few seconds previous.

1. Go to Setup mode and select **97 Control Setup**.
2. Press **Edit**, and on the CH/PRG page, set the program to **749 LiveMode Default**.
3. With the program highlighted, press **Edit** and go to the AMPENV page.
4. To make the crossfading less choppy, you want short attack and release segments: set Att1 to **0.06/100**% and Rel1 to **0.10/0**%.
5. Press **Exit** and save the program to some new ID.
6. Now go to the ARPEG page and set the Active parameter to **On**.
7. Set the Duration parameter to **99**%.
8. For this example, set Order to **Simultaneous** and Beats to **1/32**.
9. Tempo should already be **120**.

Now play C 4 and you'll hear the live signal at the correct pitch. Play G 3 and you will hear the signal pitched down a fourth. You can use the ribbon or similar controller to bend the pitch smoothly. Going above unity pitch will cause a jump back into the past.

Experiment with the Tempo, the Beats setting, the Duration value, and the AMPENV parameters to get useful variations on the program. Remember that because we set the Order to **Simultaneous**, you can play several notes at once. And finally, try setting Glissando to **On**.

Sustained Notes and Loops

If the incoming signal is a single, sustained pitch, like a saxophone note, then you can consider the Live mode keymap to be playing a normal, looped sound. In this case, the fact that an upward bend jumps back a few seconds is no big deal because the sound hasn't changed much during that time.

With this technique, melodies or chords can be played based on a segment of a live performance. Keep in mind that, unless your incoming signal is a C, notes and chords played on the K2600 keyboard will be transposed relative to the incoming pitch. Also remember that a rhythm pitched an octave down will play at half the speed, while one pitched an octave up will play twice as fast. Fifths produce a 3-against-2 pattern. To keep some sort of relative sync with the live signal, you may want to experiment with retriggering the notes, perhaps using the arpeggiator, at some appropriate tempo.

If the passage you want to play is long, and the input signal isn’t so long—say, the sax player needs to take a breath—you may run into a problem as the K2600 tries to play the buffer where the audio was interrupted. If the input signal is mono, you might be able to overcome this by using a delay line to “hold” the signal. The delay line could be part of VAST, or it could be an external device, but either way its output is sent back to the K2600 through the unused Live mode input channel.

Chord Progressions

Record a few bars of block chords—all notes under C 4—into the sequencer, using a simple quarter-note or half-note pattern. What sound you use doesn’t matter. Now replace the program on the recorded track with the Live mode default program. Play back the sequence (you will probably want it to loop), and at the same time play single notes from an external instrument into the K2600, at the same rhythm as your recorded chords. If you change the notes on the instrument, the chords will transpose. If you play intervals or chords, you're on your own as to the consequences!

Feedback

Live mode gives you the ability to feed back a live signal into itself, similar to pointing a microphone at the speaker it's sending audio to. Before you hook anything up, turn the volume down as low as you can.

Now go to the Sample page and set Source to **Internal**. Go to a multi-layer ROM program of your choice, and go to the Import page. Import Layer 1 from the Live-mode default program.

Play one note, then a few. As you play more notes, the noise will build up. You’ll have a better time controlling the feedback loop if you have a healthy delay, with no dry path around it, in the loop. Perhaps add a little modulation of the loop to provide some pitch shifting, a big reverb, and a compressor to keep from blowing your ears out. Inject a little something from the synthesizer to get things started—and you are instantly transported to an alien dimension.

For more complexity, split the incoming signal and run it through multiple VAST layers in parallel—you can use up to 32, each one processing, panning, and routing the signal differently. You can crosslink the inputs and outputs (right into left, left into right) to create a double feedback loop for even more fun.

Chapter 15
Advanced KDFX

This chapter describes the organization of KDFX in detail.

Personalities

KDFX can be said to have two distinct "personalities," depending on the setting of the FX Mode parameter on the Effects-mode page.

If FX Mode is set to **Master**, then all KDFX parameters are set within the current studio, and remain unchanged unless you edit the studio or any of its FX presets.

If FX Mode is set to anything else —**Program**, **Setup**, or **Auto—**then one or more parameters within the studio may be under the control of an outside source, such as MIDI or one of the K2600's control sources, and can be continuously changed in real time without editing the studio or any FX presets.

For the sake of clarity, we'll begin by discussing KDFX only in Master mode. For information about real-time control, turn to page 15-21.

*For starters, set the FX Mode parameter on the Effects-mode page to **Master** and the Channel FX Chan parameter to **None** before going further into this chapter.*

Navigating KDFX

The largest component in KDFX is the studio. The studio encompasses all of KDFX's signal routings, processing algorithms, and processing parameters. When you change any parameter in a studio, you are potentially creating a new studio, just as changing a parameter in a program creates a different program (if you save it). The user interface within a studio is organized according to the following diagram:

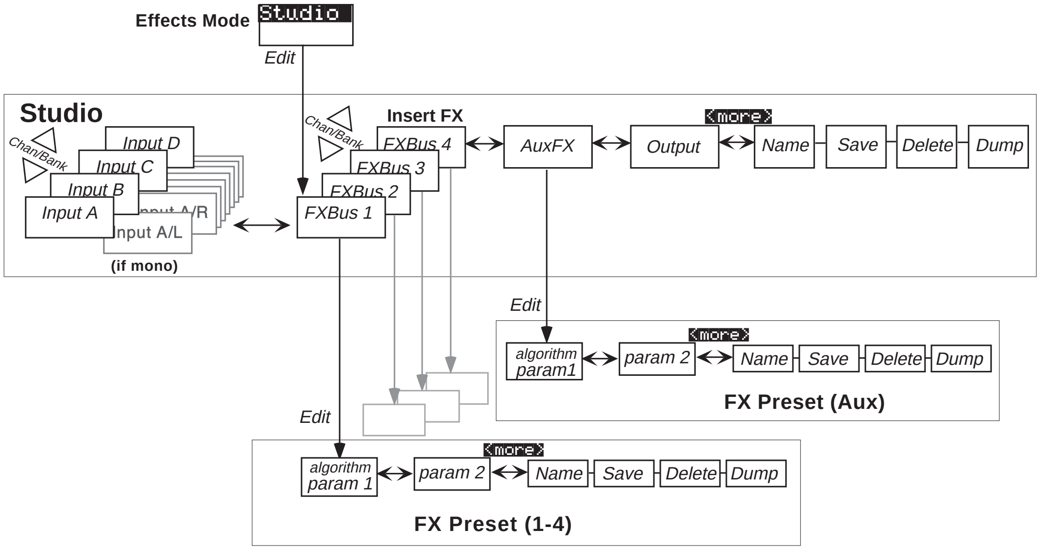
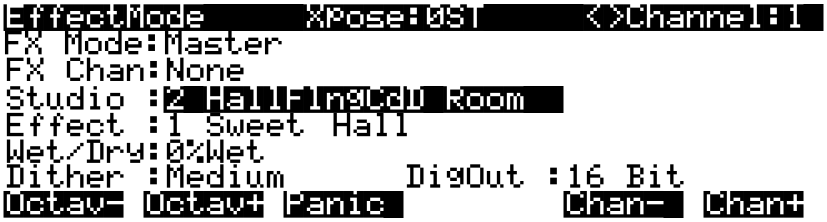


Figure 15-1 The KDFX User Interface

To select a studio, go to the main Effects page if you're not already there (from Program mode, press **Effects** to go into Effects mode), and move the cursor so the studio name is highlighted.

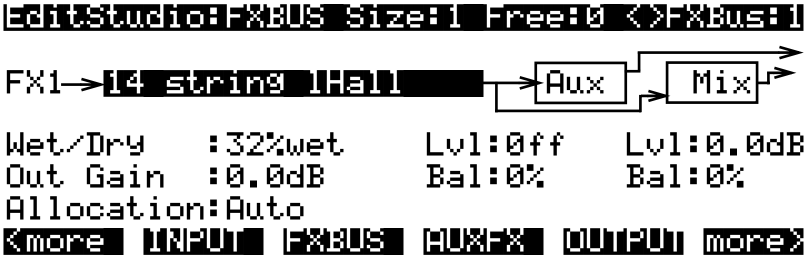


```
EffectMode      XPose:0ST      <>Channel:1
FX Mode:Master
FX Chan:None
Studio :2 HallFlngCdD Room
Effect :1 Sweet Hall
Wet/Dry:0%Wet
Dither :Medium       DigOut :16 Bit
Octav- Octav+ Panic          Chan-  Chan+
```

FXBus

When the studio is highlighted, pressing **Edit** goes to the Studio Editor (EditStudio) level, and the FXBUS page. There are four FXBUS pages, one for each FXBus in the studio. These four buses are the inputs to KDFX, and receive the output from the K2600's sound engine, as defined on the OUTPUT page in the Program Editor.

Select the desired FXBUS page using the **Channel**/**Bank Up** and **Down** buttons. If you have just entered the Studio Editor, the first FXBUS page you see will be the one for FXbus1. Once you are inside the Studio Editor, however, when you press the **FXBUS** soft button from another page, it will take you to whichever FXBUS page you were *last* looking at.



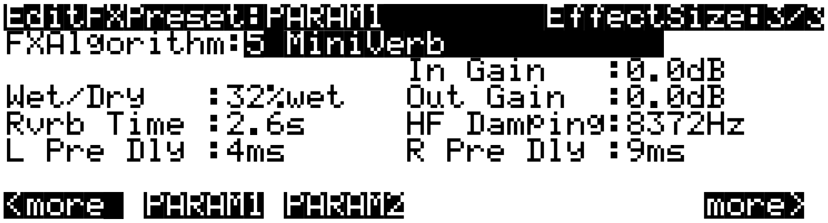
```
EditStudio:FXBUS Size:1 Free:0 <>FXBus:1
FX1→14 string 1Hall     →Aux   →Mix
Wet/Dry   :32%wet    Lvl:Off    Lvl:0.0dB
Out Gain  :0.0dB     Bal:0%     Bal:0%
Allocation:Auto
```

The four FXBuses are the equivalent of four effects processors inserted into the effects loop of a mixing console. Therefore, they are also known as the *Insert effects*.

Parameters

When you are on an FXBUS page, highlighting the name of the FX preset (or any of the parameters directly below it) and pressing **Edit** accesses the first page of parameters for that FX preset. An FX preset is an object within the studio, much like a keymap is an object within a program. The same FX preset can be used in more than one studio, or more than once in the same studio (provided you don't run out of PAUs).

The first EditFXPreset Parameter page includes the algorithm on which the FX preset is based. The soft buttons take you to additional pages of parameters. Depending on how complex the algorithm is, there may be as many as four parameter pages in an FX preset. Algorithms are in the KDFX ROM, and are not normally changeable, deletable, or saveable by the user. Like ROM samples, they are simply always there. (You can load additional algorithms from disk, however, as they become available from Kurzweil.)



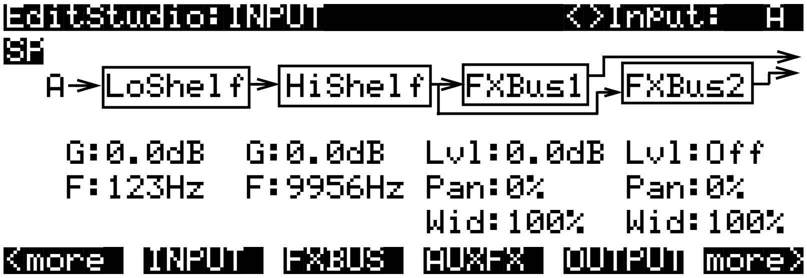
```
EditFXPreset:PARAM1          EffectSize:3/3
FXAlgorithm:5 MiniVerb
                     In Gain   :0.0dB
Wet/Dry   :32%wet    Out Gain  :0.0dB
Rvrb Time :2.6s      HF Damping:8372Hz
L Pre Dly :4ms       R Pre Dly :9ms
```

Pressing the **<more>** soft buttons gives you access to the Name, Save, Delete, and Dump pages for the FX preset.

Pressing **Exit** goes back to the FXBUS page, and if you have made any changes in the FX preset you will be prompted to save it. If the FX preset in this bus is the same as on another bus (either in this studio or another), then any changes you make (and save) will affect all buses using that FX preset.

Input

From the FXBUS page, or anywhere inside the Studio Editor, pressing the **INPUT** soft button brings you to the Input A page.



```
EditStudio:INPUT                <>Input:   A
SF
  A→LoShelf→HiShelf→FXBus1→FXBus2

    G:0.0dB  G:0.0dB  Lvl:0.0dB Lvl:Off
    F:123Hz  F:9956Hz Pan:0%    Pan:0%
                      Wid:100%  Wid:100%
<more   INPUT   FXBUS   AUXFX   OUTPUT  more>
```

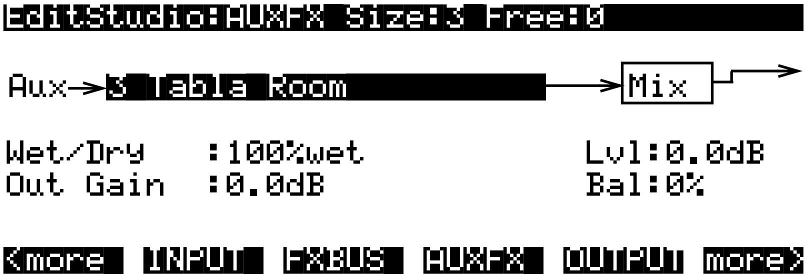
The other Input pages—B, C, and D—are selected by using the **Chan**/**Bank Up** and **Down** buttons. Depending on how the inputs are configured—stereo or mono—there will be from four to eight Input pages.

Similar to the FXBUS pages, the first time you look at an input page after entering the Studio Editor, it will be the Input A page (or, if it's mono, the Input A/L page). Once you are in the Studio Editor, when you press the **INPUT** soft button from another page, it will take you to whichever input page you were *last* looking at.

Aux FX

From inside the Studio Editor, a soft button accesses the AUXFX page. This is a separate effects bus, which can be used by itself, or in a chain following one or more of the FXBuses.

Like the FXBuses, you can view and edit the FX preset's parameters, including its algorithm, by highlighting the FX preset's name and pressing **Edit**. As on the insert FXBuses, the FX preset on the Aux bus has up to four pages of parameters, and the **<more>** soft buttons access Name, Save, Delete, and Dump pages for the FX preset. The same FX preset can be used in the Aux bus as in any of the insert FXBuses.



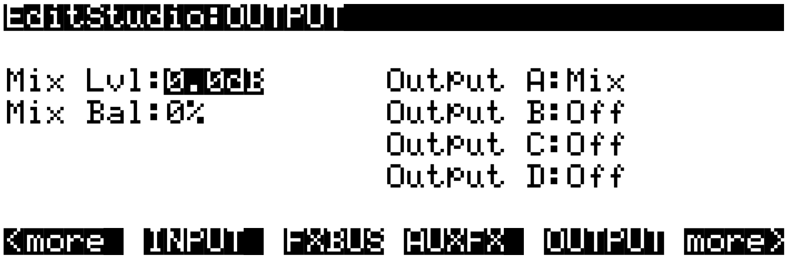
```
EditStudio:AUXFX Size:3 Free:0

Aux→3 Tabla Room            →Mix

Wet/Dry    :100%wet          Lvl:0.0dB
Out Gain   :0.0dB            Bal:0%

<more   INPUT   FXBUS   AUXFX   OUTPUT  more>
```

The AUXFX page can also be accessed from any of the FXBUS pages by placing the cursor on the box labelled Aux and pressing **Edit**.

Pressing **Exit** goes back to the AUXFX page on the EditStudio level. If you have made any changes in the FX preset, you will be prompted to save the FX preset.

Output

In the Studio Editor, pressing the **OUTPUT** soft button accesses the OUTPUT page, where the KDFX’s “virtual” outputs are assigned to the K2600’s physical outputs.



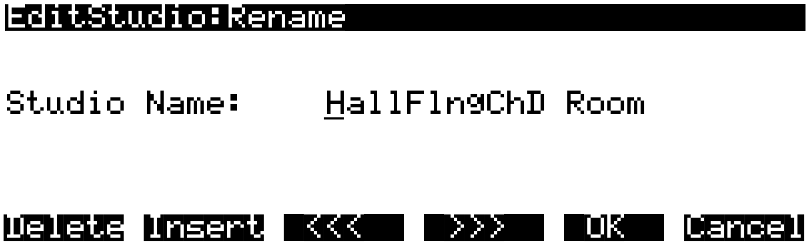
```
EditStudio:OUTPUT

Mix Lvl:0.0dB        Output A:Mix
Mix Bal:0%           Output B:Off
                     Output C:Off
                     Output D:Off
```

The OUTPUT page can also be accessed from any of the FXBUS pages by placing the cursor on the box labelled Mix and pressing **Edit**.

Name, Save, Delete, Dump

From any of the EditStudio pages, pressing either of the **<more>** soft buttons accesses Name, Save, Delete, and Dump pages for the studio. Studios are stored in RAM, like K2600 programs, and when a studio is recalled, all of its associated FX presets and parameters are recalled with it. Studios in ROM occupy slots in the zeros, 100s, 700s, and 800s banks. You may override these if you like, or use the RAM banks (200-699 and 800-999) for your studios. The ROM studios are always there, and if you delete a studio that you’ve stored in a ROM slot, the original ROM studio will pop up in its place.

When you save a studio, you can also rename it, using the standard naming dialog:



```
EditStudio:Rename

Studio Name:    HallFlngChD Room
```

The Compare and FX Bypass Buttons

As with all K2600 objects, the **Compare** (**Disk**) button lets you switch back and forth between the last saved version of whatever you are editing and its current state.

If you are on a page at the EditStudio level, **Compare** toggles between the last-saved and current versions of the studio. If you are inside an FX preset, on the EditFXPreset level, **Compare** toggles between the last-saved and current versions of the FX preset.

If you have changed any algorithms in an FX preset or studio during the current editing session, the **Compare** button will switch back to the old algorithms. This can create some short-term “holes” in the audio output when the signal momentarily goes dry—see the section on switching studios in real time on page 15-21.

The **Effects**/**FX Bypass** button, when you are in the Studio Editor, bypasses all of the FX presets (all of the Insert FX and the Aux FX) in the current studio, so that you can hear the signals without processing. It does not, however, change the EQs, gains and balances, or signal routings—those will continue to affect the signal you hear.

Exploring the Studio Parameters

We'll explore the parameters within the studio in the order in which they affect the signal path, starting on the Input page.

Input Section

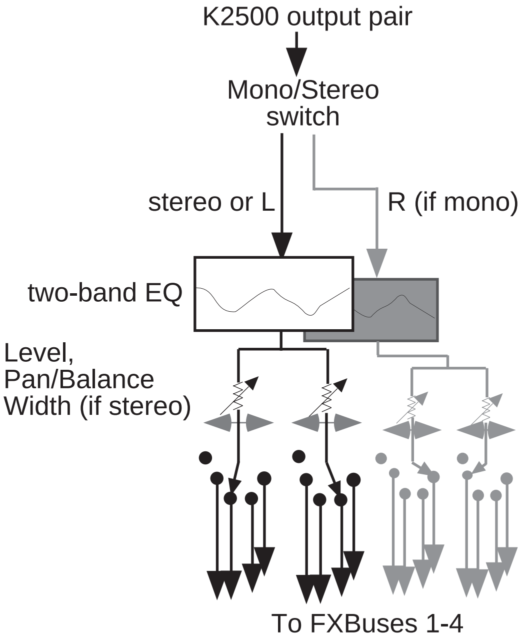


Figure 15-2 The KDFX Input Editor

Inputs are referred to as A, B, C, and D, and correspond to the four output buses (KDFX-A through KDFX-D) from the K2600's Program Editor. These signals can be treated as stereo pairs or as individual mono signals. This is determined by the first parameter on an Input page, the Mono/Stereo switch.

If this switch is set to **M**, then the left and right channels of the selected program output pair are split up, and each is given its own Input page, with EQ and FXBus routings. If it is set to **SP** (Stereo with Pan) or **SB** (Stereo with Balance), the two channels of the pair are processed in parallel.

Selecting the Mono/Stereo mode on one input bus does not affect any of the others, and you can have any combination of stereo and mono inputs in a studio. Therefore, there can be anywhere from four to eight Input pages in a studio.

The **Chan**/**Bank Up** and **Down** buttons let you move among the Input pages.

The Arrow Meter

On an Input page, whenever there is signal present on its bus, the arrow next to the letter of the bus flashes. This is a good way to check that you have set up your program output routings correctly. More on this later.

EQ

The input signal first passes through two equalizers. These equalizers are independent of each other, but the signal is chained: it goes through the left one, then the right one. Each equalizer has a frequency (F) control and, depending on the mode, a gain (G) control. The mode of each EQ module is changed by placing the cursor in the appropriate block and turning the Alpha wheel or using the **Plus/Minus** buttons. There are eight modes for the first EQ, and six for the second:

None No effect, the signal passes through unchanged.

LoShelf Boosts or cuts frequencies below the F value by G decibels.

HiShelf Boosts or cuts frequencies above the F value by G decibels.

LoPass1 Cuts frequencies above the F level with a 6 dB/oct (1-pole) slope.

LoPass2 Cuts frequencies above the F level with a 12 dB/oct (2-pole) slope.

HiPass1 Cuts frequencies below the F level with a 6 dB/oct (1-pole) slope.

HiPass2 Not available on the second EQ. Cuts frequencies below the F level with a steeper 12 db/oct (2-pole) slope.

ParaMid Not available on the second EQ. Provides a cut or boost centered around the F frequency. The bandwidth of the equalizer is two octaves. There is an illustration of the action of this equalizer mode on page 16-28.

FXBus Sends

Following the equalizers are the insert FXBus sends. Each input has two sends. Change the destination of each send by placing the cursor on it and doing the usual thing with the Alpha wheel or **Plus/Minus** buttons.

Either send can be assigned to any of the four FXBuses, or to **None**, with one exception: the two sends on a particular input cannot both be assigned to the same FXBus. So, for example, if the first send on Input B/L is assigned to FXBus2, the second send from Input B/L cannot also be assigned to FXBus2. You can, however, assign as many *different* inputs to the same FXBus as you like—including the two channels from a mono pair.

Each of the FXBus sends has a level parameter (Lvl) that determines the gain of the signal going to that send. Maximum level is **24.0 dB**, and minimum is **-79.0 dB—**there is also an **Off** position. **0.0 dB** is unity gain.

The FXBus sends are stereo, and if an input is stereo, both channels go to the send.

Mono Inputs (M)

If an input is mono, then each of its FXBus sends has a Pan parameter. This determines how the signal is distributed between the left and right channels going to the FXBus: **-100%** is left channel only, **100%** is right channel only, and **0%** is both channels equally.

Stereo Inputs with Pan (SP)

If the input is set to **SP**, then each FXBus send has a Pan parameter and a Width parameter. The Width parameter determines how much separation there will be between the left and right input signals as they are sent to the FXBus: assuming Pan is set to **0%**, a Width of **100%** means the signals will be completely separate, while **0%** means they will be combined into “dual mono.” Negative numbers flip the channels around, so that **-100%** means the channels are separate, but with left and right reversed, while **-50%** means they are reversed and partially blended.

The Pan control maintains the stereo image, but “tilts” it one direction or the other. At **0%** there is no change to the signal, while at **100%** it all goes to the right channel. At **50%**, what had been hard left will now be in the center, and what had been in the center will now be halfway between center and right. Negative values tilt the signal to the left.

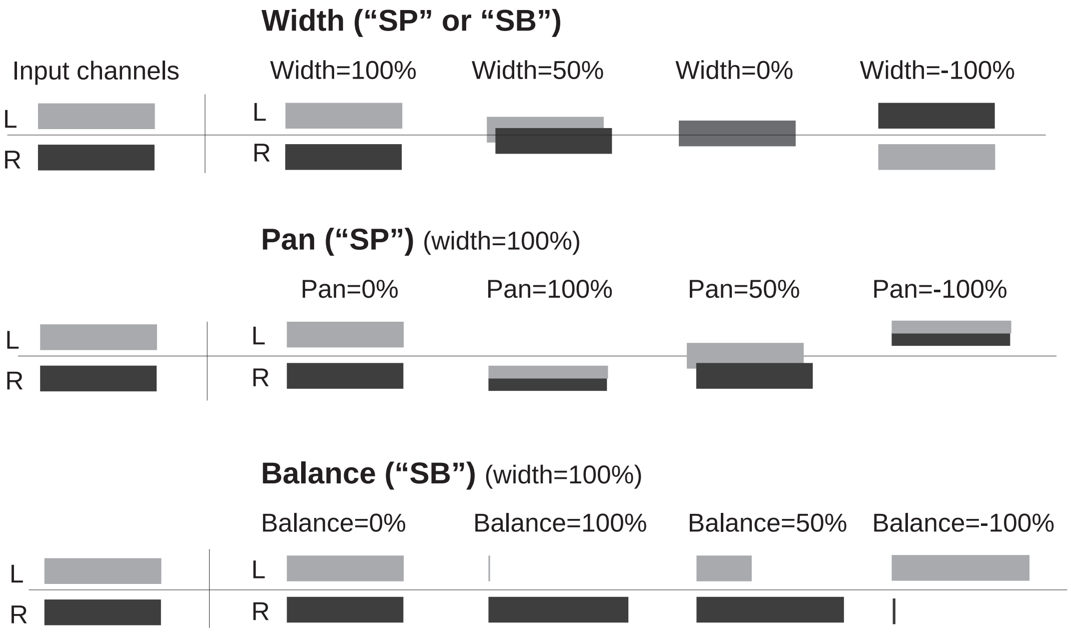


Figure 15-3 Width, Pan, and Balance controls

A Word About Gain

The Pan, Balance, and Width controls all have constant power curves, so that the combined signal level doesn’t change when you move the signals from side to side. However, if you use several Pans or Balances on a signal to keep the channels isolated throughout the entire signal chain (for example, if the Input send is panned 100%, and so are the Aux send, the Mix send, and the Mix output), you can increase the gain of the signal considerably.

Each stage of hard-panning adds 3 dB, so the increase in gain when the signal reaches the final output can be as high as 12 dB. In this case, you may want to trim the level at various stages to keep the signals from getting too hot.

Effects Buses

The four insert Effects buses (FXBuses) receive the signals from the Input Editor and process them. Press the **FXBUS** soft button to go to one of the FXBUS pages—the first time you do this after entering the Studio Editor, it will be the **FXBus1** page.To go to the other FXBUS pages, use the **Chan**/**Bank Up** and **Down** buttons. The number of the FXBus appears in the upper right corner.

The Arrow Meter

There are arrow meters on the FXBUS pages as well, right next to the number of the FXBus. These tell you when signal is coming into the bus, and also when signal is present *inside* the bus, so if you have a long reverb or repeating delay, for example, the arrow will keep flashing as long as the processing is going on.

FX Preset

The first parameter on an FXBUS page is the FXBus's FX preset. Set the cursor on it, and use the Alpha wheel to scroll through the FX presets currently in memory. If an FX preset name comes up in parentheses, for example, **(Really Big Plate)**, it means there is not enough processing power (PAUs) available at the moment to use this FX preset in this FXBus. We'll get to PAUs in a moment. **199 No Effect** is a "blank" FX preset, in which all signals pass straight through without any processing. It can be used as a starting point for creating your own FX preset. If you want to set up a "dummy" effects bus to pass signal directly to the Aux bus, use **0 None**.

Bus Outputs

The parameters on the right side of this page determine how the effected signal gets to the KDFX outputs. Each FXBus has four outputs, all of which are stereo:

- Its own dry (pre-effect) output
- Its own wet (post-effect) output
- The Mix bus
- The Aux bus

The output to the Mix effects bus has a level control with a range of **-79.0** to **+24.0 dB**, and an **Off** position. It also has a Balance control that works similarly to the Balance control on the inputs, by setting the relative levels of the two output channels. The signal is mixed with similar signals from the other FXBuses onto the Mix bus, which can be accessed on the OUTPUT page.

The output to the Aux bus has an identical pair of controls. Its signal goes to the global Aux effects bus, where it is mixed with similar signals from the other FXBuses, and then put through the Auxiliary Effects processor. From there it can be accessed on the OUTPUT page.

There is no external level control over the output of the FXBus itself—it just shows up, in pre-effect and post-effect versions, on the OUTPUT page.

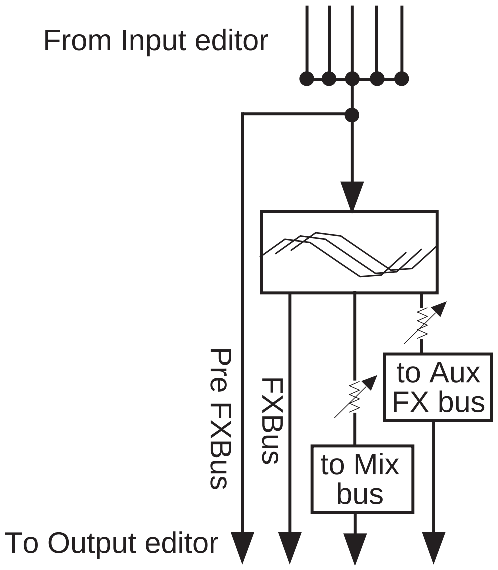


Figure 15-4 FXBus Signal Routing

The Aux Bus

The Auxiliary effects bus is accessed by pressing the **AUXFX** soft button on any page in the Studio Editor. It contains an FX preset, which is separate from those in the insert FXBuses—although it can be the same FX preset that is in use on one or more of the insert FXBuses. There is no Allocation parameter, because the Aux bus has a fixed allocation of 3 PAUs. Only a very few highly complex FX presets require more than 3 PAUs, so as you scroll through the FX presets here you won't see many names in parentheses.

Aux Bus Outputs

There are two outputs from the Aux bus: itself, and a feed (post-effect) to the Mix bus. The feed to the Mix bus, where it is combined with other Mix bus feeds from the four FXBuses, has level and balance controls. The Aux bus's own output has no post-effect controls, and goes right to the OUTPUT page.

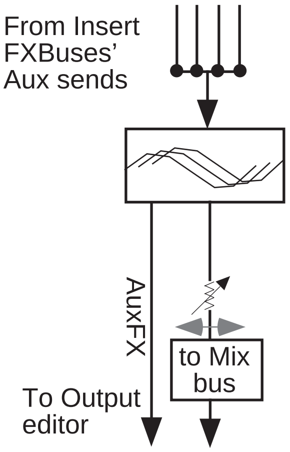


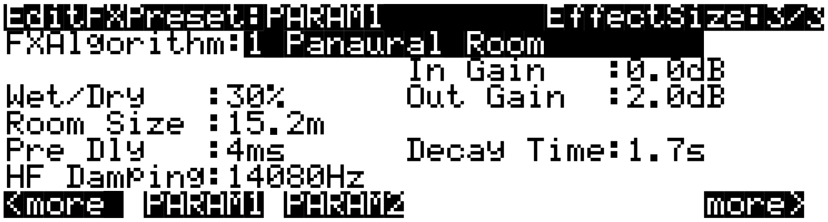
Figure 15-5 Aux Bus Outputs

About FX Presets

FX presets are where the processing takes place in KDFX. Each of the insert FXBuses and the Aux bus have their own FX preset, and they are all independent of each other. If you think of a studio as being the equivalent of a K2600 program, then an FX preset is the equivalent of a layer or keymap.

In any of the FXBuses, the FX preset is edited by placing the cursor on the name of the FX preset and pressing **Edit**.

The first parameter page, or PARAM1, appears. The algorithm that the FX preset is based on is selected at the top of this page. Highlight the name on the FXAlgorithm line, and scroll through the various algorithms. The algorithms are in the K2600's ROM (like ROM samples), and are not changeable by the user. As you change algorithms, the parameters displayed on the page change accordingly.



```
EditFXPreset:PARAM1         EffectSize:3/3
FXAlgorithm:1 Panaural Room
                     In Gain   :0.0dB
Wet/Dry   :30%       Out Gain  :2.0dB
Room Size :15.2m
Pre Dly   :4ms       Decay Time:1.7s
HF Damping:14080Hz
<more  PARAM1 PARAM2             more>
```

Notice also that as you scroll the algorithms, the EffectSize parameter in the upper-right corner changes. This parameter shows how many Processing Allocation Units (PAUs) the currently selected algorithm requires, followed by how many are available for this FX preset. If, for example, EffectSize is **2/3**, that means the algorithm requires 2 PAUs, and there are 3 PAUs available. More about PAUs soon.

Just below and to the right of the algorithm name is an Input Gain parameter, which adjusts the level of the signal coming into the FX preset from the input(s) sending to it. The relative level of the various inputs is determined on the Input pages, but you can change the overall level here—

for example, if you've combined several inputs into one FXBus and the signal is too hot for the FX preset, you can pad it down here. The Trim is adjustable from **Off**/**-79.0 dB** to **+24 dB**.

Below the Input Gain is usually (but not always) an Output Gain, which sets the level of the signal leaving the FX preset, which can be further modified by the Mix and Aux sends on the FXBUS page.

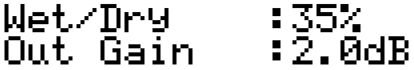
The other parameters that appear on this page are determined by the algorithm. Each algorithm has its own set of parameters, which may take up as many as four pages, accessed using the soft buttons **PARAM2**, **PARAM3**, etc. The parameters associated with each algorithm are discussed in detail beginning on page 15-35, and at the beginning of Chapter 10 of the *Musician's Reference*.

When you change a parameter on one of these pages, you have changed the FX preset, and if you want the change to be permanent, you must save the FX preset. Pressing either of the **<more>** soft buttons accesses Name, Save, Delete, and Dump pages for the FX preset. FX presets are stored in RAM, just like K2600 keymaps. When an FX preset is recalled, either by itself (from within the Studio Editor) or as part of a studio, its associated algorithm and all parameters are recalled with it.

There is another way to edit an FX preset's parameters without altering the FX preset itself, and that is by using bus overrides.

Bus Overrides (Bus Mods)

We haven't yet talked about the two parameters that are on the Insert FXBUS and AUXFX pages, directly underneath the name of the FX preset. These are called "bus overrides" or "bus mods," and they allow you to change parameters within an FX preset without actually going into the FX preset.



```
Wet/Dry     :35%
Out Gain    :2.0dB
```

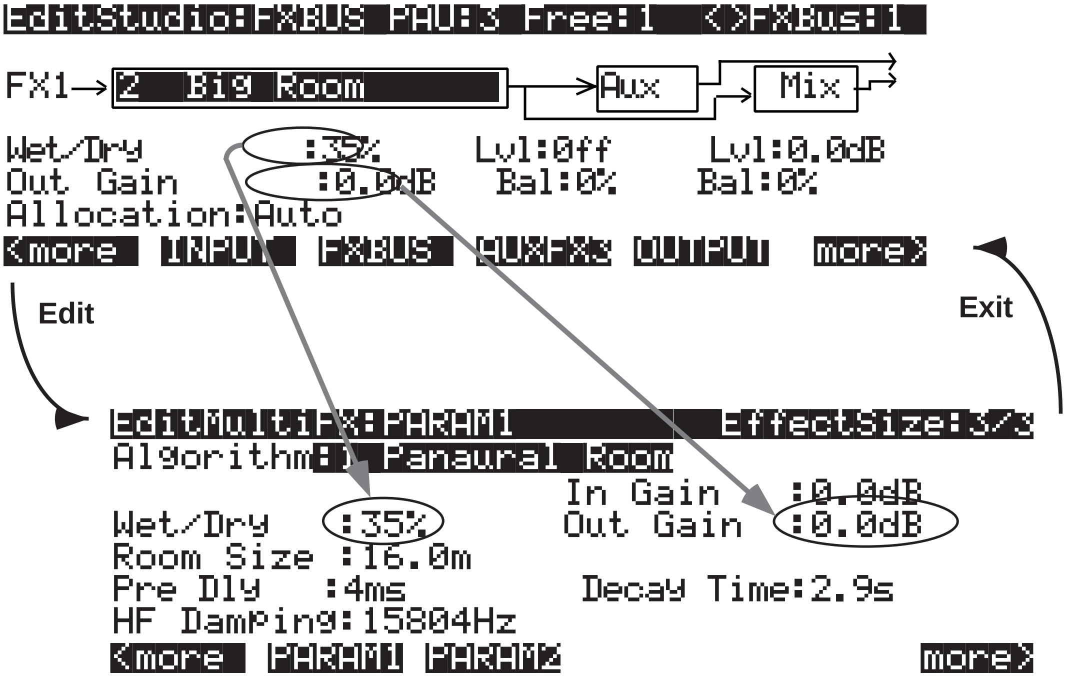
For example, the bus overrides on FX1 are often Wet/Dry mix and Output gain. Normally, these parameters would be found inside the FX preset, and if you changed them, you'd have to save the new FX preset in order to keep the changes.

Instead, using bus overrides, you can adjust these two parameters and hear what they sound like while you are adjusting them *without* going into the FX preset. When you save the studio, these parameter values are saved, but they are not part of the FX preset—they are part of the *studio*. Therefore, the FX preset remains unchanged (and if the FX preset is in use elsewhere, it hasn't changed there), but these two parameters *in this particular FXBus* have been altered.

Wet/Dry and Out Gain are the default bus overrides you will encounter most often, but in some algorithms and FX presets, other parameters are accessed as bus overrides. For example, on some compressor algorithms, the first bus override is an In/Out switch; and on some dual-channel delay and filter algorithms, the overrides are separate Wet/Dry controls for the left and right channels.

Making and Breaking Bus Overrides

Some studios supplied with KDFX, when you first encounter them, have the bus mods in place, but they are not engaged—that is, they're not actually overriding anything, but instead simply show the values of the corresponding parameters inside the FX preset unchanged. To see this, choose an FX preset on the FXBUS page and look at the values of the override parameters, **Wet/Dry** and **Out Gain**. Now go inside the FX preset by highlighting the FX preset's name and pressing **Edit**, and find those two parameters on the first parameter page (PARAM1). You'll see their values are the same as on the FXBUS page.



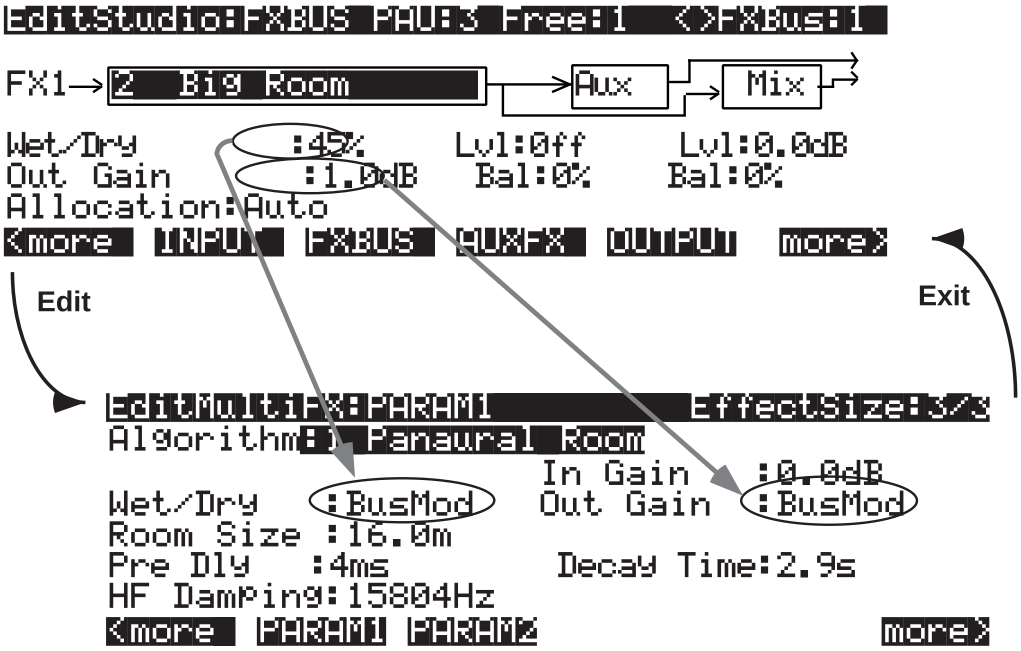
```
EditStudio:FXBUS PAU:3 Free:1  <>FXBus:1
FX1→2  Big Room          →Aux    Mix
Wet/Dry      :35%   Lvl:Off    Lvl:0.0dB
Out Gain     :0.0dB  Bal:0%    Bal:0%
Allocation:Auto
<more  INPUT  FXBUS  AUXFXs OUTPUT  more>
```

```
EditMultiFX:PARAM1          EffectSize:3/3
Algorithm:1 Panaural Room
                    In Gain   :0.0dB
Wet/Dry   :35%      Out Gain  :0.0dB
Room Size :16.0m
Pre Dly   :4ms       Decay Time:2.9s
HF Damping:15804Hz
<more  PARAM1 PARAM2                more>
```

Figure 15-6 Bus Overrides in Place, But Not Engaged

Press **Exit** to go back to the FXBUS page.

To engage a bus override, you have to use it, which you do by moving the parameter away from its nominal value on the FXBUS page. Do that, and then go back inside the FX preset.You'll see that the value of the parameter you've changed is now shown as **BusMod**, meaning that the FX preset parameter is under the control of the bus override.



```
EditStudio:FXBUS PAU:3 Free:1  <>FXBus:1

FX1→ 2  Big Room      → Aux → Mix

Wet/Dry       :45%    Lvl:Off    Lvl:0.0dB
Out Gain      :1.0dB  Bal:0%     Bal:0%
Allocation:Auto
<more  INPUT  FXBUS  AUXFX  OUTPUT  more>
```

```
EditMultiFX:PARAM1           EffectSize:3/3
Algorithm:1 Panaural Room
                       In Gain   :0.0dB
Wet/Dry   :BusMod      Out Gain  :BusMod
Room Size :16.0m
Pre Dly   :4ms         Decay Time:2.9s
HF Damping:15804Hz
<more  PARAM1  PARAM2                 more>
```

Figure 15-7 Bus Overrides Engaged

If you now change the parameter value from *inside* the FX preset, it *breaks* the bus override. By doing this, however, you have changed the FX preset, and when you leave the FX preset you will be asked if you wish to save it. If you then save the FX preset, the bus mod will be permanently broken, but if you *don't* save the FX preset, it remains in its previous state, which means the bus mod *remains intact*.

Since bus overrides are part of the studio, they are not saved permanently until the studio is saved.

Changing the Bus Overrides

The default bus overrides that come up when you are working in a studio are often convenient, but you aren't required to use them as they are. You can assign *any* two of the FX preset's parameters to the bus overrides. (You cannot, however, assign both bus overrides to the same parameter.) When you save the studio, the parameters you've selected for the bus overrides on each FXBus are saved, along with their values.

If you set up a bus override and adjust a parameter's value, and then change your mind and assign a *different* parameter to that override, the value of the first parameter (the one no longer assigned to a bus override) will revert to its original value—that is, the value set *inside* the FX preset. This can be a helpful feature, in that it means you can use bus overrides as a "window" into an FX preset by scrolling through the various parameters. But keep in mind that you can't use this method to tweak parameters, because as soon as you move on, the parameter you tweaked gets "untweaked."

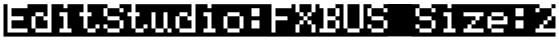
Allocation

Understanding PAU allocation and how it works is very important for using KDFX to its fullest potential.

What’s a PAU?

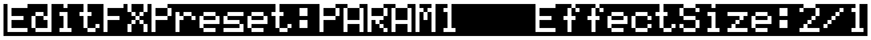
The basic unit of signal processing in KDFX is the Processing Allocation Unit, or PAU. There are a total of 4 PAUs that can be split among the four insert FXBuses, and another 3 PAUs for the Aux bus. These two sets of PAUs are distinct and are not interchangeable.

The number of PAUs on an FXBus is determined by the algorithm inside its FX preset. Different algorithms require different numbers of PAUs, as shown in the Size parameter at the top of the screen when you are on the FXBus edit page:



EditStudio:FXBUS Size:2

or inside an FX preset:



EditFXPreset:PARAM1   EffectSize:2/1

A simple delay, flanger, compressor, exciter, or small reverb uses only 1 PAU. A complex phaser, pitcher, or multiband tone control uses 2 PAUs. A really complicated reverb or graphic equalizer may use 3 PAUs. Only a handful of really wild algorithms use 4 PAUs.

The Allocation parameter on each FXBUS page determines how many PAUs are available for the FX preset assigned to that bus. If the parameter is set to **1**, then only FX presets that use algorithms requiring 1 PAU will be available for the bus. If you try to assign an FX preset that requires more PAUs, its name will show up in parentheses, and the sound will pass through the bus unprocessed.

If you know you want a certain FX preset in a particular FXBus, you can select it, and then set the Allocation parameter for that bus to match the PAU requirements of the FX preset.

A PAU is a Terrible Thing to Waste

Be careful not to set the Allocation parameter *too* high. If you set it on some bus to 3, for example, and you are using an FX preset on that bus that requires only 1 PAU, the other 2 PAUs are being wasted, since they are not available to be assigned to other buses. The other buses’ Allocation parameters will not go higher than 1, and therefore no FX presets whose Size is greater than 1 can be selected for any of them.

PAUs are normally allocated on a first-come, first-served basis. If you set one FXBus to a PAU of **3**, then you will be able to set the Allocation on the next bus you set to **0** or **1** only. If you then set that second bus to **1**, then you won’t be able to set a value greater than **0** for *any* of the other buses.

If you know that you won’t be using an FXBus, or that you’ll be using it only as a “dummy” to route signals somewhere else, you can set its Allocation to **0**.

Auto Allocation

On any of the buses, you can set Allocation to **Auto**. **Auto** means that the PAU allocation for that bus will automatically adjust itself to the currently selected FX preset. However, **Auto** obviously cannot create PAUs when they are in use elsewhere, and Allocations that have been manually set take precedence over Auto Allocations. So for example, if you set the Allocation of FXBus1 to **3**, and set the Allocation of FXBus2 to **Auto**, the maximum number of PAUs available to FXBus 2 is still only 1, and if you try to load a Size-2 FX preset into FXBus2, it won't work and the FX preset's name will show up in parentheses.

If *all* buses' Allocations are set to **Auto**, then PAUs are not allocated first-come, first-served, but instead are allocated in *numerical* bus order: if an FX preset requiring 3 PAUs is loaded into FXBus1, then only 1 PAU will be available for the other buses, regardless of which FX preset got assigned to which bus first.

A parameter called Free appears at the top of every FXBUS page, telling you how many PAUs in the current studio are unallocated and available.

Effect Size

While you're editing an FX preset and selecting algorithms for it, the EffectSize parameter at the upper right of the EditFXPreset page informs you of the PAU situation. The first digit in this parameter's value is the number of PAUs the currently selected algorithm requires, and the second digit is the number that have been allocated to the bus, either manually or automatically. If the first digit is larger than the second, the algorithm is not available, and if you choose it, its name will show up in parentheses and the sound will pass through unprocessed—just like an unavailable FX preset on an FXBUS page.

If the FXBus's Allocation is **Auto**, when you change the algorithm inside the FXPreset Editor, *both* digits of the EffectSize display will change. If you call up an algorithm that requires more PAUs than are currently available, the second digit will change to **0**—since the algorithm can't be loaded, the PAUs are freed up for use elsewhere.

Designing with PAUs

One simple way to use Allocations when designing a KDFX studio is to put all of the buses in Auto, and start with your most complex processing on FXBus 1, then assign other FX presets to the other buses as they are available.

PAUs on the Aux Bus

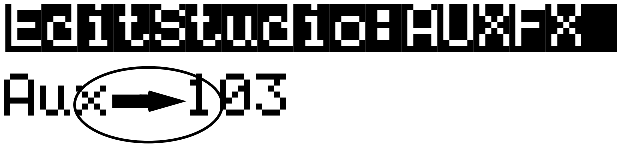
The Aux bus is a whole separate processor with 3 of its own PAUs. Allocation is not an issue with the Aux bus, since its 3 PAUs are assigned to it permanently. (There is a fourth PAU in the Aux bus, but it is used for mixing and routing, and so it's not available.)

Any FX preset with a PAU requirement of 1, 2, or 3 can be used. When editing the Aux bus, there's no need for an Allocation parameter or a Free parameter. However, when you are scrolling through algorithms, the EffectSize parameter will be displayed, with its second digit always **3**. Should you try to access an algorithm on the Aux bus that requires 4 PAUs, the name of that algorithm will be displayed in parentheses.

The Aux bus is generally used as a global processor, but it doesn't have to be. You can set up one FXBus as a "dummy"—set its FX preset parameter to **0 None**—and send its output to the Aux bus, and at the same time turn off the Aux sends from all the *other* buses. Now the Aux bus can function as an insert FXBus, with 3 PAUs all its own. So if you run into trouble with PAUs on the insert FXBuses and can sacrifice a global effect, this is one solution.

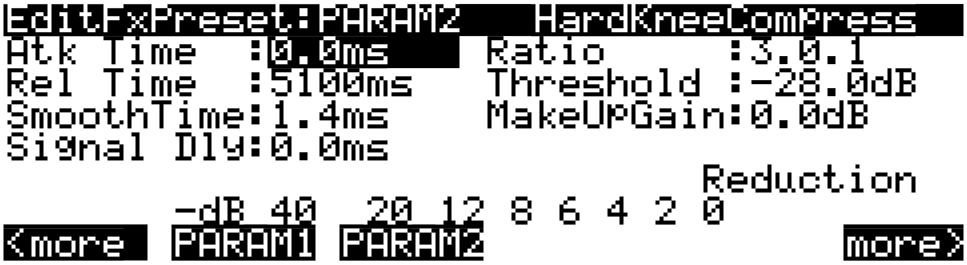
Metering

All Input and FXBUS pages have rudimentary signal meters (the arrows), as we've seen. On the Input pages, the arrow next to the input letter flashes when there is signal present at the input. On the FXBUS and AUXFX pages, the arrow next to the FXBus number flashes as long as there is signal being processed—in effect, it's an output meter. The arrows flash whenever the signal level exceeds 14 bits below full scale, which is -84 dB relative to the maximum level the KDFX can handle. Since typically the K2600 operates with about 20 dB of headroom, this translates to about -64 dB relative to normal operating level



```
EditStudio:AUXFX
Aux→103
```

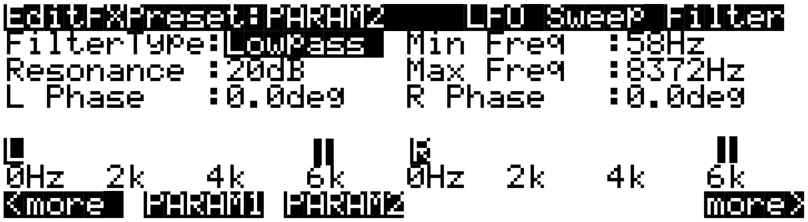
In addition, a number of KDFX's algorithms include a fast visual metering system for monitoring what the algorithms are doing. Many of these algorithms deal with dynamics control like compression and limiting. An example is Algorithm **950 HardKneeCompress**. Meters are most often found on the PARAM2 page of the FXPreset Editor:



```
EditFxPreset:PARAM2    HardKneeCompress
Atk Time   :0.0ms     Ratio      :3.0.1
Rel Time   :5100ms    Threshold :-28.0dB
SmoothTime:1.4ms      MakeUpGain:0.0dB
Signal Dly:0.0ms
                                Reduction
       -dB 40   20 12  8  6  4  2  0
<more   PARAM1  PARAM2                 more>
```

In this algorithm, the meter is showing the gain reduction that the compressor imposes on the signal coming into it.

Metering is also used for other purposes in some other algorithms, for example **902 LFO Sweep Filter**. In this algorithm, the meters show the current center frequency of the left and right filters as they sweep up and down the spectrum:



```
EditFXPreset:PARAM2     LFO Sweep Filter
FilterType:LowPass  Min Freq   :58Hz
Resonance :20dB     Max Freq   :8372Hz
L Phase   :0.0deg   R Phase    :0.0deg

L                   R
0Hz  2k   4k   6k   0Hz  2k   4k   6k
<more  PARAM1 PARAM2               more>
```

Meters use up a certain amount of KDFX's processing power, although less than a whole PAU, and so they are only used in algorithms where extra power is available. If an algorithm would need to increase its Size—say, from 2 PAUs to 3—with the addition of metering, then metering has not been included in that algorithm. This is to ensure that the maximum amount of processing power is available for actual signal processing.

Output Section

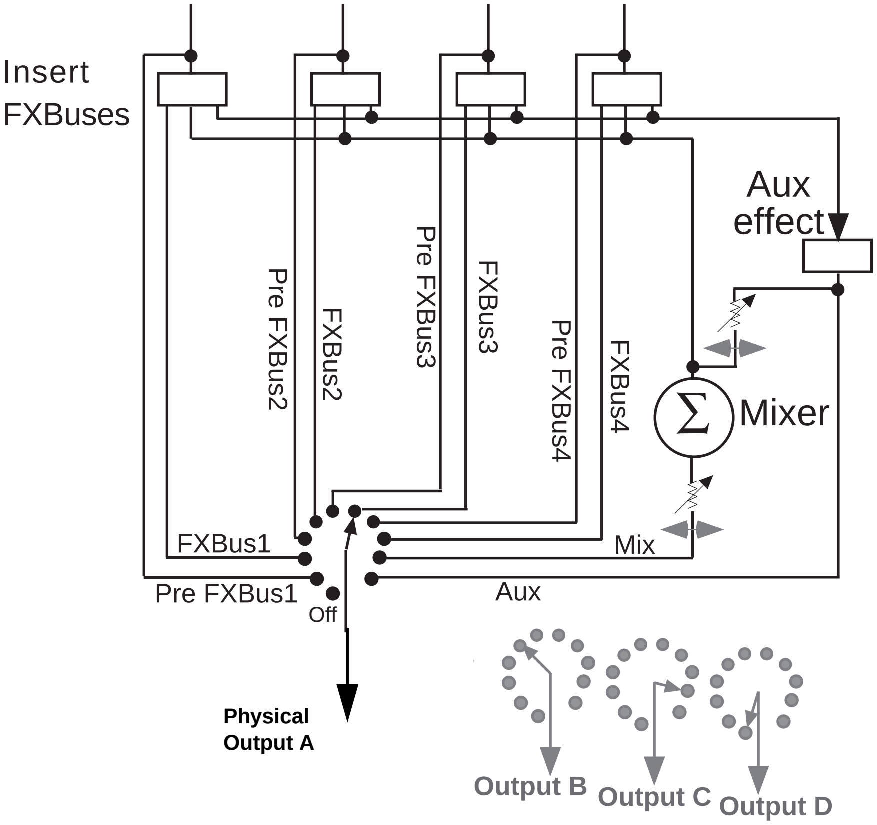


Figure 15-8 FXBuses and Physical Outputs

In the Output section, the various signal paths are routed to the K2600's physical outputs. Each of the four output pairs, A, B, C, and D, has a selector switch to determine which signal it will carry. All signals are stereo. The choices are:

Off No output.

PreFX*n* The signal coming *into* the FX preset on FXBus1, 2, 3, or 4. All of the EQs and pan/width/balance settings of the input modules that are assigned to FXBus*n* are active on this signal path, but the FX preset is not.

FXBus*n* The direct output, post-FX preset, from any of the four insert FXBuses.

Mix The signal from the Mix output, which can include the sum of any or all of the insert FXBuses, and/or the Aux bus. The Level and Balance parameters on this page control this output.

AuxFX The outputs of the Aux bus.

There are no restrictions on the settings. If you like, all four outputs can be carrying the same signal: PreFX1, for example.

Separate Analog and KDS Outputs

The K2600's separate analog outputs (four pairs, A–D) are wired in parallel and are "live" at all times. So are the eight digital outputs (also four stereo pairs) available with the digital I/O (KDS) upgrade option. The parameters on the OUTPUT page affect both the separate analog and KDS outputs.

Analog Mix Output

The K2600's analog Mix output combines the four output pairs into a single stereo analog pair. Each of the analog outputs, A–D, carries the signal from the KDFX bus specified for each output pair on the OUTPUT page of the Studio Editor. The signal at the Mix output always carries the summed signal of outputs A–D; connecting cables to Outputs A–D does not remove the corresponding portion of the signal from the Mix outputs.

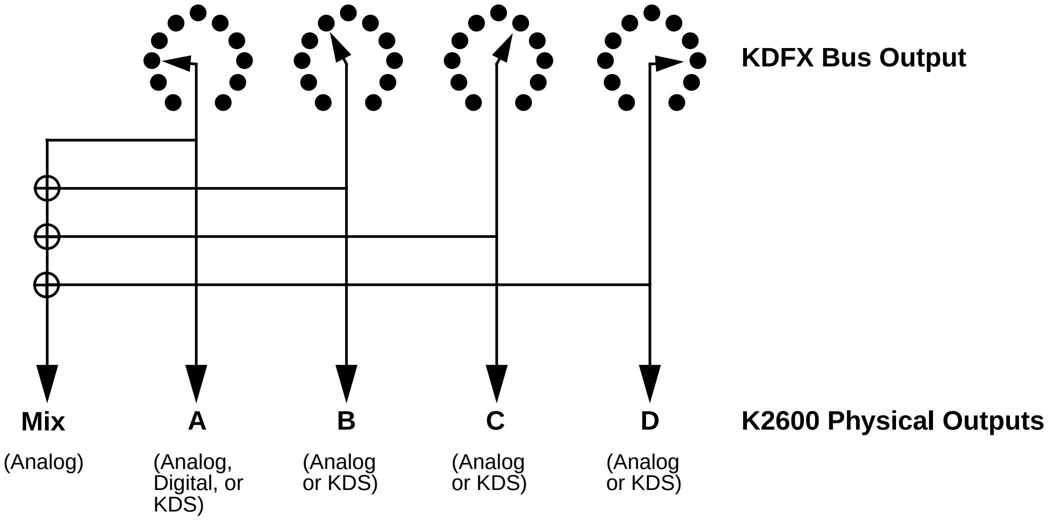


Figure 15-9 K2600 Analog, Digital, and KDS Outputs

Digital Output

For digital output, you'll need either the sampling option or the digital I/O (KDS) option. In either case, the digital output of the K2600 (a single stereo pair) is wired in parallel with output pair A. It gets whatever signal is routed to Output A on the OUTPUT page of the Studio Editor.

With the digital I/O option, you can use the XLR or optical outputs, or the separate KDS outputs (although this requires the Kurzweil DMTi digital interface for full functionality). With the sampling option, you can use the XLR or optical outputs.

The sampling option also enables you to sample the K2600's own output.

1. On the Sample-mode page, set the Input parameter to **Analog**, and the Src parameter to **Int**. This routes the signal at the K2600's Mix outputs to the analog sampling input, ignoring the external sampling inputs on the rear panel.

2. Set the Input parameter to **Digital**, then set the Out parameter, if available, to **A/D**. This causes the digital outputs to mirror the signal coming from the K2600's Mix outputs. (Set the Out parameter to **Dir** to mirror the Output A signal at the digital outs.)

***Note**: If you don't have the Digital I/O option, you won't see the Out parameter, which means that the digital output mirrors the K2600's Mix outputs.*

When the Input parameter is set to **Analog** and the Src parameter is set to **Ext**, the signal that appears at the digital output is the signal coming into the K2600's *external* analog inputs, and the K2600 is acting as a very fancy, very high-quality A/D converter.

Value of Out Parameter	Value of Src Parameter	Signal at Digital Output
Dir	N / A	KDFX Output A
A / D	Int	K2600 Mix Output
A / D	Ext	K2600 Analog Input

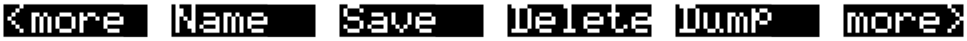
Table 15-1 Digital Output Switch Settings

Saving Studios and Other objects

Saving

Saving a studio or an FX preset is handled the same as any other K2600 object—see *Saving and Naming* on page 5-3 if you need help with this.

To save an object, from any of the editing pages within the object, press **<more** or **more>**, until you see the file-management pages:



Name lets you rename the object, using the standard K2600 naming window. **Save** lets you save the object to its present slot or to a different one, and also gives you the option of renaming it before saving it, or accessing the Object utilities—see Chapter 13.

Delete deletes the object from RAM. If it is in a slot where there is a ROM studio or FX preset, then the slot will revert to the ROM studio or FX preset.

Dump performs a MIDI System Exclusive dump of the object over the MIDI Out cable for external storage.

The Save option is automatically invoked if you have done any editing within an object and you press **Exit**.

Do I Always Have to Make a New Studio?

While it may sound as if every time you make a change in a studio parameter, you have to create a new studio to save it, this is not necessarily true. Studios do take up room in program RAM, and although the number of studios that can be maintained inside the K2600 is large, it is not unlimited. Besides, it can be very cumbersome keeping track of a large number of studios which have only small differences between them.

A way to avoid this problem is to use Program or Setup mode, and use FXMods—studio-parameter controls that live inside a program or setup—to make changes in the studio parameters. When you load the program or setup, the studio changes accordingly. This means you don’t have to save each variation in the studio as a separate studio; the variations live in the program or setup.

See *Real-time Control of KDFX* below for more information.

Disk Functions

Studios and FX presets are loaded from and saved to disk exactly the same as other K2600 objects. FX presets are dependent objects within studios, so when you save a studio, you are given the opportunity to save the dependent FX presets as well. (Algorithms are in ROM, so they don’t need to be saved.) Again, see Chapter 13 if you need help.

Real-time Control of KDFX

Studios can be static, but they don’t have to be. One of the most powerful features of KDFX is the ability to change any of the operating parameters in a studio in real time. Mix levels and pans, EQ values, effects parameters, and almost any other settings can be controlled dynamically and smoothly, giving KDFX a high degree of flexibility in performance and automation.

KDFX parameters can be controlled from the following

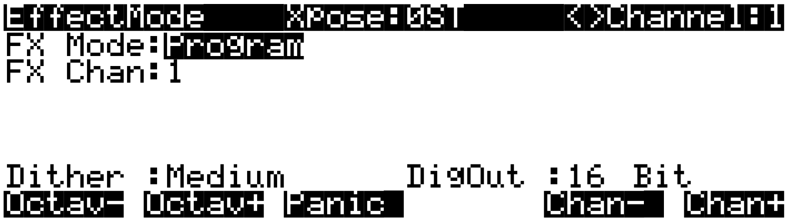
- The K2600’s sliders, wheels, ribbons, and pedals
- External MIDI sources like another controller or a sequencer
- Internal functions like LFOs, envelopes, and FUN generators

Real-time control of a studio is called effects modulation, and a link between a program or setup and a studio parameter is called an FXMod.

Linking to Programs and Setups

If you’re familiar with real-time control of the K2500’s original internal effects, you’ll be happy to know that real-time control of studios works the same way.

Real-time KDFX studio control normally originates within a program or setup. For the link between a studio and a program to work, the FX Mode parameter on the Effects-mode page must be set to **Program**. If you want to link a studio and a setup, the FX Mode parameter on the Effects-mode page must be set to **Setup**. Setting the FX Mode parameter on the Effects-mode page to **Auto** means that it will follow whatever mode the K2600 is in—Program or Setup—which can be particularly useful when you are switching between programs and setups in a Quick Access bank, or when you just don’t want to worry about which mode you’re in.



```
EffectMode     Xpose:0ST       <>Channel:1
FX Mode:Program
FX Chan:1



Dither :Medium       DigOut :16 Bit
Octav- Octav+ Panic         Chan-  Chan+
```

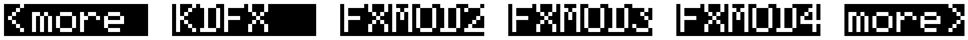
Notice that when FX Mode is set to **Program** or **Setup**, the studio is not shown, as it is when the FX Mode is set to **Master**. So you can't go into a studio and edit it from this page—you have to go through the program or setup, from its KDFX page.

Note: When FX Mode is set to **Auto**, then FX Mode follows the K2600's operating mode—it's either **Program** or **Setup**. When you are on *this* page, however, FX Mode is actually **Master**. But don't screw up your brain thinking about this—we'll get back to it later in this chapter (*KDFX in Auto Mode* on page 15-34).

For the sake of clarity, during most of this chapter we will talk about linking KDFX only with programs. The procedures for linking KDFX to a setup are essentially identical, except for the setting of the FX Mode.

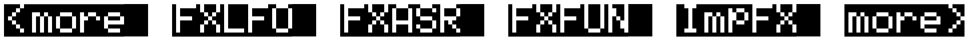
The KDFX Pages

If you look inside any program, you will see several pages that handle the studio and KDFX control assignments. Open up a program with the **Edit** button and press **more>** until you see this at the bottom of the display:



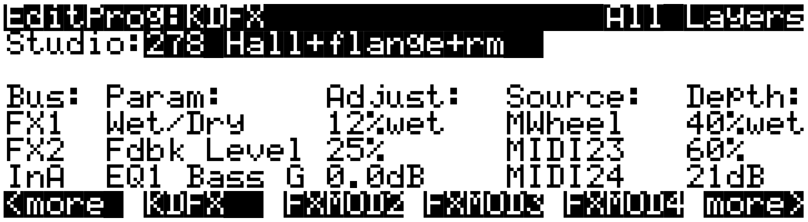
```
<more   KDFX    FXMOD2  FXMOD3  FXMOD4  more>
```

These are the FXMod setup pages, and in fact there are eight of them—press **more>** again to see the others.



```
<more   FXLFO   FXASR   FXFUN   ImpFX   more>
```

Go back to the first set (press **<more**), and press the **KDFX** soft button to look at the first of the FXMod setup pages:



```
EditProg:KDFX                     All Layers
Studio:278 Hall+flange+rm

Bus: Param:      Adjust:  Source:  Depth:
FX1  Wet/Dry     12%wet   MWheel   40%wet
FX2  Fdbk Level  25%      MIDI23   60%
InA  EQ1 Bass G  0.0dB    MIDI24   21dB
<more KDFX  FXMOD2 FXMOD3 FXMOD4 more>
```

The top line tells us we're in the KDFX section of the Program Editor. The second line shows the studio that is linked to this program. Any changes made on this page do not directly affect the studio, they affect only this program's *control* over the studio.

The last three lines (ignoring the Soft buttons) show us which FXMods are active in this program: one is controlling Wet/Dry mix on insert FXBus1, one is controlling feedback level on insert FXBus2, and one is controlling the Bass Gain on the first EQ of Input A.

Looking Into the Studio

You can go into this studio, to see what the parameters are doing—a good idea when you are setting up FXMods so you understand them in context—or to alter a fixed parameter. You can do this without leaving the Program Editor: highlight any parameter on this page and press **Edit**. When you are done with the studio, pressing **Exit** will bring you back to this page. If you have made any changes in the studio, you will be prompted to save the studio, and if you don't do so, the changes will be disregarded.

If you do save the studio, either in the same numbered location or a new one, the new studio will now be linked to this program. If you have changed the studio's number, then you'll also have to save the program when you leave the Program Editor, so that the program knows which studio to link to the next time you call it up.

Setting Up FXMods

The KDFX page allows three different FXMod control assignments to be made. More FXMod assignments are available on the FXMOD2, FXMOD3, and FXMOD4 pages, each of which has five setups. This gives a grand total of 18 studio parameters that can be under real-time control. Don't worry, you don't have to use them all.

Bus Assignments

The first column lets you choose which bus inside the studio you want this FXMod to control: Input A, B, C, or D (if any of the Inputs is set to Mono, you will get to choose individual channels, for example A/L and A/R); FXBus 1, 2, 3, or 4; the Aux bus; or the Mix bus.

Parameter Assignments

The second column chooses the specific parameter on the selected bus. Scrolling through the choices shows that this selection is context-sensitive: it shows only parameters that are being used in the current studio on the selected bus, so you can't make assignments to irrelevant or nonexistent parameters. It "knows" which EQs are active, and what modes they are set to; which Input sends are assigned and whether the sends are in Pan or Balance mode; what parameters are being used in the FX preset on the particular bus; etc. If you need to know more about how the studio is set up, you can go into it and look around, as explained above.

Here's an example: set the FX Studio to **201*RngMd/PFD/Plt** which we looked at on page 9-16. Set the bus on the first line to **InA**. Set the cursor under Param: and scroll the choices. They correspond exactly to the parameters available in the Input Editor page: level and frequency for the two EQs, and SendLvl, Pan, and Width for the two FX sends. Note that you *cannot* change the FXBus assignments on an input from here; you must do that within the studio itself.

Change the bus to **FX1**, and all of the parameters from the FX preset on FXBus1 are available for selection, including Mod Mode and the various settings for the preset's internal oscillators, as well as the bus's output controls: Mix Level, Mix Balance, Aux Level, and Aux Balance. Change the bus to **AuxFX**, and the Aux bus's parameters are available: levels, delays, room types, etc.

Setting the bus to **Mix** makes available the Level and Balance controls from that page.

What Can’t Be Controlled

You cannot change any parameters through KDFX that would involve a major reconfiguration of the studio:

- Bus assignments on the Input pages
- Selecting FX presets on the FXBus pages
- Allocation on the FXBus pages
- Selecting algorithms within the FX presets
- Bus assignments on the Output page

If you want to be able to change any of these in real time, you will have to create a new studio, link it to a different program or setup, and then call it up using a Program Change command.

In addition, there are a few parameters that can cause serious glitching if they are changed in real time. The most common of these are the “Room Type” settings in reverb algorithms. While there is nothing in the software to prevent you from assigning an FXMod to Room Type, you need to be aware of the potential consequences. See *Static FXMods* on page 15-27.

Adjust

The Adjust setting is the starting value of the selected parameter when it is under KDFX control, similar to the entry value of a controller in a setup. This might very well be different from the value of the parameter when the studio is *not* under FXMod control, so don’t get confused. If you are in Program mode, and this is the current program, the Adjust value takes precedence over the studio’s fixed value, and it’s the Adjust value that will be called up when you call up the program.

Source

The Source parameter determines which real-time control—internal, MIDI, etc.—is going to affect the selected studio parameter. As with all K2600 real-time controls, the range of control sources is very large:

- OFF (the parameter is not affected by any source and stays at its Adjust value)
- ON (the parameter is set to the maximum value determined by adding the Adjust and Depth values)
- MIDI Continuous Controllers 1-95 (*see Note*)
- Channel State
- Pressure
- Pitch Wheel
- The usual list of controllers, as described in Chapter 4 of the *Musician’s Reference*: ASRs, FUNctions, Clocks, LFOs, Internal Controls, Random Generators, etc.

***Note**: Under some circumstances, particularly when the K2600 is in Setup mode, there are certain restrictions on which MIDI sources you can use.*

Dedicated FXMod Control Sources

There are a few control sources that apply exclusively to FXMods:

- FXLFO1, FXLFO1ph, FXLFO2, and FXLFO2ph—two LFOs and their phases.
- FXASR1 and FXASR2— two three-stage (Attack/Sustain/Release) envelopes with selectable triggers and Normal, Hold, and Repeat modes.
- FXFUN1, FXFUN2, FXFUN3, and FXFUN4—Functions. Yes, more Fun with KDFX!

The ASRs, FUNs, and LFOs work exactly the same way they do in any other part of a program, except these are extra control sources for use *only* with FXMods, and are not available for other program functions. They are global for all of KDFX—you can't apply these controls to just one FXBus.

The parameter values for these controls are saved with the program. As we saw earlier, you get to the pages for the FXMod control sources by pressing the **more>** soft button until you see these soft buttons:

In the Program Editor, you can also get to these pages directly from the KDFX page or one of the FXMOD pages: select one of those parameters as a source, and then press **Edit**.

Tempo-based Parameters

There are several different ways KDFX can respond to tempo information, from the internal sequencer or an external one. These are discussed later in this chapter (see *Tempo-based Control of KDFX* on page 15-32).

Depth

The Depth parameter lets you specify a range of change in values that the real-time control will make, using the Adjust value as a minimum or starting point. This range can be positive or negative, and the values are displayed in the context of the studio parameter: seconds, dB, %, Hz, cents, etc.

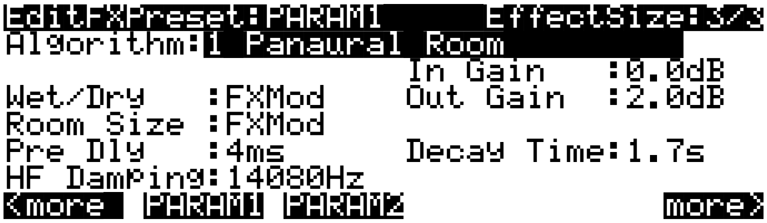
At the maximum setting of the Source (for example, Mod Wheel all the way up), the value of the parameter = Adjust + Depth. So if the parameter is Out Gain, the Adjust is 1.0 dB, the Source is Mod Wheel, and the Depth is 4.0 dB, then at the Mod Wheel's highest point, the output gain will be 5.0 dB.

For Source values less than maximum, the formula is: parameter = Adjust + (Depth x Source), where the Source is considered to be varying between 0 and 1 (or in some cases, such as Pitch Wheel, between -1 and +1). So using the same example, when the Mod Wheel is halfway up (MIDI value 64), the gain is 1.0 + (4.0 x 1/2) = 3.0 dB, and when it is all the way down (MIDI value 0), the gain is 1.0 + (4.0 x 0) = 1.0 dB.

The formula works the same way for negative Depth values. Given the same example, but with a Depth of -4.0dB, at the Mod Wheel's minimum point, the gain will be 1.0 dB; at its halfway point it will be -1.0 dB (1.0 + (-4.0 x 1/2)); and at its maximum point it will be -3.0dB.

Showing Who's in Control

When you are in Program or Setup mode and you look inside the current studio or its FX presets, any parameters that are under FXMod control will not display numerical values, but instead will say **FXMod**. Don’t touch any of those parameters for now—we’ll explain why in a moment.



```
EditFXPreset:PARAM1       EffectSize:3/3
Algorithm:1 Panaural Room
                          In Gain   :0.0dB
Wet/Dry   :FXMod          Out Gain  :2.0dB
Room Size :FXMod
Pre Dly   :4ms            Decay Time:1.7s
HF Damping:14080Hz
<more    PARAM1  PARAM2              more>
```

Note that if you look at the studio when FX Mode is in Master, the FXMods will not be displayed, because the FXMods are not in effect! So don’t get confused.

Breaking the Links

When FX Mode is **Program** (or **Setup**) and you are in the Studio Editor, if you change a parameter that has been assigned an FXMod, it breaks the link—the parameter is no longer under FXMod control, but is now a static value. If you now save the studio (either in the same location or a different one), the link stays broken. However, if you leave the studio *without* saving it, and go back to the FXMod pages of the Program Editor, the link automatically reestablishes itself.

If you change the FX preset on an FXBus, the FXMod links to the preset on that bus may or may not break. If the preset you’re calling up uses *the same algorithm* as the preset you’re leaving, the links will stay intact. If it uses a different algorithm—even a similar one—the links will break. On the other hand, if you are inside a preset, and you change its algorithm, that breaks the links. Again, the links will be reestablished if you leave the preset without saving it.

Links to the Aux and Mix parameters on an FXBus do not get interrupted when you change the FXpreset on that bus.

The same rule applies when you are on the KDFX page in a program or setup and you change the studio. Links to the Input and Output pages and the Level parameters on the FXBus pages will be retained, but links to processing parameters will be retained on a given FXBus only if the algorithm inside the preset on that bus isn’t changed.

What About Bus Overrides?

Bus overrides are transparent, as far as KDFX is concerned. Any FXMods that involve a parameter inside an FX preset are set up directly between the program and the FX preset itself. If the parameter happens also to be assigned a bus override, it doesn’t make a difference to KDFX—the program will control the parameter as if the bus override wasn’t there.

However, bus overrides are still active *within a studio*, and changing a bus override value in a studio will, as usual, change its associated FX preset parameter. If there happens to be an FXMod *also* controlling that parameter, *the FXMod link will be broken*, just as if you reset the parameter from inside the FX preset.

Using FXMods So You Don’t Have to Change Studios

We’ve mentioned that FXMods are a good way to get around the need to create a new studio every time you want to make a small change in a studio’s parameters. The same studio can be used for many different purposes if you use FXMods to control it. FXMods can set the gain of the signals going to the various FXBuses, set panning and output levels of the signals from the buses, configure the mix going to the Aux bus, and even turn FXBuses on or off, as well as set processing parameters.

Since any parameter except the ones that reconfigure a studio (see *What Can't Be Controlled* on page 15-24) can be under FXMod control, the amount of variation between the various program- or setup-controlled versions of a single studio can be very great.

Static FXMods

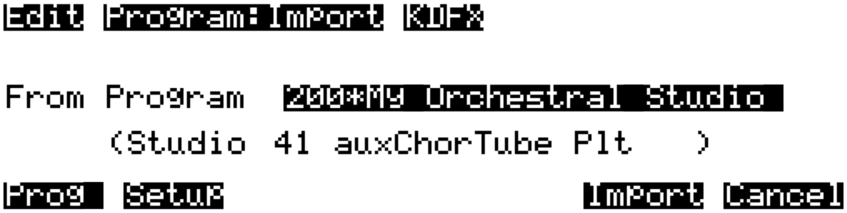
Static FXMods, that is, FXMods that are just going to be used to "set and forget" studio parameters when a program or studio is called up (as opposed to dynamically controlling them), are easy to set up. On the FXMOD page, select the bus and parameter you want to control, set the Adjust value to the parameter value you want, and set Source to **OFF**. When you are done, save the program or setup normally. Now whenever you select the program or setup, the parameters in the FXMods will be immediately reset to the values you've specified.

Room Type and other potentially glitch-producing parameters work much better as static FXMods than they do under dynamic control. If you need to change a Room Type in a reverb when you change a program, you can do so with a static FXMod without hearing horrible glitches, as long as you make sure there is no signal passing through the reverb at the moment of the program change.

Importing Studios From Another Program or Setup

If you have a killer studio with FXMods set up in one program (or setup) and you want to use the same studio and FXMods in a different program, you don't have to rebuild the studio by hand: you can copy studios between programs or setups in one operation.

Press the **ImpFX** soft button, and you'll see this:



```
Edit Program:Import KDFX

From Program  200*My Orchestral Studio
     (Studio 41 auxChorTube Plt   )
Prog  Setup                  Import Cancel
```

Use the soft buttons at the left to select either a program or a setup as the source for the studio and FXMods you are importing. Select the number of the program or setup you want to import from. As you change the program or setup, the studio associated with each program or setup will display in parentheses below the program or setup.

The studio will also kick in, so you can hear it affecting the program you're listening to.

When you've chosen a program or setup to import a studio from, press the **Import** soft button. "KDFX from…imported" will flash on the screen, and you will be returned to the Program Editor of the program or setup you are working on.

Studios from setups can be imported into programs without restriction, and vice versa.

FXMods on Imported Studios

If you have imported a studio into a program, the FXMods that were in the program you've imported *into* are now all gone. They are replaced with the FXMods associated with the program you imported *from*. It doesn't matter if there were more FXMods in the old program than in the new—even if the new program has *zero* FXMods, the old ones are all erased.

Using KDFX Live From the K2600 Keyboard

The sliders and wheels on the K2600 keyboard, and the pedals connected to the keyboard, can be extremely useful with KDFX, if you assign FXMods to the K2600's controllers. The ribbon, for example, can be used in a "Pitcher" FX preset to change the pitch of the signal, while sliders and pedals can be used to control reverb time, flanger feedback, EQ, or any of the myriad parameters available in KDFX.

KDFX in Setup Mode

To use KDFX with a setup, make sure that FX Mode (on the Effect-mode page) is set to **Setup** or **Auto**.

Like programs, every setup has a set of FXMod routings associated with it. These are identical to the routings available in programs: there are four FXMod pages, three dedicated FX function pages, and an Import FX page. The procedure for setting up the routings is the same as in a program. KDFX settings can be imported into a setup from either another setup or a program.

When KDFX is under setup control, any FXMods in the programs *within* the setup are ignored. If you have a program that contains a studio and FXMods that you would like to use while playing a setup, import the KDFX from the program into the current setup.

MIDI Control and Setup Mode

You'll notice that when you set FX Mode to **Setup**, FX Chan goes to **None**, and can't be changed. The MIDI receive channel for controlling KDFX is not determined here, it's determined in the setup itself: it's the channel used by Zone 1 of the setup. Any incoming MIDI commands on other channels, while they may play *sounds* in the setup, will not affect KDFX parameters.

If for some reason you don't want KDFX to be in Setup mode while you are playing a setup—for example, if you want it to be under control of an external MIDI device and not change studios when you change setups—you can set the FX Mode to **Program**. In this case the *program that is in the zone assigned to the FX Channel*, (which may or may not be part of the setup) will control KDFX.

So for example, say a setup has three zones, which are assigned to channels 2, 4, and 6. If FX Chan is set to **6**, then the studio (and FXMods) assigned to the program in Zone 3 will be active. If none of the zones' channel assignments match the FX Channel—say it's set to **16**—then nothing you do on the K2600 will control the studio and FXMods. However, an external MIDI source, like a sequencer, sending on channel 16 *will* control the studio and FXMods, and the program that is on channel 16 will determine what they are.

This technique can be useful when you want to have a sequencer control KDFX, while you are also playing along on the K2600 keyboard in Setup mode.

KDFX in Program Mode

If you are playing the K2600 in Program mode, then you must make sure that the current channel of the K2600 (at the upper right of the Program-mode page) agrees with the FX Channel (on the Effects-mode page). Otherwise the local keyboard commands, while they will control the other current program parameters, will not address KDFX. If FX Chan is set to **Current** then you don’t have to worry about this.

Performance Modes and Effects Control

Here’s how to tell what’s controlling what, depending on the various modes. See *FX Channel* below for information about the MIDI channel for external control of KDFX. Note that you can use SysEx commands to control KDFX regardless of the setting for FX Mode.

K2600 Mode	FX Mode Value	What Sets Physical Control Assignments (Sliders, Wheels, etc.)	What Controls KDFX	MIDI Channel for Control of KDFX
Program	Program or Auto	Control setup	Program’s FXMods	FX Chan
Setup	Setup or Auto	Setup	Setup’s FXMods	Channel used by Zone 1
Setup	Program	Setup	FXMods of program on channel assigned as FXChan	FX Chan
Program	Setup	Control setup	The last setup that was selected in Setup mode	Channel used by Zone 1 of the last setup
Program	Master	Control setup	Nothing	None
Setup	Master	Setup	Nothing	None

Table 15-2 Modes, Control Assignments, and Effects Control

Using KDFX With a Sequencer

If you use an external MIDI sequencer, you are probably thinking about how powerful it will be putting KDFX under sequencer control. Certainly being able to record, edit, and automate a studio’s parameters as part of a MIDI sequence is one of the most attractive aspects of KDFX.

FX Channel

Any program on any channel can be the one that controls KDFX. On the Effects-mode page, set FX Mode to **Program**, select the FX Channel you want to control KDFX with, and put your KDFX-controlling program on that MIDI channel. Now any MIDI commands coming from the sequencer on that MIDI channel will be sent to KDFX.

Dedicating a Program and Channel

Perhaps the most efficient and least confusing way to do this is to have a dedicated program that controls *only* KDFX, on a channel that is otherwise not being used to play music. Many K2500 and K2000 users know this trick for automating the old internal effects, but it becomes even more important given the complex nature of KDFX. It requires sacrificing a MIDI channel, but few users should have a problem with that.

Again, go to the Effects-mode page and, keeping the FX Mode set to **Program**, set the channel you’re going to use as an FX Chan—in this case, **15** is often a good choice. Go to Program Mode and select Program **199 Default Program**. Press **Edit**, and then **KEYMAP**, and set the Keymap parameter to a value of **0 None**. This program will now make no sound in response to MIDI notes, and use up none of the K2600's polyphony. Use the **<more>** buttons to get to the KDFX page, and start setting up your studio and FXMods, or, if you have another program already set up with the studio and FXMods you want, use Import FX to bring that studio and FXMods into the current program.

Now save the program to a new location, giving it a name like **Studio Controller1**. You can now use this studio with your sequencer: start by calling up its bank and program number, and then put appropriate MIDI commands into the sequencer for controlling the studio's parameters.

If the MIDI sources for the FXMods are also K2600 on-board controllers, then you can record your parameter changes from the K2600 into the sequencer.

There's a step-by-step discussion of this procedure, beginning on page 12-21.

Changing Studios With a Sequencer

If you need more than one studio available in a piece, or you want a selection of studios to use in different pieces, simply create new programs the same way, with different studios specified on their KDFX pages. To switch from one to the other, send an appropriate Program Change command on channel 16 from your sequencer.

Setting up static FXMods in different programs, all of which address the same studio, as we saw earlier, is a good way to make one-shot changes in KDFX parameters without having to construct a bunch of different studios.

Also as we saw earlier, this technique can be useful even when you are playing the K2600 in Setup mode along with a sequencer. If FX Mode is **Program**, and the FX channel is **16**, the sequencer can control the FXMods independently of which setups are in use on the keyboard.

Preventing Glitches When Changing Studios

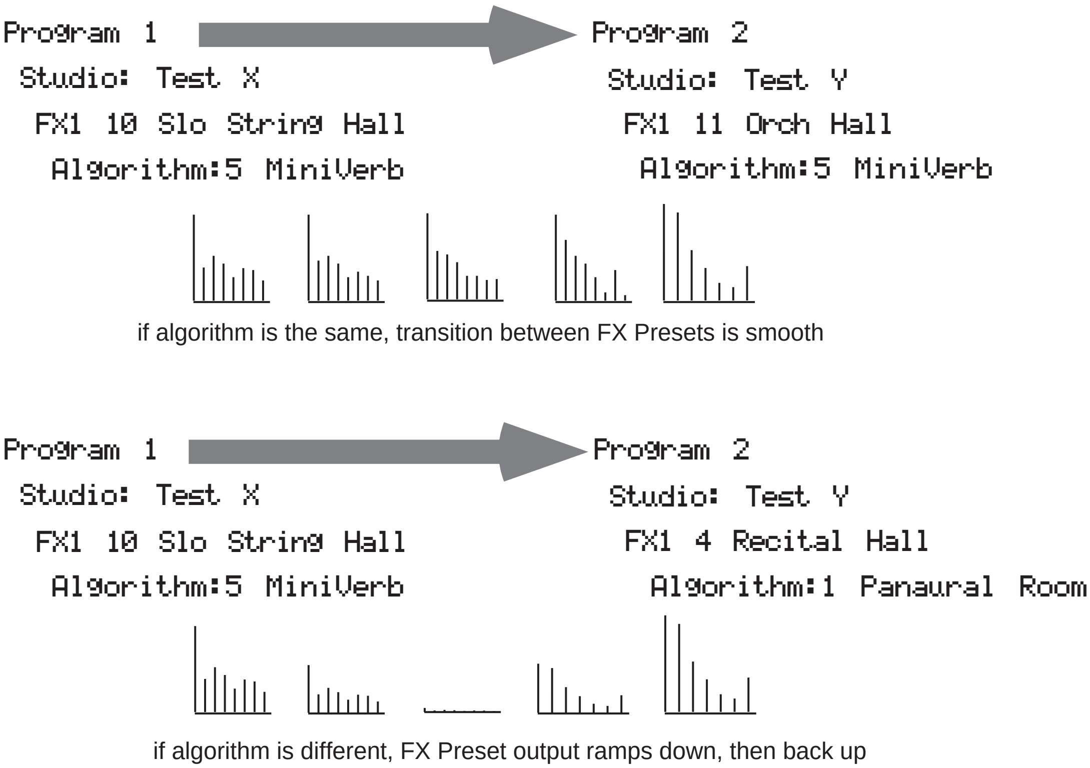
As with any digital effects unit, you need to take some care when you are sending KDFX real-time commands that radically alter the nature of the processing it is doing. The trickiest situations will occur when you are changing studios, and calling up new FX presets, algorithms, and/or signal routings.

Under the best of circumstances, the transition between two studios can be seamless, and the effects in one will "morph" into the effects of the other. Under the worst of circumstances, there will be a momentary "hole" in the sound, as the effects from the first studio are cleared out and the effects of the second studio build up.

The chances of a smooth transition between studios will be highest if the algorithms on the FXBuses in the two studios are the same. (In some reverb algorithms, the Room Type parameter should also be the same.) For example, if FXBus1 in Studio X uses an FX preset based on the reverb in algorithm 4, and FXBus1 in Studio Y uses a different FX preset based on the same reverb algorithm 4, then when you switch from Studio X to Studio Y, the signals going through FXBus1 should experience a smooth reverb change.

However, if the FX preset for FXBus1 in Studio Y uses a *different* algorithm, say a multitap delay in Algorithm 35, then at the moment the studio changes, the reverb effect will ramp down quickly, there will be a very brief point at which the signal will pass dry, and then the multitap will quickly ramp up. This ramping will take place any time the two algorithms (or Room Types) are different—even if on the surface they seem very similar, like a Room and a Hall.

Even if the algorithms are the same, however, there is a chance that the transition will not be completely smooth. Some parameters cause glitches if they are changed in real time even within a studio—delay lengths, for example—and so if a parameter like that changes when you *switch* studios, the software will clear it out as the studio changes, and a small hole will appear. In some cases, instead of a hole, you will hear a momentary pitch shift as the studio changes. A bit of experimentation will help you determine how to achieve acceptable transitions between studios.



if algorithm is the same, transition between FX Presets is smooth

if algorithm is different, FX Preset output ramps down, then back up

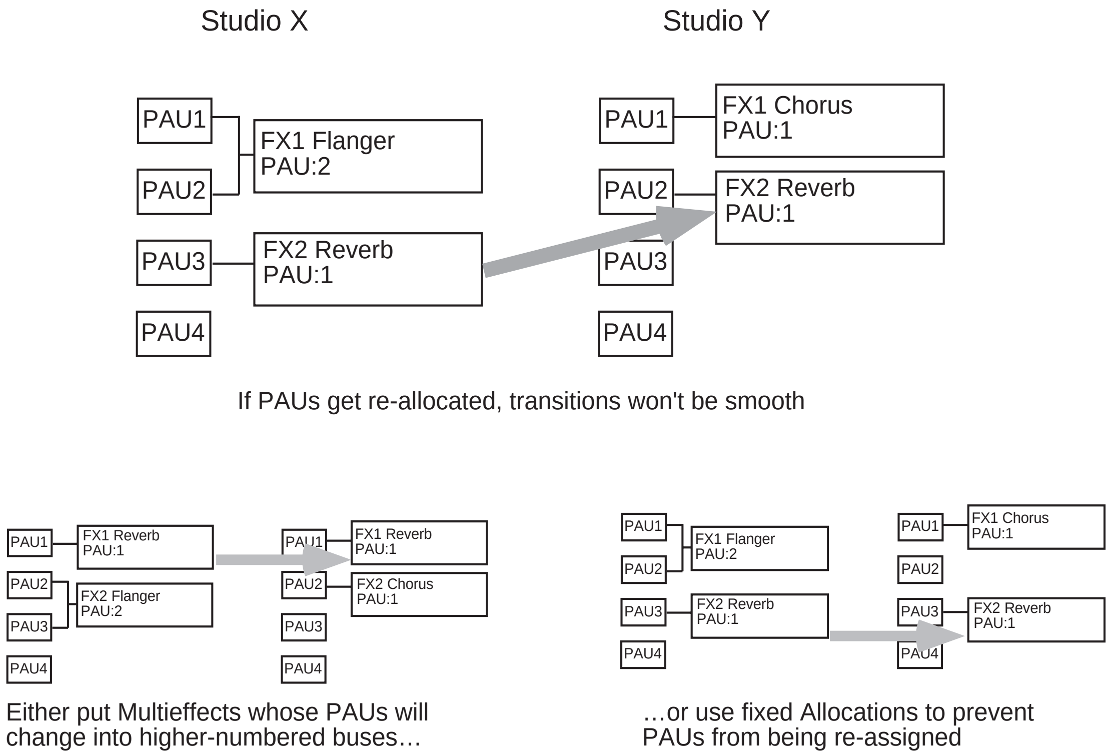
Figure 15-10 Effect of FX Algorithm on Transition Between FX Presets

Changing PAU Allocations

There are other circumstances under which using the same algorithm will not ensure a smooth transition. If for any reason the PAUs have to be reallocated when you move from one studio to another, then even if the algorithms on a given FXBus are the same in both studios, there will be a ramp-down/ramp-up. This can happen when there is a change in the number of PAUs being used on any FXBus that has a *lower number* than the FXBus you’re concerned with. That’s because reallocating PAUs on the fly forces some of the processors to reconfigure themselves, inasmuch as they are now being called upon to do different functions.

Here’s an example: say Studio X’s FXBus1 uses a Flanger that requires 2 PAUs, while FXBus2 has a small reverb that uses 1 PAU. In Studio Y, FXBus1 has a Chorus that uses only 1 PAU, while FXBus2 uses the same small reverb as Studio X. When you switch from Studio X to Studio Y, one of the PAUs that was previously being used for FXBus1 is now being used for FXBus2—it’s been reallocated. Since it’s not the *same* PAU that’s handling the reverb, the transition isn’t going to be smooth—even though the algorithm hasn’t changed.

There are a couple of ways around this. First is to set up your studios so that any transitions between them that are going to force changes in PAU allocations occur in the *higher-numbered* FXBuses—if the situation just described were reversed, and the reverb was in FXBus1, there would be no problem, because that same PAU would be used in FXBus1 in both studios. A second method is to *manually* allocate the PAUs in the lower-numbered FXBuses, rather than using Auto allocation. Using the same example, if Studio Y had 2 PAUs hard-assigned to FXBus1, even though the FX preset is only using one of the PAUs, then that second PAU would not get reassigned to the reverb, and the reverb's transition would be smooth.



If PAUs get re-allocated, transitions won't be smooth

Either put Multieffects whose PAUs will change into higher-numbered buses…

…or use fixed Allocations to prevent PAUs from being re-assigned

Figure 15-11 Avoiding Transition Problems When PAUs Get Reallocated

Tempo-based Control of KDFX

Many parameters within KDFX can be set up to respond to musical tempo. The tempo information can come from a number of sources: it can be fixed within a studio, it can come from the K2600's internal sequencer, or it can come from an external MIDI sequencer that has been configured to send MIDI Timing Clocks to the K2600.

The procedures for using tempo from the internal sequencer and from an external sequencer are essentially the same, with one crucial difference in one setting: in Song mode, on the MISC page. The Clock parameter must be set to **Int** when using the internal sequencer and to **Ext** when using an external sequencer.

Note that MIDI Timing Clocks (otherwise known as MIDI Sync) is not the same as MIDI Time Code (or MTC). MIDI Sync, which is used for syncing sequencers together, changes its speed in response to tempo changes; MIDI Time Code, which is used for syncing a sequencer to an audio or video tape or disk system, does not. KDFX (and the K2600 in general) responds to MIDI Sync, but not to MIDI Time Code.

Tempo Within an FX Preset

Some algorithms have a Tempo parameter, which allows certain time-based functions, such as LFOs, envelopes, or delay lengths, to be expressed in beats per minute (BPM). A second parameter immediately below Tempo, Period, multiplies or divides the tempo. The Period is expressed in beats (bts) and ranges from 1/24 (the actual time is 24 times faster than the Tempo) to 32 (the time is 32 times slower than the Tempo) in a sliding scale based on 24ths of a beat. So for example, if an LFO has a Tempo setting of 120 BPM and a Period of 12/24 bts, the LFO will oscillate 240 times per minute, or 4 Hz.

Variable Tempos Using System

You can also have these parameters respond to tempo in real time, by setting the Tempo parameter to **System**, which is set by scrolling below 1 BPM (or pressing **0** and **Enter** on the alphanumeric buttonpad). Now instead of Tempo being a fixed value, it will vary according to the system clock—which, depending on the setting of the Clock parameter on the Song mode's MISC page as described above, will either be the internal sequencer's clock, or tempo coming from an external source of MIDI Clock. The System Tempo will be multiplied or divided by the Rate or Beats setting, to determine the parameter's actual speed, just as if it were a fixed value.

If the Clock setting is **Int**, then the current tempo of the internal sequencer (as shown on the Song Mode's MAIN page) will be in effect regardless of whether the sequencer is running.

If the Clock setting is **Ext**, the external sequencer (assuming it is sending MIDI Clock) controls the Tempo parameter. If *no* MIDI Clock is being received by the K2600, the tempo will not go to **0**; instead, it will retain the last valid value it saw—either the tempo of the internal sequencer at the time when you switched the Clock to **Ext**, or the last tempo sent by an external sequencer before it stopped.

Tempo as an FXMod Source

KDFX algorithms that don't have a Tempo parameter can also respond to tempo information, using FXMods.

In the Source list, there is a value called Tempo, located at number 55. This source reads the current tempo (fixed or variable, internal or external) and turns it into a numerical value between **0** and **1**, which is then applied to the parameter under the control of the FXMod. The "virtual" range of this parameter is 3.75 to 240 BPM—that is, a tempo of 3.75 BPM (or less) will set the parameter value to **0**, while a tempo of 240 BPM (or greater) will set it to **1**. Of course, most parameters don't actually vary between **0** and **1**, so you have to scale the real values appropriately.

This can get pretty confusing, but there is a relatively simple method to follow when using this feature: set the Adjust value of an FXMod to the value of the parameter that you want when the tempo is 3.75 BPM, and set the Depth value so that the sum Adjust+Depth equals the value that you want when the tempo is 240 BPM.

For example, let's look at an algorithm that has an LFO Rate parameter in Hz. Let's say we want to the LFO speed to be twice as fast as the tempo. Here are the values for the FXMod:

Parameter **LFO Rate**

Adjust The slowest vibrato we want is (3.75 BPM/60) x 2, or 0.125 Hz. **0.13 Hz** is the nearest value available.

Source **Tempo**

Depth The fastest vibrato we want is (240BPM/60) x 2, or 8.00 Hz, so this parameter is 8.00–0.13, or 7.87 Hz; the nearest value available is **7.80 Hz**.

The Tempo value exists in Control Source lists throughout the K2600's operating system, not just in FXMod pages, so you can use it for controlling just about any function, including pitches of sample loops, envelope lengths, filters, volume, and more.

Tap Tempo

A real-time tap on a footswitch or button can also be used to control the tempo of a parameter in KDFX. Like Tempo, Tap Tempo is not KDFX-specific, but can be used throughout the K2600's operating system, including the sequencer.

Tap Tempo is on the Control Source list for any controller in a setup or control setup, located at number 39. Tapping or pressing on a controller that is set to **Tap Tempo** sends tempo information, based on the average time between taps, to the Internal Clock.

It takes four taps to determine the tempo—if you tap fewer than four times, no information is sent. If you tap more than four times, the average of *all* of your taps is continuously calculated and sent to the Clock. Later taps are weighted more heavily than earlier ones, so that if you change the rate at which you tap, the tempo information being sent will change accordingly, but there is a "flywheel" effect, and the Clock's tempo changes will lag somewhat behind the changes in your tapping rate.

In order for Tap Tempo to have any effect, the Clock parameter on the Song Mode:MISC page must be set to **Internal**.

KDFX in Master Mode

So far we've been telling you that setting FX Mode to **Master** on the Effects page disables real-time control over KDFX, but this is not strictly true. In that mode, KDFX will respond to specific system-exclusive (Sysex) commands. If you are a serious Sysex jockey, you can take advantage of this by using Sysex commands.

Sysex, in fact, can be used to control KDFX *regardless* of the FX Mode setting, so if you need to tweak certain parameters in real time without setting up an FXMod to do it, you can use Sysex and also use the FXMods in a program or setup.

See Appendix B in the *Musician's Reference* for details about using SysEx to control KDFX.

KDFX in Auto Mode

When the FX Mode is set to **Auto**, then control of KDFX changes according to where you take the K2600: either Program mode or Setup mode. But when you are on the Effects-mode page itself, you are actually in *Master* mode. Therefore, you can go into the studio whose name appears on the page and tweak its parameters, but there won't be any FXMods. You can change a parameter that has an FXMod link through a program (or setup), but, unlike when you are going into the studio through a program, that link will *not* be disturbed.

When you leave the studio, and then go into Program or Setup mode and call up a program or setup that links to that studio, any changes you have just made in the studio will show up, and any FXMods previously in place will also still be there.

From Auto Mode to Song Mode

When you are on the Effects-mode page and if FX Mode is set to **Auto**, and then you go into Song Mode, something slightly different happens: KDFX is now under control of the program that is on the effect channel within the song—from Song mode press **Edit** and look at the EffectChan parameter.

The effect channel is saved with the song, so make sure that it is set so that the program that you want to be in control of KDFX is active.

Using the Algorithms

This section will discuss the processing algorithms found in KDFX: what their parameters mean, and how to use them. It is designed to be read through so that you can get a general understanding of the algorithms. You can find a more complete reference, with every algorithm and the meaning and range of every parameter, arranged in the order they appear on the screen, in the *Musician's Reference*.

But first, a word about how the KDFX studios in ROM are organized.

Conventional Studio Structure

Many of the studios provided in ROM follow an overall organizational plan, which uses KDFX's resources efficiently and clearly. While by no means are you required to follow this structure when creating your own studios, it's a good idea to become familiar with it, and see how Kurzweil's own engineers have approached the complex issue of studio organization. And certainly it can serve as a good starting point when you start creating your own studios.

The organization of the ROM studios follows the following guidelines Not all studios are set up this way, but most of them are.

- ROM programs and setups are assigned to an output (KDFX-A through KDFX-D on the Program:Output or Setup:CH/PRG page) based on the type of effects processing that would most likely be appropriate for that sound.
- All inputs go to their respective FXBuses (Input A to FXBus 1, Input B to FXBus 2, etc.), and *only* to their respective FXBuses—the Lvl parameter for the second FXBus on each Input page is turned Off.
- FXBus1 contains a relatively simple reverb with a low Size requirement.
- FXBus2 contains an effect that does not increase the "length" of the sound (that is, no reverb or delay). Possibilities include chorus, flange, phaser, distortion, shaper, pitcher, enhancer, EQ, or EQ morpher.
- FXBus3 contains effects that take up lots of time, such as delays, delays with reverb, or other lead sounds.
- FXBus4 is dry, since the first three FXBuses have probably used up all the PAUs.
- The Aux bus contains a larger reverb (Size: 2 or 3), a compressor, or a graphic EQ. It can often be used instead of an FXBus reverb, such as the one on FXBus1. If you use it in this way (set the Aux Lvl on FXBus1 to 0dB or higher), it frees up FXBus1 for use as an Enhancer, Stereo Image, Flanger, etc.

General Parameters

Some parameters show up in all algorithms, and we'll deal with those first.

Wet/Dry balances the levels of the processed and unprocessed signals passing through the FXpreset. The range is 0% wet (the signal is unprocessed) through 100% wet (no dry signal is present). A setting of 50% wet means the dry and processed signals are equal in level. In some algorithms, separate Wet/Dry parameters are provided for the Left and Right input channels.

In Gain sets the level of the signal coming into the FXpreset from the Input page. As with most Gain controls in KDFX, the range is -79.0 dB to 24.0 dB, and there is an **Off** position. **0.0 dB** is unity gain.

Out Gain sets the level of the signal after it passes through the FXpreset. From here the signal goes directly to the Output page, if one of the Outputs is set to FXBus*n*. The level can be further changed before it goes to the Mix and/or Aux buses.

In/Out enables or disables the effect. It's like a Wet/Dry parameter with only two positions: 100% (In) and 0% (Out).

HF Damping is the cutoff (-3 dB) frequency of a 6dB/octave lowpass filter that's inserted before the processor. In the case of processors where multiple iterations of the signal are heard, such as in a delay, each iteration of the signal will pass through the filter, and will therefore be duller.

XCouple (Cross Couple). In stereo effects, this controls how much of any signal that is being fed back goes to the channel opposite to the one where it first appeared. At 100%, all feedback from signals at the left input goes to the right channel, and vice versa, causing a "spreading" or in the case of delay lines, a ping-pong effect. At 0%, fed-back signals stay with the channel they came in on.

A->B cfg (configuration). In combination algorithms that contain two components, and whose name uses **<>**, as in **713 Flange<>Shaper**, the order in which the signal passes through the two components can be changed. For example, this algorithm can be configured so the signal passes through the reverb first and then the compressor, or through the compressor first and then the reverb. The A->B cfg parameter determines the configuration, and its value is context-sensitive—in this example, the choices would be **Rvb->Cmp** and **Cmp->Rvb**.

A/Dry->B is also found in many combination algorithms, and controls the amount of signal that will pass dry (unprocessed) through the first component into the second component. Different combination algorithms use different variations on this parameter, depending on the context. The range is 0 to 100%.

Reverbs

Room Type changes the configuration of the algorithm to simulate a wide array of room types and sizes including booths, small rooms, chambers, halls and large spaces can be selected. Use this parameter to choose different types of reverb—if all other parameters remain at their nominal values, what you get when you change this parameter will always sound great. Because this parameter changes the structure of the reverb algorithm, it cannot be assigned an FXMod. Room types in different algorithms with similar names do not necessarily sound the same.

Rvrb Time is the RT_{60}—the time it takes for the reverb to decay to 60 dB below its initial level—in seconds. It is accurate assuming that several other parameters (HF Damping, Diff Scale, Size Scale, and Density) are at their nominal levels. It is adjustable up to Inf, which creates an infinitely-sustaining reverb.

LateRvbTim adjusts the basic decay time of the late portion of the reverb after diffusion.

L/R Pre Dly (PreDelay) is the time between the start of a sound and the output of the first reverb reflections from that sound. Longer predelays can help make larger spaces sound more realistic. Longer times can also help improve the clarity of a mix by separating the reverb signal from the dry signal, so the dry signal is not obscured.

EarRef Lvl adjusts the mix level of the early reflection portion of algorithms offering early reflections.

Late Lvl adjusts the mix level of the late reverb portion of algorithms offering early reflections.

Diff Scale scales the diffusion of the early reflections, that is, how spread out they are as a group over time. At very low settings, the early reflections start to sound quite discrete, and at higher settings the early reflections are seamless. It is adjustable from 0.00 to 2.00, with 1.00 being nominal for the given Room Type.

Density controls how tightly the early reflections are packed in time. Low Density settings group the early reflections close together, while higher values spread the reflections for a smoother reverb. It is adjustable from 0.00 to 4.00, with 1.00 being nominal (and usually optimal) for the given Room Type.

Expanse controls the amount of late reverb energy biased toward the edges of the stereo image. A setting of 0% will bias energy towards the center. Moving away from 0% will bias energy towards the sides. Positive and negative values will have a different character.

Build adjusts the envelope of certain portions of the reverb. Positive values speed up the envelope, and negative values slow it down.

Size Scale changes the size of the current room. Altering this parameter will cause a subtle coloration of the reverb. It is adjustable from 0.00 to 4.00, with 1.00 being nominal (and usually optimal) for the given Room Type.

InfinDecay, when set to **On**, causes the reverb tail to decay indefinitely. When it's **Off**, the decay time is determined by the Rvrb Time or LateRvbTim parameters.

Wet Bal (Wet Balance). Some reverb algorithms are actually two stereo reverbs in one, with each one receiving a different mono signal. This balances the outputs of the two reverbs—0% means they are being mixed equally.

Delays

There are two types of taps in the Multitap delays: The Loop tap, which can be repeated, and the numbered taps, which play a single iteration.

Fdbk (Feedback) **Level** controls the repeating function of the Loop Tap. A setting of **0**% means there will only be a single delay, while a setting of **100**% means the signal keeps repeating without ever stopping.

Both types of taps are individually adjustable from 0 to 2.55 seconds. The Loop Crs and Tap*n* Crs (*n* being the number of the tap) parameters set the coarse value of the loop in 20-ms increments, while the Loop Fine and Tapn Fine parameters set the fine value in 0.2-ms increments.

In Delay algorithms that use tempo to determine tap lengths, there is a Tempo parameter, which can be set from **1** to **255** BPM or to **System**. The Loop Length and Tap*n* Delays are then expressed in beats relative to that overall Tempo. See *Tempo-based Parameters* on page 15-25 for more information about tempo control of KDFX parameters.

Hold is a switch that, when turned on, "locks" any signal currently in the delay and plays it until Hold is turned off. When Hold is on, no signal can enter the delay and Feedback is set to 100%.

Dry Bal (Balance) is the left/right balance of the dry signal. At -**100**%, only the left dry signal goes to the left output, while at **100**% only the right dry signal passes to the right output, and at **0**%, equal amounts of the left and right dry signals pass to their respective outputs.

Tap*n* Level is the level of each numbered tap, from **0**% to **100**%, relative to the overall output of the effect.

Tap*n* Bal is the left/right balance of each of the numbered taps. At **-100**%, only the left tap goes to the left output, while at **100**% only the right tap goes to the right output, and at **0**%, equal amounts of the left and right taps pass to their respective outputs. In some delays, pairs of taps (1 and 5, 2 and 6, etc.) are controlled together as stereo pairs.

DelayScale lets you change the lengths of all the taps together. Its range is **0** to **10x**.

Note that it is possible for KDFX to run out of delay memory with over-generous settings of DelayScale or very slow Tempos. If this happens, delay times will be automatically cut in half.

Complex Echo

This algorithm has two feedback taps per channel as well as three independent taps, and also a feedback diffuser for “smearing” the delays. Feedback line 1 feeds the signal back to the delay input of the same channel, while feedback line 2 feeds the signal back to the opposite channel.

FB2/FB1>FB is a balance control between feedback lines 1 and 2. **0**% (minimum) turns off feedback line 2, only allowing use of feedback line 1. **50**% is an even mix of both lines, and **100**% (maximum) turns off line 1.

L Diff Dly and **R Diff Dly** adjusts the delay lengths of the diffusers. Range is **0** to **100 ms**.

Diff Amt adjusts the diffuser intensity. Range is **0** to **100**%.

***N* Fdbk*n* Dly** adjusts the delay length of the *N* channel’s *n*th feedback tap, fed back to the *N* channel’s delay input. Range is **0** to **2600 ms**.

Spectral Multitap Delays

These 4- and 6-tap delays have their feedback and output taps modified with shapers and filters. In the feedback path of each tap are a diffuser, hipass filter, lopass filter, and imager. Each delay tap has a shaper, comb filter, and balance and level controls.

Fdbk Image sets the amount the stereo image is shifted each time it passes through the feedback line. Range is **-100** to **100**%.

Tap *n* Shapr adjusts the intensity of the shaper at each output tap. Range is **0.10** to **6.00 x**.

Tap *n* Pitch adjusts the frequency of the comb filter at each output tap. Range is **C -1** to **C 8**, in semitones.

Tap *n* PtAmt adjusts the intensity of the comb filter at each output tap. Range is **0** to **100**%.

Equalizers (EQ)

KDFX has both Graphic and Parametric EQ algorithms. Parametric EQ sections are also available on a number of combination algorithms.

The Graphic equalizer is available as stereo (linked parameters for left and right) or dual mono (independent controls for left and right). It has 10 bandpass filters per channel, each of whose gain is adjustable from **-12 dB** to **+24 dB**.

Like all graphic equalizers, the filter response is not perfectly flat when all gains are set to the same level (except at **0 dB**), but rather has ripple from band to band. To minimize this ripple, it is best to center the overall settings around **0 dB**.

The Parametric equalizer (5-Band EQ) has two bands of shelving filters and three bands of true parametric EQ.

Treb Freq and **Bass Freq** set the center frequencies for the shelving filters. Both of these are adjustable over the full range of **16** to **25088 Hz**, in increments of a semitone.

Treb Gain and **Bass Gain** control the amount of cut or boost above (Treb) or below (Bass) the center frequency. The range is **-79** to **+24 dB**.

Mid*n* Gain sets the cut or boost for the parametric band *n*, with a range of **-79** to **+24 dB**.

Mid*n* Freq sets the center frequency for parametric band *n*, with a range of **16** to **25088 Hz**, in increments of a semitone.

Mid*n* Width sets the bandwidth of the filter on band *n*, with a range of **0.01** to **5 octaves**.

Enhancers

Enhancers modify the spectral content of the input signal by boosting existing spectral content, or stimulating new ones. Two- and three-band versions are provided.

Drive adjusts the input into each band. Increasing the drive will increase the effects. Range is **-79.0** to 2**4.0 dB**.

Xfer adjusts the intensity of the transfer curves. Range is **-100** to **100**%.

EQ Morpher

This algorithm uses two four-band bandpass filters, A and B, and moves between them, which among other things, can produce a very convincing simulation of a human vocal tract.

FreqScale offsets the filter frequencies for each set of filters. After setting the filter parameters (Freq, Gain, and Width), the FreqScale parameters will move each of the four filter frequencies together by the same relative pitch. Range is **-8600** to **8600 cents**.

Morph A>B. When set to **0**% the A parameters are controlling the filters, and when set to **100**%, the B parameters control the filters. Between **0** and **100**%, the filters are at interpolated positions. When morphing from A to B settings, the A filter #1 will change to the B filter #1, A filter #2 moves to B filter #2, and so on. Range is **0** to **100**%.

Compressors, Expanders, and Gates

A wide range of Compression and Expansion effects is available in KDFX. The various algorithms include different combinations of:

- Compressor with either soft-knee or hard-knee characteristic
- Expander
- Multiband compressor that breaks the signal up into three frequency bands and compresses them all separately
- Sidechain or output EQ
- Reverb and compressor in combination
- Gate
- Gated reverb

All of the compression algorithms use these parameters:

FdbkComprs (Feedback Compression) selects whether to use feed-forward (set this to **Out**) or feedback (set this to **In**) compression. The feed-forward configuration uses the input signal as the side-chain source. The feedback configuration uses the compressor *output* as the side-chain source. Feedback compression tends to be more subtle, but it does not allow instantaneous attack compression.

Atk (Attack) **Time** for the compressor is adjustable from **0.0** to **228.0 ms**.

Rel (Release) **Time** for the compressor is adjustable from **0** to **3000 ms**.

Smooth Time smooths the output of the expander's envelope detector by putting a lowpass filter in the control signal path. Smoothing will affect the Attack or Release times only when this parameter is longer than one of the other times. The range is **0.0** to **228.0 ms**.

Signal Dly (Delay) puts a small delay in the signal relative to the sidechain processing, so that the compressor (or gate) "knows" what the input signal is going to be before it has to act on it. This means the compression can kick in before an attack transient arrives. In a compressor, it is only really useful in feedback configuration (FdbkComprs is **In**). The range is **0** to **25 ms**.

Ratio is the amount of gain reduction imposed on the compressed signal, adjustable from **1.0:1** (no reduction) to **100:1**, and **Inf:1**.

Threshold is the level in dBFS (decibels relative to full scale) above which the signal begins to be compressed. Adjustable from **-79.0** to **0 dB**.

MakeUpGain allows additional output gain to compensate for gain reduction in the compressor. It works in conjunction (additive, in dB) with the Out Gain parameter. The range is **-79.0** to **+24.0 dB**.

Expansion

Algorithms containing Expanders have these controls:

Exp Atk (Attack), how fast the expander turns off when the input signal rises above the threshold level, adjustable from **0.0** to **228.0** ms.

Exp Rel (Release), how fast the expander turns back on after the signal drops below the threshold level, adjustable from **0** to **3000 ms**.

Exp Ratio, how much the gain is reduced below the expansion threshold, adjustable from **1:1.1** (slight downward expansion) to **1:17** (extreme downward expansion).

Exp Threshold, the level below which the signal is expanded, adjustable from **-79.0** to **0 dB**.

In addition, the two-segment compressors with expander have separate Ratio and Threshold controls for each of the compression segments.

Multiband Compression

The Multiband Compression algorithm has Attack, Release, Smooth, Signal Delay, Ratio, Threshold, and MakeUp Gain parameters for each of the three bands (Low, Mid, and High). In addition, it has the following:

Crossover1 and **Crossover2**. These set the frequencies which divide the three compression frequency bands. The two parameters are interchangeable, so either may contain the higher frequency value. The range is **16** to **25088 Hz**, in increments of a semitone.

Gates

SC Input lets you select which input channel(s) will control the sidechain, which is responsible for opening and closing the gate. It can be set to **L**, **R**, or the average of the two channels, **(L+R)/2**.

Gate Time is the time that the gate will stay open when the sidechain signal reaches the Threshold. Its range is **0** to **3000ms**.

Ducking reverses the action of the gate. Normally this if set to **Off**, and the gate opens when the input signal rises above the threshold. But when this is **On**, the gate *closes* when the input signal rises above the threshold.

Env Time is the amount of time it takes for the sidechain signal envelope to drop below the threshold. If this time is too short, the gate can close and open too quickly from amplitude modulation in the sidechain signal. If it is too long, the gate may stay closed until the envelope has a chance to fall, and some signals would not get through. This parameter is only in effect when Retrigger is **Off**.

Retrigger determines whether the gate timer will reset itself each time the sidechain signal goes above the threshold. If it is **On**, the timer resets itself, and therefore the gate stays open as long as the signal is above the threshold, or keeps going above the threshold, within the interval specified by Gate Time. If it is **Off**, the gate closes down after Env Time has elapsed—regardless of the sidechain level—and the sidechain level must fall below the threshold and come back up again before the gate will open again.

Chorus

Chorus is an effect that gives the illusion of multiple voices playing in unison. The effect is achieved by detuning copies of the original signal and summing the detuned copies back with the original. Low frequency oscillators (LFOs) are used to modulate the positions of output taps from a delay line. The delay line tap modulation causes the pitch of the signal to shift up and down, producing the required detuning.

The choruses are available as stereo or dual mono. The stereo choruses have the parameters for the left and right channels ganged, while the dual mono choruses have separate left and right controls.

Fdbk Level is the level of the feedback signal from the LFO1 delay tap into the delay line. Negative values polarity-invert the feedback signal.

Tap Lvl sets the levels of the LFO-modulated delay taps. Negative values polarity-invert the signal. Setting any tap level to **0**% effectively turns it off.

Tap Pan sets the stereo position for a given tap's output. The range is **-100**% for fully left to **100**% for fully right.

LFO Rate sets the speed of modulation of the delay lines with a range of **0.01** to **10 Hz**.

LFO Dpth sets the maximum detuning depth of the LFO-modulated delay lines, with a range from **0** to **50 cents** (1/2 semitone).

Tap Dly adds extra delay in front of the LFO modulated delay taps from **0** to **230 ms**.

L/R Phase or **LFO*n* LRPhs** adjusts the relative phases of the LFOs for the left and right channels in the stereo Choruses.

Flanger

Flanging is the process of adding or subtracting a signal with a time-displaced replica of itself, which results in a series of notches in the frequency spectrum, generally referred to as a comb filter. In KDFX, the flanger is a multi-tap delay line, all (but one) of whose taps can have their lengths modulated up and down by a low frequency oscillator (LFO). The rate of the LFO is expressed in Tempo.

StatDlyLvl (Static Delay Level) is the level of the first, nonmoving tap. Negative values invert the polarity of the tap. The range is **-100** to **100**%; **0**% turns the tap off.

DlyCrs and **DlyFin** are the coarse and fine length controls for the Static delay (StatDly…) and for the minimum value of the moving delays (Dlyn…). The coarse range is **0** to **228 ms**, and the fine range adjusts the coarse range in samples (1/48,000 sec, or 20.8μsec) from **-127** to **127**.

Xcurs Crs and **Xcrs Fin** determine how far the LFO-modulated delay taps can move from the center of their ranges (this is called *excursion*). The total range of the LFO sweep is twice the excursion. If the excursion is set to **0**, the LFO does not move and the tap behaves like a simple delay line set to the minimum delay. The coarse range is **0** to **228 ms**; the range **0** to **5 ms** is most effective for flanging. The fine range adjusts the coarse range in samples from **-127** to **127**.

Quantize + Flange

The Quantize portion of this algorithm produces digital distortion known as quantization noise by limiting the number of bits available to the signal.

DynamRange (dynamic range) controls how many bits to remove from the signal data words. At 0 dB the hottest of signals will toggle between only two bit (or quantization) levels, thereby producing a square wave. Every 6 dB added doubles the number of quantization levels, reducing the noise and getting closer to the original signal. If the signal has a lot of headroom (available signal level before digital clipping), then not all quantization levels will be reached. Range is **0** to **144 dB**.

Headroom sets the available signal level before digital clipping. Use this in conjunction with DynamRange to keep the signal level from getting too loud at low levels of DynamRange. Range is **0** to **144 dB**.

DC Offset adds a positive DC Offset to the input signal, which allows you to alter the position where digital zero is with respect to your signal. At low DynamRange settings, this can cause the output to sputter. Range is **Off/-79.0** to **0.0 dB**.

LaserVerb

LaserVerb is a new kind of reverb which produces a delayed train of closely spaced reflections, or impulses. As time passes, the spacing between the impulses gets wider, which creates a discernible buzzy pitch that gets lower as the spacing increases. The signal can be fed back into itself to extend the effect.

Dly Coarse is the overall delay length, which controls the duration or decay time. 0.5 sec is a good starting point. Range is **0** to **1.3 seconds** in the 2 PAU version of the algorithm, and **0** to **2 seconds** in the 3-PAU version.

Dly Fine adjusts the delay with a resolution down to 0.1 ms. Range is **-20.0** to **20.0 ms**.

Spacing determines the starting pitch of the descending buzz and how fast it descends, by setting the initial separation of impulses and the subsequent rate of increasing impulse separation. The spacing between impulses is given in samples (20.8μs), with a resolution of 0.1

sample. At low values, the buzz starts at high frequencies and drops slowly, while at high values the buzz starts at a lower pitch and drops rapidly. Range is **0.0** to **40.0 samples**.

Contour controls the overall shape of the reverb. When set to a high value, sounds passed through the reverb start at a high level and slowly decay. As the control value is reduced, it takes more time for the effect to build up before decaying. At a value of around **34**, the reverb behaves like a reverse reverb, building up to a hit. When it is set to **0**, the algorithm acts like a simple delay. Range is **0** to **100**%.

Filters

There are four types of Resonant Filter algorithms in KDFX.

Resonant Filter

Filter Type (**FiltType**) can be **Lowpass**, **Highpass**, **Bandpass**, or **Notch** (band-limit).

Frequency (**Freq**) is the resonant frequency of the filter. Its range is **58** to **8372 Hz**.

Resonance is the resonance of the filter, adjustable from **0** to **50 dB**.

Envelope Filter

Envelope Filter is a resonant filter whose center frequency can be made to vary according to the level of the incoming signal.

Filter Type can be **Lowpass**, **Highpass**, **Bandpass**, or **Notch** (band-limit).

Min Freq is the minimum resonant frequency of the filter, that is, the filter frequency when the input gain is below the triggering threshold. Its range is **58** to **8372 Hz**.

Sweep determines how far the resonant frequency moves when the input level increases. At positive levels it moves up in pitch, and at negative levels it moves down. The highest possible resonant frequency is 8372 Hz,; the lowest is 0 Hz. This parameter’s range is **-100**% to **+100**%.

Resonance is the resonance of the filter, adjustable from **0** to **50 dB**.

Atk Rate adjusts the upward slew of the attack portion of the envelope detector. Range is **0** to **300.0 dB/sec**.

Rel Rate adjusts the downward slew of the release portion. Range is **0** to **300.0 dB/sec**.

Smooth Rate slows down the envelope follower. If it is set to a lower rate than Atk Rate or Rel Rate, it can dominate those parameters. Range is **0** to **300.0 dB/sec**.

Trig Filt

The Triggered Filter is a sweeping resonant filter that triggers when a certain input threshold is reached, and then follows its *own* envelope, consisting of an instantaneous attack and an exponential release, rather than the envelope of the input signal.

Max Freq is the resonant frequency of the filter at the peak of the internal envelope. It can be set lower than Min Freq (above), in which case the filter will sweep downwards, then back up. Range is **58** to **8372 Hz**.

Trigger is the input-signal threshold at which the envelope detector triggers. Range is **-79** to **0 dB**.

Retrigger is the input-signal threshold at which the envelope detector resets, so that it can trigger again. This parameter is only useful when it is set below the value of Trigger. Range is from **-79** to **0 dB**.

Env Rate is the envelope detector decay rate. This can be used to prevent false triggering. When the signal envelope falls below the retrigger level, the filter can be triggered again when the signal rises above the trigger level. Since the input signal can fluctuate rapidly, it is necessary to adjust the rate at which the signal envelope can fall to the retrigger level. The range is **0** to **300.0 dB/sec**.

Rel Rate is the downward slew (release) rate of the triggered envelope generator. The range is **0** to **300.0 dB/sec**.

Smth Rate slows down the envelope follower. If set lower than the release rate, it will dominate it. You can also use the smoothing rate to lengthen the attack of the internal envelope. The range is **0** to **300.0 dB/sec**.

LFO Filter

The LFO filter is continuously swept between two resonant frequencies over a period of time. The LFO frequency, expressed in BPM and beats, can be fixed or set to follow System tempo. (See *Tempo-based Parameters* on page 15-25for more information about tempo control of KDFX parameters.)

Min Freq and **Max Freq** are the low and high limits of the resonant frequency as the filter is swept. It actually doesn’t matter which is higher; the effect will be the same. The range for both is **58** to **8372 Hz**.

LFO Shape is the waveform type for the LFO. Choices are **Sine**, **Saw+**, **Saw-**, **Pulse**, and **Tri**.

LFO PlsWid (Pulse Width). When the LFO Shape is set to **Pulse**, this sets the pulse width as a percentage of the waveform period. When the width is set to **50%**, the result is a square wave. This parameter has no effect if other waveform types are chosen. Range is **0** to **100%**.

LFO Smooth smooths (removes the higher harmonics from) the Saw+, Saw-, and Pulse waveforms. A Sawtooth wave looks more like a triangle wave, and a Pulse wave looks more like a sine wave. Range is **0** to **100%**.

Distortion

Distortion algorithms on KDFX may include a parametric equalizer or a cabinet simulator.

Dist Drive applies a boost to the input signal to overdrive the distortion algorithm into soft clipping. Since distortion drive will make your signal very loud, you may have to reduce the Out Gain as the drive is increased. Range is **0** to **96 dB**.

Warmth is a lowpass filter in the distortion control path. This filter may be used to reduce some of the harshness of some distortion settings without reducing the bandwidth of the signal. Range is **16** to **25088 Hz**.

Highpass allows you to reduce the bass content of the distortion content in the smaller distortion algorithms that don’t have true parametric EQ. Range is **16** to **25088 Hz**.

Cab Preset selects from eight cabinet simulations which have been created based on measurements of real guitar amplifier cabinets. The presets are: **Plain**, **Lead 12**, **2x12**, **Open 12**, **Open 10**, **4x12**, **Hot 2x12**, and **Hot 12**.

Cab Bypass switches on and off the cabinet-simulation part of the algorithm. When this is set to **In**, the cabinet simulation is active; when it is **Out**, there is no cabinet filtering.

Cabinet HP and **Cabinet LP** are highpass and lowpass filters to set the frequency response limits of the cabinets. Range of both filters is **16** to **25088 Hz**.

Polydistort

This is a more complex distortion algorithm that provides two, four, or six stages of distortion.

Curve *n* controls the curvature of the individual distortion stages. **0**% is no curvature (no distortion at all). At **100**%, the curve bends over smoothly and becomes perfectly flat right before it goes into clipping. Maximum value is **127**%.

LP *n* Freq are shelving frequencies for one-pole lowpass filters on each of the distortion stages. LP0 Freq handles the initial low pass prior to the first distortion stage. The other low pass controls follow their respective distortion stages. Range is **16** to **25088 Hz**.

Rotating Speakers

An algorithm that includes Rotating Speakers breaks the signal into two frequency bands, “rotates” each band separately through a virtual speaker, and then combines the outputs with a pair of virtual “microphones” whose angle relative to the speakers is adjustable.

Xover (Crossover) is the frequency at which high and low frequency bands are split and sent to separate rotating drivers. The range is **16** to **25088 Hz**.

Lo Gain and **Hi Gain** are the gains of the signal passing through the rotating woofer or tweeter, respectively. The range is **Off**/**-79.0** to **24.0 dB**.

Lo Rate and **Hi Rate** are the rotation rates of the rotating woofer and tweeter. Each driver woofer can rotate clockwise or counter-clockwise, which is determined by the sign of this parameter: assuming the microphones are positioned in front of the driver and the microphones are panned positively (positive numbers go to the right), then a positive value for this parameter causes the driver to spin clockwise when viewed from the top. The range is **-10.00** to **10.00 Hz**.

Lo Size and **Hi Size** are the effective sizes (radius of rotation) of the rotating speakers in millimeters. This affects the amount of Doppler shift or vibrato of the low frequency signal. The range is **0** to **250 mm**.

Lo Trem and **Hi Trem** control the depth of tremolo (amplitude modulation) of the signals. It is expressed as a percentage of full scale tremolo. The range is **0** to **100**%.

Mic Angle is the angle of the virtual microphones in degrees from the “front” of the rotating speaker. For the left microphone the angle increases clockwise (when viewed from the top), while for the right microphone the angle increases counter-clockwise. You should not assign an FXMod to this parameter because adjustments to it will result in large sample skips, which will cause clicks in the signal passing through. The range is **0** to **360.0 degrees**.

LoResonate and **HiResonate** are simulations of cabinet resonant modes express as a percentage. For realism, you should use very low settings. The range is **0** to **100**%.

Lo Res Dly and **Hi Res Delay** are the number of samples of delay in each resonator circuit in addition to the rotation excursion delay. The range is **10** to **2550 samples**.

LoResXcurs and **HiResXcur**s are the number of samples of delay to sweep through the resonator at the rotation rate of each rotating speaker. The range is **0** to **510 samples**.

ResH/LPhs sets the relative phases of the high and low resonators. The angle value in degrees is somewhat arbitrary and you can expect the effect of this parameter to be rather subtle. The range is **0** to **360.0 degrees**.

Vibrato/Chorus

The Vibrato/Chorus algorithm (and also the KB3 effects algorithm) simulates the vibrato and chorus effects on a Hammond organ, and is used in conjunction with the Rotary Speaker. It has several unique parameters:

VibChInOut is an in/out switch for the Vibrato/Chorus effect.

Vib/Chor is the type of Vibrato/Chorus effect to be used. The choices are from three vibratos, **V1**, **V2**, **V3**, or three choruses, **C1**, **C2**, **C3**.

Roto InOut engages or bypasses the rotary speaker effect.

Lo Beam W and **Hi Beam W** set the acoustic radiation patterns (beam width) of the two drivers in the rotating speaker. If you imagine looking down on the rotating speaker, this angle is the angle between the -6 dB levels of the beam. The range is from **45°** to **360°**. At **360°**, the driver is omnidirectional.

There are four virtual microphones, with two each on the woofer (**LoMic A** and **LoMic B**) and on the tweeter (**HiMic A** and **HiMic B**). Each microphone has the following parameters:

- **Pos** (position), the angle of the microphone from the front of the virtual speaker, from **-180** to **180 degrees**
- **Lvl** (level) from 0 to 100%
- **Pan**, the left/right panning of the microphone's output, from -100% (full left) to 100% (full right)

Tremolo and AutoPan

Tremolo is amplitude modulation using an LFO. AutoPan moves the signal between the left and right channels, using an LFO. They have several parameters in common and several unique parameters.

LFO Rate is the rate of the LFO. The range is **0** to **10.00 Hz**, or in Tremolo BPM algorithm, **0** to **12.00 x** the tempo.

Rate Scale multiplies the speed of the LFO rate into the audio range. The range is **1** to **25088 x**. When above **19x**, the values increment in semitone steps. These steps are accurate when LFO Rate is set to **1.00 Hz**.

LFO Shape is the waveform type for the LFO. Choices are **Sine**, **Saw+**, **Saw-**, **Pulse**, and **Tri**.

LFO PlsWid or **Pulse Width**: When the LFO Shape is set to **Pulse**, this sets the pulse width as a percentage of the waveform period. When the width is set to **50%**, the result is a square wave. This parameter has no effect if other waveform types are chosen. Range is **0** to **100%**.

Origin (AutoPan) determines the axis for the panning motion. At **0%**, the panning is centered between the speakers. Positive values shift the axis to the right, while negative values shift it to the left. At **-100%** or **+100%** (the range limits), there is no panning action.

ImageWidth (AutoPan) is the width of the original input program material before it is auto-panned. At **0%** (minimum), the input image is shrunk to a single point source, allowing maximum panning excursion. At **100%** (maximum), the original width is maintained so no panning can occur.

Pan Width (AutoPan) controls the amount of pan excursion. It is the percentage of total panning motion available after Origin and ImageWidth are set. Range is **0** to **100%**.

CentrAtten (Attenuation) (AutoPan) is the amount the signal level drops as it is panned through the center of the stereo image. For the smoothest tracking, a widely accepted subjective reference is **-3dB**. Values above **-3dB** will cause somewhat of a bump in level as an image passes through the center, while values below **-3dB** will cause a dip. Range is **-12** to **0 dB**.

Depth (Tremolo) controls the amount of attenuation applied when the LFO is at its deepest excursion point. Range is **0** to **100**%.

LFO Phase (Tremolo BPM) shifts the phase of the tremolo LFO relative to the beat reference. It is most useful when Tempo is set to **System**. Range is **0.0** to **360.0 degrees**.

50% **Weight** (Tremolo) is the relative amount of attenuation added when the LFO is at the -6dB point. This causes the LFO shape to bow up (positive values) or down (negative values). Range is **-16** to **3 dB**.

L/R Phase sets the phase relationship of the channels. **In** flips the left channel's LFO out of phase, with the result that the effect turns into an auto-balancer. **Out** leaves the left LFO alone.

Pitch Shifter (Pitcher)

Pitcher shifts the pitch of the incoming signal to the specified note.

Pitch is the fundamental pitch imposed upon the input, in MIDI note numbers from **C -1** to **G 9**.

Ptch Offst is an offset from the pitch frequency in semitones, from **-12.0** to **12.0**. It can be useful to assign pitch bend or a continuous controller to this parameter through an FXMod.

Odd Wts, **Pair Wts**, **Quartr Wts**, **Half Wts** are parameters that control the exact shape of the frequency response of Pitcher. An exact description of what each one does is, unfortunately, impossible, since there is a great deal of interaction between them. For more information and examples, see the algorithm documentation.

Ring Modulation

Ring modulation multiplies two signals (the "carrier" and the "modulator") together to produce unusual, often nonharmonic, overtones. The Ring Modulator algorithm in KDFX has two modes: **L*R**, in which two mono signals are modulated together; and **Osc**, in which the input is stereo, and it is modulated with the sum of five waveforms which are generated within the algorithm itself. Four of these are sine waves, and one (Oscillator 1) has a configurable waveform.

Wet/Dry: In L*R mode, the left signal is passed dry through this control.

Mod Mode selects between the two modes.

Osc1 Lvl is the level of Oscillator 1, from **0** to **100**%.

Osc1 Freq is the frequency of Oscillator 1, from **16** to **25088 Hz**.

Osc1 Shape is the waveshape of Oscillator 1, selectable from **Sine**, **Saw+**, **Saw-**, **Pulse**, and **Tri**.

Osc1PlsWid (Pulse Width). When Osc1 Shape is set to **Pulse**, this sets the pulse width as a percentage of the waveform period. When the width is set to **50**%, the result is a square wave. This parameter has no effect if other waveform types are chosen. Range is **0** to **100**%.

Osc1Smooth smooths (removes the higher harmonics from) the Saw+, Saw-, and Pulse waveforms. A Sawtooth wave looks more like a triangle wave, and a Pulse wave looks more like a sine wave. Range is **0** to **100**%.

The other four oscillators, **Sine2** through **Sine5**, each have Lvl and Freq controls.

SRS (Sound Retrieval System)

SRS, which is used under license from SRS Labs, Inc., is a single-ended processing system that produces a fully immersive, three-dimensional sound image from any audio source—mono, stereo, surround sound or encoded with any other audio enhancement technology—with two or more standard speakers.

The four parameters control the ambience of the image, and may have different optimal settings depending on the amount of stereo content in the inputs. To match the optimal settings specified by SRS Labs, the bass and treble gains should be set to 0 dB. This algorithm will have no effect on mono signals. All ranges are **-79.0** to **24.0 dB**.

Center varies the amount of “center channel” in the output. It has an **Off** position.

Space controls the width of the image. It, too, has an **Off** position.

Bass Gain and **Treb Gain** set the amount of ambience added to the low and high frequencies, respectively, in the signal. Setting both of these to **0 dB** gives a best match to the optimizations of SRS Labs.

Stereo Simulation

The Mono to Stereo algorithm converts a monaural input to simulated stereo output.

In Select selects the input signal to be “stereoized”. It can be **Left**, **Right**, or both (**(L+R)/2)**).

CenterGain is the level of the summed left and right channels. Range is **Off**/**-79.0** to **24.0 dB**.

Diff Gain is the level of the difference signal produced, which is the spatial component of the stereo signal. Range is **Off**/**-79.0** to **24.0 dB**.

DiffBassG is a gain parameter for a bass-shelf filter on the difference signal. By boosting the low frequency components of the difference signal, you can increase the sense of acoustic envelopment. Range is **-79.0** to **24.0 dB**.

DiffBassF is the transition frequency for the bass-shelf frequency. Range is **16** to **25088 Hz**.

The processed signal is split into three frequency bands—Lo, Mid, and High—each of which can be delayed and panned separately.

Crossover1 and **Crossover2** are the two Crossover frequencies at which the band-split filters split the signal into three bands. The two parameters are interchangeable: either may have a higher frequency than the other. Range is **16** to **25088 Hz**.

Pan band sets the pan position for each band. Range is **-100%** (fully left) to **100%** (fully right).

Delay band sets the delay for each band. Range is **0** to **1000 ms**.

Stereo Analyze

In this algorithm you can look at the two channels of a stereo signal, and also their inversions, sums, and differences. You can adjust their gains, and apply small delays to either or both channels.

L Invert and **R Invert** invert the phase of the channels.

L Out Mode and **R Out Mode** determine which signal is going to be metered and sent to the output of each of the channels. The choices for each are: **L** (left), **R** (right), **(L+R)/2** (normalized sum), **(L-R)/2** (normalized difference), and polarity inverted versions of these.

L/R Delay “time balances” the two signals. At negative values, the right channel is delayed, while at positive values, the left channel is delayed. The range is **-500** to **500 samples**.

RMS Settle controls how fast the RMS meters can rise or fall with changing signal levels. Range is **0** to **300 dB/second**.

The Stereo Image algorithm borrows some features from this algorithm and some from Mono to Stereo, and provides a stereo correlation meter.

FXMod Diagnostic

This algorithm allows you to view the current levels of any data sliders, MIDI controls, switches, or internally generated VAST LFOs, ASRs, FUNs, etc. which are available as modulation sources. It has no effect on any signal being routed through it.

Up to eight modulation sources may be monitored simultaneously. Meters #1 through #4 can monitor bipolar sources, meaning sources that can have both positive and negative values. The range of the bipolar meters is -1 to +1. Four monopolar meters #5 through #8 provide better resolution, but the range is limited to 0 though +1. Use the monopolar meters for sources which you do not expect to go negative.

Eight parameters are provided to connect modulation sources to the meters. The parameter values are fixed at **NoDpth** and have no function except to connect sources to meters. To use the algorithm, save an FX preset and studio containing the algorithm, then go to one of the FXMod pages of your program or setup (with the studio selected). Select the FXBus that contains the FX preset using the FXMod Diagnostic algorithm, and choose one of the meter parameters (Bipole N or Monopole N). You will not be able to modify the Adjust or Depth fields, but you can select any source you want. Finally press the **Edit** button to reenter the studio and Multieffect Editor where you can view the meters on parameter page 2.

Bipole1 through **Bipole4** attach bipolar modulation sources (those that can go positive or negative) to the bipolar meters. The parameters are not adjustable.

Monopole5 through **Monopole8** attach monopolar modulation sources (can go positive only) to the monopolar meters. The parameters are not adjustable.

Special Topics

Song Mode

We’ve referred to the special aspects of KDFX in Song mode elsewhere in the manual. Here is a summary of what’s been said, along with some new information.

FX Mode in Song Mode

If FX Mode on the Effects-mode page is set to **Program**, and you put the K2600 into Song mode, control of KDFX remains with the program that is on the current FX Channel, as set on the Effects-mode page. However, if FX Mode is set to **Auto**, and you put the K2600 into Song Mode, KDFX is under the control of the program that is on the effect channel *within* the song—from Song mode press **Edit** to go to the EditSong:COMMON page, and look at the EffectChan parameter. In fact, when FX Mode is set to **Auto**, the program on the song’s EffectChan controls KDFX *any* time a song is playing, even if you’re not in Song mode.

The effect channel is saved with the song, so make sure that it is set so that the program that you want to be in control of KDFX is active on that channel.

Clock Setting for tempo-based parameters

The clock setting in Song mode determines how tempo (BPM)-based parameters behave. If you are using the K2600’s internal tempo clock, the Clock parameter on the MISC page in Song mode must be set to **Int**. If you are using an external MIDI timing source, such as a sequencer, then that parameter must be set to **Ext**.

If you are using Tap Tempo as a tempo source, then the Clock parameter must be set to **Internal**.

Recording a Setup in Song Mode

In order to record a setup into the internal sequencer, you first need to set the RecTrk parameter on the MAIN Song page to **Mult**. (If you are using an external keyboard that sends on only one MIDI channel, you must also go to the RECEIVE page in MIDI mode and turn on the LocalKbdCh parameter, setting it to agree with the MIDI Out channel of your keyboard.)

There is a complication, however, if you want to record FXMods while you’re recording your track using the setup. The sequencer will faithfully record all of your controller movements, but when it plays them back, it plays them as part of a single track. That track contains a program, not a setup, because the sequencer doesn’t know how to play setups.

Since all of the FXMods were part of the setup, even though the data on the track is being played back, it’s not going anywhere, and no FXMods are being played.

The solution is to have a *program* playing back the data you’ve just recorded, which has the identical KDFX studio and FXMods as the setup you used to record with. The data has been recorded on the MIDI channel of Zone 1 of the setup. Find the program on that channel, press **Edit** to get inside it, and then import the KDFX studio from the setup you used originally—that’s what the ImpFX function is for.

Now save that program (in the same location or a new one) and go back to the song. Make sure that the program’s channel and the song’s effect channel agree. The FXMods should now all play correctly.

Studios and Songs in Master Mode

When you are using KDFX in Master mode, the studio does not get saved with the song, since it is not a dependent object of the song, or of anything in it. Therefore, if you want to recall a studio whenever you load in a particular song, you either have to:

- Remember to recall the studio by hand, or
- Go into FXMode: Program, and save the studio as part of a program which is on one of the tracks of the song.

Remember, just because you are in FXMode:Program doesn’t mean you *have* to use FXMods—you can leave the studio alone, just as you would in Master mode.

KB3 Effects

In the K2600’s KB3 mode, KDFX processing is being used to its utmost. Several studios have been included for use with KB3. These studios have a special characteristic: the FXBuses and Aux bus are used together, forming a “virtual” single processing algorithm with 7 (count ’em) PAUs.

There are two algorithms that are designed for this special use: Algorithm **779 KB3 FXBus** (to be used on an insert FXBus), and Algorithm **780 KB3 AuxFX** (to be used on the Aux bus). They don’t work very well individually, and should probably not be used that way.

Chapter 16
DSP Functions

This chapter explains the DSP functions that can be inserted into the algorithms in the Program Editor. As you configure each algorithm, the DSP functions you select determine the type of synthesis you apply to your sounds. Deciding which algorithm to use depends on what you want to do; there's no hard and fast rule. If you want to create a classic analog sound, for example, you'll choose one of the algorithms containing one or more blocks that can have filter functions assigned to them. If you want real-time panning effects, choose an algorithm that includes the PANNER function in the F3 block. Your best approach is to study the algorithm charts in the *Musician's Reference*, and choose the algorithm that includes the functions you want to work with.

Before we get to the explanations of the DSP functions, we've included a brief discussion of a few general concepts of sound synthesis. This should help you understand the workings of the DSP functions. We'll refer to these concepts repeatedly as we go along.

Any single sound waveform is composed of numerous sine wave components, each at a different frequency. These components are called partials. The lowest frequency is perceived by the ear as the pitch of the sound, and is called the fundamental. The other components are called harmonics. The relative amplitudes (volume) of each of the partials in a sound determine its timbre, its most recognizable characteristic. When you think of the difference between the sound of a piano and a saxophone, you're thinking about their different timbres. A dull sound has a strong fundamental and weak harmonics, while a bright sound has strong harmonics.

Sound synthesis can be most simply described as the manipulation of either the amplitude or phase of one or more of the partials constituting a sound. The K2600's various DSP functions give you a variety of methods for manipulating those partials. We've grouped our explanations of the DSP functions according to the types of specialized manipulation they enable you to perform on a given sound. The categories are as follows:

Filters

Equalization (EQ)

Pitch / Amplitude / Pan Position

Mixers

Waveforms

Added Waveforms

Nonlinear Functions

Waveforms with Nonlinear Inputs

MIxers with Nonlinear Inputs

Synchronizing (Hard Sync) Functions

Introduction to Algorithm Programming

Programming the algorithms is a multi-step process. The first step is selecting an algorithm. Changing the algorithm of an existing program's layer is likely to alter the sound of the layer dramatically. As a rule, then, you won't want to change a layer's algorithm unless you're building a sound from scratch. Furthermore, when you change a layer's algorithm, the values for each of the DSP functions within the algorithm may be set at nonmusical values; you should lower the K2600's volume slider before changing algorithms.

Deciding which algorithm to use for a new sound is primarily a process of planning a layer's signal path through the sound engine. The real sound manipulation is done by the DSP functions you insert into the algorithm. The algorithm simply lays a framework that determines how the DSP functions interact.

Once you know which algorithm you're going to work with, you'll assign various DSP functions to each of the stages of the algorithm. These stages, as you recall, are represented by the rectangular blocks you see on the ALG page. The arrows pointing down at the blocks represent control inputs that affect the behavior of the DSP functions. For each arrow, there's a page of parameters controlling some aspect of the DSP function's behavior. Every DSP function has at least one control input; several have two or three.

The ALG page is where you select algorithms and assign DSP functions to the algorithm's various stages. To assign a DSP function, move the cursor to select the stage you want to modify, then use any data entry method to scroll through the list of available DSP functions for that stage. You'll normally hear the effect of each selection as soon as you make it. If you don't hear a difference, it's because the function's control parameters aren't set to significant values. Once you adjust some of these parameters, the function will have a noticeable effect on the sound. Keep in mind that not all DSP functions are available at every stage of every algorithm.

When you have each stage of the current algorithm set up to your liking, you can begin to program the control inputs of each DSP function. This is done by selecting the control-input page(s) for the currently selected DSP function, and adjusting the parameters on the page. There are two ways to select the control-input pages: you can move the cursor to select the DSP function you want to tweak, and press **Edit**. The selected DSP function's control-input page will appear (if it's a multi-stage DSP function, its first control-input page will appear). Or you can use the soft buttons to select the pages. The **PITCH** soft button always selects the pitch control-input page, since the first stage of every algorithm is invariably the pitch control. The **F1–F4** soft buttons select the control-input pages corresponding to the remaining four arrows, which point down at the subsequent four variable control inputs.

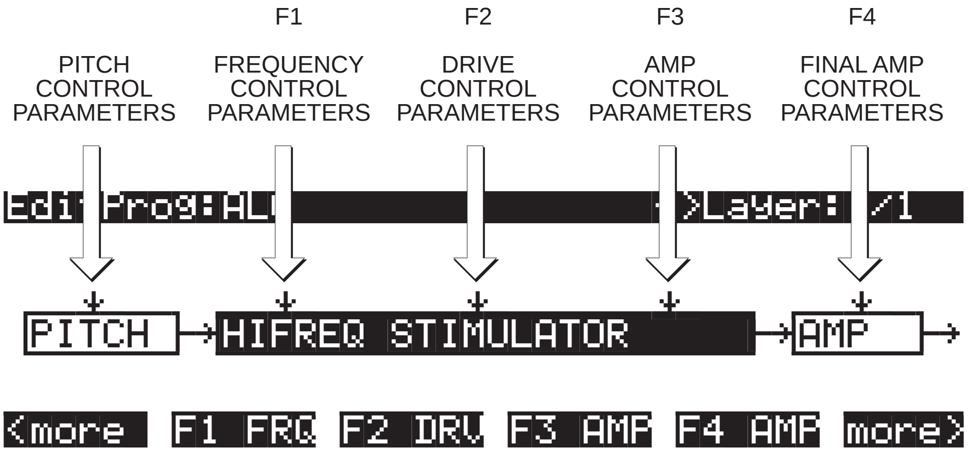


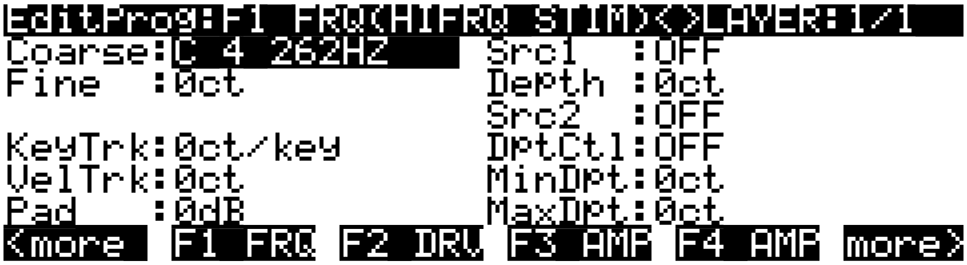
Figure 16-1 Input Control for DSP Functions

Each control-input page contains several parameters, which affect some aspect of the behavior of the DSP function named on the top line of the page. Most of these parameters are the common DSP control parameters; for a review, see *Common DSP Control Parameters* on page 6-14.

The possibilities are truly enormous, given the number of different combinations of functions you can assign to any particular layer (not to mention multi-layer programs, each layer of which has its own algorithm). You can create completely new sounds just by tweaking the parameters on the control-input page for a single DSP function. When you begin adjusting these parameters, it’s a good idea to start with all of them set to **0** (or the value that minimizes their effects), then adjust them one by one. This will help you understand exactly what effect each parameter has, and will give you an idea of the variety of effects each parameter can produce. Then you can start combining the effects of multiple parameters, and quite possibly take your sound in a whole new direction. You’ll quickly become familiar with the control-input pages for the DSP functions, since most of them contain the same parameters, with just a few variations. You’ll find that they all look much alike. The top line of each page, however, will indicate which DSP control input it represents (PITCH, or F1–F4).

For example, on the page below, the top line tells you that the currently selected DSP function is the high-frequency stimulator—its name is abbreviated and enclosed in parentheses. You can also see that you’re looking at F1, which in this case controls the frequency of the high-frequency stimulator. So the top line of these pages always shows three things:

1. The currently selected control input (PITCH or F1–F4);
2. The aspect of the current DSP function controlled by the input (this varies depending on the current DSP function);
3. The currently selected DSP function (usually abbreviated, and in parentheses). Items 1 and 2 match the label of the soft buttons that select each page. The page below, for example, is selected with the soft button labeled **F1 FRQ**



```
EditProg:F1 FRQ(HIFRQ STIM)< >LAYER:1/1
Coarse:C 4 262HZ    Src1  :OFF
Fine  :0ct          Depth :0ct
                    Src2  :OFF
KeyTrk:0ct/key      DptCtl:OFF
VelTrk:0ct          MinDpt:0ct
Pad   :0dB          MaxDpt:0ct
<more  F1 FRQ  F2 DRV  F3 AMP  F4 AMP  more>
```

Additional Parameters

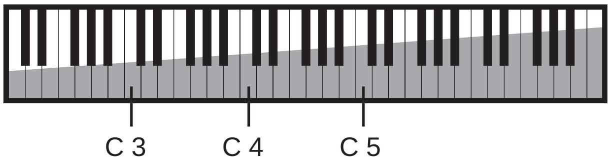
In addition to the common DSP control parameters you’ll find on each page, you’ll also see a few others on various pages. They’re described here, since programming them is the same regardless of the page on which they appear. Depending on the DSP function they affect, they’ll have different ranges of values and different units of measurement (%, dB, etc.).

Pad

Many of the DSP functions boost the signal as it passes through. Depending on the signal’s input level and the amount of gain (boost) introduced by any given DSP function, its output may clip, which will alter the sound considerably. Clipping may also occur as a result of phase shifting, but this is not as common as clipping caused by gain. While you may find clipping to be a useful component of some sounds, you’ll want to remove it from others. The Pad parameter, which appears on the control-input pages of many DSP functions, lets you attenuate (reduce the amplitude of) the signal by 6, 12, or 18 dB at the input of those functions. Use the Pad parameter to reduce or eliminate any undesired clipping caused by the currently selected DSP function.

Key Track Start (KStart)

This parameter appears on many control-input pages, and gives you added control over the effect of key tracking. For each note you play, it multiplies the value of the KeyTrk parameter by a number that varies with the note's MIDI key number. If KeyTrk is set to **0**, this parameter will have no effect. When KeyTrk is a nonzero value, KStart will modify the normal key tracking curve, which is shown is the diagram below. The effect of normal key tracking reaches its minimum at C -1, and its maximum at C 9. You can use KStart to dampen the effects of key tracking at one end of the keyboard. If key tracking causes a sound to clip or distort toward the high end of the keyboard, for example, you can use KStart to reduce the effect of the key tracking at the upper end without changing its effect on the lower end. To do this you would set a negative value for KeyTrk, and a unipolar value for KStart.



Normal KeyTrk curve
(Positive KeyTrk value)

Unipolar Keystart

The range of values for KStart is **C 1** to **C 9** unipolar, and **C -1** to **C 9** bipolar. Unipolar and bipolar values have different effects on the key tracking. The next three diagrams illustrate the effect of three different *unipolar* keystart values on the key tracking curve when a positive value is assigned for the KeyTrk parameter. At a KStart value of **C 4**, for example, there is no key tracking effect below Middle C (it multiplies the key tracking amount by a key number value of **0**). The key tracking value is multiplied by 0 at C 4 (normal key tracking), by 1 at C$^{\#}$ 4, by 2 at D 4, and so on to a maximum of 64 at 5 1/3 octaves above the KStart value. For higher notes, the key tracking effect would still increase on its own, but the effect of keystart would not increase further. At a KStart value of C 3, the key tracking value would be multiplied by 0 for C 3 and all notes below, by 1 for C$^{\#}$ 3, and so on. The key number value would reach its maximum of 64 before reaching C 9. When KStart is set above **C 4**, its effect on key tracking will continue to increase up to C 9, but will not reach full scale at C 9.

You'll use unipolar values for KStart when you want to cancel the key tracking effect on a DSP function over a sizable portion of the keyboard, but have it increase or decrease throughout the rest of the keyboard's range. Set high unipolar values for KStart when you want to remove key tracking from the lower notes, applying it only to the higher notes. If you have set a positive value for KeyTrk, set low unipolar values when you want to apply key tracking to the lower notes and pin it at its maximum throughout the upper range of the keyboard. You may want to use low values of key tracking in this case, depending on the DSP function you're applying.

When the value of the KeyTrk parameter is negative, remember that the key tracking is at its minimum effect at C 9, and maximum at C -1. In this case, the key tracking effect will be reduced for notes above the KStart setting. For notes below the keystart note, the normal key tracking amount will apply.

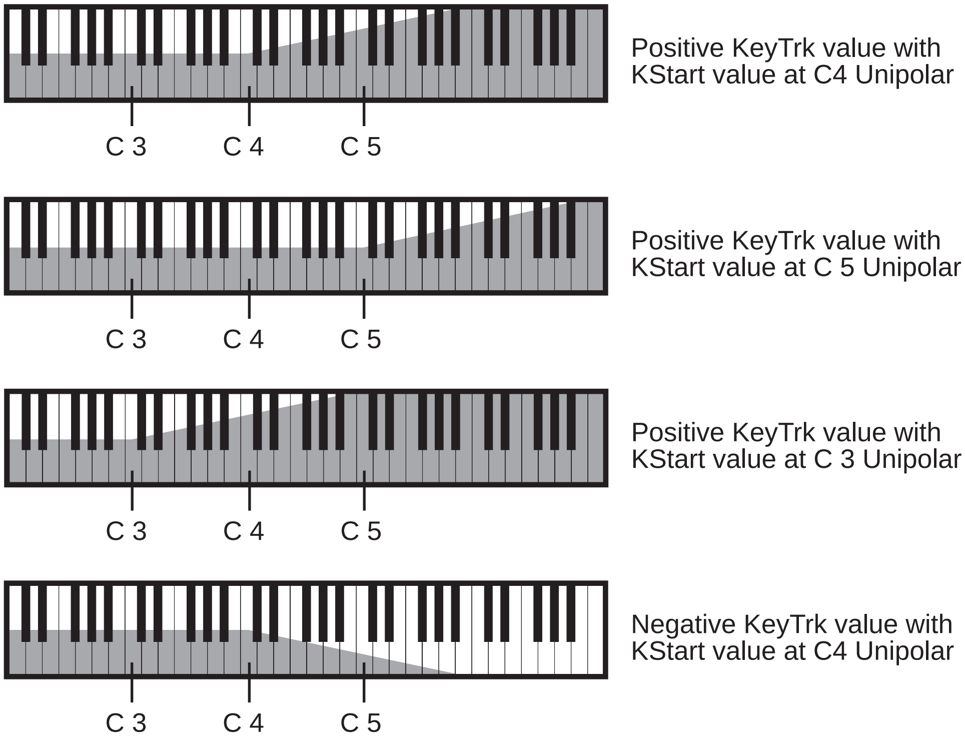

Positive KeyTrk value with KStart value at C4 Unipolar
C 3
C 4
C 5
Positive KeyTrk value with KStart value at C 5 Unipolar
C 3
C 4
C 5
Positive KeyTrk value with KStart value at C 3 Unipolar
C 3
C 4
C 5
Negative KeyTrk value with KStart value at C4 Unipolar
C 3
C 4
C 5

Figure 16-2 Unipolar Keystart

Bipolar Keystart

For bipolar KStart values with positive key tracking values, the effect on key tracking depends on whether the KStart parameter is set above or below **C 4**. When it's above, the effect on key tracking will be minimum at **C -1**, reaching its maximum effect on key tracking at the keystart setting. The normal key tracking curve applies above the keystart setting. When KStart is set below **C 4**, the effect on key tracking is maximum at C 9, decreasing with each successive note closer to the keystart setting, and remaining constant at the keystart setting and below. The normal key tracking curve applies below the keystart setting.

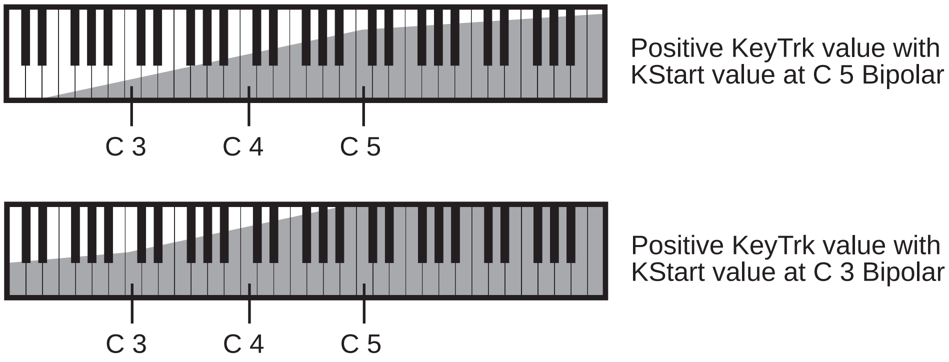

Positive KeyTrk value with KStart value at C 5 Bipolar
C 3
C 4
C 5
Positive KeyTrk value with KStart value at C 3 Bipolar
C 3
C 4
C 5

Figure 16-3 Bipolar Keystart

Use bipolar settings for KStart when you want to gradually increase or decrease the key tracking effect of the currently selected DSP function across the entire keyboard range. With KStart at **C 4** bipolar, playing C 4 will apply the DSP function at the level you set with the Adjust parameter, and will increase or decrease with higher or lower notes, depending on your settings for KeyTrk.

When KeyTrk is set to a negative value, the effect on key tracking is reversed. For keystart settings above **C 4**, the effect on key tracking will be maximum at C -1, decreasing with each note closer to the keystart setting, and remaining constant at and above the keystart setting. For keystart settings below **C 4**, the effect on key tracking will be minimum at C 9, increasing with each note closer to the keystart setting, and remaining constant for notes at and below the keystart setting.

KStart is available for many of the ***nonlinear DSP functions***, like SHAPER and WRAP. If you like the control it gives you, you can simulate its effect by using the FUNs. To simulate unipolar keystart, assign Key Number (KeyNum) as one of the inputs to a FUN, and select one of the diode equations for the FUN's Function parameter. To simulate bipolar keystart, assign Bipolar Key Number (BKeyNum) as one of the inputs of a FUN. Then assign those FUNs to some other control-source parameter.

The DSP Functions

Filters

One-pole Lowpass
Two-pole Lowpass
Two-pole Lowpass, -6 dB resonance
Two-pole Lowpass, +12 dB resonance
Four-pole Lowpass with separation
Gated Lowpass
One-pole Highpass
Two-pole Highpass
Four-pole Highpass with separation

One-pole Allpass
Two-pole Allpass
Two-pole Notch
Two-pole Notch, fixed width
Double Notch with separation
Two-pole Bandpass
Two-pole Bandpass, fixed width
Twin Peaks Bandpass

Filters are widely used in synthesis to change the timbre of a sound by manipulating the amplitude of specific partials. When using filters, you always set a reference point (cutoff or center frequency) that determines which partials the filters affect. Here's a quick summary of the effects of the filter functions.

Lowpass filters cut the levels of all partials above the cutoff frequency without affecting the partials at or below the cutoff frequency (the low frequencies pass through). Highpass filters do the opposite; they cut the levels of all partials *below* the cutoff frequency without affecting the partials at or above the cutoff frequency.

Notch filters, as the name implies, cut the levels of partials in a range between high and low frequency. Consequently the "cutoff" frequency is referred to as the center frequency. With notch filters, the levels of partials at the center frequency are cut, while the levels of partials above and below the center frequency are unaffected. Bandpass filters are the opposite of notch filters; they leave the levels of partials at the center frequency unchanged, and cut the levels of partials above and below the center frequency.

The use of lowpass, highpass, notch, and bandpass filters is often referred to as subtractive synthesis, since the timbre of a sound is changed by removing certain partials.

Allpass filters, instead of cutting or boosting the partials of a sound, change the phase of the partials as their frequencies pass through the center frequency.

Filter Terminology

Rolloff Filters do not usually cut all frequencies precisely at their cutoff point. Instead, the amplitude of the frequencies above (or below, in case of a hi pass filter) the cutoff decrease by a fixed amount per octave—for example, 6 dB per octave. This curve of lessening amplitude is called a rolloff.

Poles The number of poles in a filter affect how sharp the rolloff is. The more poles there are, the sharper the rolloff, meaning that the cutoff will have a more dramatic effect on the sound. The K2600 has one-pole, two-pole, and four-pole filters available. A one-pole filter has a 6 dB per octave cutoff; a two-pole is 12 dB per octave; and a four-pole is 24 dB per octave.

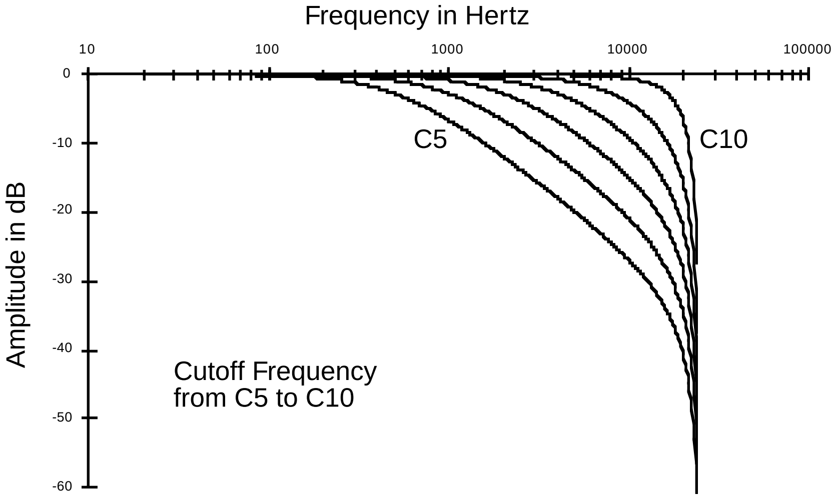
Resonance In a filter that has resonance, the frequencies near the cutoff are given an increase or decrease in amplitude. If you decrease these frequencies, you are essentially creating a longer rolloff. But if you increase those frequencies thereby emphasizing them, it creates a distinctive sound that you will very likely recognize. Resonance is also sometimes called Emphasis or Q on various synthesizers. Resonance on the K2600 is implemented in one of two ways. On some filters, the resonance is fixed, adding or subtracting a specific amount of dB to the affected frequencies (the ones near the cutoff). On other filters, you can control the amount of resonance applied. In the case of these filters, there will always be a separate control page for the resonance.

Separation Four of the filters in the K2600 (both Four-Pole filters, the Double Notch, and the Twin Peaks) are actually two filters combined into one DSP function. For these filters, you will find a control page called Separation. This allows you to shift the cutoff frequency of the second filter, creating a separation in the cutoff frequencies of the two filters. In the case of the Notch and Band Pass filters, this can be used to create two separate notches or band passes. In the case of the four-pole filters, it affects the shape of the roll off. For the four-pole filters, separation set to **0** creates sharp rolloff of 24dB per octave.

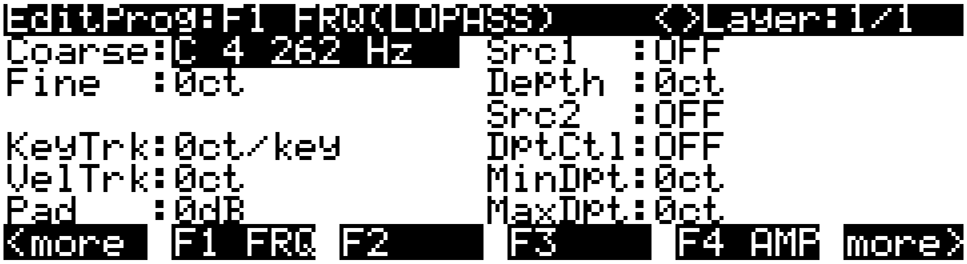
How to Read the Graphs

The graphs show the rolloff curve, using several different values to show how they change the shape of the curve. Amplitude is always on the vertical axis. Frequency is always on the horizontal axis. You will notice on several graphs that the curve becomes more dramatic as the cutoff frequency is set at a higher value. This is because the highest frequency the K2600 can produce is 20Khz, so as the cutoff is set to higher values, there are fewer frequencies available before it is past the range of the K2600.

One-pole Lowpass Filter (LOPASS)



Frequencies below the cutoff frequency are unaffected by this filter. At the cutoff frequency, the signal is attenuated 3 dB. There’s a rolloff of 6 dB per octave above the cutoff frequency—that is, the signal is attenuated 6 dB with each octave above the cutoff. The resonance—the amount of cut or boost at the cutoff frequency—is fixed at -3dB. When the cutoff frequency is well below the lowest-frequency partials of a sound, lowering the cutoff further will not affect the timbre of the sound, but will reduce its overall amplitude.

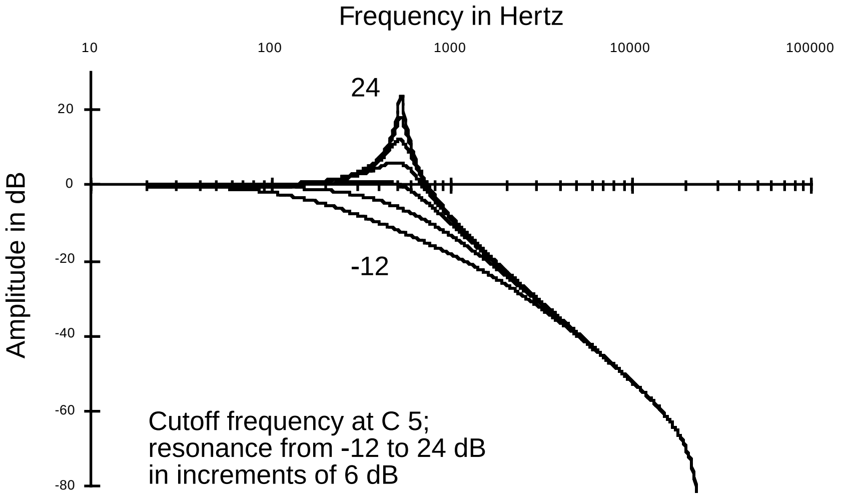


```
EditProg:F1 FRQ(LOPASS)       <>Layer:1/1
Coarse:C 4 262 Hz   Src1   :OFF
Fine  :0ct          Depth  :0ct
                    Src2   :OFF
KeyTrk:0ct/key      DptCtl:OFF
VelTrk:0ct          MinDpt:0ct
Pad   :0dB          MaxDpt:0ct
<more  F1 FRQ  F2     F3     F4 AMP  more>
```

Parameter	Range of Values
Coarse Adjust	C 0 16 Hz to G 10 25088 Hz
Fine Adjust	± 100 cents
Key Tracking	± 250 cents per key
Velocity Tracking	± 10800 cents
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 10800 cents
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 10800 cents
Maximum Depth, Source 2	± 10800 cents

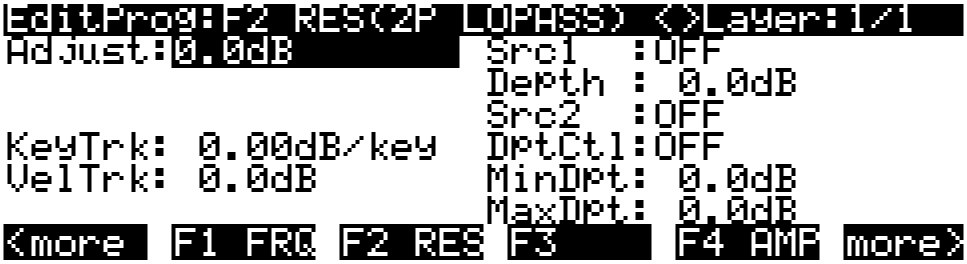
The Coarse Adjust parameter sets the cutoff frequency in terms of a key name. The remaining parameters (except Pad) alter the cutoff frequency in increments of cents. You'll notice that positive values for key tracking have an interesting effect on the function of lowpass filters; positive key tracking values raise the cutoff frequency for high notes and lower it for low notes. More specifically, a value of **100 cents per key** on this page, when filtering a constant waveform like a sawtooth, would result in waveforms of exactly the same shape for all pitches of the waveform. The cutoff frequency moves in sync with the frequencies of the waveform's partials as different pitches are generated. Negative key tracking values will steepen the rolloff of lowpass filters above the cutoff. The Pad parameter, as always, attenuates the signal at the input to the function. These parameters affect all the lowpass filters similarly.

Two-pole Lowpass Filter (2POLE LOWPASS)



The two-pole lowpass filter has a rolloff of 12 dB per octave above the cutoff frequency. This is a two-stage function, so it has two control-input pages. The first affects the cutoff frequency, and has the same parameters as the one-pole lowpass. The second control-input page (F2 RES) affects the resonance of the filter. Resonance is a cut or boost in amplitude of the partials in the vicinity of the cutoff frequency.

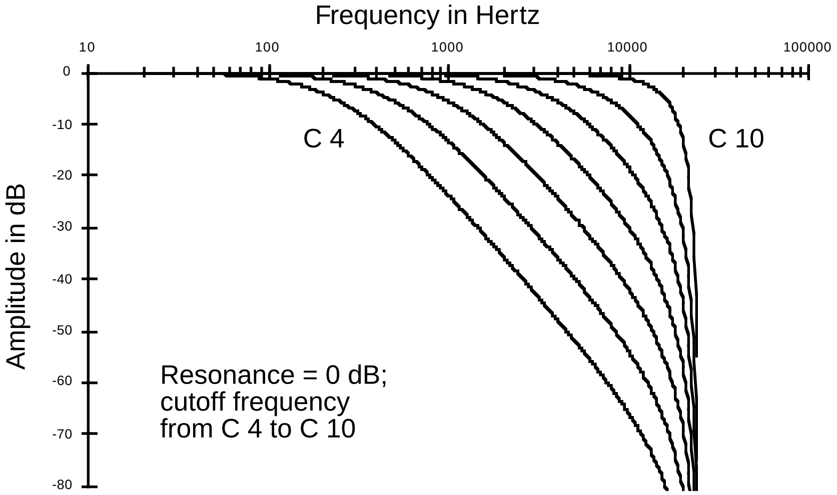
Set the resonance with the Adjust parameter on the F2 RES control-input page, and use the other parameters to assign various controls to alter it. If a boost is applied at frequencies where there are high-amplitude partials, the signal may clip. The Pad parameter on the F1 FRQ page will reduce the clipping, but there's no harm in keeping it if you like the sound.



```
EditProg:F2 RES(2P LOPASS) <>Layer:1/1
Adjust:0.0dB          Src1   :OFF
                      Depth : 0.0dB
                      Src2   :OFF
KeyTrk: 0.00dB/key    DptCtl:OFF
VelTrk: 0.0dB         MinDpt: 0.0dB
                      MaxDpt: 0.0dB
<more   F1 FRQ  F2 RES  F3      F4 AMP  more>
```

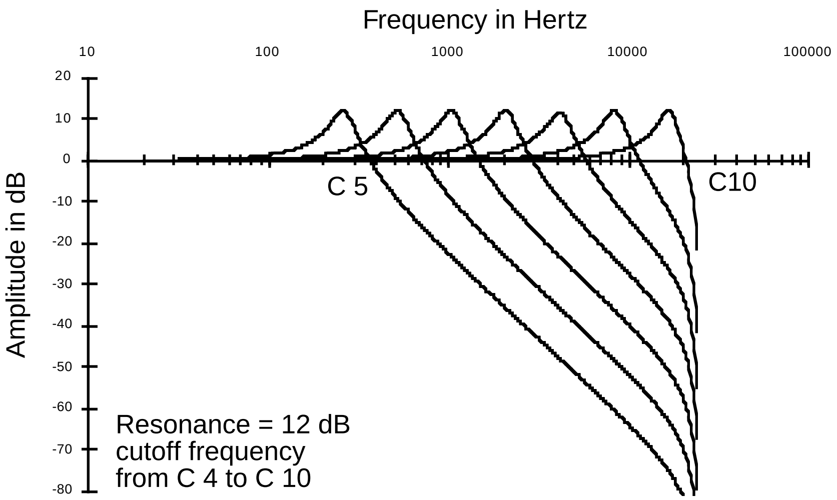
Parameter	Range of Values
Adjust	-12 to +48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	-30 to +60 dB
Source 1	Control Source list
Source 1 Depth	-30 to +60 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	-30 to +60 dB
Maximum Depth, Source 2	-30 to +60 dB

Two-pole Lowpass Filter, -6 dB Resonance (LOPAS2)



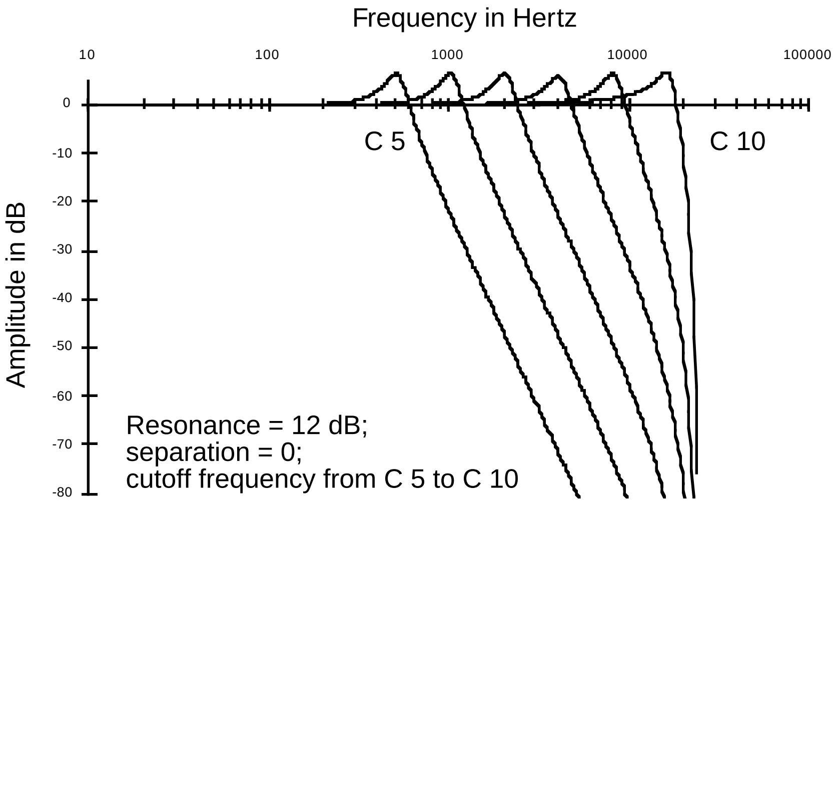
Using this filter is the same as using two one-pole lowpass filters in successive algorithm blocks. Since its resonance is fixed at -6 dB, it's also the same as using 2POLE LOWPASS with the resonance set to -6 dB. You'd use this filter when you want a 12 dB per octave rolloff but don't need to set a resonance level. This would leave you free to use another DSP function in the algorithm, since it's a one-stage function.

Two-pole Lowpass Filter, +12 dB Resonance (LP2RES)

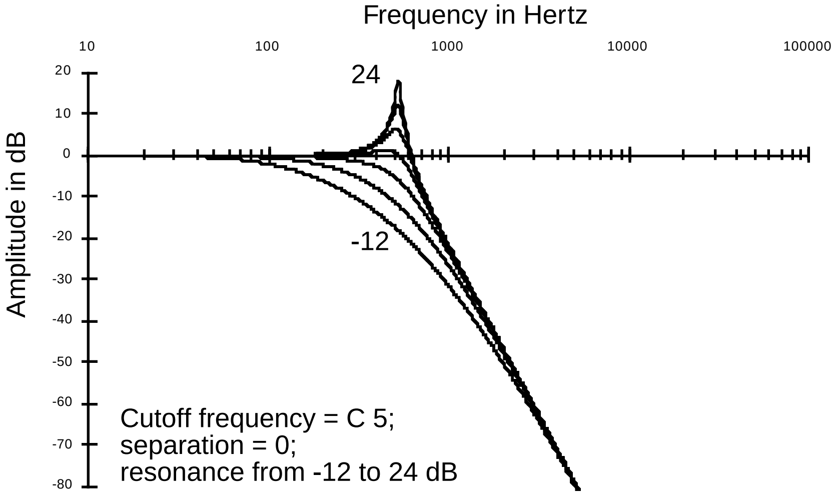


This is similar to LOPAS2; the only difference is that its resonance is fixed at +12 dB.

Four-pole Lowpass Filter with Separation (4POLE LOPASS W/ SEP)

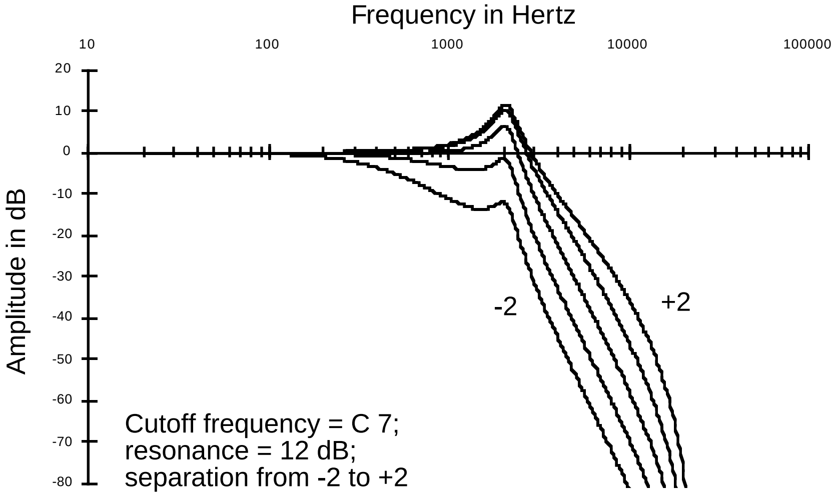


4-Pole Lowpass Filter: Resonance



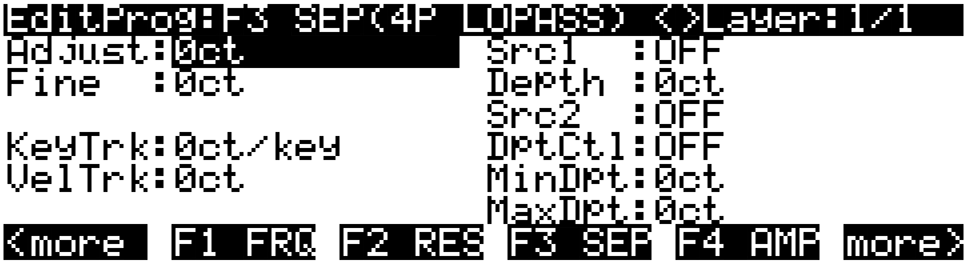
Cutoff frequency = C 5;
separation = 0;
resonance from -12 to 24 dB

4-Pole Lowpass Filter: Separation in Octaves



Cutoff frequency = C 7;
resonance = 12 dB;
separation from -2 to +2

This combines 2POLE LOWPASS and LOPAS2 in one three-stage function. The parameters on the F1 FRQ control-input page affect the cutoff frequencies of both filters. The parameters on the F2 RES page affect the resonance of 2POLE LOWPASS. The parameters on the F3 SEP page shift the cutoff frequency of LOPAS2, creating a separation between the cutoff frequencies of the two filters. Positive values raise the cutoff frequency of LOPAS2, while negative values lower it. If no separation is applied, there's a 24 dB per octave rolloff above the cutoff frequency.



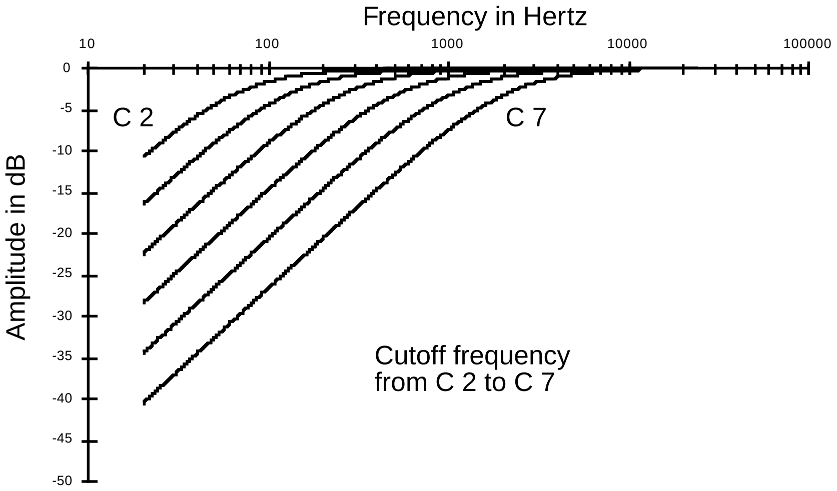
```
EditProg:F3 SEP(4P LOPASS) <>Layer:1/1
Adjust:0ct          Src1  :OFF
Fine  :0ct          Depth :0ct
                    Src2  :OFF
KeyTrk:0ct/key      DptCtl:OFF
VelTrk:0ct          MinDpt:0ct
                    MaxDpt:0ct
<more  F1 FRQ  F2 RES  F3 SEP  F4 AMP  more>
```

Parameter	Range of Values
Coarse Adjust	± 10800 cents
Fine Adjust	± 100 cents
Key Tracking	± 250 cents per key
Velocity Tracking	± 10800 cents
Source 1	Control Source list
Source 1 Depth	± 10800 cents
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 10800 cents
Maximum Depth, Source 2	± 10800 cents

Gated Lowpass Filter (LPGATE)

You may be familiar with gates as applied to effects like reverb, where the effect shuts off abruptly after a specified time. The gated lowpass filter produces a somewhat similar effect in terms of the sound's amplitude. The filter's cutoff frequency is controlled by the AMPENV. When the AMPENV is at 100%, the cutoff frequency is high, so most of the partials are heard. When the AMPENV decays or releases to 0%, the cutoff frequency is low, so only the lowest partials are heard. You'll hear the distinct effect of the filter closing as the amplitude envelope releases.

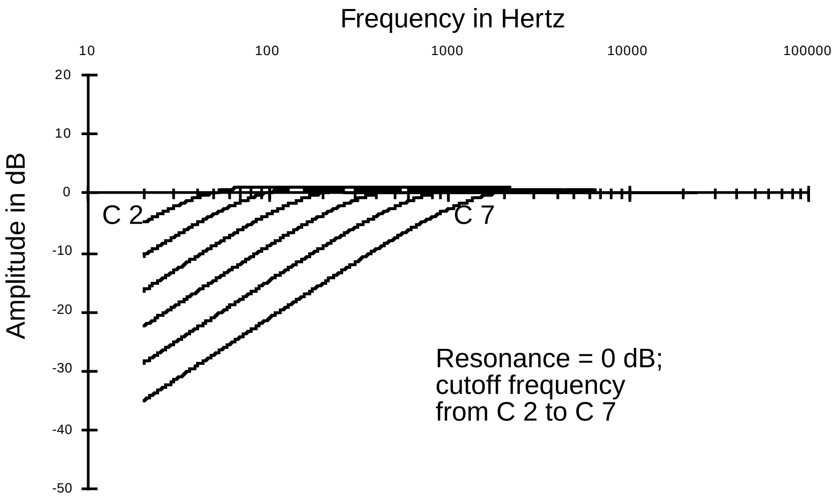
One-pole Highpass Filter (HIPASS)



High-frequency partials pass through this filter unaffected. At the cutoff frequency, the signal is attenuated 3 dB. There's a roll-off of 6 dB per octave below the cutoff frequency. The resonance is fixed at -3dB. When the cutoff frequency is well above the highest-frequency partials of a sound, raising the cutoff further will not affect the timbre of the sound, but will merely attenuate it further.

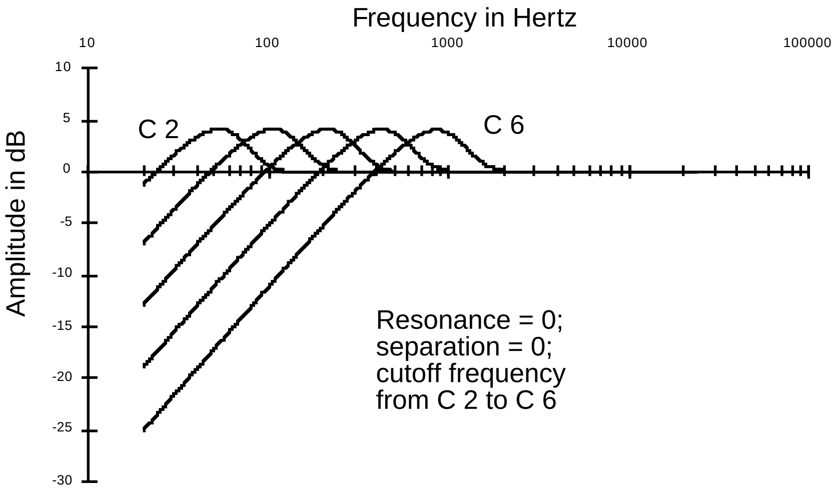
The Coarse Adjust parameter sets the cutoff frequency in terms of a key name. The remaining parameters (except Pad) alter the cutoff frequency in increments of cents. Positive key tracking values raise the cutoff frequency for high notes and lower it for low notes. More specifically, a value of 100 cents per key on this page, when filtering a constant waveform like a sawtooth, would result in waveforms of exactly the same shape for all pitches of the waveform. The cutoff frequency moves in sync with the frequencies of the waveform's partials as different pitches are generated. Negative key tracking values will steepen the rolloff of highpass filters below the cutoff. The Pad parameter, as always, attenuates the signal at the input to the function. These parameters affect all the highpass filters similarly.

Two-pole Highpass Filter (HIPAS2)

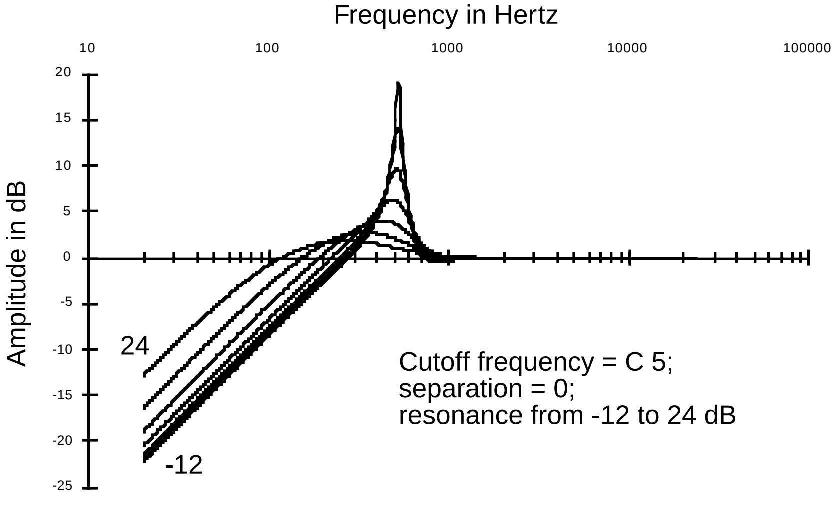


This is very similar to HIPASS. The primary difference is in the steepness of the rolloff at the cutoff frequency. Below the cutoff frequency, the rolloff is similar to that of HIPASS, except that there’s a one-octave shift—that is, HIPASS with a cutoff frequency of C 3 will sound nearly the same as HIPAS2 with a cutoff of C 4. In other words, HIPASS gives you greater attenuation of low frequencies when set to the same cutoff frequency as HIPAS2.

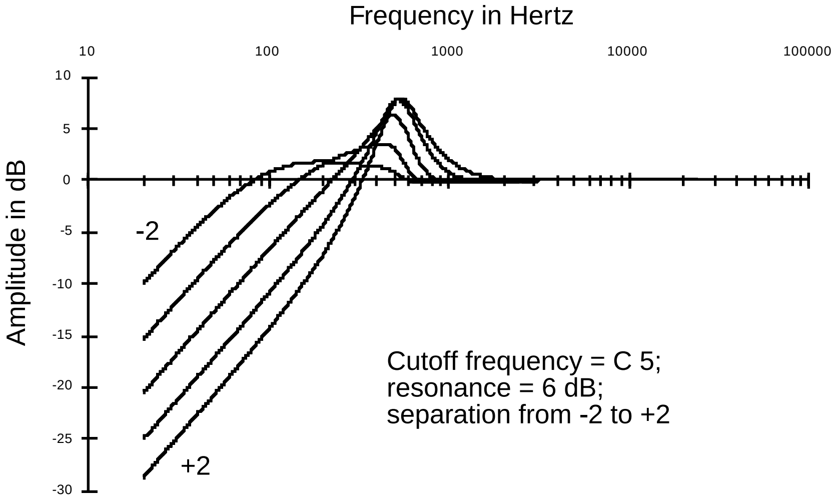
Four-pole Highpass Filter with Separation (4POLE HIPASS W/ SEP)



4-Pole Highpass Filter: Resonance

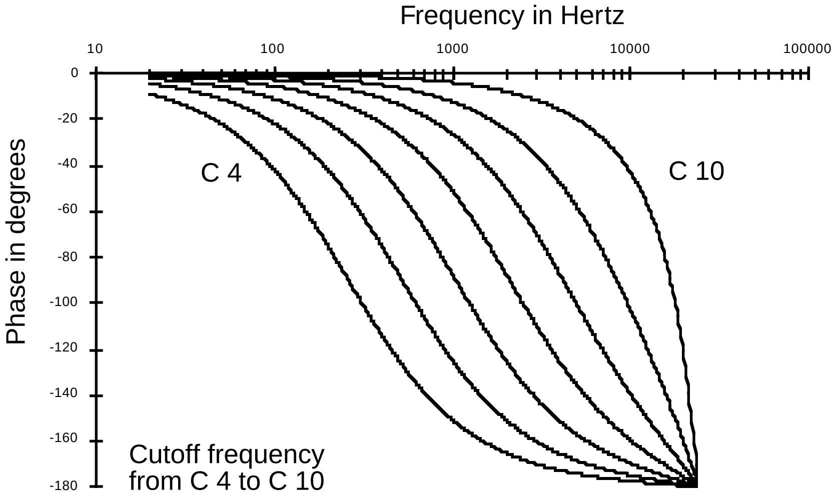


4-Pole Highpass Filter: Resonance



This combines two of the 2POLE HIPASS filters into one three-stage function. It has a rolloff of 6 dB per octave below the cutoff frequency. The parameters on the F1 FRQ control-input page affect the cutoff frequencies of both filters. The parameters on the F2 RES page affect the resonances of the first filter. There will always be a slight extra boost of partials at the cutoff frequency, even at low resonance settings. The parameters on the F3 SEP page shift the cutoff frequency of the second 2POLE HIPASS, creating a separation between the cutoff frequencies of the two filters. Positive values raise the cutoff frequency of the second 2POLE HIPASS, while negative values lower it. If no separation is applied, there's a 24 dB per octave rolloff below the cutoff frequency. A variety of responses can be produced by adjusting the resonance and separation settings.

One-pole Allpass Filter (ALPASS)

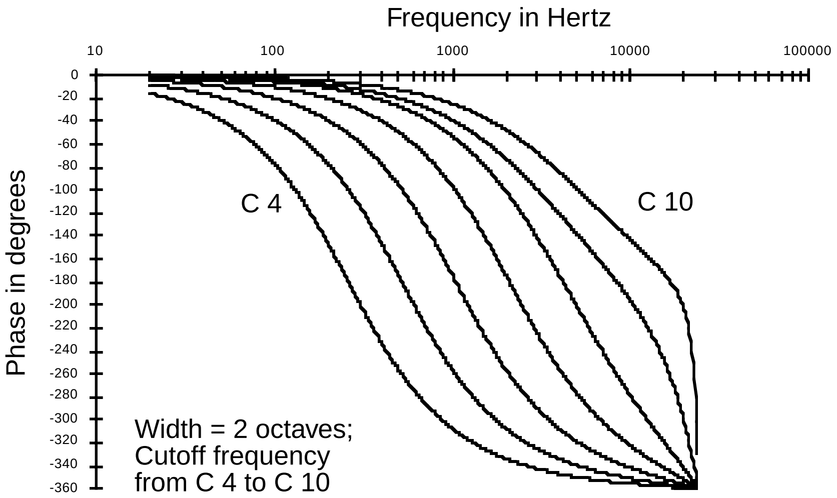


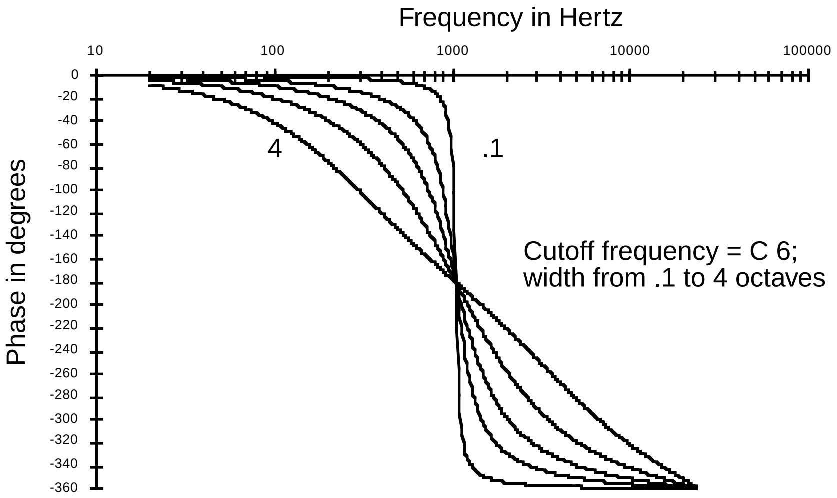
Allpass filters do not affect a sound’s frequency response (the amplitude of partials at various frequencies), but change the phase of each partial depending on its proximity to the center frequency. The phase shift is -90 degrees for partials at the center frequency. It rises toward 0 degrees for partials at frequencies below the center, and falls toward -180 degrees for partials at frequencies above the center. With low-frequency waveforms, you’ll be able to hear this phase shift. As a rule, however, the ear is not sensitive to phase shifts unless they’re changing, so you’ll usually want to use Source 1 or 2, and assign an LFO to sweep the center frequency up and down.

Periodic phase shifts like those induced by an LFO on the center frequency will cause a vibrato-like variation in the pitch of a sine wave input. This vibrato effect will be less regular for more complex partials. The greater the depth setting of the control source using the LFO, the greater the vibrato effect.

The amount of vibrato effect also depends on the speed and amount of the phase shift. Try adjusting the rate of the LFO controlling the center frequency. Another way to increase the amount of phase shift is to use the two-pole allpass filter, or to use the one-pole allpass filter in more than one algorithm block.

Two-pole Allpass Filter (2POLE ALLPASS)

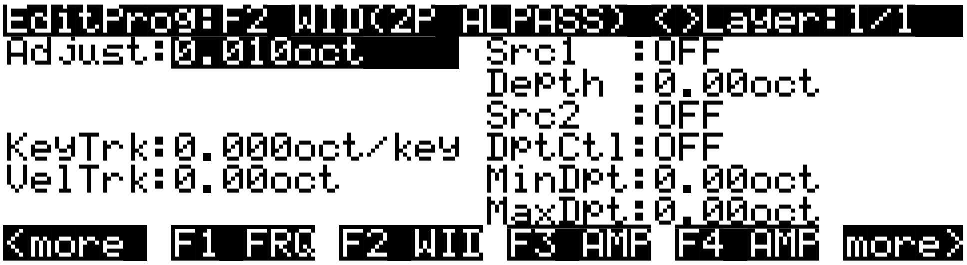




Using 2POLE ALLPASS is very similar to using ALPASS, with two differences. First, the phase shift is -180 degrees for partials at the center frequency, approaching 0 degrees for partials at low frequencies, and approaching -360 degrees for partials at high frequencies.

Second, since this is a two-stage function, there's an additional control-input page (F2 WID) which controls the filter width. The parameters on this page affect the frequency range, measured in octaves, where most of the phase shifting occurs. Small values cause a drop from 0 to -360 in the phase shift to occur near the center frequency, while large values spread the drop in the phase shift over a broader frequency range. Small values tend to affect just a few partials, leaving others mostly untouched. The affected partials seem to become detached from the others, creating the illusion of an additional sound source.

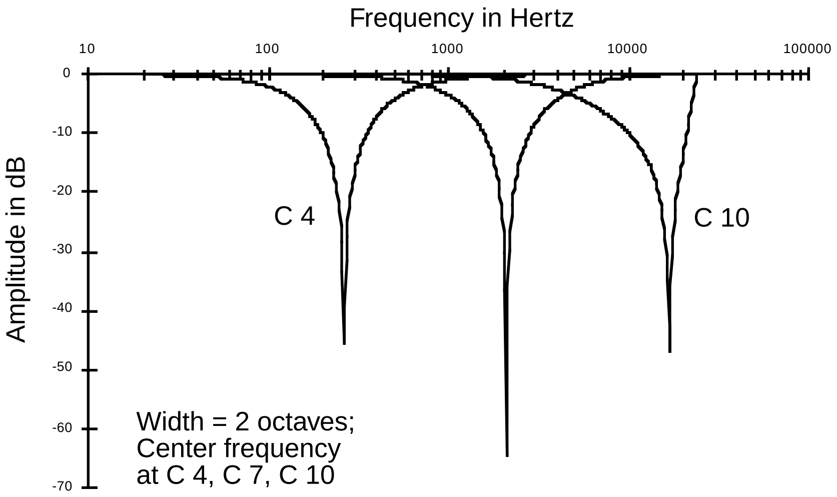
If you leave the center frequency constant and assign an LFO to vary the width, partials with frequencies above the center will shift their pitches in the opposite direction of partials below the center frequency.



```
EditProg:F2 WID(2P ALPASS) <>Layer:1/1
Adjust:0.010oct    Src1  :OFF
                   Depth :0.00oct
                   Src2  :OFF
KeyTrk:0.000oct/key DptCtl:OFF
VelTrk:0.00oct     MinDpt:0.00oct
                   MaxDpt:0.00oct
<more  F1 FRQ  F2 WID  F3 AMP  F4 AMP  more>
```

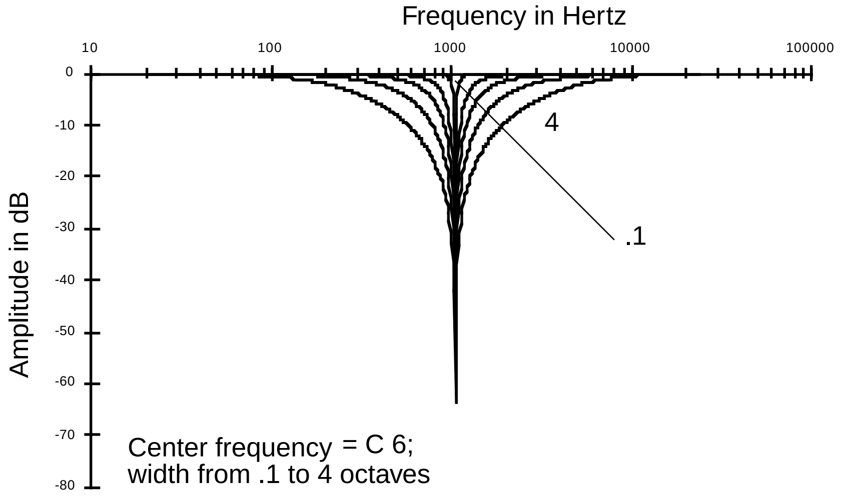
Parameter	Range of Values
Adjust	0.010 to 5.000 octaves
Key Tracking	± .200 octaves per key
Velocity Tracking	± 5.00 octaves
Source 1	Control Source list
Source 1 Depth	± 5.00 octaves
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 5.00 octaves
Maximum Depth, Source 2	± 5.00 octaves

Two-pole Notch Filter (NOTCH FILTER)



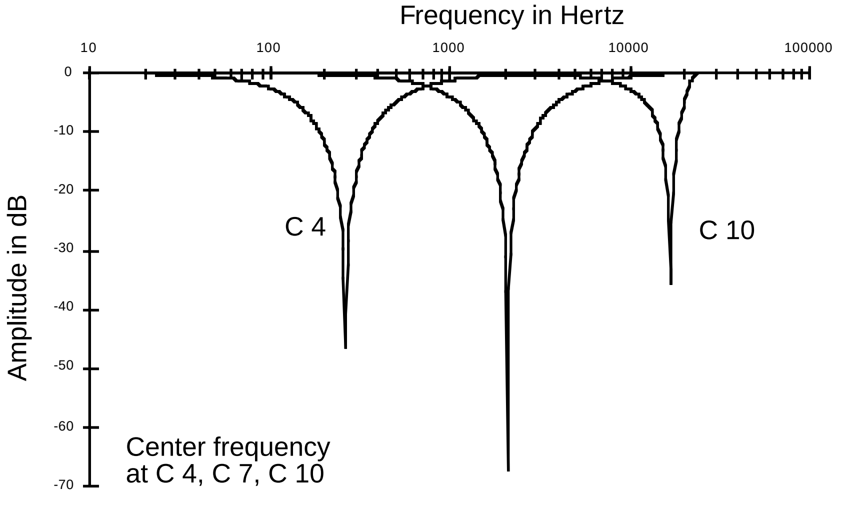
Width = 2 octaves;
Center frequency
at C 4, C 7, C 10

2-Pole Notch Filter: Width in octaves



The two-pole notch filter has two control-input pages, one for center frequency, one for width. Partials with frequencies above or below the notch will be unaffected. Within the notch, partials will be attenuated according to the width of the notch. The width is defined in terms of the number of octaves between the points on the signal's attenuation curve where the attenuation is 3 dB (see the explanation of F2 WID for the PARAMETRIC EQ function—page 16-26). For example, if the width is set at four octaves, then the attenuation will be 3 dB at two octaves in either direction from the center frequency. There's no attenuation of partials at more than two octaves in either direction from the center frequency.

Two-pole Notch Filter, Fixed Width (NOTCH2)



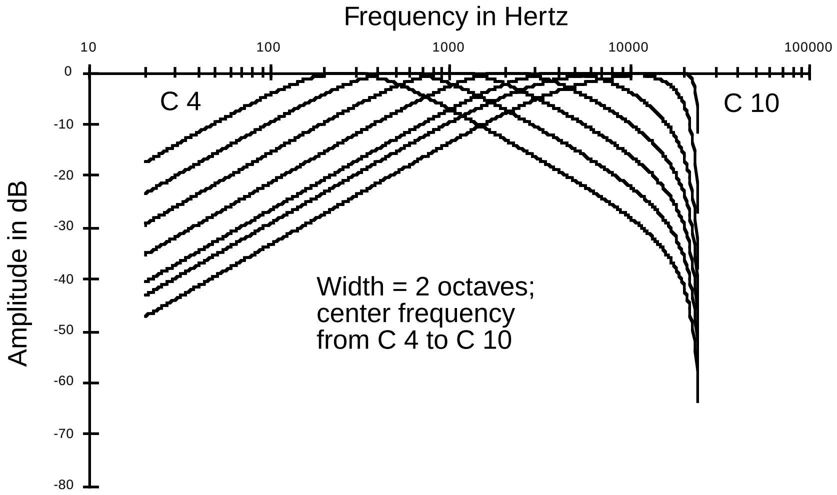
The only functional difference between NOTCH2 and NOTCH FILTER is that the width of NOTCH2 is fixed at 2.2 octaves. This gives you a one-stage notch filter function.

Two-pole Bandpass Filter (BANDPASS FILTER)

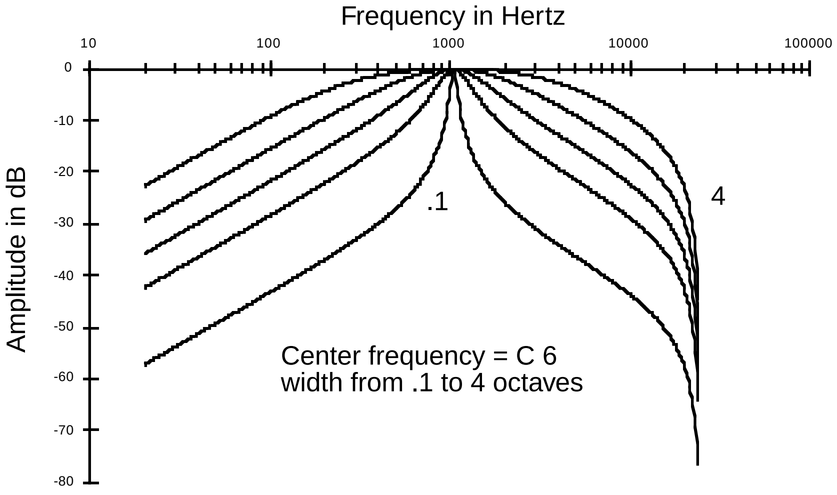
This is essentially the opposite of a notch filter; it passes all partials at the center frequency, and cut the levels of partials above or below the center frequency. The width is defined the same as for the double notch filter.

The gain at the center frequency is 0 dB. Small values for width (a narrow bandpass) may produce a very quiet signal unless the center frequency matches the frequency of a strong sine wave partial. Wide bandpasses may result in a quiet signal if they're centered in a region of the sound where the partials are weak. You can easily boost these quiet signals with the parameters on the F4 AMP page.

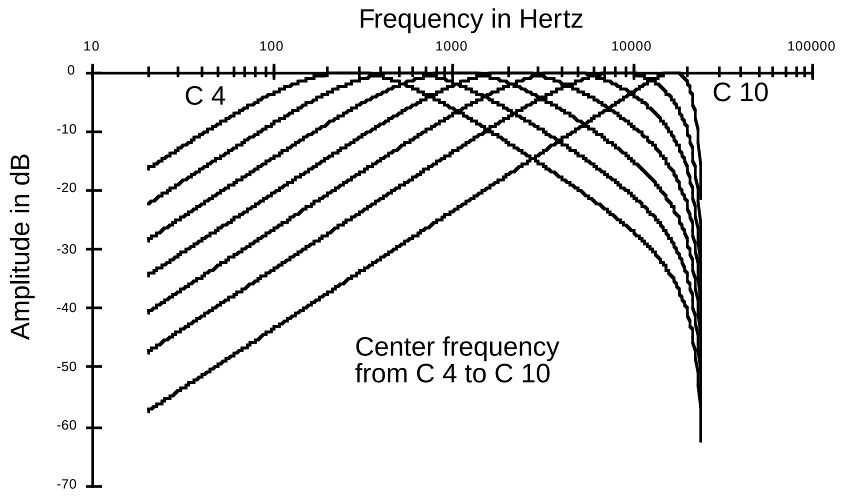
Bandpass Filter: Frequency



Bandpass Filter: Width in octaves

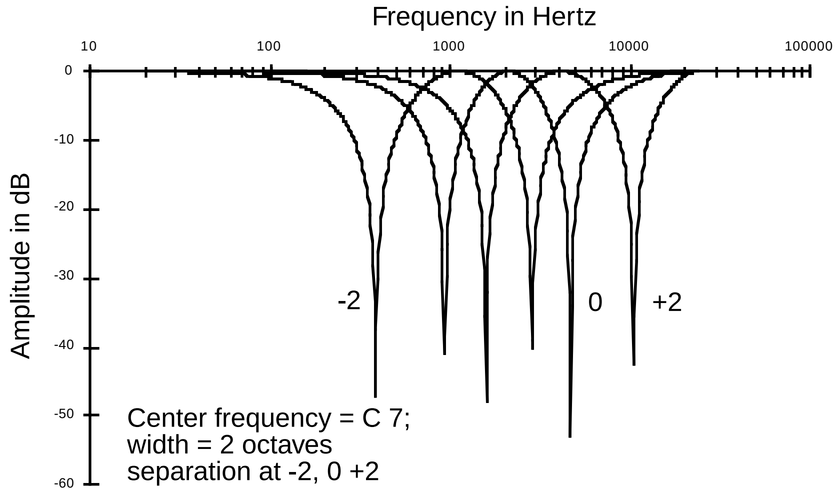


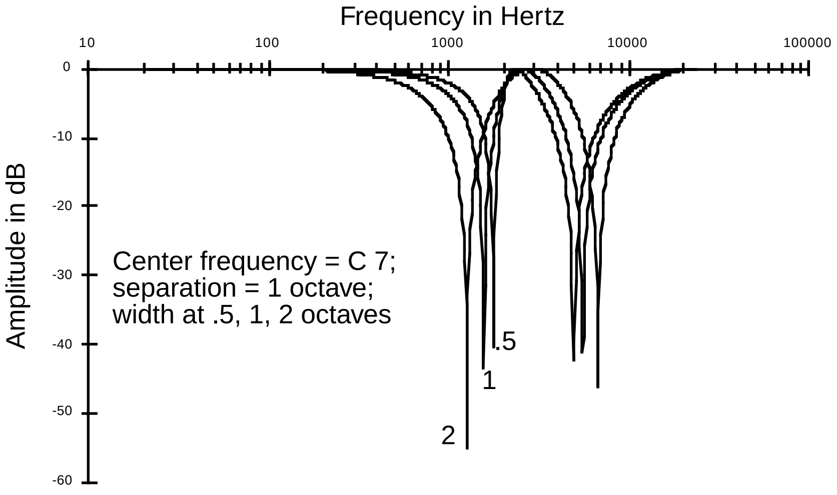
Two-pole Bandpass Filter, Fixed Width (BAND2)



The only functional difference between BAND2 and BANDPASS FILTER is that the width of BAND2 is fixed at 2.2 octaves. This gives you a one-stage bandpass filter function.

Double Notch Filter with Separation (DOUBLE NOTCH W/ SEP)



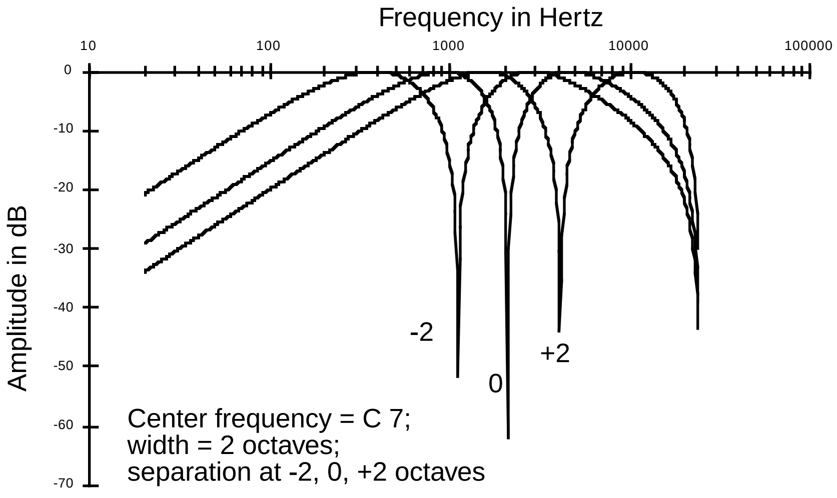


This is a three-stage function that puts two notches in the frequency response. As with NOTCH FILTER and NOTCH2, there are control-input pages for frequency and width. A third control-input page affects the separation of the notches.

Setting the center frequency on the F1 FRQ page defines the frequency that's halfway between the notches. The settings for the separation affect the behavior of the width control parameters. When the separation is 0, the notches are close to the center frequency, and the width control parameters control the widths of both notches equally. Positive values for separation move the notches apart, and cause the width control parameters to affect the width of the higher-frequency notch more than the width of the lower-frequency notch. Negative values for separation will move the notches apart to the same extent, but will cause the width control parameters to affect the width of the lower-frequency notch more than the width of the higher-frequency notch.

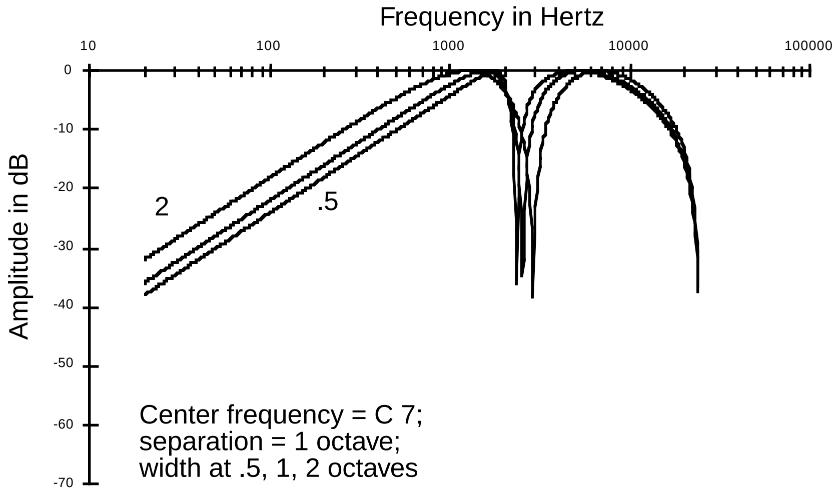
Twin Peaks Bandpass Filter (TWIN PEAKS BANDPASS)

The control parameters for TWIN PEAKS BANDPASS work the same way as for DOUBLE NOTCH FILTER, but of course, you get peaks instead of notches—that is, the amplitudes of partials near the center frequency are high, and the amplitudes are increasingly attenuated at frequencies farther from the center.



Center frequency = C 7;
width = 2 octaves;
separation at -2, 0, +2 octaves

Twin Peaks Bandpass Filter: Separation



Center frequency = C 7;
separation = 1 octave;
width at .5, 1, 2 octaves

Twin Peaks Bandpass Filter: Width

Equalization (EQ)

Equalization is a specialized filtering process that lets you boost or cut the amplitude of a specified range of frequencies.

Parametric EQ

Mid-range Parametric EQ

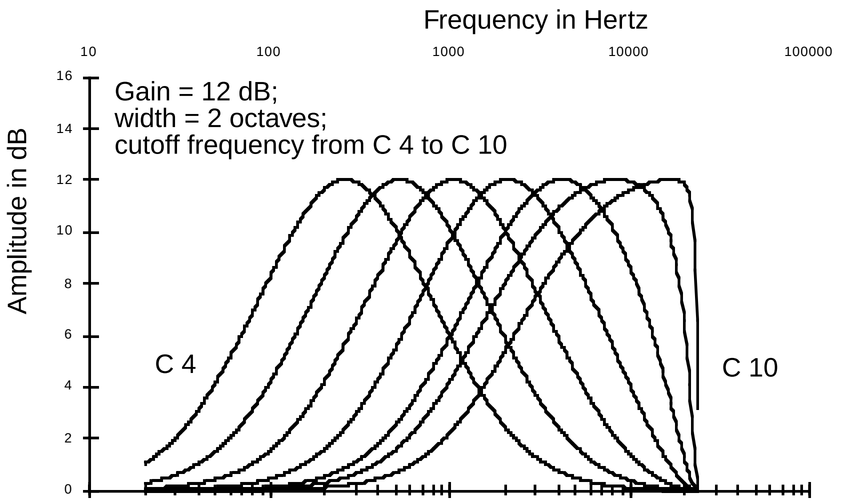
Bass Tone Control

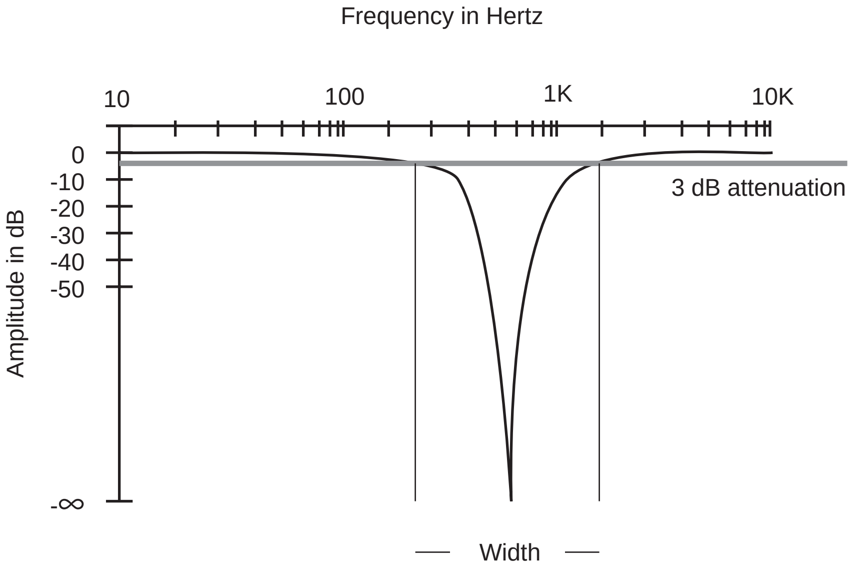
Treble Tone Control

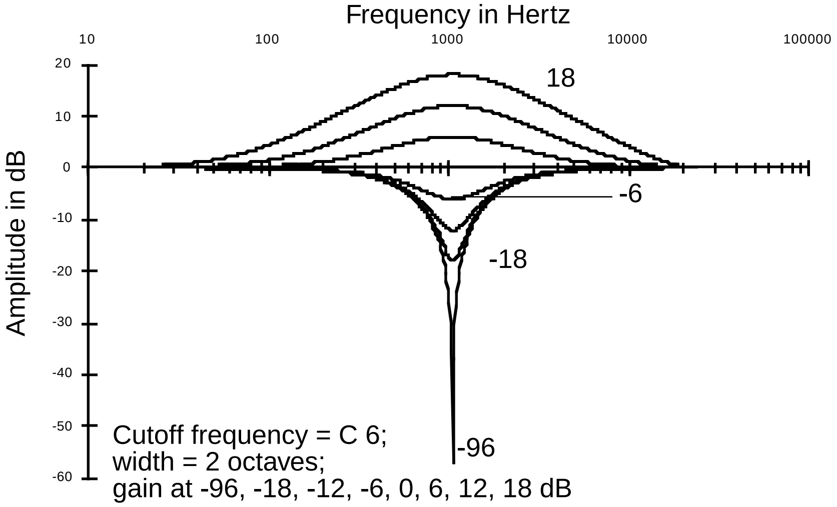
Steep Bass Tone Control

Parametric Equalizer (PARAMETRIC EQ)

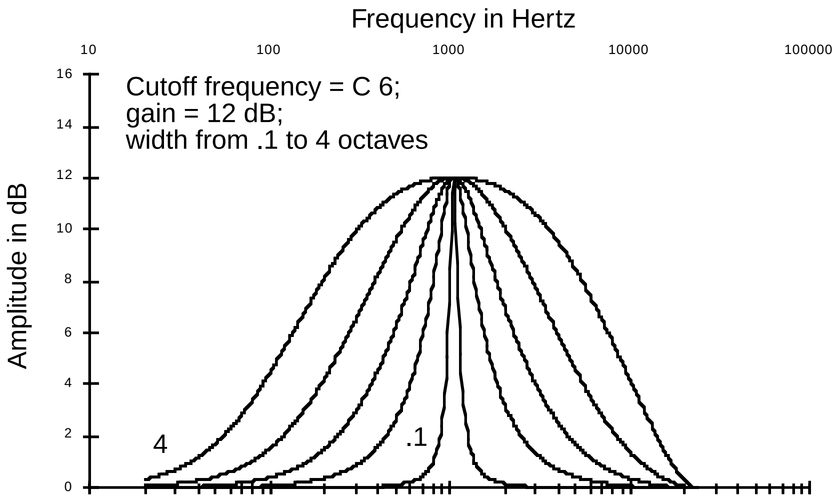
This function has three interacting parameters, each with its own control-input page: center frequency, width, and amplitude. The center frequency is the center of the range of frequencies that will be boosted or cut by the amplitude setting. The width is the entire range of frequencies that will be affected by the amplitude setting. For the K2600, the width is defined by imagining an amplitude curve with a level (in dB) of -infinity (minus infinity) at the center frequency, then measuring the distance (in octaves) between the points on the curve where the amplitude is attenuated by 3dB. See the diagram below.

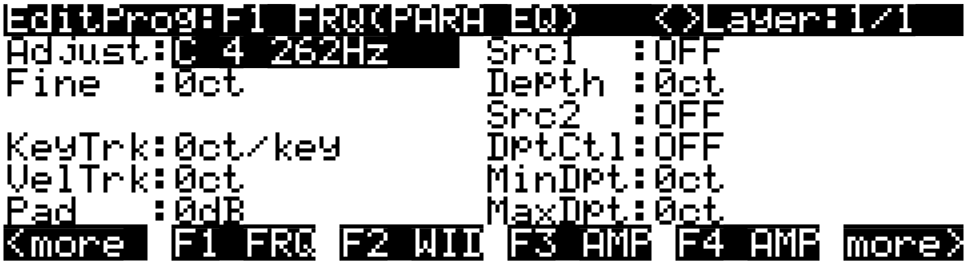






When you're using the Parametric EQ, you might use the following sequence. Set the center frequency (press the **F1 FRQ** soft button to select its control-input page). The frequency is measured in terms of each note of the keyboard. The frequency in Hertz of each note appears with the note name as the value for the Adjust parameter. Next, select the width control-input page (the **F2 WID** soft button) to determine how wide a range will be affected by the amplitude adjustment. Then select the amplitude control-input page (**F3 AMP** soft button), and adjust the amplitude of the range you specified with the center frequency and width settings. You'll probably jump back and forth between these three pages until your ear is satisfied with the sound.

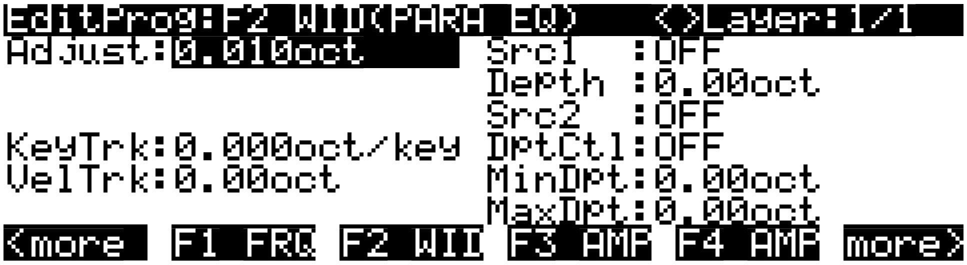




```
EditProg:F1 FRQ(PARA EQ)    <>Layer:1/1
Adjust:C 4 262Hz       Src1  :OFF
Fine  :0ct             Depth :0ct
                       Src2  :OFF
KeyTrk:0ct/key         DptCtl:OFF
VelTrk:0ct             MinDpt:0ct
Pad   :0dB             MaxDpt:0ct
<more  F1 FRQ  F2 WID  F3 AMP  F4 AMP  more>
```

Parameter	Range of Values
Coarse Adjust	C 0 16 Hz to G 10 25088 Hz
Fine Adjust	± 100 cents
Key Tracking	± 250 cents per key
Velocity Tracking	± 10800 cents (9 octaves)
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 10800 cents
Maximum Depth, Source 2	± 10800 cents

The Fine Adjust parameter gives you one-cent precision in setting the center frequency.



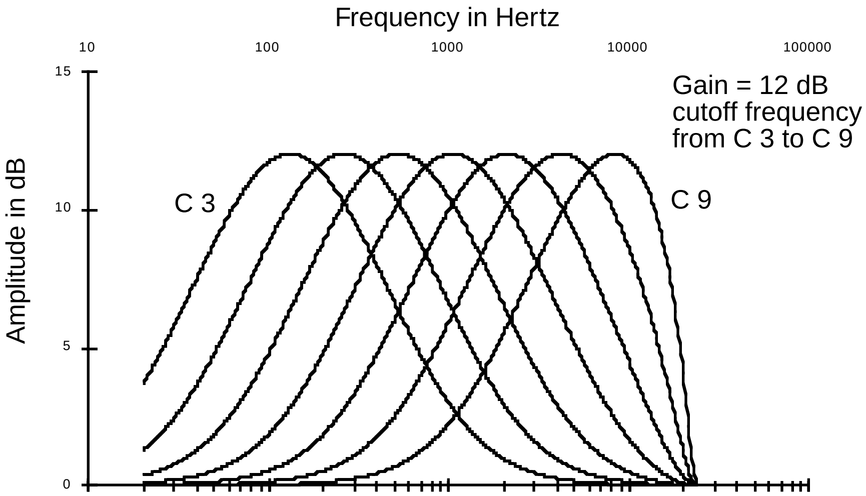
```
EditProg:F2 WID(PARA EQ)     <>Layer:1/1
Adjust:0.010oct       Src1   :OFF
                      Depth  :0.00oct
                      Src2   :OFF
KeyTrk:0.000oct/key   DptCtl:OFF
VelTrk:0.00oct        MinDpt:0.00oct
                      MaxDpt:0.00oct
<more   F1 FRQ  F2 WID  F3 AMP  F4 AMP  more>
```

Parameter	Range of Values
Adjust	0.010 to 5.000 octaves
Key Tracking	± .200 octaves per key
Velocity Tracking	± 5.00 octaves
Source 1	Control Source list
Source 1 Depth	± 5.00 octaves
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 5.00 octaves
Maximum Depth, Source 2	± 5.00 octaves

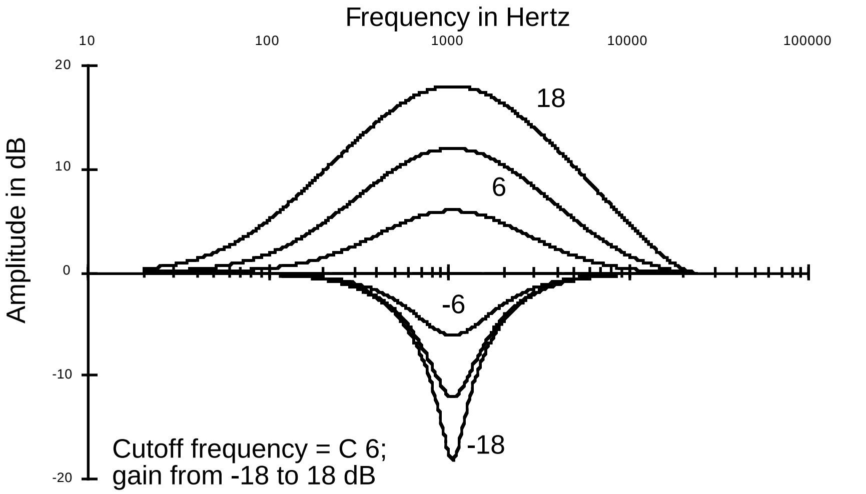
The control-input page for the amplitude stage (F3 AMP) is identical to the AMP page described previously, except that there’s no Pad parameter.

Mid-range Parametric EQ (PARA MID)

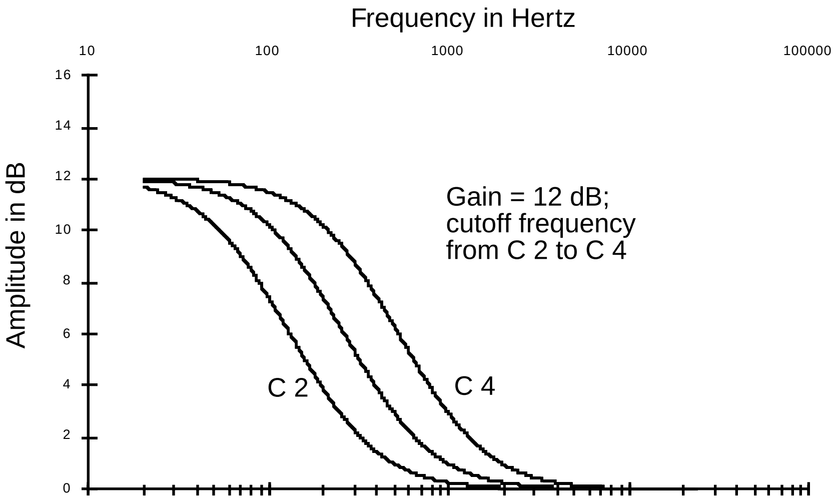
This two-stage function is almost identical to the three-stage Parametric EQ function. The only difference is that the width of PARA MID is fixed at 2.2 octaves. Consequently there’s no control-input page for the width.



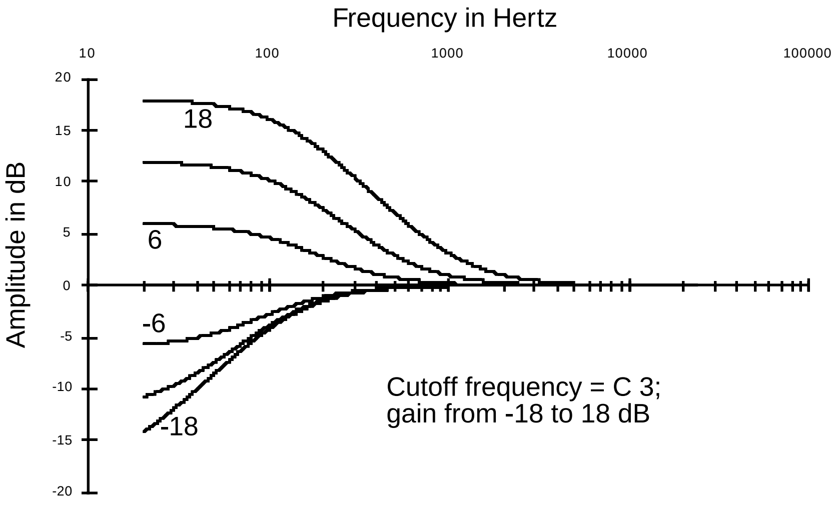
Para Mid: Gain



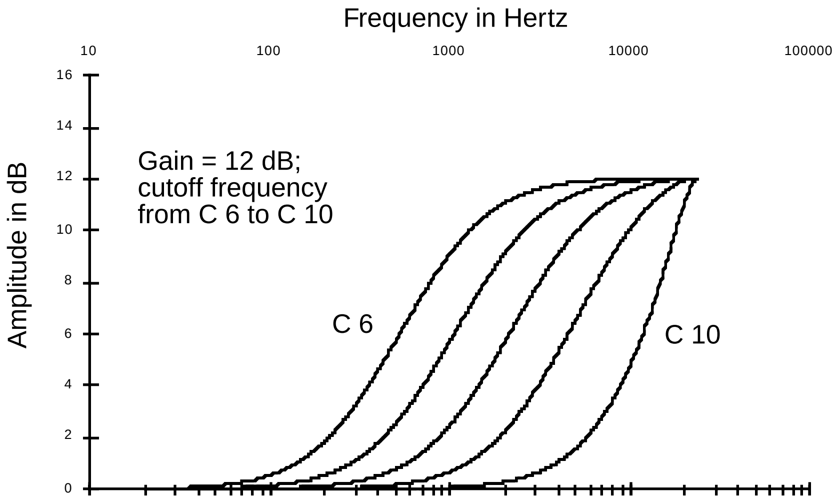
Bass Tone Control (PARA BASS)



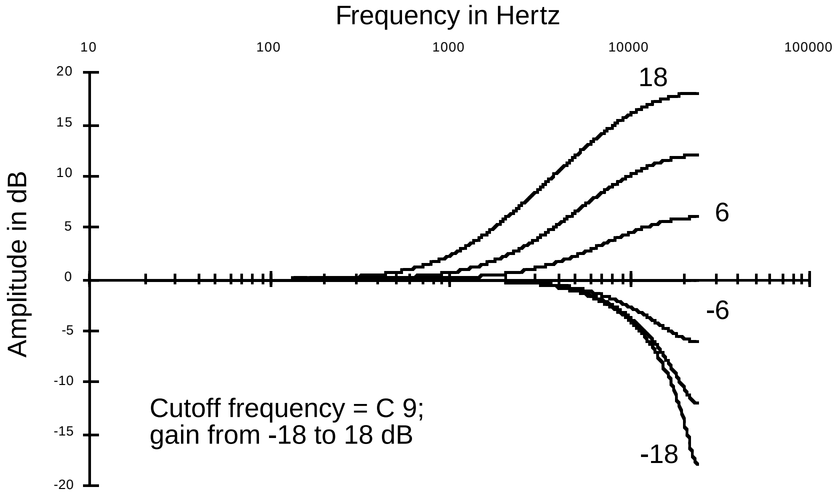
This is a two-stage function, with control-input pages for frequency and amplitude. These pages are the same as those for frequency and amplitude in PARA EQ. On the frequency control-input page, you'll set the cutoff frequency. For notes above this frequency, the amplitude setting has a diminished effect. On the amplitude control-input page, you'll set the amount of cut or boost that's applied to notes below the cutoff frequency. There's a gradual increase in the bass response for each successively lower note. The location of the cutoff frequency will change somewhat as you change the amplitude settings, although the value for the Adjust parameter on the frequency control-input page will not reflect the change.



Treble Tone Control (PARA TREBLE)

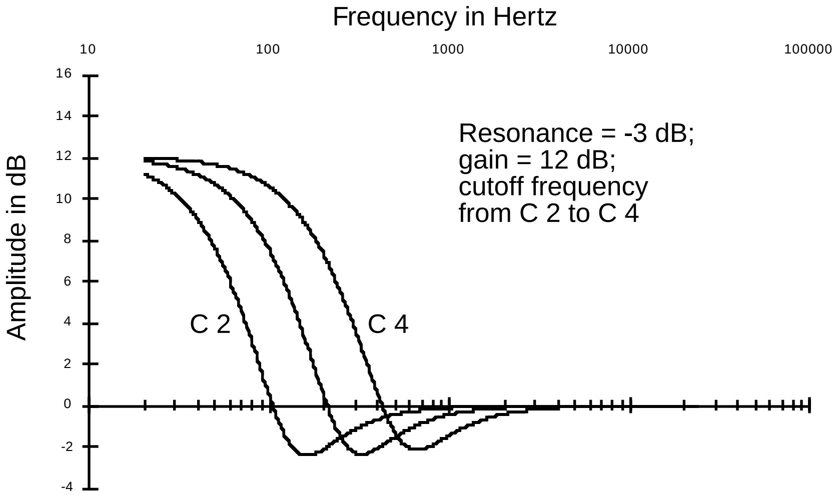


PARA TREBLE is very similar to PARA BASS; the only difference is that the amplitude setting affects notes *above* the cutoff frequency.

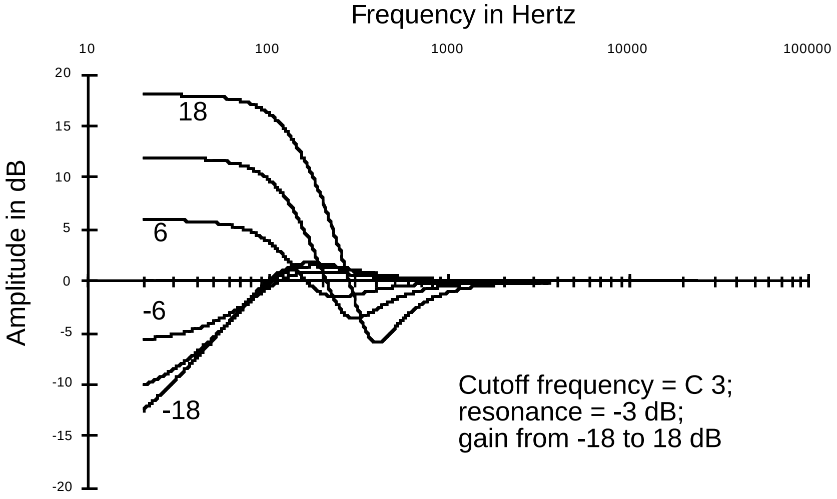


Steep Bass Tone Control (STEEP RESONANT BASS)

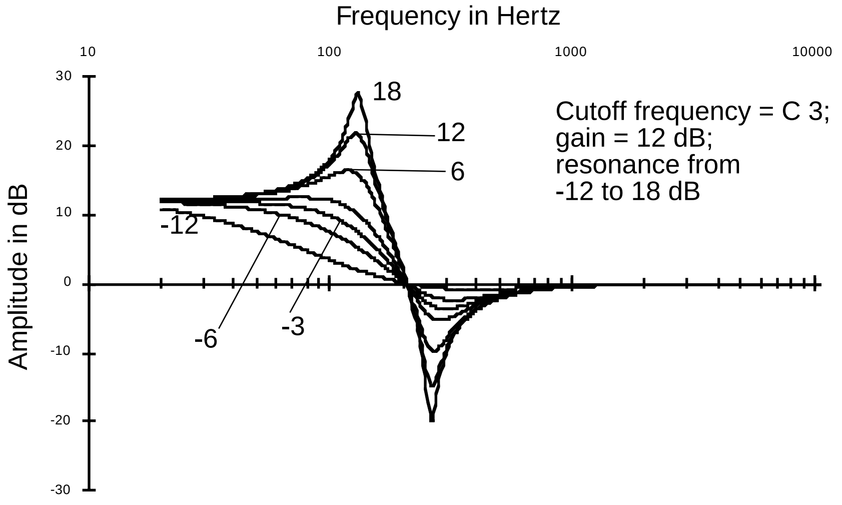
Steep Resonant Bass: Frequency



Steep Resonant Bass: Gain

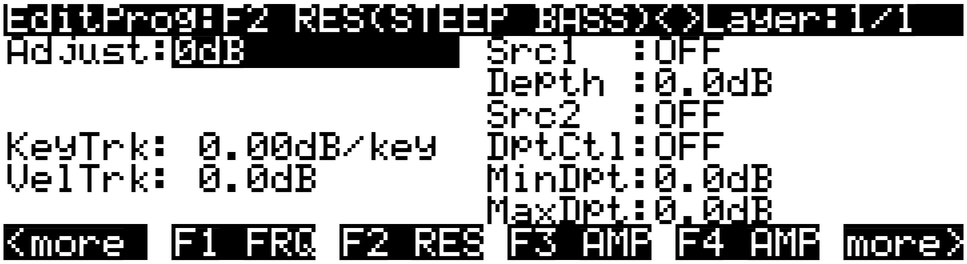


Steep Resonant Bass: Resonance



Cutoff frequency = C 3;
gain = 12 dB;
resonance from
-12 to 18 dB

This function uses a two-pole lowpass filter to give you a sharper transition in bass response than PARA BASS. Like PARA BASS, there are control-input pages for cutoff frequency and amplitude, which are identical to those for PARA BASS. There is also a control-input page for resonance (also known as “q”), which can boost or cut the amplitude of the partials near the cutoff frequency.



```
EditProg:F2 RES(STEEP BASS)<>Layer:1/1
Adjust:0dB            Src1   :OFF
                      Depth  :0.0dB
                      Src2   :OFF
KeyTrk: 0.00dB/key    DptCtl:OFF
VelTrk: 0.0dB         MinDpt:0.0dB
                      MaxDpt:0.0dB
<more  F1 FRQ  F2 RES  F3 AMP  F4 AMP  more>
```

Parameter	Range of Values
Adjust	-12 to 24 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 30 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 30 dB
Maximum Depth, Source 2	± 30 dB

You’ll get the best transition in bass response with a resonance setting of -6 dB. There’s a small frequency range above the cutoff frequency where the response reverses direction (if you’re cutting the amplitude, for example, you’ll get a slight boost just above the cutoff frequency). The higher you set the resonance, the larger this reversal will be, resulting in unusual—but possibly useful—response curves at high resonance values.

Pitch / Amplitude / Panner

PITCH

AMP

PANNER

UPPER AND LOWER AMP

BALANCE AND AMP

GAIN

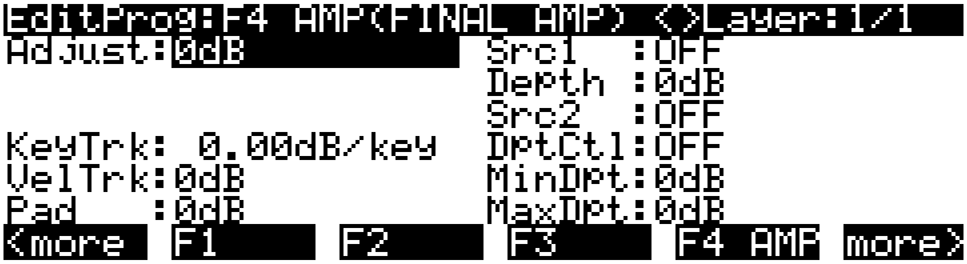
PITCH

We used the PITCH control-input page as an example to introduce the common DSP control parameters in Chapter 6 (*Common DSP Control Parameters* on page 6-14), so we won't add much here. The PITCH function modifies the pitch of the layer's keymap as it passes through the sound engine. The PITCH stage of each algorithm is always the first stage. Algorithms 26–31, the Sync algorithms, don't show the PITCH stage on the ALG page, since these algorithms generate their own sawtooth waves, and do not use keymaps.

AMP

The AMP function is the final stage in every single-output algorithm, and controls the overall amplitude (volume) of the layer. This is an easy way to boost the signal to a more desirable level if it's not loud enough for your purposes. Large values for the Adjust parameter can cause a sound to clip, which will distort most sounds considerably. You may want this effect, and using it won't damage anything, but as a rule, you'll want to avoid clipping your sounds with the AMP (or GAIN) function. There are many other ways to distort your sounds, like DIST, SHAPER, and WRAP, to name a few.

The settings for the parameters on the F4 AMP page affect the gain level for the currently selected layer. So do the settings on the AMPENV page. Compare this with the effect of GAIN, described on page 16-37.



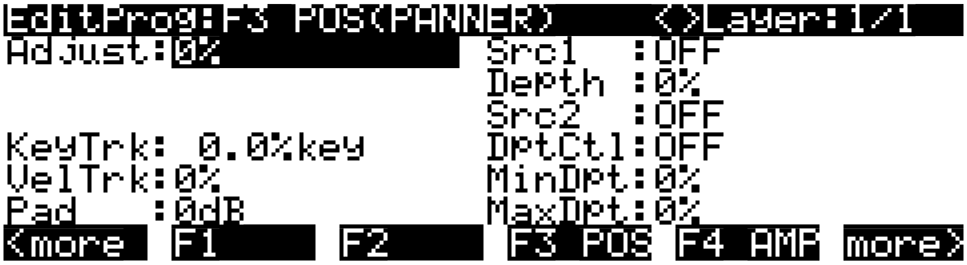
```
EditProg:F4 AMP(FINAL AMP) <>Layer:1/1
Adjust:0dB               Src1  :OFF
                         Depth :0dB
                         Src2  :OFF
KeyTrk: 0.00dB/key       DptCtl:OFF
VelTrk:0dB               MinDpt:0dB
Pad   :0dB               MaxDpt:0dB
<more  F1    F2    F3    F4 AMP  more>
```

Parameter	Range of Values
Adjust	-96 to 48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

PANNER

This single-stage function converts a single wire at its input into a double wire at its output, splitting the signal between an "upper" and "lower" wire. This creates a double-output algorithm, as discussed on page 6-31. The parameters on the PANNER page enable you to modify the signal's routing through the upper and lower wires. By itself the PANNER doesn't change the pan position of the sound. It just defines what percentage of the currently selected layer's sound goes to each wire. When you select one of these double-output algorithms, the OUTPUT page for the layer changes to enable you to make pan settings for each wire independently. So when you use the PANNER function, you'll also want to adjust the Pan parameters on the OUTPUT page, setting the upper wire's pan fully right, and the lower wire's pan fully left. This will enable you to hear the effect of the PANNER function.

The PANNER function is available only in algorithms 2, 13, 24, and 26, and always appears in the block before the final AMP function. Consequently, it will always be selected with the **F3** soft button, which is labeled **F3 POS** (position).



```
EditProg:F3 POS(PANNER)      <>Layer:1/1
Adjust:0%              Src1  :OFF
                       Depth :0%
                       Src2  :OFF
KeyTrk: 0.0%key        DptCtl:OFF
VelTrk:0%              MinDpt:0%
Pad   :0dB             MaxDpt:0%
<more  F1     F2     F3 POS F4 AMP more>
```

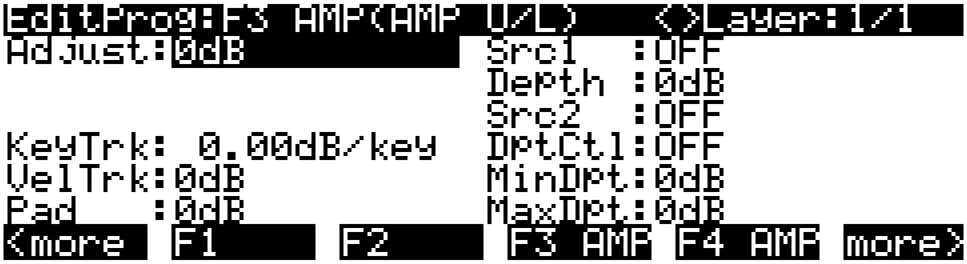
Parameter	Range of Values
Adjust	± 100%
Key Tracking	± 16% per key
Velocity Tracking	± 200%
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 200%
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 200%
Maximum Depth, Source 2	± 200%

The Adjust parameter sets the initial routing of the layer to the upper or lower wire. **-100**% is the lower, and **100**% is the upper. The KeyTrk parameter lets you shift the layer's sound from one wire to the other based on the MIDI key number of each note. For positive values of KeyTrk, the higher above Middle C, the more sound goes to the upper wire.

The remaining parameters have ranges from **-200**% to **200**%. This lets you start with a sound that's fully on the lower wire, for example, and shift it completely to the upper wire. The VelTrk parameter shifts notes between wires based on the attack velocity of each note. For positive values, the higher the attack velocity, the more sound goes to the upper wire. The Src1 and Src2 parameters let you assign controls to reroute the sound relative to the initial routing. Setting their depth parameters to positive values will shift the sound to the upper wire when the controls assigned to them approach their maximum values.

Upper and Lower Amp (AMP U   AMP L)

This two-stage function is similar to the AMP function described above, but it appears in algorithms that have split the signal to two wires and has sent them through different DSP functions in the F2 and F3 blocks. This function enables you to set the final amplitude independently for each wire, and keeps the two signals separate at its output, giving you added flexibility for mixing and panning. Like the AMP function, UPPER AND LOWER AMP always appears as the last block in an algorithm. Since it’s a two-stage function, it has two control-input pages. **F3** selects the control-input page for the lower wire, and **F4** the control-input page for the upper wire.

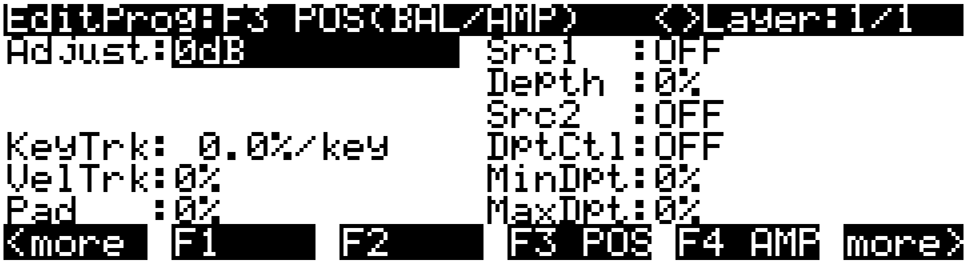


```
EditProg:F3 AMP(AMP U/L)    <>Layer:1/1
Adjust:0dB           Src1  :OFF
                     Depth :0dB
                     Src2  :OFF
KeyTrk: 0.00dB/key   DptCtl:OFF
VelTrk:0dB           MinDpt:0dB
Pad   :0dB           MaxDpt:0dB
<more  F1     F2     F3 AMP  F4 AMP  more>
```

Parameter	Range of Values
Adjust	-96 to 48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

Balance and Amp (BAL AMP)

This function has a two-wire input and a two-wire output. The parameters on its control-input page affect the amount of gain applied to each wire between input and output. A value of **0%** applies equal gain to both the upper and lower wires; at a value of **100%** only the upper wire’s sound will be audible, at **-100%**, only the lower wire’s sound will be audible. This works like the balance control on any stereo system; as the gain increases for one wire, it decreases for the other. It’s also similar to the PANNER and XFADE functions. The **F3** soft button selects the control-input page for the balance stage of this function.



```
EditProg:F3 POS(BAL/AMP)    <>Layer:1/1
Adjust:0dB          Src1  :OFF
                    Depth :0%
                    Src2  :OFF
KeyTrk: 0.0%/key    DptCtl:OFF
VelTrk:0%           MinDpt:0%
Pad   :0%           MaxDpt:0%
<more  F1     F2     F3 POS F4 AMP more>
```

Parameter	Range of Values
Adjust	± 100%
Key Tracking	± 16.00% per key
Velocity Tracking	± 200%
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 200%
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 200%
Maximum Depth, Source 2	± 200%

The AMP stage sets the overall amplitude applied to both wires, and is programmed exactly like the AMP function described above. Their control-input pages are almost identical, including the ranges of values. The only difference is that there’s no Pad parameter for the AMP stage of the BAL AMP function. The **F4** soft button selects the control-input page for the AMP stage.

GAIN

This function, like AMP, can boost or cut the amplitude of the signal as it passes through the algorithm. Unlike AMP, however, the layer’s amplitude envelope doesn’t effect the gain settings. GAIN can be used to introduce clipping into a signal, or for adjusting the amplitude of an added waveform. Use a gain function when you want to boost or cut all of a sound’s partials uniformly. The control-input page for GAIN is identical to the one for AMP.

Mixers

+AMP

+GAIN

CROSSFADE

The functions in the Mixer category combine two-wire signals in various ways. They have double wires at their inputs, and they mix and amplify the signals from the two wires, then combine them for output to a single wire. Depending on where you assign one of these functions, they can be used to combine two-wire signals for the F4 AMP block, or to enable you to apply another DSP function to the combined signals before the F4 AMP block.

There's a Pad parameter on the control-input pages for these functions, which attenuates the lower wire's signal at its input to the function.

+AMP

The two input signals to this function are multiplied by .5 (to reduce the likelihood of clipping), then added together. The resulting signal is then multiplied by a gain factor (the combined values for the parameters on the control-input page), and multiplied by 2. Any clipping that occurs can be eliminated by lowering the value of the Adjust parameter. If the Adjust value is –6 dB or lower, the signal will never clip. The control parameters are affected by the settings and controls on the AMPENV page.

+GAIN

This function operates in almost the same way as +AMP, the only difference being that the signal is not affected by the settings on the AMPENV page, since it occurs before the final AMP block.

Crossfade (XFADE)

This function adds the signals from the upper and lower wires after evaluating the combined values of the parameters on its control-input page. If those values add up to -100%, only the lower wire's signal is sent to the output. If they add up to 100%, only the upper wire's signal is sent to the output. If they add up to 0%, both signals are attenuated 6 dB, then added and sent to the output.

Waveforms

SINE
LOW FREQUENCY SINE
SAWTOOTH
LOW FREQUENCY SAWTOOTH
SQUARE
LOW FREQUENCY SQUARE

In this category of DSP functions are three standard synth waveforms—Sine, Sawtooth, and Square—with high- and low-frequency variations of each. These are all one-stage functions. They can be assigned in several different positions and combinations in many of the algorithms.

One important fact to keep in mind is that assigning one of these waveforms to a layer's algorithm may remove the original sample from the signal, since they don't have input signals to send to their outputs (they send only the waveform that they generate themselves). If, for example, you were editing the Acoustic Piano program, and you assigned SINE in the F1 block, you would no longer hear the piano timbre, only the sine wave (unless the signal splits before the F1 block, as in Algorithm 10). Consequently, you'll tend to use these waveforms when you want to build a sound from scratch. If you want to add a waveform to the original timbre of a sound, use one of the added waveform functions described in the next section, or use one of the split signal algorithms.

These waveforms can range in frequency from .1 Hz to 20 KHz. They're not samples like the instrumental sounds and other waveforms; they're generated by oscillators. Since the DSP function waveforms aren't produced by playing back multi-sample keymaps, there are no sample root transitions as you play notes in different keyboard ranges. This makes them especially suitable for use with portamento and wide pitch bend ranges.

Since these waveform functions generate an output signal only, and don't receive an input signal to pass along, the algorithms are arranged so you won't inadvertently assign a series of waveforms that interfere with each other. You'll usually find, for example, that if you can assign a waveform in the F1 block, all subsequent blocks will allow you to assign only the added waveforms. Or, if the subsequent blocks allow you to assign the "regular" waveforms, it's because the wiring of the algorithm is split so that the two waveforms pass through in parallel (as in Algorithm 10).

This next point is a small one, but important, and may make it easier for you to understand the way the waveform functions operate, especially if you've been carefully studying the wiring paths of the algorithms. In several algorithms where the waveforms are available the wiring paths of the algorithms (the horizontal arrows) appear to send a signal to an input of the waveforms. This is *not* the case, and anywhere one of these waveforms is assigned, you should view the algorithm as if there were no horizontal arrow pointing to the left (input) side of the block where the waveform is assigned. The diagrams below will clarify this point. The only difference in the DSP function assignments is in the F1 block, where the first and second diagrams show the SAW waveform, and the third diagram shows the SAW+ added waveform (described in the next section). In the first diagram, the PITCH function's output (passing the sample signal from the keymap) appears to be connected to the input of the F1 block (the SAW function), as well as splitting and passing to the +GAIN function in the F2 block. This is what you would see on the ALG page.

In fact, the actual signal path does *not* pass from the PITCH function through the SAW function; it splits and bypasses the SAW function, as shown in the second diagram. The third diagram shows the same algorithm with the SAW+ added waveform assigned to the F1 block. In this case, the diagram is accurate; the signal passes from the output of the PITCH function, and splits into a two-wire signal. The upper wire passes through the F1 block where the sawtooth wave is added, and into the +GAIN function in the F2 block. The lower wire bypasses the F1 block, and passes directly to the F2 block, where it is combined with the upper wire signal.

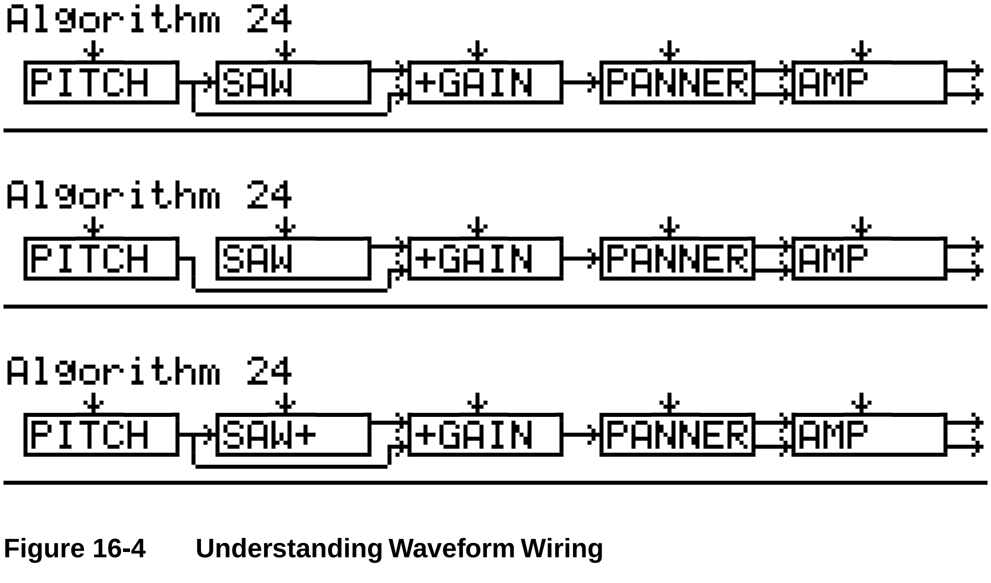
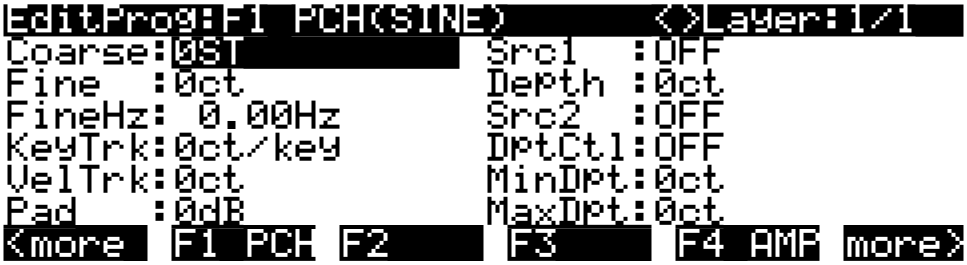


Figure 16-4 Understanding Waveform Wiring

The six waveforms in this category are Sine, Sawtooth, Square, Low Frequency Sine, Low Frequency Sawtooth, and Low Frequency Square. The control-input pages for all six waveforms affect the pitch of the waveforms. The control-input pages for the first three waveforms are identical, as are the control-input pages for the three low frequency waveforms are identical.

SINE, Sawtooth (SAW), SQUARE

There’s only one parameter on this control-input page that may still be unfamiliar to you: Fine Hz. This is discussed on page 6-27. It can tune the pitch of the waveform in terms of its actual frequency in Hertz, as opposed to the usual method of tuning by key names. The advantage to using the Fine Hz parameter is that you can maintain constant beat frequencies across much of the keyboard when you have a program with slightly detuned multiple layers (or multiple waveforms in one layer).



```
EditProg:F1 PCH(SINE)         <>Layer:1/1
Coarse:0ST               Src1   :OFF
Fine   :0ct              Depth  :0ct
FineHz: 0.00Hz           Src2   :OFF
KeyTrk:0ct/key           DptCtl:OFF
VelTrk:0ct               MinDpt:0ct
Pad    :0dB              MaxDpt:0ct
<more   F1 PCH   F2       F3       F4 AMP   more>
```

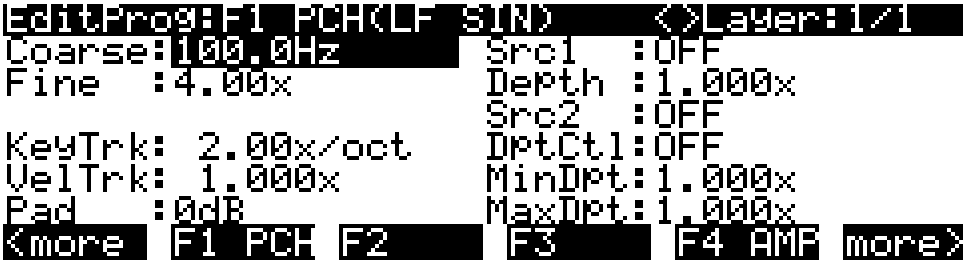
Parameter	Range of Values
Coarse Adjust	–120 to 60 semitones
Fine Adjust	± 100 cents
Fine Adjust in Hertz	± 6.00 Hertz
Key Tracking	± 2400 cents per key (2 octaves)
Velocity Tracking	± 7200 cents (6 octaves)
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 7200 cents
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 7200 cents
Maximum Depth, Source 2	± 7200 cents

If you want the waveform to play with standard twelve-tone octaves, set the KeyTrk parameter to **100 cents per key**. Different values for KeyTrk will result in nonstandard tunings.

Low Frequency Waveforms: Sine (LF SIN), Sawtooth (LF SAW), Square (LF SQR)

These can be used like the waveforms above, since their frequency ranges are similar, but they’re intended to be used not for their timbres, but for the shape of their waveforms. By using low frequency values for these waveforms, you’re basically getting extra LFOs with very precise control parameters. They’re intended to be used as inputs to drive the DSP functions in the subsequent algorithm blocks. They’re especially useful with the nonlinear DSP functions, such as xAMP.

The parameters on this page affect the pitch of the low frequency waveform in a slightly different manner. They’re all tied to the value of the Coarse Adjust parameter, so when you’re working with this page, you’ll want to set the Coarse Adjust first, then set the values of the other parameters to modify the initial setting. The Coarse Adjust value is multiplied by the values of the other parameters to determine the effect on the pitch, as indicated by the “x” after the parameters’ values. More parameter descriptions follow below.



```
EditProg:F1 PCH(LF SIN)     <>Layer:1/1
Coarse:100.0Hz         Src1  :OFF
Fine  :4.00x           Depth :1.000x
                       Src2  :OFF
KeyTrk: 2.00x/oct      DptCtl:OFF
VelTrk: 1.000x         MinDpt:1.000x
Pad   :0dB             MaxDpt:1.000x
<more   F1 PCH   F2      F3      F4 AMP   more>
```

Parameter	Range of Values
Coarse Adjust	0.1, 1.0, 10.0, 100.0, 1000.0 Hertz
Fine Adjust	1.00 to 20.00 x
Key Tracking	0.1 to 10.0 x per octave
Velocity Tracking	0.010 to 32.000 x
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	0.010 to 32.000 x
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	0.010 to 32.000 x
Maximum Depth, Source 2	0.010 to 32.000 x

Coarse Adjust (Coarse)

There are only five values to choose from. They’re expressed in terms of their frequencies in Hertz. Each value has a frequency ten times higher or lower than adjacent values.

Fine Adjust (Fine)

Set this parameter’s value to **1.00 x** if you don’t want it to affect the Coarse Adjust setting. Doubling this value (**2.00 x**, **4.00 x**, etc.) will raise the pitch in octaves. By using Fine Adjust in tandem with Coarse Adjust, you can achieve frequencies from .1 Hz (well below the audible range) to 20 KHz.

Key Tracking (KeyTrk)

A value of **1.00 x per octave** will keep the waveform’s pitch uniform across the entire keyboard. A value of **2.00 x per octave** will give you the normal twelve-tone octave. Other values will give you nonstandard tunings.

Added Waveforms

SINE+

SAW+

NOISE+

There are three DSP functions that add waveforms to a layer’s existing sample: SINE+, SAW+, and NOISE+.

The parameters on the control-input page for the SINE+ function affect the pitch of the sine waveform without affecting the pitch of the existing sample. The control-input page for the SINE+ function is similar to those for the regular waveforms above. There are parameters for coarse adjust, key tracking, velocity tracking, Source 1 and Source 2, and a pad. There are also parameters for fine adjust and fine Hertz adjust.

The SAW+ function is virtually identical to the SINE+ function; the only difference is in the shape of the waveform.

The NOISE+ function is tied to the level of the sample to which it’s added. It will generate fairly white noise (that is, nearly equal amplitude at all audible frequencies) as long as the amplitude of the sample is nonzero. The amplitude of the noise is multiplied by its gain control (the Adjust parameter on its control-input page), then added to the signal. To add a short burst of noise at the beginning of a sound, assign ENV2 as the value of one of the Source parameters, then edit ENV2 to produce an envelope with a rapid decay.

The control parameters for NOISE+ are similar to those for SINE+ and SAW+, except that there are no parameters for fine adjust or fine Hertz adjust.

Nonlinear Functions

High Frequency Stimulator

Distortion

Shaper

Double Shaper

Two-parameter Shaper

Wrap

Lowpass Filter with Clipping

Pulse Width Modulation

The functions in this category have a variety of effects on the signal. What they have in common is that they can add partials to the signal that were not present at their inputs.

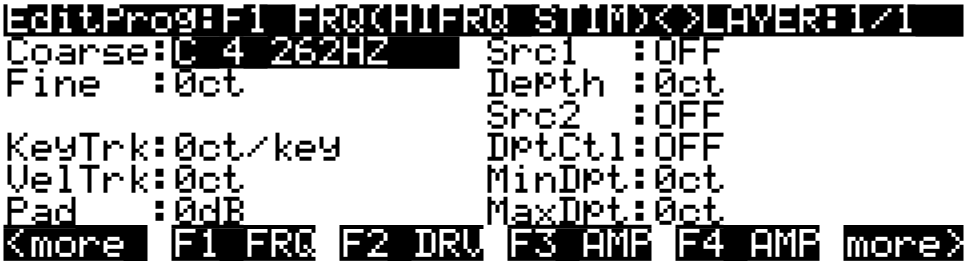
The nonlinear functions can produce dramatic changes in timbre, resulting in all sorts of new and modified sounds. One thing to keep in mind is that sounds with a large number of high-frequency partials can be subject to distortion at the high end of the keyboard, especially when you’re using more than one of the nonlinear DSP functions. You might also hear a bit of ***aliasing*** with some sounds. Aliasing refers to unintended partials that occur below the fundamental pitch of a sound. The easiest way to remove this distortion or aliasing is to reduce the level of the Adjust parameter on the control-input page for whichever nonlinear DSP functions you’re working with. When you’re using PWM followed by DIST or SHAPER, you’d reduce the level of the Adjust parameter for the DIST or SHAPER function. You can also use key tracking (KeyTrk, usually with a negative value), and key tracking in combination with the Keytrack Start (KStart) parameter that’s described on page 16-4.

Even with the damping effects of KeyTrk and KStart, you may come up with sounds that are fantastic in the low range, but gritty in the high range. You can transpose the keymap down to counteract this, but that’s the nature of the nonlinear functions. In extreme cases, you can lower the HiKey of the layer to disable the high end completely.

High Frequency Stimulator (HIFREQ STIMULATOR)

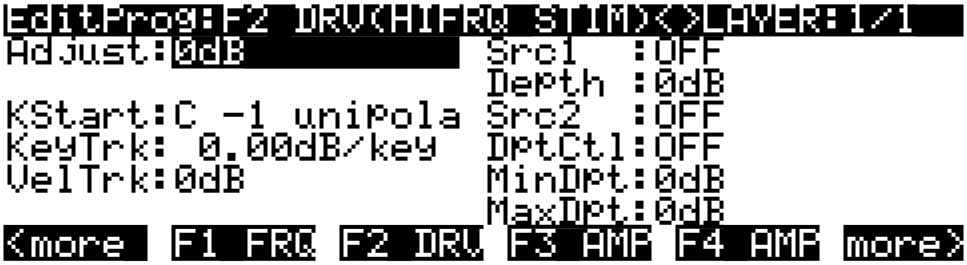
The overall effect of this three-stage function is to boost the high frequency partials of the signal, and depending on the settings of the control inputs, it can add high-frequency partials to the signal as well. It’s useful for building sounds that cut through the mix, and have a bright crisp nature.

There’s more to the High Frequency Stimulator than meets the eye. It works like this: the signal is run through a high-pass filter, then through a distortion function, then through a second high-pass filter. Finally, it’s mixed with the original signal after passing through the final AMP stage of the algorithm. The three control-input pages let you adjust the cutoff frequency of the first high-pass filter (F1 FRQ), the amount (drive) of the distortion function (F2 DRV), and the mix (relative amplitude) of the stimulated signal with the original (F3 AMP).



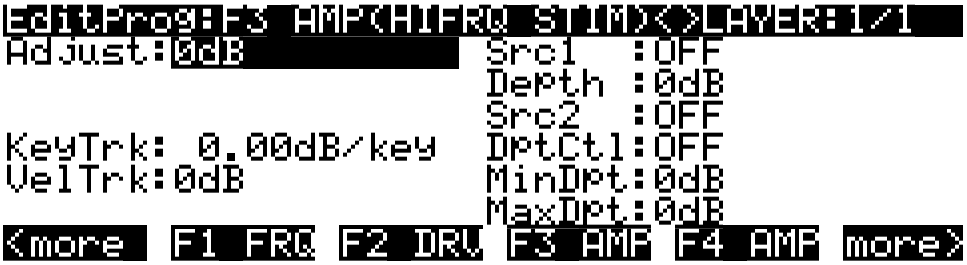
```
EditProg:F1 FRQ(HIFRQ STIM)<>LAYER:1/1
Coarse:C 4 262HZ      Src1   :OFF
Fine  :0ct            Depth :0ct
                      Src2   :OFF
KeyTrk:0ct/key        DptCtl:OFF
VelTrk:0ct            MinDpt:0ct
Pad   :0dB            MaxDpt:0ct
<more  F1 FRQ  F2 DRV  F3 AMP  F4 AMP  more>
```

Parameter	Range of Values
Coarse Adjust	C -1 16 Hz to G 10 25088 Hz
Fine Adjust	± 100 cents
KEY Tracking	± 250 cents per key
Velocity Tracking	± 10800 cents
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 10800 cents
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 10800 cents
Maximum Depth, Source 2	± 10800 cents



```
EditProg:F2 DRV(HIFRQ STIM)< >LAYER:1/1
Adjust:0dB           Src1  :OFF
                     Depth :0dB
KStart:C -1 unipola  Src2  :OFF
KeyTrk: 0.00dB/key   DptCtl:OFF
VelTrk:0dB           MinDpt:0dB
                     MaxDpt:0dB
<more  F1 FRQ  F2 DRV  F3 AMP  F4 AMP  more>
```

Parameter	Range of Values
Adjust	–96 to 48 dB
Keytrack Start	C -1 to C 9 unipolar, C -1 to C 9 bipolar
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

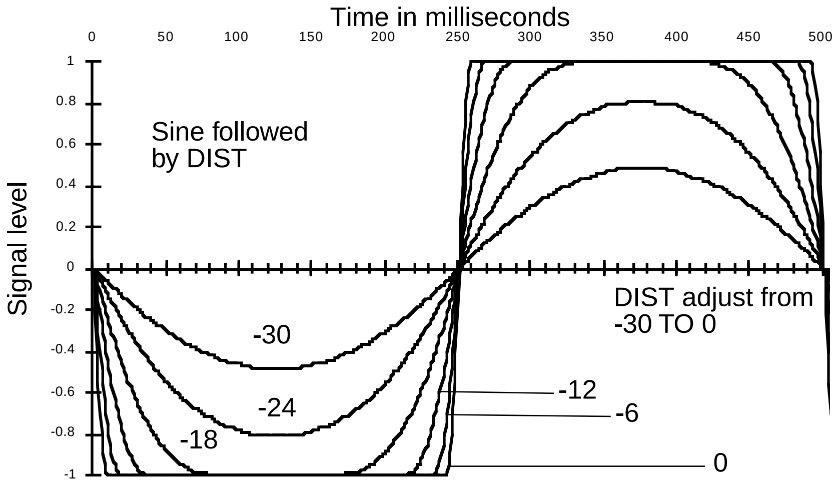


```
EditProg:F3 AMP(HIFRQ STIM)< >LAYER:1/1
Adjust:0dB           Src1  :OFF
                     Depth :0dB
                     Src2  :OFF
KeyTrk: 0.00dB/key   DptCtl:OFF
VelTrk:0dB           MinDpt:0dB
                     MaxDpt:0dB
<more  F1 FRQ  F2 DRV  F3 AMP  F4 AMP  more>
```

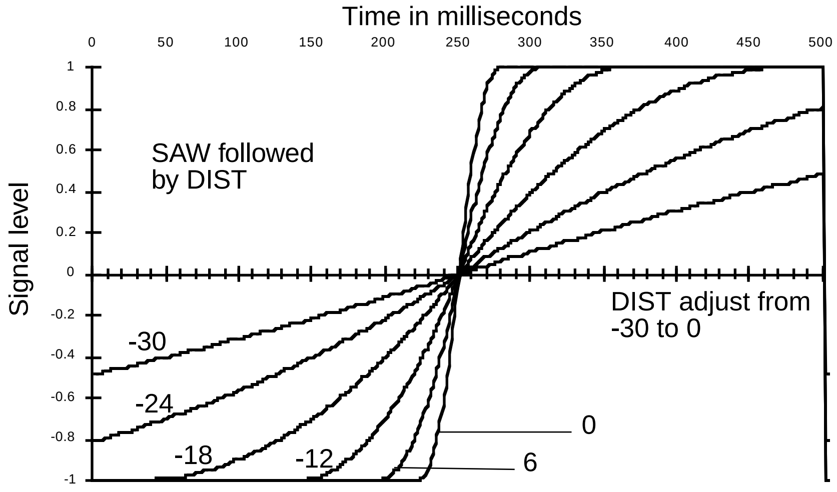
Parameter	Range of Values
Adjust	–96 to 48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Source 1	Control Source list
Source 1 DEPTH	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

Distortion (DIST)

Distorted Sine wave



Distorted Sawtooth wave

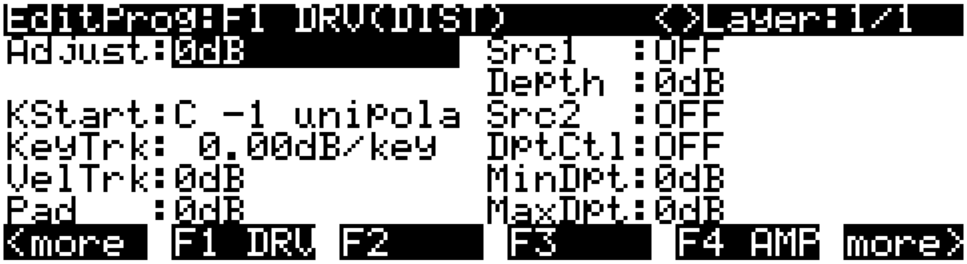


Using this function is much like overdriving an ordinary keyboard or guitar amplifier. The input signal is multiplied by the gain control (the combined values of the parameters on its control-input page labeled DRV, for Drive), then passes into a distortion mapper. Large Adjust values will cause serious amounts of distortion.

Different sounds are affected differently by DIST. Waveforms that are static (waveforms with shapes that repeat regularly and are not evolving) when they enter the DIST function will undergo more of a timbre change than the familiar sound of distortion.

The DIST function distorts each note separately, unlike a fuzz box, which adds several notes together then applies a uniform amount of distortion to all of them. Consequently your power chords may sound a little different from your expectations, but you can also get some great effects with key and velocity tracking (not to mention Sources 1 and 2!) that aren’t possible with other distortion devices.

The page below shows the DIST function in the F1 block, but it can appear in other blocks as well.



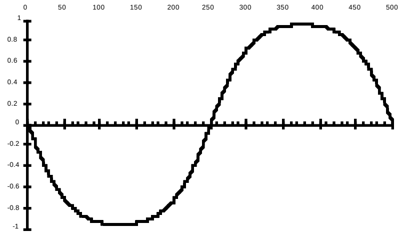
```
EditProg:F1 DRV(DIST)         <>Layer:1/1
Adjust:0dB          Src1  :OFF
                    Depth :0dB
KStart:C -1 unipola Src2  :OFF
KeyTrk: 0.00dB/key  DptCtl:OFF
VelTrk:0dB          MinDpt:0dB
Pad   :0dB          MaxDpt:0dB
<more   F1 DRV  F2      F3      F4 AMP  more>
```

Parameter	Range of Values
Adjust	–96 to 48 dB
Keytrack Start	C -1 to C 9 unipolar, C -1 to C 9 bipolar
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
PAD	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

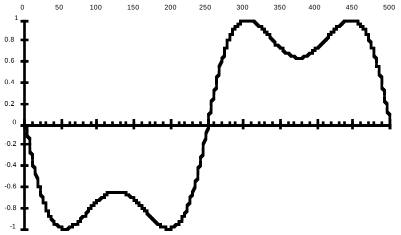
SHAPER

The effect of SHAPER can be very unpredictable, and the mechanics of its operation lend themselves toward explanations that are more numerical than verbal. The best way for you to get a feel for the SHAPER is to start with single-cycle waveform keymaps and experiment with different values for the parameters on its control-input page (labeled AMT, for Amount), and listen to the results. SHAPER tends to work best with the single-cycle waveform sounds (keymaps with IDs 112—166), and is usually less effective with acoustic instrumental sounds. SHAPER often produces numerous peaks throughout the frequency range, even at frequencies that didn't have much amplitude to begin with. These peaks can sound like resonant filters, and can even sound voice-like.

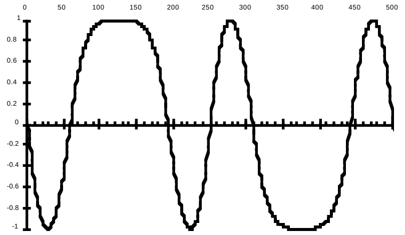
The two series of graphs that follow show the effect of SHAPER on two typical single-cycle waveforms. The first set of six graphs just below shows the evolution of a sine wave input as the value of the Amount parameter is increased. The following set of six graphs shows the effect when the Adjust parameter is increased. Each graph plots a 500-millisecond segment of waveforms cycling at frequencies of 2 Hz. Of course, these are just a few of the countless modulations you can apply to different waveforms at different frequencies.



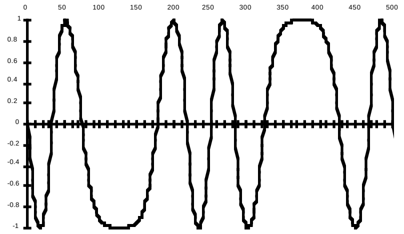
Amount = .1



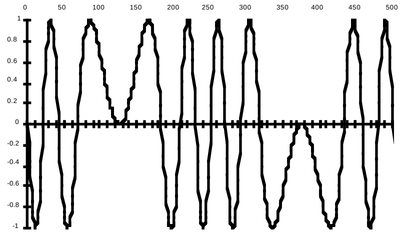
Amount = .2



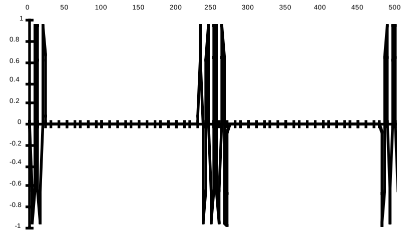
Amount = .375



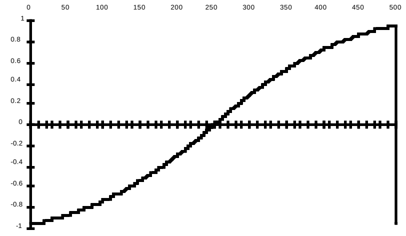
Amount = .625



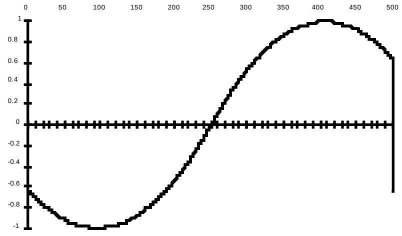
Amount = .1



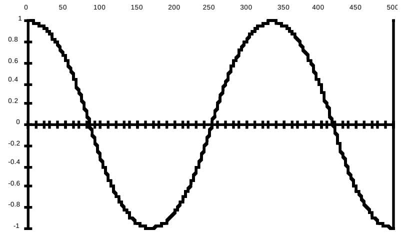
Amount = 4



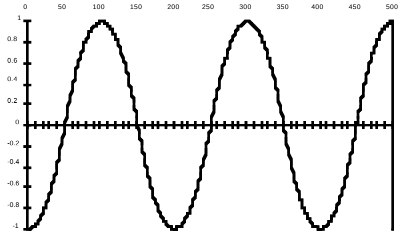
Adjust = .1



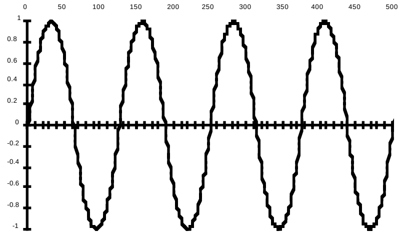
Adjust = .2



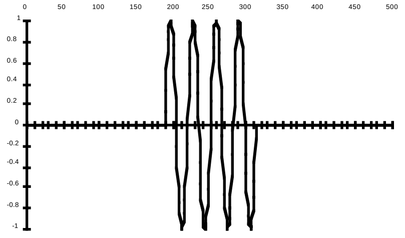
Adjust = .375



Adjust = .625



Adjust = 1

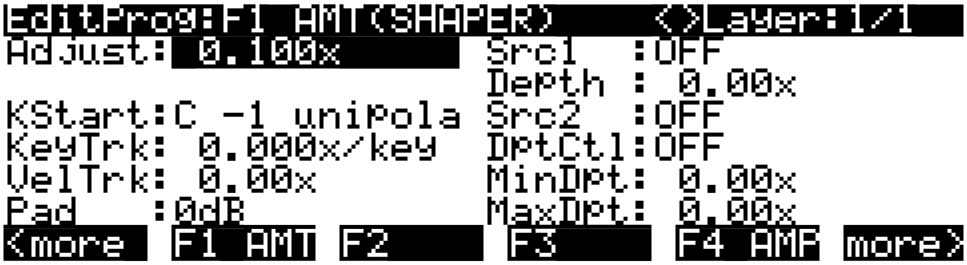


Adjust = 4

As the SHAPER receives input signals, it evaluates the signal's level according to its own internal scale. When the SHAPER's Adjust value is at **.25**, an input signal moving from negative full scale to positive full scale (a sawtooth) will map to an output curve with a single-cycle sine wave shape. At an adjust value of **.5**, the same input signal would map to a 2-cycle sine wave output signal. Adjust values of **.75** and **1.0** for the SHAPER would map to 3-and 4-cycle sine wave output signals, respectively. Beyond values of **1.0**, some portions of the output will pin at zero-scale.

Small Adjust values for the SHAPER can sound much like the DIST function, but larger values will introduce dramatic changes in timbre, while DIST will have a less pronounced effect on timbre.

The following F1 page shows the SHAPER function in the F1 block, but it can appear in other blocks as well.



```
EditProg:F1 AMT(SHAPER)      <>Layer:1/1
Adjust: 0.100x      Src1  :OFF
                    Depth : 0.00x
KStart:C -1 unipola Src2  :OFF
KeyTrk: 0.000x/key  DptCtl:OFF
VelTrk: 0.00x       MinDpt: 0.00x
Pad    :0dB         MaxDpt: 0.00x
<more  F1 AMT F2     F3     F4 AMP more>
```

Parameter	Range of Values
Adjust	0.100 x to 4.000 x
Keytrack Start	C -1 to C 9 unipolar, C -1 to C 9 bipolar
Key Tracking	± 0.200 x
Velocity Tracking	± 4.00 x
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 4.00 x
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 4.00 x
Maximum Depth, Source 2	± 4.00 x

The values for each of the parameters on the SHAPER's control-input page are expressed in arbitrary quantities that represent a multiplication of the amount of shaping applied.

Double Shaper (SHAPE2)

This is simply a series of two SHAPERs. The first is programmed exactly like SHAPER. The values of the control parameters for the second are fixed at .75 times those of the first. This can produce effects that the single SHAPER can't. If, for example, you set the Adjust parameter of SHAPE2 to **1.000**, it will process the input signal with a value of **1.000**, then again with a value of **.75**. This is not the same as processing the input signal once with an Adjust value of **1.75**.

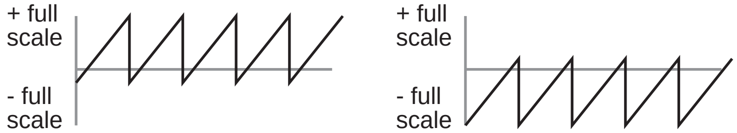
Two-parameter Shaper (2PARAM SHAPER)

This function is similar to the SHAPERs described above, but it has two control-input pages instead of one. The F1 EVN control parameters enable you to add distortion to sine wave partials that are even harmonics of the input signal, and the F2 ODD control parameters let you add distortion to sine wave partials that are odd harmonics of the input signal.

In simpler terms, the control parameters behave like those of the regular SHAPER, but they can shape the signal about six times more than SHAPER can. 2PARAM SHAPER works by separately multiplying the input signal by the combined values of the two sets of control parameters, adding the resulting signals, and multiplying that sum by a constant, then wrapping the signal values that exceed positive or negative full scale (see the WRAP function below).

Experimentation is the key here. Start with very low values for each of the Adjust parameters, and increase them until you begin to hear an effect. Some values will create a DC offset (see diagram below) in the signal—that is, the signal won’t oscillate around the normal zero-point of the scale, but will be shifted toward positive or negative full scale. This may cause a click or thump in sounds with rapid attacks, decays, or releases. To reduce the click or thump, you can edit the AMPENV to produce a more gradual envelope.

2PARAM SHAPER works best with the single-cycle waveform keymaps (IDs 112—166).



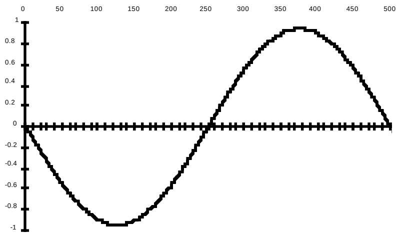
SAWTOOTH WAVE

SAME WAVEFORM WITH NEGATIVE DC OFFSET

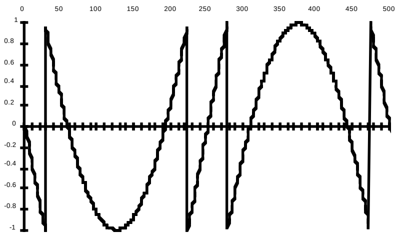
Figure 16-5 DC Offset

Waveform Wraparound (WRAP)

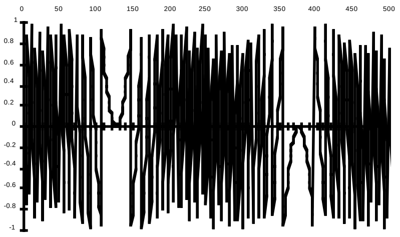
The next three graphs show the effect of various amounts of WRAP on a sine wave.



Adjust = -30

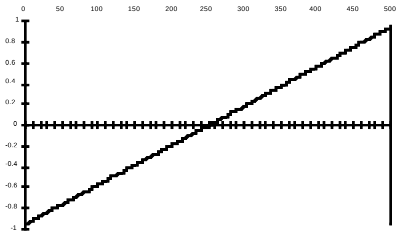


Adjust = -20

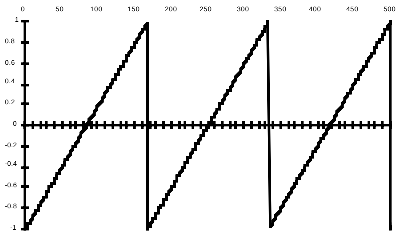


Adjust = 0

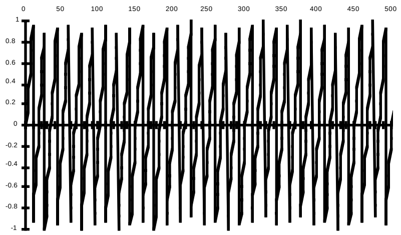
The following three graphs show the effect of WRAP on a sawtooth wave at the same frequency.



Adjust = -30



Adjust = -20



Adjust = 0

With this function you can completely mutilate a sound, and with large amounts of wrap, turn anything into white noise. At the input of the WRAP function, the signal is multiplied by the combined values of the parameters on the WRAP control-input page, then multiplied by an additional gain factor of 30. If the resulting value is greater than full scale (in other words, if it's sufficiently high to clip), then instead of clipping, the waveform "wraps" back around to negative full scale, and it continues to evolve from that point. Likewise, if the resulting value is less than negative full scale, it wraps to positive full scale and proceeds from there. For any waveform, several of these wraparounds can take place before the waveform fits into the allowable range.

You'll want to try different values of the Adjust parameter to get a feel for the results of different amounts of wraparound. Look for the value that introduces a very slight amount (it will tend to be well below 0). The sound will start to buzz here and there, as a few segments of the input wrap around. As you increase the Adjust value, the buzz will increase, and the pitch of the sound will begin to disappear. Keep adding to the Adjust value, and you'll end up with white noise, regardless of the starting timbre.

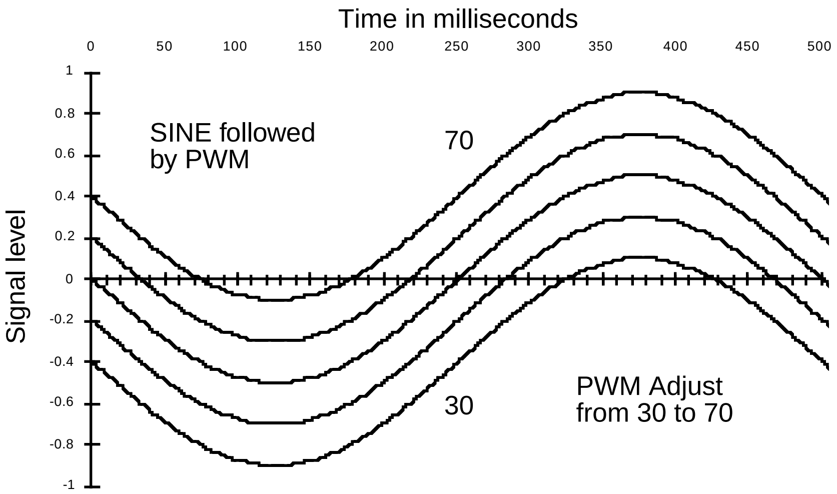
The bright buzzy nature of wrapped sounds is due to the discontinuities in the waveforms of the partials as they wrap around from positive full scale to negative full scale and vice versa. If you want, you can reduce or eliminate the buzz by sending the signal through SHAPER after it goes through WRAP (assign SHAPER as the DSP function in the following algorithm block). Set the SHAPER's Adjust parameter to **.25**. This will map both positive full scale and negative full scale amplitudes to a level of zero, eliminating the wraparound discontinuities, but preserving the alterations in the waveform produced by WRAP. SHAPER will add its own effects as well.

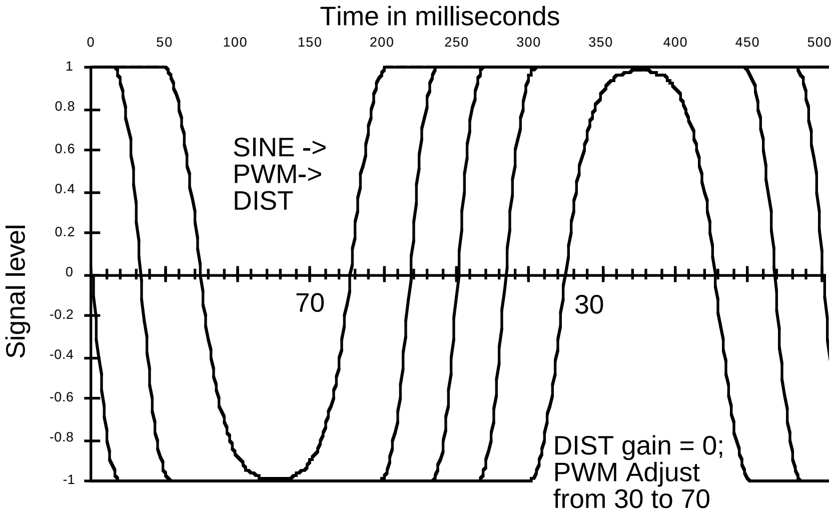
The control-input page for WRAP uses the same set of parameters and ranges of values as DIST.

Lowpass Filter with Clipping (LPCLIP)

This is a one-pole filter, which is programmed just like LOPASS. The difference with LPCLIP is that the amplitude of the input signal is multiplied by 4 before the filter. This can cause the signal to clip, which can produce interesting results. Naturally, you'd use this function only when you want to induce clipping intentionally as a component of your sound.

Pulse Width Modulation (PWM)





Pulse width modulation can produce some classic synth sounds, and can break new sonic ground as well. Its operation is very simple; it adds an oscillating DC offset to the input signal (shifts it toward positive or negative full scale). Unless it causes the signal to clip, it won't have much effect on the signal. It's designed to be followed by DIST, with the Adjust parameter for DIST set to a fairly high level. The DIST function will drive all positive elements of the signal toward positive full scale, and negative elements toward negative full scale. The result is a rectangular wave with a width that varies according to the Adjust level of PWM.

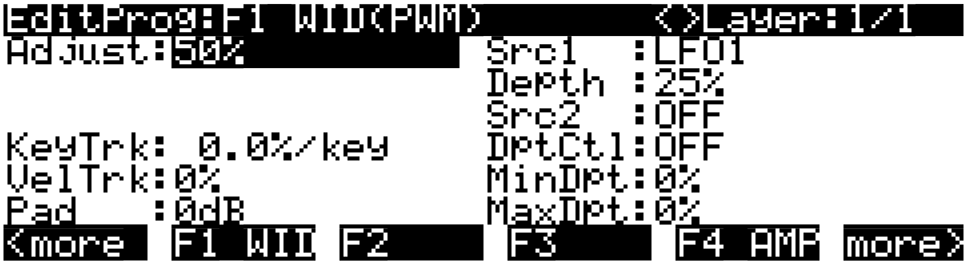
PWM won't affect a square wave input. Sine and triangle waves will produce familiar PWM sounds. More complicated waveforms will result in discontinuous rectangular waveforms.

You can also follow a PWM algorithm block with SHAPER, since SHAPER's output is affected by the DC level of the signal.

The parameters on the PWM control-input page affect the DC offset of the signal, in terms of the percentage of shift from no offset to maximum offset. At a value of **0%**, there is an offset positive full scale. At **100%**, the offset is negative full scale, and at **50%**, there is no offset.

A typical control configuration for PWM has the Coarse Adjust set to **50%**, an LFO assigned to Src1, and Src1's Depth parameter set to a value of **25%**.

The page below shows the PWM function in the F1 block, but it can appear in other blocks as well.



```
EditProg:F1 WII(PWM)          <>Layer:1/1
Adjust:50%          Src1   :LFO1
                    Depth  :25%
                    Src2   :OFF
KeyTrk: 0.0%/key    DptCtl:OFF
VelTrk:0%           MinDpt:0%
Pad    :0dB         MaxDpt:0%
<more  F1 WII  F2     F3     F4 AMP  more>
```

Parameter	Range of Values
Adjust	0 to 100%
Key Tracking	± 8.00% per key
Velocity Tracking	± 100%
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 100%
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 100%
Maximum Depth, Source 2	± 100%

Waveforms Combined with Nonlinear Functions

Added Sawtooth with Nonlinearity (SW+DST)

Added Sawtooth plus Shaper (SW+SHP)

Shape-modulated Oscillator

x Shape Mod Osc

+ Shape Mod Osc

Amp-modulated Oscillator

The six functions in this category do one of two things: they combine samples and waveforms with a nonlinear DSP function, or they use a waveform or sample as inputs to drive nonlinear functions.

Added Sawtooth Wave with Nonlinearity (SW+DST)

This function starts by adding a sawtooth wave to the layer's sample input. When the input signal from the sample is added to the sawtooth, the signal may exceed full scale, so a wraparound function similar to WRAP is performed. The result is then squared to remove any discontinuities from the wraparound. The resulting signal has a large DC offset, so a constant of 3/8 is subtracted.

The parameters on the control-input page for SW+DST control the pitch of the sawtooth wave.

Added Sawtooth Wave Plus SHAPER (SW+SHP)

For this function, the sample input is combined with a sawtooth wave, then passed into the SHAPER function. The SHAPER has a constant Adjust value of .25. First, the sample input is multiplied by a constant, which may cause it to clip. Any clipping becomes part of the signal. This result is added to the sawtooth wave, which may cause the waveform to exceed full scale. If it does, the signal wraps around as in the WRAP function. This result is then put through the SHAPER.

You may want to use the Pad parameter with this function, depending on the sample you use. Pastorius fans should try this function using one of the Electric Bass keymaps. Use Algorithm 8, and start with the second and third algorithm blocks set to **NONE**. Set the fourth block to **SW+SHP**. Try setting the keymap transposition to **-12**, and the SW+SHP key tracking to **100 cents per key**.

Shape-modulated Oscillator (SHAPE MOD OSC)

This function combines the sample input with a sine wave at 1/4 scale, does a combination of additions and multiplications on both signals, then puts the result through a SHAPER function. The amount of shaping depends on the levels of both the input signals. First, the SINE value is multiplied by the sample input value, then multiplied by a constant—any samples exceeding full scale will wrap around. The result is added to the wrapped product of the SINE value times a constant. The entire resulting waveform is then passed through the SHAPER, whose Adjust value is set by the level of the sample input. You might think of this function as an oscillator whose shape is controlled by the sample input signal.

Even if the depth value goes to zero, there will still be a nonzero final output, due to the addition of the multiplied SINE value. In this case, the output would be a slightly distorted sine wave. As the value of depth increases, the harmonic content of the output signal will rise, assuming the pitches of the sample input and the sine wave are simply related—unison, octaves, etc. Slight detuning between the two pitches will cause a slow beat frequency.

The parameters on the F2 PCH control-input page affect the pitch of the sine wave. Those on the F3 DEP page affect the level of the sample input, and consequently the amount of SHAPER that's applied. If the DEPTH values exceed +5 dB, then the product of SAMPLE INPUT x DEPTH may clip, adding further harmonics through a mechanism that differs from the addition of harmonics below the +5 dB DEPTH level.

x SHAPE MOD OSC

Available only in Algorithm 18, this function is similar to SHAPE MOD OSC, except that it multiplies its two input signals and uses that result as its input.

+ SHAPE MOD OSC

Also available only in Algorithm 18, + SHAPE MOD OSC is similar to x SHAPE MOD OSC, except that it adds its two input signals and uses that sum as its input. With this and all the modulated oscillators, let your ears be your guide.

Amp Modulated Oscillator / Final Amp (AMP MOD OSC)

This function is available only in Algorithm 17. The sample input is multiplied by the output of a sine wave oscillator. This result is scaled by the parameters on the F3 DEP control-input page, and the result is added to the original sample input and sent to the final AMP function. The parameters on the F2 PCH control-input page affect the pitch of the sine wave, and consequently all subsequent results.

Mixers with Nonlinear Inputs

x AMP

x GAIN

! AMP

Amplitude Modulation

x AMP

This function can be used in the final algorithm block when it mixes two input wires into a single output. The two input signals are multiplied. The control-input parameters affect the gain of the multiplied signals. The final amplitude is also affected by the settings for AMPENV and ENVCTL. Multiplying the two signals can result in outputs that differ dramatically from the input signals. You can get a wide variety of effects from this function, for example, turning an acoustic sample and a waveform (or two waveforms) into a sound that has little resemblance to the input sounds.

When two signals are multiplied, the resulting signal consists of the sums and differences of the frequencies of each partial of each signal. The frequencies of the original signals do not come through, unless they have one or more DC components (nonoscillating partials). And of course, if one of the signals has zero amplitude, the resulting signal also has zero amplitude.

If the fundamental frequencies of the two input signals are related by simple fractions (that is, if the ratios between their frequencies are something like 1/1, 2/1, 3/1, 4/1, 1/2, 1/3, 1/4, 2/3, 3/2) the resulting signal will be a harmonic sound. Its partials will be multiples one of the original fundamentals, or possibly a new fundamental. If the frequency ratios of the original signals are nearly but not quite one of these fractions, some beat frequencies will be perceived, which may or may not be useful. Of course, with equal temperament, the ratios given above are not perfectly precise (a perfect fifth, for example, has a frequency ratio of 1.4983, not 3/2). If the frequencies of the original signals are not at least closely related, the result of X AMP will be, shall we say, less than harmonious.

If the frequency of one of the original signals is below the audible range, then the result of X AMP is not a matter of harmony, but of amplitude. In this case, a tremolo effect (amplitude modulation) would be heard, because the resulting signal would periodically dip below the audible range. In fact, when you're using X AMP in the final algorithm block, you can use any sample as an LFO source by setting the Adjust parameter on the PITCH page to its minimum. To make this work, the algorithm must use one of the waveform functions in one of the blocks, and the sample signal must be routed to the x AMP block. Results will vary.

x GAIN

This function operates like X AMP, except that it is not affected by the settings for AMPENV, since it occurs before the F4 AMP block.

SHAPER / Final Amp (! AMP)

This function also appears in the final algorithm block when it mixes two input wires to a single output wire. The two inputs are added, then put through the SHAPER function with a fixed Adjust value of **.25**, then amplified according to the values for the parameters on the F4 AMP control-input page.

Amplitude Modulation (AMP MOD)

The AMP MOD function multiplies its two input signals, and the result is multiplied by a gain value that is determined by the parameters on the AMPMOD's control-input page. This result determines the balance between the upper and lower wires. AMP MOD can clip the signal, so you may need to use the Pad parameter.

Hard Sync Functions

SYNC M AND SYNC S

These two functions appear in Algorithms 26–31, and always work in tandem. Each is a rising sawtooth oscillator. SYNC M is the "master" waveform, and SYNC S is the "slave." These terms stem from the fact that the pitch (frequency) of the master waveform determines the repetition rate, and thus the shape, of the slave's waveform. These functions generate their own waveforms, and do not pass the sample input through the algorithm. Consequently the PITCH function does not appear for these algorithms.

Every time the master waveform falls from positive full scale to negative full scale, the slave waveform is forced to negative full scale. You can create a wide variety of timbres by adjusting the pitches of the waveforms relative to each other. This is done with the parameters on the F1 PCH and F2 PCH control-input pages. F1 is for the master, and F2 is for the slave. Pitch control is really a bit of a misnomer for the slave waveform, because its pitch is determined by the pitch of the master waveform. The fundamental of the slave waveform is forced to be the same as that of the master, since they always have the same frequencies, although the shapes of their waveforms differ.

To clarify this, assume for now that the pitch of the master waveform remains constant. When you trigger a note, both waveforms start at negative full scale. If the slave's "pitch" control is set to a much lower value than that of the master, the master waveform will reach positive full scale before the slave. So the shape of the slave waveform will be that of a more slowly rising sawtooth wave with a relatively large negative DC offset (most of the waveform will be in the negative portion of the scale).

If the slave's "pitch" control is set to a value only slightly lower than that of the master, the waveforms will be very similar, and the slave waveform will have a small negative DC offset. When the pitch settings are identical, the waveforms are identical.

If the slave's pitch setting is higher than that of the master (which usually gives more interesting results), the slave's waveform will alternate between a complete sawtooth cycle and a fraction of the subsequent cycle. At twice the frequency of the master, the waveform will have twice the frequency, and only the even harmonics of the master frequency will be pronounced. When the slave/master frequency ratio is nearly, but not exactly three, all harmonics will be present, but the 3rd, 6th, 9th (all multiples of 3) harmonics will be much louder than the others. This will sound like a resonant filter with multiple resonance peaks.

Because the pitch of the slave waveform is forced to be nearly that of the master, you can adjust the key tracking of the slave to values less than **100 cents per key** without affecting the pitch. This will help reduce some of the harshness at the high end of the keyboard.

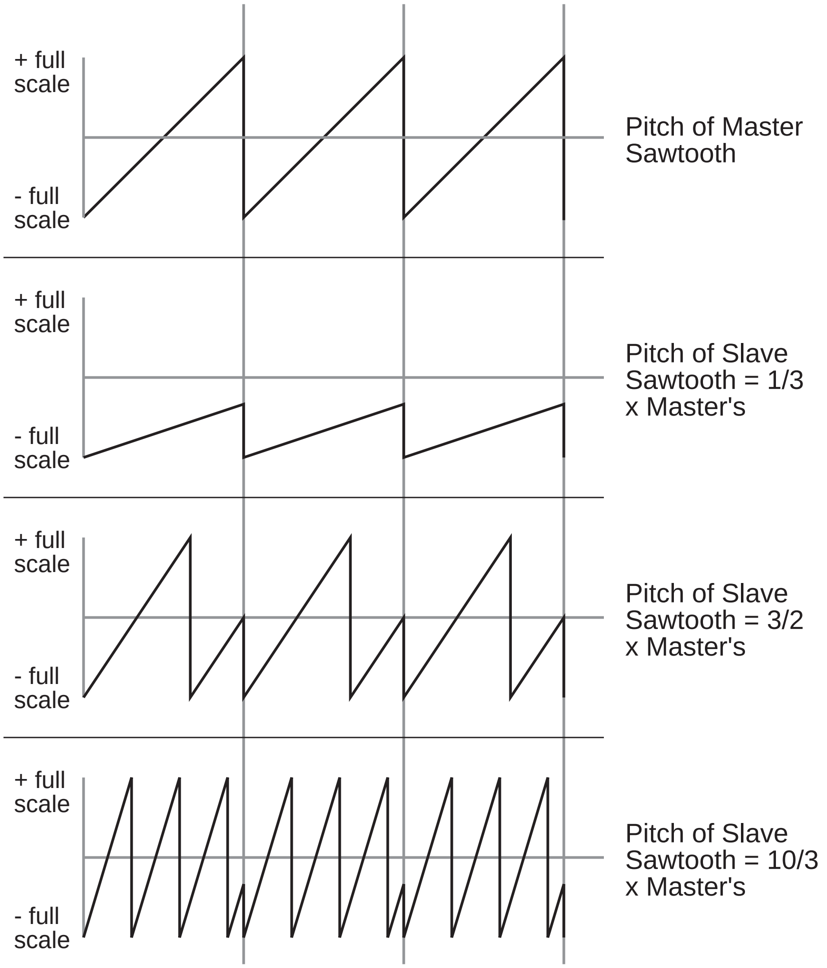

+ full scale
- full scale
Pitch of Master Sawtooth
+ full scale
- full scale
Pitch of Slave Sawtooth = 1/3 x Master's
+ full scale
- full scale
Pitch of Slave Sawtooth = 3/2 x Master's
+ full scale
- full scale
Pitch of Slave Sawtooth = 10/3 x Master's

Figure 16-6 Hard Sync Functions

Chapter 17 FUNS

The name “FUN” is an abbreviation for “Function.” FUNs are a series of equations that you can use to modulate control-source signals.

We’ve discussed various control sources throughout this manual, from the physical controls like the Mod Wheel to the software control sources like LFOs and attack velocity. You can assign them to affect your sounds in all sorts of ways.

The FUNs take the control sources one level further. By setting up a FUN as a control source, you can mix the signals of two control sources, and perform one of 50 functions on the combined signals. The result of that function becomes the new control source value. Because they can radically change their combined input values, FUNs can have a profound effect on your sounds.

You may find that experimenting with the various FUN equations gives you a better idea of their effects than reading the explanations. Although there’s some serious mathematics behind the FUNs, the most important consideration is how they affect your sounds. The more you play around with them, the better you’ll understand how powerful they are.

The Mechanics of Control Sources

We’ll return for a minute to the notion that the K2600 is an integrated system consisting of a MIDI-driven sound engine and a MIDI-driven effects processor. The sound engine responds to MIDI messages received at the MIDI In port and from the front panel, as does the effects processor.

The K2600’s control sources use their own internal signal format for interpreting control messages and communicating them to the sound engine. Every control source sent from your MIDI controller to the K2600’s sound engine is translated to a value in the range from -1 to +1. This consistency enables the sound engine to process control source signals very efficiently. Conversely, the K2600’s internal control source signals are translated to MIDI values before being sent to the MIDI Out port.

A control signal value of 0 represents minimum effect; it’s equivalent to the control source being turned off or disconnected. A control signal value of +1 represents the maximum positive effect of a control source, while a value of -1 represents the maximum negative effect of a control source.

Unipolar and Bipolar Control Sources

There are two kinds of control source signals: unipolar and bipolar. A unipolar signal has a value between 0 and +1. A bipolar signal has a value between -1 and +1.

A switch pedal is unipolar; its control signal value will never go below 0. Since it’s a switch control, it has only two possible values: 0, which corresponds to off or minimum, and +1, which corresponds to on or maximum. When you depress your MIDI controller’s sustain pedal, for example, it sends a control signal value of +1 to the K2600’s sound engine.

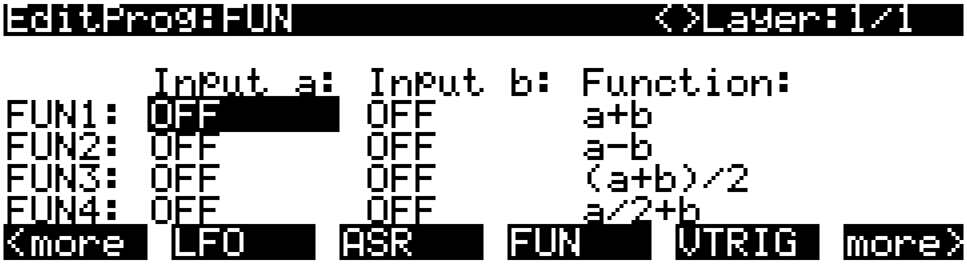
Continuous controls can be unipolar or bipolar. Consider your MIDI controller’s Mod and Pitch Wheels as examples. Normally, the Mod Wheel affects the K2600 as a unipolar control source; it sends a control signal value that’s interpreted as 0 when it’s fully down, and values interpreted between 0 and +1 as you push it up. When fully up, it sends a value that's interpreted as +1. It can be used as a bipolar control source by assigning a value of **Bi-Mwl** to any control source parameter.

The Pitch Wheel is normally bipolar; it sends a control signal value that's interpreted as 0 when it's centered, values interpreted between 0 and -1 as it's pulled downward, and values interpreted between 0 and +1 as it's pushed upward. It can be used as a unipolar control source by assigning a value of **AbsPwl** to any control source parameter.

The FUNs can act as unipolar or bipolar control sources; it depends on the values of the input signals and the nature of the function you choose. Depending on the function you choose to process the input signals, the output signal value can exceed +1 or -1. Normally the signal merely pins at +1 or -1; that is, it won't go any higher or lower. In some cases, however, the output signal value is wrapped around instead of pinning; we'll mention these cases as we get to them. You can assume that the output signal values of the functions listed below will pin at -1 or +1, unless specified otherwise.

Programming the FUNs

Start by entering the Program Editor, then use the soft buttons to select the FUN page. Setting up a FUN as a control source is a two-step process: assigning a FUN as the value for one or more control source parameters in the Program Editor, then programming the FUN on the FUN page, by assigning control sources to two inputs—a and b, and choosing a function (equation) that will process the combined signals from Input a and Input b.



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

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

There are four FUNS; you can combine and process four different pairs of control source signals. FUNs 1 and 3 are always local, that is, they affect each note in their respective layers independently. FUNs 2 and 4 are local by default, but they can be made global by setting a value of **On** for the Globals parameter on the COMMON page in the Program Editor. A global FUN affects all notes in its layer equally and simultaneously.

The best way to understand the use of the FUNs is to set up a simple test model, then plug in the different equations and listen to their effects. We'll walk you through the programming of a FUN and assigning it to control pitch. Then you can scroll through the list of equations at your leisure.

Start in Program mode and select Program **199**. Press **Edit** to enter the Program Editor. Select the KEYMAP page, and change the keymap to **152 Dull Sawtooth**. Then select the PITCH page, and assign a value of **FUN1** for the Src1 parameter (a shortcut is to press **1**, **1**, **2**, **Enter** on the alphanumeric pad). Select the Depth parameter and change the value to **1200 cents**. Next, select the FUN page, and select the Input a parameter for FUN1. Assign a value of **MWheel** (the quickest way is to hold the K2600's **Enter** button and move your MIDI controller's Mod Wheel). Next, select the Input b parameter for FUN1, and assign a value of **Data**. This assumes your MIDI controller either has a data slider, or a programmable control that you set to send Data messages (MIDI 06). If you don't have a data slider or a programmable control, you can set the value of Input b to AbsPwl, and use your Pitch Wheel to control Input b. If you do this, you'll need to go to the LAYER page and set the PBMode parameter to a value of **Off** to keep Pitch Wheel messages from interfering with the test model.

Now select the Function parameter, and scroll through the list of equations. Move your MIDI controller's Mod Wheel and Data slider as you play, and listen to their effects. Actually listening to the various effects while reading the explanations below will help your understanding. In the model we've set up here, Inputs a and b are both unipolar. The effect of each equation will differ depending on the type of controls you assign to the inputs. There are four possible combinations: both inputs unipolar; both inputs bipolar; Input a unipolar with Input b bipolar; Input a bipolar with Input b unipolar.

The FUN Equations

In this section we'll describe how each of the FUN equations works. In some cases, a small graph will accompany the explanation. Here's how to interpret the graphs.

Each graph shows a curve illustrating the effect of the equations on the input signals. The horizontal axis represents the possible values of the input to the FUN (the combined control signals of Inputs a and b). The vertical axis represents the possible values of the FUN's output signal. The four elements in the diagram below show you how to read these graphs:

a) the curve representing the effect of the FUN's equation on every possible input value

b) one point on that curve, representing a single input value and the corresponding output value generated by the FUN's equation

c) the input value represented by the point

d) the output value represented by the point

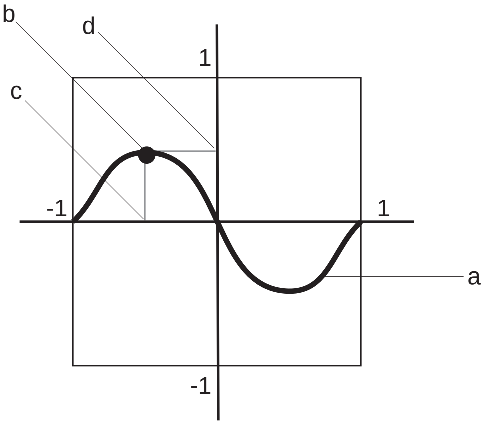


Figure 17-1 Interpreting FUN Graphs

For any point on the equation's curve, you can determine the input value by tracing a line from the point to the horizontal axis. Similarly, you can determine the output value by tracing a line from the point to the vertical axis. For the point shown in the example above, the combined values of the control signals of Inputs a and b equal about -.5, which translates to an output value of +.5. An input value of -1 gives an output value of 0, as do input values of 0 and +1. An input value of +.5 gives an output value of -.5.

Basic FUN Equations

The first six equations are weighted sums and differences—that is, the signal values of Inputs a and b are added to or subtracted from each other, and are divided in turn by various amounts to alter their effects relative to each other. These equations give you several different types of mixers for combining the signals of the two inputs.

a + b

The values of Inputs a and b are added, creating a simple mixer. For example, you might have **LFO1** assigned for the Src2 parameter on a layer's PITCH page, and a FUN assigned for the DptCtl parameter. On the FUN page, if you set Input a to a value of **MWheel**, and Input b to a value of **MPress**, then this equation will let you modulate the depth of the LFO's pitch modulation with your MIDI controller's Mod Wheel or with mono pressure. You could set a fixed initial depth with the Mod Wheel and alter it further with mono pressure. In this case the output signal would pin at +1 or -1 fairly quickly.

a - b

This operates similarly to the previous equation, but the value of Input b is subtracted from the value of Input a. This equation will reverse the normal effect of the control source assigned to Input b. For example, if Input a is off, and Input b is assigned to a unipolar control source like **MWheel**, then the Mod Wheel will generate a control signal of -1 when fully down, and 0 when fully up.

(a + b) / 2

The values of Inputs a and b are added, and the sum is divided by 2. This gives you the same kind of control as the previous two equations, but the output signal will reach +1 or -1 half as often as with the equation a + b.

a / 2 + b

The value of Input a is divided by 2, and the result is added to the value of Input b. Input a has half the effect of Input b.

a / 4 + b / 2

The value of Input a is divided by 4, and the value of Input b is divided by 2. The two results are added to give the output value. Input a has half the effect of Input b, and the total result has half the effect of the previous equation.

(a + 2b) / 3

The value of Input b is multiplied by 2, and the result is added to the value of Input a. This sum is then divided by 3. Input a has half the effect of Input b, and the total result has somewhat more effect than the previous equation, but less effect than a + b.

a * b

The values of Inputs a and b are multiplied. If you like using Src2 and DptCtl, this equation can be used to create a similar type of control source (it's equivalent to the Src2/DptCtl pair with the MinDpt parameter set to **0**).

-a * b

The value of Input a is multiplied by -1, then multiplied by the value of Input b. This will reverse the normal effect of the control source assigned to Input a. This equation also produces an effect like that of Src2 and DptCtl with the MinDpt parameter set to **0**.

a * 10^b

The actual equation is: a x $(10^{(2 \times b)} \div 100)$. This is an exponential curve. 10 is raised to the (2 x b) power, then divided by 100. This result is then multiplied by a. Another way to express this is as follows: a change of 1 in the value of Input b results in a hundredfold change in the output value. Here are a few possible output values:

Input Values		Output Values
a = +1	b= +1	+1
a = +1	b= 0	.01
a = +1	b= -1	.0001
a = 0	b = +1	0
a = 0	b = 0	0
a = 0	b = -+1	0
a = -1	b = +1	-1
a = -1	b = 0	-.01
a = -1	b = -1	-.0001

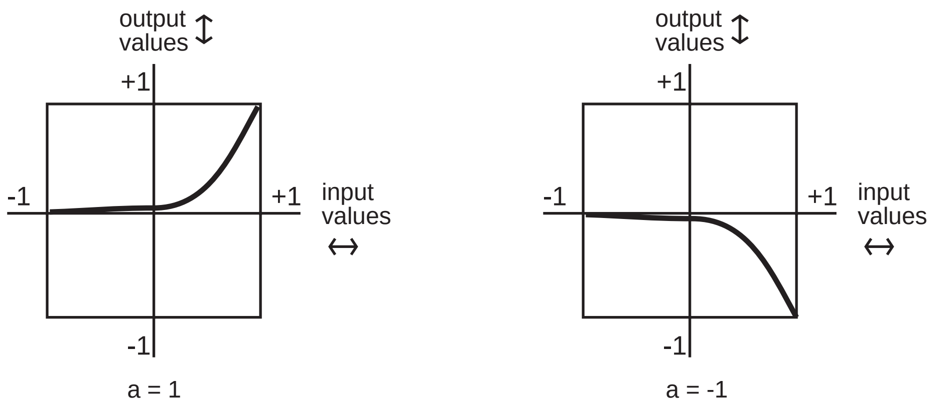


Figure 17-2 a * 10 ^ b

| a + b |

The values of Inputs a and b are added, and the absolute value of the sum is taken. If the sum is negative, it is multiplied by -1. This makes the FUN a unipolar control source.

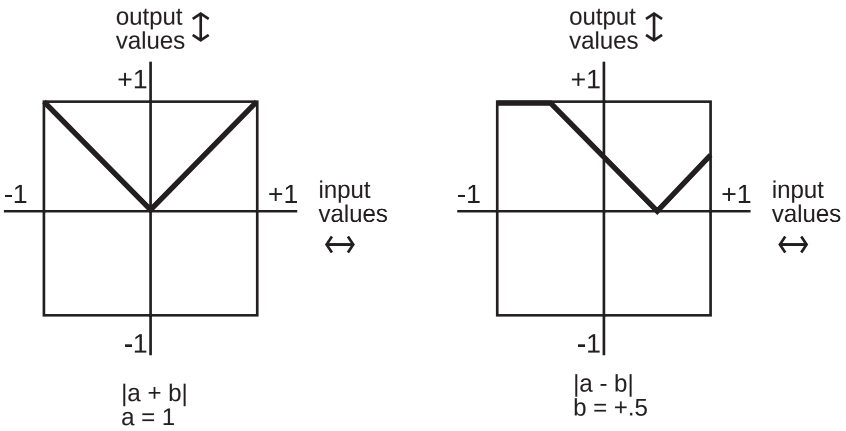


Figure 17-3 | a + b |, | a - b |

| a - b |

The value of Input b is subtracted from the value of Input a, and the absolute value is taken. If the difference is negative, it is multiplied by -1. This also makes the FUN unipolar.

min (a, b)

The values of Inputs a and b are compared, and the smaller value becomes the output value. This can be used to limit the value range of a control source. If, for example the value of the control source assigned to Input b is left at +.5, then when the value of the control source assigned to Input a is between -1 and +.5, its value will be used. As soon as its value exceeds +.5, the value of Input b is used.

max (a, b)

This is the opposite of the previous equation. The values of Inputs a and b are compared, and the larger value becomes the output value.

Quantize b to a

This turns the control source assigned to Input b into a stepped control source. Instead of smooth transitions from minimum to maximum, it will jump from minimum to maximum in some number of equal steps. The number of steps is determined by the value of Input a. The normal real-time application of this is to set a stationary value for Input a to set the number of steps in the effect. Then use the control source assigned to Input b as a real-time control to induce the stepped effect. Changing the value of Input a in real time will produce an extraneous (but possibly useful) effect.

Range of Values for Input a		Number of Steps as Input b Moves from Min to Max	
From	To	When Input B is Bipolar	When Input B is Unipolar)
0	.0625	1 (no effect)	1
.0625	.125	2	1*
.125	.1875	3	2
.1875	.25	4	2*
.25	.3125	5	3
.3125	.375	6	3*
.375	.4375	7	4
.4375	.5	8	4*
.5	.5625	9	5
.5625	.625	10	5*
.625	.6875	11	6
.6875	.75	12	6*
.75	.8125	13	7
.8125	.875	14	7*
.875	.9375	15	8
.9375	1	16	8*

Table 17-1 Quantizing b to a

As an example, consider the FUN we set up at the beginning of the previous section: the Mod Wheel was assigned as Input a, and the data slider as Input b. The FUN was assigned as Src1 on the PITCH page, and the depth of Src1 was set to **1200 cents**. If you push the Mod Wheel all the way up, the value of Input a will be +1. This will set the number of steps at 8, since the data slider sends a unipolar control signal. With your MIDI controller's data slider at minimum, play and sustain a note. Then move the data slider slowly up. The pitch of the note will jump up an octave in 8 steps as you move the data slider all the way up.

If the value of Input a is negative, it's multiplied by -1, so its value always falls within the ranges above. When Input b is bipolar and the resulting number of steps is an odd number, the steps are centered around a value of 0—that is, the center step is equivalent to no effect from Input b. When the number of steps is even, a value of zero is not included in the steps. This is also true for the values marked by an asterisk when Input b is unipolar.

lowpass (f = a, b)

This equation might be called a lag equation. Its effect is to introduce a delay in the K2600's response to changes in the value of Input b. It works by filtering (reducing) higher values of Input b. The value of Input a determines the degree to which the values of Input b are filtered. Low values for Input a will induce a long lag when the value of Input b changes. High values will shorten the lag. When Input b remains constant at a high level, low values of Input a will cause the FUN to sweep up slowly from 0 to the value of Input b. Higher values for Input a will cause the FUN to sweep more rapidly.

The four graphs below show the effect of different values for Input a on the change of Input b. In each graph, the value of Input b jumps from 0 to +1. In graph 1, the value of Input a is +1. Each successive graph represents the same change in the value of Input b, at successively lower values for Input a.

This equation works as intended only when the value of Input a is 0 or positive. Negative values for Input a will result in a much less predictable response than positive values. You might like the effect, but it won't be anything like what we've just described.

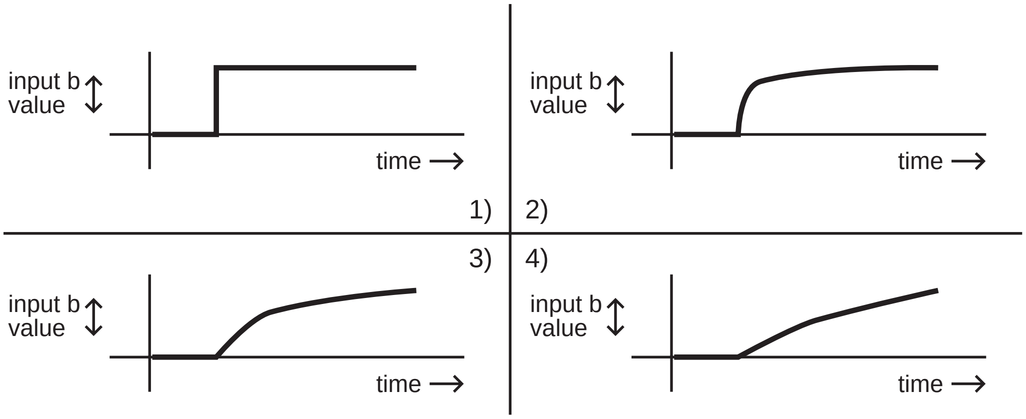


Figure 17-4 lowpass (f = a, b)

hipass (f = a, b)

With this equation the low values of Input b are filtered according to the value of Input a. This causes somewhat different results compared with the lowpass equation above. At low values for Input a, low values for Input b will have little effect, while high values for Input b will cause the FUN to quickly reach full effect then slowly sweep down to its starting level. At high values for Input a, a rapid change in the value of Input b will have little effect. At low values for Input a, rapid changes in the value of Input b will cause the FUN to respond quickly to the change, then slowly fade back to minimum effect. Listening to the effects at different values for each input will give you the best understanding.

The four graphs below show the effect of different values for Input a on the change of Input b. In each graph, the value of Input b drops from +1 to 0. In graph 1, the value of Input a is +1. Each successive graph represents the same change in the value of Input b, at successively lower values for Input a.

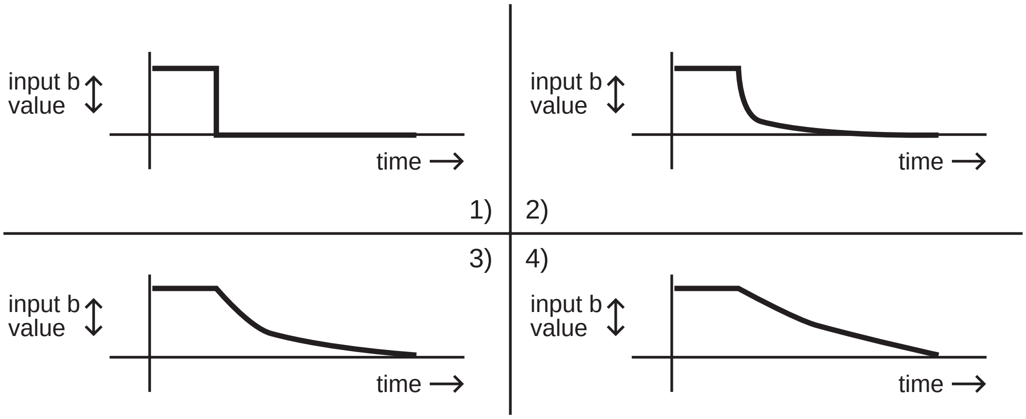


Figure 17-5 hipass (f = a, b)

b / (1 - a)

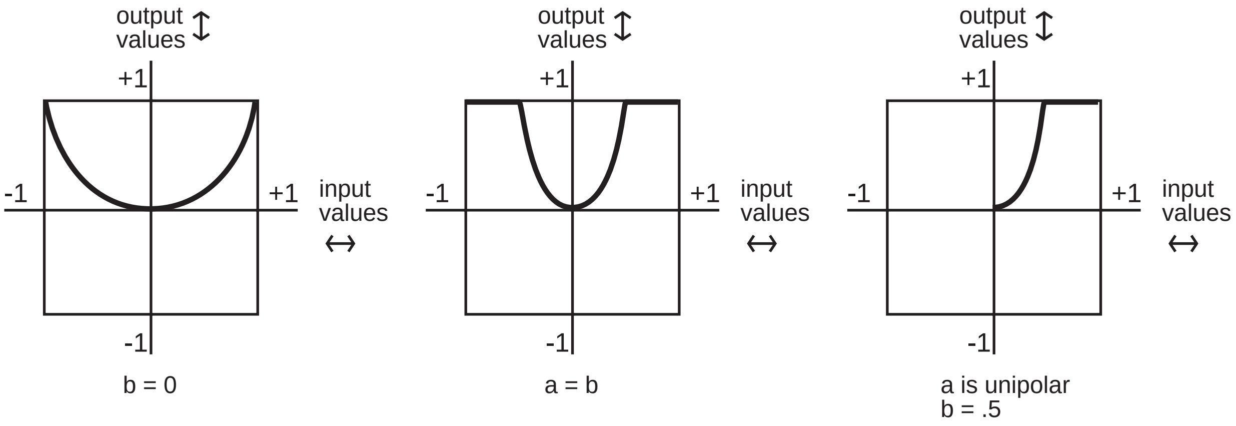
This is another weighted difference equation similar to the first six. The value of Input a is subtracted from 1. The value of Input b is then divided by the difference. You'll get considerably different results for different input values of a and b.

a(b-y)

Think of this equation as reading "y is replaced by the result of the function a(b-y)." The value of y indicates the value of the FUN's output signal. Every 20 milliseconds, the K2600 takes the current value of y, runs the equation, calculates a new value of y, and inserts the new value into the equation. Consequently the value of y will change every twenty milliseconds. Here's an example. When you play a note, the K2600 starts running the FUN. The first value for y is always 0. We'll assume the value of Input a is +.5, and the value of Input b is +1. The first time the K2600 evaluates the FUN, the result of the equation is .5 x (+1 - 0), or .5. So the FUN's output value after the first evaluation is .5. This becomes the new value for y, and when the K2600 does its next evaluation of the FUN, the equation becomes .5 x (+1 - .5), or .25. The resulting output value is .25, which becomes the new value for y. For the next evaluation, the equation is .5 x (+1- .25), or .375.

(a + b)^2

The values of Inputs a and b are added, and the result is squared (multiplied by itself). This will change the linear curve of a unipolar control signal into a curve that's lower at its midpoint (by a factor of 2). Bipolar control signals will generate curves that are high at both ends, and 0 in the middle.



b = 0

a = b

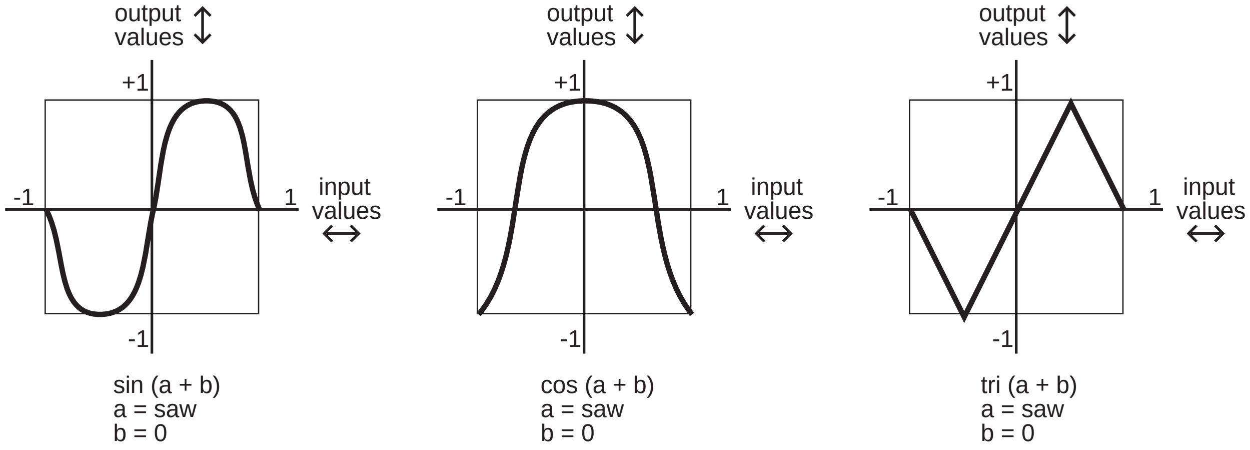
a is unipolar
b = .5

Figure 17-6 (a + b)^2

sin (a + b), cos (a + b), tri (a + b)

These equations are intended to be used with inputs that are sawtooth waves—for example, Input a might be LFO1 with its shape set as a sawtooth. Each equation will map a sawtooth-shaped input into a sine-, cosine-, or triangle-shaped output. Other input waveform shapes will result in outputs with more complex waveform shapes.

Other ways to get sawtooth shapes as inputs to these FUNs are to use other FUNs as the inputs, with their equations set as any of the ramp equations described later in this section (see the note on page 17-18 about the evaluation order of the FUNs). You could also use LFOph1 or LFOph2 as inputs. The first three graphs below show the result of these functions when Input a is a rising sawtooth wave, and the value of Input b is 0. The fourth shows the result of the sin (f=a + b) equation when the value of Input b is 0 and Input a is a sine wave.



sin (a + b)
a = saw
b = 0

cos (a + b)
a = saw
b = 0

tri (a + b)
a = saw
b = 0

Figure 17-7 sin (a + b), cos (a + b), tri (a + b)

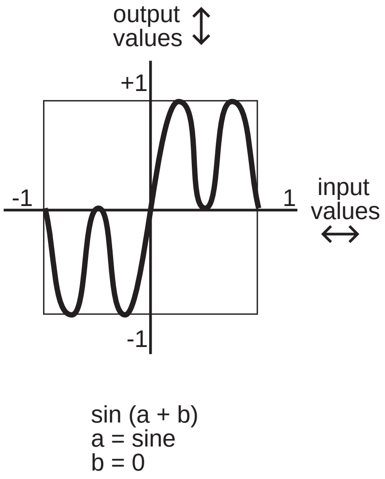


Figure 17-8 sin (a + b)

Warp FUN Equations

The next five equations all behave similarly, and are intended to be used as follows: the value of Input a is the controlling value, and normally remains constant, although it doesn't have to. The value of Input b is expected to change over time; Input b might be an LFO, for example. The value for Input a affects how the FUN calculates its output value while the value of Input b changes.

warp1(a, b)

We call this the Vari-slope™ equation. The value of Input a controls the mapping of values for Input b. If Input b is a sawtooth wave, different values for Input a will change it into a triangle wave. If Input b is a more complicated waveform, the output waveform is also more complicated.

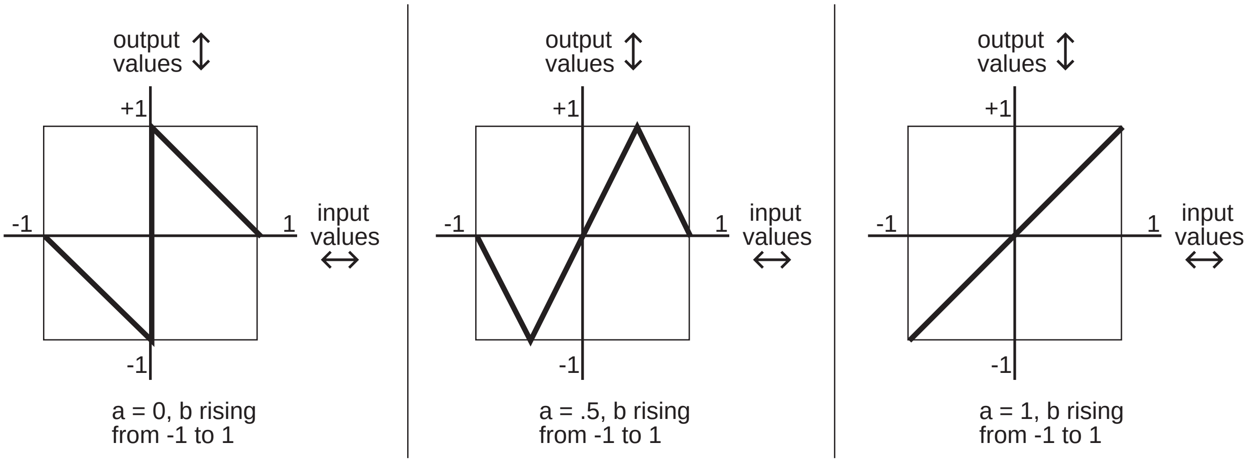


Figure 17-9 warp1(a, b)

warp2(a, b)

We call this equation Slant-square.™ Again, the value of Input a controls the mapping of values for Input b. If Input b is a sawtooth wave, different values for Input a will turn it into a number of variations on square waves.

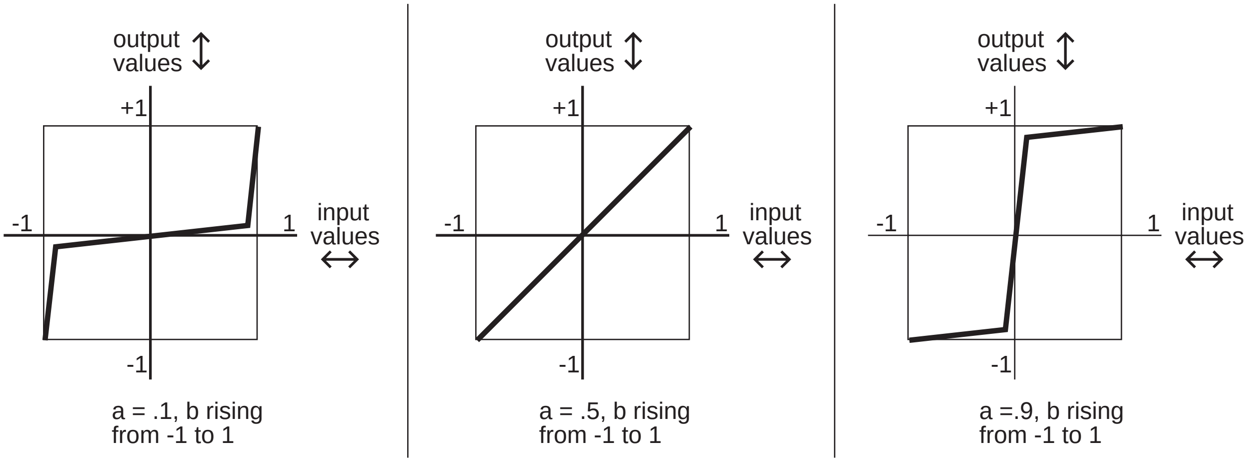


Figure 17-10 warp2(a, b)

warp3(a, b)

We call this one the Variable Inverter.™ It looks at the binary numbers that represent the values of Inputs a and b, compares the corresponding bits in each number, and performs an XOR operation on them (we'll explain that below). The resulting number is converted into the output value. This can produce some erratic results, but if variety is what you're after, this equation will give it to you. You'll get your best results when Input b is an LFO with a slow rate.

The XOR operation is a subprogram that applies a truth table to each of the digits in the binary numbers that represent the values of Inputs a and b. Each of these numbers is a string of 16 digits (bits); each bit is either a 0 or a 1. The subprogram looks at the first bit of each number. If they're both 0s, the resulting value is 1. If one is a 0 and the other is a 1, the result is 1. If they're both 1s, the result is 0. This process is repeated for the remaining 15 bits of each number, and a new 16-bit number is generated. This number represents the output value of the FUN.

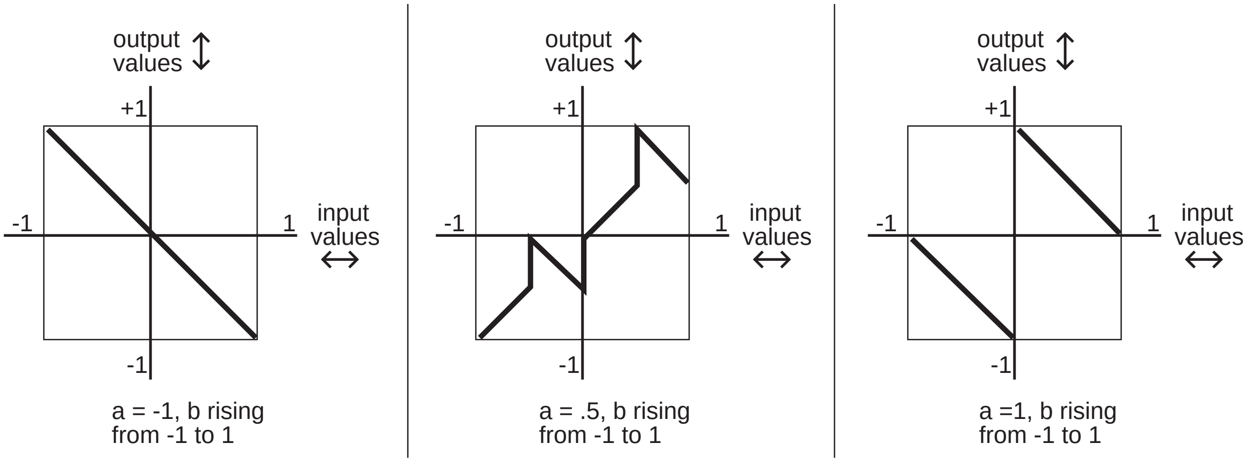


Figure 17-11 warp3(a, b)

warp4(a, b)

This equation, the Period Inverter,™ is based on repeated evaluations of the value of Input b. The K2600 compares each new value of Input b with the value from the previous evaluation. If the absolute value (always a positive number) of the difference between the two is greater than the value of Input a, the current value of Input b is multiplied by -1.

The primary feature of this equation is that it will take a discontinuous signal and make it continuous. If, for example, FUN1 uses an equation like a(y + b), its output can wrap around from +1 to -1, or vice versa. You could set **FUN1** as Input b for FUN2, set Input a of FUN2 to **ON** (+1), and FUN2 would remove the discontinuity from the signal. The first graph below shows a hypothetical output signal with such a discontinuity, and the second shows how FUN2 in this case would make the signal continuous without drastically changing its shape.

If, on the other hand you *want* the signal to become discontinuous, you can use the warp4(a, b) equation in a single FUN, with Input a set to **OFF** (**0**), and the signal would be multiplied by -1 with each evaluation of Input b.

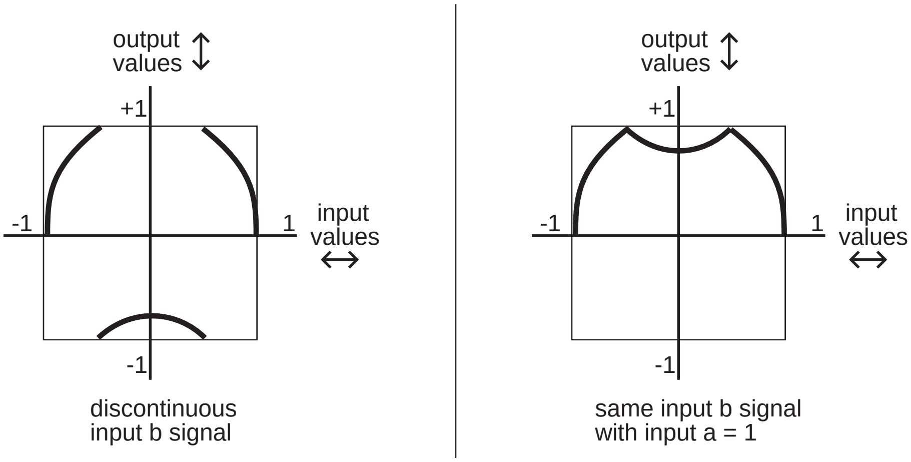


Figure 17-12 warp4(a, b)

warp8(a, b)

This relatively simple equation is a x b x 8. If the result is beyond the range of -1 to +1, it wraps around from +1 to -1 (or vice versa), until it's within the allowable range. The table below shows some examples of how this works.

a x b x 8 =	Final Output Value
-7.4	.4
-4.2	-.2
-1.8	.8
-.6	-.6
.4	.4
1.2	-.2
2.6	.6
5.4	-.4

Table 17-2 warp8(a, b)

Boolean FUN Equations

a AND b

The values of Inputs a and b are interpreted as logical quantities—they're considered TRUE if they're greater than +.5, and FALSE otherwise. This turns the FUN into an on/off switch. In the model we set up in the previous section, **FUN1** was assigned to control Src1 on the PITCH page, and Src1's depth was set to **1200 cents**. With this equation, both Input a (the Mod Wheel in this case) and Input b (the data slider in this case) would have to be more than halfway up for the FUN to switch on. The pitch would jump 1200 cents as soon as both control sources moved above their halfway points. As soon as one of them moved below its halfway point, the pitch would jump back to its original level.

This equation can be used to trigger ASRs, or as a layer enable control, or for any control source that toggles on and off. If you set one of the inputs to an LFO, the FUN would switch on and off every time the LFO's signal went above +.5 (as long as the other input was also above +.5).

a OR b

This equation is very similar to a AND b. The only difference is that the FUN will switch on when the value of either Input a *or* Input b moves above +.5.

Sawtooth LFO FUN Equations

The next six equations case the FUN to generate a sawtooth LFO as its output signal. Each performs a different operation on the values of Inputs a and b, and the resulting value is multiplied by 25. The result determines the frequency of the LFO. If the value is a positive number, the LFO has a rising sawtooth shape. If the value is negative, the LFO has a falling sawtooth shape. When the resulting values are large (above 10 or so), the output waveform is not a pure sawtooth; a bit of distortion occurs.

ramp(f=a + b)

The values of Inputs a and b are added, then multiplied by 25.

ramp(f=a - b)

The value of Input b is subtracted from the value of Input a, and the difference is multiplied by 25.

ramp(f=(a + b) / 4)

The values of Inputs a and b are added, and the sum is divided by 4. This value is multiplied by 25.

ramp(f=a * b)

The values of Inputs a and b are multiplied, and the result is multiplied by 25.

ramp(f=-a * b)

The value of Input a is multiplied by -1, then multiplied by the value of Input b. The result is multiplied by 25.

ramp(f=a * 10^b)

10 is raised to the power of b, then multiplied by the value of Input a. The result is multiplied by 25.

Random / Chaotic LFO FUNs

The next five equations function somewhat like the equation a(b-y) described earlier, in that they start with a value of 0 for y, evaluate the equation, and use the result as the new value of y for the next evaluation. Although they all can function as LFOs (they can have a repeating cycle of output values), they can become chaotic depending on the input values.

a(y + b)

The values of y and b are added, then multiplied by the value of a.

ay + b

The values of a and y are multiplied, then added to the value of b.

(a + 1)y + b

1 is added to the value of a. The sum is multiplied by the value of y. The result is added to the value of b.

y +a(y + b)

The values of y and b are added. The sum is multiplied by the value of a. The result is added to the value of y.

a |y| + b

The absolute value of y is taken (if it’s a negative value, it’s multiplied by -1). The absolute value of y is multiplied by the value of a. the result is added to the value of b.

Sample b On a

This is a sample and hold function. The values of Inputs a and b are interpreted as logical quantities, as described for the equations a AND b, a OR b. When the value of Input a changes from FALSE to TRUE (goes above +.5), the value of Input b at that moment is sampled (recorded), and becomes the FUN's output value. This value remains constant until Input a makes another transition from FALSE to TRUE.

Sample b On ~a

This works like the previous equation, but the value of Input b is sampled whenever the value of Input a makes a transition from TRUE to FALSE.

Track b While a

This equation also interprets the values of Inputs a and b as logical quantities. While the value of Input a is TRUE, the value of Input b is used as the FUN's output value. The output value changes exactly as the value of Input b changes. When the value of Input a goes FALSE, the FUN's output value freezes and remains constant until the value of Input a becomes TRUE again. The FUN's output value then continues to track the value of Input b.

Track b While ~a

This is the opposite of the previous equation. The FUN's output value tracks the value of Input b as long as the value of Input a is FALSE.

Diode Equation FUNs

The remaining equations perform a diode-like function; only positive input values are significant. If the result of the equation is negative, the FUN's output value is 0. You can use these equations to limit bipolar control signals to unipolar values. Normally you'll use these by setting Input a or b to **ON** or **OFF**, and assigning some control source to the other input. These will enable you to produce a variety of output curves.

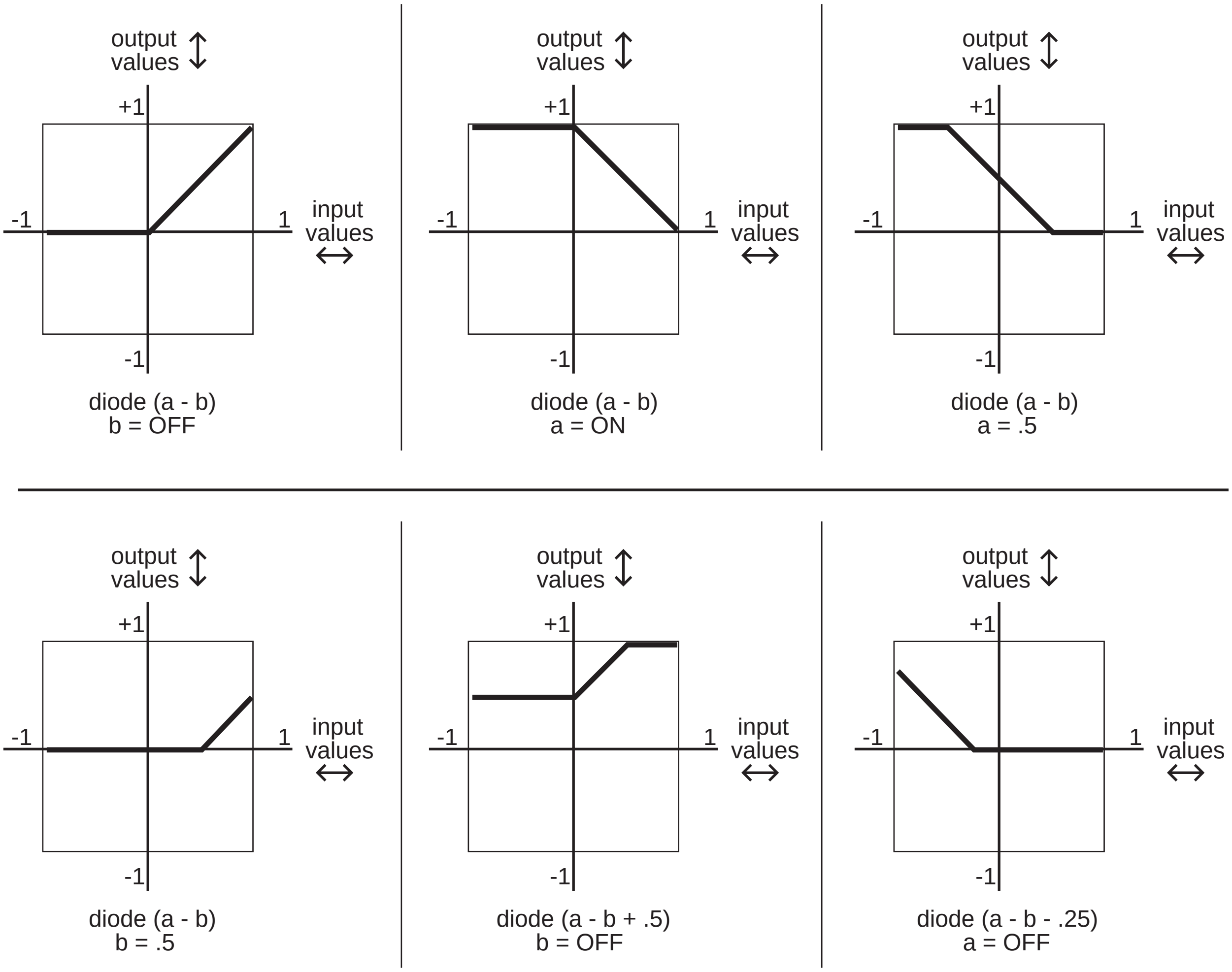


Figure 17-13 Diode Equations

Diode (a - b) Subtracts the value of Input b from the value of Input a. If the difference is less than 0, the output value is 0.

Diode (a - b + .5) Adds a constant of +.5 to the difference of (a - b), then maps all negative values to 0. The curve is the same shape as Diode (a - b), but shifted upward 1/4 of the range between -1 and +1.

Diode (a - b - .5) Same curve as diode (a - b), shifted downward 1/4 of the range.

Diode (a - b + .25) Shifts the curve up 1/8 of the range, and diode (a - b - .25) shifts the curve down 1/8 of the range.

The Order of Evaluation for FUNs

The K2600 is a computer, and processes information at very high speeds. Every 20 milliseconds, it checks the condition of every active parameter, evaluates any changes, and processes the new information. This is done according to a rigid set of priorities that determines the sequence in which the parameters are evaluated.

The status of each control source is evaluated in turn, according to the sequence in which they appear in the Control Source list. In the case of the FUNs, they are evaluated in the following order: FUN1, FUN2, FUN3, FUN4 (although there are a few control sources that get evaluated between FUN2 and FUN3).

This sequence of evaluation becomes significant if you assign a FUN as the input for another FUN. You should always assign FUNs as inputs for higher-numbered FUNs. For example, you might assign FUN1 as Input a for FUN2. Since the K2600 needs to know the value of FUN1's output before it can evaluate FUN2, assigning the FUNs in this way will ensure that the K2600 can evaluate both FUNs as quickly as possible.

If you were to assign FUN2 as an input for FUN1, then when the time came for the K2600 to evaluate FUN1, it wouldn't know the current value for FUN2's output, so it would evaluate FUN1 according to the previous value of FUN2. There would be a delay of one evaluation cycle before a change in FUN2 would be reflected in FUN1. This might have an adverse affect on the start of notes as you play. You won't harm anything, but you might not hear what you expect to hear.

Chapter 18
Other Editors

In addiion to the editors accessible from their respective modes, there are three editors that enable you to modify other performance parameters of the K2600. They are the Intonation Table Editor, the Velocity Map Editor, and the Pressure Map Editor.

The Intonation Table Editor

Intonation tables define the interval between the notes in each octave. The default intonation table is **1 Equal**, which sets precisely equal intervals between notes—the standard for modern western music. If you're interested in playing other styles of music, you can use any of the K2600's other factory intonation tables, or create you own with the Intonation Table Editor.

The intonation table is selected in Master mode with the Intonation parameter. Here's the list of available factory intonation tables (check the *Musician's Reference* for brief descriptions):

1	Equal	9	Indian Raga
2	Just	10	Arabic
3	Just b/7th	11	Bali/Java 1
4	Harmonic	12	Bali/Java 2
5	Just Harmonic	13	Bali/Java 3
6	Werkmeister	14	Tibetan
7	1/5th Comma	15	CarlosAlpha
8	1/4th Comma	17	Pyth/dim5

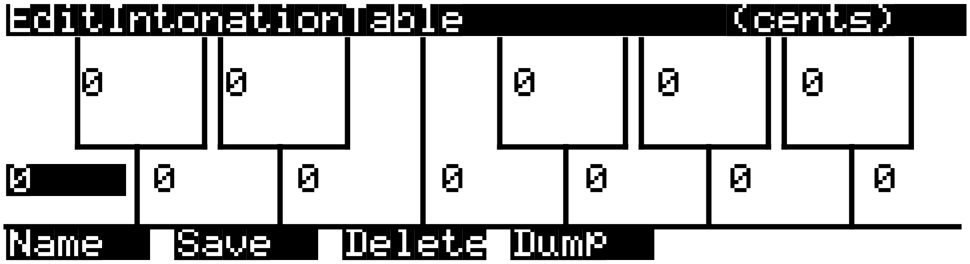
As you scroll through the options for the Intonation parameter, you'll notice an eighteenth entry with a name like **Obj v*n.nn***. This isn't really an intonation table; it's simply a convenient place for the K2600 to keep a record of the current version of ROM objects. If you ever need to find out the version of ROM objects loaded into your unit, this is the place to check. Just remember, if you're using a nonstandard intonation table, that you'll need to reset the Intonation parameter after you check the ROM object version list at "Table" 18.

Depending on the number of options you've installed in your K2600, you may see several intonation-table entries numbered as high as 22.

It's important to emphasize that the detuning values for each parameter affect the intervals in each octave independently. That is, notes that are an octave apart will remain perfectly tuned regardless of the detuning between notes within the octave. Consequently, editing intonation tables will not enable you to create tunings with more (or fewer) than 12 notes per octave. If you want to do this, use the Keymap Editor, assign each key to its own key range, then tune the samples in each key range.

When you're ready to edit an intonation table, select Master mode, then select the Intonation parameter. Use any data entry method to select the intonation table you want to edit. Check the value of the Intonation Key parameter (IntonaKey). This sets the tonic, or reference note for the parameters on the Intonation Table Editor page. We'll explain this below.

Enter the Intonation Table Editor by pressing the **Edit** button. The Intonation Table Editor page displays a graphic representation of a C octave, with the low C always representing the tonic. The values for these parameters indicate the amount of detuning applied to each note relative to perfectly equal intonation. If you look at the values for the intonation table 1 Equal (as illustrated below), you'll see that the value of each parameter is **0**. None of the notes is detuned, so the intervals between each note are exactly the same. Take a minute to look at the values for some of the other intonation tables, and listen to the effect on the tuning.



EditIntonationTable (cents)

0 0 0 0 0

0 0 0 0 0 0 0

The top line of the display gives you the usual reminder of your location. The bottom line labels the soft buttons for the Intonation Table Editor, which let you perform the usual librarian functions of naming, saving, and deleting the currently selected intonation table, or dumping it via SysEx.

Assume for now that the IntonaKey parameter is set to a value of **C** (which is the default). This means that each C is the reference point for defining the intervals of all the other notes.

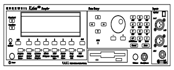
Let's say you want to create an intonation table that flats the 5th in each octave. Select the value on the G note displayed by the Intonation Table Editor, and set its value to a negative number—for example, **-12 cents**. This will reduce the interval between the tonic (C, in this case) and the 5th (G, in this case) by 12/100ths of a semitone. All the Gs will remain exactly an octave apart from each other, but they'll be 12 cents flat from their normal pitches.

Press the **Compare** (Disk mode) button to hear the difference between the edited table and the original. Press the **Save** soft button to begin the save dialog, where you can rename and save your edited intonation table.

The changes we've discussed here are based on a value of **C** for the IntonaKey parameter. If you were to select Master mode and change the value of the IntonaKey parameter, you'd change the tonic note that the intonation table uses to define the intervals of the other notes. Let's say you change IntonaKey from **C** to **G**. The tonic changes to G, and the intervals in each octave are defined relative to G. The minor 2nd is now G#, the major 2nd is A, and so on. In the example we gave above, all the Ds would now be 12 cents flat.

A good reference source for descriptions of various alternative intonations is a book by Scott R. Wilkinson entitled *Tuning In: Microtonality in Electronic Music*, available from the Hal Leonard Publishing Corporation.

The Velocity Map Editor



Velocity maps affect the way the K2600 interprets attack velocity values. Velocity maps are assigned as the values for two different parameters: VelocMap on the TRANSMIT page in MIDI mode, and VelocityMap on the RECEIVE page in MIDI mode. (They also apply to the VelTouch parameter in Master mode, but since the VelTouch parameter doesn't affect the K2600R, you shouldn't change its default value.) There's only one list of velocity maps, and all three of these parameters use this list. Even so, certain maps are intended for use with the VelTouch parameter in the keyboard version of the K2600, and others for use with the MIDI-mode parameters. You can think of these as internal maps and MIDI maps. With the K2600R, you should use only the MIDI maps, and unless you have a problem with the K2600R's velocity response, or that of MIDI devices connected to the K2600R's MIDI Out port, you should leave both the Receive Velocity map and the Transmit Velocity map set to their default values.

Every attack velocity value generated by your MIDI controller is mapped through the velocity map assigned on the MIDI-mode RECEIVE page before being passed to the sound engine. If you're using the Local Keyboard Channel feature, attack velocity values received at the K2600's MIDI in port go first through the Receive Velocity map, then to the sound engine, then through the Transmit Velocity map (as selected with the VelocMap parameter on the TRANSMIT page in MIDI mode), and to the MIDI Out port.

The VelocMap parameter on the TRANSMIT page in MIDI mode affects only the velocity values the K2600 sends to its MIDI Out port. Normally you'll leave its value set to **1 Linear**, especially when you're recording sequences with a personal computer or hardware sequencer. You might want to adjust this parameter, however, when you need to change the response of MIDI slaves that you're driving from the K2600. If, for example, you're overdriving a DX7 that you have slaved to the K2600, try the map **6 Hard2**.

The VelocityMap parameter on the RECEIVE page in MIDI mode affects how the K2600 responds to attack velocity values received at its MIDI In port. Normally you'll leave this set to **Linear**, as with the VelocMap parameter on the MIDI-mode TRANSMIT page. You might want to adjust this parameter, for example, to boost or cut the volume response of the K2600 during playback in Song mode, or when playing a sequence from an external sequencer.

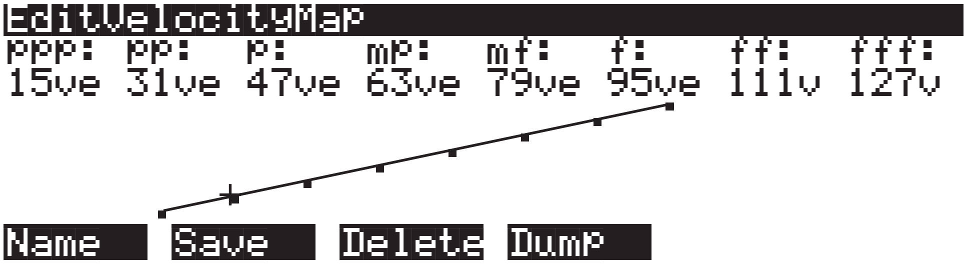
Here's a list of the velocity maps stored in ROM (the factory presets). Keep in mind, however, that you should use only the linear MIDI maps; the internal maps are meant for the K2600.

1	Linear (the default on the TRANSMIT and RECEIVE pages in MIDI mode): Velocity values are passed through this MIDI map unchanged.
2	Light1
3	Light2
4	Light3
5	Hard1
6	Hard2
7	Hard3

The attack velocity value of every note received by the K2600's sound engine is routed through the map assigned to the VelocityMap parameter (on the RECEIVE page in MIDI mode). The velocity maps convert each attack velocity value into a new value, depending on the curve of the map. By editing velocity maps, you can change the attack velocity value that the maps calculate.

Using the Velocity Map Editor

To get to the Velocity Map Editor, select any of the parameters we've mentioned above—VelTouch (Master mode) VelocMap (TRANSMIT page, MIDI mode), or VelocityMap (RECEIVE page, MIDI mode). Use any data entry method to select the map you want to edit, then press **Edit**.



```
EditVelocityMap
ppp: pp:  p:   mp:  mf:  f:   ff:  fff:
15ve 31ve 47ve 63ve 79ve 95ve 111v 127v
```

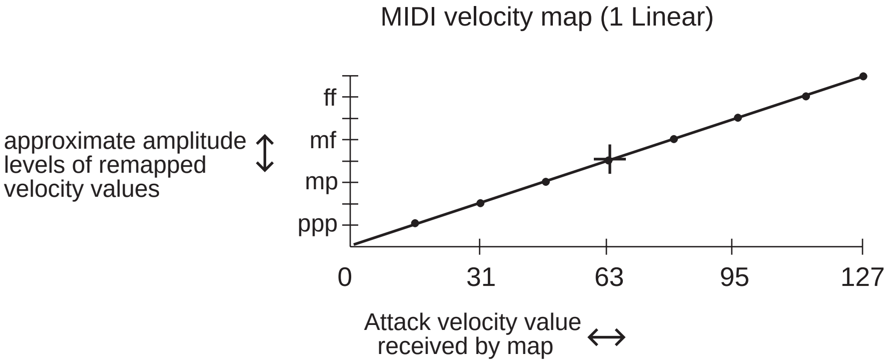
The top line of the display gives you the usual reminder of your location. The bottom line labels the soft buttons, which execute the normal librarian functions of naming, saving, or deleting velocity maps, or dumping them via SysEx. The values you see in this diagram are the settings for the velocity map **1 Linear**.

The eight parameters on the Velocity Map Editor page are expressed in terms of the eight dynamic levels of standard musical notation. As far as the velocity maps are concerned, the dynamic levels represented by the parameter names are simply reference points—for example, when using the map **1 Linear**, an attack velocity value of 63 will result in a remapped velocity value at the mp level of the K2600's dynamic range.

The values for these parameters are expressed in "vels," which correspond to the standard MIDI attack velocity values of 1–127. The values you set for each parameter determine the velocity value that must be generated to achieve the dynamic level for that parameter. For example, if you're using the map **1 Linear** as the value for the VelocityMap parameter, then the K2600 will play notes at full volume only when the attack velocity values for those notes are 127. Notes with attack velocities from 1 to 15 will be played at the lowest volumes. They wouldn't all play at the *same* volume; their volumes would be graduated, but would be at the low end of the scale.

The values for these parameters have another function as well. They determine the velocity thresholds for the velocity triggers (VTRIG), as well as the velocity crossovers for multi-velocity keymaps. Once again using the example of the map **1 Linear**, if you have the VelCrossover parameter in the Keymap Editor set to a value of **mp**, then attack velocity values higher than 63 will trigger the high-velocity range of a dual-velocity keymap, while velocity values of 63 or lower will trigger the low-velocity range.

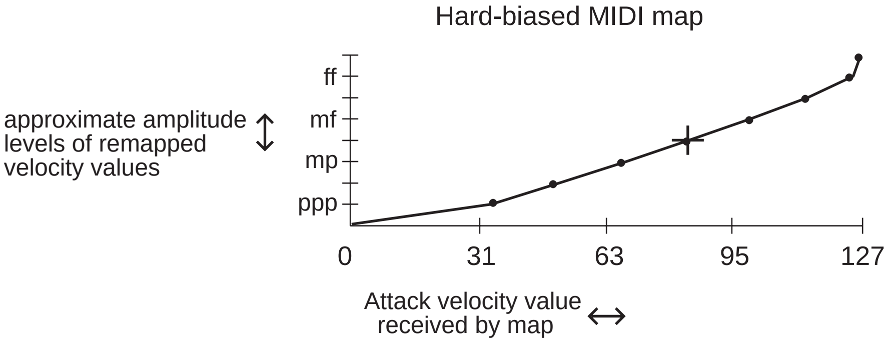
The curve that represents the velocity map is shown in the center of the page. Think of this curve as a plot on a graph, as shown in the diagram below. The horizontal axis gives the attack velocity values received by the velocity map. The vertical axis gives the approximate dynamic level at which the K2600 will play the note (in response to the remapped velocity value).



When you select a parameter, the small crosshairs move to indicate the position on the curve of the dynamic level represented by that parameter. Each of the eight levels is marked on the curve by a small dot. The crosshairs jump to one of the dots when you select the corresponding parameter. The dots always have the same vertical location, but move to the left or right depending on the velocity value assigned to them. The farther to the right that one of these dots is located, the higher the attack velocity value required to play a note at that level. As the curve suggests, velocity values between those set for the eight dynamic levels will be remapped to a new velocity represented by the height of the curve.

As you change the value of the currently selected parameter, the shape of the curve changes to reflect the new value. As an example, let’s assume that you’re driving a MIDI slave synth from the K2600’s MIDI Out port, and you get full-amplitude sounds from the slave, even when you play softly on your MIDI controller. To fix this, you’d edit the map to give it a harder bias (or use one of the preset MIDI maps with a hard bias). That is, you want to have to strike your controller’s keys harder to get full amplitudes.

The diagram below shows a hard-biased linear map similar to the **Hard3** map. Although the even progression from low to high remapped values remains, the entire curve is shifted so that a given input velocity value will remap to a lower value.



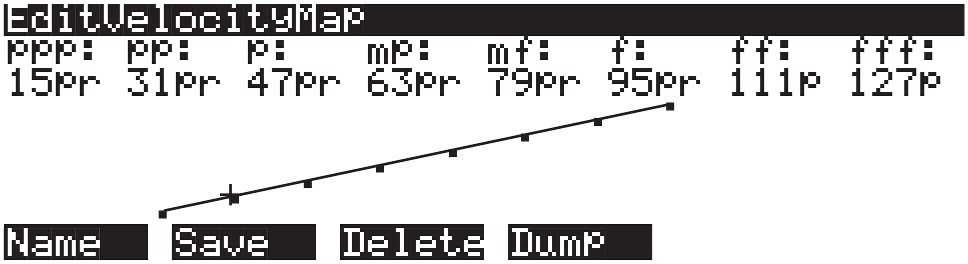
The Pressure Map Editor

Pressure maps function in much the same way as velocity maps, but they affect the K2600’s response to mono pressure messages received from your MIDI controller. Like the velocity maps, pressure maps convert pressure values of 0–127 to new values that are sent to the sound engine. Also like the velocity maps, pressure messages received at the K2600’s MIDI In port from your MIDI controller pass through the Receive Pressure map, then to the sound engine. If you’re using the Local Keyboard Channel feature, the pressure messages pass first through the Receive Pressure map, then through the Transmit Pressure map, then to the MIDI Out port.

Pressure maps are assigned as the values for the PressTouch parameter in Master mode, the PressMap parameter on the TRANSMIT page in MIDI mode, and the PressureMap parameter on the RECEIVE page in MIDI mode. There’s one list of pressure maps, which is used to select the pressure map for all three parameters. As with the velocity maps, you should use only the MIDI map (ID 1) for the MIDI-mode parameters, leave the value of the PressTouch parameter at its default value. Here’s the list of all seven maps. Keep in mind, however, that you should use only the linear MIDI maps; the internal maps are meant for the K2600.

1	Linear: The default internal and MIDI pressure map.
2	Easy
3	Easier
4	Easiest
5	Hard
6	Harder
7	Hardest

To edit a pressure map, select one of the three pressure map parameters, and set its value to the pressure map you want to edit. Edit the map **1 Linear** if you want to use your edited map with the MIDI pressure map parameters. Press **Edit**, and the Pressure Map Editor page will appear.



```
EditVelocityMap
ppp: pp:  p:   mp:  mf:  f:   ff:  fff:
15pr 31pr 47pr 63pr 79pr 95pr 111p 127p
```

The top line of the display gives you the usual reminder of your location. The bottom line labels the soft buttons, which execute the normal librarian functions of naming, saving, or deleting the current pressure map, or dumping it via SysEx.

The eight parameters are expressed in terms of the dynamic levels of standard musical notation. They don’t correspond to actual volume levels, they’re just a reference point (ppp corresponds to minimum pressure effect, and fff corresponds to maximum). Their values are expressed as “prs” units, and range from 0–127, corresponding to the values of mono or poly pressure messages. The values you set for these parameters determine the pressure message values that must be generated to achieve the dynamic levels expressed by the parameters. Values in between those indicated by the parameters will be remapped as indicated by the curve.

Chapter 19
Audio Outputs

Audio Configurations

There are several ways to get audio output from the K2600. The most common configuration is a pair of mono or stereo 1/4-inch cables connecting the Mix outputs of the K2600 to inputs on a mixer or keyboard amp. The Mix outputs carry the sum of all the signals routed to the separate analog outputs (A–D), including effects. Another common configuration is to use one or more of the separate analog outputs. Connecting to one of the separate outputs does not remove the corresponding portion of the signal from the MIX outputs (for example, if you connect cables to the A pair, you'll get the Output A signal at both the A outputs and the Mix outputs).

The audio output routing of the K2600 depends primarily on two parameters:

- The Pair parameter on the OUTPUT page in the Program Editor; this routes the signal from programs to Inputs A–D in KDFX
- The Output parameters (A–D) on the OUTPUT page in the Studio Editor; this routes the KDFX output to the physical audio outputs (optionally bypassing KDFX, or adding effects from the KDFX Aux bus)

In other words, individual programs route the audio signal from the K2600's sound engine into the effects processor (KDFX), while the studios assigned to those programs route the signal from KDFX to the jacks on the rear panel.

Of course, there are other options: you can set the value of the Outpair parameter (on the CHANNELS page in MIDI mode) to **KDFX-A**, **KDFX-B**, **KDFX-C**, or **KDFX-D**. If you set Outpair for Channel 1 to **KDFX-A**, for example, then every program on Channel 1 sends its audio signal from the sound engine to Input A of KDFX—overriding the program's routing.

You can also use the Out parameter on the CH/PRG page in the Setup Editor in the same way, forcing each zone of a setup to send its output to a particular KDFX input, overriding the settings of the programs in each zone.

***Note**: we recommend that you make the cable connection to the K2600 (or any instrument) after you've made your other audio connections, since this reduces the chance of creating static electricity that can cause an audible "pop" (and, in extreme cases, cause equipment damage).*

Audio Routing: Programs to KDFX

1. In any mode (typically Program mode), highlight a program name with the cursor, then press **Edit** to enter the Program Editor. Note how many layers there are in the program.

2. Press one of the **more** soft buttons until you see OUTPUT at the bottom of the display. Press the corresponding soft button to view the OUTPUT page for the current layer.

3. Set the value of the Pair parameter as desired. This value determines which KDFX input (A–D) gets the output from the current program layer.

4. Repeat this process for each layer in the program (or, if you're editing a setup, for every layer of every program in the setup).

Audio Routing: KDFX to Audio Outputs

Every program that uses KDFX has a studio assigned to it. The studio defines the KDFX parameters for the program to which it's assigned.

1. In the Program Editor, press one of the **more** soft buttons until you see KDFX at the bottom of the display. Press the corresponding soft button to view the KDFX page for the current layer.

2. Highlight the Studio parameter, then press **Edit** to enter the Studio Editor.

3. Press the **OUTPUT** soft button to view the OUTPUT page for the current studio. Note that it controls all layers of the program.

4. Set the values for each of the four Output parameters. These parameters represent the four pairs of analog outputs; the parameters' values specify which KDFX output bus gets routed to each of the analog outputs.

Using the Optional Digital Outputs

Digital audio output is available for K2600s that have either the sampling option or the digital I/O option. Each option gives you different possibilities for digital output configuration.

The sampling option provides 16-bit stereo digital output, via coaxial (XLR) or optical output jacks that are supplied with the sampling option. The output is routed through the sampler's analog-to-digital converter (ADC). The stereo digital out carries either the K2600's analog Mix output, or an external analog signal converted to digital. You can choose either SPDIF or AES/EBU format for the output. You can set the output sample rate at 48, 44.1, 32, or 29.6 KHz, or you can slave the K2600 to an external clock via one of the stereo digital inputs. These inputs also enable you to sample digital signals, as described on page 14-8.

The digital I/O option provides 16- or 20-bit eight-channel digital output in Kurzweil's proprietary KDS format, via a serial port supplied with the digital I/O option. You'll need to connect a Kurzweil DMTi or other KDS-compatible device to the serial port. The digital I/O option also provides direct stereo digital output (optical or coaxial).

With the digital I/O option, the output sample rate is fixed at 48 KHz. The digital I/O option also provides eight channels of KDS-format digital input, which you can route into VAST programs. See *Playback Mode* on page 6-26.

If your K2600 has both the sampling option and the digital I/O option, you can route the signal to the stereo digital outputs either through the sampler, or directly.

K2600 With Sampling Option Only

1. Go to the SampleMode page (in Program, Setup, Quick Access, or Master mode, press the **Sample** soft button).
2. Set the value of the Input parameter to **Digital**. This enables the Format parameter.
3. Set Format as needed. **SPDIF** is the most common setting; you may want to use **AES/EBU** in professional environments.
4. Set Input to **Analog** to enable the Src and Rate parameters.
5. Set Src as needed. Use **Int** to send the K2600's analog Mix output to the digital outs (via the sampler), or use **Ext** to convert an external analog signal—received at either of the Analog inputs—to digital before sending it to the digital outs.
6. If the K2600 is going to be the master (the device controlling the output sample rate), set the Rate parameter as desired. This results in a constant output sample rate. If you want an external master to control the K2600's output sample rate, set Input back to **Digital**, and connect the master device to one of the stereo digital inputs. Set the value of Cable accordingly: **Coaxial** for the XLR input, or **Optical** for the optical input. Now the K2600 adjusts its output sample rate according to the clock signal received at the stereo digital in.

If you're using a Coaxial cable and are going from the Kurzweil's XLR out to an RCA in, be sure the cable is wired as described in *Cables and Input Jacks* on page 14-1.

K2600 With Digital I/O Option Only

The K2600's audio output goes to both the stereo digital outs (XLR and optical) and the eight-channel KDS outs (it goes to the analog outs as well). The stereo output is a copy of the signal routed to analog out A—so if you want the full K2600 output at the stereo digital outs, you'll need to make sure that the current studio routes all of its output to out A (on the OUTPUT page in the Studio Editor, set the value of the Output A parameter to **Mix**). The eight-channel output is a copy of the signal routed to analog outs A (Left and Right) through D (Left and Right).

The output sample rate is fixed at 48KHz. In any situation requiring different sample rates, you'll need to use a sample rate converter (like the DMTi). When you need a clock signal to synchronize two or more instruments, there are two options:

- Make the K2600 the master; it can't be slaved to an external clock signal. Use a sample rate converter, if necessary, to match the sample rates of your other instruments.
- Make another instrument the master. The K2600 won't respond to the clock signal. Use a sample rate converter, if necessary, to match the K2600's output rate to the master's rate.

When you need to slave one or more instruments or devices to an external master clock, the K2600 is necessarily the master, because it can't be slaved to an external clock. If you need to use sample rates other than 48KHz, you'll need to use the DMTi or another sample-rate converter to convert the K2600's output to the desired rate.

When you have the digital I/O option, the Effects-mode page includes an additional parameter—DigOut—which enables you to switch the digital output's word length to 16 bits or 20 bits. Most devices support 16-bit, and many support 20-bit.

You may also want to adjust the value of the Dither parameter on the Effects-mode page, to adjust the digital noise floor.

K2600 with Sampling Option and Digital I/O Option

When your K2600 has both the sampling and digital I/O options, you have both eight-channel direct digital output, and stereo digital output, with two options for the stereo output. These options are defined by an additional parameter—Out—that appears on the SampleMode page when Input is set to **Digital**. Set the Out parameter to **A/D**, and the K2600 routes its analog Mix output through the sampler before sending it to the stereo digital outs. This converts the signal to analog and back to digital before it reaches the stereo outs. Set Out to Direct (**Dir**), and the K2600 sends the signal from analog out A to the stereo digital outs, without the extra conversion. The eight-channel KDS output is always direct.

Chapter 20
Programming Examples

The other chapters in this manual have described the K2600's features in detail. This tutorial chapter will take you step-by step through several programming operations.

Each of the following examples will begin from the same starting point: the default program with ID 199. This program is included specifically for the purpose of giving you a programming template. Most of its parameters have been set at values that don't affect the sound of the program.

You may want to adjust some of the parameters of Program 199, to create your own customized programming template. Even if you don't, it's a good idea to begin with Program 199 when you're building a new sound, so you'll know exactly what you have from the start.

Keep in mind that none of these examples provides you with a usable program. Instead, the examples are designed to give you tools and concepts which you can apply to your own sounds. Once you become familiar with the programming basics in this chapter, analyze a few of the factory presets by moving through the Program Editor, and observing how those presets were designed. This may help you learn more techniques for creating new sounds.

While in the Program Editor, there are several editing shortcuts you can use. To call up a control source, enter its number on the alphanumeric buttonpad, or hold **Enter** and strike a key on the keyboard (see Control Sources in Chapter 4 of the *Musician's Reference*). When a highlighted parameter has a Control Source as its value, press **Edit**, and you will go directly to that Control Source page. You can also use the **Previous Pg**, **Mark**, and **Jump** buttons (see page 5-7).

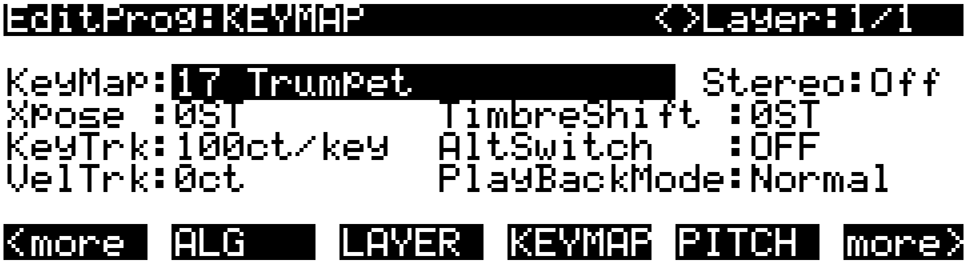
Example 1

Trumpet with Delayed Vibrato and Velocity-triggered Falls

Vibrato is a regular oscillation in pitch that adds dimension to any sound. Brass players will often "fall off" from a note, punching it then letting the pitch roll down smoothly or in small fast steps.

To create these effects, we'll use an LFO to control the pitch, (this is the typical way to create vibrato), and delay it with an ASR. This way you'll hear the vibrato only on notes that you hold for a second or so. The stab will be done with a second ASR controlling pitch and amplitude. The stab's ASR will be triggered by a velocity trigger (VTRIG), so only those notes you play at fortissimo will stab.

Start by selecting Program 199 and pressing **Edit**. The ALG page will appear. The first task is to change the keymap. Press the **KEYMAP** soft button to select the KEYMAP page.



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

KeyMap:17 Trumpet              Stereo:Off
Xpose :0ST        TimbreShift :0ST
KeyTrk:100ct/key  AltSwitch   :OFF
VelTrk:0ct        PlayBackMode:Normal

<more  ALG    LAYER  KEYMAP PITCH  more>
```

The KeyMap parameter is already selected, and as you can see, the Default program uses the Grand Piano keymap. Use any data entry method to change the keymap to **Trumpet**, which has ID 17. The KEYMAP page should look like the diagram above when you're done. Remember that you can play your MIDI controller's keyboard at any time while editing, so you can listen to each change as you make it.

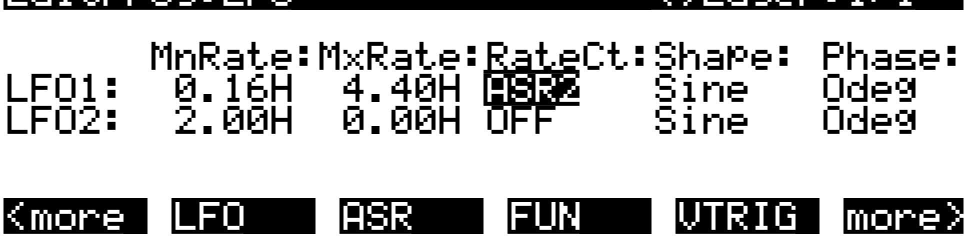
Next set up the vibrato. Start by selecting the PITCH page (press the **PITCH** soft button). Use the cursor buttons to move the cursor to the Src2 parameter. Use any data entry method to select **LFO1** as its value by pressing **1**, **1**, **4**, **Enter** on the alphanumeric buttonpad (or hold **Enter** and strike B 5 on the keyboard). This assigns LFO1 to control the pitch of the trumpet sample.

The next step is to set the depth of the vibrato. Select the MaxDpt parameter and assign a value of **10 cents** (**1**, **0**, **Enter**). Since the default program is preset to have your controller's Mod Wheel control the depth of Src2, you can hear the vibrato by pushing the Mod Wheel fully up (LFO1 has nonzero default values in the default program, otherwise, you wouldn't hear the vibrato). If you're not sure you hear the vibrato, try setting the MaxDpt parameter to a larger value.

Next, select the DptCtl parameter and assign a value of **ASR2** by pressing **1**, **1**, **1**, then **Enter** (or hold **Enter** and strike G# 5 on the keyboard). This will cause ASR2 to control the depth of the vibrato. At this point, the default values for ASR2 will cause the vibrato to fade in and out.

There are two more steps to programming the delayed vibrato: adjusting the rate of LFO1 and setting up ASR2 to control the vibrato's delay. First, highlight Src2 and press **Edit**; this brings up the LFO page. The default value for LFO1's minimum rate (the MnRate parameter) is **2 seconds**. Select this parameter with the cursor buttons, and set its value to **.16** seconds (**1**, **6**, **Enter**). Select the MxRate parameter, and set its value to **4.40 Hz** (**4**, **4**, **0**, **Enter**). Select the RateCt parameter, and assign a value of **ASR2** (**1**, **1**, **1**, **Enter**). The vibrato will still fade in and out because of the default settings of ASR2.

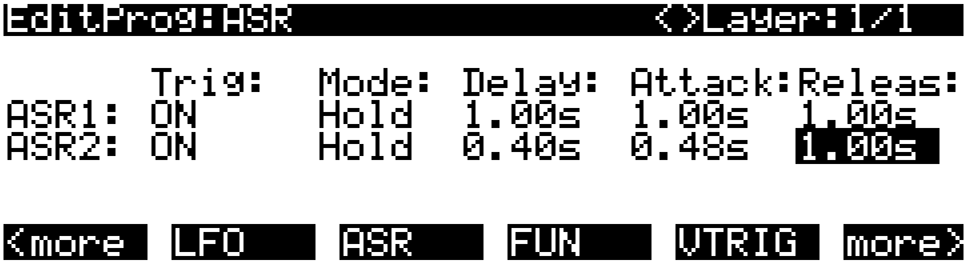
The LFO page should now look like this:



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

      MnRate:MxRate:RateCt:Shape: Phase:
LFO1:  0.16H  4.40H ASR2   Sine   0deg
LFO2:  2.00H  0.00H OFF    Sine   0deg

<more  LFO    ASR    FUN    VTRIG  more>
```

Now select the ASR page to adjust the settings for ASR2. Since the cursor is highlighting ASR2 as the value for the RateCt parameter, you can select the ASR page by pressing **Edit**.

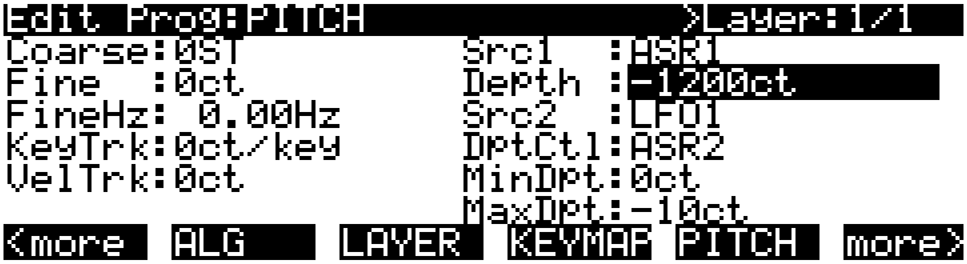
To program a realistic delayed vibrato, you need to adjust the Mode, Delay, and Attack parameters. Select the Mode parameter and change its value to **Hold** (use the Alpha Wheel or **Plus**/**Minus** buttons). This will prevent the vibrato from fading as it did. (This fading was caused by the ASR repeating, which was the default setting.) Now select the Delay parameter and set its value to **.4** seconds (**4**, **0**, **Enter**). Select the Attack parameter and change its value to **.48** seconds (**4**, **8**, **Enter**). The vibrato should now begin to fade in gradually after a short delay, then remain constant at a rate of 4.40 cycles per second. The ASR page should look like the page below. We're finished with the delayed vibrato; next is the velocity stab.



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

       Trig:   Mode:  Delay:  Attack:Releas:
ASR1:  ON      Hold   1.00s   1.00s   1.00s
ASR2:  ON      Hold   0.40s   0.48s   1.00s
```

We want the stab to drop the trumpet's pitch *and* amplitude, but only when notes are played fortissimo or louder. This is done by using ASR1 as a control source on both the PITCH and AMP pages, then using a velocity trigger (VTRIG) to control ASR1.

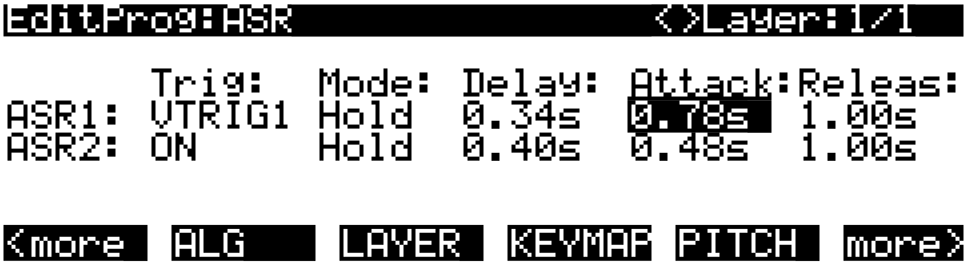
First return to the PITCH page (if you're still on the ASR page, press the **more>** soft button three times, and the **PITCH** soft button will appear). Press **PITCH**, then select the Src1 parameter, and set its value to **ASR1** (press **1**, **1**, **0**, **Enter** or hold **Enter** and strike G 5). Then select the Depth parameter, and set its value to **–1200 cents** (**+/-**, **1**, **2**, **0**, **0**, **Enter**). ASR1 has nonzero default values in Program 199, so you'll hear the pitch drop an octave if you strike and hold a note. The PITCH page should look like this:



```
Edit Prog:PITCH                    >Layer:1/1
Coarse:0ST            Src1  :ASR1
Fine  :0ct            Depth :-1200ct
FineHz: 0.00Hz        Src2  :LFO1
KeyTrk:0ct/key        DptCtl:ASR2
VelTrk:0ct            MinDpt:0ct
                      MaxDpt:-10ct
```

The next step is to adjust the characteristics of the stab by programming ASR1. Select the Src1 parameter again, and press **Edit** to select the ASR page.

First, select the Trig parameter, and assign a value of **VTRIG1** (press **1**, **0**, **6**, **Enter** or hold **Enter** and strike D$^{\#}$ 5). Select the Delay parameter and set its value to **.34** seconds (**3**, **4**, **Enter**). Set the Attack parameter to a value of **.78** seconds. The ASR page should now look like this:



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

      Trig:   Mode:  Delay:  Attack:Releas:
ASR1: VTRIG1  Hold   0.34s   0.78s  1.00s
ASR2: ON      Hold   0.40s   0.48s  1.00s

<more  ALG    LAYER  KEYMAP  PITCH  more>
```

Notice that the soft buttons are identical to those on the PITCH page; this is because you pressed **Edit** to call up the ASR page, instead of using the **<more>** soft buttons.

To make the stab sound realistic, we'll drop the amplitude at the same rate as the pitch. To do this, select the F4 AMP page (press the **more>** soft button once, then press the **F4 AMP** soft button). Select the Adjust parameter, and assign a value of **8 dB**. This will give the trumpet a bit more punch. Next, select the Src1 parameter, and set its value to **ASR1**. Then select the Depth parameter and set its value to **-68 dB**.

Here's an important point to remember: ASR1 is being used to control both the drop in pitch for the stab, and the drop in amplitude as well. All of the control sources can be similarly assigned for as many parameters as you like.

The final step in this example is to set the velocity threshold of the stab. Right now the stab is occurring on almost every note, but we want it to happen only when playing fortissimo or louder. To do this, press the **more>** soft button three times, then press the **VTRIG** soft button to select the VTRIG page. The VTrig1 Level parameter is already selected, so just turn the Alpha Wheel until the VTrig1 Level value is **ff**. Now you can play softly without triggering the stab.

That's it for Example 1. If you want to save your work, the easiest way is to press **Exit**. The K2600 will ask you if you want to save. You should probably press the **Rename** soft button, give your program a new name, then save it with a new ID. See page 5-3 if you need help with the Save dialog.

Example 2

Lowpass Filter, Envelopes

This example will show you how to assign a DSP function to an algorithm block (the 4-pole lowpass filter), and adjust its control parameters. You'll also set up an envelope to control the cutoff frequency of the filter.

Start with Program **199**, press **Edit**, and select the KEYMAP page (This is explained in a bit more detail in the previous example, if you haven't read it.) Change the value to **6 Ensemble Strings**. String sounds are especially responsive to lowpass filtering, because they have a great deal of activity in the higher frequencies. Lowpass filters, depending on their cutoff frequencies, attenuate high frequencies, so a bit of adjustment can alter a string sound considerably.

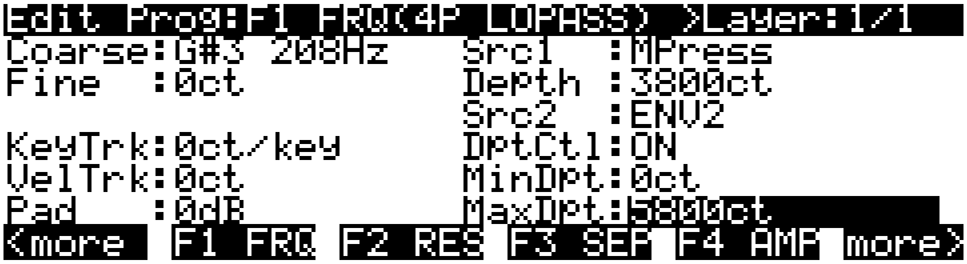
Next, press the **ALG** soft button to select the ALG page. The center DSP function block will already be selected. Set its value to **4POLE LOWPASS W/ SEP**. Notice how the sound changes.

Press the **more>** soft button, and the three soft buttons that select the control-input pages for the lowpass filter will appear. Press the **F1 FRQ** soft button.

The Coarse Adjust parameter will already be selected. Try a few different values for this parameter, to get a feel for its effect on the sound. Then set it to a value of **G$^{\#}$3 208 Hz**. Use the cursor buttons to select the Src1 parameter, and set its value to **MPress** (**3**, **3**, **Enter**). Cursor down once to the Depth parameter, and set it to a moderate value, like **3800 cents**. Now try playing and applying pressure to the keys. (Obviously, this will have no effect if your MIDI controller doesn't send mono pressure.)

Now we're going to use an envelope to control the sweep of the filter. The K2600 has one envelope dedicated to shaping the output of the sound; this is called AMPENV. However, two other envelopes (ENV 2 and ENV3) can be assigned to control other parameters. You can assign AMPENV as a Control Source, but using ENV2 and ENV3 lets you assign separate envelopes to do different things.

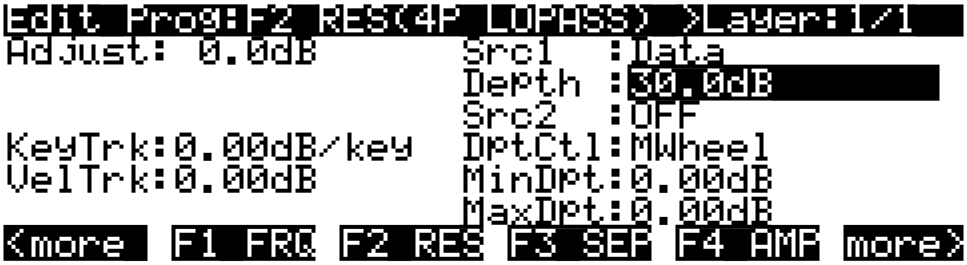
Cursor down to the Src2 parameter, and set its value to **ENV2** (**1**, **2**, **1**, **Enter**). Move the cursor down to the DptCtl parameter and set its value to **ON**. Cursor down to the MaxDpt parameter, and set its value to **5800 cents**. As soon as you set a depth, you'll hear the envelope sweep the cutoff frequency. We'll adjust it further in a minute. The F1 FRQ page should now look like this:



```
Edit Prog:F1 FRQ(4P LOPASS)  >Layer:1/1
Coarse:G#3 208Hz   Src1  :MPress
Fine  :0ct         Depth :3800ct
                   Src2  :ENV2
KeyTrk:0ct/key     DptCtl:ON
VelTrk:0ct         MinDpt:0ct
Pad   :0dB         MaxDpt:5800ct
<more  F1 FRQ  F2 RES  F3 SEP  F4 AMP  more>
```

Next, press the **F2 RES** soft button to select the page for adjusting the resonance of the lowpass filter. Select the Src1 parameter, and set its value to **Data** (**6**, **Enter**). Select the Depth parameter and set its value to **30.0 dB**. This means that the partials at the cutoff frequency will be boosted 30 dB. The F2 RES page should look like the following diagram.

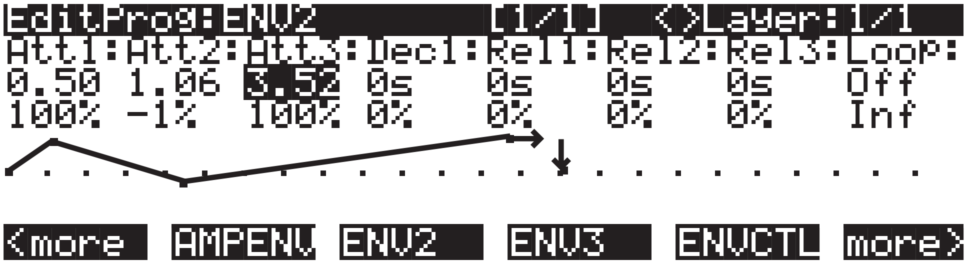
If your MIDI controller has a data slider (or a control that can be programmed to send MIDI 06), the data slider (or programmable control) will now affect the resonance of the filter. Be aware though that the change in resonance is subtle; resonance affects some waveforms more noticeably than others. You may want to set the Pad parameter on F1 FRQ page to a value of 18dB or so, then set the Gain parameter on the OUTPUT page to the same value; this reduces the potential of clipping due to resonance.



```
Edit Prog:F2 RES(4P LOPASS)  >Layer:1/1
Adjust: 0.0dB       Src1  :Data
                    Depth :30.0dB
                    Src2  :OFF
KeyTrk:0.00dB/key   DptCtl:MWheel
VelTrk:0.00dB       MinDpt:0.00dB
                    MaxDpt:0.00dB
<more  F1 FRQ  F2 RES  F3 SEP  F4 AMP  more>
```

Next press either of the **<more>** soft buttons until you see the **ENV2** soft button. Press it to select the ENV2 page. Here you'll program Envelope 2 to control the filter's cutoff frequency.

The Att1 time parameter will already be selected; set its value to **0.50 seconds**. Press the right cursor button once to select the Att2 time parameter and set its value to **1.06 seconds**. Cursor down once to select the Att2 level parameter and set its value to **-1%**. Cursor right once, then up once, selecting the Att3 time parameter. Set its value to **3.52 seconds**. The ENV2 page should look like the diagram below. Play around with the sound a bit. Hold a few notes through the full length of ENV2; the sound will continue to change for several seconds. Work your controller's Mod Wheel and data slider through several different positions to get an idea of the range of variation you can add by using the two control sources in tandem.



```
EditProg:ENV2          [1/1]  <>Layer:1/1
Att1:Att2:Att3:Dec1:Rel1:Rel2:Rel3:Loop:
0.50 1.06 3.52 0s   0s   0s   0s   Off
100% -1%  100% 0%   0%   0%   0%   Inf

<more  AMPENV  ENV2  ENV3  ENVCTL  more>
```

Example 3

Sample and Hold Using a FUN

This example will use one of the FUNs to create a sample and hold program. As usual, start with Program 199, and press **Edit**. While you're on the ALG page, select a value of **PARAMETRIC EQ** for the center DSP function block. Then select the KEYMAP page, and select the keymap **152 Dull Sawtooth**.

Now press the **more>** soft button, then the **F1 FRQ** soft button. This enables you to set the frequency for the parametric EQ, which will set the depth of the modulation for the sample and hold function. The Coarse parameter will already be selected; set its value to **D$^{\#}$4 311 Hz**. Cursor down to the KeyTrk parameter, and set its value to **100 cents per key**. Cursor over to the Src1 parameter, and set its value to **FUN1** (**1**, **1**, **2**, **Enter**). Cursor down once to the Depth parameter, and set its value to a fairly large negative number, like **-4000** cents.

Next, press the **F3 AMP** soft button, to set the amplitude for the Parametric EQ (make sure you don't select **F4 FINAL AMP**). Set the value of the Adjust parameter to about **17 dB**.

Now press either **<more>** soft button until you see the **LFO** soft button. Press the **LFO** button to select the LFO page. We'll program LFO1 as a square wave, then use it as one of the FUN's inputs.

Select the MnRate parameter for LFO1 and set it to a value of **5.00 Hz**. Select the MxRate parameter and set its value to **15.00 Hz**. Select the RateCt parameter and set it to a value of **Data**. Select the Shape parameter and set its value to **Square**. That's it for the LFO page.

Next press the **FUN** soft button to select the FUN page. Select the Input a parameter for FUN1, and set its value to **LFO1** (**1**, **1**, **4**, **Enter**). Select the Input b parameter and set its value to **RandV1** (**1**, **0**, **8**, **Enter**). Finally, select the Function parameter, and set its value to **Sample B On A**. Now when you play and hold a note, you'll hear the sample and hold effect. Use your MIDI controller's data slider (or a programmable control set to MIDI 06) to control the rate of the effect.

Here’s what’s happening. The square wave of the LFO cycles from +1 to -1. Every time this occurs, the random signal generator (RandV1) randomly picks a value which changes the frequency of the parametric EQ. There are lots of ways to set up sample and hold effects, but the FUN is the basic element. In this case, every time the value of the FUN’s Input a (LFO1) becomes greater than +.5, the value of Input b (RandV1) is sampled. That value becomes the FUN’s output until the next time the value of Input a becomes greater than +.5 (see Chapter 17 for more). In this example, the FUN’s output modulates the frequency of the EQ, and causes rapid changes in timbre.

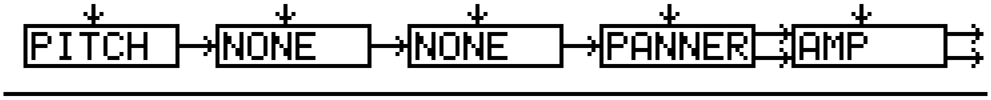
Now we’ll apply the FUN to control the pitch of the sound as well. Press either of the **<more>** soft buttons until the **PITCH** soft button is visible, and press it to select the PITCH page. Select the Src1 parameter and set its value to **FUN1**. Select the Depth parameter and set its value to **400 cents**. Now the sound’s pitch will fluctuate in sync with the EQ effect.

Example 4

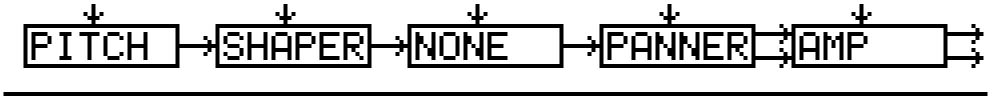
SHAPER and PANNER

Our next example incorporates two of the DSP functions, and will give you a general overview of using the algorithms to build sounds.

Starting with Program 199, press **Edit**, and while you’re on the ALG page, cursor up to the Algorithm parameter, and select **Algorithm 13**. Select values of **NONE** in the F1 and F2 blocks. The value of the F3 block is already set to **PANNER**, as shown below.



Next press the **KEYMAP** soft button to select the KEYMAP page, and select **163 Sine Wave** as the keymap. Play a few notes to accustom you ear to the sound of the unshaped sine wave. Then return to the ALG page, and select **SHAPER** for the algorithm’s F1 block. You’ll notice a bit of an effect right away. The SHAPER DSP function adds additional high frequency partials to the waveform in unpredictable ways, resulting in large changes in timbre. For more on the SHAPER, see page 16-46.



Press the **more>** soft button, then press the **F1 AMT** soft button. This enables you to adjust the amount of shaping applied to the sine wave. Try different values for the Adjust parameter, then set it to its minimum—**0.100x**. Then cursor down to the VelTrk parameter, and set a value of **0.70**. Listen for the variation in the effect as you play with different attack velocities.

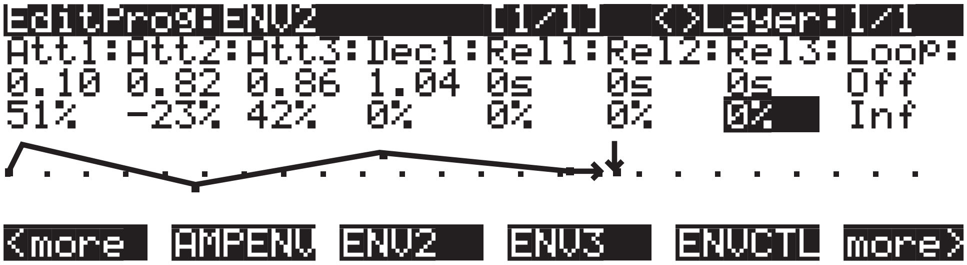
Now go back to the KEYMAP page, and select different keymaps so you can hear the SHAPER’s effect on different sounds. When you’re finished experimenting, set the Keymap parameter at **152 Dull Sawtooth**.

Notice how notes on the high end tend to break up. If you return to the F1 AMT page you can reduce this distortion and/or aliasing with the KStart and KeyTrk parameters.

Select the KStart parameter and set a value of **C 2 Unipolar**. This limits the amount of shaping applied to notes above or below C 2, depending on the value of the KeyTrk parameter. Next set the value for the KeyTrk parameter to **-0.018x per key**. Since we're using a negative value, the amount of shaping will decrease with higher notes.

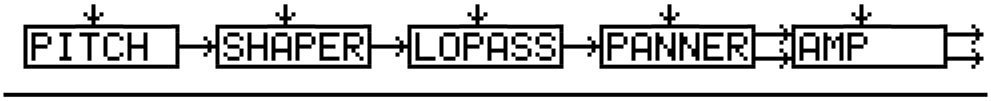
Next we'll program an envelope to change the SHAPER in real time. While still on the F1 AMT page, select the Src2 parameter, and set it to a value of **ENV2** (**1**, **2**, **1**, **Enter**). Cursor down to the DptCtl parameter and set it to a value of **AttVel** (**1**, **0**, **0**, **Enter**). Set the MinDpt parameter to **0.00x** and the MaxDpt parameter to a value of **1.70 x**. This will let you use attack velocity to control the amount of the envelope's effect on the SHAPER.

Next, use the **<more>** soft buttons to locate the **ENV2** soft button, then press it to select the ENV2 page. Set up the parameters as shown in the diagram on the following page.



```
EditProg:ENV2          [1/1]  <>Layer:1/1
Att1:Att2:Att3:Dec1:Rel1:Rel2:Rel3:Loop:
0.10 0.82 0.86 1.04 0s   0s   0s   Off
51%  -23% 42%  0%   0%   0%   0%   Inf

<more  AMPENV  ENV2   ENV3   ENVCTL  more>
```

This can still be a little harsh on the high end when you play with high attack velocities. One way to smooth it out would be to go back to the ALG page, select a lowpass filter for the F2 block, and adjust its cutoff frequency to about F$^{\#}$ 6. This is done by pressing **Edit** when the F2 block is selected, then selecting the Adjust parameter and changing the value with any data entry method.



That's it for the SHAPER example; we'll continue with this program to describe the PANNER. You'll notice that in Algorithm 13, **PANNER** is the only value available for the F3 block. The PANNER function takes a single signal from the sound engine, and splits it into two. These are referred to as the upper and lower wires. The upper and lower wires pass independently into the final Amp block, and from there to the audio outputs.

The parameters on the control-input page for the PANNER let you distribute the signal between the upper and lower wires. You can send the signal all to the lower wire (an Adjust value of-**100**%), all to the upper wire (**100**%), or anywhere in between. This in itself won't necessarily change the pan position of the current layer. It works in tandem with the Pan parameter on the layer's OUTPUT page.

When a layer uses an algorithm that contains the PANNER function, you always have two wires going through the final Amp and to the audio outputs. Consequently, on the layer's OUTPUT page, there are parameters to assign the output pair and pan position of each wire. When you have one wire panned hard left, and the other hard right, changing the parameters on the PANNER control-input page will enable you to move the layer's pan position in real time. The closer a layer's output is to the center of the stereo field, the less effective the PANNER function will be.

The first step in our PANNER example, therefore, will be to select the OUTPUT page. Select the Pan parameter for the upper wire, and set it all the way to the right. Select the Pan parameter for the lower wire and set it all the way to the left.

Now you can select the F3 POS page, and program it to move the sound around. First try changing the value of the Adjust parameter. You should hear the sound move to the left when the value is negative, and to the right when it's positive. Set it back to **0%**, then select the Src1 parameter. Select **LFO1** as the value, then select the Depth parameter and set it to **100%**.

Now select the Src1 parameter again, and press **Edit** (or use the **more>** soft button to locate the **LFO** button, then press it). You will now see the LFO page. Set the MnRate parameter for LFO1 to a value of **0.1 Hz**. Set the MxRate parameter to a value of **2.00 Hz**, and the RateCt parameter to **Data**. Leave the Shape parameter set to **Sine**. You should hear the sound shift slowly from left to right as the LFO cycles. You can adjust the speed of the shift with your MIDI controller's data slider (or a programmable control set to MIDI 06).

Example 5

Building a Drum Program; Using the Keymap Editor

With our next example, you'll learn how to build a drum program using the Program and Keymap Editors. To keep the example as brief as possible, we'll include only a few timbres and DSP examples. This won't make for a terribly realistic drum program, but it will give you the basic ideas you need to build your own. In this example, you'll create a four-layer program, with a different percussion timbre in each layer, each timbre having a different set of DSP functions applied.

Start with the default program 199. Press **Edit**, then press the **KEYMAP** soft button. Select a value of **168 Silence**. This gives you a keymap with a single key range from C 0 to G 10. Select the KeyTrk parameter, and change the value to **0**. This will make the pitches of all the samples you assign the same on each key (you won't hear anything until you assign the samples). Next, press the **<more** soft button once, then press the **DupLyr** soft button. Layer 2 will be created. Repeat this twice, until you have four layers.

When you've created the four layers, you'll notice that the top line of the display tells you that you're looking at Layer 4 of a 4-layer program. Press the **Chan**/**Bank Up** button to return to Layer 1. Press the **LAYER** soft button to select the LAYER page. Set the LoKey and HiKey parameters to **C 4** and **D 4**. The easiest way to do this is to select the LoKey parameter, hold the **Enter** button, and strike C 4 on your MIDI controller. Do the same for the HiKey parameter, striking D 4. Press the **Chan**/**Bank Up** button to select Layer 2. Set its LoKey and HiKey to **D# 4** and **F 4**. Repeat this for Layer 3, setting its LoKey and HiKey parameters to **F# 4** and **G# 4**. Do the same for Layer 4, setting its LoKey and HiKey parameters to **A 4** and **B 4**. This might be a good time to save what you've done so far.

Next, return to Layer 1 (**Chan**/**Bank** button), press the **KEYMAP** soft button, select the Keymap parameter, and press **Edit** to enter the Keymap Editor. Select the Sample parameter, and select a value of **64 15in Dry Tom-C 4**. Press the **Save** soft button, and the K2600 will prompt you to save the keymap. Rename it as **Tom**, and save it to an unused ID (don't replace **168 Silence**). Press **Exit** to return to the Program Editor, and select Layer 2 with the **Chan**/**Bank** buttons. Press **Edit** to return to the Keymap Editor, select the Sample parameter, and assign a value of **47 Dry Kick 1 C 4**. Save the keymap, renaming it **Kick**. Press **Exit** to return to the Program Editor, and select Layer 3. Return to the Keymap Editor, select the Sample parameter, assign a value of **55 Dry Snare 2-C 4**, save the keymap—renaming it **Snare**. Return to the Program Editor, select Layer 4, select the Keymap parameter, and assign a value of **42 Closed Hihat-C 4**. Save the keymap, renaming it **HiHat**. You now have a four-layer program, each layer having its own keymap with a different sample assigned to each one.

This is the basic process for creating any keymap and incorporating it into a program. In this case, we don’t want the layers to overlap, and we want each layer to use a distinct keymap with its own sample assignment. In other programs, you might want to create a keymap with different timbres in a single layer, and you might want the layers to overlap.

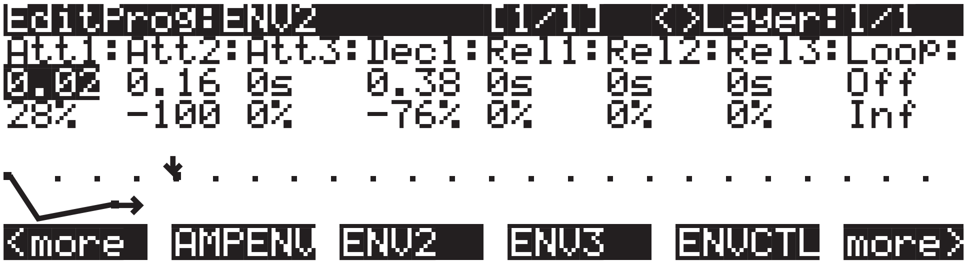
As an example of how to quickly set up a multi-sample keymap, we’ll change the sample assignment in Layer 1. Return to the Program Editor, and select Layer 1. Return to the Keymap Editor (by pressing **Edit** while the Keymap parameter is highlighted on the EditProg✱KEYMAP page), then press the **NewRng** soft button. The K2600 will prompt you to strike a low and high key on your MIDI controller. Strike C 4 and C$^{\#}$ 4. Notice that the value of the Key Range parameter changes to reflect the new range assignment. Now select the sample parameter, and turn the Alpha Wheel one click to the right, to select the sample **65 13in Amb Tom-C 4**. Save the keymap, replacing the earlier version. You can repeat this process to create as many new key ranges as you like (in this example, doing so would have no effect, since we’ve limited each layer to a narrow three-key span). Make sure the value of the KeyTrk parameter on the Keymap page is set to **100ct/key.**

If you wanted the layers to overlap, you would simply set each layer’s LoKey and HiKey parameters (on the LAYER page in the Program Editor) to the same respective values. For example, you might set each LoKey parameter to **C 2**, and each HiKey parameter to **C 7**, causing all layers to play across five octaves.

Now we’ll add some processing to some of the layers in our drum program example. The fact that each sound is on a different layer enables us to use a different algorithm for each layer, giving us enormous control over each sound.

Return to the Program Editor and select Layer 1. Press the **PITCH** soft button, and select the Coarse transpose parameter. Set its value to **-3 ST**. Select the KeyTrk parameter, and set its value to **400 cents per key**. This will give you a much different pitch on each key.

Next, select the PITCH page and select the Src2 parameter. Set its value to **ENV2** (**1**, **2**, **1**, **Enter**). Select the MaxDpt parameter and set its value to **300 cents**. Select the DptCtl parameter, and set its value to **VTRIG1** (**1**, **0**, **6**, **Enter**). Press **Edit** to select the VTRIG page, and set the VTrig1 Level parameter to a value of **fff**. Return to the PITCH page, select the Src2 parameter, and press **Edit** to select the ENV2 page. Set it up as shown in the ENV2 page below.



```
EditProg:ENV2         [1/1]   <>Layer:1/1
Att1:Att2:Att3:Dec1:Rel1:Rel2:Rel3:Loop:
0.02 0.16 0s   0.38 0s   0s   0s   Off
28%  -100 0%   -76% 0%   0%   0%   Inf
```

Next, select Layer 2 and select the F4 AMP page. Set the Adjust parameter to **8 dB** to give the kick a little more presence. Select the Src1 parameter and set it to a value of **AttVel** (**1**, **0**, **0**, **Enter**). Select the Depth parameter and set a value of **10 dB**. The kick will get considerably louder as you strike the keys harder.

Select Layer 3, and press the **ALG** soft button. Select the center, 3-stage DSP function block and assign a value of **HIFREQ STIMULATOR**. Press the **more>** soft button once, then press the **F1 FRQ** soft button. Select the Coarse Adjust parameter, and assign a value of **G 10 25088 Hz**. Select the Src1 parameter and assign a value of **MWheel**. Select the Depth parameter and set it to a value of **-10800 cents**. Move your MIDI controller's Mod Wheel to bring out the snare's high end.

Example 6

Editing a Setup for KB3 Control

1. Go to Setup mode, and select **97 ControlSetup**.
2. Press **Edit** to enter the Setup Editor.
3. Press the **more>** soft button, then press the **FOOTSW** soft button.
4. Use the cursor buttons to highlight **TapTempo** (it's the value of the Dest parameter for FtSw4).
5. On the alphanumeric button pad, press **6 8 Enter**. This changes the value to **LegatoSw**, which is MIDI 68 (for VAST programs, MIDI 68 will now control Legato Switch, but for KB3 programs, it will switch between slow and fast rotary speaker effect).
6. Press the **more>** soft button, then press the **RIBBON** soft button.
7. Highlight **AuxBend2** (it's the value of the Dest parameter for SmRib).
8. On the alphanumeric button pad, press **1 6 Enter**. This changes the value to **Ctl A**, which is MIDI 16.
9. Highlight **MPress** (it's the value of the Dest parameter for SmPrs).
10. On the alphanumeric button pad, press **1 7 Enter**. This changes the value to **Ctl B**, which is MIDI 17.
11. Highlight **AuxBend1** (it's the value of the Dest parameter for LgRib—or Sect1 if the large ribbon is configured in three sections).
12. On the alphanumeric button pad, press **1 8 Enter**. This changes the value to **Ctl C**, which is MIDI 18.
13. Press the **SWITCH** soft button.
14. Highlight **ArpSw** (it's the value of the Dest parameter for PSw1—Button 9 above the Pitch Wheel).
15. On the alphanumeric button pad, press **6 9 Enter**. This changes the value to **FrezPd**, which is MIDI 69.
16. Highlight **MIDI29** (it's the value of the Dest parameter for PSw2—Button 10 above the Mod Wheel).
17. On the alphanumeric button pad, press **7 0 Enter**. This changes the value to **MIDI70**.
18. Press **Exit**. The K2600 will ask you if you want to save the setup. Press the **Rename** soft button.
19. Press either **Chan/Bank** buttons until the KbdNaming parameter in the top line of the display shows a value of **Adv**. This lets you rename the setup by playing notes (in this

example, we’re going to rename it as **KB3 Setup**). Of course, if you prefer, you can use soft buttons and data entry (Alpha Wheel, cursor buttons, or alphanumeric button pad) to rename the setup.

20. Hold down G 6, then strike D 4 (or do the equivalent on your MIDI source). This enters a **K** as the first character in the name.
21. Hold down G 6 and strike B 2 to enter a **B**.
22. Release G 6, then press A# 2 to enter a **3**.
23. Strike G# 6 three times to delete three characters. Notice how the remaining characters shift to the left.
24. Strike G 2 to enter a space. You’re finished renaming.
25. Press **OK**. The save dialog reappears. Change the setup’s ID if you want, then press **Save**. You’ll return to Setup mode.

Example 7

Using the KB3 Setup From the Accessory Disk

1. Insert the floppy disk called **K2600 Additional Programs and Factory RAM Banks** into the K2600’s floppy drive.
2. Set the value of the CurrentDisk parameter to **Floppy**, if it isn’t already there.
3. Use the cursor buttons to highlight the directory called **MOREPRGS**, then press **Open** or **OK**.
4. Highlight the file **KB3.K26**, if it isn’t already highlighted. Press **OK**. The K2600 prompts you to select a memory bank.
5. Select a different bank if you wish, then press **OK**.
6. Press **Fill**, which determines how the K2600 assigns an ID to the file you’re loading. This loads the file into the first available ID in the bank you selected—without affecting any objects already stored in the bank. When the file has loaded, the K2600 will return to the Disk-mode page.
7. Press the **MIDI** mode button, then press the **XMIT** soft button.
8. Highlight the value of the CtlSetup parameter, then press **2 0 0 Enter** on the alphanumeric buttonpad.

Appendix A
K2600 Boot Loader

The boot loader lets you install a new K2600 operating system and/or new K2600 ROM objects from a SCSI device or the floppy drive into flash ROM. The boot loader also provides hard reset and diagnostics options.

Note: *Your K2600 comes from the factory with the operating system and ROM objects already installed. You do not need to run the K2600 boot loader to start up a new K2600.*

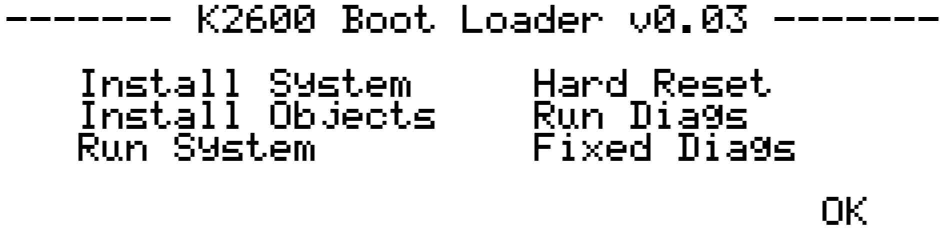
Starting the Boot Loader

When you start the K2600, it displays a “Please wait...” message and waits for approximately two seconds. Press and release the **Exit** button while the “Please wait...” message is displayed to start the boot loader. Otherwise, the K2600 will start up normally.

The boot loader’s menus resemble K2600 dialog boxes. That is, they consist of a series of options and a cursor that is used to select one of the options. You can use the cursor buttons to move the cursor up, down, right, and left. Press the **OK** soft button to invoke the highlighted menu option. You can also use the Alpha Wheel and **Enter** button to select options.

Boot Loader Main Menu

When the boot loader starts up, its main menu appears

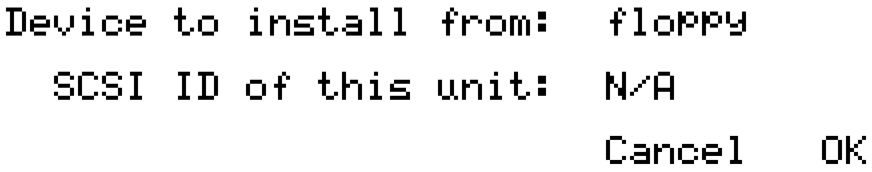


```
------- K2600 Boot Loader v0.03 -------

   Install System      Hard Reset
   Install Objects     Run Diags
   Run System          Fixed Diags

                                  OK
```

Initially, the Install System option is highlighted. After invoking various actions, however, other options may be selected by default, as the context suggests, upon returning to this menu.

Install System / Install Objects

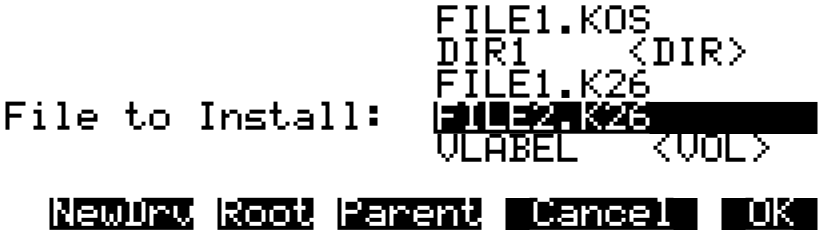
Use **Install System** to install the K2600 operating system and diagnostics files by loading them from floppy or from a SCSI device and programming them into Flash ROM. Use **Install Objects** to install ROM objects into Flash ROM. After either option is invoked, a dialog box such as the following will be displayed.



```
Device to install from:   floppy
  SCSI ID of this unit:   N/A
```

From this dialog box, you can select which device contains the system or ROM-objects file. You can use the Alpha Wheel, the **Plus/Minus** buttons, or the cursor buttons the same way you can in normal operating modes. If you choose to install from a SCSI device, the SCSI ID of this unit will change from **N/A** to **SCSI 6**. If there is another device on your SCSI bus with an ID of 6, you will need to change the SCSI ID of this unit to a free SCSI ID. At this point you can continue by pressing **OK**, or abort the operation by pressing **Cancel**. Canceling the operation returns you to the boot loader menu, with the Run System option selected. Use this option to return to normal operation.

When you select an option and press **OK**, the K2600 tries to access the specified drive and scan it for files. If it is successful, the following display appears:



```
                    FILE1.KOS
                    DIR1     <DIR>
                    FILE1.K26
File to Install:    FILE2.K26
                    VLABEL    <VOL>
```

While this page is displayed, you can use the **Up** and **Down** cursor buttons, the Alpha Wheel, or the **Plus/Minus** buttons to highlight different items in the file list. Names that have **<DIR>** displayed to their right are directories. If the disk has a volume label, it will be displayed in the list, marked with **<VOL>**. Operating system files typically have a **.KOS** extension, while ROM object files have a **.KRZ** or **.K26** extension. If you're loading a new operating system, be sure to load both files: there's an A file and a B file, both of which must be loaded. Typically, you'll load the A file first, although the order isn't significant.

The **OK** soft button selects the highlighted file for loading, or changes to the highlighted directory. The **Cancel** button aborts the installation and returns to the menu. **Parent** changes to the current directory's parent directory, while **Root** changes to the disk's root directory. The **NewDrv** button returns to the previous page, allowing you to choose a different drive from which to install.

After you select a file by pressing the **OK** button, you will be asked to confirm the installation. The K2600 will display a message of the form **Ready to install operating system from** ***filename*** **on** *device* or **Ready to install ROM objects from** ***filename*** **on** *device***.** When it displays this message, it will fill in *filename* and *device* with the name of the file and device you have selected. To confirm that you want to proceed, press the **OK** soft-button. If you do so, you will not be able to interrupt the installation. Alternatively, the **Cancel** button will return you to the menu, or the **Choose** button will allow you to select a different device or file from which to install.

While the K2600 is installing the file into Flash ROM, the top line of the display will gradually fill to indicate the progress of the operation. Note that operating system files and ROM object files are of different formats. If you try to install a file using the incorrect menu option, the K2600 will think the file is corrupt and will refuse to install it into Flash ROM. If this happens, simply choose the correct menu option from the boot loader's main menu, and repeat the installation.

If there is a problem accessing the drive or installing the file, an error message will be displayed. At this point, you can (1) abort the operation with the **Abort** soft button and return to the menu, (2) press the **Retry** button to try the operation again, or (3) choose another device or file by pressing the **Choose** button.

If you load a new operating system, you should then load all the object files corresponding to the ROM programs you have (always load the basic objects file, then load the Live-mode, Orchestral-block, and Contemporary-block object files if you have the sampling option and the Orchestral and Contemporary ROM blocks.

Run System

The Run System option executes the engine software. When this option is invoked, the operating system and ROM setups are checked for integrity. If either does not appear to be installed, a message will be displayed explaining this. If this happens, you can (1) double press the **Run Anyhow** soft buttons and be given a chance to try to run the system anyhow, knowing that the machine might crash, (2) install the uninstalled portion with the **Instal** button, or (3) return to the menu by using the **Menu** button. In the last case, the menu will be displayed with the Load System or Load Setups option highlighted as appropriate.

Hard Reset

The Hard Reset option causes all user RAM to be erased. When you select this option, a strict warning message is displayed. You must press the **Up** and **Down** cursor buttons simultaneously to actually erase the RAM. If you push any other button or turn the data wheel, the operation is aborted, and the menu is displayed with the Run System option highlighted. Otherwise, the RAM will be marked for reinitialization, and a message declaring this fact will appear prior to return to the menu. When the menu reappears, it will have the Run System option highlighted.

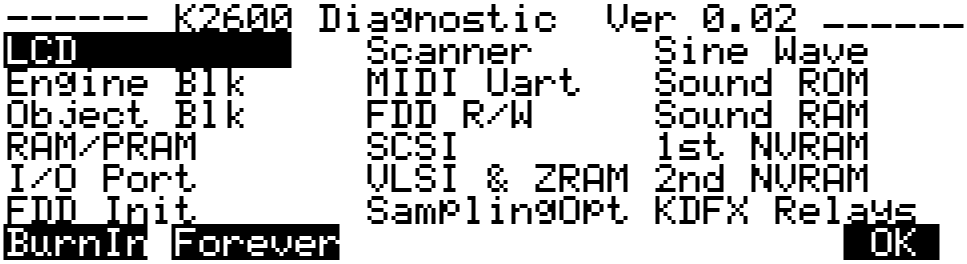
Run Diags

The **Run Diags** option first runs a diagnostic test of the system software. If there's no system software currently intalled, you'll see a corresponding message, at which pont wyou can do one of the following:

- Press **Run Anyhow** to execute the system diagnostic. The instrument may crash.
- Press **Instal** to begin the process of installing sytem software.
- Press **Menu** to return to the Boot Loader page.

After the system diagnostic finishes, you'll see a warning about erasing RAM objects during the diagnostics. If you don't want to lose any data in RAM, you should save everything before exeduting any of the diags. Press any button on the K2600's fron panel to contiue to the

Dagnistics page, where you can choose from a series of diagnostic tests. These tests can be useful in identifying problems you might be having with your instrument.



```
------ K2600 Diagnostic  Ver 0.02 ------
LCD            Scanner     Sine Wave
Engine Blk     MIDI Uart   Sound ROM
Object Blk     FDD R/W     Sound RAM
RAM/PRAM       SCSI        1st NVRAM
I/O Port       VLSI & ZRAM 2nd NVRAM
FDD Init       SamplingOpt KDFX Relays
BurnIn Forever                  OK
```

A Few Precautions

1. Turn down your sound system before beginning diagnostics!
2. Before executing the MIDI UART diagnostic, connect a MIDI cable from the K2600’s MIDI Out port to its MIDI In port. If you do not do this, the MIDI UART test will fail.

Running the Tests

Select the desired test with the Alpha Wheel or cursor buttons, then press OK to begin the test.

When you’re finished, restart your K2600 to return to regular operation.

Index