

Chapter 13

Disk Mode

Disk mode lets you load, save, back up, and copy files of objects between the K2661 and the outside world, through SmartMedia or the K2661's SCSI port. The K2661 works with 3.3v SmartMedia cards (the most common type) having a minimum size of 4 MB.

Most SCSI (Small Computer System Interface) devices will operate with the K2661 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 K2661. The K2661 will treat a CD-ROM drive like any other SCSI device (except that you can't save files to it). The K2661 can read writable CDs (CD-Rs), although it can't write to them.

The K2661 can address up to 8 Gigabytes (8 G) of hard-disk space, in 2-G partitions. This is true for any hard disk formatted with the DOS-compatible FAT-16 format. Hard disks larger than 8 G can be formatted to make 8 G (in four partitions) accessible to the K2661. You can connect up to seven hard disks. See page 13-76 for more information about Disk Partitioning.

Disk mode in the K2661 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 SmartMedia slot
- One SCSI port
- 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 reading writable CDs (CD-Rs): reading, copying, and backing up

Disk Mode Page

To enter Disk mode, press the **Disk** button, and the Disk-mode page will appear:

```
DiskMode    Samples:131072K Memory:14800K

CurrentDisk:SMedia      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. To the right of the top line you see the amount of memory available for storing all other RAM objects.

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 SmartMedia card, 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 K2661 will interact with the SCSI storage device whose SCSI ID matches the value of the Current Disk parameter. If you choose a value of **SMedia**, the K2661 will communicate with SmartMedia.

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.

Using SmartMedia Cards

You can use SmartMedia cards for all your backup and storage requirements. SmartMedia cards are sold in a variety of sizes; the K2661 will work with any size, so long as it's 4M or larger. Also, you should double-check to make sure that you always buy 3.3v cards, which is the most common type.

The SmartMedia card slot is on the back panel of the K2661, but it is easily accessible from the front of the instrument – just look for the SmartMedia logo and the blue LED. The gold contacts on the card must be facing up when you insert it into the K2661. You can remove a SmartMedia card anytime the blue LED is unlit.



Caution: Do not remove a SmartMedia card while the blue LED is lit; this can cause data corruption.

Formatting a SmartMedia Card

SmartMedia cards come formatted and ready to use with the K2661. If you ever need to format a card, however, insert the SmartMedia card (with gold contacts up) into the K2661's SmartMedia slot, or in the SmartMedia slot of any computer with SmartMedia formatting capability. Make sure the card does not have a write-protection sticker attached.

Press the **Disk** button to enter Disk mode. Make sure the Current disk parameter says **SMedia**, so you don't accidentally format any SCSI devices you might have connected! Press the soft button labeled **Format**. The K2661 will ask you if you want to format, and a pair of **Yes/No** soft buttons will appear. Press the **Yes** soft button when you are ready to begin.

The K2661 will remind you that formatting will erase the SmartMedia card, and will give you two more chances to cancel the formatting procedure—we want to make sure you don't accidentally erase any cards. Press the **Yes** soft button to continue formatting. When formatting begins, the display will tell you that the card is being formatted. The blue SmartMedia LED will light.

Connecting a SCSI Device

It's easy to connect SCSI devices to the K2661'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 K2661. 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 K2661's SCSI ports, and the other end to your SCSI device. 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*, as well as at www.kurzweilmusicsystems.com.

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 K2661 is always terminated.

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 are common sizes).

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 computer dealer for confirmation. If your SCSI system includes only the K2661 and an internally terminated computer, you're probably all set.

If you have an internally terminated computer, a K2661 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 K2661'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 K2661 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. Don't forget that the computer may have one or more internal SCSI drives; these must also be counted.

If you're planning to buy an external SCSI hard disk to use with your K2661, 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 8 to select SmartMedia. 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 K2661 in a SCSI System

SCSI IDs

All devices in a chain of SCSI devices must have different SCSI IDs, including the K2661. The K2661'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. Generally, PCs with SCSI cards will also use SCSI ID 7 for their interface.

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 SmartMedia cards, once the SCSI device is selected with the CurrentDisk parameter. The K2661 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 K2661's formatting time for SCSI disks is surprisingly short.

See page 13-76 for information about Disk Partitioning.

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 K2661 Format SCSI drives and SmartMedia cards. You can even create directories within directories; these are called subdirectories.

Directories are very useful for organizing your sample, song, and program files. The K2661 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 K2661 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 K2661 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 K2661 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 13-70). If this is set to **None**, then the K2661 will power-up in a normal fashion. If this is set to a SCSI device or **SMedia**, when the K2661 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 K2661'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 SmartMedia), or by either a "Library" or "Unspecified" designation (see *Macros* on page 13-41). 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 SmartMedia card or another SCSI disk that referenced CD-ROM files containing samples or keymaps.

Verify

Set Verify to **On** when you want the K2661 to verify saves, copies, and backups (the K2661 can't verify loads). The operations take longer, but it provides insurance against corrupted files.

Disk Drive Information

For SCSI disks, you'll see specific information about the current disk's manufacturer, model number and internal mechanism; for Smart Media cards, the manufacturer and card size are displayed. The K2661 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 K2661.

Macro On Indicator

When (Macro on) is visible, the K2661 records all file-loading operations in its macro table. See *Macros* on page 13-41.

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 K2661 memory.
Save	Save banks of objects, selected objects, or a macro as a K2661 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 K2661 disk. See <i>Deleting Files and Directories</i> on page 13-66.
Rename	Change the filename of a file on a K2661 disk. See <i>Renaming Files</i> 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 K2661 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 K2661 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 K2661 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 K2661 will ask you to wait while the device's disk starts spinning. As soon as the disk is spinning at full speed, the K2661 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
Select Root Parent Open OK Cancel
```

When you enter this dialog, the K2661 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 cards or removable hard disks), the K2661 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 K2661 has just been powered on).

The display for all disks (including SmartMedia) shows the 3-character extension of all files in the directory (except directories themselves). Extensions are created when the file is saved by the K2661. You cannot modify the extensions on the K2661. This is because the K2661 uses the extensions to tell it what kind of data the files contain.

Directories created by the K2661 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 K2661:

- .AIF** Audio Interchange File Format (AIFF)
- .KOS** Kurzweil K2500 or K2661 operating system file
- .KRZ** Kurzweil K2000 format file
- .K25** Kurzweil K2500 format file containing objects and/or sample data
- .K26** Kurzweil K2600/K2661 format file containing objects and/or sample data
- .MAC** Kurzweil K2500, K2600, or K2661 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 K2661 can read these file formats.*

When loading files, the K2661 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 K2661 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:..\HHIS\      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
Select  Root  Parent  Open  OK  Cancel
```

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 K2661 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 13-59).

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
Select Root Parent Open OK Cancel
```

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
Select Root Parent Open OK Cancel
```

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 to 1 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 K2661 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 K2661. 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 K2661 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 K2661 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 K2661 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 13-59.

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 K2661 files, whether you are using SCSI or SmartMedia. You can create directories on any disk formatted by a K2661, 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 K2661 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 13-26). 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 K2661 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 K2661 has created the directory, then the Save dialog reappears.

When you create a directory from within the Save dialog, the K2661 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
ZORK     .K26     .5K
Total:21034K
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 K2661 file system, it is time to discuss some of the features provided in the disk functions themselves.

Loading Files

The **Load** button instructs the K2661 to copy a file from the current disk to the K2661'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 K2661 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 K2661's internal RAM size.

You can select individual objects or groups of objects (samples, programs, keymaps, effects, songs) for loading from within a single K2661 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: SAMES .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 K2661 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 K2661 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 K2661 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 K2661. 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 K2661. 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:LOHU 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 K2661, 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 K2661 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: LOH0	Self: 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
Select	Next	Type Multi OK Cancel

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 13-35). 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/Right** 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 13-29). 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 K2661 will ask the following question:

Load dependent objects?

Yes **No**

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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661. It is also possible to see the following error if the K2661 object RAM is full or very near full:

Not enough memory to audition

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 most banks; the only completely empty banks are 200 and 300.

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:

```
Load this file as: 200...299*
                  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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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?

Yes No

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?

Yes No

More Load Function Features

There are more features having to do with the Load function that are described later on in this chapter, such as loading macro files and loading AIFF files.

Saving Files

The **Save** button starts the process of saving from the K2661 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
Export Macro Object NewDir OK Cancel

```

The **Macro** soft button will be present only if macro recording is turned on. (See page 13-41 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 13-28 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:Uff
Save as:          NEWFILE
Delete Insert >>End Choose OK Cancel

```

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 K2661 adds an extension (.K26) to the filename. This enables the K2661 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 K2661 (or another K2661) 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.

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 13-72
- 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 K2661’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 13-11.
- 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 13-72), **Macro** (page 13-41), and **NewDir** (page 13-11) are all explained elsewhere in this guide. This section will describe the process of saving K2661 objects into K2661 format disk files.

Saving Individual Objects

You can select any group of objects in the K2661’s RAM for saving into a single file. To save individual objects, from the above dialog, press **Object**. The K2661 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 13-14 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/Right** 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 K2661 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 K2661 will access the current disk directory and display the following:

```
Dir:\ Sel:0/10 Index: 1

Choose file name: BOTTLE .K26 48K
                  CLAU .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 K2661'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 K2661 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 K2661 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.

While the K2661 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. First of all, make sure you have enough space available (on card or disk) to hold whatever RAM samples you are saving. 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 K2661 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 K2661 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?

Names Yes No

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 K2661 at only one time: when the file is loaded. At that time, the K2661 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 K2661 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 K2661'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 K2661 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 K2661 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 K2661'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
```

```
Select Next Type Multi OK Cancel
```

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 13-17); 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
OK
```

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 K2661'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 K2661 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 K2661'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 13-35) 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 K2661, 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:LOH0    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

                                OK

```

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 K2661 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 K2661 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
Quick Access Banks Songs Velocity Maps Pressure Maps Intonation Tables	255	1–75	900–999
			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 11-15)

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

All    Type Toggle Clear Set Cancel
```

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 13-39 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

All    Type Toggle Clear Set Cancel
```

You can't use this mode with the Load function, since the K2661 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 13-40 for more information.

If you set the Select parameter to a value of Everything or SearchStrg, all other parameters disappear. In Everything mode, the K2661 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 K2661 selects every object whose name contains a user-defined string of characters. In this case, when you press **Set**, the K2661 prompts you to enter a string of characters using the alphanumeric buttonpad. Enter the characters, and press **OK**. The K2661 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:

All **Type** **Toggle** **Clear** **Set** **Cancel**

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 K2661 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 K2661 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. <i>The actual range used for selections when Toggle, Set, or Clear is pressed is taken from the setting of the StartId and EndId parameters.</i> 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 K2661 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 K2661'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 K2661 lets you create lists of disk files called macros. The files can be located on any SCSI-based disk or SmartMedia card. Files from SCSI disks in Roland and Akai format can also appear in macros. Ensoniq files are not currently supported by K2661 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 K2661’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 K2661 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 K2661’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
<more> Load Save Macro Delete more>

```

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 K2661 has three macro modes: Record, Pause, and Off.

Off There is no Macro table in the K2661.

Record A Macro table exists, and the K2661 adds all file-loading operations to the Macro table.

Pause A Macro table exists, but the K2661 *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: SMedia          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 K2661'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 K2661. 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 K2661 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 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 K2661 via the battery-backed memory.

A macro can hold as many entries as there is space for in your K2661'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:
MULTIWOX .K26*    98K
NOISE      .K26*   36K
RESONANT   .K26*  109K
SYNAPSE    .K26*  421K
Total: 664K
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 K2661, *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 K2661.

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 13-55 for more information.

Once you have pressed either **Macro** or **OK**, the loading-mode buttons appear (**OverFill**, **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 K2661 executes the **OK** or **Macro** command you entered earlier. If you had pressed **Macro**, the K2661 would add the selected objects to the Macro table. If you had pressed **OK**, the K2661 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 K2661'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 K2661 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 K2661 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 13-53.

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 K2661 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 K2661 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 K2661 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 13-57).

When the macro is done loading, you’ll see this display:

Macro SYNTH.MAC completed...

The K2661 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 K2661’s memory.

Macro Support for Multiple Removable Disks

You can use a single macro to load files from different removable disks.

1. In Disk mode, press **Load** to begin the Load dialog. You'll see a prompt asking you to select a file to load.
2. Select the macro file to load, and press **OK**. The K2661 displays a prompt asking you to select a bank for loading the macro.
3. Select a bank (or select **specified** or **Everything**), and press **OK**.
4. The K2661 begins loading files. When it tries to load a file that's not on the disk, the display shows you the name of the file that it couldn't find. Press **OK**.
5. Another prompt asks you if you want to retry loading after changing disks. Press **Yes**.
6. Change the removable disk, and press **OK**, as instructed by the prompt in the display. The K2661 resumes loading files according to the macro entries.
7. Repeat Steps 4 through 6 as often as necessary to load all the files specified by the macro. The display returns to the Disk-mode page when loading is complete.

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. Each macro entry also has a partition ID, which is recorded when you create the macro. The partition ID is part of the macro entry's name; it appears after the SCSI ID. You can view macro entries (if there's a current macro table) by pressing the **Macro** soft button while in Disk mode.

When editing a macro entry (using the Modify option on the MACRO function page in Disk mode), you can edit the partition ID along with other properties of the entry. If you change the partition ID of a macro entry, make sure it's a valid partition ID.

The following diagram shows how the MACRO page might look once four files have been recorded into the RAM Macro table:

Func:MHURU [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:		
Select	Modify	Load	Pause	Off	Exit	

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 K2661 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 SmartMedia. The letter U means Unspecified disk ID (see page 13-50). The letter L means the Library disk (see page 13-51).
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 K2661 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 your 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 K2661’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 K2661 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 K2661 lets you create macro files with entries that don't specify a Disk ID. When the K2661 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 K2661, 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 K2661 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 K2661, 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 13-53 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 SmartMedia card, 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 SmartMedia cards 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 K2661 will be able to load files directly from the CD-ROM, then load the programs from the SmartMedia, 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:SMedia          Startup:Off
                             Library:SCSI 3

<more  Load  Save  Macro  Delete  more>

```

We'll assume that SCSI 3 is your CD-ROM drive, and that you have a SmartMedia card 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 K2661 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 K2661 loads the file, and then loads `\PNOPROGS.K26` from the SmartMedia card.

If the K2661 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 K2661 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 13-53 for more information.

Although the Unspecified and Library Disk IDs are meant to be used with distributable media such as CD-ROMs and SmartMedia cards, 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
Select Root Parent Open OK Cancel
```

The K2661 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:
Select Check All OK Exit
```

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 K2661.

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\MULTIWOX.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 K2661 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:MMHURU [Record]      Index:    1

*4:\NEWMIX\TRASHX12.K26    200:F:
*4:\POTS\TEHPOT.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.

Delete OK Cancel
  
```

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?

Yes No

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	SMedia, 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:\ARCFIELD.K26            200:F:
5:\METALIC.K26             200:F:
5:\STEREO2X.K26           300:F:Obj
5:\STRINGS\DLBASS.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:MHURU [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 K2661 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?

Yes **No**

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?

Yes **No**

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 K2661 from a sequencer. This can be useful, for example, if the K2661 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 K2661 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 K2661 will display: "Send SysEx Load Macro command?" Press the **Yes** soft button and the K2661 will add the LOADMACRO SysEx command to the sequence.



Note: You cannot remotely load a macro to the K2661 while the K2661 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:
```

```
Info Find List Free Done
```

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.

A soft button labeled **Part'n** will be displayed for a partitioned hard disk (if your hard disk is 2 G or smaller, you won't see the **Part'n** soft button). Pressing this button displays a prompt asking you to select a partition (like pressing **Root** or **Parent** when you're in the root directory).

Info

Provides general information about the current disk, such as the Disk ID and formatting specifications.

The disk report also includes the total number of K2661 partitions on the disk, and which partition you're currently viewing. This information is on the last line of the second page of information.

Find

Enables you to search for files that match a certain character string in their filenames.

Free

Pressing the **Free** soft button displays the free space on the disk (or current partition). There are three components to the free-space information: percentage used, kilobytes free, and total kilobytes.

If you change partitions while the free-space information is visible, the K2661 automatically calculates and displays the free-space information for the new partition. Exiting the DiskUtil page disables this automatic updating until the next time you use the Free utility.

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.

The List utility is a convenient way to view the directories and files on your hard disk. It's also a time saver for disk operations. When you press **List**, the display shows a hierarchical listing of all the files and directories in the current partition (this can take a while, depending on the size and complexity of your directories). Directory names are followed by a backslash (\).

Use the **Up** and **Down** cursor buttons to select different objects in the list. The index number on the far right of the display's top line shows where you are in the list (each object is numbered according to its position in the directory hierarchy). Using the cursor buttons, you can quickly find files on the hard disk without having to change directories.

Changing partitions while using the List utility resets the index number to 1. In other words, no matter where you are in the list for the current partition, if you change partitions you'll start at the first entry in that partition's list.

You can use the alphanumeric buttonpad to navigate through the list. Press the index number you want to view, then press **Enter**.

Use the **Up** and **Down** *soft* buttons (not the cursor buttons) to navigate through the partition. Pressing **Down** when a directory is highlighted takes you into that directory (pressing **Down** when a file is highlighted doesn't do anything). Pressing **Up** always takes you to the parent directory of the current directory (if you're at the partition's root directory, the K2661 prompts you to select a partition.). As you've probably guessed, pressing **Root** takes to the root directory of the current partition.

Press the **Go To** button to change the current path to the highlighted directory (or to the directory that contains the highlighted file). You'll save a lot of time if you use the List utility to navigate to the desired directory before executing a command (instead of executing the command first, then navigating to the desired directory in the resulting dialog).

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:Uff

```

```

Search string: RAT_

```

```

Delete Insert >>End Choose OK Cancel

```

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 K2661 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

FindNext Go To Cancel

Found directory:

\CRATES

FindNext Go To Cancel

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 K2661 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.
```

```
OK
```

If no matching files were found, you will also see

```
(No files found)
```

The K2661 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
Root Up Down Go To Exit
```

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:
```

```
Info Find List Free Done
```

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.

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)
Select  Root  Parent  Open  OK  Cancel

```

Press **OK** when you have made your selection. Press **Cancel** to return to the Disk-mode page.

The K2661 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\

Change  OK  Cancel

```

Press **OK** to use the default.

Press **Change** if you want to select a different destination from the default shown. The K2661 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)
Root  Parent  Open  Current  SetDir  Exit

```

The move operation begins when you press either **Current** or **SetDir**.

If the default destination directory is the same as the source directory, the K2661 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 K2661 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 K2661 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
  Root  Parent  Open  OK  Cancel
```

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 K2661 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
Select Root Parent Open OK Cancel

```

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 K2661 will ask you if you're sure. Press Yes and the K2661 begins deleting the selected objects. If one of the selected objects is a directory that contains files, the K2661 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)
                                PLANKTON (dir)
Total:9040K
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?
```

```
Help Replace Increm Cancel
```

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.

Disk Mode

Backup and Copy Functions

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 0
Mode =RePlace
Source=\
Dest  =\
```

SetFile **OK** **Cancel**

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:ZYXWUITS .K26      418K
                PLANKTON      (dir)
                ANIMALS      .K26      1097K
Total:9040K     UEGGIES      .K26      2801K
                Root      Parent      Open      OK      Cancel
```

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).

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 K2661 asks "Replace existing *filename* on *destination disk*?"

```
Dir:\          Sel:0/15  Index: 1

File to copy:ZYXWUUTS .K26 418K
              PLANKTON (dir)
Total:9040K  ANIMALS .K26 1097K
Select Root Parent Open OK Cancel
              VEGGIES .K26 2801K
```

Creating a Startup File

You can create a macro file that will be automatically loaded when you power up your K2661. This file, called the startup file, or boot macro, can be on a SmartMedia card 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 K2661 when powered on. When you save the macro file, just name the file **BOOT**, and the K2661 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 SmartMedia card, set the Startup parameter to **SMedia** and make sure that you have the correct SmartMedia card in the drive when you next turn the K2661 on.

When the K2661 is powered on, it will display the following message (after the introductory VAST logo):

```
About to load startup file...
```

Cancel

The K2661 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 K2661 is turned on, by pressing the **Cancel** button.

If **BOOT.MAC** is found by the K2661, 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 K2661 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 K2661 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 K2661’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 K2661. 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 K2661 is compatible with fixed and removable disk drives that use the FAT-16 DOS 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. See page 13-76 for more information about Disk Partitioning. Working from the K2661 front panel with an MS-DOS formatted disk will appear the same as working with a disk that has been formatted with the K2661'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 K2661 format. If the **(DOS)** indicator is displayed, it means the K2661 has determined that the disk is a DOS-format hard disk with at least one primary DOS partition.

Some advantages of working with an MS-DOS compatible disk format over the standard K2661 format are:

- Easier sharing of K2661 files with other users over computer communications lines
- Ability to use graphical file management interfaces for organizing files and directories
- Ability to back up K2661 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

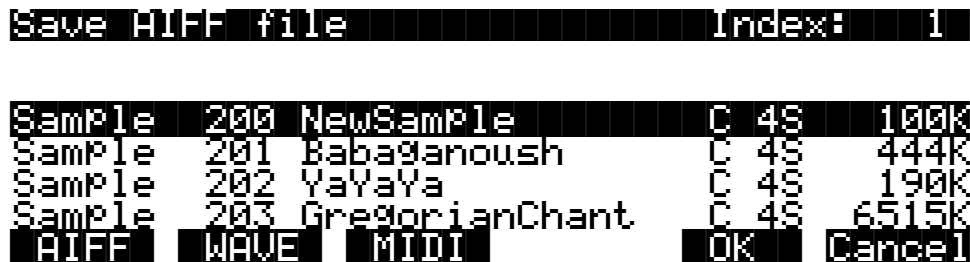
DOS format does not support space characters in filenames. The K2661, though, allows spaces to be used within filenames. If you plan to transfer files between the K2661 and a DOS compatible computer, it is recommended that you use only filenames without space characters in them. Otherwise, a computer may have trouble identifying the files.

Importing and Exporting Data Using Standard File Formats

The K2661 supports three common data interchange file formats, Audio 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 K2661 can recognize these file types automatically on loading, regardless of the file extension. You can load these files as you would any standard K2661 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**.



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 K2661 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 K2661 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 K2661 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 K2661 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 K2661. The K2661 will save 16-bit AIFF files, either mono or stereo.

WAVE Files

The K2661 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 K2661 reads and writes MIDI Type 0 (single-track) and Type 1 (multi-track) files. The K2661 supports all musical timestamp resolutions, and automatically scales imported information to the K2661'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 K2661 will load samples from Akai, Roland, and Ensoniq EPS and EPS-16 Plus SCSI disks (including ASR-10 “Ensoniq” format), using the Load operation. 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 K2661 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 K2661 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 K2661 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: **HDdrive**, **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 **HDdrive** 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 **HDdrive** 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 K2661 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 K2661 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 K2661 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 K2661 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.

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.

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.

Disk Partitioning

The K2661 can address up to 8 G of hard-disk space, in 2-G partitions. This is true for any hard disk formatted with the DOS-compatible FAT-16 format. Hard disks larger than 8 G can be formatted to make 8 G (in four partitions) accessible to the K2661.

When properly partitioned, the K2661 files on your hard disk should be accessible to both the K2661 and your personal computer, if you're using one. Using a personal computer makes it much easier to manage your K2661 files. It also enables you to use other disk management tools on your K2661 partitions—for example, disk defragmenters and system integrity checkers.

We strongly recommend that you partition your hard disk from the K2661, although it's possible to partition it from a PC (see page 13-78).



***Note:** Keep in mind that if you partition a hard disk, K2600 series instruments without v2.0 (as well as K2000s and K2500s) will be able to use only the first partition of that disk.*

Once you've connected a hard disk drive to the K2661's SCSI port, you're ready to begin partitioning. If you need information about using your K2661 and a PC together in a SCSI chain, see the following sections in this chapter:

- Connecting a SCSI Device 13-3
- SCSI Termination 13-3
- Using your K2661 in a SCSI System 13-4

Partitioning is part of the formatting procedure. When you format a disk from the K2661, you can choose whether to add partitions. When partitioning a disk, the K2661 creates as many 2-G partitions as possible, to a maximum of 8 G (even if your disk is larger than 8 G, the K2661 formats only the first 8 G).



***Note:** If you happen to have a 2-G hard disk, you might wonder whether it's worth the effort to partition it. If you want the disk to be fully DOS/Windows-compatible, you should partition it. If DOS/Windows compatibility isn't important, partitioning isn't necessary (for example, when you know you'll never use the hard disk with a computer).*

The last partition includes whatever disk space (up to the 8-G limit) remains after the last full 2-G partition is formatted. The last partition is usually slightly less than 2 G, because each partition contains some data for disk management. For example, if you format a new 10-G hard disk from the K2661, you end up with three 2-G partitions, and a fourth partition of slightly under 2 G. The K2661 doesn't format the remaining 2 G. You may be able to use a personal computer to format the remaining space for use with that computer. See *Partitioning Large Disks* on page 13-79.



***Note:** There are two schools of thought about describing the size of storage devices. Some people think of a kilobyte as 1,000 bytes, and others think of it as 1,024 bytes (which is the actual number). If you consider a kilobyte to be 1,000 bytes, then a gigabyte is 1,000³, or a billion bytes. If you think of a kilobyte as 1,024 bytes (as we do), a gigabyte is 1,024³, or 1,073,741,824 bytes. So a 2-G partition stores about 2.15 billion bytes, which you might prefer to think of as 2.1 G.*

Doing a Hard Format

There are two kinds of formatting; we'll call them normal formatting and "hard" formatting. Normal formatting is what usually happens when you format a disk. Although normal formatting makes an entire hard disk accessible for data storage, it doesn't actually *delete* files; instead, normal formatting deletes underlying system-level data about how and where the files are stored. Although this makes the files on the disk unusable, much of the files' contents remain on the disk (that's why you can often recover a file even after you've deleted it).

Hard formatting physically erases every bit of data on the disk. Once a disk is hard formatted, there's no data to recover; it's completely blank, and must be formatted normally before it can store data again. Hard formatting can sometimes make a corrupted disk usable, or make it easier to use a disk on a different operating system.

The *first* time you format a disk for use with the K2661—whether it's new or previously formatted with a computer—you should begin with a hard format, then format it normally with the K2661. (You should do this because any residual file data left by normal formatting could cause the K2661 to miscalculate partition boundaries.)

Any time you're using a PC or a Mac to format or reformat a disk for use with the K2661, you should start with a hard format to remove residual file data.

When using the K2661 to reformat a disk or partition that's been formatted *with the K2661*, you can probably skip the hard format, unless you're having problems saving and loading files.

The time required for hard formatting varies widely depending on the hard disk. Newer drives in the 10-G range will generally take from 30 to 60 minutes, although your results may differ.

1. Press the **Disk** mode button to enter Disk mode.
2. Select the Current Disk parameter, if necessary, then select the SCSI ID of the disk you want to format.
3. Press the **Chan/Bank** buttons simultaneously. You'll see several prompts asking you whether you want to do a hard format.
4. Answer **Yes** in each case. The display informs you when formatting begins. When it's finished, the display returns to Disk mode.

The Partitioning Procedure



Caution: *This procedure destroys all data files on the disk. Before formatting, make sure to back up any data that you want to preserve.*

1. If you haven't already done a hard format, do so now, as described in the preceding procedure. Doing a hard format is important for ensuring data integrity. When the hard format is complete, the K2661 returns to Disk mode, and you can proceed with Step 2.
2. Press one of the **more** soft buttons until you see the **Format** soft button. Press it. The display prompts you to decide whether to format the disk. Press **Yes**.
3. The display prompts you to choose whether to *partition* the disk. Press **Yes**.
4. The display prompts you with a reminder that formatting erases everything on the hard disk, and asks whether you want to continue. Press **Yes**.

5. The display shows **Formatting Drive X with N Partitions**. X is the SCSI ID of the disk, and N is the number of partitions that the disk will contain. This is the last chance to abort the formatting process. If you do a soft reset while this message is visible, you'll cancel the format. Each partition takes about 30 seconds to format.

When Disks Are Already Partitioned

If the hard disk you've selected for formatting is already partitioned, the display you see at Step 2 prompts you to decide whether to reformat the entire disk. Press **Yes** if you want to reformat the whole disk, then proceed with Step 3.

If you don't want to reformat the entire disk, press **No** at Step 2. The display prompts you to decide whether to format a single partition. Press **Yes**, and the display prompts you to select a partition to format. By the way, this is a quick method for deleting all the files in a single partition.

Partitioning With a Computer

You can partition disks for the K2661 using a DOS-capable computer that's compatible with the FAT-16 file format. This works well on DOS and Windows machines (consequently that's the focus of this description). We don't have any reliable recommendations for partitioning disks with a Macintosh.[®] Nevertheless, there's some Mac-specific information beginning on page 13-82.

Partitioning with a PC gives you greater flexibility in defining the number and size of partitions on your hard disk. Unless this flexibility is extremely important to you, however, we strongly recommend that you use the K2661 to partition your hard disk for use with either a PC or a Mac. Our tests show that this provides the greatest cross-platform compatibility.

For example, partitions created by the K2661 are visible to any PC that's compatible with the FAT-16 file format. These partitions are also visible to any Mac running System 8.5 (or later), and version 3.0.2 (or later) of the File Exchange extension. On the other hand, partitions created on a PC are less likely to be accessible on older Macs.

Formatting from the PC requires **FDISK**, a disk-partitioning utility that's built into Windows 95 and Windows 98.

1. If you haven't already done a hard format, do so now, as described on page 13-77. Doing a hard format is important for ensuring data integrity. When the hard format is complete, the K2661 returns to Disk mode, and you can proceed with Step 2.
2. On the PC, start **FDISK** by typing **FDISK** at a DOS command prompt (or in the Run menu). **FDISK** prompts you to decide whether to enable support for large hard disks.
3. Respond with **N** (No), because you don't want support for large hard disks (although **FDISK** doesn't mention it by name, large-hard-disk format is *FAT-32*—answering **No** specifies *FAT-16* format, which is compatible with the K2661).
4. Use Menu Option 5 to select the drive to format.
5. Use Menu Option 1 to create a primary partition. Specify a size up to 2 G.
6. When the primary partition is complete, choose Option 2 to create an extended partition. At the prompt, select any size (up to the 8 G limit; including the primary partition), or select the default size to use the remainder of the drive for the extended partition. Press **Escape** to begin partitioning. When the partition is complete, you'll have the opportunity

to create one or more logical drives within the extended partition (as described in Step 7). The K2661 recognizes each of these logical drives as a separate partition).

7. When the extended partition is complete, you'll see a prompt asking you to specify the size for a logical drive. Specify a size up to 2 G, then press **Escape**. When the logical drive is complete, the prompt returns. Repeat this step until you've used the remaining space in the extended partition. Each of these partitions (logical drives) gets a drive letter as you create it. Keep track of these, because you'll need to format each partition. At any time in this process, you can use FDISK Menu 3 to display the current list of partitions.
8. When you're finished partitioning, press escape several times to exit **FDISK**.
9. Restart the computer.
10. Format each of the new partitions (using the drive letters you created with Steps 5 through 7). You can do this from the DOS prompt, or from Windows Explorer.

Partitioning from a PC is more time-consuming, but gives you more control over the number and size of your partitions.

Partitioning Large Disks

The K2661 can address a maximum of 8 G of disk space, so there's not much point in connecting a larger disk to the K2661—unless you like to do a bit of hacking, in which case you may be able to use a personal computer to format the extra space for use with that computer.

There are several disk-formatting utilities available for the Mac and PC. You may be able to use one of these (including **FDISK** for the PC) to format a large hard disk with 8 G of space for the K2661, and additional space that's accessible to your computer.

Working With Partitions

Once your hard disk is formatted and partitioned, you should verify each partition before attempting to store files. The quickest way to do this is to execute a Load command for each partition. If the K2661 can read the disk (even though there are no files), the partition is OK. The display informs you if there's a problem addressing the partition.

Once you've verified each partition, you can start using the disk with your K2661, and optionally with a personal computer. If your disk is attached to the K2661 and nothing else, you'll interact with the hard disk exclusively through Disk mode. If you're in Disk mode and you can't get to a partition, try reformatting the partition (see *When Disks Are Already Partitioned* on page 13-78). If that doesn't work, do a hard format on the disk, then format and partition it again.

If your hard disk is also attached to a computer, the computer should be able to address each partition. PCs can address each partition as an individual drive (at a DOS prompt or via icons in Windows), while Macs display a file icon for each partition.

Having a computer hooked up to the hard disk gives you another way to verify the formatting and partitioning of the disk. Look at the disk from the K2661, then from the computer. If they don't have the same partition data (number of partitions, size of each partition), you could have a formatting problem. In this case, you should start over by doing a hard format, then proceeding to formatting and partitioning.

Managing a SCSI Chain that Includes a Computer

If you create a SCSI chain that contains both a K2661 and a computer, you'll need to use caution to prevent file corruption. We'll explain how this risk of file corruption occurs, then describe how to avoid it.

When you boot up your computer, it addresses the SCSI devices to which it's connected, and attempts to mount every SCSI device whose format it recognizes. While mounting a disk, the computer examines the disk's file allocation table (FAT). The FAT contains the locations of all the data on the disk. Consequently, the FAT gets updated every time you write data to the disk. Here's where the risk of file corruption occurs.

When you write to your hard disk from the K2661, the K2661 updates the disk's FAT. But there's no way to instruct the computer that the FAT has been changed, so the K2661 and the computer have different versions of the disk's FAT. If you write to the disk from the computer at this time, you run the risk of writing over the data you just stored from the K2661.

You can eliminate this risk by ensuring that the K2661 and the computer have the same version of the disk's FAT before writing from the computer. The most foolproof way to ensure this is to adopt the following rule: *Whenever* you write to the hard disk from the K2661, unmount and remount that disk before writing to it from your computer. If it's a removable-media drive, ejecting the disk and reinserting it will update the computer's version of the FAT.

You might be able to unmount and remount the hard disk from the desktop, or using a software utility—or you might have to restart the computer. If you're using Windows, you can probably designate the hard disk as a removable-media disk (as described below), which is convenient because Windows machines always read the FATs of removable-media disks before writing to them.

Designating a Disk as a Removable-Media Disk (Windows only)

1. On your PC, run the Device Manager.
2. Expand the Disk Drives category.
3. Double click icon for the K2661's internal hard disk to bring up the Properties dialog.
4. Select the **Settings** tab.
5. Check the **Removable** checkbox.

Configuring Device Drivers (Windows 95 and 98 Only)

PCs running Windows 95 or Windows 98 require a SCSI driver for every device in a SCSI chain. There is no SCSI driver for the K2661, because it's not required for normal SCSI operations. Consequently, the first time you boot up your PC after connecting it to the K2661 via SCSI, you'll need to do some configuring to enable the PC to recognize the K2661. This configuration isn't required, but it enables the PC and the K2661 to interact more smoothly.

A Bit of Background

When you start up your PC, Windows scans the PC's SCSI port(s) for connected devices. For each SCSI device it finds, Windows checks a set of entries in the Device Manager, to verify that the appropriate SCSI drivers are installed. Windows 95 and 98 require that the Device Manager contains an entry for each SCSI Logical Unit of each SCSI device. The K2661 has eight possible SCSI IDs (0–7), so your Device Manager should contain eight entries corresponding to the K2661. Each of these entries must specify a valid SCSI driver—otherwise Windows will report one or more unknown devices as it boots up.

The K2661 requires no SCSI drivers for disk operations, but Windows 95 and 98 require them nonetheless. You can satisfy this requirement by specifying a generic “Unsupported Device” Windows driver for each of the K2661’s entries in the Device Manager. The following procedure describes how.

The Configuration Procedure

We’ll assume you’ve already connected your K2661, your PC, and your hard disk in a SCSI chain (with whatever SCSI termination is required), and that everything is powered down.

1. Power up the hard disk. Wait until it’s fully up to speed.
2. Power up the K2661. Wait until it reaches Program mode.
3. Power up your PC. During the boot-up process, Windows detects the K2661 as an unknown SCSI device. The Update Device Driver Wizard appears, prompting you to specify a SCSI device driver to install.
4. Make sure that the PC’s floppy and CD-ROM drives are empty and click **Next**—that is, proceed without specifying a SCSI driver.
5. The PC displays a dialog informing you that Windows did not locate a driver for the SCSI device. Click **Finish** to proceed without installing a driver.
6. Repeat Steps 4 and 5 seven times—once for each of the K2661’s remaining SCSI Logical Units. When you finish the last dialog, Windows finishes booting up.
7. Start the Device Manager (in the Start menu, select **Settings**, then **Control Panels**).
8. Select the **System** icon in the Control Panels window.
9. Select the **Device Manager** tab in the System Properties window.
10. Click the icon for **Other Devices**. You’ll see the eight entries for the K2661. Note the exclamation marks, indicating problems with the corresponding device drivers.
11. Double-click on one of the entries. This opens the Properties window for that entry.
12. Click the **Drivers** tab. The status for this entry indicates that the driver isn’t working properly, and that the driver files either are not required or haven’t been loaded.
13. Click **Update Driver**. This opens the Update Device Driver Wizard, which asks whether Windows should search for the driver.
14. Select the option **No, select driver from list**, then click **Next**. A list of driver types appears. The default is **Other devices**.
15. Click **Next** to accept the default. A list of devices appears. The default is **Unsupported device**.
16. Click **Finish**. This “installs” the driver and returns to the Properties window for the current entry.
17. Click **Close**, which returns you to the Device Manager.
18. Repeat Steps 11 through 17 for the remaining entries.
19. Close the Device Manager and Control Panels window.

There's a page on our website where you can find more information about including a K2661 and a PC in a SCSI chain:

http://www.kurzweilmusicsystems.com/html/scsi_help.html

If a Partition is Inaccessible

On the PC

You're not likely to have problems addressing K2661 partitions from your PC, especially if you created those partitions on the K2661 itself. If you can't mount a partition on your PC, try reformatting the partition, using the K2661. If that doesn't work, do a hard format, then format and partition the disk again.

On the Mac

System 8.5 or later—when used with the File Exchange extension—supports DOS partitions, so you should be able to view partitions as file icons. If they aren't visible, try reformatting the partitions using the K2661

If you're running an operating system older than 8.5, your Mac will probably not be able to address the last partition on the disk. If you have the Formatter 5 utility from Software Architects, you may be able to make the last partition visible (we can't guarantee that this will work):

1. Start with a hard format (see the procedure on page 13-77).
2. Format the disk with Formatter 5 (see the Formatter 5 instructions).
3. Use Mac/PC manager (a utility provided with Formatter 5) or DOS Mounter 95 (also available from Software Architects) to view the partitions.

Other Macintosh Issues

There are several things to keep mind when attempting to use a Mac with disks formatted for the K2661.

First, we recommend using MacOS 8.5 or later, and version 3.0.2 (or later) of the File Exchange system extension. This combination has given the best results, especially for disks formatted with the K2661. With this configuration you can read most partitioned DOS disks, but you can't format them.

On Macs running operating systems prior to 8.5, you can use Formatter 5 (and its accompanying extension, Mac/PC manager). these work reasonably well for formatting and reading small disks. Results are unreliable for disks larger than 2 G, however. Formatter 5 often has trouble reading any partition but the first, even on disks formatted with the K2661. You might also try DOS Mounter 95, also available from Software Architects.

Partition Support in Disk Mode

Basic Operations

This is how the Disk mode page looks when you're viewing a hard disk. Note the partition ID (0, which indicates the first partition), which is included in the path information. The partition ID for unpartitioned disks is always 0. There's no real limit to the number of partitions your disk can contain; the K2661 should be able to read all of them. Nevertheless, 20 or so partitions is a reasonable maximum.

```

DiskMode                               Memory:478K
Path = 0\PROGS\TRIPLES

CurrentDisk:SCSI 1                     Startup:Off
                                         Library:Off
Direct Access, 10000MB                 Verify :Off
MegaDisk 10                            J.02
<more  Rename  Move  Util  NewDir  more>

```

The K2661 records path information up to the disk level. In other words, the K2661 tracks your movement within directories and partitions (but not from disk to disk). As long as you're addressing the same disk, the K2661 has all the information it needs to load or save files (or any other Disk-mode command).

When you select a different disk, the first command you execute causes the K2661 to prompt you to select a partition. Once you do so, you won't see the prompt again until the next time you change disks and execute another command.

Navigation

Many disk-operation dialogs (the Save dialog, for example) include file lists, where you select the file to save, delete, copy, or whatever. Any time you're navigating through the directories in a file list, you can select a different partition.

1. Press the **Root** soft button to select the root directory for the current partition. The path information in the top line of the display will look something like **Dir:0**. The 0 (or some other numeral) is the partition ID, and the backslash (\) denotes the root directory. (You can also press **Parent** one or more times to get to the root directory.)
2. Press **Root** or **Parent** to display the partition selector, which prompts you to select a partition to mount.
3. Select a partition ID, then press **OK**.

The quickest way to get to the partition selector is to press **Root** twice in succession. This brings you to the partition selector from anywhere in the file list.

Backups and Copies

You can use the Backup and Copy operations to copy files to different partitions on the same disk, or even to some directories within the same partition.

You can always back up or copy files to a different partition, regardless of location. If you're backing up or copying to the same partition, there's a restriction: you can't back up or copy to the same directory, or to a subdirectory of the current directory (you *can* back up and copy to parent directories, however).

For example, suppose you had several program files in the root directory of a partition, and you wanted to copy some of them to a directory called PROGS, also in the root directory. You couldn't use Backup or Copy to make copies of the programs in the PROGS directory, but you could use Move to *move* them into PROGS, then use Backup or Copy to make copies of them in the root directory.

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 K2661. 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 K2661 sampling option. Even without the sampling option, however, you still have access to all of the sample editing features covered later in this chapter. Samples can be loaded from disk, or dumped into the K2661 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* in 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 K2661'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 will be sampling from a digital source in AES digital format (either AES/EBU or S/PDIF), connect the input cable to the AES/SPDIF In jack in the sampling section of the rear panel. This jack is covered by a small plug which is easily removed before connecting the cable. This plug should be left in place whenever the optical input is not in use, since dust and 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 K2661'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 K2661 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 K2661. The best results will be achieved by sampling from a digital source, using one of the K2661'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
Mode :Stereo
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 K2661's sampling inputs. Use a value of **Int** if you want to sample the K2661'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 K2661'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 K2661 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 K2661’s transposition range during sample playback. The K2661 transposes samples by changing the sample playback rate; the higher the playback rate, the higher the pitch of the sample. The K2661 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 K2661 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 be able to add more sample memory to your K2661 (if it’s not already maxed out at 128 MB). 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 K2661 Sample Memory* in the *K2661 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 the most common sample RAM configurations and their total sample time capacity (in seconds) at various sample rates.

Total RAM	Sampling Mode	Sampling Rate in KHz				Total Sampling Time (min:sec)
		29.4	32.0	44.1	48.0	
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 K2661’s sampling inputs (assuming you have the Sampling Option), and use that input just like a regular VAST program (the input goes through a DSP algorithm, then through KDFX, then to the audio

outputs). Set Mode to LiveIn to use Live mode. See page 14-42 for more information about Live mode.

Threshold (Thresh)

The Thresh parameter controls when the K2661 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 K2661 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 K2661 calculates this automatically, and limits the maximum value of the Time parameter accordingly. At a value of **0** for this parameter, the K2661 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 K2661 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 K2661 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 14-5.

If you don't have enough free program RAM, you may be unable to audition samples, since the K2661 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 K2661'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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 13-35.
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 K2661 assumes you want to preview them all.
3. Press **OK**. If you've selected more than one sample, the K2661 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 K2661 returns to the Bank dialog, where you can select a different bank if you want, then press **OK**. The K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661's Output

You can sample the K2661'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 K2661's “sample-while-play” capabilities offer a number of useful possibilities. It allows you, for example, to create composite sounds made up of several K2661 sounds or even sequences. This can help you make efficient use of the K2661's polyphony. By building composite sounds from other composite sounds, you could actually cause a frighteningly large number of K2661 sounds to become a single sample. The only constraints are your imagination—and the amount of sample RAM installed in your K2661.

You can also sample the K2661 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.

```
SampleMode Samples:131072K Channel=2
Sample:None Src:Ext
Input :Digital Time:1s

Format:AES/EBU
Mode :Stereo
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 K2661 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.

Format

Use the Format parameter to tell the K2661 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 K2661 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 K2661 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 K2661 still samples the input correctly. You might need to adjust the coarse tune to get the proper root pitch.)

Src

The possible values for the Src parameter are Internal (Int) or External (Ext). Choose a value of **Ext** if you want to sample the signal from an external source that's connected to the AES/SPDIF In optical jack. Use a value of **Int** if you want to resample K2661 internal audio data. Note that the digital internal sampling source corresponds to output A from KDFX. Other outputs will not be sampled digitally.

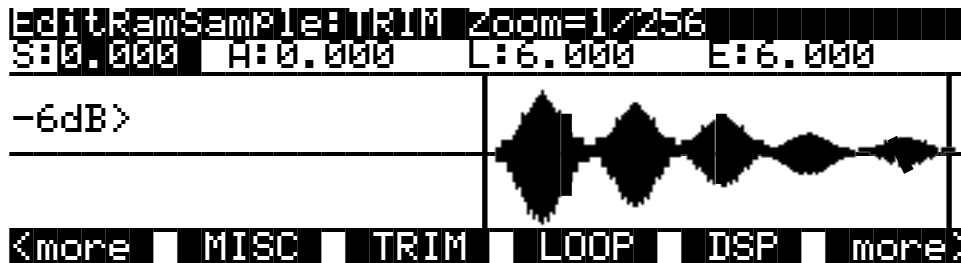
If you are resampling digitally, you should be sure to set the Digital Output Length parameter on the Master2 page to 16 bits (see page 11-11). This is because the sampler is only 16 bits, and you want to make sure that the signal is properly dithered before resampling.

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.



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 K2661 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 K2661 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 K2661 splits it into left and right sides. If it's a block of samples, the K2661 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 13-35.
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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 will ask you if you want to copy the sample data. If you answer **Yes**, the K2661 will make a separate copy of the sample. If you answer **No**, the K2661 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 K2661 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 K2661, 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.

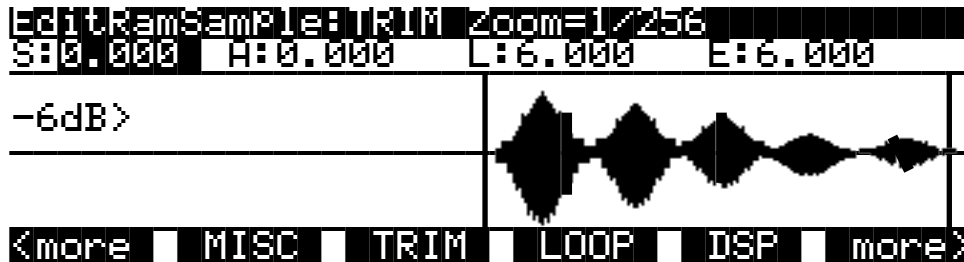
Sample Size and Sample Rate

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.



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.

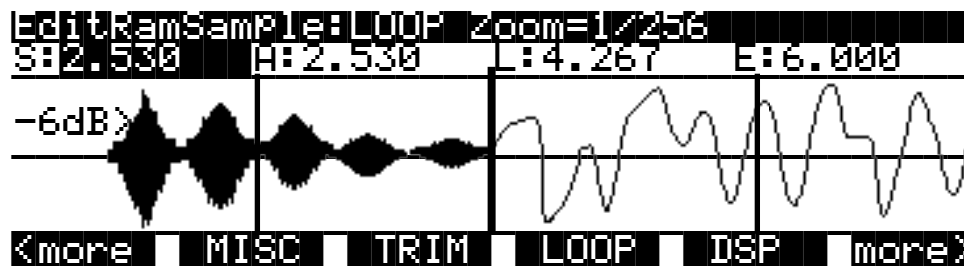
The End (E) parameter sets the point at which the current sample will stop playback. Typically you'll use this parameter to trim unwanted silence off the end of a sample, although you can use it to shorten a sample as much as you want.

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.



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 K2661 will search for the next zero-crossing to the left. Press the **Plus** button again to search toward the right.



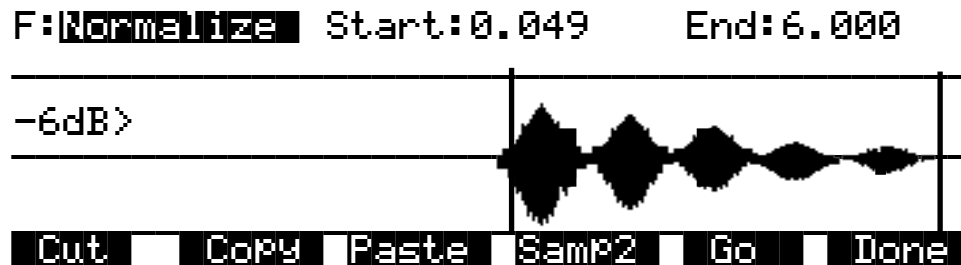
***Note:** The Gain affects the sensitivity of the zero-crossing search algorithm. The higher the gain, the more zero crossings it will find. At low gain settings, the algorithm will fail to find many or even all zero crossings.*

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.



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 K2661 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 K2661 is shut off. If you accidentally 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 K2661 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 K2661 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 K2661 will prompt you to keep

the changes. You can audition your changes and decide whether to keep them, then press **Yes** or **No**.

Done

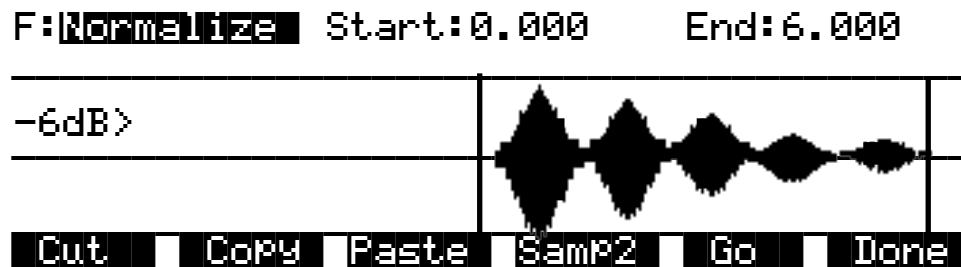
Press the **Done** soft button to return to the previously selected Sample-editor page when you're finished with the DSP functions.

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.

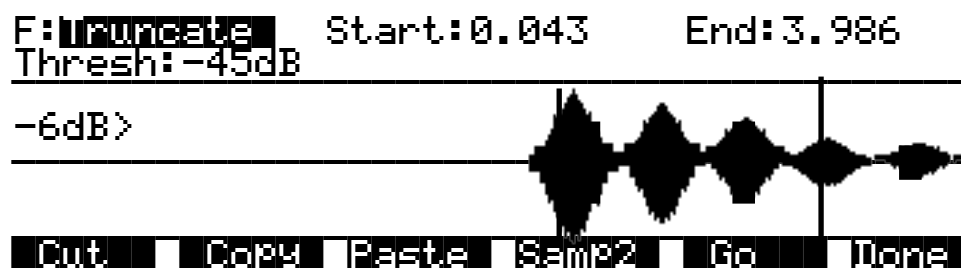
Remember that the DSP functions operate only on RAM samples. You'll notice that if you're editing a ROM sample, there is no **DSP** soft button; instead, there's an extra **Link** soft button for your convenience.

1 Normalize



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

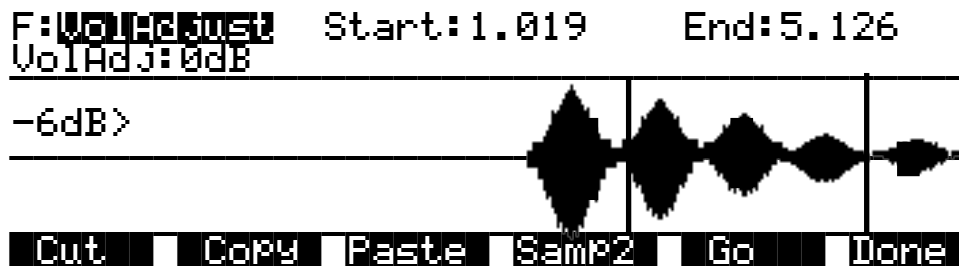


The Truncate function will automatically reset the Start (S) and End (E) points of the sample. This can be quicker than trimming the sample manually on the TRIM page.

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 K2661 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**.

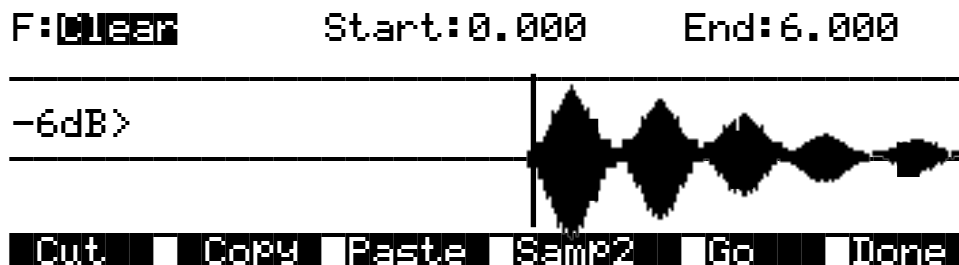
3 Volume Adjust



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 K2661, 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.

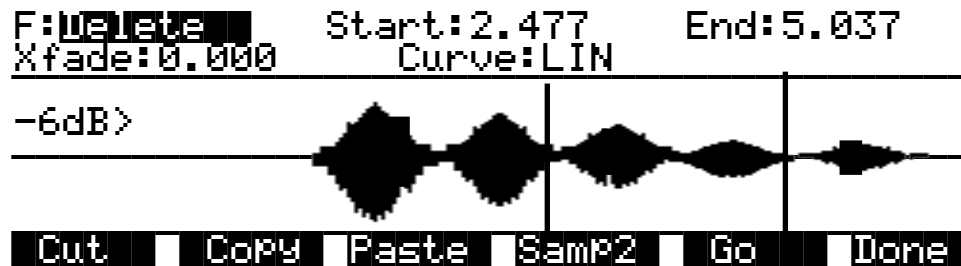
When you've selected the range to be adjusted, select the VolAdjust parameter with the cursor buttons, and use the Alpha Wheel to adjust the volume of the selected range. You can cut/boost the volume from -96 to 96 dB.

4 Clear



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

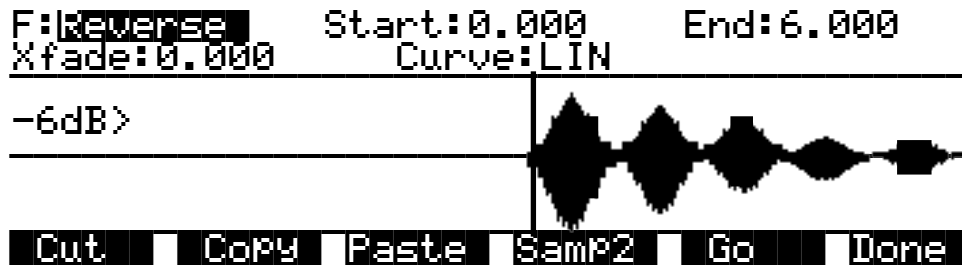


Unlike the Clear function, this will erase the samples within the selected range and shorten the sample, like cutting a section out of a tape and splicing the ends. If you want to silence a segment of the sample without shortening it, use the Clear function.

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.

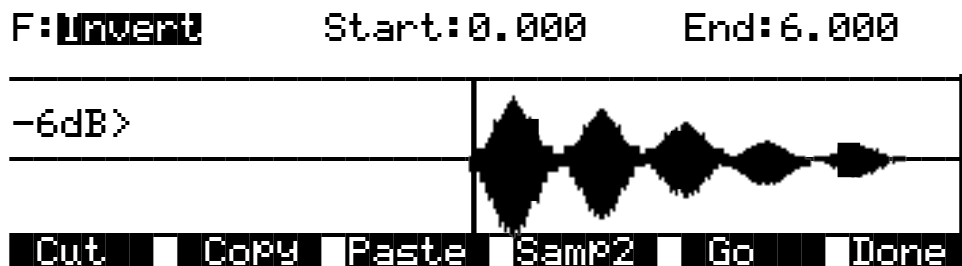
The Curve parameter lets you choose from a variety of crossfade curves that affect the nature of the crossfade. A value of **LIN** gives a straight linear curve that fades uniformly from start to finish. A value of **EXP** sets an exponential curve that starts gradually and steepens toward the finish. 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

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

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



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



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.

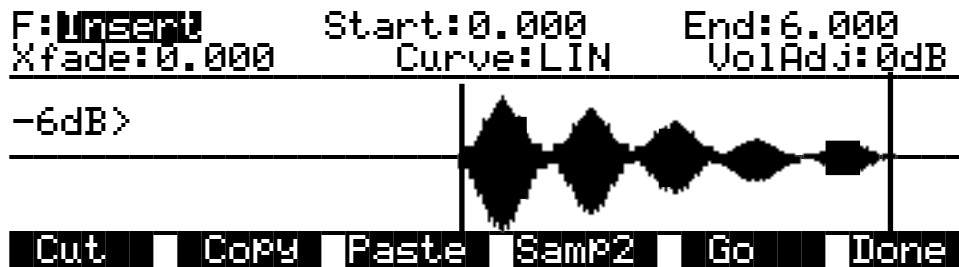
Use the In and Out parameters to specify the length of time it takes Sample 2 to reach full amplitude and to fade to silence. The Curve parameter selects the curve of the fades. The available values are **LIN**, **EXP**, **COS**, **EQL**, and **MIX**. These curves are described on page 14-34.

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 K2661 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

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 K2661 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

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)

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



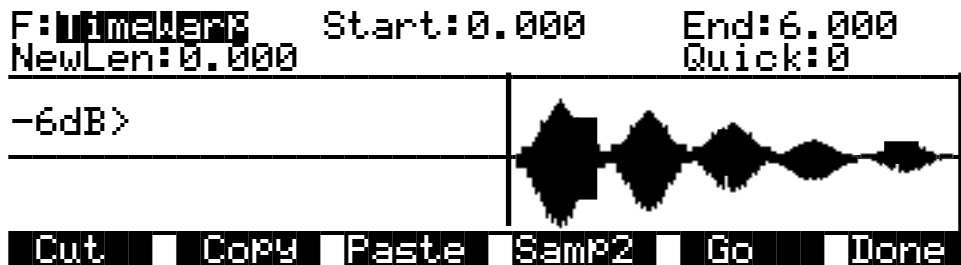
Use this function to change the sample rate of the samples in the selected segment. This is convenient for converting samples to new rates matching those of other samples to be mixed with or inserted into. It's also useful for saving memory and altering the timbre of a sample.

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 K2661 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 K2661 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



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



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



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.

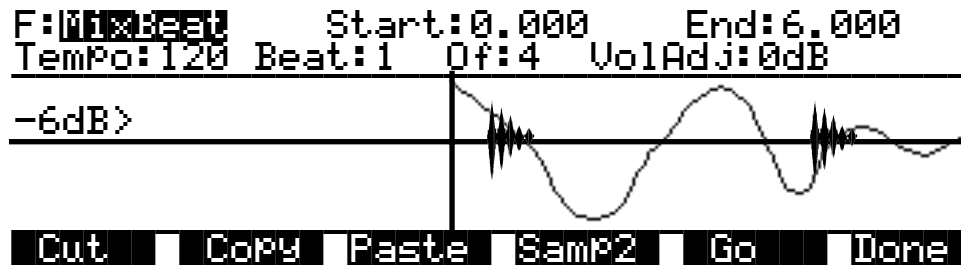
Now set the values of the Beat and Of parameters to determine how the mixed sample will repeat. The Beat parameter determines the beat (s) on which Sample 2 will be mixed—from 1 to 9999. The Of parameter determines the measure length—also from 1 to 9999.

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.



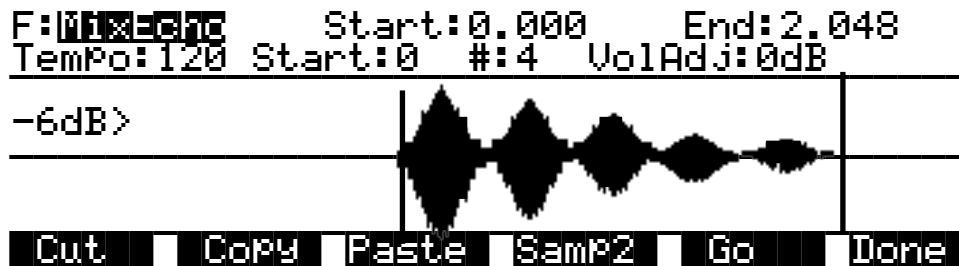
If you change Beat to 3, Sample 2 will play on beats 3, 7, and 11. If you set Beat to 1 and Of to 6, Sample 2 will play on beats 1 and 7. If you set Beat to -1, Sample 2 will still play on beats 1 and 7, but the sound of the sample will change each time, as the start point of the Sample 2 segment moves 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



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

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.

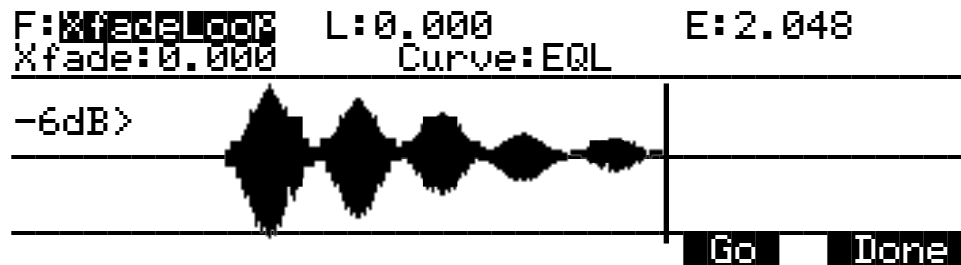
If you set the # parameter to a positive value, the adjustment in volume begins on the second mixed-in beat (the first beat plays at the initial level). If you set a negative value, the first mix of Sample 2 will be cut or boosted by the Vol adjust amount.

19 Beat Volume Adjust

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)



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



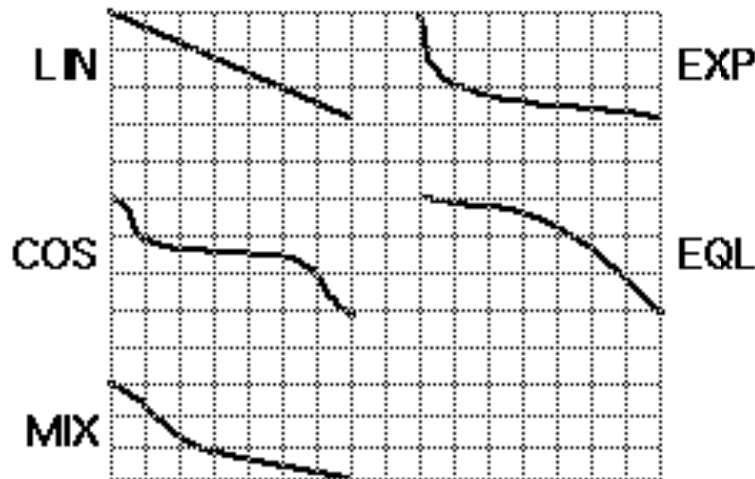
The Dynamics function is a general-purpose compressor with a few features that make it suitable for creating smooth sample loops. First, while most compressors set a threshold relative to 0 dB, the Dynamics function sets its threshold relative to the peak of the signal you're compressing. Secondly, after 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 K2661'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 K2661 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 K2661 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 K2661 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 K2661, 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 K2661 chooses between, according to the attack velocity of the note. 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 14-38 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 to 127 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 key ranges. The velocity range is set with the Velocity Crossover parameter(s), described on page 14-38.

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 K2661 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 K2661 sets Coarse Tune so that the note plays the sample without transposition. If the sample is assigned to a range of notes, the K2661 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.

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 K2661 (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 K2661 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 K2661 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 K2661 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 K2661 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 K2661 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 K2661. 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 K2661. 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 K2661 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 K2661’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 the same 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 K2661’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 K2661’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 Stlm 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. 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 K2661 on any kind of input signal.

For starters, you can simply hook up a CD player to one of the K2661'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 K2661 makes it possible to alter the pitch of the incoming signal in real time. But the K2661 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 K2661 plays buffered data from the input source, which enables the K2661 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 K2661 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 K2661 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 K2661 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 K2661, 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 K2661’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 15-20.



*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:

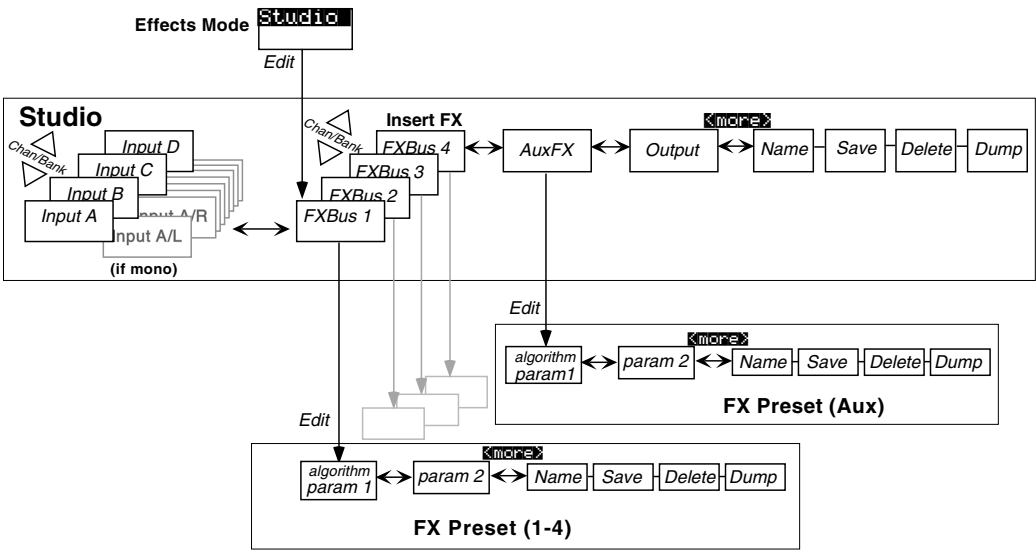


Figure 15-1 The KDFX User Interface

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).

```
KDFX Mode: MAIN FXCtrl: Auto <> Enable
Studio: 113 PltEnvFI4T Plate Free: 0
FX1 43 Plebe Chamber - Size: 1
FX2 902 Synth Env Filter B Size: 2
FX3 735 Bap ba-da-dap - Size: 1
FX4 0 None B Size: 0
Aux 103 RigPredelayPlate B Size: 3
MAIN CTRL EQBYP FXBYP BUSMUT Enable
```

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 K2661’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.

```

EditStudioFXBUS Size:1 Free:0 <>FXBUS:1
FX1→14 string 1Hall→Aux→Mix→
Wet/Dry :32%wet Lvl:0ff Lvl:0.0dB
Out Gain :0.0dB Bal:0% Bal:0%
Allocation:Auto
<more> INPUT FXBUS AUXFX OUTPUT more>

```

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.

```

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
<more> PARAM1 PARAM2 more>

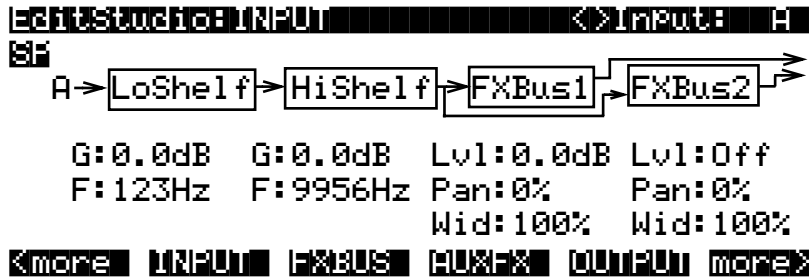
```

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.



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.



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 K2661's physical outputs.

```

EditStudio:OUTPUT
Mix Lvl:0.000 Output A:Mix
Mix Bal:0%    Output B:Off
              Output C:Off
              Output D:Off

<more> INPUT FxBUS AUXFX OUTPUT <more>

```

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 K2661 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

Delete Insert <<< >>> OK Cancel

```

The Compare and FX Bypass Buttons

As with all K2661 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 15-20.

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

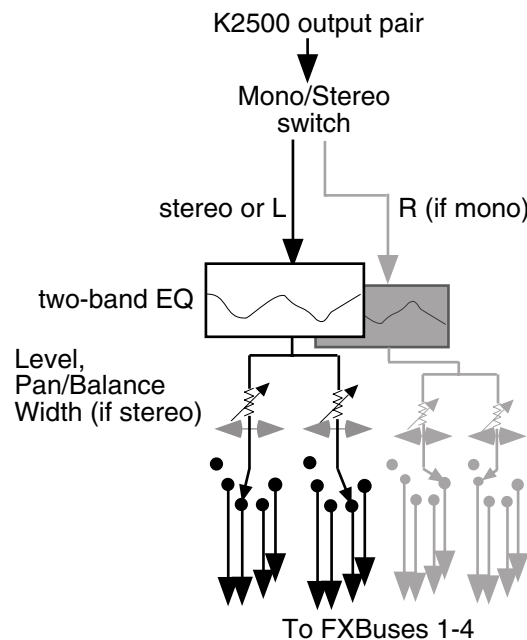


Figure 15-2 The KDFX Input Editor

Inputs are referred to as A, B, C, and D, and correspond to the four output buses (KDFX-A through KDFX-D) from the K2661'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.

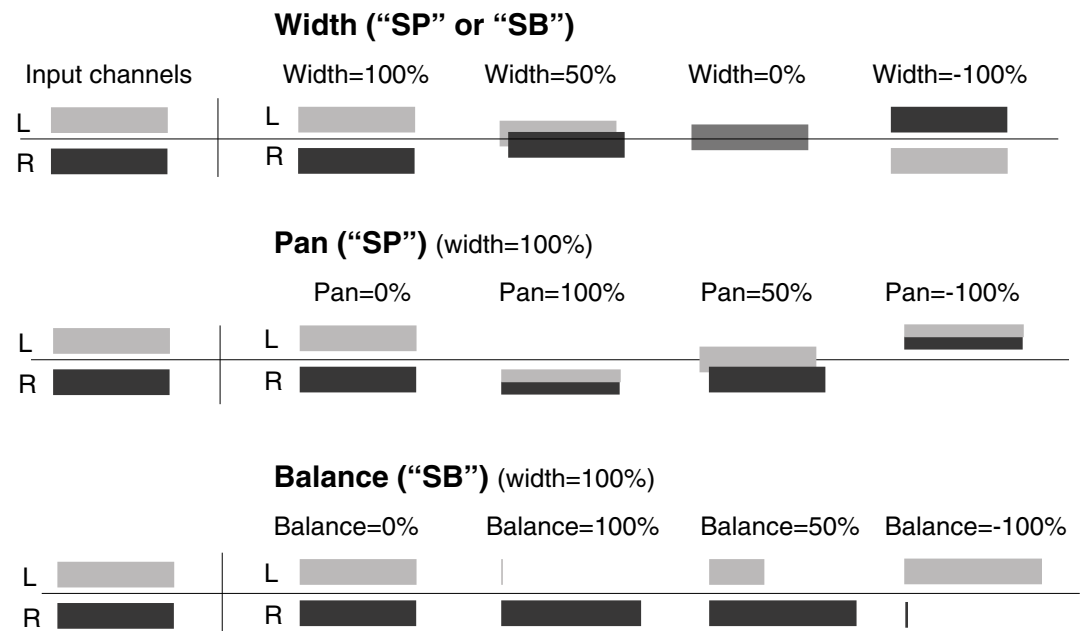


Figure 15-3 Width, Pan, and Balance controls

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.

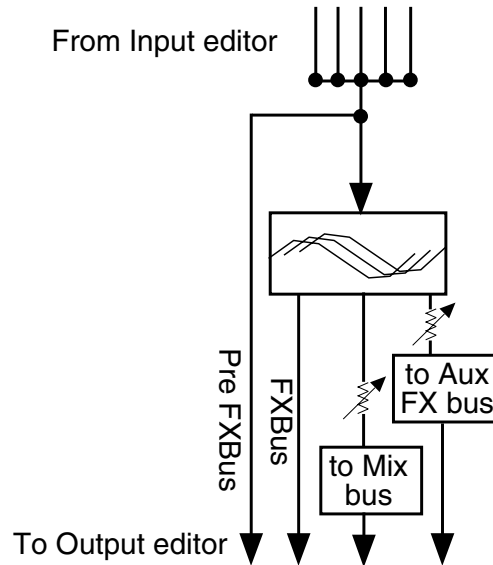


Figure 15-4 FXBus Signal Routing

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.

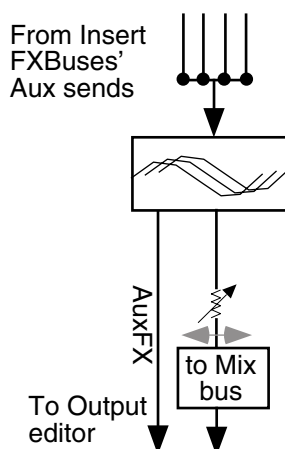


Figure 15-5 Aux Bus Outputs

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 K2661 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 K2661's ROM (like ROM samples), and are not changeable by the user. As you change algorithms, the parameters displayed on the page change accordingly.

```

Edit:FXPreset:PARAM1      EffectSize:3/3
FXAlgorithm:1 Panaural Room
Wet/Dry   :30%           In Gain   :0.0dB
Room Size :15.2m         Out Gain  :2.0dB
Pre Dly   :4ms           Decay Time:1.7s
HF Dampng: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 K2661 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.

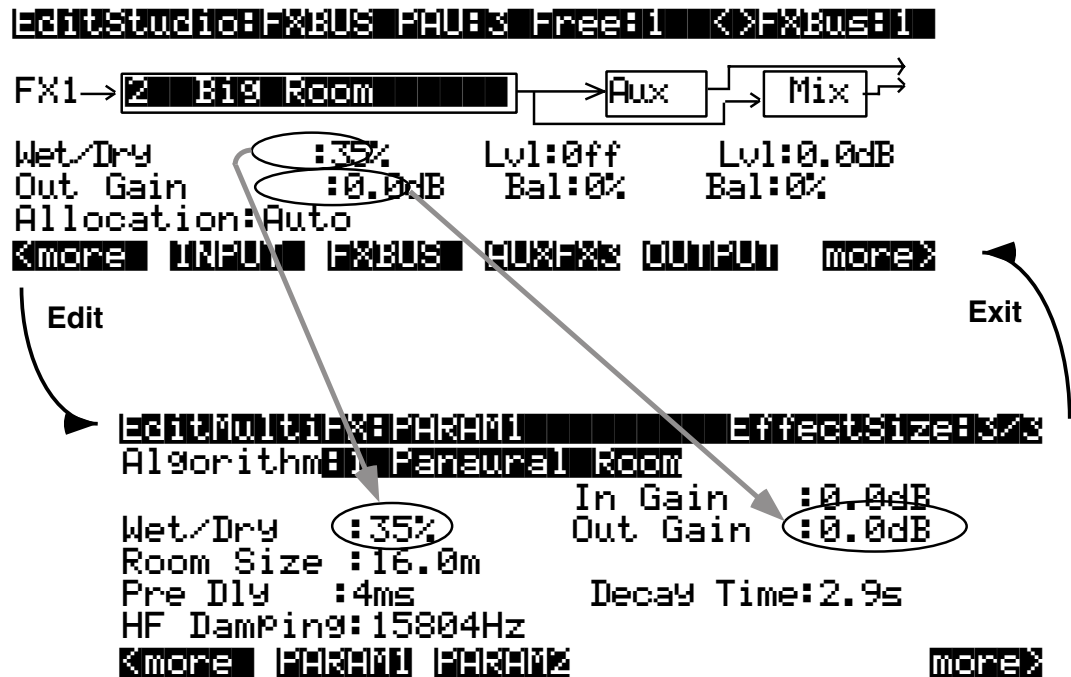


Figure 15-6 Bus Overrides in Place, But Not Engaged

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.

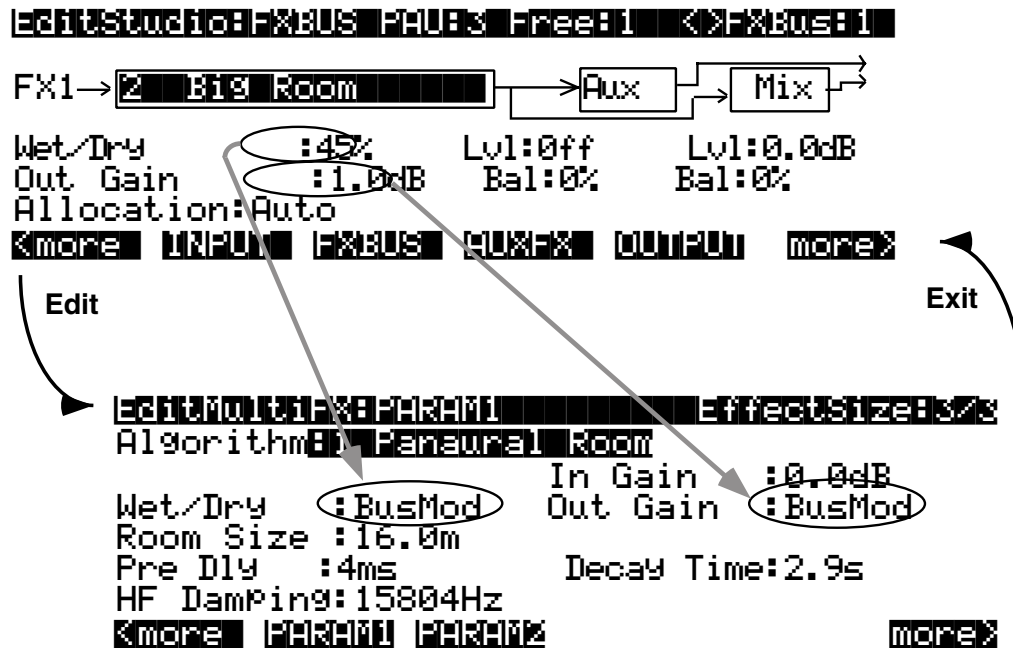


Figure 15-7 Bus Overrides Engaged

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:2A
```

or inside an FX preset:

```
EditFXPreset:HHHMM EffectSize:2/1A
```

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.

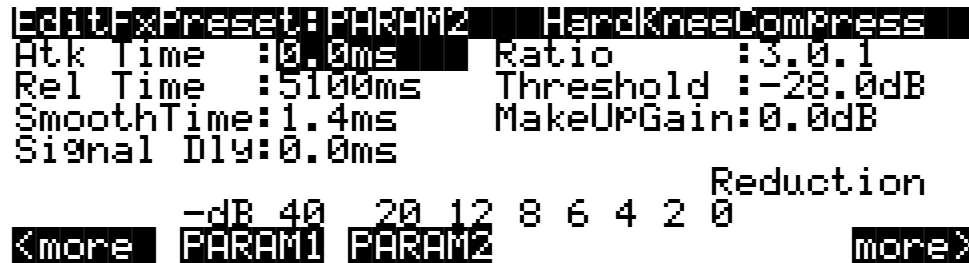
The Aux bus is generally used as a global processor, but it doesn't have to be. You can set up one FXBus as a "dummy"—set its FX preset parameter to 0 **None**—and send its output to the Aux bus, and at the same time turn off the Aux sends from all the *other* buses. Now the Aux bus can function as an 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 K2661 operates with about 20 dB of headroom, this translates to about -64 dB relative to normal operating level.

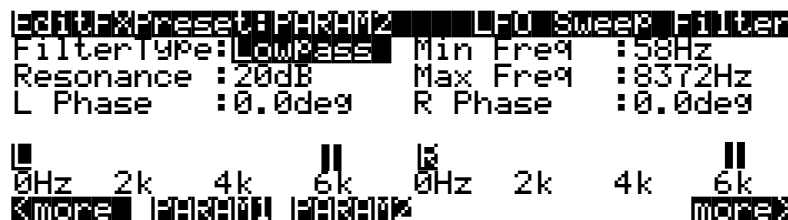


In addition, a number of KDFX's algorithms include a fast visual metering system for monitoring what the algorithms are doing. Many of these algorithms deal with dynamics control like compression and limiting. An example is Algorithm **950 HardKneeCompress**. Meters are most often found on the PARAM2 page of the FXPreset Editor:



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:



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

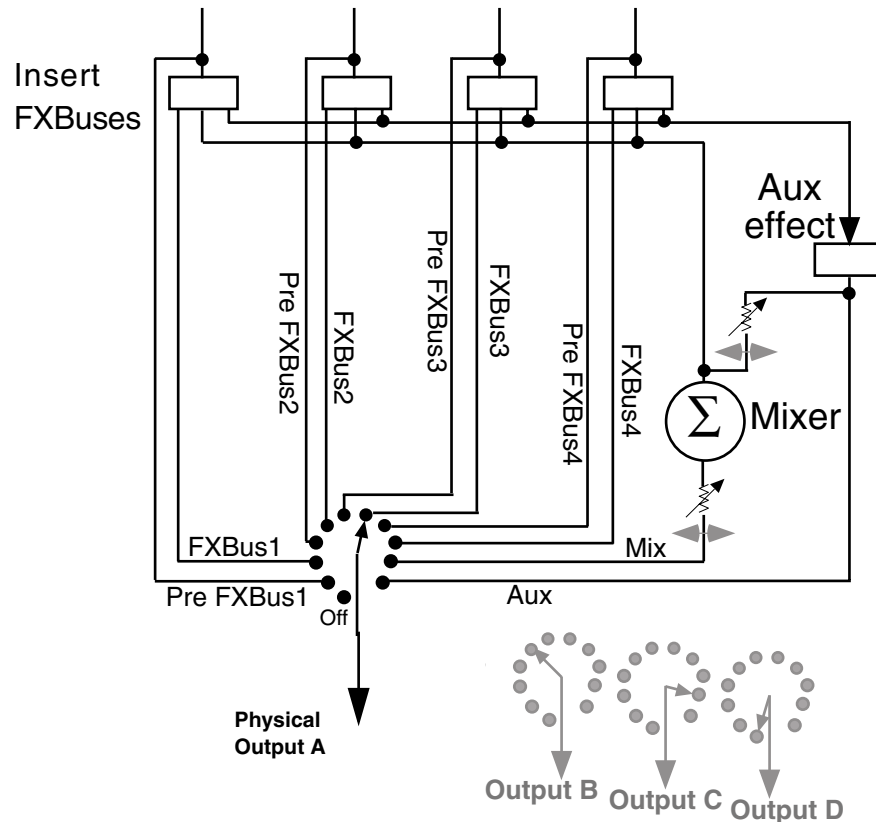


Figure 15-8 FXBuses and Physical Outputs

In the Output section, the various signal paths are routed to the K2661'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 Digital Outputs

The K2661's separate analog outputs (two pairs: A and B) are wired in parallel and are "live" at all times. So are the eight digital outputs (four stereo pairs) available in ADAT format at the ADAT/AES optical jack. Also available at the ADAT/AES Out jack is a single stereo pair (output A) in AES/EBU or S/PDIF format. The parameters on the OUTPUT page affect both the separate analog and digital outputs.

Analog Mix Output

The K2661's analog Mix output combines output pairs A and B into a single stereo analog pair. Each of the analog outputs, A and B, 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 and B; connecting cables to Outputs A and B does not remove the corresponding portion of the signal from the Mix outputs.

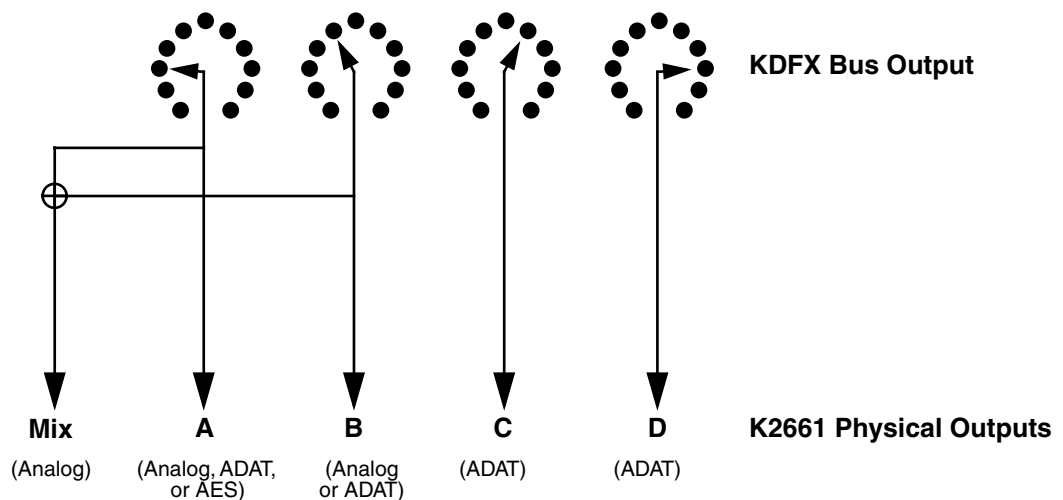


Figure 15-9 K2661 Analog and Digital Outputs

Digital Output

Digital output is always available at the ADAT/AES Out optical jack, in ADAT 8-channel format or in AES 2-channel format. Both AES/EBU Pro and Consumer (S/PDIF) formats are supported. You may also choose a digital word length to match the resolution of your other digital equipment: 16, 20, or 24 bits. The choice of digital output format and digital word length is made on the second Master Page menu. See *Digital Output Format* on page 11-11 for details.

Saving Studios and Other objects

Saving

Saving a studio or an FX preset is handled the same as any other K2661 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:

<more **Name** **Save** **Delete** **Dump** **more>**

Name lets you rename the object, using the standard K2661 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 K2661 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 K2661 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 K2661'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 K2661'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 K2661 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:081  <>Channel:1
FX Mode:Program
FX Chan:1
```

```
          DigOut :16A 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 K2661'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>A
```

These are the FXMod setup pages, and in fact there are eight of them—press **more>** again to see the others.

```
<more  FXLFU  FXHSR  FXFUN  IMPFX  more>A
```

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 9-26. 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 15-26.

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 K2661 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 K2661 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:

<more **FXLFO** **FXASR** **FXFUN** **IMPFX** **more>**

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 $1.0 + (4.0 \times 1/2) = 3.0$ dB, and when it is all the way down (MIDI value 0), the gain is $1.0 + (4.0 \times 0) = 1.0$ 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 ($1.0 + (-4.0 \times 1/2)$); 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.

```

Edit:FXPreset:PHXHM1 EffectSize:3/8
Algorithm:1 Pansaural Room
Wet/Dry :FXMod In Gain :0.0dB
Room Size :FXMod Out Gain :2.0dB
Pre Dly :4ms Decay Time:1.7s
HF Damp:14080Hz
<more> PHXHM1 PHXHM2 <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 15-23) 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

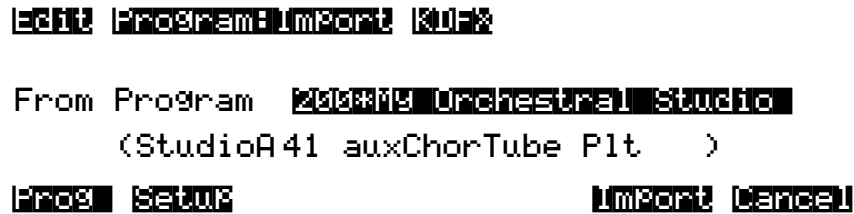
Static FXMods, that is, FXMods that are just going to be used to “set and forget” studio parameters when a program or studio is called up (as opposed to dynamically controlling them), are easy to set up. On the FXMOD page, select the bus and parameter you want to control, set the Adjust value to the 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:



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 K2661 Keyboard

The sliders and wheels on the K2661 keyboard, and the pedals connected to the keyboard, can be extremely useful with KDFX, if you assign FXMods to the K2661'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 K2661 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 K2661 keyboard in Setup mode.

KDFX in Program Mode

If you are playing the K2661 in Program mode, then you must make sure that the current channel of the K2661 (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.

K2661 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

Table 15-1 Modes, Control Assignments, and Effects Control

K2661 Mode	FX Mode Value	What Sets Physical Control Assignments (Sliders, Wheels, etc.)	What Controls KDFX	MIDI Channel for Control of KDFX
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 15-1 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 K2661'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 K2661 on-board controllers, then you can record your parameter changes from the K2661 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 K2661 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.

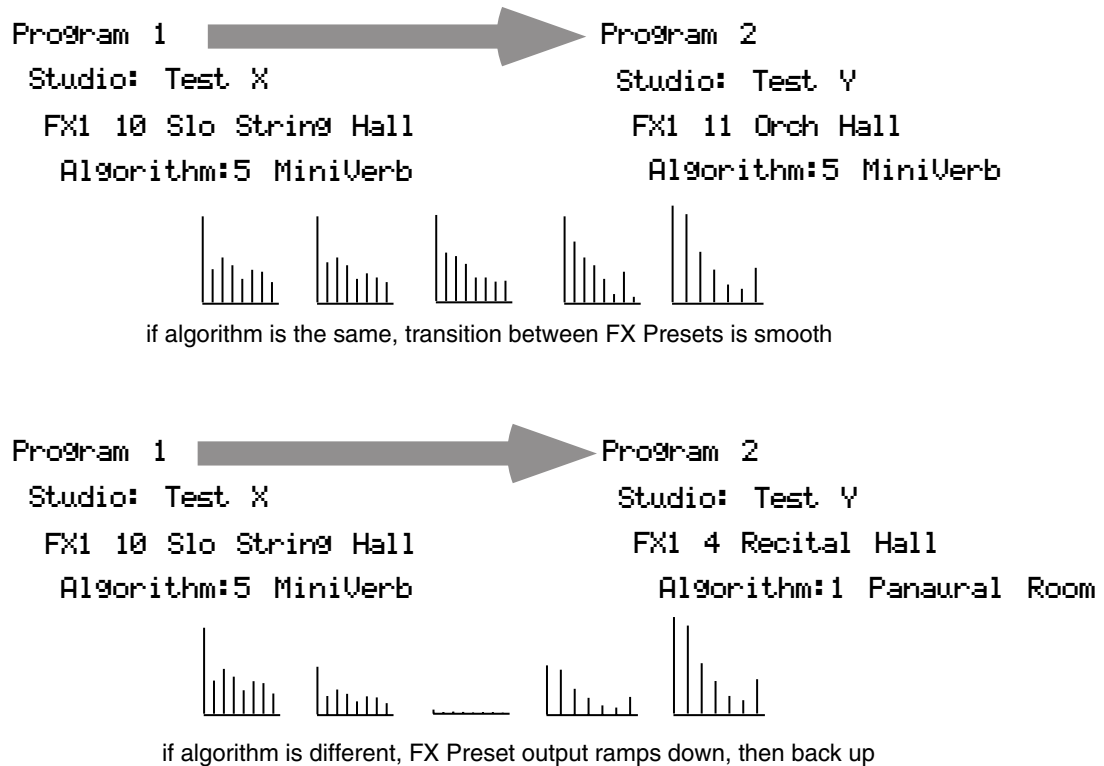


Figure 15-10 Effect of FX Algorithm on Transition Between FX Presets

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.

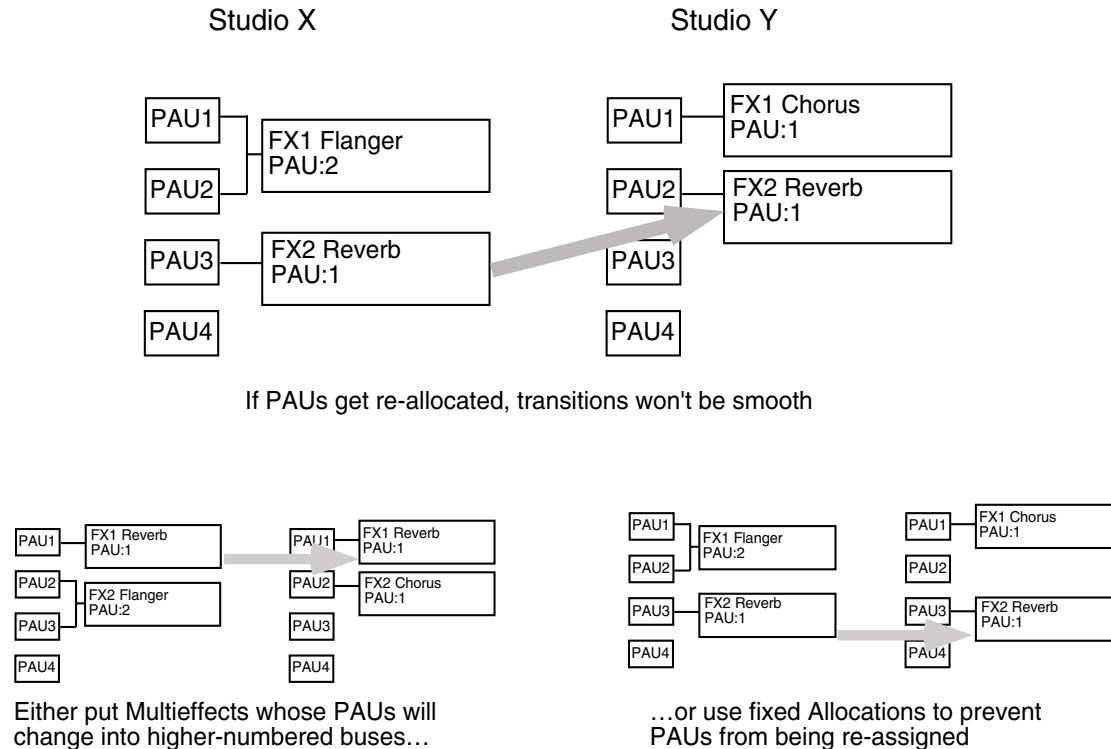


Figure 15-11 Avoiding Transition Problems When PAUs Get Reallocated

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 K2661's internal sequencer, or it can come from an external MIDI sequencer that has been configured to send MIDI Timing Clocks to the K2661.

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 K2661 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 K2661, 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 $(3.75 \text{ BPM} / 60) \times 2$, or 0.125 Hz. **0.13 Hz** is the nearest value available.

Source **Tempo**

Depth The fastest vibrato we want is $(240 \text{ BPM} / 60) \times 2$, 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 K2661'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 K2661'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 K2661: 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.

Expans 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 Tap n 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 Fdbkn 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 24.0 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 is 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 $20.8\mu\text{sec}$) from **-127 to 127**.

Xcrs 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 ($20.8\mu\text{s}$), 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 15-25 for 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 **HiResXcurs** 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 K2661 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 K2661 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 K2661'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 K2661'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.

Note that Triple Mode offers even more algorithmic possibilities than those described here; see Chapter 12 of the *Musician's Reference* for details.

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 K2661'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	Added Waveforms
Equalization (EQ)	Nonlinear Functions
Pitch / Amplitude / Pan Position	Waveforms with Nonlinear Inputs
Mixers	Mixers with Nonlinear Inputs
Waveforms	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 K2661'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.

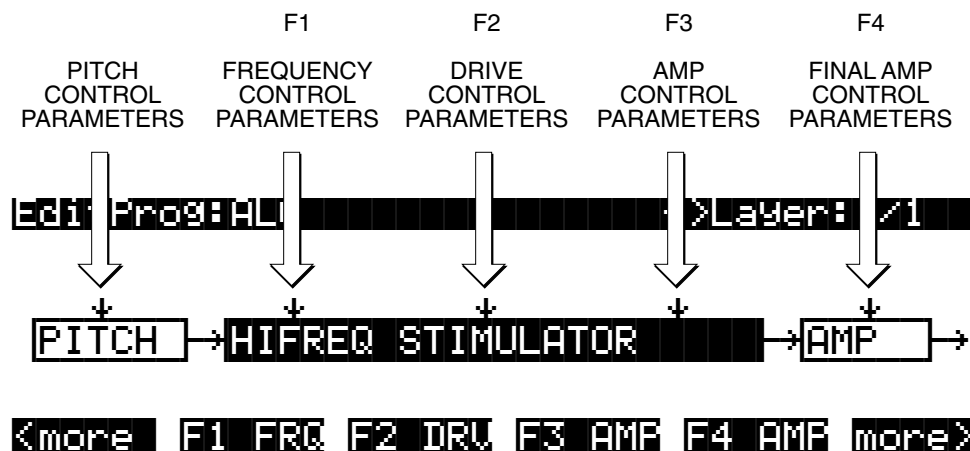


Figure 16-1 Input Control for DSP Functions

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, as well as Triple Mode, which amalgamates the DSP capabilities of three layers). 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(HI-FRQ S LMD<>LAYER:1/1
Coarse:C 4 262HZ Src1 :OFF
Fine :Oct Depth :Oct
KeyTrk:Oct/key Src2 :OFF
VelTrk:Oct DptCt1:OFF
Pad :Oct MinDpt:Oct
MaxDpt:Oct
<more F1 FRQ F2 DRU 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 in 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.



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.

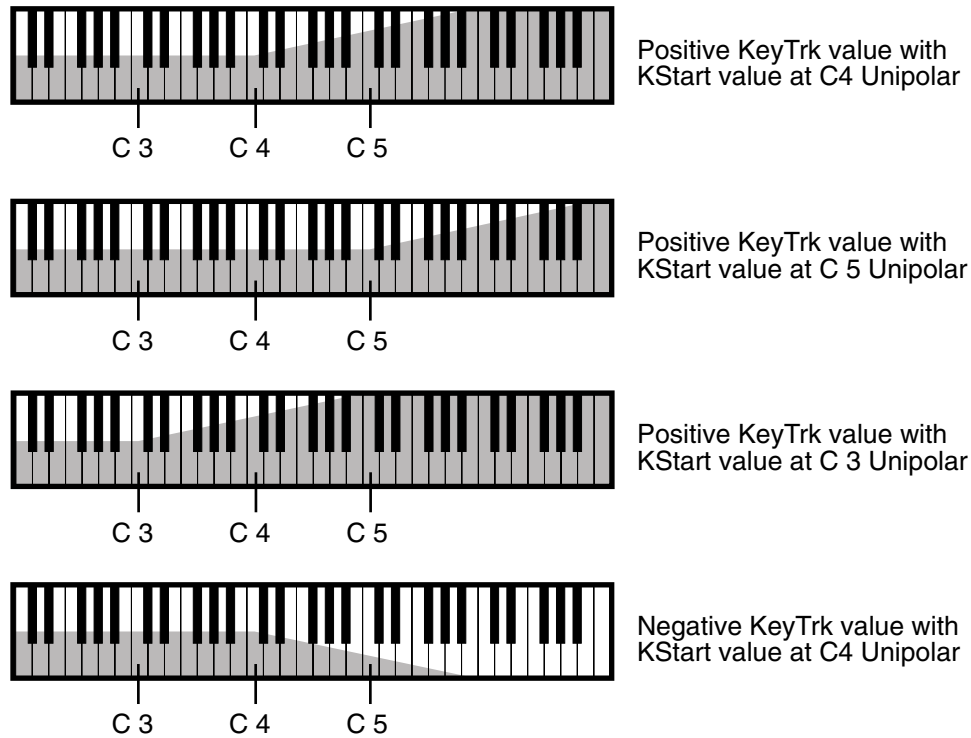


Figure 16-2 Unipolar Keystart

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.

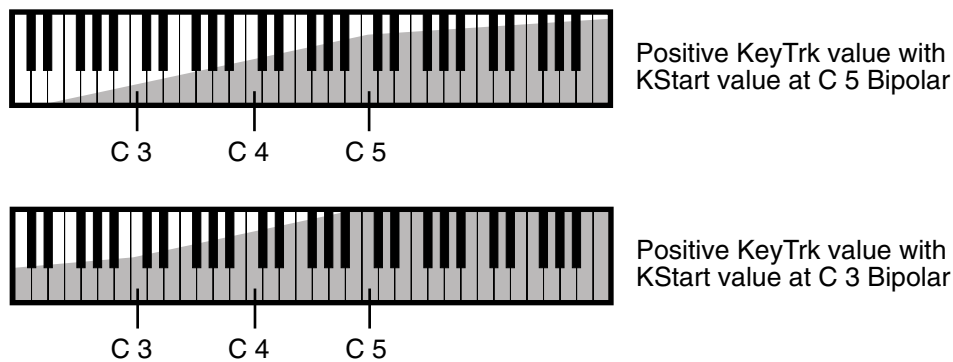


Figure 16-3 Bipolar Keystart

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	One-pole Allpass
Two-pole Lowpass	Two-pole Allpass
Two-pole Lowpass, -6 dB resonance	Two-pole Notch
Two-pole Lowpass, +12 dB resonance	Two-pole Notch, fixed width
Four-pole Lowpass with separation	Double Notch with separation
Gated Lowpass	Two-pole Bandpass
One-pole Highpass	Two-pole Bandpass, fixed width
Two-pole Highpass	Twin Peaks Bandpass
Four-pole Highpass with separation	

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.

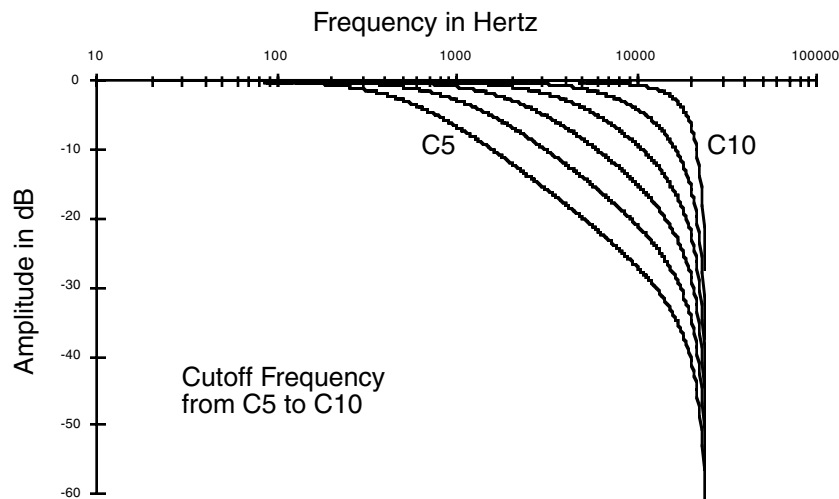
Filter Terminology

- 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 K2661 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 K2661 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 K2661 (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 K2661 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 K2661.

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.

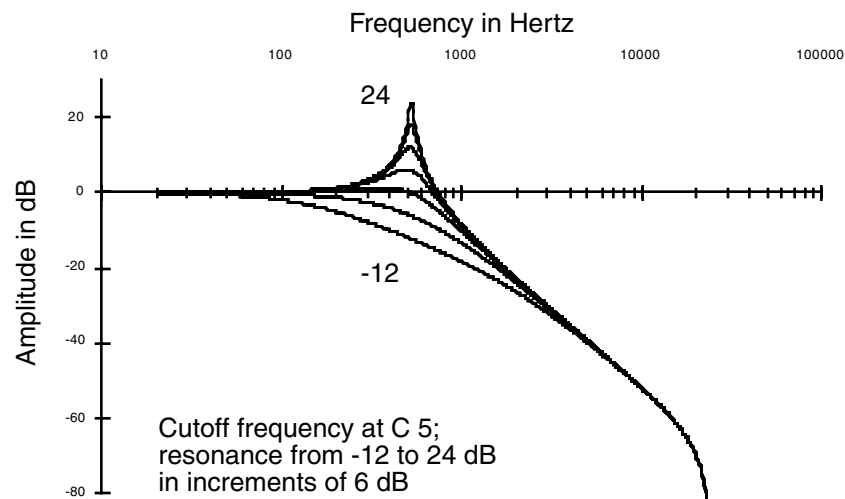
```

EditProg:F1 FRQ(LOPASS) <>Layer:1/1
Coarse:C 4 262 Hz Src1 :OFF
Fine :Oct Depth :Oct
KeyTrk:Oct/key Src2 :OFF
VelTrk:Oct DptCt1:OFF
Pad :0dB MinDpt:Oct
MaxDpt:Oct
<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

DSP Functions

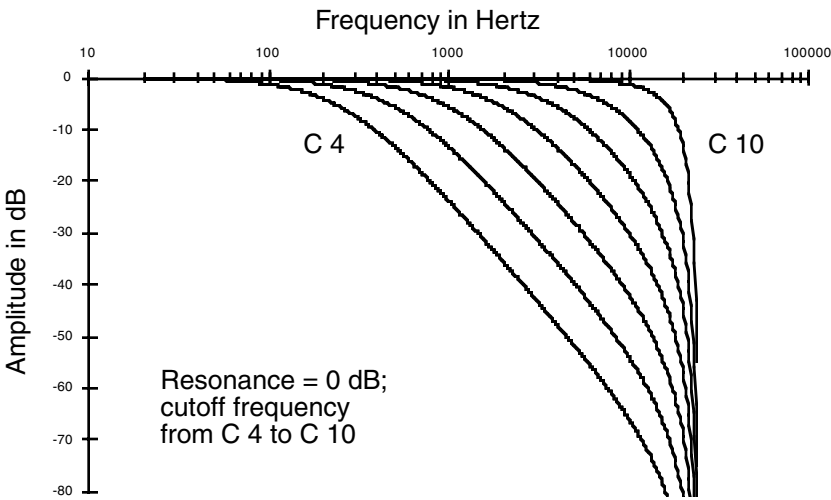
The DSP Functions

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
HdJust: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>
```

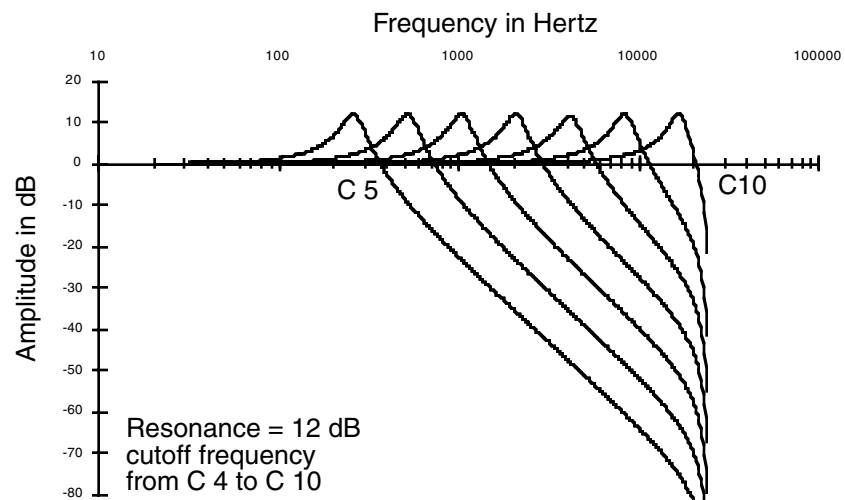
Parameter	Range of Values
Adjust	-12 to +48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	-30 to +60 dB
Source 1	Control Source list
Source 1 Depth	-30 to +60 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	-30 to +60 dB
Maximum Depth, Source 2	-30 to +60 dB

Two-pole Lowpass Filter, -6 dB Resonance (LOPAS2)



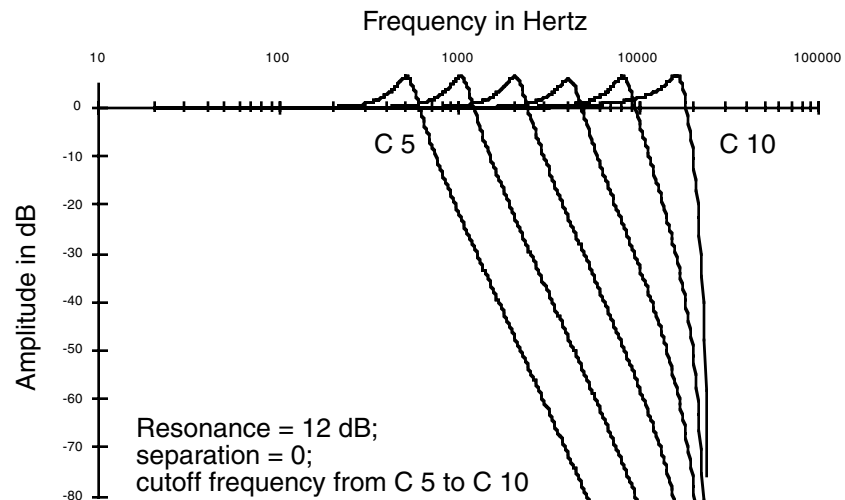
Using this filter is the same as using two one-pole lowpass filters in successive algorithm blocks. Since its resonance is fixed at -6 dB, it's also the same as using 2POLE LOWPASS with the resonance set to -6 dB. You'd use this filter when you want a 12 dB per octave rolloff but don't need to set a 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)



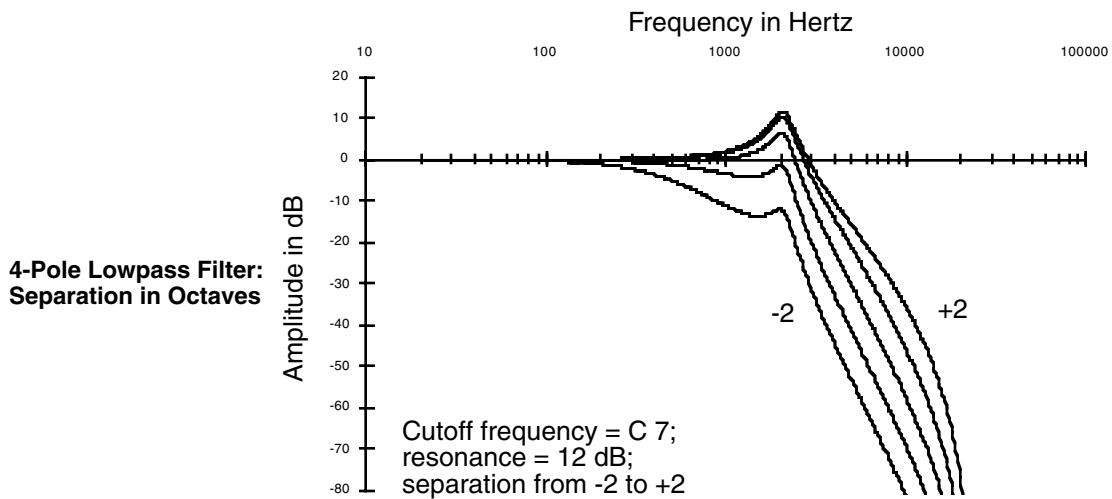
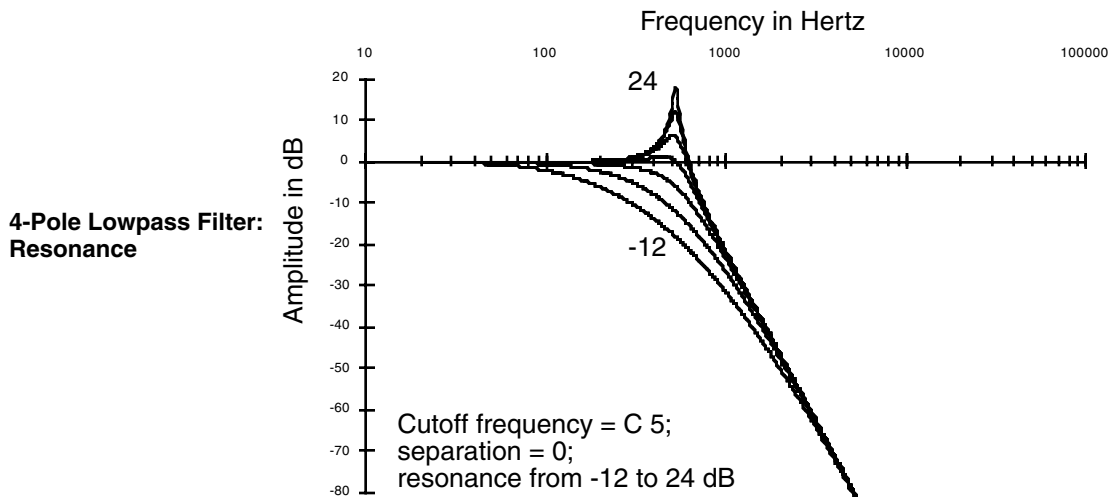
This is similar to LOPAS2; the only difference is that its resonance is fixed at +12 dB.

Four-pole Lowpass Filter with Separation (4POLE LOPASS W/ SEP)



DSP Functions

The DSP Functions



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 LUPHSS) <>Layer:1/1
HdJust:0ct Src1 :OFF
Fine :0ct Depth :0ct
KeyTrk:0ct/key DptCt1:OFF
VelTrk:0ct MinDpt:0ct
MaxDpt:0ct
<more F1 FRQ F2 RES F3 SEP F4 AMP more>

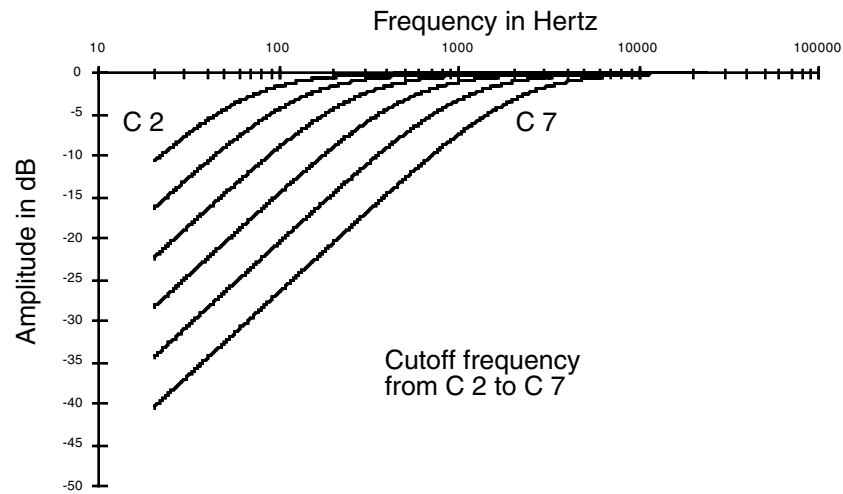
```

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.

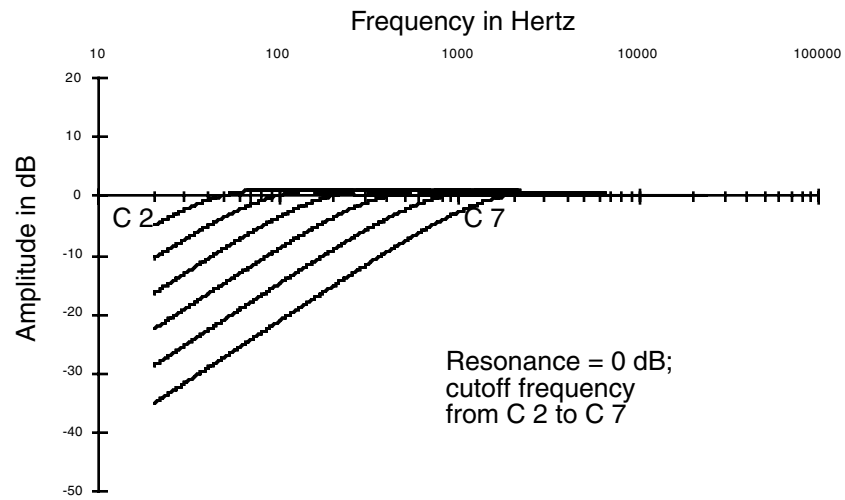
One-pole Highpass Filter (HIPASS)



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.

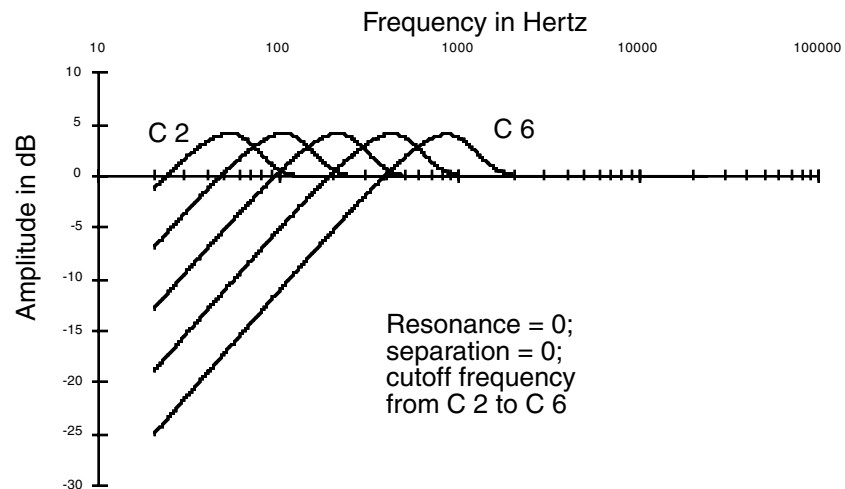
The Coarse Adjust parameter sets the cutoff frequency in terms of a key name. The remaining parameters (except Pad) alter the cutoff frequency in increments of cents. Positive key tracking values raise the cutoff frequency for high notes and lower it for low notes. More specifically, a value of 100 cents 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)



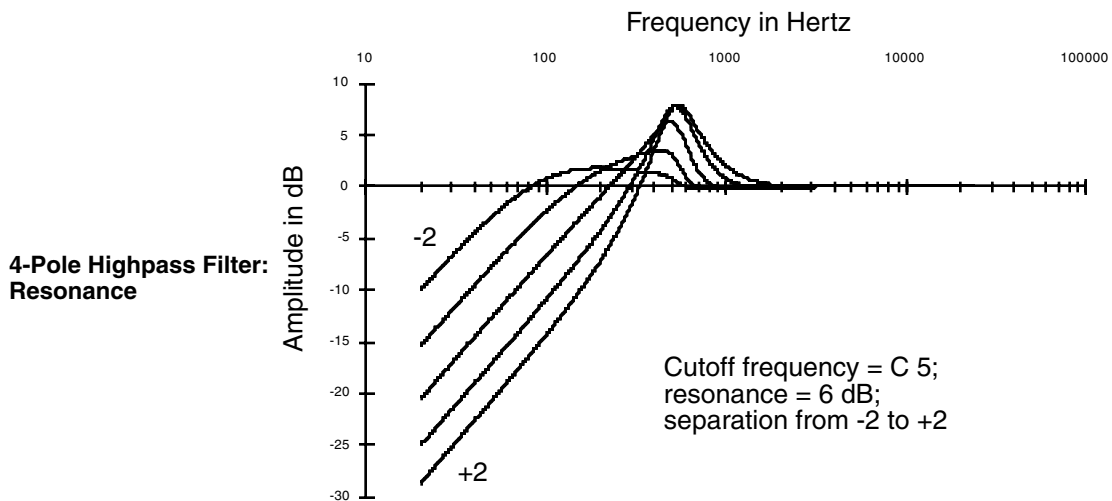
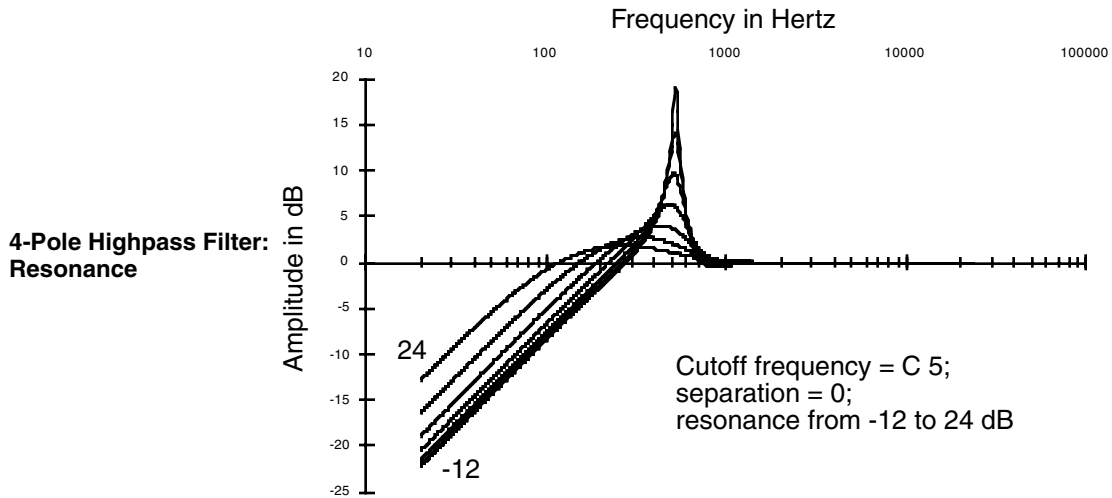
This is very similar to HIPASS. The primary difference is in the steepness of the rolloff at the cutoff frequency. Below the cutoff frequency, the rolloff is similar to that of HIPASS, except that there's a one-octave shift—that is, HIPASS with a cutoff frequency of C 3 will sound nearly the same as 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)



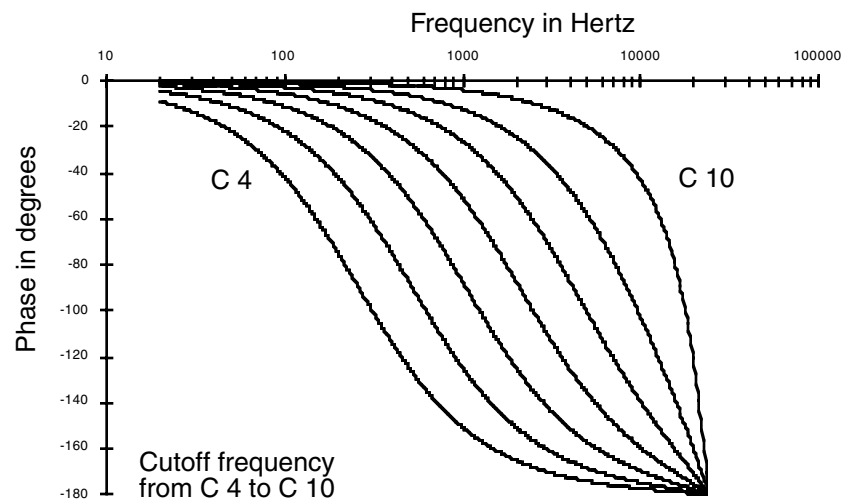
DSP Functions

The DSP Functions



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)

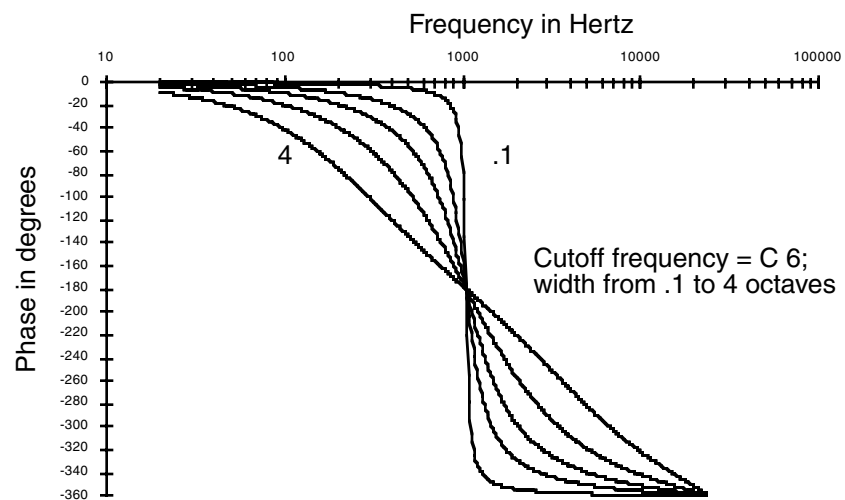
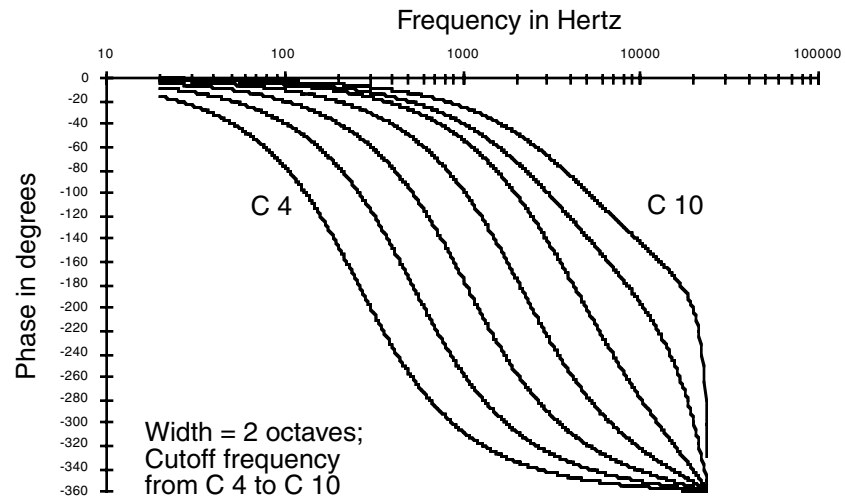


Allpass filters do not affect a sound's frequency response (the amplitude of partials at various frequencies), but change the phase of each partial depending on its proximity to the center frequency. The phase shift is -90 degrees for partials at the center frequency. It rises toward 0 degrees for partials 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.

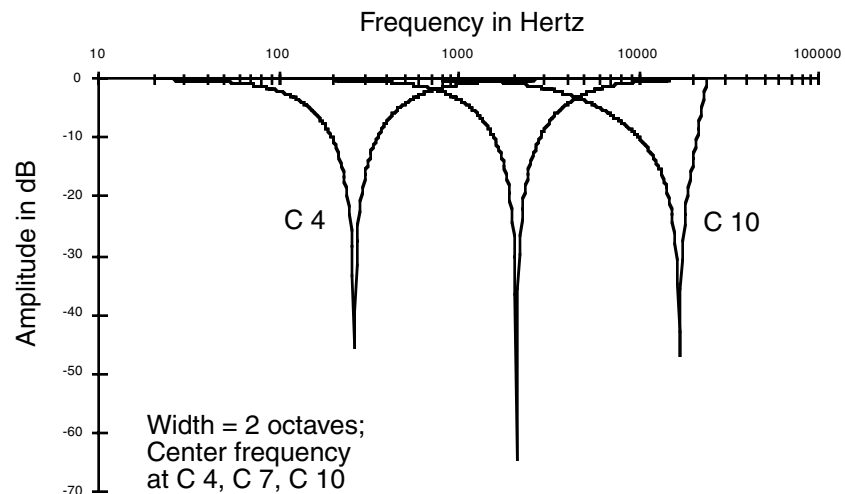
If you leave the center frequency constant and assign an LFO to vary the width, partials with frequencies above the center will shift their pitches in the opposite direction of partials below the center frequency.

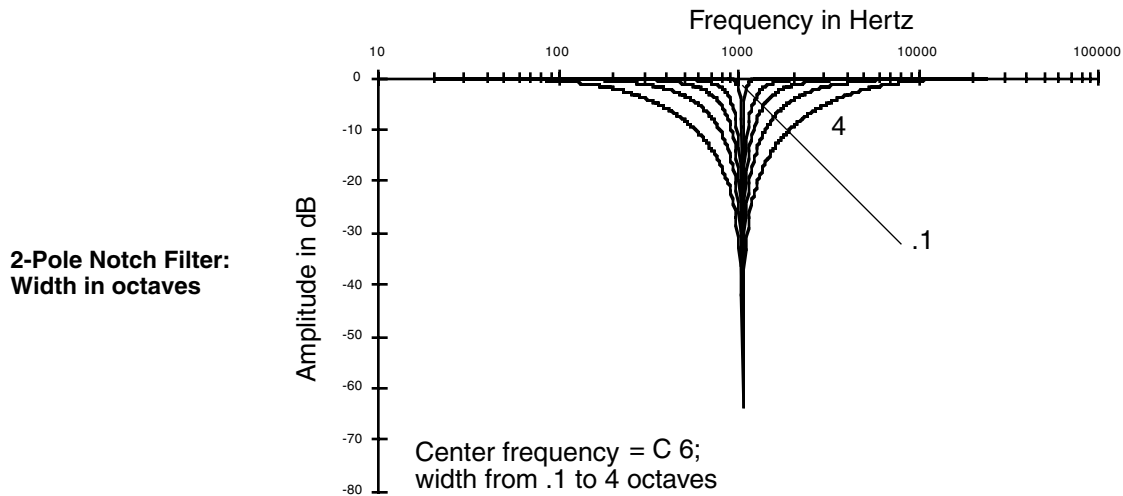
```

EditProg:F2 WII(2P HL PASS) <>Layer:1/1
Adjust:0.010oct Src1 :OFF
Depth :0.00oct
Src2 :OFF
KeyTrk:0.000oct/key DptCt1:OFF
VelTrk:0.00oct MinDpt:0.00oct
MaxDpt:0.00oct
<more F1 FRQ F2 WII 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

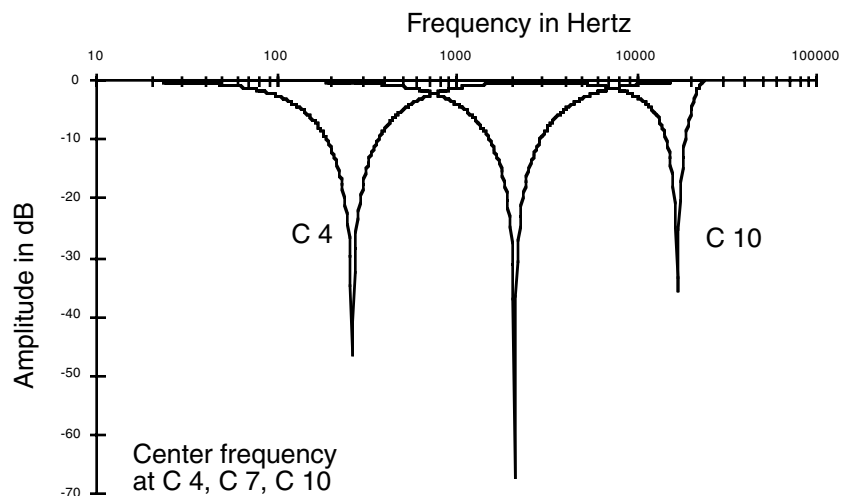
Two-pole Notch Filter (NOTCH FILTER)





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)



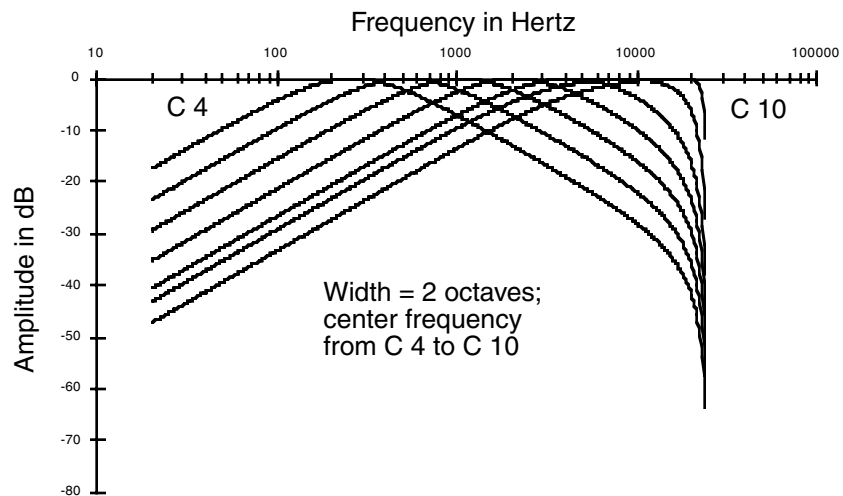
The only functional difference between NOTCH2 and NOTCH FILTER is that the width of NOTCH2 is fixed at 2.2 octaves. This gives you a one-stage notch filter function.

Two-pole Bandpass Filter (BANDPASS FILTER)

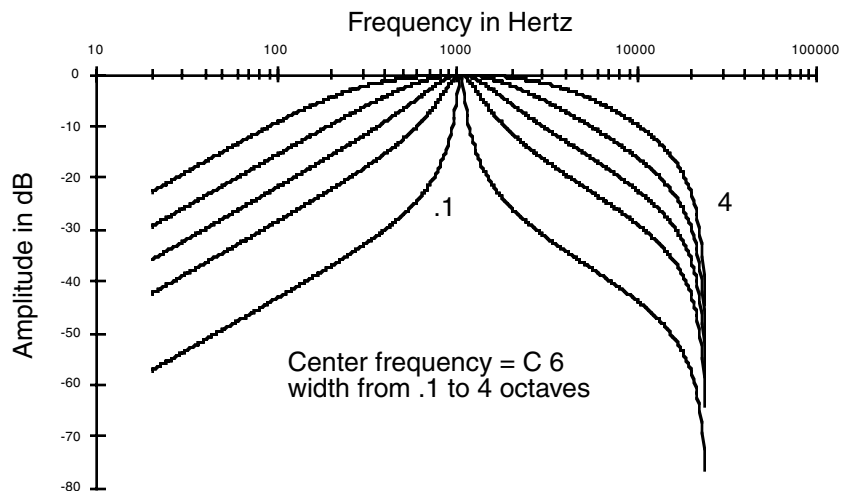
This is essentially the opposite of a notch filter; it passes all partials at the center frequency, and cut the levels of partials above or below the center frequency. The width is defined the same as for the double notch filter.

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.

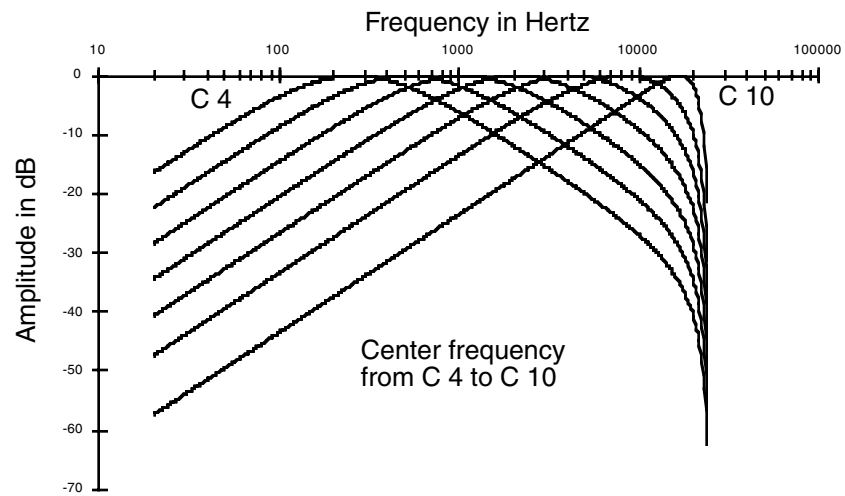
**Bandpass Filter:
Frequency**



**Bandpass Filter:
Width in octaves**

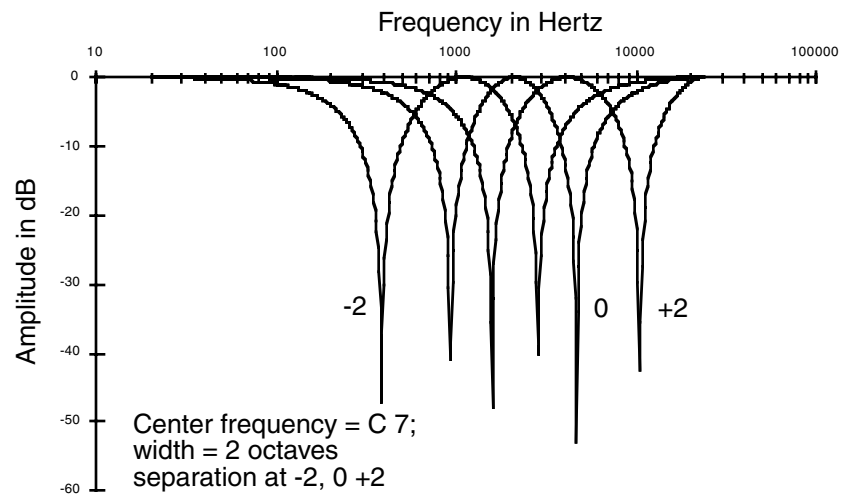


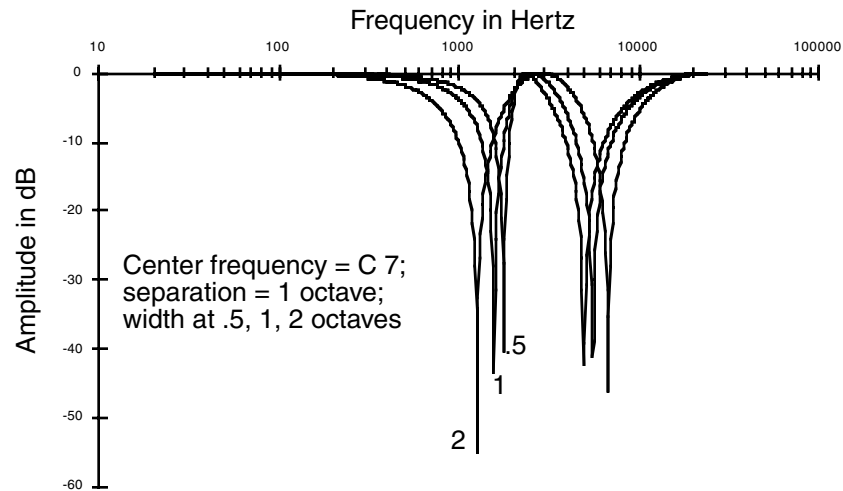
Two-pole Bandpass Filter, Fixed Width (BAND2)



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.

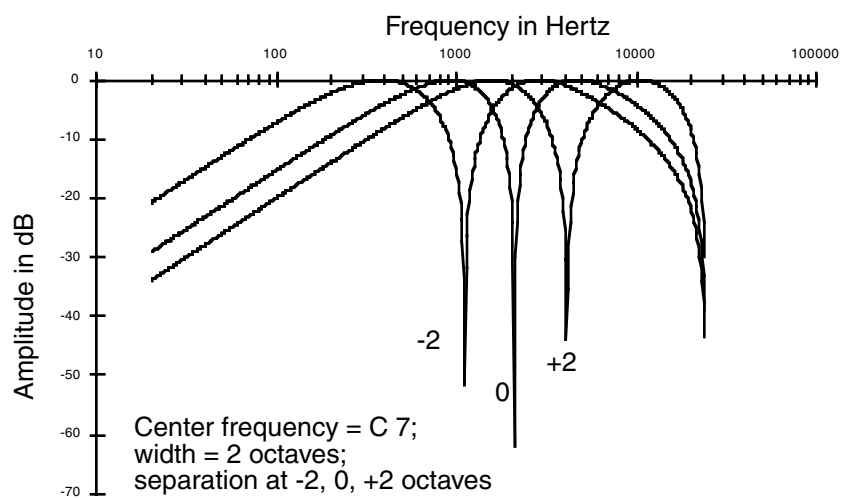
Twin Peaks Bandpass Filter (TWIN PEAKS BANDPASS)

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.

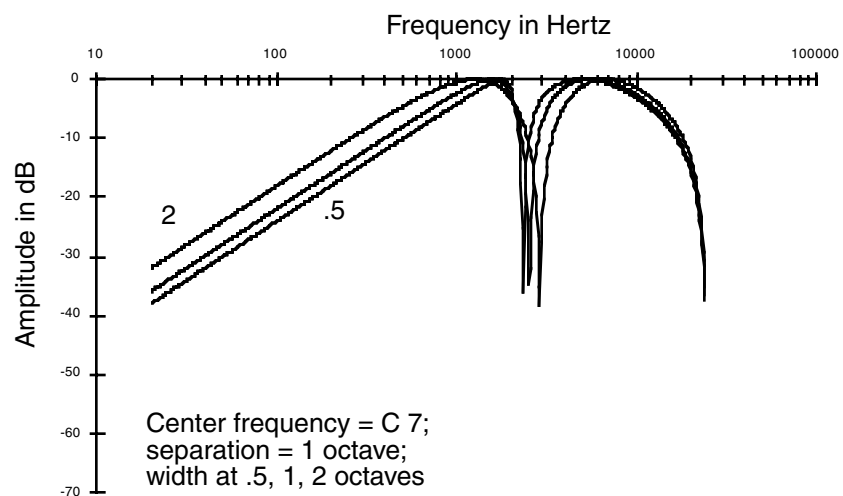
DSP Functions

The DSP Functions

**Twin Peaks
Bandpass Filter:
Separation**



**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

Treble Tone Control

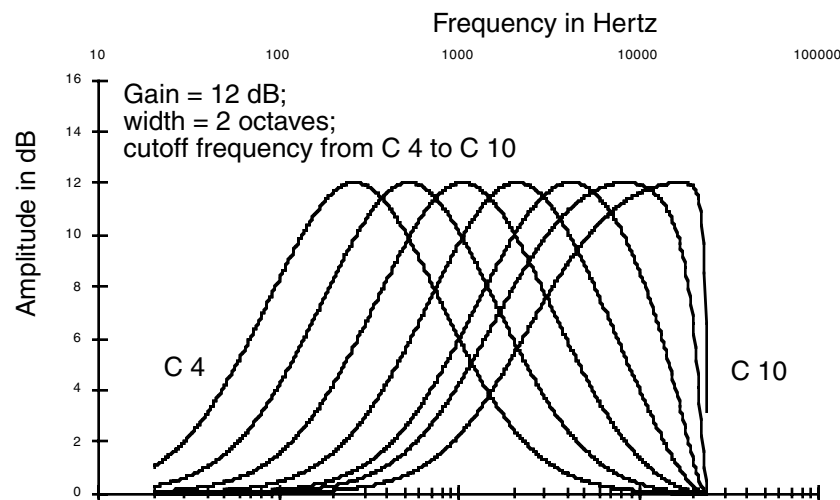
Mid-range Parametric EQ

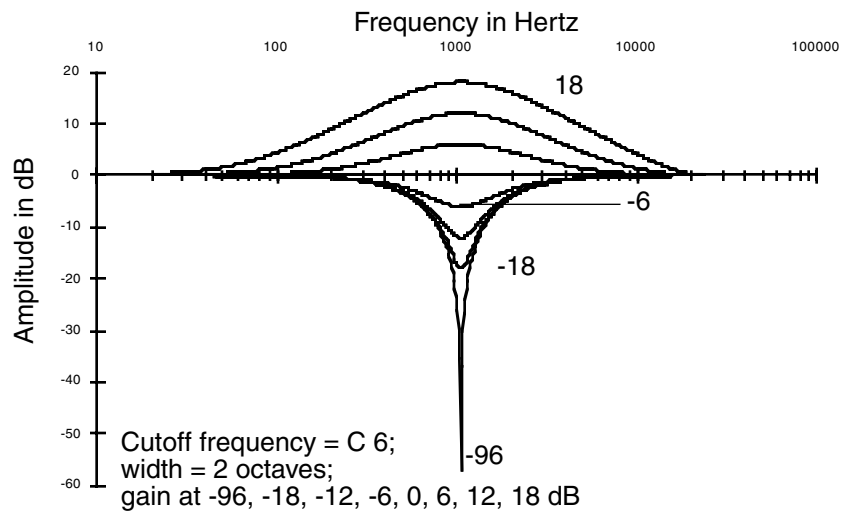
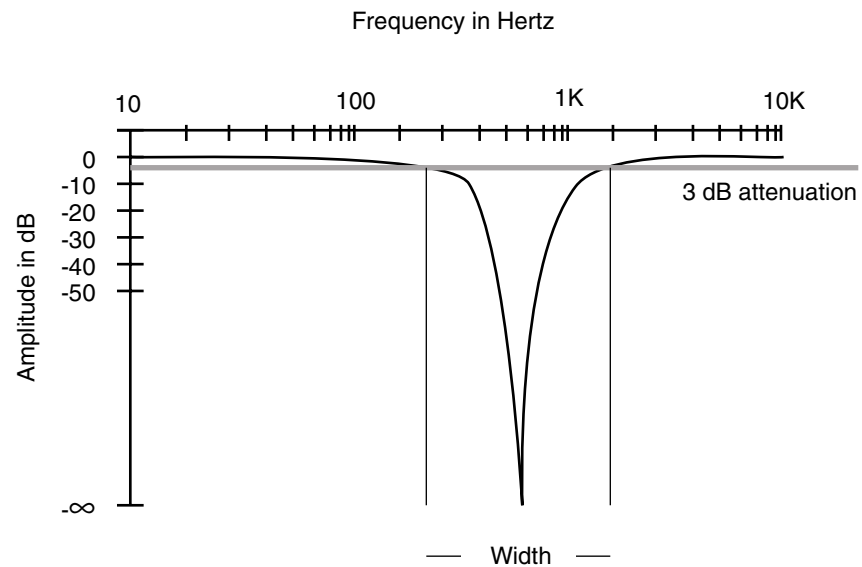
Steep Bass Tone Control

Bass Tone Control

Parametric Equalizer (PARAMETRIC EQ)

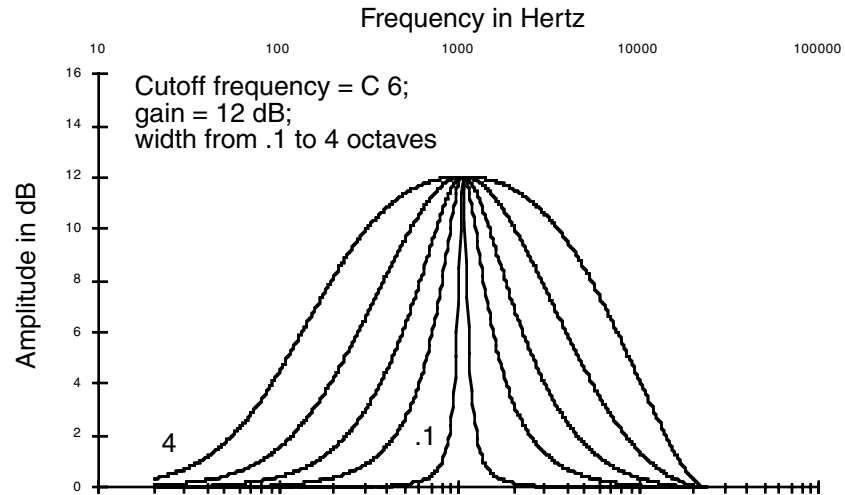
This function has three interacting parameters, each with its own control-input page: center frequency, width, and amplitude. The center frequency is the center of the range of frequencies that will be boosted or cut by the amplitude setting. The width is the entire range of frequencies that will be affected by the amplitude setting. For the K2661, 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(PHRH EQ) <>Layer:1/1
Adjust:C 4 262Hz Src1 :OFF
Fine :Oct Depth :Oct
KeyTrk:Oct/key Src2 :OFF
VelTrk:Oct DptCt1:OFF
Pad :0dB MinDpt:Oct
MaxDpt:Oct
<more F1 FRQ F2 WII 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

DSP Functions

The DSP Functions

The Fine Adjust parameter gives you one-cent precision in setting the center frequency.

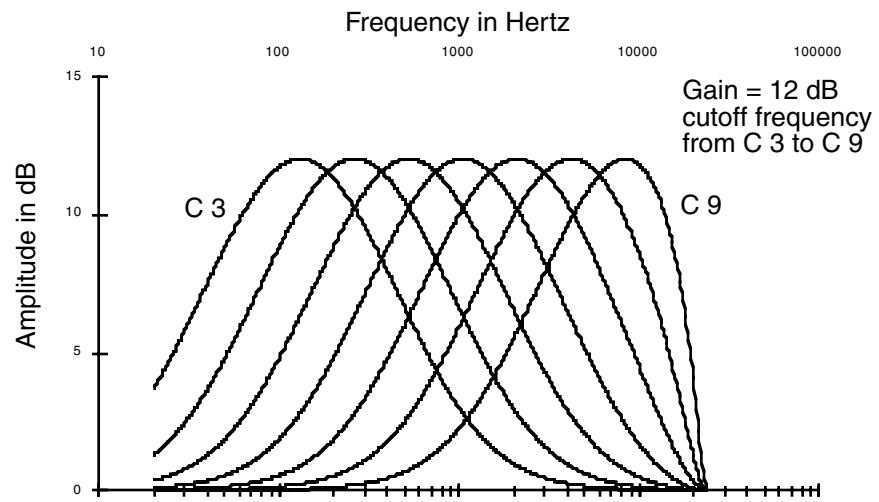
```
EditProg:F2 WII(PARA EQ) <>Layer:1/1
Adjust:0.010oct Src1 :OFF
Depth :0.00oct
Src2 :OFF
KeyTrk:0.000oct/key DptCtl:OFF
VelTrk:0.00oct MinDpt:0.00oct
MaxDpt:0.00oct
<more F1 FRC F2 WII F3 AMP F4 AMP more>
```

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

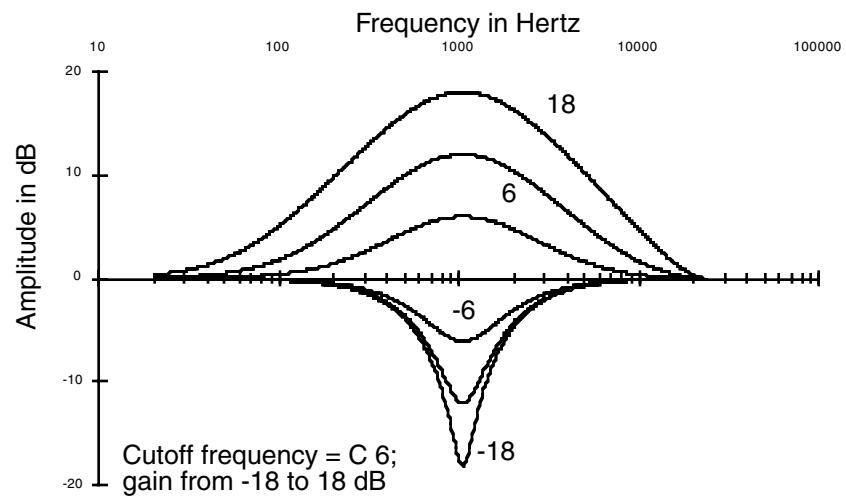
The control-input page for the amplitude stage (F3 AMP) is identical to the AMP page described previously, except that there's no Pad parameter.

Mid-range Parametric EQ (PARA MID)

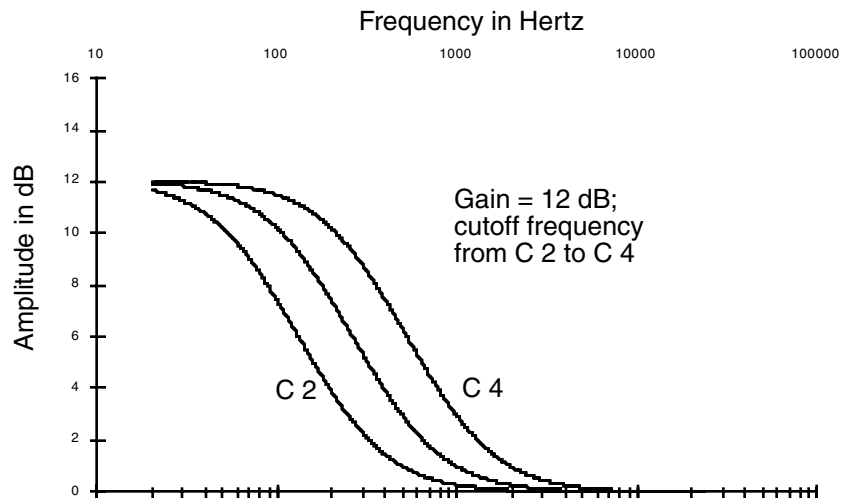
This two-stage function is almost identical to the three-stage Parametric EQ function. The only difference is that the width of PARA MID is fixed at 2.2 octaves. Consequently there's no control-input page for the width.



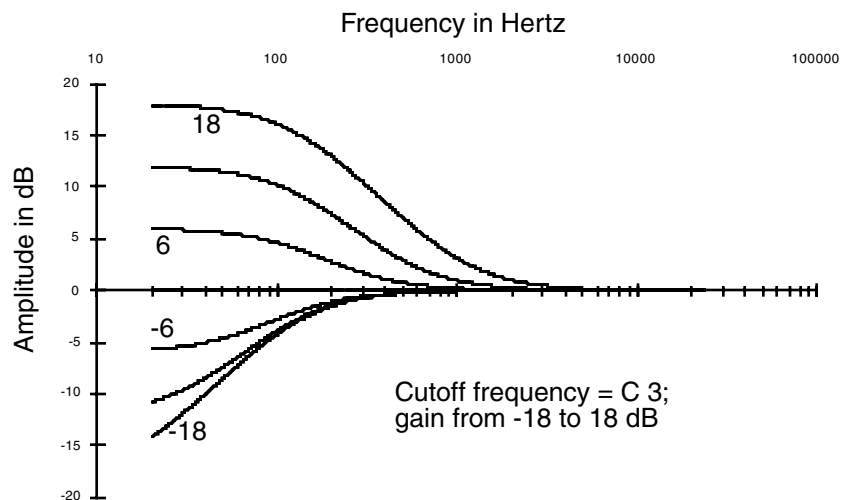
Para Mid: Gain



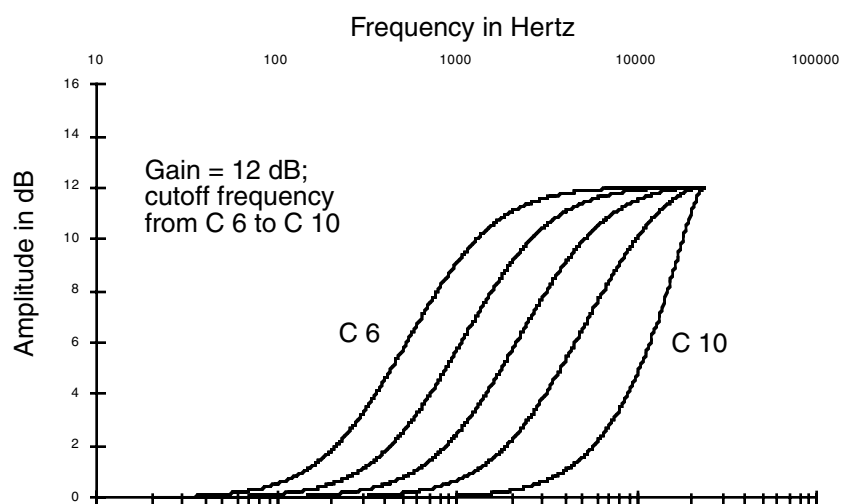
Bass Tone Control (PARA BASS)



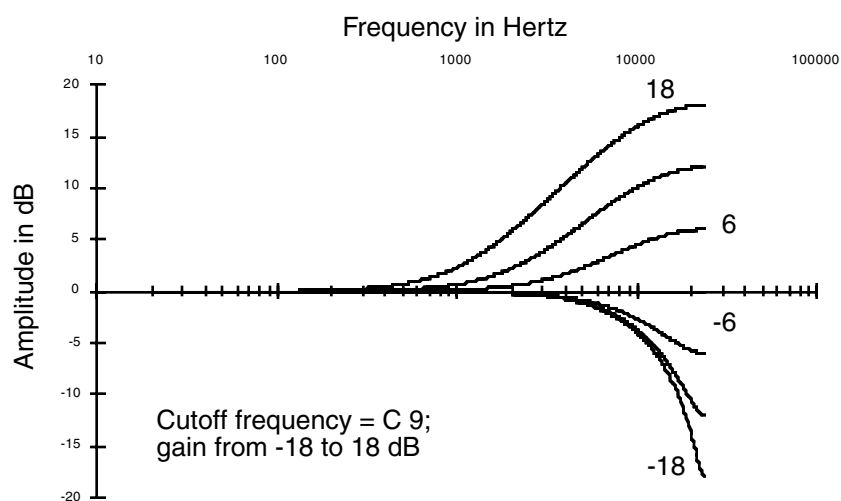
This is a two-stage function, with control-input pages for frequency and amplitude. These pages are the same as those for frequency and amplitude in PARA EQ. On the frequency control-input page, you'll set the cutoff frequency. For notes above this frequency, the amplitude setting has a diminished effect. 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.



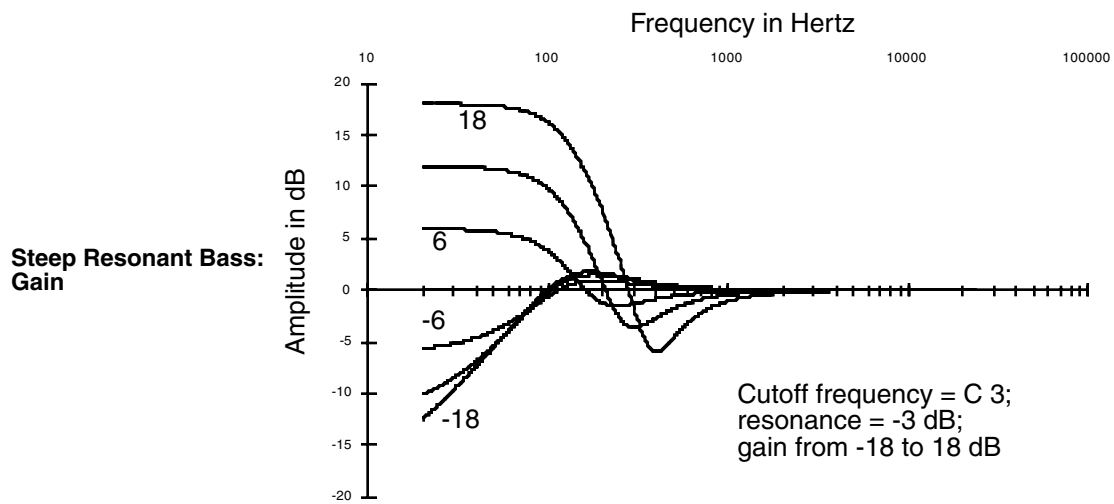
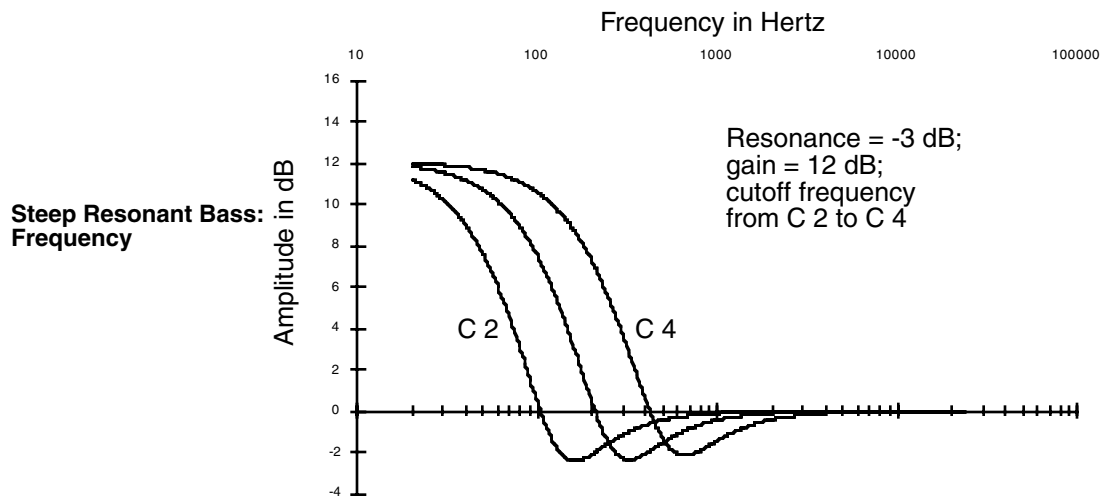
Treble Tone Control (PARA TREBLE)

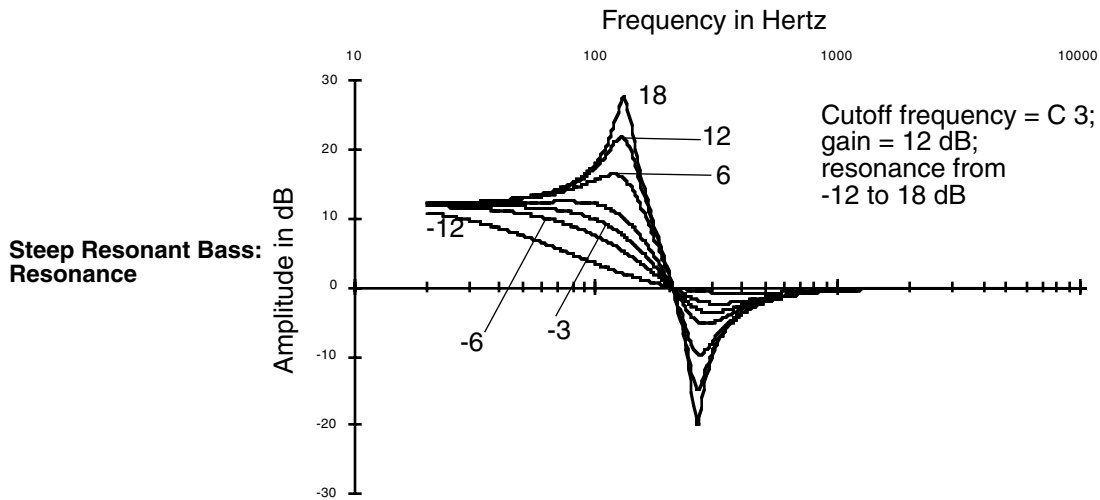


PARA TREBLE is very similar to PARA BASS; the only difference is that the amplitude setting affects notes *above* the cutoff frequency.



Steep Bass Tone Control (STEEP RESONANT BASS)





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
HdJust: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	UPPER AND LOWER AMP
AMP	BALANCE AND AMP
PANNER	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(FINHL AMP) <>Layer:1/1
Adj: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
DptCtl:OFF
KeyTrk: 0.0%key MinDpt:0%
VelTrk:0% MaxDpt:0%
Pad :0dB
<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/HMP) <>Layer:1/1
HdJust: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.

Algorithm 24



Algorithm 24



Algorithm 24



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:0SI Src1 :OFF
Fine :0ct Depth :0ct
FineHz: 0.00Hz Src2 :OFF
KeyTrk:0ct/key DptCt1: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	Two-parameter Shaper
Distortion	Wrap
Shaper	Lowpass Filter with Clipping
Double Shaper	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(HI)F2 STIM)<>LAYER:1/1
Coarse:C 4 262HZ Src1 :OFF
Fine :Oct Depth :Oct
KeyTrk:Oct/key Src2 :OFF
VelTrk:Oct DptCtl:OFF
Pad :Oct MinDpt:Oct
MaxDpt:Oct
<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

DSP Functions

The DSP Functions

```
EditProg:F2 DRU(HI-FREQ S1 UN)<>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 FREQ F2 DRU F3 AMP F4 AMP more>
```

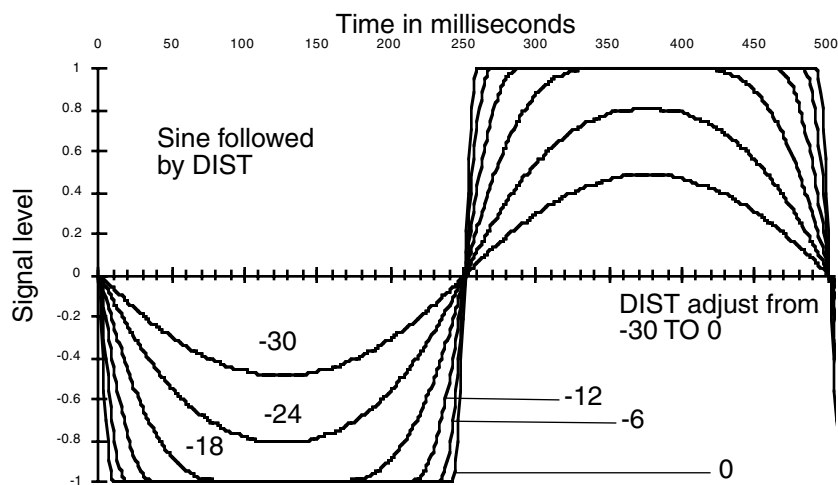
Parameter	Range of Values
Adjust	-96 to 48 dB
Keytrack Start	C -1 to C 9 unipolar, C -1 to C 9 bipolar
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

```
EditProg:F3 AMP(HI-FREQ S1 UN)<>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 FREQ F2 DRU F3 AMP F4 AMP more>
```

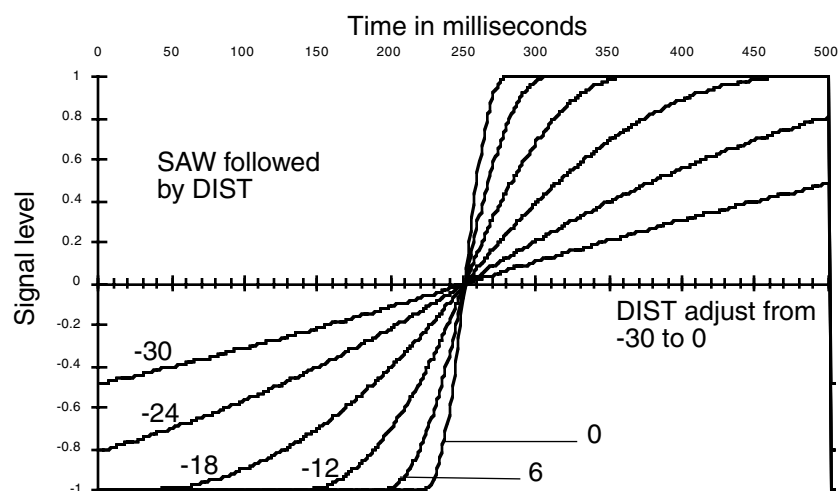
Parameter	Range of Values
Adjust	-96 to 48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Source 1	Control Source list
Source 1 DEPTH	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

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 DRU(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 DRU F2 F3 F4 AMP more>

```

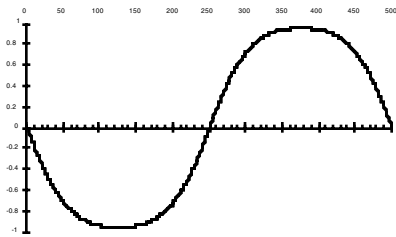
Parameter	Range of Values
Adjust	–96 to 48 dB
Keytrack Start	C -1 to C 9 unipolar, C -1 to C 9 bipolar
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
PAD	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

SHAPER

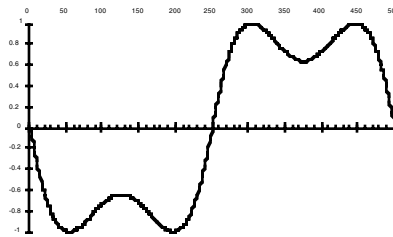
The effect of SHAPER can be very unpredictable, and the mechanics of its operation lend themselves toward explanations that are more numerical than verbal. The best way for you to get a feel for the SHAPER is to start with single-cycle waveform keymaps and experiment with different values for the parameters 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

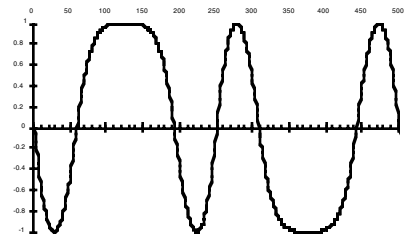
waveforms cycling at frequencies of 2 Hz. Of course, these are just a few of the countless modulations you can apply to different waveforms at different frequencies.



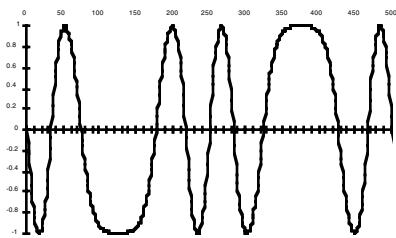
Amount = .1



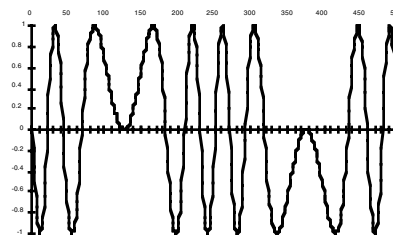
Amount = .2



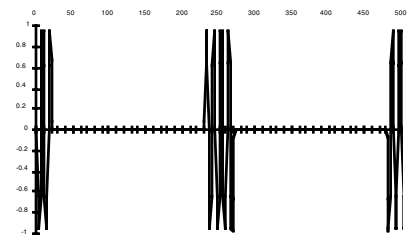
Amount = .375



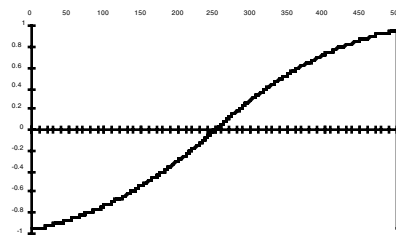
Amount = .625



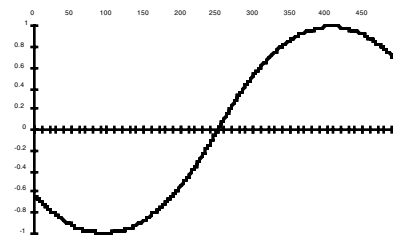
Amount = .1



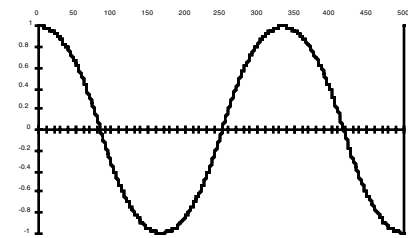
Amount = 4



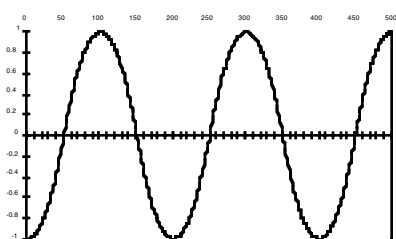
Adjust = .1



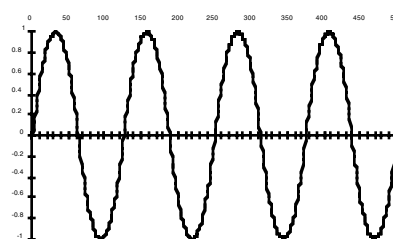
Adjust = .2



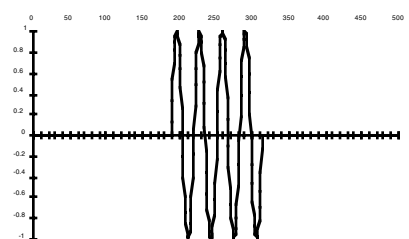
Adjust = .375



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).

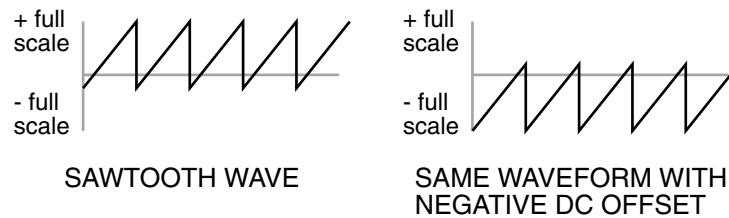
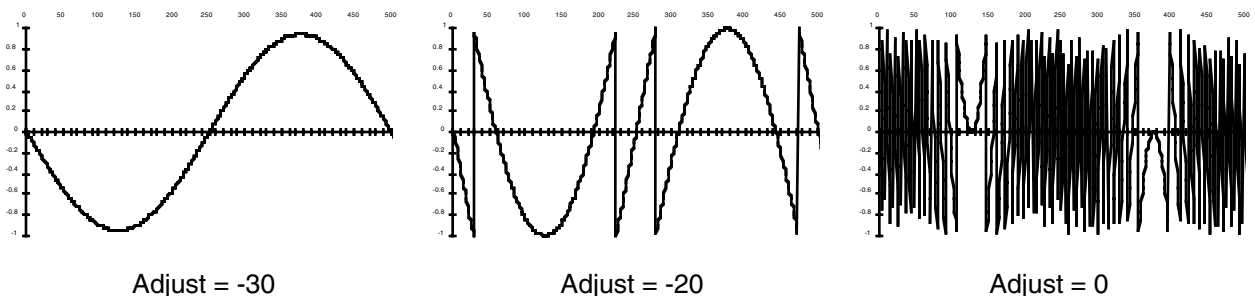


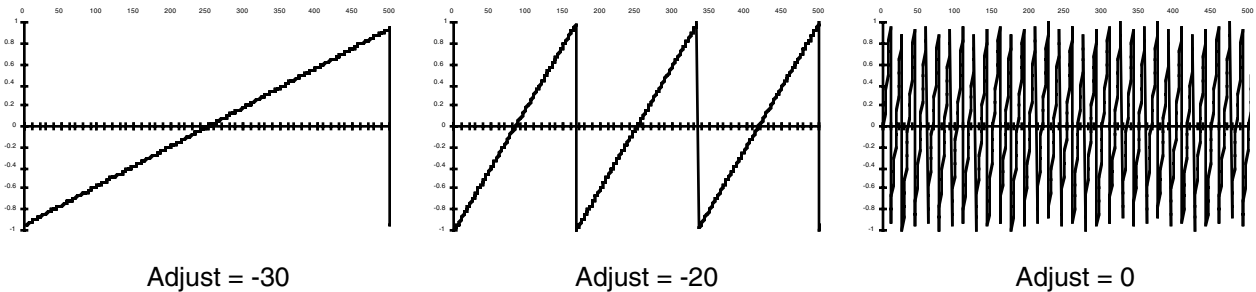
Figure 16-5 DC Offset

Waveform Wraparound (WRAP)

The next three graphs show the effect of various amounts of WRAP on a sine wave.



The following three graphs show the effect of WRAP on a sawtooth wave at the same frequency.



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.

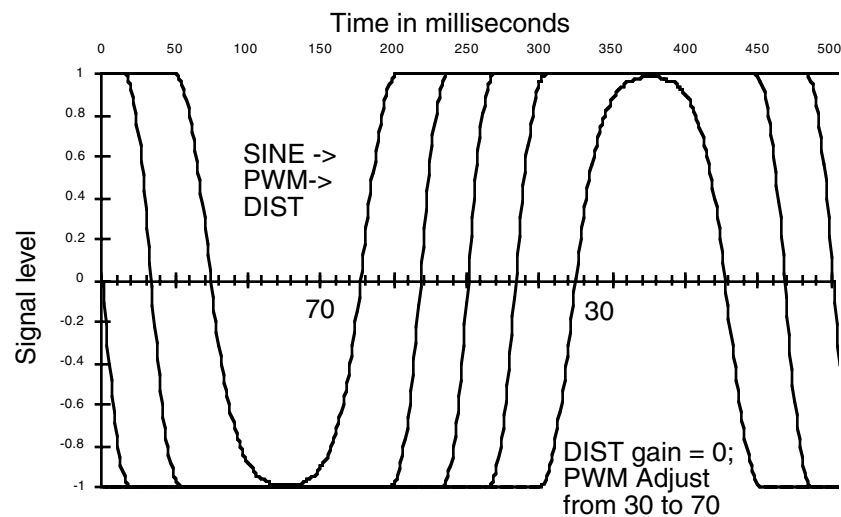
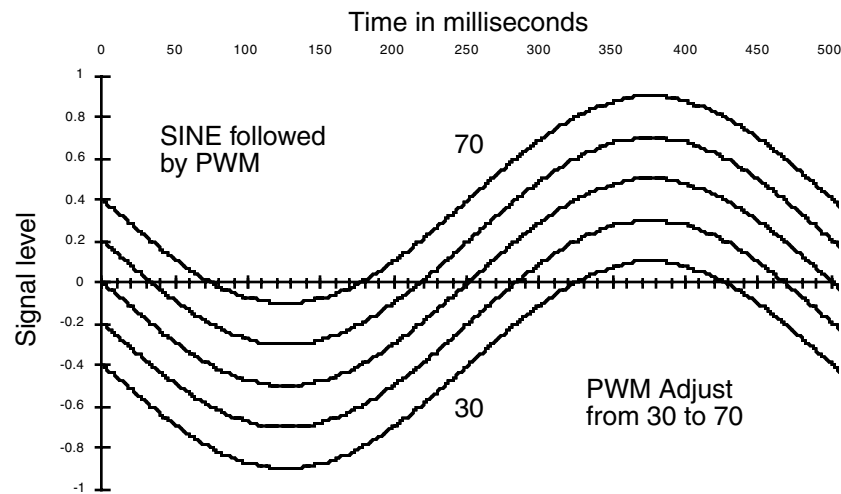
The bright buzzy nature of wrapped sounds is due to the discontinuities in the waveforms of the partials as they wrap around from positive full scale to negative full scale and vice versa. If you want, you can reduce or eliminate the buzz by sending the signal through SHAPER after it goes through WRAP (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 W10(PWM) <>Layer:1/1
Adjust:50% Src1 :LF01
Depth :25%
Src2 :OFF
DptCtl:OFF
KeyTrk: 0.0%/key MinDpt:0%
VelTrk:0% MaxDpt:0%
Pad :0dB
<more F1 W11 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)	x Shape Mod Osc
Added Sawtooth plus Shaper (SW+SHP)	+ Shape Mod Osc
Shape-modulated Oscillator	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

! AMP

x GAIN

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.

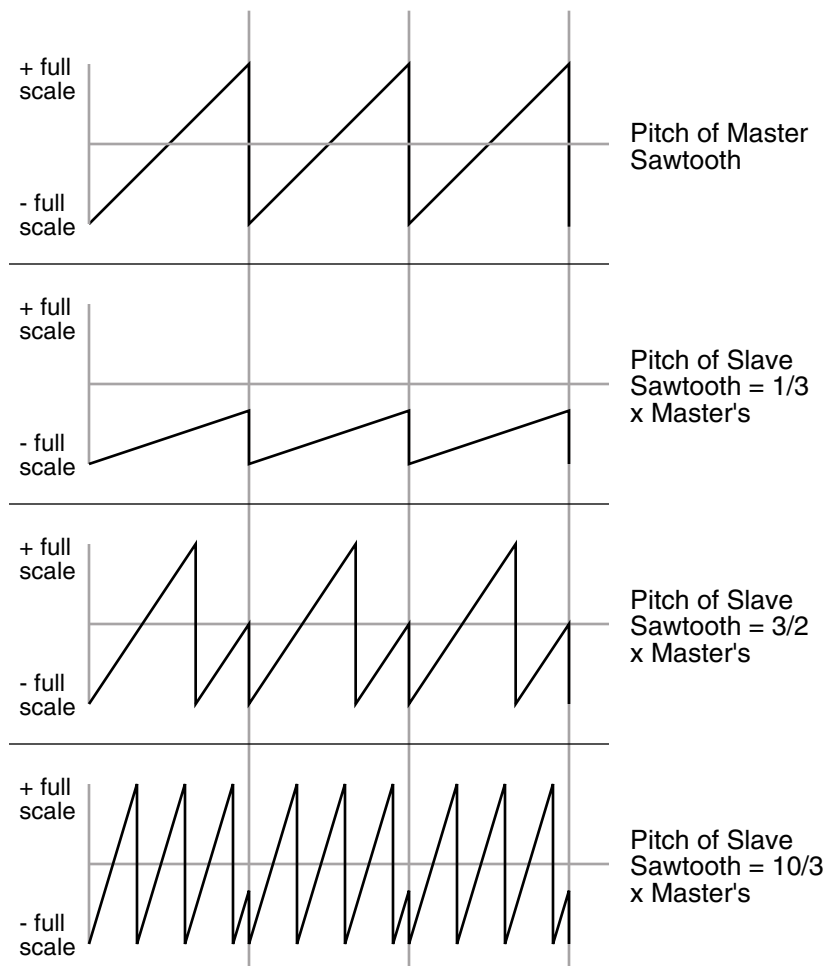


Figure 16-6 Hard Sync Functions

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 K2661 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 K2661’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 K2661’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 K2661’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 K2661’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 K2661 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
<more LFO ASR FUN UTRIG more>

```

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 K2661'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:

- the curve representing the effect of the FUN's equation on every possible input value
- one point on that curve, representing a single input value and the corresponding output value generated by the FUN's equation
- the input value represented by the point
- the output value represented by the point

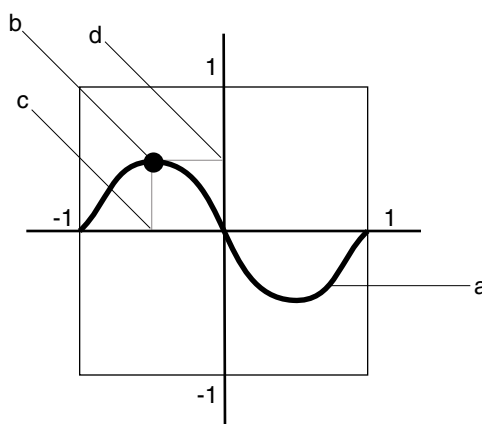


Figure 17-1 Interpreting FUN Graphs

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 -0.5, 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: $a \times (10^{(2 \times b) \div 100})$. This is an exponential curve. 10 is raised to the $(2 \times b)$ 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

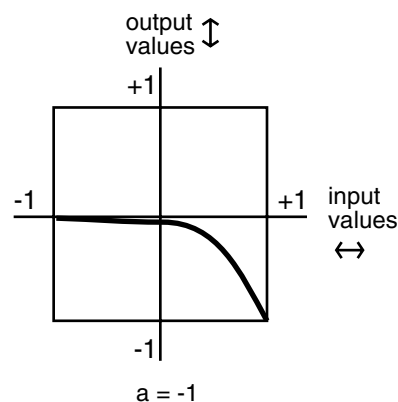
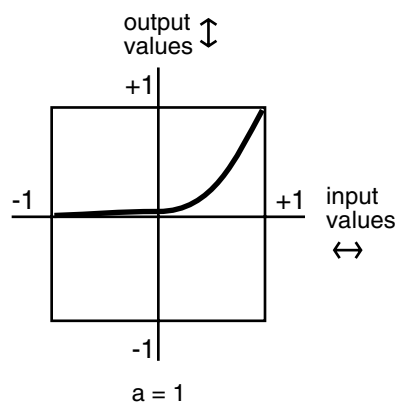


Figure 17-2 **a * 10 ^ b**

$|a + b|$

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.

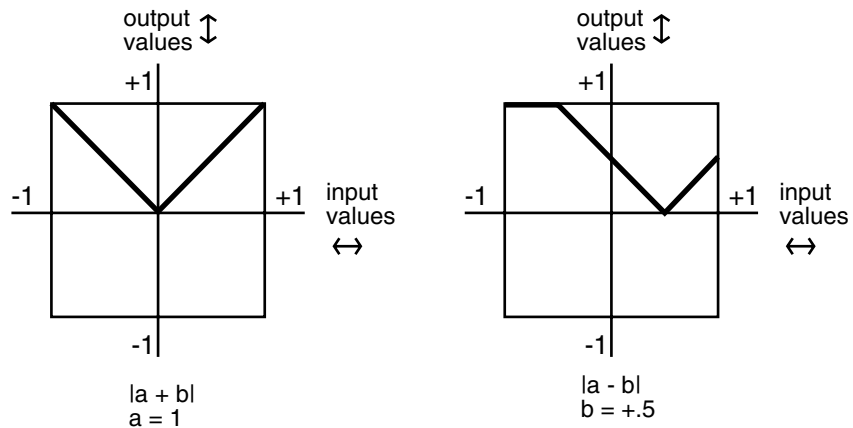


Figure 17-3 $|a + b|$, $|a - b|$

 $|a - b|$

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 K2661'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.

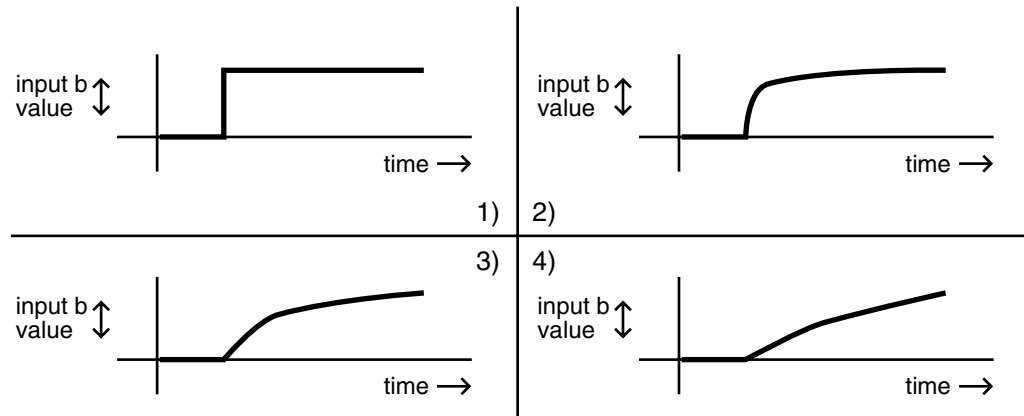


Figure 17-4 lowpass ($f = a, b$)

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.

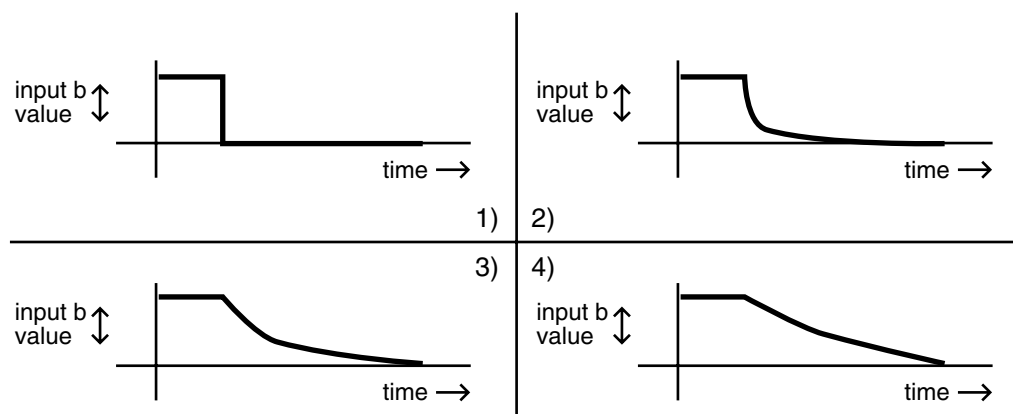


Figure 17-5 hipass ($f = a, b$)

$b / (1 - a)$

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 K2661 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 K2661 starts running the FUN. The first value for y is always 0. We'll assume the value of Input a is $+0.5$, and the value of Input b is $+1$. The first time the K2661 evaluates the FUN, the result of the equation is $0.5 \times (+1 - 0)$, or 0.5 . So the FUN's output value after the first evaluation is 0.5 . This becomes the new value for y , and when the K2661 does its next evaluation of the FUN, the equation becomes $0.5 \times (+1 - 0.5)$, or 0.25 . The resulting output value is 0.25 , which becomes the new value for y . For the next evaluation, the equation is $0.5 \times (+1 - 0.25)$, or 0.375 .

$(a + b)^2$

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.

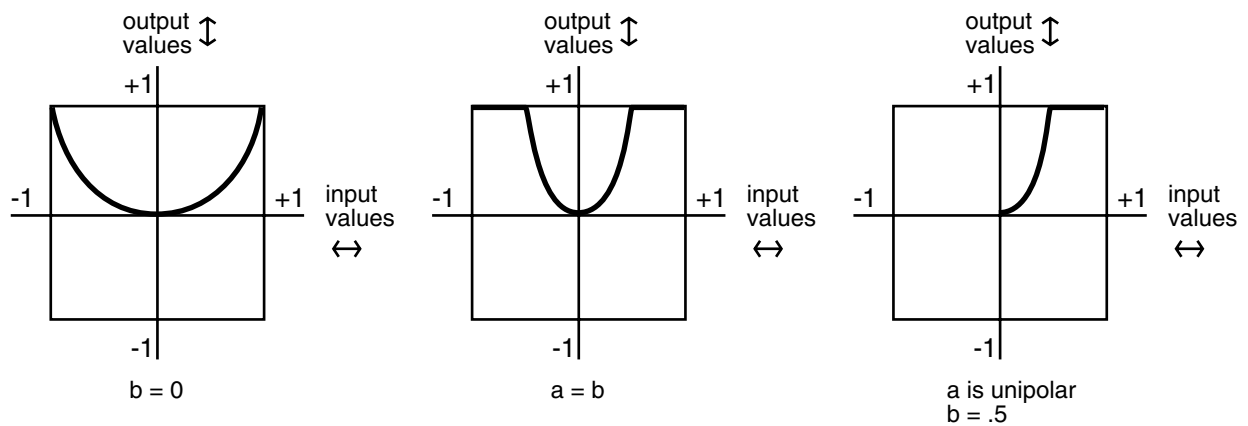


Figure 17-6 $(a + b)^2$

$\sin(a + b)$, $\cos(a + b)$, $\text{tri}(a + b)$

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.

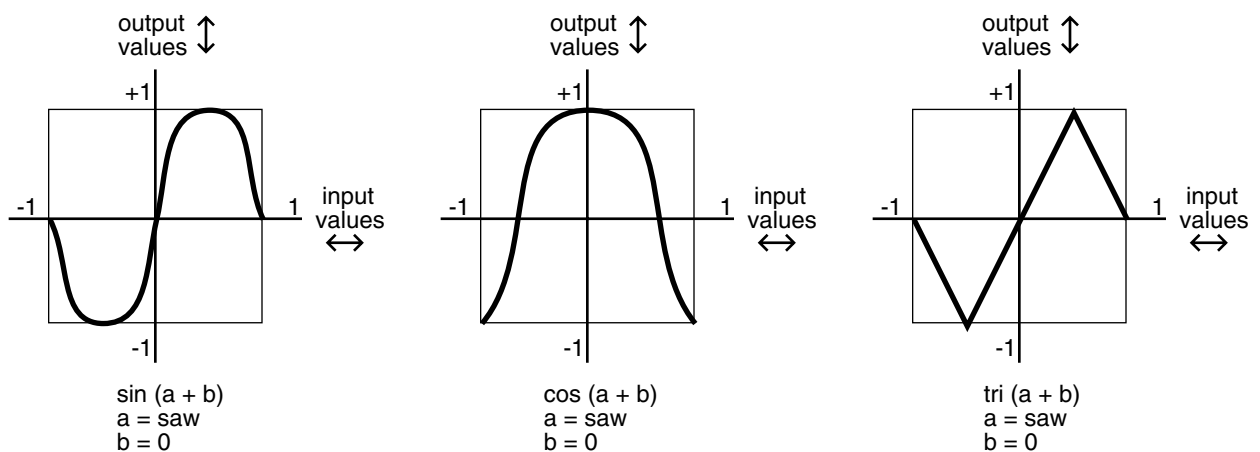
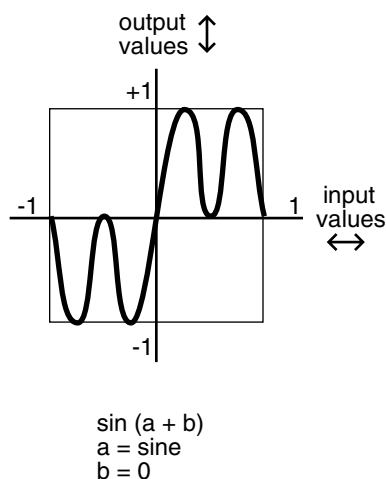


Figure 17-7 $\sin(a + b)$, $\cos(a + b)$, $\text{tri}(a + b)$

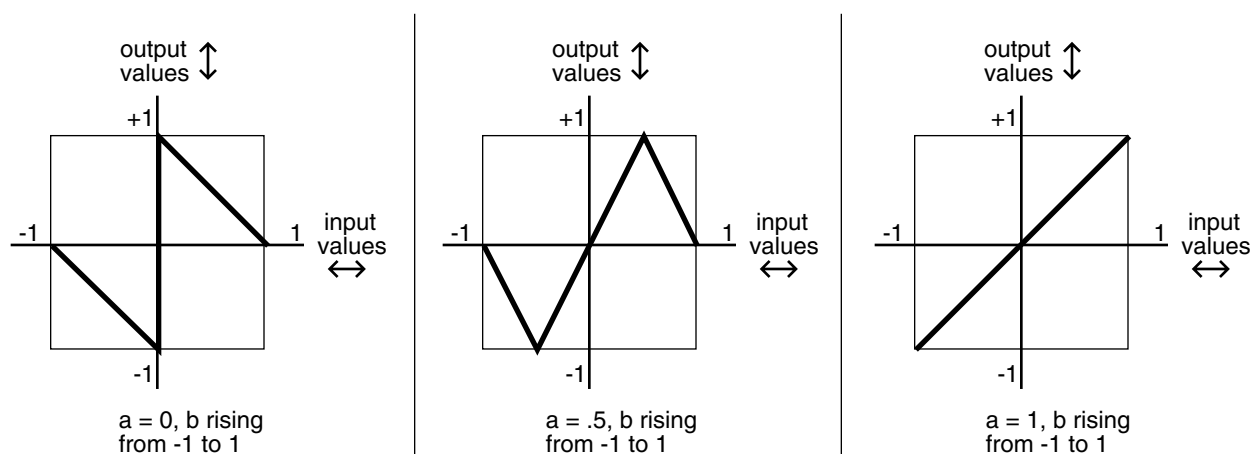
Figure 17-8 $\sin(a + b)$

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 $\text{warp1}(a, b)$

warp2(a, b)

We call this equation Slant-square.TM 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.

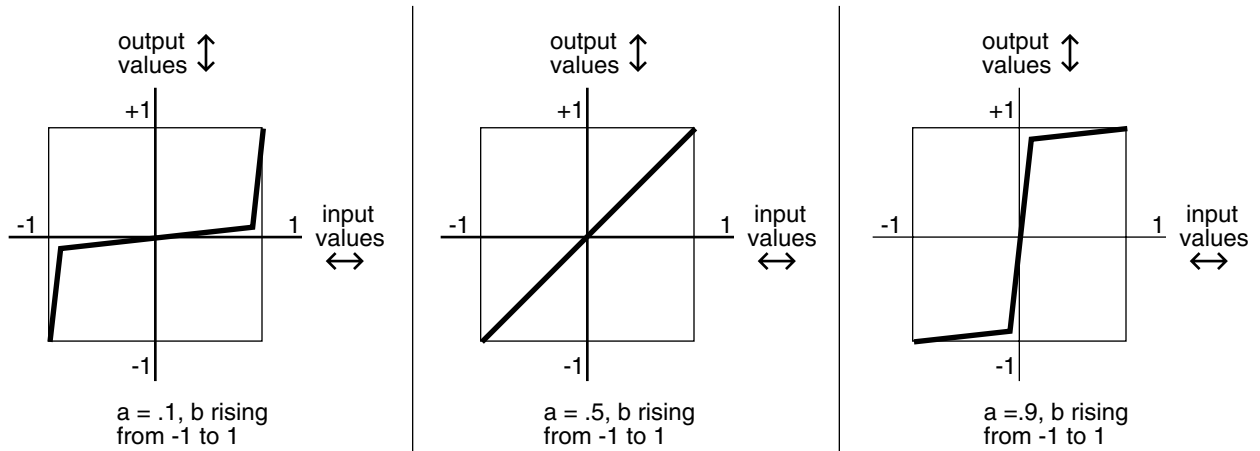
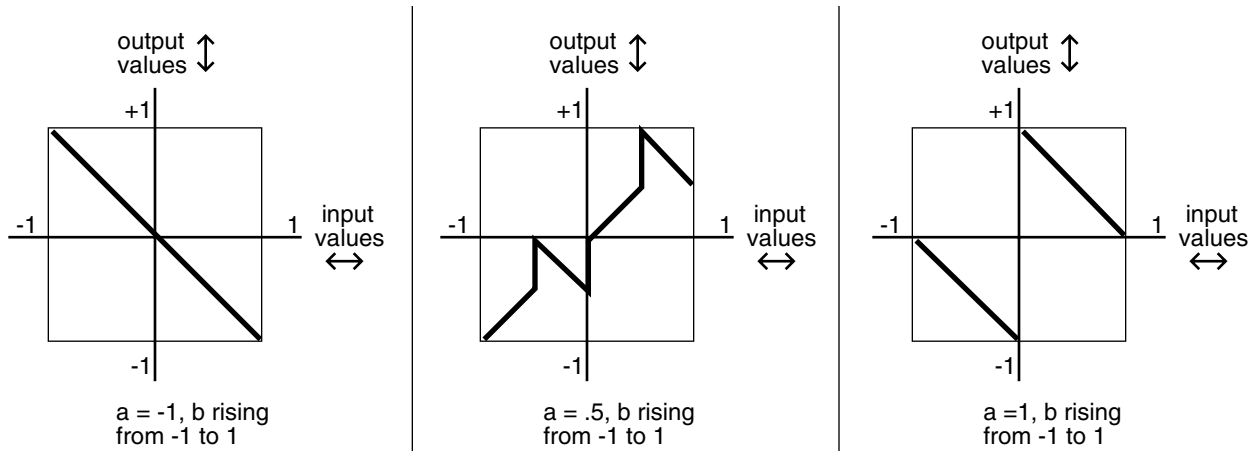


Figure 17-10 warp2(a, b)

warp3(a, b)

We call this one the Variable Inverter.TM 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.

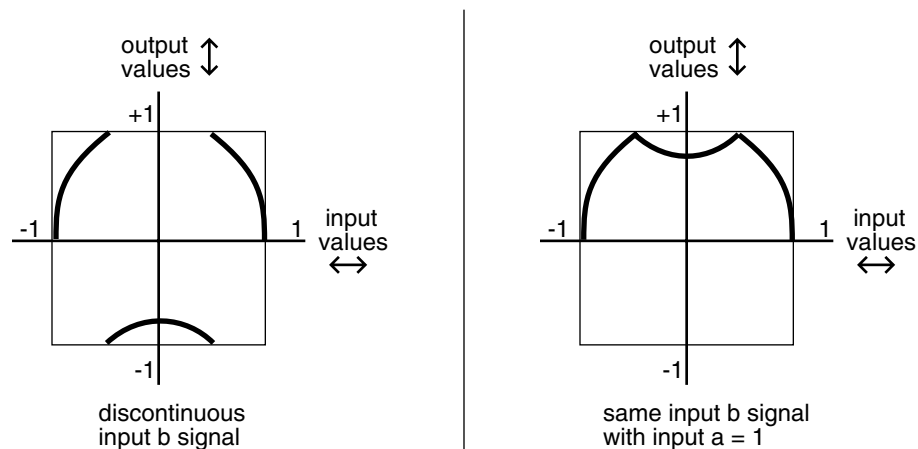
The XOR operation is a subprogram that applies a truth table to each of the digits in the binary numbers that represent the values of Inputs a and b. Each of these numbers is a string of 16 digits (bits); each bit is either a 0 or a 1. The subprogram looks at the first bit of each number. If they're 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 $\text{warp3}(a, b)$ **warp4(a, b)**

This equation, the Period Inverter,TM is based on repeated evaluations of the value of Input b . The K2661 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 $\text{warp4}(a, b)$ 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 $\text{warp4}(a, b)$

warp8(a, b)

This relatively simple equation is $a \times b \times 8$. 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.

$a \times b \times 8 =$	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.

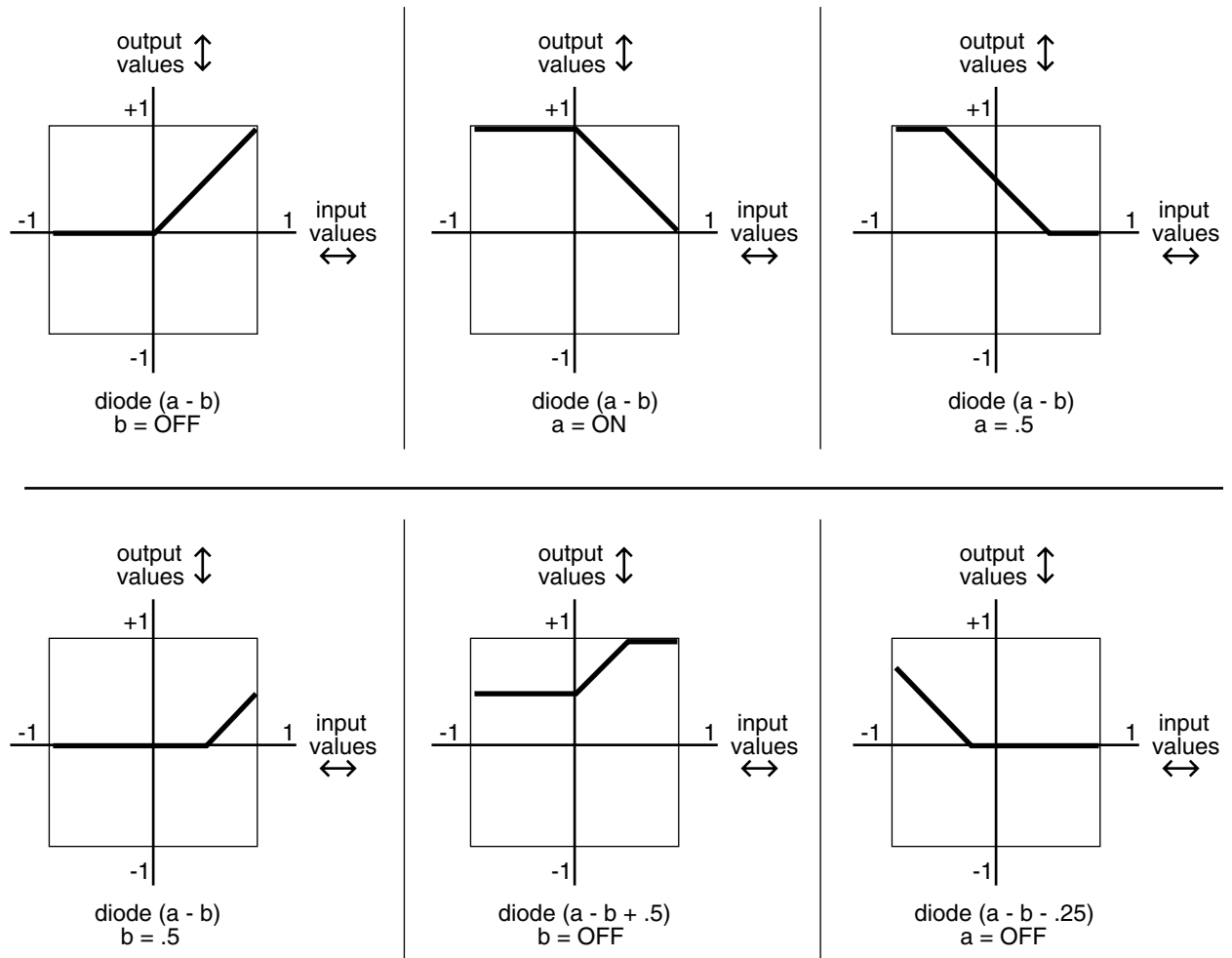


Figure 17-13 Diode Equations

- 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 K2661 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 K2661 needs to know the value of FUN1's output before it can evaluate FUN2, assigning the FUNs in this way will ensure that the K2661 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 K2661 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 addition to the editors accessible from their respective modes, there are three editors that enable you to modify other performance parameters of the K2661. 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 K2661’s other factory intonation tables, or create your 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 vn.mn**. This isn’t really an intonation table; it’s simply a convenient place for the K2661 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 K2661, you may see several intonation-table entries numbered as high as 24.

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.

Edit Intonation Table (cents)									
	0		0			0	0		0
0		0		0		0		0	
Name	Save	Delete	Dump						

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

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 K2661'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 K2661 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 K2661. If, for example, you're overdriving a DX7 that you have slaved to the K2661, try the map **6 Hard2**.

The VelocityMap parameter on the RECEIVE page in MIDI mode affects how the K2661 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 K2661 during playback in Song mode, or when playing a sequence from an external sequencer.

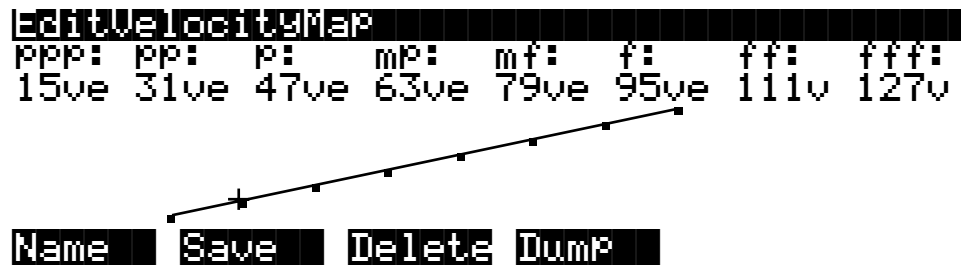
Here's a list of the velocity maps stored in ROM (the factory presets). Keep in mind, however, that you should use only the linear MIDI maps; the internal maps are meant for the K2661.

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 K2661'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**.



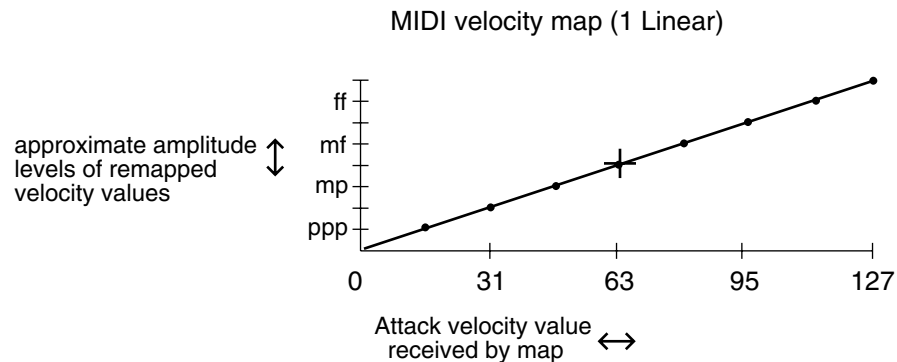
The top line of the display gives you the usual reminder of your location. The bottom line labels the soft buttons, which execute the normal librarian functions of naming, saving, or deleting velocity maps, or dumping them via SysEx. The values you see in this diagram are the settings for the velocity 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 K2661'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 K2661 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.

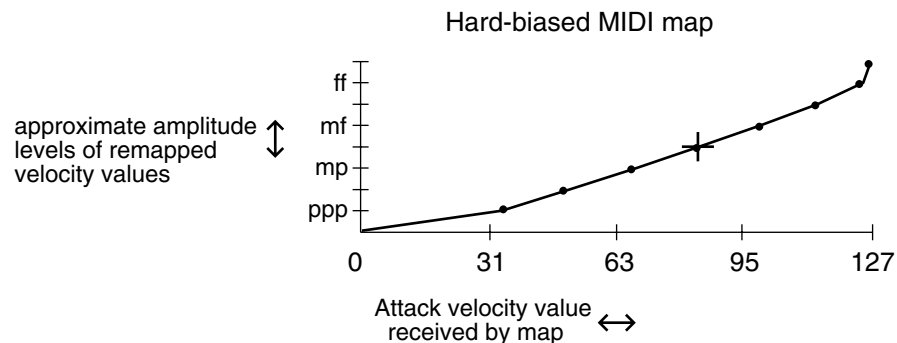
The curve that represents the velocity map is shown in the center of the page. Think of this curve as a plot on a graph, as shown in the diagram below. The horizontal axis gives the attack velocity values received by the velocity map. The vertical axis gives the approximate dynamic level at which the K2661 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 K2661'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.



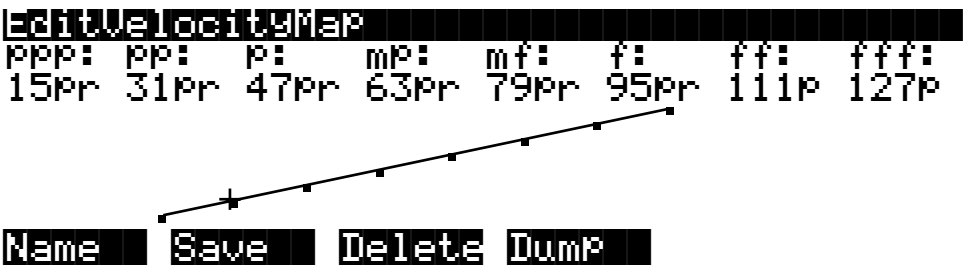
The Pressure Map Editor

Pressure maps function in much the same way as velocity maps, but they affect the K2661'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 K2661'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 K2661.

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.



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 parameters 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 K2661. The most common configuration is a pair of mono or stereo 1/4-inch cables connecting the Mix outputs of the K2661 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 and B), 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 K2661 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 K2661'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 K2661 (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 outputs; the parameters’ values specify which KDFX output bus gets routed to each of the analog outputs.

Using the Digital Outputs

Digital audio output is available at the ADAT / AES Out optical jack on the rear panel of the K2661. The format of the digital output stream can be chosen to match your other digital audio equipment. Formats supported by the K2661 include ADAT 8-channel, AES / EBU Professional 2-channel, and AES Consumer (also known as S/PDIF) 2-channel.

In ADAT digital format, the 8 channels correspond to the 4 stereo outputs found on the KDFX Output page. The AES 2-channel formats correspond to the output A stereo pair.

The table below summarizes the K2661's digital outputs:

	A Left	A Right	B Left	B Right	C Left	C Right	D Left	D Right
ADAT	1	2	3	4	5	6	7	8
AES	1	2						



Note: To use ADAT In, the ADAT Out cable must be connected to the sending device.

The word length of the digital data can be set to match your other equipment. It is generally best to use 24-bit digital formats, since it increases dynamic range and reduces the effects of noise. However, some older equipment may not be compatible with 24-bit data and therefore the K2661 supports 16 and 20 bit digital word lengths.

Choosing digital format and word length is done in Master Mode. See *Digital Output Format* on page 11-11 for details.

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 K2661 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 K2661 won't respond to the clock signal. Use a sample rate converter, if necessary, to match the K2661'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 K2661 is necessarily the master, because it can't be slaved to an external clock.

Audio Outputs

Using the Digital Outputs

Chapter 20

Programming Examples

The other chapters in this manual have described the K2661'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 5-8).

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 K2661'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   Odeg
LFO2:  2.00H  0.00H  OFF    Sine   Odeg

<more  LFO  ASR  FUN  UTRIG  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

<more>  LFO  ASR  FUN  UTRIG  more>

```

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         DptCt1:ASR2
VelTrk:0ct             MinDpt:0ct
                          MaxDpt:-10ct
<more>  ALG  LAYER  KEYMAP  PITCH  more>

```

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

ASR1:  Trig:  Mode:  Delay:  Attack:Releas:
      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 K2661 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 K2661 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 LUPHSS) >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.

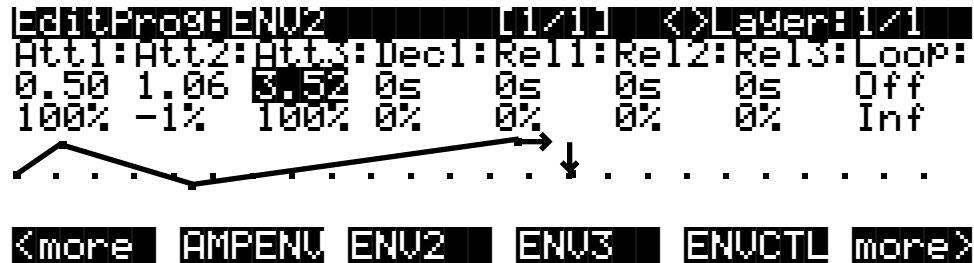
Slider A (Data / MIDI 06) will now affect the resonance of the filter. Be aware though that the change in resonance is subtle; resonance affects some waveforms more noticeably than others. You may want to set the Pad parameter on F1 FRQ page to a value of 18dB or so, then set the Gain parameter on the OUTPUT page to the same value; this reduces the potential of clipping due to resonance.

```

Edit Prog:F2 RES(4P LUPHSS) >Layer:1/1
Hdjust: 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.



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 Slider A (Data / 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.

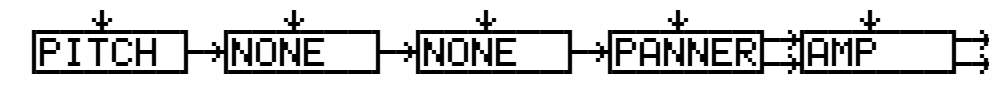
Now we'll apply the FUN to control the pitch of the sound as well. Press either of the **<more>** soft buttons until the **PITCH** soft button is visible, and press it to select the PITCH page. Select the Src1 parameter and set its value to **FUN1**. Select the Depth parameter and set its value to **400 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 your 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**.

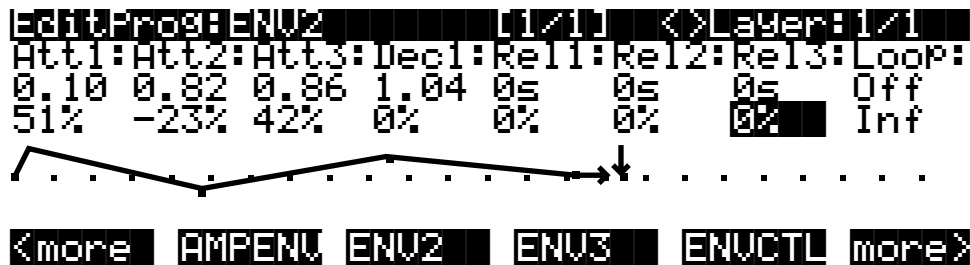
Notice how notes on the high end tend to break up. If you return to the F1 AMT page you can reduce this distortion and/or aliasing with the KStart and KeyTrk parameters.

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.



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 Slider A (Data / 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 K2661 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 K2661 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

```

↓

```

<more> AMPENV ENV2 ENV3 ENVCTL more>

```

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 K2661'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 K2661 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 SmartMedia Card

1. Insert the SmartMedia card provided with your K2661 into the K2661's SmartMedia slot.
2. Set the value of the CurrentDisk parameter to **SMedia**, 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 K2661 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 K2661 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 K2661 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

K2661 Boot Block

The Boot Block is a part of the K2661 software that lets you update the K2661 operating system and objects from either a SCSI device or the SmartMedia drive. The Boot Block also provides diagnostics options for service personnel and a reset option.



***Note:** Your K2661 comes from the factory with the operating system and ROM objects already installed. You do not need to run the K2661 Boot Block to start up a new K2661.*

Starting the Boot Block



***Note:** Before entering the Boot Block, you should turn down the volume on your PA or monitoring system.*

When you start the K2661, 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 Block. Otherwise, the K2661 will start up normally.

When the Boot Block starts, it will test the K2661’s files to make sure they are valid. Press the **OK** soft button to invoke the highlighted menu option.

```
Boot Block Main
K2661 Boot Block v1.00          Valid
K2661 Hardware Config v1.00    Valid
K2661 Engine v1.00             Valid
User Objects                    Valid

Install Run Reset DIAGS
```

Boot Block Main Menu

The Main Menu looks and functions similarly to other K2661 menus. Press one of these soft buttons to access a function:

- **Install** – lets you update the K2661’s operating system, Boot Block, and/or objects from a SmartMedia card or disk drive connected via SCSI.
- **Run** – starts the K2661 in its regular operating mode.
- **Reset** – performs a hard reset.
- **DIAGS** – runs diagnostic tests for troubleshooting issues with the K2661.

Updating K2661 Software

From time-to-time, Kurzweil Music Systems may release updates to the K2661's operating system, Boot Block, and/or objects. Generally, these will be posted at our web site:

<http://www.kurzweilmusicsystems.com/>

Use the Boot Block, as described in this section, to install any software update. Updates can include:

- K2661 Operating System
- K2661 Objects (programs, setups, songs, FX studios, etc.)
- K2661 Boot Block

To load from a SmartMedia card you'll first need a way to copy files (e.g., updates that you've downloaded from the Kurzweil web site) to a SmartMedia card. Fortunately, SmartMedia drives are readily and inexpensively available from a variety of sources.

File types

There are three different types of files, each distinguished by a unique three-character extension, that you may encounter when loading software into the K2661:

- **.KOS** – K2661 operating system files
- **.K26** – K2661 object files
- **.KBB** – K2661 Boot Block files

Always check for special instructions that may be included with a software update, since some updates may require a hard reset or other action.

To load new K2661 software:

1. Press the **Install** soft button on the Boot Block Main Menu.
2. The K2661 will display a screen that lets you indicate the device from which you are installing. Use the alpha wheel to scroll to the device name (either SMedia if you are installing from SmartMedia, or a SCSI ID if you are installing from a SCSI device).

If you are installing from a SCSI device, you may also need to set the SCSI ID of the K2661 on this page (SCSI ID 6 will be selected by default; if you've never changed the SCSI ID of your K2661, this should be alright).

3. After you press the **OK** button, the K2661 will list all the files in the top level directory on the SmartMedia card or SCSI device. You can use the alpha wheel, or the up, down, increment (+), or decrement (-) keys to navigate to the file(s) you want to load.
4. Use the Root, Parent, and Open soft buttons to move between directories:
 - **Root** takes you to the top level directory on the card.
 - **Parent** moves you up one directory level.
 - **Open** opens the currently selected directory.

5. Highlight a filename, then press the **Select** button. Press **Select** a second time to deselect an item.

You can select multiple files from the selection list. The status line at the top of the screen will show the current directory, how many files are in this directory, and how many files you have currently selected. An index counter shows you where in the list the cursor is currently located.

You can also double press the left and right cursor keys to select all the files in the current directory, with one exception. The exception is KBB files; if there is a single KBB file in the current directory, then it will be highlighted along with all the other files when you perform the double press. If there are several KBB files in the current directory, however, then the select-all double press will not select any of the KBB files.

6. Press the **OK** soft button when you're ready to load the selected file(s). The Boot Block will first test each segment of an OS or Object file before loading. If any problem is detected it will report that segment as corrupt.

When the load is complete, press the **Done** soft button, then press the **Run** soft button to start the K2661 in its regular operating mode.

Note: When you install a KBB file (Boot Block) the unit automatically restarts, running the new Boot Block.

Running Diagnostic Tests

The **DIAGS** soft button from the Boot Block Main Menu provides a list of available diagnostic tests. Since these tests are intended for service personnel, they are not described in this manual.

Resetting the K2661

Press the **Reset** soft button to perform a hard reset. This will restart your K2661, reset everything, and empty the unit's memory of any objects (program, setups, songs, etc.) you may have created. Therefore, you want to be absolutely sure that you want to perform a hard reset before you confirm this operation.

This option is the same as the Hard Reset option available from the Master page. There is also a less severe "soft" reset available by pressing **+/-**, **0**, and **Clear** simultaneously.

Index

Symbols

! AMP 16-54
+ SHAPE MOD OSC 16-53

Numerics

2PARAM SHAPER 16-49
2POLE ALPASS 16-18
2POLE LOWPASS 16-9
32 track sequencing 12-64
4POLE HIPASS W/SEP 16-15
4POLE LOPASS W/SEP 16-11
50% Weight 15-47

A

A/Dry->B parameter 15-36
A->B cfg parameter 15-36
Aborting a Macro Load 13-57
Aborting a Multiple File Load 13-22
Aborting a sample recording 14-6
Accessory disks 1-5
ADAT digital I/O 11-11
ADAT In 6-26, 11-11
Added waveforms 16-41
Adding layers 6-47
Adjust (parameter adjustment) 15-23
Adjust parameter (EnvCtl) 6-40
Adjust parameter (F4 Amp) 6-28
Adjust, coarse and fine 6-15
Adjusting sample volumes 14-38
AES/EBU Outputs 15-19
Aftertouch 11-2
AIFF files 13-73
AIFF-C files 13-73
Akai 13-74
Algorithm 9-3
Algorithm control inputs 6-13
Algorithm output 6-12
Algorithm page (Program Editor) 6-17
Algorithm programming 16-1
Algorithm wiring 6-12
Algorithms
 Common control parameters 6-14
 Double-output 6-31
 KDFX 15-35
 single and double output 6-12
All Controllers Off 10-6
All Notes Off 10-6
Allocation 15-15
Allocation parameter 9-22
Allpass filters 16-17
ALPASS 16-17
Alpha Wheel 2-6, 3-4
Alphabetic entry 3-5, 5-4
Alphanumeric pad 3-5
Alternative Attack parameter (Keymap) 6-26
Amp control 16-34

AMP MOD 16-55
AMP MOD OSC 16-53
AMP U AMP L 16-36
AMPENV page (Program Editor) 6-35
Amplitude envelope
 Decay segment 6-37
 Natural 6-35
 User 6-35
Amplitude envelope parameters 6-37, 6-38
Amplitude envelopes 6-35
Analog Mix Output 15-19
Append soft button 13-20
Are you sure? dialog
 suppressing with Confirm on Master page 11-4
Arp Active 7-36
Arpeggiator 7-35, 14-45
 Active 7-36
 Beats 7-38
 Duration 7-40
 Glissando 7-40
 Latch 7-37
 Limit Option 7-39
 Low Key (LoKey) and High Key (HiKey) 7-36
 Note Shift 7-38
 Order 7-38
 Shift Limit 7-38
 shifting notes 7-38
 Tempo 7-38
 Velocity 7-40
ARRANGE Page
 Song Mode 12-64
Arrangement Feature
 Edit Song Common parameters used with 12-43
Arrow Meter 15-7, 15-9
ASCII characters 5-4
ASR page (Program Editor) 6-42
ASR parameters 6-42, 6-43
Assign sample 14-37
Assigning samples to keymaps 14-38
Atk Rate 15-43
Atk Time 15-40
Attack 15-40
Attack parameter (ASR) 6-43
Attack Portamento parameter (Common) 6-34
Attack Time and Level parameters (AmpEnv) 6-37
Attack velocity 11-2
Audio cables 2-2
Audio Configurations 19-1
Audio levels 9-24
Audio output 6-29
Audio output parameters 6-30-??, 6-31
Auditioning Objects in RAM 13-27
Auditioning Samples from a Disk File 13-18
Auto Mode
 KDFX 15-34
AutoPan 15-46
Aux FX 15-4
Aux FX Bus 9-3
AuxFX Bus 9-28
AuxFX bus 15-10
AUXFX page (Studio Editor) 9-25

Index**B**

- backing up files 13-84
- Backup 13-66
- Balance and Amp (BAL AMP) 16-36
- BAND2 16-22
- BANDPASS FILTER 16-21
- Bandpass filters 16-21
- Bank Select
 - MIDI Receive page 10-8
- Bank Status Indicator 13-19
- Basic editing 5-1
- Basic FUN Equations 17-4
- Basic MIDI channel 10-5
- Bass Freq 15-38
- Bass Gain 15-39, 15-48
- Bass tone control 16-30
- Beat Volume Adjust 14-32
- Bipolar control sources 17-1
- Bipolar keystart 16-5
- Bipole 15-49
- Boldface italic type 1-4
- Boolean FUN equations 17-14
- Boot Loader A-1
- BOOT.MAC 13-70
- Bottom line of display 3-3
- Bounce
 - Song Mode TRACK Page 12-53
- brightness 2-3
- Build 15-37
- Building a Drum Program 20-9
- Building a keymap 14-39
- Bus Mods 15-12
- Bus Overrides 9-4, 15-26
- Bus overrides 15-12
- BusMod 9-4
- Buttons
 - Panic 6-10
- Bypass 9-12
- Bypass effects 5-8

C

- Cab Bypass 15-44
- Cab Preset 15-44
- Cabinet HP 15-45
- Cabinet LP 15-45
- Cabinet simulations 15-44
- Cables and Input Jacks 14-1
- Cancel
 - soft button 13-10
- Cancel soft button 13-20
- Carrier
 - ring modulation 15-47
- CD-ROM 13-1
- Center 15-48
- CenterGain 15-48
- CentrAtten 15-47
- Chan/Bank button
 - for bypassing 9-6
- CHAN/BANK buttons 3-3, 5-8
- Change
 - Song Mode TRACK Page 12-57
- Changing intonation key 11-4

- Changing song playback programs 12-20
- Channel/Program (CH/PROG) Page 7-6
 - Setup Mode 7-2
- Channels on and off 10-10
- Character substitution 3-7
- Characters 5-4
- Choose File Name Function 13-26
- Chorus 15-41
- Clear 14-22
- Clicking during portamento 6-34
- CLIP indicator
 - Sampling page 14-3
- Clock
 - Song mode 12-37
- Coarse adjust 6-15
- Common DSP control parameters 6-14
- COMMON Page
 - Song Mode 12-41
- COMMON page (Program Editor) 6-32
- Common parameters 6-33, 6-34
- Compare 5-8
- Compare button 15-5
- Complex Echo 15-38
- Compressing samples 14-33
- Compressors 15-39
- computers and hard disks 13-80
- Configuring audio output 6-29
- Configuring control sources 6-10
- configuring device drivers 13-80
- Confirmations 11-4
- Connecting MIDI 2-2
- Connecting SCSI devices 13-3
- Contour 15-43
- contrast 2-3
- Control input pages 16-2
- Control inputs 6-13
- Control Setup 6-10
 - Setup Editor 7-2
- Control sources
 - Common parameters 6-14
 - Configuring 6-10
 - dedicated to KDFX 15-24
 - FUNs 6-43
 - Key tracking 6-16
 - Source 1 and Source 2 6-16
 - Unipolar vs Bipolar 17-1
 - Velocity tracking 6-16
- Control sources—Overview 17-1
- Controlling KDFX 9-19
- Controlling LFO rate 6-41
- conventions for editing objects 5-1
- Copy
 - disk mode 13-66
 - on Object Utilities page 11-17
 - Song Editor TRACK Page 12-52
- copying files 13-84
- Copying layers 6-48
- Creating a Macro 13-44
- Creating Directories 13-11
- Creating layers 6-47
- Creating multi-root samples 14-12
- Crescendo/Decrescendo (Crescendo) 14-27
- Cross Couple parameter 15-36
- Crossfade 16-38

Crossfade and Volume Adjust Curves 14-34
 Crossfade Loop (XfadeLoop) 14-33
 Crossfade parameter (Output) 6-31
 Crossfade Sense parameter (Output) 6-31
 Crossover 15-45
 Crossover1 15-40, 15-48
 Crossover2 15-40
 Cursor buttons 3-3
 Curve n 15-45

D

Data entry 3-4
 DC Offset 15-42
 DC offset 16-49, 16-51, 16-52, 16-55
 Decay segment 6-37
 Decay Time and Level parameters (AmpEnv) 6-37
 Delay band 15-48
 Delay parameter (ASR) 6-43
 Delayed vibrato 20-1
 Delays 15-37
 DelayScale 15-38
 Delete 14-23
 on Object Utilities page 11-18
 Delete Layer soft button 6-48
 Delete soft button 6-47
 Deleting Banks in a Startup File 13-70
 Deleting Files and Directories 13-66
 Deleting layers 6-48
 Deleting objects 5-6, 11-11
 Deleting programs 6-47
 Density 15-37
 Dependent Objects
 loading 13-18
 Dependent objects
 saving 13-28
 Depth 15-25, 15-47
 Depth Control parameters (F4 Amp) 6-29
 designating hard disks as removable 13-80
 Designing with PAUs 15-16
 device drivers, configuring 13-80
 Device Manager (Windows) 13-80
 Diagnostic tests A-3
 Dialogs
 Save 5-3
 Diff Amt 15-38
 Diff Gain 15-48
 Diff Scale 15-36
 DiffBassF 15-48
 DiffBassG 15-48
 Digital Output 15-19
 Digital Output Format 11-11
 Digital Output Length 11-11
 Digital Outputs
 using 19-3
 Diode FUN equations 17-16
 Directories 13-4
 creating 13-11
 deleting 13-66
 Directory Selection Dialog 13-13
 directory structure 13-60
 Disk button 5-8
 Disk Drive Information 13-5
 Disk format requirements 13-1
 Disk Function soft buttons 13-6
 Disk Functions 15-20
 disk information 13-59
 Disk Mode 5-7, 13-2
 Disk mode 4-4, 13-83
 Disk Mode Functions 13-14
 disk partitioning (overview) 13-76
 disk space 13-59
 Disk Utilities 13-59
 Disks larger than 8 G 13-79
 Display 3-2
 Dist Drive 15-44
 Distortion 15-44
 Distortion (DIST) 16-45
 Dly Coarse 15-42
 Dly Fine 15-42
 DlyCrs 15-42
 DlyFi 15-42
 DOS 13-78
 Double button presses 3-5
 Double notch filter 16-22
 DOUBLE NOTCH W / SEP 16-22
 Double shaper 16-48
 Double-output algorithm 6-12
 Double-output DSP algorithms 6-31
 DRAWBR page (Program Editor) 6-51
 Drive 15-39
 Drum channel 11-2
 Drum Loop
 example of recording 12-4
 Drum programs
 creating 20-9
 Drum Tracks
 Song mode 12-44
 Drum trigger 14-4
 Dry Bal 15-37
 DSP algorithm control inputs 6-13
 DSP algorithms
 Common control parameters 6-14
 Double-output 6-31
 single and double output 6-12
 DSP control inputs 6-27
 DSP control parameters 16-2, 16-3
 DSP Functions 14-21
 DSP functions 16-6
 inputs and outputs 6-12
 Ducking 15-41
 Dummy effects bus 15-9, 15-16
 Dump
 on Object Utilities page 11-19
 Dump soft button 6-47
 Dumping programs 6-47
 Duplicate Layer soft button 6-48
 Duplicating layers 6-48
 Dynamic voice allocation 11-12
 Dynamics (Sample DSP function) 14-33
 DynamRange 15-42

E

EarRef Lvl 15-36
 EDIT button 3-4
 Edit button 5-8
 Edit compare 5-8

Index

Edit Song

- ARRANGE Page 12-64
- COMMON Page 12-41
- EVENT Page 12-45
- STEP Page 12-61

EditFXPreset Parameter pages 15-3

Editing 5-1

editing conventions 5-1

Editing KB3 programs 6-48

Editing Macros 13-53

Editing Samples 14-10

Editing VAST programs 6-11

Editors, nested 4-3

Effect Selection During Recording and Playback 12-20

EffectChan

- in Song mode 12-42

Effects buses 15-9

Effects button 5-8

Effects bypass 5-8

Effects Bypass button 15-5

Effects mode 4-4

Electrical grounding 2-2

Embedded Editors

- with KDFX 9-9

Enable MIDI channels 10-10

Enable parameter (Layer) 6-21

Enable Sense parameter (Layer) 6-21

Enhancers 15-39

Env Rate 15-44

Env Time 15-41

ENV2 and ENV3 pages (Program Editor) 6-38

ENVCTL page (Program Editor) 6-39

Envelope control 6-39

Envelope control parameters 6-40

Envelope Filter 15-43

Envelopes 6-35, 6-38, 20-4

EPS 13-75

EQ 15-7

- Bass tone control 16-30
- Mid-range parametric 16-28
- Parametric 16-25
- Steep bass tone control 16-32
- Treble tone control 16-31

EQ Morpher 15-39

EQ page (Program Editor) 6-59

EQ parameters 15-7

EQs 15-38

Equalization 9-23, 15-7, 16-25

Equalization parameters 15-7

Equalizers 15-38

Erase

- Song Mode TRACK Page 12-52

Erase soft button

- Song mode 12-30

Evaluation of FUNs 17-18

EVENT Page

- Song Mode 12-45

Everything

- loading objects as 13-21

Everything files 13-24

EXIT button 3-4

Exit values 7-25

Exp Atk 15-40

Exp Ratio 15-40

Exp Rel 15-40

Exp Threshold 15-40

Expanders 15-39

Expanse 15-37

Expansion 15-40

Exporting data 13-72

Extended program changes 10-12

Extended sample loop 14-15

Extensions

- used by or accepted by the K2600 13-7

F

F1–F3 pages (Program Editor) 6-27

F4 AMP page (Program Editor) 6-28

F4 Amp parameters 6-28, 6-29

FAT 13-80

FAT-16 13-78

FB2/FB1>FB 15-38

Fdbk Image 15-38

Fdbk Level 15-37, 15-41

FdbkComprs 15-39

FDISK 13-78

file allocation table (FAT) 13-80

File Backup 13-66

File Copy 13-68

File Formats

- supported by K2600 13-72

File Index 13-8

File List Dialog 13-7

File types A-2

Files

- deleting 13-66
- Everything 13-24
- Loading 5-7, 13-14
- Master 13-24
- moving between directories 13-63
- renaming 13-65
- Saving 5-7
- saving 13-24

Fill Mode 13-19

Fill soft button 13-20

Filter Type 15-43

Filters 15-43, 16-6

- Allpass 16-17

- Bandpass 16-21

- Double notch 16-22

- Highpass 16-14

- Lowpass 16-8

- Notch 16-19

- Twin peaks bandpass 16-23

Filters—A review 16-6

Find Files 13-60

Finding objects 3-7

Fine adjust 6-15

Fine Hz parameter (Pitch) 6-27

Flanger 15-42

Format 14-9

formatting

- hard format 13-77

formatting a hard disk with a computer 13-77

Formatting a SmartMedia Card 13-2

Formatting SCSI devices 13-4

Four-pole highpass filter 16-15

Four-pole lowpass filter 16-11
 free disk space 13-59
 Free utility 13-59, 13-63
 Freeze Pedal parameter (Layer) 6-22
 FreqScale 15-39
 Front panel navigation 3-2
 FUN diagram 17-3
 FUN equations 17-3
 FUN page (Program Editor) 6-43
 Function soft buttons 6-47
 Function Soft Buttons in the Sample Editor 14-10
 FUNS
 For sample and hold 20-6
 FUNs
 Basic equations 17-4
 Boolean equations 17-14
 Diode equations 17-16
 Evaluation sequence 17-18
 Random/Chaotic equations 17-15
 Sawtooth LFO equations 17-14
 Warp equations 17-11
 FX Bus 9-3, 15-2
 FX Bus 1 9-26
 FX Bus 2 9-27
 FX bypass 5-8
 FX Bypass button 15-5
 FX Channel 15-29
 FX Mode 9-4
 FX Mode in Song Mode 15-50
 FX Preset 9-3
 FX Presets 15-11
 FXBus Editor 9-22
 FXBus sends 15-7
 FXCtrl 9-9, 9-33
 FXLFO page (Program Editor) 6-46
 FXMod control assignments 15-22
 FXMod Diagnostic 15-49
 FXMOD2-4 pages (Program Editor) 6-46
 FXMode setting 9-4
 FXMODs 9-4
 FXSEND Page 9-24

G

g 13-2
 GAIN 16-37
 Gain 14-3, 15-8
 Gain parameter (Output) 6-31
 Gate Time 15-41
 Gated lowpass filter 16-13
 Gates 15-39
 General MIDI 11-13
 General synthesis concepts 16-1
 Glitches
 preventing 15-30
 with guitar or wind controller 11-14
 Globals parameter (Common) 6-34
 Glossary entries 1-4
 GM 11-13
 Grab
 Song Mode TRACK Page 12-60
 Graphic equalizer 15-38
 Grounding 2-2
 Guitar/Wind Controller Mode 11-14

H

Half Wts 15-47
 hard format 13-77
 Hard reset 11-14, A-3
 from boot loader A-3
 Hard sync functions 16-55
 SYNC M and SYNC S 16-55
 hard-disk information 13-59
 Headroom 15-42
 HF Damping parameter 15-36
 Hi Beam W 15-46
 Hi Gain 15-45
 Hi Rate 15-45
 Hi Res Delay 15-45
 Hi Size 15-45
 Hi Trem 15-45
 hierarchy of directories 13-60
 High and low notes, setting 6-47
 High frequency stimulator 16-42
 High Key parameter (Layer) 6-19
 High Velocity parameter (Layer) 6-20
 Highpass 15-44
 Highpass filters 16-14
 HiMic A 15-46
 HiMic B 15-46
 HIPAS2 16-15
 HIPASS 16-14
 HiPass1 and HiPass2 EQ 15-7
 HiResonate 15-45
 HiResXcurs 15-45
 HiShelf EQ 15-7
 Hold 15-37
 Hold Through Attack parameter (Layer) 6-22
 Hold Until Decay parameter (Layer) 6-23
 How the K2600 works 1-3

I

Ignore release 6-22
 ImageWidth 15-46
 Impact parameter (EnvCtl) 6-40
 ImpFX soft button 15-27
 Import Layer soft button 6-48
 Importing data 13-72
 Importing layers 6-48
 Importing samples from other manufacturers 13-74
 Importing Studios 15-27
 ImportKDFX page (Program Editor) 6-46
 In Gain parameter 15-35
 In Select 15-48
 In/Out parameter 15-36
 inaccessible partitions 13-82
 InfinDecay 15-37
 Info utility 13-59
 Input
 Sample mode page 14-3
 INPUT page (Studio Editor) 9-23
 Input Quantize 12-35
 Input section 15-6
 Inputs 9-27
 Insert 14-26
 Song Mode TRACK Page 12-54
 with Macros 13-55

Index

Insert FX Bus 9-3
Insert Zero 14-25
Intonation key 11-4
Intonation Table Editor 18-1
Intonation tables 11-2, 11-3
Introduction to editing 5-1
Intuitive data entry 3-6
Invert 14-24

J

Joining samples 14-12
Jump to page 5-8

K

K1000
 bank mode 7-9
K2000
 bank mode 7-9
K2600 Features 1-1
K2600 Overview 1-1
KB3 Effects 15-51
KB3 programs 2-5, 6-4, 11-2
KBB file type A-2
KDFX 9-24
 AUXFXpage 9-25
 controlling 9-19
 definition 9-1
 INPUT page 9-23
 real-time control of 15-20
KDFX Algorithms 15-35
KDFX in Auto mode 15-34
KDFX in Master mode 15-34
KDFX in Song mode 15-34
KDFX page (Program Editor) 6-44
KDFX pages
 in Program editor 15-22
KDFX structure 9-16
KDFX tutorial 9-21
Key range 14-37
Key tracking 6-16
Key tracking parameter (EnvCtl) 6-40
Key tracking parameter (F4 Amp) 6-28
Key Tracking parameter (Keymap) 6-24
Key/Velocity (KEY/VEL) Page 7-10
Keyboard
 using to trigger sampler 14-5
Keyboard naming 5-5
KEYCLK page (Program Editor) 6-54
Keymap 14-39
Keymap Editor 14-34, 20-9
 Assign sample 14-37
 New range 14-36
Keymap Editor Parameters 14-37
KEYMAP page (Program Editor) 6-23
Keymap parameter (Keymap) 6-23
Keymap parameters 6-23, 6-26
Keymaps 6-2
 multi-velocity 14-38
Keymaps, stereo 6-24
keymaps, stereo 6-31
Keystart

 bipolar 16-5
 unipolar 16-4
Keystart start (KStart) 16-4
KOS file type A-2
KRZ files
 saving and loading a macro table in 13-57
KSP file type A-2
Kurzweil program changes 10-12

L

L Diff Dly 15-38
L Invert 15-49
L Out Mode 15-49
L/R Delay 15-49
L/R Phase 15-41, 15-47
L/R Pre Dly 15-36
LaserVerb 15-42
Late Lvl 15-36
LateRvbTim 15-36
Layer delay control 6-20
LAYER page (Program Editor) 6-19
Layer parameters 6-19, 6-23
Layers
 Adding 6-47
 Deleting 6-48
 Duplicating 6-48
 Importing 6-48
 moving between in multi-layer programs 3-7
 Muting 5-8
 Setting ranges 6-47
LCD 3-2
Legato play parameter (Common) 6-33
Leslie effect 15-45
LF SAW 16-40
LF SIN 16-40
LF SQR 16-40
LFO Dpth 15-41
LFO Filter 15-44
LFO page (Program Editor) 6-40
LFO parameters 6-41, 6-42
LFO Phase 15-47
LFO PlsWid 15-44, 15-46
LFO Rate 15-41, 15-46
LFO Shape 15-44, 15-46
LFO Smooth 15-44
LFO on LRPPhs 15-41
LFOs 6-40, 6-42
Library 13-5
Library Disk 13-51
Line cord 2-2
List Utility 13-62
List utility 13-60
Live mode programs 2-6
Lo Beam W 15-46
Lo Gain 15-45
Lo Rate 15-45
Lo Res Dly 15-45
Lo Size 15-45
Lo Trem 15-45
Load Function Dialog 13-19
Loading a Macro File 13-46
Loading Dependent Objects 13-18
Loading files 5-7, 13-14

Loading Individual Objects 13-14
 Loading MIDI files 12-23
 Loading Older Setup Versions 7-2
 Loading Selected Entries from a Macro File 13-52
 Local control 10-2
 Local Keyboard Channel
 with Song Mode 12-1
 Lock parameters 10-11
 LoMic A 15-46
 LoMic B 15-46
 LOOP 14-17
 Loop Type parameter (AmpEnv) 6-37
 Looping samples 6-26
 LOPAS2 16-10
 LOPASS 16-8
 LoPass1 and LoPass2 EQ 15-7
 LoResonate 15-45
 LoResXcurs 15-45
 LoShelf EQ 15-7
 Low frequency waveforms
 Coarse adjust 16-41
 Sawtooth 16-40
 Sine 16-40
 Square 16-40
 Low Key parameter (Layer) 6-19
 Low Velocity parameter (Layer) 6-20
 Lowpass filters 20-4
 LP n Freq 15-45
 LP2RES 16-11
 LPCLIP 16-50
 LPGATE 16-13
 Lvl 15-46

M

Macintosh 13-78, 13-79, 13-82
 Macro Insert 13-55
 Macro Load
 aborting 13-57
 remote 13-58
 Macro On Indicator 13-6
 Macros 13-41
 editing 13-53
 MakeUp Gain 15-40
 MakeUpGain 15-40
 making backups 13-84
 making copies 13-84
 Marking pages 5-8
 Master button 5-8
 Master files 13-24
 Master mode 4-4
 KDFX 15-34
 Master mode page 11-1
 Master mode—Soft buttons 11-11
 Master Transpose 11-2
 Max Freq 15-43, 15-44
 Maximum delay parameter (Layer) 6-20
 Maximum Rate parameter (LFO) 6-41
 Memory bank ID allotments 13-34
 Memory banks 5-7
 Memory display 11-12
 Memory Limits 12-23
 Merge soft button 13-20
 Metering 15-17
 Mic Angle 15-45
 Microtonal tunings 11-2
 MID file type A-2
 MIDI
 All Notes Off 10-6
 Audio output gain 10-10
 Audio output pair 10-10
 Basic channel 10-5
 Channel enable 10-10
 Pan 10-10
 Parameter locks 10-11
 Program change formats 10-11
 Program changes 10-13
 Receive mode 10-5
 Reset channels 10-18
 Song recording 12-23
 Transmit parameters 10-1
 Volume 10-10
 MIDI button 5-8
 MIDI channel 10-2
 MIDI channel parameters 10-9
 MIDI connections 2-2
 MIDI control 10-2
 MIDI Control and Setup Mode 15-28
 MIDI dumps 6-47
 MIDI Faders Page 3-9
 MIDI files
 Loading 12-23
 MIDI loop 10-2
 MIDI mode 4-4
 Soft buttons 10-17
 MIDI program changes 10-3, 10-15
 MIDI Receive page 10-8
 Power Mode 10-9
 MIDI Receive parameters 10-5
 MIDI Type 0 and Type 1 Files 12-23, 13-73
 MIDIScope 11-12
 Midn Freq 15-39
 Midn Gain 15-39
 Midn Width 15-39
 Mid-range parametric EQ 16-28
 Min Freq 15-43, 15-44
 Minimum delay parameter (Layer) 6-20
 Minimum Rate parameter (LFO) 6-41
 MISC Page
 Song Mode 12-32
 MISC page (Program Editor) 6-57
 Miscellaneous (MISC) 14-13
 Mix 14-25
 Mix Beat 14-29
 MIX Page
 Song Mode 12-39
 Mixdown Page 3-9
 Mixers 16-37
 Mixers with nonlinear inputs 16-54
 !AMP 16-54
 AMP MOD 16-55
 x AMP 16-54
 x GAIN 16-54
 Mod Mode 15-47
 Mode 14-4
 Mode buttons 3-2
 Mode parameter (ASR) 6-43
 Mode Selection 3-1

Index

- Mode selection 4-1
- Modes 4-1
 - Program 6-1
- Modes, using 4-3
- Modulator
 - ring modulation 15-47
- Momentary
 - switch type 7-30
- mon 6-17
- Monitor (Mon) 14-5
- Mono inputs 15-7
- Mono sound systems 2-2
- Monophonic parameter (Common) 6-33
- Monophonic programs 6-33
- Monopole 15-49
- Morph A>B 15-39
- Move
 - on Object Utilities page 11-16
- Moving Files Between Directories 13-63
- MPressure 7-20
- MS-DOS File System Compatibility 13-71
- Multiband Compression 15-40
- Multiple Object Selector
 - entering selection criteria in 13-39
 - soft buttons 13-37
- Multiple Object Selector Page 13-35
- multiple removable disks 13-48
- Multiple Selection of Files to Load 13-22
- Multi-velocity keymaps 14-38
- Mute 9-12
- Muting layers 5-8
- Muting Setup zones 5-8
- Muting tracks 12-40

N

- N Fdbkn Dly 15-38
- Name
 - on Object Utilities page 11-18
- Name soft button 6-47
- Name Table 13-29
 - not loading 13-34
- Naming objects using the keyboard 5-5
- Natural amplitude envelope 6-35
- navigating directories 13-60
- navigating disk mode 13-83
- Navigation 3-2
- Nested editors 4-3
- New Layer soft button 6-47
- Noise generator 6-26
- Non-harmonic overtones
 - creating with ring modulator 15-47
- Nonlinear functions 16-42
 - 2PARAM SHAPER 16-49
 - DIST 16-45
 - HIFREQ STIMULATOR 16-42
 - LPCLIP 16-50
 - PWM 16-51
 - SHAPE2 16-48
 - SHAPER 16-46
 - WRAP 16-49
- Normalize 14-21
- Notch filters 16-19
- NOTCH2 16-20

- Note triggering 6-20
- Number of loops parameter (AmpEnv) 6-38
- Numeric entry 3-5

O

- Object type and ID 5-2
- Object Utilities 11-15
 - while editing objects 11-20
- Objects 5-1
 - auditioning 13-27
 - Deleting 5-6, 11-11
 - loading individually 13-14
 - Naming 5-3
 - Numbering in memory banks 13-34
 - RAM 5-2, 5-4
 - Renaming 5-3
 - ROM 5-2, 5-4
 - saving individually 13-25
 - soft button on Utility page 11-12
- Objects, storing 13-34
- Odd Wts 15-47
- OK
 - soft button 13-10
- One-pole allpass filter 16-17
- One-pole highpass filter 16-14
- One-pole lowpass filter 16-8
- Opaque parameter (Layer) 6-22
- Open
 - soft button 13-10
- operating system
 - updating A-2
- Origin 15-46
- Osc1 Freq 15-47
- Osc1 Lvl 15-47
- Osc1 Shape 15-47
- Osc1PlsWid 15-47
- Osc1Smooth 15-47
- Out Gain parameter 15-36
- Output assignment 7-8
- output assignments 6-29
- Output gain 10-10
- Output page 15-5
- OUTPUT page (Program Editor) 6-29
- OUTPUT page (Program Editor: KB3) 6-56
- Output pair 10-10
- Output section 15-18
- Outputs 9-29
- Overview 1-1
- Overwrt soft button 13-20
- OvFill soft button 13-20

P

- Pad 16-3
- Pad parameter (F4 Amp) 6-29
- Page Buttons 14-13
- Pages 3-2
 - Algorithm (ALG) 6-17
 - AMP (KB3) 6-56
 - AMPENV (Amplitude envelope) 6-35
 - ASR (Attack, Sustain, Release) 6-42
 - COMMON 6-32

- DRAWBR 6-51
- ENV2 and ENV3 (Envelopes) 6-38
- ENVCTL (Envelope control) 6-39
- EQ 6-59
- F1–F3 6-27
- F4 AMP 6-28
- FUN (Attack, Sustain, Release) 6-43
- FXLFO 6-46
- FXMOD2–4 6-46
- ImportKDFX 6-46
- Jumping to 5-8
- KDFX 6-44
- KEYCLK 6-54
- KEYMAP 6-23
- LAYER 6-19
- LFO 6-40
- Marking 5-8
- Master mode 11-1
- MIDI CHANLS 10-9
- MIDI RECV 10-5
- MIDI XMIT 10-1
- MISC 6-57
- OUTPUT 6-29
- OUTPUT (KB3) 6-56
- PERC 6-52
- PERC2 6-54
- PITCH 6-14, 6-27
- PITCH (KB3) 6-52
- Previous 5-8
- TONEWL 6-48
- VTRIG (Velocity Trigger) 6-44
- Pair parameter (Output) 6-30
- Pair Wts 15-47
- Pan 15-46
 - MIDI 10-10
- Pan band 15-48
- Pan lock 10-11
- Pan Mode parameter (Output) 6-30
- Pan parameter (Output) 6-30
- Pan Position
 - MIX page 12-39
- Pan Width 15-46
- Panic button 6-10, 10-18
- PANNER 16-35, 20-7
- PARA BASS 16-30
- PARA MID 16-28
- PARA TREBLE 16-31
- Parameter locks 10-11
- Parameters 9-3
 - Adjust 15-23
 - AmpEnv 6-37, 6-38
 - ASR 6-42, 6-43
 - Audio output 6-30, 6-31
 - Common 6-33, 6-34
 - DSP control 6-14
 - Envelope control 6-40
 - EQ 15-7
 - F4 Amp 6-28, 6-29
 - Keymap 6-23, 6-26
 - Layer 6-19, 6-23
 - LFO 6-41, 6-42
 - Source 15-24
 - Velocity triggers 6-44
- Parametric EQ 16-25
- Parametric equalizer 15-38
- ParaMid EQ 15-7
- Parent
 - soft button 13-10
- parent directory 13-83
- Parentheses, enclosing program name 6-9
- partition size 13-76
- partitioning (overview) 13-76
- partitioning with a PC 13-78
- partitions, inaccessible 13-82
- Patch Through
 - with Song mode 12-2
- Path 13-4
- PAU Allocations
 - changing 15-31
- PAUs 9-3, 15-15
 - designing with 15-16
- PAUs on the Aux Bus 15-16
- PC 13-78, 13-79, 13-80, 13-82
- PCH 10-12
- Pedals 7-46
- PERC page (Program Editor) 6-52
- PERC page (Program Editor: KB3) 6-56
- PERC2 page (Program Editor) 6-54
- Performance modes and Effects control 15-28
- Phase parameter (LFO) 6-42
- Pitch 15-47
- Pitch Bend Mode parameter (Layer) 6-20
- Pitch Bend Range parameter (Common) 6-33
- Pitch changing
 - with Live Mode 14-43
- Pitch control 16-34
- PITCH page 6-14
- PITCH page (Program Editor) 6-27
- PITCH page (Program Editor: KB3) 6-52
- Pitch Shift 14-29
- Pitch Shifter 15-47
- Pitch, amplitude, panner 16-34
- Pitcher 15-47
- Playback loops 6-26
- Playback Mode parameter (Keymap) 6-26
- Playing songs 12-19
- Plus /Minus buttons 3-4
- Polydistort 15-45
- Polyphony 6-2, 11-12
- Portamento click 6-34
- Portamento parameter (Common) 6-33
- Portamento Rate parameter (Common) 6-34
- Pos 15-46
- Power cable 2-2
- Power Mode
 - MIDI Receive page 10-9
- Pressure (Press) Page
 - Setup Mode 7-29
- Pressure Map Editor 18-6
- Pressure maps 10-3, 10-6
- Pressure sensitivity 11-2
- Previous page 5-8
- Processor Allocation Units
 - see PAUs
- Program button 5-8
- Program change formats 10-11
- Program change type 10-6
- Program changes

Index

- Extended 10-12
 - Kurzweil 10-12
 - MIDI 10-3, 10-13, 10-15
 - Quick Access banks 10-13, 10-15
 - Quick Access mode 8-1
 - Program Editor
 - ALG page 6-17
 - AMP page (KB3) 6-56
 - AMPENV page 6-35
 - ASR page 6-42
 - COMMON page 6-32
 - DRAWBR page 6-51
 - ENV2 and ENV3 pages 6-38
 - ENVCTL page 6-39
 - EQ page 6-59
 - F1-F3 pages 6-27
 - F4 AMP page 6-28
 - FUN page 6-43
 - FXLFO page 6-46
 - FXMOD2-4 pages 6-46
 - ImportKDFX page 6-46
 - KDFX page 6-44
 - KEYCLK page 6-54
 - KEYMAP page 6-23
 - LAYER page 6-19
 - LFO page 6-40
 - MISC page 6-57
 - OUTPUT page 6-29
 - OUTPUT page (KB3) 6-56
 - PERC page 6-52
 - PERC2 page 6-54
 - PITCH page 6-27
 - PITCH page (KB3) 6-52
 - Soft buttons 6-12
 - TONEWL page 6-48
 - VTRIG page 6-44
 - "wires" 6-12
 - Program lock 10-11
 - Program Mode
 - with KDFX 15-28
 - Program mode 2-4, 4-3, 6-1, 6-9
 - Soft buttons 6-10
 - Program mode page 6-9
 - Program names in parentheses 6-9
 - Programming algorithms 16-1
 - Programming FUNs 17-2
 - Programs 2-4
 - Adding layers 6-47
 - Deleting 6-47
 - Deleting layers 6-48
 - Dumping 6-47
 - Duplicating layers 6-48
 - Editing (KB3) 6-48
 - Editing (VAST) 6-11
 - Importing layers 6-48
 - KB3 2-5, 6-4
 - Live mode 2-6
 - Renaming 6-47
 - Saving 6-47
 - Selecting 2-6
 - VAST 2-4
 - Ptch Offst 15-47
 - Pulse Width 15-44, 15-46
 - Punctuation 5-4
 - PWM 16-51
- ## Q
- QA 0-127 10-15
 - Quantize
 - Input 12-35
 - Song Mode TRACK Page 12-55
 - Quantize + Flange 15-42
 - Quartr Wts 15-47
 - Quick Access bank program changes 10-13, 10-15
 - Quick Access button 5-8
 - Quick Access Editor 8-1
 - Quick Access mode 2-7, 4-4, 8-1
 - Quick Mute
 - Song Mode MIX Page 12-40
 - Quick Mute Track Bank Indicator
 - MIX page 12-40
 - Quick Scrolling to Subdirectories 13-11
- ## R
- R Diff DI 15-38
 - R Invert 15-49
 - R Out Mode 15-49
 - RAM objects 5-2, 5-4
 - Random/Chaotic LFO FUN equations 17-15
 - Rate 14-3
 - Rate Control parameter (LFO) 6-41
 - Rate Scale 15-46
 - Ratio 15-40
 - Real-time Control of KDFX 15-20
 - Recording Multi-timbral Sequences via MIDI 12-23
 - Recording songs 3-10
 - Recording songs via MIDI 12-23
 - Reference Quantize
 - Song Mode TRACK Page 12-56
 - Region/Criteria window
 - Song mode 12-49
 - Rel Rate 15-43, 15-44
 - Rel Time 15-40
 - Release 15-40
 - Release parameter (ASR) 6-43
 - Release Time and Level parameters (AmpEnv) 6-37
 - Releasing ASRs 6-43
 - Relink-by-Name 13-30
 - Remap
 - Song Mode TRACK Page 12-59
 - removable media 13-48, 13-80
 - Renaming Files 13-65
 - Renaming multiple objects 3-7
 - Renaming objects 5-3
 - with Name utility 11-18
 - Renaming programs 6-47
 - Repeating ASRs 6-43
 - Resample 14-28
 - resampling K2661 internal audio data 14-9
 - Reset
 - hard 11-14, A-3
 - soft A-3
 - Reset MIDI channels 10-18
 - Resetting the KSP8 A-3
 - ResH/LPhs 15-45

Resonance 15-43
 Resonant Filter 15-43
 Retrigger 15-41, 15-44
 Reverb types 15-36
 Reverse 14-24
 Reversing samples 6-26
 Ribbon Configuration (RIBCFG) Page
 Setup Mode 7-42
 RIBBON Page
 Setup Mode 7-42
 Ribbons
 Center 7-43
 Large Ribbon Configuration 7-42
 Position Mode (PosMode) 7-42
 Ribbon Configuration (RIBCFG) Page 7-42
 Spring 7-43
 Ring Modulation 15-47
 RMS Settle 15-49
 Roland 13-75
 ROM objects 5-4
 ROM objects, saving 5-2
 Room Type 15-36
 Root
 soft button 13-10
 root directory 13-83
 Rotating Speakers 15-45
 Roto InOut 15-46
 Rvrb Time 15-36

S

Safety precautions 2-2
 Sample 14-5
 Sample and Hold (FUNs) 20-6
 Sample Editor 14-39
 Sample memory 14-4
 Sample recording
 aborting 14-6
 Sample Skipping parameter (Keymap) 6-24
 Sampler
 entering from Master Mode 11-13
 triggering from keyboard 14-5
 Samples
 Adjusting volume 14-38
 Assigning to keymaps 14-38
 Playback mode 6-26
 triggering from analog inputs 14-40
 triggering from audio signals 14-4
 Tuning 14-38
 Sampling Analog Signals 14-2
 Save dialog 5-3
 Save soft button 6-47
 Saving and naming Objects
 Saving 5-3
 Saving Dependent Objects 13-28
 Saving Files 13-24
 Saving files 5-7
 Saving Individual Objects 13-25
 Saving Master and Everything Files 13-24
 Saving objects
 RAM 5-4
 ROM 5-4
 Saving programs 6-47
 Saving RAM objects 5-2

Saving ROM objects 5-2
 Saving Studios 15-20
 SAW 16-40
 Sawtooth LFO FUN equations 17-14
 SC Input 15-40
 SCSI 13-80
 SCSI chain 13-80
 SCSI connections 2-3
 SCSI devices 13-1
 Formatting 13-4
 Termination 13-3
 SCSI ID 10-6
 SCSI IDs 13-4
 SCSI Musical Data Interchange 1-2
 Search function 3-7
 Select soft button 13-9
 Selecting a Song for Playback 12-19
 Selecting Modes 3-1
 Selecting modes 4-1
 Selecting parameters 3-2
 Selecting programs 2-6
 Selection by Dependents 13-40
 Selection by Type/Range 13-39
 Sends 9-24
 Sequencer
 32 track sequencing 12-64
 controlling KDFX with 9-20
 tutorial 12-1
 using KDFX with 15-29, 15-32
 Set drawbars 6-51
 Set Range (SetRng) soft button 6-47
 Setting FXBus levels 9-24
 Setting layer range 6-47
 Setting Up For Sampling 14-1
 Setup button 5-8
 Setup Editor 7-5
 Clear Soft Button 7-45
 Control Setup 7-2
 Copy and Paste Soft Buttons 7-44
 Delete Soft Button 7-43
 Delete Zone (DelZn) Soft Button 7-44
 Dump Soft Button 7-44
 Duplicate Zone (DupZn) Soft Button 7-44
 Import Zone (ImpZn) Soft Button 7-44
 Local Program (LocalPrg) 7-6
 Name Soft Button 7-43
 New Zone (NewZn) Soft Button 7-44
 Set Range (SetRng) Soft Button 7-45
 Soft buttons 7-43
 Setup Mode
 ARPEG Page 7-35
 Channel/Program (CH/PROG) Page 7-2
 Pressure (Press) Page 7-29
 Ribbon Configuration (RIBCFG) Page 7-42
 RIBBON Page 7-42
 using with KDFX 15-27
 WHEEL Page 7-29
 Setup mode 4-3, 7-1
 Setups 2-6
 Aux Bend 1 7-19
 Aux Bend 2 7-19
 BEND Page 7-18
 Bend Range 7-18
 COMMON Page 7-33

Index

- Continuous Control Pedal (CPEDAL) Page 7-27
- Continuous Controller Parameters 7-23
- Controllers 7-19
- Controlling KB3 features 20-11
- Curve (Curv) 7-24
- Destination 7-9, 7-30
- Destination (Dest) 7-24
- Editing Hints and Suggestions 7-45
- Entry (Ent) and Exit States 7-31
- Entry Pan, Exit Pan 7-17
- Entry Program Change (EntryProgChg) 7-9
- Entry values 7-25
- Entry Volume, Exit Volume 7-17
- Footswitch (FOOTSW) Page 7-31
- Low and high key 7-10
- Low Velocity (LoVel), HighVelocity (HiVel) 7-16
- MIDI Bank 7-7
- MIDI Bank Mode 7-9
- MIDI channel 7-6
- MIDI Control Source List 7-21
- MIDI Program (MIDIProg) 7-8
- Muting zones 5-8
- Note Maps 7-11
- Off Value 7-31
- Offset (Add) 7-24
- On Value 7-31
- Pan/Volume (PAN/VOL) Page 7-17
- physical controllers 7-20
- RIBBON Page 7-27
- Save Soft Button 7-43
- Scale 7-24
- Selecting via MIDI 10-13, 10-15
- SLIDER and SLID/2 Pages 7-25
- SONG 7-33
- Status 7-8
- Switch controllers 7-30
- SWITCH Page 7-32
- Switch Type (SwType) 7-30
- Transpose 7-11
- Transposing 7-2
- using KDFX with 15-50
- Velocity Curve 7-14
- Velocity Offset 7-12
- Velocity Scale (VelScale) 7-12
- Zone Arpeggiation (ZoneArpeg) 7-10
- SHAPE MOD OSC 16-53
- Shape parameter parameter (LFO) 6-41
- SHAPE2 16-48
- SHAPER 16-46, 20-7
- sharing your hard disk with the K2600 and a computer 13-80
- Shift
 - Song Mode TRACK Page 12-57
- Shifting notes
 - in the Arpeggiator 7-38
- Signal Delay 15-40
- Signal Dly 15-40
- Signal-to-noise ratio 2-3
- SIMMs 1-2, 14-4
- SINE 16-40
- Single-output algorithm 6-12
- Size of algorithms 9-3
- size of partitions 13-76
- Size Scale 15-37
- Sleep soft button 13-6
- Sliders 7-20
- SmartMedia 2-3, 13-2
 - formatting 13-2
- SMDI 1-2
- Smooth 15-40
- Smooth Rate 15-43
- Smooth Time 15-40
- Smth Rate 15-44
- SngCtl
 - Setup Common Page 7-34
- Soft buttons 3-3
 - Disk Mode page 13-6
 - Keymap Editor 14-36
 - Master mode 11-11
 - MIDI mode 10-17
 - in Program Editor 6-12
 - in Program Mode 6-10
 - Setup Editor 7-43
 - Special functions 6-47
- Soft Buttons on the DSP Page 14-20
- Soft reset A-3
- Soft Through
 - with Song Mode 12-2
- software updates A-2
- Software Upgrades 2-8
- Soloing current track 12-41
- Song button 5-8
- Song Control
 - Setup Common Page 7-34
- Song Editor 12-41
 - TRACK Page 12-48
- Song Mode 12-1, 15-50
 - KDFX 15-34
 - MISC Page 12-32
 - MIX Page 12-39
- Song mode 4-4
- Song playback 12-19
- Songs
 - Memory limits 12-23
 - Recording 3-10
 - Recording via MIDI 12-23
 - Synchronizing 12-20
- Sostenuto Pedal parameter (Layer) 6-22
- Sound Retrieval System 15-48
- Source 1 6-16
- Source 1 and Source 2 parameters (F4 Amp) 6-29
- Source and Depth parameters (EnvCtl) 6-40
- Source parameter 15-24
- Space 15-48
- Spacing 15-42
- Special button functions 5-8
- Special-function soft buttons 6-47
- Spectral Multitap Delays 15-38
- SQUARE 16-40
- Src parameter
 - Sample page 14-3
- SRS 15-48
- Standard MIDI Files 13-73
- Startup 2-1, 13-5
- Startup File
 - creating 13-70
 - deleting banks in 13-70
- StatDlyLvl 15-42
- Static FX Mods 15-26

Stealer
 soft button on Utility page 11-12
 Steep bass tone control 16-32
 STEP Page
 Song Mode 12-61
 Stereo Analyze 15-49
 Stereo inputs with Pan 15-8
 Stereo keymaps 6-31
 Stereo parameter (Keymap) 6-24
 Stereo Simulation 15-48
 Storing Objects in Memory Banks 13-34
 Studio 9-2
 tutorial example 9-21
 Studio Editor 9-4
 AUXFX page 9-25
 INPUT page 9-23
 Studio Parameters 15-6
 Studio Structure 15-35
 Studios 9-16, 15-1
 Importing 15-27
 saving 15-20
 Substituting characters 3-7
 support for multiple removable disks 13-48
 Sustain not working 10-6
 Sustain Pedal parameter (Layer) 6-22
 Sustaining ASRs 6-43
 SW+DST 16-52
 SW+SHP 16-53
 Sweep 15-43
 SYNC M and SYNC S 16-55
 Synchronizing songs 12-20, 12-36
 Synthesis, General concepts 16-1
 SysEx dumps 6-47
 System
 tempo parameter 15-33
 System Exclusive 10-4
 System Exclusive dumps 11-19
 System Exclusive ID 10-6

T

Tap Dly 15-41
 Tap Lvl 15-41
 Tap n Pitch 15-38
 Tap n PtAmt 15-38
 Tap n Shapr 15-38
 Tap Pan 15-41
 Tap Tempo 15-34
 Tapn Bal 15-37
 Tapn Level 15-37
 Tempo
 as an FXMod Source 15-33
 Tempo control 9-22
 Tempo within an FX Preset 15-32
 Tempo-based control of KDFX 15-32
 Terminating SCSI devices 13-3
 Thin
 Song Mode TRACK Page 12-59
 Threshold 15-40
 Threshold (Thresh) 14-5
 Timbre Shift parameter (Keymap) 6-25
 Time 14-5
 Time Warp 14-28
 Toggle

 switch type 7-30
 TONEWL page (Program Editor) 6-48
 Top line of display 3-3
 Total size of all files 13-10
 Track mute 12-40
 TRACK Page
 Song Editor 12-48
 Track Status Indicators
 MIX page 12-40
 Transpose
 Master 11-2
 MIDI 10-2
 Song Mode TRACK Page 12-57
 Transpose parameter (Keymap) 6-23
 Transposing setups 7-2
 Treb Freq 15-38
 Treb Gain 15-39, 15-48
 Treble tone control 16-31
 Tremolo 15-46
 Trig Filt 15-43
 Trig parameter (Layer) 6-20
 Trigger 15-43
 Trigger parameter (ASR) 6-42
 Triggered Filter 15-43
 Triggering notes on startup 6-20
 Triggering sampler from keyboard 14-5
 Triggering samples from analog inputs 14-40
 Triggering samples from audio signals 14-4
 TRIM 14-16
 troubleshooting 13-82
 Truncate 14-21
 Tuning samples 14-38
 Tuning to other instruments 11-1
 Tutorial 9-21
 Twin peaks bandpass filter 16-23
 Two-parameter shaper 16-49
 Two-pole allpass filter 16-18
 Two-pole bandpass filter 16-21
 Two-pole bandpass filter, fixed width 16-22
 Two-pole highpass filter 16-15
 Two-pole lowpass filter 16-9
 Two-pole notch filter 16-19
 Two-pole notch filter, fixed width 16-20

U

Unipolar control sources 17-1
 Unipolar keystart 16-4
 Upper and lower amp 16-36
 User amplitude envelope 6-35
 Using the Digital Outputs 19-3
 Using the modes 4-3
 Using the Velocity Map Editor 18-3
 Using your K2600 in a SCSI System 13-4
 Utilities 11-12

V

Variable Architecture Synthesis 1-1, 1-3
 Variable tempos 15-33
 VAST 1-1, 1-3
 VAST program structure 6-2
 VAST programs 2-4

Index

Velocity crossover 14-38
Velocity Map Editor 18-3
Velocity maps 10-3, 10-6
Velocity sensitivity 11-2
Velocity stabs 20-1
Velocity tracking 6-16
Velocity tracking parameter (EnvCtl) 6-40
Velocity tracking parameter (F4 Amp) 6-29
Velocity tracking parameter (Keymap) 6-24
Velocity trigger parameters 6-44
Velocity triggers 6-44
Ventilation 2-2
Vib/Chor 15-46
VibChInOut 15-46
Vibrato/Chorus 15-46
View soft button 11-10
viewing directories 13-60
Viewing Selected Objects 13-26
Vocoder 11-5
Voice allocation 11-12
Voice channels 6-2
Voltage levels 2-2
Volume
 MIDI 10-10
Volume Adjust 14-22
Volume Level
 MIX page 12-39
Volume lock 10-11
Volume Ramp 14-27
VTRIG page (Program Editor) 6-44

W

Warmth 15-44
Warp FUN equations 17-11
WAVE Files 13-73
Waveform wraparound 16-49
Waveforms 16-38
 Sawtooth 16-40
 Sine 16-40
 Square 16-40
Waveforms with nonlinear functions 16-52
 + SHAPE MOD OSC 16-53
 AMP MOD OSC 16-53
 SHAPE MOD OSC 16-53
 SW+DIST 16-52
 SW+SHP 16-53
 x SHAPE MOD OSC 16-53
Wet Bal 15-37
Wet/Dry 15-47
Wet/Dry control 9-22
Wet/Dry parameter 15-35
WHEEL Page
 Setup Mode 7-29
Wind controller
 controlling K2600 with 11-14
Windows 13-80
Windows: configuring device drivers 13-80
“wires” 6-12
World-Wide Web 2-8
WRAP 16-49

X

x AMP 16-54
x GAIN 16-54
x SHAPE MOD OSC 16-53
XCouple parameter 15-36
Xcrs Fin 15-42
Xcurs Crs 15-42
XFADE 16-38
Xfer 15-39
XMIT page 10-1
Xover 15-45

Z

Zone status LEDs 7-4