

# Chapter 5

## Editing Conventions

### Introduction to Editing

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

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

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

### What's an Object?

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

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

<b>Pressure maps</b>	Factory preset or user-programmed curves that affect the K2600's response to, and MIDI transmission of, pressure (aftertouch) values.
<b>Intonation tables</b>	Factory preset or user-programmed tables that affect the intervals between the twelve notes of each octave.
<b>Master tables</b>	The values that are set for the global control parameters on the Master-mode page, as well as the settings for the parameters on the CHANNELS page in MIDI mode, and the programs currently assigned to each MIDI channel.
<b>Fader tables</b>	MIDI Controller assignments defined on the MIDI Faders pages.
<b>Name tables</b>	Contains a list of dependent objects needed by the other objects in a file at the time the file was saved.
<b>Macros</b>	List of disk files to be loaded into the K2600's memory at start-up time.

## Object Type and ID

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

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

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

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

Many parameters have objects as their values—the VelTouch parameter on the Master-mode page, for example. In this case, the object's ID appears in the value field along with the object's name. You can enter objects as values by entering their IDs with the alphanumeric pad. This is especially convenient for programs, since their ID numbers are the same as their MIDI program

change numbers (when you're using the Extended or Kurzweil Program Change format—see *Program Change Formats* on page 10-11).

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

## Saving and Naming

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

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

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

```
editProgram:Exit
```

```
Save changes to Righteous Piano?
```

```
Rename Cancel Yes No
```

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

```
editProgram:Rename <>KbdNaming:Off
```

```
Program Name: Righteous Piano
```

```
Delete Insert <<< >>> OK Cancel
```

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

The cursor underlines the currently selected character. Press the **<<<** or **>>>** soft buttons to move the cursor without changing characters. Press an alphanumeric button one or more times to enter a character above the cursor. The characters that correspond to the alphanumeric buttons are labeled under each button. If the character that appears is not the one you want,

press the button again. Press the +/- button on the alphanumeric pad to switch between upper and lower case characters.

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

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

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

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

: ; < = > ? @ A through Z

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

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

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

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

```
EditProgram:Save
```

```
Save Righteous Piano as: ID#200
```

```
Object          Rename Save Cancel
```

## ROM Objects

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

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

If you decide not to cancel or change the ID, and you press the **Replace** soft button, the K2600 will write your newly edited object over the existing ROM object. Actually, it only appears that



## Deleting Objects

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

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

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

## Dependent Objects

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

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

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

## Memory Banks

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

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

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

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

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

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

## Saving and Loading Files—Disk Mode

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

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

## Special Button Functions

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

## Editing Conventions

### Special Button Functions

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

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

**Table 5-1 Special Button Functions**