

Chapter 17

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 K2vx. 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 K2vx'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 *Reference Guide* for brief descriptions):

- | | |
|----|---------------|
| 1 | Equal |
| 2 | Classic Just |
| 3 | Just b/7th |
| 4 | Harmonic |
| 5 | Just Harmonic |
| 6 | Werkmeister |
| 7 | 1/5th Comma |
| 8 | 1/4th Comma |
| 9 | Indian Raga |
| 10 | Arabic |
| 11 | 1Bali/Java |
| 12 | 2Bali/Java |
| 13 | 3Bali/Java |
| 14 | Tibetan |
| 15 | CarlosAlpha |
| 16 | Pyth/aug4 |
| 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.n*". This isn't really an intonation table; it's simply a convenient place for the K2vx 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 non-standard intonation table, that you'll need to reset the Intonation parameter after you check the ROM object version list at "table" 18.

It's important to emphasize that the detuning values for each parameter affect the intervals in each octave independently. That is, notes that are an octave apart will remain perfectly tuned regardless of the detuning between notes within the octave. Consequently, editing intonation tables will not enable you to create tunings with more (or fewer) than 12 notes per octave. If you want to do this, use the Keymap Editor, assign each key to its own key range, then tune the samples in each key range.

When you're ready to edit an intonation table, select Master mode, then select the Intonation parameter. Use any data entry method to select the intonation table you want to edit. Check the

value of the intonation key parameter (IntonaKey). This sets the tonic, or reference note for the parameters on the Intonation Table Editor page. We'll explain this below.

Enter the Intonation Table Editor by pressing the Edit button. The Intonation Table Editor page displays a graphic representation of a C octave, with the low C always representing the tonic. The values for these parameters indicate the amount of detuning applied to each note relative to perfectly equal intonation. If you look at the values for the intonation table 1 Equal (as illustrated below), you'll see that the value of each parameter is 0. None of the notes is detuned, so the intervals between each note are exactly the same. Take a minute to look at the values for some of the other intonation tables, and listen to the effect on the tuning.

EditIntonationTable (cents)									
0		0		0		0		0	
0	0	0	0	0	0	0	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

Velocity maps affect the way the K2vx interprets attack velocity values. Velocity maps are assigned as the values for two different parameters: `VelocMap` on the `XMIT` page in MIDI mode, and `VelocityMap` on the `RECV` page in MIDI mode. (They also apply to the `VelTouch` parameter in Master mode, but since the `VelTouch` parameter doesn't affect the K2vx, you shouldn't change its default value.) There's only one list of velocity maps, and all three of these parameters use this list. Even so, certain maps are intended for use with the `VelTouch` parameter in the keyboard version of the K2vx, and others for use with the MIDI mode parameters. You can think of these as internal maps and MIDI maps. With the K2vx, you should use only the MIDI maps, and unless you have a problem with the K2vx's velocity response, or that of MIDI devices connected to the K2vx's MIDI Out port, you should leave both the Receive Velocity map and the Transmit Velocity map set to their default values.

Every attack velocity value generated by your MIDI controller is mapped through the velocity map assigned on the MIDI mode `RECV` page before being passed to the sound engine. If you're using the Local Keyboard Channel feature, attack velocity values received at the K2vx'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 `XMIT` page in MIDI mode), and to the MIDI Out port.

The `VelocMap` parameter on the `XMIT` page in MIDI mode affects only the velocity values the K2vx 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 K2vx. If, for example, you're overdriving a DX7 that you have slaved to the K2vx, try the map 6 Hard2.

The `VelocityMap` parameter on the `RECV` page in MIDI mode affects how the K2vx 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 `XMIT` page. You might want to adjust this parameter, for example, to boost or cut the volume response of the K2vx during playback in Song mode, or when playing a sequence from an external sequencer.

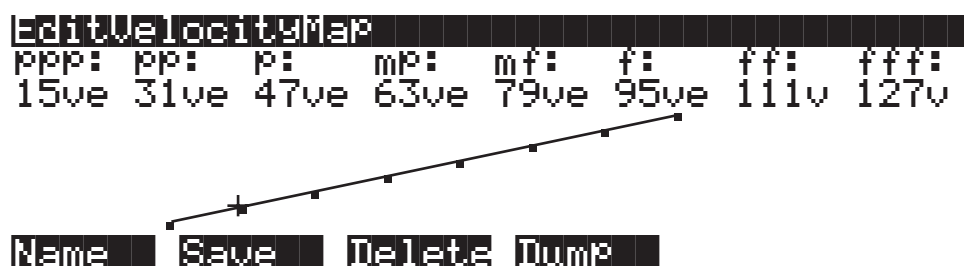
Here's a list of the velocity maps stored in ROM (the factory presets). Keep in mind, however, that you should use only the linear MIDI maps; the internal maps are meant for the K2vx.

- 1 Linear (the default on the `XMIT` and `RECV` 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 K2vx's sound engine is routed through the map assigned to the `VelocityMap` parameter on the `RECV` 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` (`XMIT` page, MIDI mode), or `VelocityMap` (`RECV` page, MIDI mode). Use any data entry method to select the map you want to edit, then press Edit.



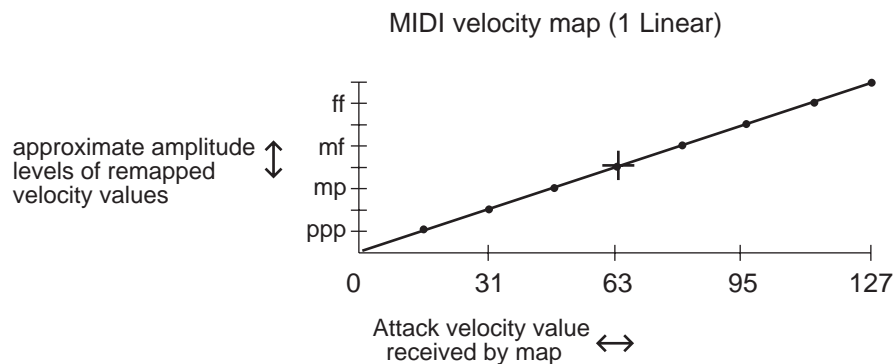
The top line of the display gives you the usual reminder of your location. The bottom line labels the soft buttons, which execute the normal librarian functions of naming, saving, or deleting velocity maps, or dumping them via SysEx. The values you see in this diagram are the settings for the velocity 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 K2vx's dynamic range.

The values for these parameters are expressed in “vels,” which correspond to the standard MIDI attack velocity values of 0—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 K2vx will play notes at full volume only when the attack velocity values for those notes are 127. Notes with attack velocities from 0 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.

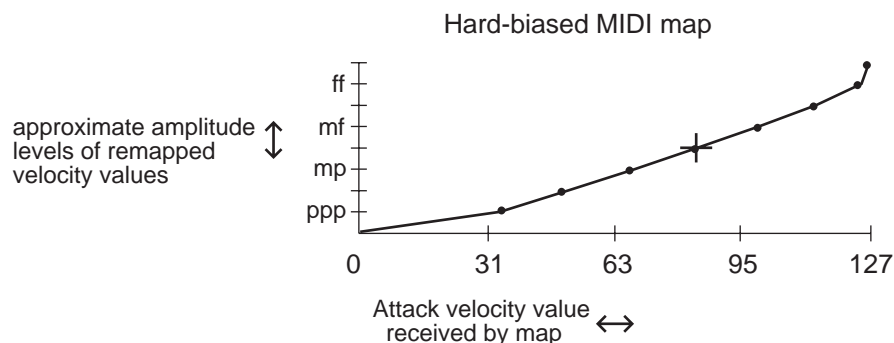
The curve that represents the velocity map is shown in the center of the page. Think of this curve as a plot on a graph, as shown in the diagram below. The horizontal axis gives the attack velocity values received by the velocity map. The vertical axis gives the approximate dynamic level at which the K2vx 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 K2vx'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.



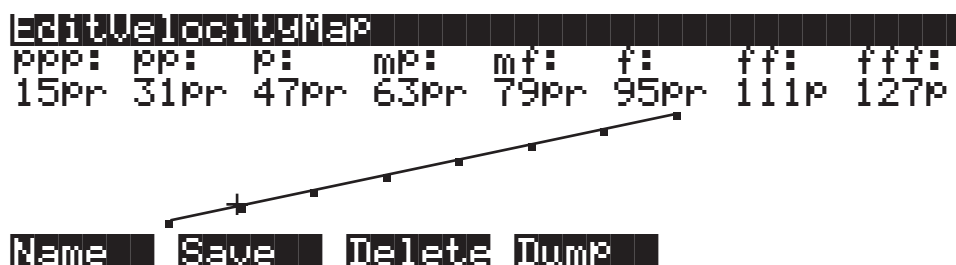
The Pressure Map Editor

Pressure maps function in much the same way as velocity maps, but they affect the K2vx'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 K2vx'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 XMIT page in MIDI mode, and the PressureMap parameter on the RECV 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 K2vx.

- 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 use any data entry method to 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 levels expressed by the parameters. Values in between those indicated by the parameters will be remapped as indicated by the curve.