Chapter 5
The Environment

The Situation

This chapter will help you to understand the basic concept of the Environment, and make full use of its options.

Your setup probably consists of a number of different MIDI devices. The purpose of the Environment is to help you manage your setup flexibly and efficiently, down to the smallest detail (e.g. the pre-delay of a specific sound parameter in a synthesizer).
Logic's Environment gives you a number of objects for sophisticated MIDI processing. These include: arpeggiators, MIDI delays, transformers, note mappers, chord memorizers, and even Touch Tracks™ for playing entire sequences at the press of a single key. Of course, you need use only the parts that you really want, and forget the rest.

The Idea

The Environment refers to Logic's "virtual" environment inside your computer. It was developed to allow you complete control over your MIDI setup from within Logic.

Imagine that the Environment window gives you a virtual view of your MIDI studio, including port objects which represent the MIDI inputs and outputs of your MIDI interface, instruments which are virtual representations of each device in your MIDI setup, and various objects for modifying the MIDI signal flow.

In order for this to work properly, you have to connect the different objects in the Environment via virtual MIDI cables. This allows you to control the overall MIDI signal flow.

In the Environment, you can also create virtual faders which generate MIDI events when you move them. The movements of these fader objects can be remote-controlled by other MIDI events, and recorded by the sequencer. In addition, there are specialized objects which can split a MIDI signal into different channels, make pre-programmed alterations, or even re-route the signal path.

Layers

As you can probably imagine, the Environment can quickly fill up with a large number of objects. To keep things organized, you can assign the objects to different display levels, which are referred to as “layers”. Think of these layers as being different, partial views of the overall Environment. Naturally, you can easily connect objects between different layers.

The Concept of the Environment

Flexibility

The Environment allows you to control your MIDI studio and Logic’s MIDI signal flow. Virtual cables give you complete control over the MIDI signal path, from the input ports to the output ports. This gives you a high degree of flexibility when connecting Logic’s individual “effect modules”.

Adaptability

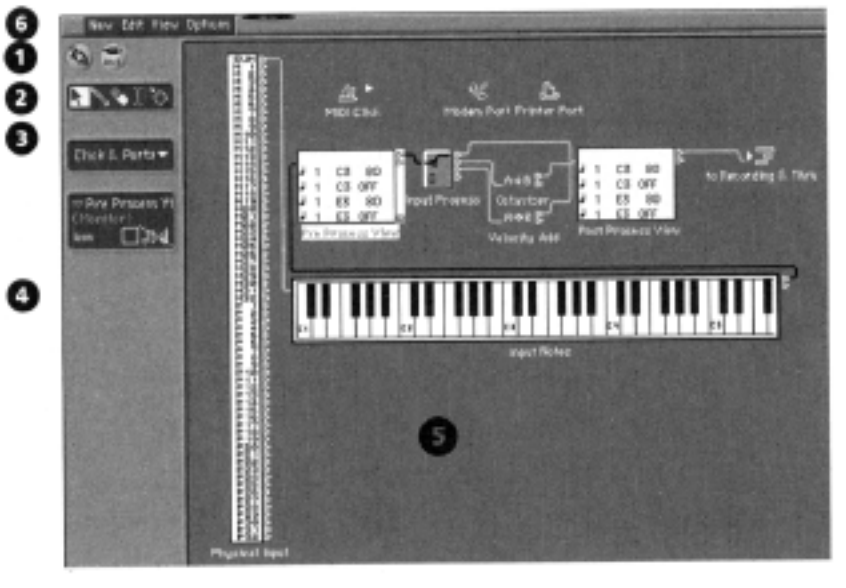
Logic addresses the MIDI ports of your interface directly via virtual instruments which you create in the Environment. These virtual instruments act like junctions: events can easily be addressed to them from the sequencer and then sent from them to a MIDI or internal (software) sound generator. Instruments include parameters which you can use to select sounds, transpositions, keyboard splits, volume, pan, MIDI delay, etc. Instruments can therefore also be thought of as virtual copies of individual sounds in your sound generator. In addition, Mapped instruments provide note assignment tables (drum maps) which can be used to adapt your preferred playing and

programming habits to the different key assignments of your ound sources. Finally, you can create virtual faders and knobs or controlling effect devices or automated mixing desks. The point is, the Environment allows you to adapt Logic to manage virtually all aspects of your MIDI setup.

5.1 Environment Overview

Opening the Environment Window

To open an Environment window, select **Windows > Open Environment** (default key command: ⌘8). Double-clicking on an nstrument in the track list of the Arrange window also opens an Environment window, to the layer containing that track instrunent. When the Environment window opens, the track Instrunent you double-clicked will be highlighted:



1. Buttons
2. Tool box
3. Layer box
4. Object parameter box
5. Layer contents (the default “Click & Ports” layer)
6. Local menus

Layers

Layers are “display levels” of the Environment window. They allow you to configure the Environment’s display so that you can view and work with specific groups of related objects (e.g. audio instruments), rather than having to deal with all Environment objects at the same time. The name of the current layer is displayed in the layer box.

The distribution of objects on different layers has no affect on their function-it’s just a system of organization.

There are some exceptions: objects in the “global objects” layer also appear on all other layers; and the “All Objects” layer shows all the Environment objects in a list, However, you can ignore both of these layers at first. They are not needed very often, even in very complex Environments.

Switching Layers

Click-holding the layer box opens the layer flip menu, which you use to switch layers.

Select **Options > Go to previous layer** to switch back to the last active layer. This allows you to toggle between two layers.

Remember you can assign key commands to this and many other Environment menu functions for faster navigation and greater ease of use.

Creating Layers

Selecting "**Create!**" from the layer flip menu creates a new, empty layer titled *"(unnamed)"* at the end of the list. You can add a new layer above the current position in the list by selecting **Options >Layer > Insert.**

Naming Layers

Double-clicking on the layer box opens the text field for entering a name for the current layer.

Deleting Layers

You can remove the current layer from the list by selecting **Options > Layer > Delete.** Because all the objects contained in the layer are deleted too, an alert box appears, to warn against deleting objects unwittingly. You must click "Delete" to complete the operation. "Cancel" or ***Escape*** cancels the operation.

Specialized Layers

The position and existence of the first two layers are protected:

All Objects

The top layer in the flip menu always displays all the objects in the Environment. The objects in this layer are normally shown in the form of a list. (You can change this in the Environment's **View** menu, but doing so is not recommended.)

If you turn off the **File > Preferences > Display > *Allow All Objects' Layer in Environment,*** the "All Objects" layer will be removed from the layer flip menu.

The function **Options > Goto Layer of Object** allows you to switch to the layer of a selected object.

Global Objects

In this second layer from the top, you can place objects which you want to be visible in all layers (e.g. the output ports). These will subsequently appear at the same position, in all layers. However, you should place as few objects as possible on this layer, due to the on-screen clutter that global objects can create.

You ***may want to temporarily move some objects to the Global Objects layer while you are building multi-layer setups then move them to one of the other layers when you are finished.***

Hiding/showing parameter fields

As in other windows, you can hide the column with the buttons, toolbox, layer menu and object parameter box by selecting **View > Parameters.** This will create more on-screen room for the Environment workspace display.

Hiding/showing the cables

By choosing View > **Cables** *(Hide/Show Cables), you* can toggle the view of the cables. This also hides/shows the positioning bar to the right of the individual objects (used for moving and sizing some Environment objects).

Protecting cabling and object positions

You can prevent the position, size and cable connections of all objects from being accidentally altered, by selecting **View > Protect Cabling/Positions.**

Background

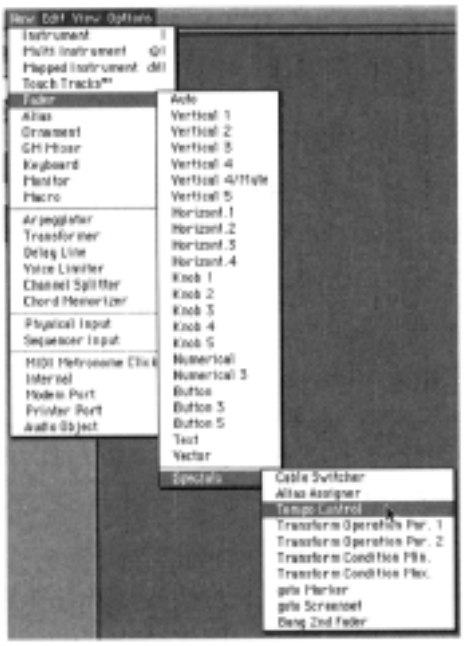
If the cabling and object positions are protected, and the cables are hidden, the background color changes. This usually looks better for virtual mixing desks and fader setups.

You can also use Ornaments and Macros to clean-up and improve an Environment control panel's appearance.

List Display

Selecting **View > by Text** switches the graphic display of the layers to a list display. The cables are not shown in the list display. This type of display is most useful in the “All Objects” layer (see section *Spezialized Layers* on page 5-5), and for importing Environments from other songs (see section *Environment Exchange* on page 5-92).

Objects



Creating Objects

When you select an object type from the New menu, an object of this type is created in the current layer (for more details see section ***Object Reference*** on page 5-24). Clicking with the pencil on the background creates a new “Standard Instrument”.

You can copy existing objects by dragging an object while holding down ***Alt*** (see also the section ***Moving Objects*** on page 5-8). Cabling is preserved; so groups of objects copied in this way will be connected in the same way as the originals. (When you copy a single object, only its output cables are preserved.)

Deleting Objects

Click with the eraser, or press ***Backspace*** to delete all selected objects.

Adjusting the Size of Objects

You can adjust the size of fader, keyboard and monitor objects by grabbing and dragging the bottom right corner (just as with windows). Changing the size with multiple objects selected (by “rubber-banding” or clicking while holding down ⌂) will alter

each object's size while preserving their sizes relative to each other.

Selecting **Options > Clean up > Size by Default** (Reset Size), sets the size each selected object to its default value.

The key commands *Object Width -1 Pixel or ...+1 Pixel* and *Object Height -1 Pixel* or *...+1 Pixel* decrease or increase the width or height of all selected objects by 1 pixel.

Moving Objects

You can move objects by grabbing their icon or name, and dragging them to the desired position. The surface of keyboard and fader objects is used for their operation. You must therefore either grab the name (if available), or the positioning bar to the right of the object.

If you hold down , you can also grab keyboard or fader objects by their surface. Remember to first deselect any other selected objects by clicking in the background, to avoid moving them as well.

Grid

Select View > Snap Positions to align the objects to an invisible grid. It's a good idea to leave snap positions switched on. You only need to switch it off if you want to manually move an object by a few pixels.

You can use the key commands Object move *left/right/up/down* to move all selected objects one pixel in the relevant direction, even when the grid is switched on.

Aligning Objects . . .

...in a row or column

To align several selected objects horizontally or vertically, select the function **Options > Clean up > Align Objects.** The top left object stays where it is. The position of the next object

determines whether the objects are aligned in a column or a row. If it is to the right of the top left object, all the objects are aligned horizontally. If it is below the top left object, all the objects are aligned vertically.

...to grid positions

If you choose **Options > Clean up > Positions by Grid, all** selected objects are aligned to the invisible grid.

Moving Objects between Layers

...with the layer selection box

To alter the layer assignment of a group of Environment objects, first select the objects, and then select the desired layer, while holding down ***Alt***. This moves the selected objects to the layer you've chosen.

...with a second Environment window

Another way of moving objects to a different layer is to open a second Environment window. Select the target layer and drag the relevant objects from one window to the other. You can also use this method to copy objects between layers (by holding ⌥).

...via the clipboard

Another option is to move objects via the clipboard **(Edit** menu or the usual key commands):

Select the objects that you want to move and select **Edit > Cut** (default key command ⌘X).
Switch to the destination layer.
Make sure that no objects are selected (by clicking in the background).
Select **Edit > Paste** (default key command ⌘V).

Replacing Objects

If any objects are already selected when you go to add objects to the current layer using **Edit > Paste** a dialog box appears asking “Replace current selection? No/Replace”. If you press ↵ or click Replace, the selected objects will be replaced by the objects in the clipboard. The existing cabling remains intact.

Assigning an Object to the Selected Track

You can assign any Environment object to the selected track in the Arrange window simply by clicking the object with the MIDI Thru tool.

You can also assign Environment objects to the selected track using the Arrange window’s Instrument List. Simply click and hold on the track name until the list appears, then scroll to the desired Environment object (see section ***Icon/Display Filter for the Instrument List*** on page ***5-23).***

Remember: the MIDI Thru function of the sequencer diverts all incoming MIDI events to the Environment object assigned to the selected track.

Special Selection Commands

Selecting All Used Instruments

With **Edit > Select Used Instruments, you** can select all objects which are assigned to tracks in the Arrange window or are connected to such objects by any series of cables.

Selecting All Unused Instruments

The **Edit > Select Unused Instruments** function allows you to select all the objects that are neither being used in the Arrange window, nor are connected to any such objects by a series of cables.

Selecting Cable Destination Objects

Edit > Select Cable Destination selects the destination object of a selected cable connection. This is particularly useful in two cases:

- The destination object is in a different layer. You can use the function to select and display the destination object in its layer;
- From the list display **View > by Text) you** can locate the destination object of a source object, because selecting the source object also selects its cable connections.

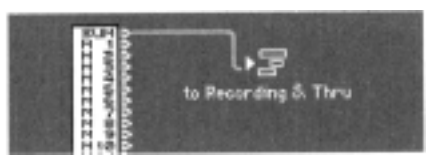
This function allows you to follow the cabling from one serially-cabled object to the next. If several objects are connected, or several cable connections are selected, the path of the top cable is followed.

Toggling your selection

Just as in the Arrange and editor windows, you can change the status of any selected objects in the current layer by choosing **Edit > Toggle Selection.**

5.2 The MIDI Signal Path

Before any MIDI events received at your computer's MIDI inputs can be recorded by the sequencer, there must be a connection between two Environment objects: the physical input and the sequencer input.



In the sequencer, the events are always directed to the selected track, where they can be recorded. The events played by the track are mixed with any incoming events and sent to the Environment object assigned to the track in the Arrange window's

Track List. From here, the events are directed to a MIDI output (see the section ***Direct Output Assignment*** below).

You can insert objects into the signal path between the physical input and the sequencer input as well as between the instrument and the MIDI Out port to accomplish other Environment processing (see section Cabling on page 5-13).

Direct Output Assignment

You can create a direct output connection from any instrument object to a physical MIDI output by setting the desired output in the ***Port*** parameter line underneath the channel parameter.

If you click and hold in this line, a flip menu will open that lists 63 individual ports and one general setting (0) for each of the three MIDI output connections as they have been set up in the communication preferences **(Options > Settings > MIDI-Communication).** Depending on the type of your computer (or rather its age...) these can be MIDI interfaces connected to the universal serial bus (USB) of your computer, to the modem port or to the printer port. Please note: Currently only USB connections to interfaces belonging to Emagic's Unitor family are supported. For more details please read section MIDI Interface Communication on page17-11.

If you have a MIDI-interface with more than one output port (or even several interfaces which can be stacked, like products belonging to the Unitor family). you can set up a direct connection to one of the individual ports (1 to 63) of that or those interface/s (for example ***"1-Modem 5"*** or ***"1-USB 23").*** If you select output 0 (zero) combined with a particular "connection" setting (e.g. ***"1-Printer 0"),*** the MIDI data of that instrument will be sent to ***all*** output ports of that particular connection.

Note: If you click in the right part of this line, the appearing flip menu will contain ***all*** available port settings (193!), even those which are not assigned to an output connection (labeled ***"n/a").*** If you click on the left side, there are only four choices: Off and the same output number of each connection as it was set here

before (for example Off, ***1-Modem 7, 2-Printer 7*** and ***3-n/a 7***). This is useful if you want to switch off the direct output connection, or to switch between identical port numbers of different connections.

For more details about the MIDI connection settings and USB support please read section ***MIDI Interface Communication*** on page 17-11, and also look for ***"ReadMe"*** files and additional update info files on your program discs.

Any object with a directly assigned output is marked by a shaded triangle on its right side. When there is no direct assignment the triangle is hollow.

Breaking the Direct Output Assignment

To break the direct output assignment, set the the Port parameter to ***off*** which is the top entry in the flip menu there.

Cabling

The cabling between Environment objects gives you control over the entire MIDI signal path. A cable is normally shown as a gray line between a source and destination object.

Objects always have their input on the left side and their output on the right side. The output of an object is marked by a small triangle pointing to the right.

If you grab the triangle, the mouse pointer turns into a patch cord representing a cable connection coming from the output. Move the mouse to the destination object to connect it with a cable. Once you have selected the destination object (this happens automatically when you touch it), release the mouse button.

If the source object has already been directly assigned to an output, a dialog box will appear telling you this, and asking whether you want to replace the direct assignment. You have three options:

Cancel the connection is not made, and the direct output assignment remains intact.

No: your cabling is carried out, but the direct output assignment remains intact. This means that the object is connected in two places-one via the visible cable and one via the Direct Port assignment.

Remove (or ***Backspace***): your cabling is carried out, and the direct output assignment is removed. (This is the default selection because you generally won't wish to have an object connected to two different destinations)

Cabling between Layers

If the destination object of the cable is on a different layer, you can connect it using the Instrument List. To do this, hold  as you grab the output triangle and scroll to the desired destination object when the Instrument List appears. (For more on the Instrument List see section ***Icon/Display Filter for the Instrument List*** on page 5 ***-23).***

You can also open a second Environment window showing the destination layer, and connect the objects graphically between the windows.

A cable connection to another layer looks like this:

Multiple Cabling

Multiple inputs

There is no limit to the number of cables that you can "plug" into a target object. All the MIDI signals are then mixed at its input.

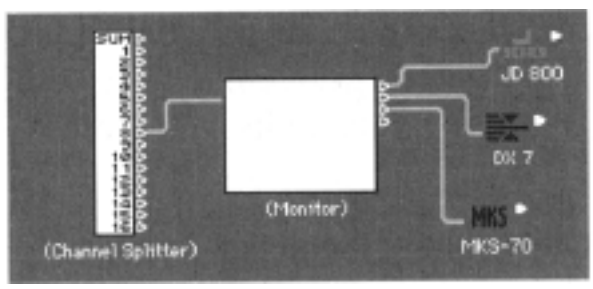
Multiple outputs

Once an output from an object is used (i.e. cabled to another object), another output triangle automatically appears. You can thus cable outputs from the object to as many other objects as you want.

Some objects have special outputs and are therefore exceptions to this rule.

Special Outputs

Some objects have several functionally ***different*** outputs, and in these cases, each output is only available once (e.g. channel splitter, cable switcher or physical input). If you want to route the signal from these outputs to several destinations you must first create a monitor **New > Monitor),** plug the cable into it, then connect it to as many other destinations as you like. The monitor object also allows you to view the MIDI events flowing through it.)



To select a common destination for several existing cables, first select the cables. (If the cables are already leading to a common destination, the simplest way of doing this is to select the destination object.) Next grab one of the cables and plug it into the new destination object. A dialog box will appear asking “Do you want to connect all selected cables with new Destination? No/Connect”. Click Connect, or press ↵.

Cabling Serially

To cable a group of objects serially, first select all the objects then select **Options > Cable serially.** The objects will be cabled in series, starting with the top-left object. This is handy for quickly connecting groups of faders in a mixing console, for example.

Cabling in Parallel

To cable a group of the same type of object (e.g. transformers, faders, etc.) to the same destination, first cable one of the objects to the destination. Next select that object and choose **Edit > Copy.** Finally, select the rest of the objects to be cabled to the same destination and choose **Options > Apply Buffer Template to > Cable(s).** (You can copy more complex cabling configurations this way, too. Just ensure that the type(s) of objects in the group you select to copy, matches the type(s) of objects in the group you select to apply the buffer template to.)

Deleting Cables

To remove a cable connection, click the cable with the eraser or select the cable and press ***Backspace***. You can also remove a cable by grabbing it and plugging it into the input of its own source object.

You can use **Edit > Clear Cables only** to remove all selected cables without also clearing any objects that happen to be selected. This is handy if you want to clear all the cables leading to or from one of more objects, because selecting objects also selects all their cable connections.

Cable color

Cables are given the same color as that of the object they came from (assuming, of course, that your monitor is set up to be used in color). You can switch off the cable coloring, and render them all gray with the **View > Colored Cables** function.

5.3 Environment Objects

Object Overview

Standard Instrument

This is the simplest track playback object. It provides a direct connection to a single MIDI port and channel, which can be set in its Parameter Box. You can also set a number of other playback parameters, including program bank & number, initial volume & pan, transpose, velocity scaling and MIDI delay (for details see page 3-12 and page 5-24).

Multi-Instrument

A multi-instrument is like a collection of 16 standard instruments (see above) in one package. There is one of these "sub-channels" for each of the 16 MIDI channels as indicated by the numbered squares on the multi-instrument icon. (Clicking one of these numbered squares selects the corresponding sub-instrument and displays its parameters in the Parameter Box.) All sub-channels of a multi-instrument use the same output port.

Multi-instruments have 1.5 name-banks each of which will hold 128 preset names (page 5-40). (Double-click the multi-instrument, or any of the sub-instrument numbers to open the name-banks window.) If the multi-instrument's program parameter (page 5-25) is checked, selecting names in the names-bank will send MIDI program change messages to the corresponding port and channel.

You will generally use multi-instruments to address multi-timbral synths or samplers in your studio. A multi-timbral synth is one that can receive on several MIDI channels at once - playing back a different sound on each channel. Since most newer MIDI devices are multi-timbral, the multi instrument

will likely be the most commonly used instrument in your Environment.

Mapped Instrument

This instrument is particularly useful for drum instruments or any drum-mode MIDI device. (A drum-mode device is one which assigns different sounds to different MIDI notes on the same MIDI channel.) Any input note may be routed to a different output note and assigned its own velocity setting, notation, name and output cable (see page 5-34). You can also protect Mapped Instruments from being accidentally transposed.

Faders

Faders are used to send MIDI events by clicking with the mouse on the screen. Faders come in different forms (e.g. knobs, sliders, numericals, and buttons). Faders will also respond to incoming MIDI events which makes them ideal for automated mixing (see page 5-78).

GM Mixer

The GM Mixer is a collection of 16 fader “modules” configured to emulate a virtual mixing desk for 16 MIDI channels. Each module has controls for volume, mute, preset, bank and four assignable knobs (one of which is typically used for pan). There are optional controls for standard XG and GS effects.

The GM Mixer comes in three styles and like standard instruments, is able to provide a direct connection to an output port which can be set in its Parameter Box.

You will most likely use Logic’s Adaptive Track Mixer for most of your MIDI & audio mixing but the GM Mixer can be handy when you want a dedicated mixer for a specific, multi-channel MIDI device.

Keyboard

The keyboard object is used for creating notes with the mouse. It will also display all notes passing through it. In this sense you can think of it as a real-time, note-on monitor (see page 5-40).

Monitor

The monitor object displays all MIDI events passing through it. and retains the last 32 such events. You will need to resize it to see more than 5 events at a time (see page 5-41).

Channel Splitter

The channel splitter routes MIDI events according to their channel. It has one output for each of the 16 MIDI channels and an additional “SUM” output. If a particular channel output is not connected to anything then events for that channel will appear at the SUM output (see page 5-41).

Transformer

The transformer is used to select and transform MIDI events passing through it. Its select-conditions allow it to act as a MIDI filter. Using its transform-operations along with the select-conditions allows you to transform some MIDI events while letting others pass through unchanged. Using what are known as "meta-events" (page 5-46) you can control many of the transformer’s parameters in real time (see page 5-42).

Voice Limiter

The voice limiter limits the number of MIDI notes (1 to 32) that can be held simultaneously. It does this by “note stealing" newly arriving notes will cause some currently held notes to be turned off once the voice limit is reached. The choice of what notes are turned off is controlled by the priority parameter, which can be set to last, top or bottom (see page 5-51).

Chord Memorizer

The chord memorizer maps single notes into chords containing up to 12 notes. It operates on a "pitch-class" basis, meaning that all notes with the same name (e.g. all C's, all F#s, etc.) will be mapped to the same chord but transposed relative to the incoming note's octave. The chord memorizer can also be used as a "scale filter" by mapping certain (non-scale) notes to nothing (see page 5-51).

Arpeggiator

The arpeggiator turns chords into arpeggios. It plays the currently held notes individually in a selectable pattern (up, down, random, etc.) and at a selectable speed between whole-notes and 768th-notes. It has a number of useful parameters including note-length, repeat on/off and octave doubling. All of its parameters can be changed in real time using MIDI controller messages (see page 5-47).

Logic's transport must be running for the arpeggiator to work.

Delay Line

The delay line echoes MIDI events passing through it at intervals ranging from one tick to 256 whole notes. There are parameters for the number of echoes, as well as their transposition and velocity offset. The echoes of each incoming event will be cycled through each of the cabled outputs of the delay line (see page 5-50).

Logic's transport must be running for the delay line to work.

Touch Tracks

The touch tracks object allows you to assign any selection of sequences and folders to a single MIDI note. You can then use your MIDI keyboard to trigger playback of these selections.

You can not use touch tracks to play audio regions.

Logic's transport must be running for the touch tracks to work.

Physical Input

The physical input represents the inputs of your computer's MIDI interface, and is normally connected to the sequencer input (see page 5-58).

Sequencer Input

The sequencer input represents the input to Logic's sequencer (i.e. to Arrange window tracks). MIDI events arriving here are routed to the selected track, where they may be recorded. They will also be sent to whatever Environment object is assigned to that track (see page 5-58).

MIDI Metronome Click

The MIDI metronome click generates MIDI note events at Logic's bar, beat and grid divisions. The pitch and velocity for each division can be set separately and the divisions can be turned on or off individually. The metronome itself is toggled on and off in Logic's transport (see page 5-59).

Logic's transport must be running for the MIDI metronome click to work.

Internal Objects

Internal objects allow MIDI to be routed to sound cards and software synthesizers inside your computer. The selection of devices (e.g. Quicktime, Wavemaker, Retro AS-l, etc.) depends on the hardware and software installed in your computer (see page 5-60).

Modem/Printer Port

Represents the physical outputs of the MIDI interfaces, which are connected to the port of the same name on your computer. If you are using an interface with several ports, you can create

one of these **objects** for each individual port, up to a maximum of 99 (see page 5-62).

Audio Objects

Audio objects are the only Environment objects that are not part of the MIDI signal flow. They allow MIDI controller messages to control aspects of Logic's audio playback such as volume, pan, muting and insert parameters. They are also used to create audio tracks in the Arrange window-audio tracks are just tracks with audio objects assigned to them.

Ornament Objects

The ornament object is not part of the MIDI signal path. It is simply a display object which allows you to create backgrounds behind other Environment objects. Its purpose is to allow you to create more attractive and useful control panels.

Macro Objects

Macros are not actually Environment objects at all. They are collections of Environment objects together with all their cables. Macros behave like single Environment objects. You can turn any reasonably sized (up to about 100) collection of Environment objects (including other macros) into a macro, then move it around, duplicate it and connect cables to and from it like any other object. You can also convert macros back into regular objects and cables for editing.

Aliases

As with sequences in the Arrange window, aliases of Environment objects represent the original object, but also have some properties of their own. Aliases are most often used for faders in which case they have their own value and cabling but send the same MIDI events as the original fader. In the case of text faders, aliases provide a significant savings in memory.

Object Parameters

Each Environment object has several parameters that control its operation. These parameters can be viewed and changed in the object's Parameter Box. The Parameter Box appears in the Arrange window when you select a track assigned to the object. It also appears in the Environment window when you select the object. These are the *same* parameters-changes in one location will be reflected in the other.

Opening/Closing the Object Parameter Box

By clicking the triangle in the top left, you can hide all of the parameters, except name and type. This reduces the box to its minimum vertical size.

Common Parameters

The following parameters are the same for all object types:

Name

The name of the object is shown next to the triangle, and can be selected for editing by clicking on it. You can also edit the name by clicking on the object with the text tool.

Object Type

The object type is shown in brackets, and cannot be edited.

Icon/Display Filter for the Instrument List

Click-holding on the icon in the Parameter Box opens a flip menu, where you can choose an icon to represent the object in the Environment and in the Arrange window's track list.

Placing a check in the box next to the icon means that the object will appear on the Instrument List, and can therefore be assigned to a track by using this menu. (Objects not on the track list can still be assigned to tracks by dragging them from the Environment window to the track list or by using the Environment's MIDI Thru tool.)

If the box is not checked, the object will not appear on the Instrument List, and a diagonal line will appear through the icon. This will not hide the object or its icon from view in the Environment.

You should only check the icon box if the object will be used as a track instrument, or you need to connect cables to its outputs using the instrument selection menu (see page 5-14).

5.4 Object Reference

Standard Instrument

To create a new standard instrument, select **New > Instrument,** or click in the background with the pencil tool. Instruments serve as the interface between Logic's tracks and your MIDI sound sources.

> Please note: In older versions of Logic the name of this object was just "Instrument". The term "Standard Instrument" is now used both in the program itself and in this book to clearly point out the difference between the "Standard Instrument", "Multi Instrument" and "Mapped Instrument" objects.

The Standard Instrument Object Parameter box will be familiar to you from the Arrange window. Here are the details of the individual parameters.

Cha — Use the *Cha* parameter to set the MIDI channel for the instrument's output. All events at the output of the standard instrument will be sent with the channel that is set here. Exception: If you set this parameter to *All*, all events will be sent with their original channel settings as they appear in the Event List window.

Port — Below the channel parameter, a direct connection to one of the MIDI output ports can be set. Remember that you can also directly cable the standard instrument to other Environment

objects for MIDI processing. If no MIDI output port is defined in the object parameter box for that instrument, the ***Port*** parameter will display "off". For details please read section ***MIDI Interface Communication*** on page 17-11 and section ***Direct Output Assignment*** on page 5 -12.

Prg, Vol and Pan The ***Prg, Vol and Pan*** parameters transmit program changes, volume controller (#7) and pan controller (#10). If there is no check in a box, you can edit its value, but it will not be sent until you check the box (by clicking on it). If the box is already checked, any value alterations will be sent immediately, and the values will also be sent whenever the track is selected.

To the left of the program number, directly to the right of the box, is the parameter for bank select. If your MIDI sound sources recognize Bank Select messages (check in their manuals), you will be able to switch between sound banks. If your sound sources respond to the standard Bank Select message (Controller #32), you will be able to use this parameter directly. If not, please read the section ***Defining Your Own Bank Select Commands*** on page 5-32.

Transpose The ***Transpose*** parameter allows you to define the number of semitones by which all the note events will be transposed as they are output. Negative values transpose downwards.

Velocity The ***Velocity*** parameter allows you to increase or decrease the “note on” velocities of all note events by an amount adjustable between -99 and 99.

Lim The two note values of the ***Lim*** parameter define a pitch range. All notes outside this range will be ignored by the instrument when it plays a sequence.

VLim The two values of the *VLim* parameter define a velocity range. All notes whose velocity is outside this range will not be played by the instrument.

Delay The *Delay* parameter causes all MIDI events to be sent early or late by from -99 to 99 ticks. This allows you to adjust for any differences in reaction time between your various MIDI devices. For delay effects use the sequence parameter of the same name, as this allows longer delay times.

No Seq Trp If the box next to the *No Sequence Transpose* parameter is checked, all sequences on any tracks played by this instrument are protected from transposition. In other words, the "transpose" sequence parameter is ignored. This is very useful for instruments assigned to drum or other multi-timbral samples, as transposition will cause the sounds rather than the pitches to change for these instruments.

No Reset If the box next to the No *Reset* parameter is checked, no reset messages will be sent to this instrument. This can be useful if controllers are being used for non-musical purposes, for example when an Instrument is used for mixer automation. The **Preferences > Reset Messages** window controls what reset messages are normally sent, but these messages are not sent to "No *Reset*" instruments.

Default Score Style The ***Default Score Style*** parameter is located at the bottom of the Parameter Box. It displays *Auto Style* by default but can be changed to any of the available score styles. Whenever a sequence is created on one of the instrument's tracks it will be assigned the score style

displayed here. In the case of ***Auto Style***, Logic will pick an appropriate style based on the pitch range of the notes in the sequence.

Don't forget: you can change a sequence's Score Style at any time in the Display Parameter box in the Score window.

You can change the Score Styles of all sequences on an instrument's tracks by holding down ***Alt***, when selecting the instrument's default Score Style.

Multi-Instruments

To create a new multi-instrument, choose **New > Multi Instrument.**

A multi-instrument is intended for use with multi-channel MIDI sound devices. It combines 16 separate sub-instruments (one for each MIDI channel) with 15 name-banks, and a number of common bank-select options.

A multi-instrument consists of 16 sub-channels, each of which has a complete set of instrument parameters. The name, MIDI channel and output of each sub-channel is determined by the multi-instrument. All sub-channels share the 15 banks of program names, and use the same format for their bank select message.

Multi-Instrument Parameter Box

Once you have created a multi-instrument, you will see the multi-instrument parameter box shown here.

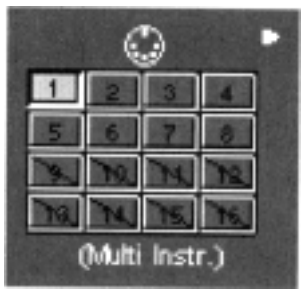
You should already be familiar with the meaning and operation of the individual parameters (if not refer to the description on page 5-34). The settings made here apply to all sub-channels and except for the output port setting, should be left at their neutral, default setting. Set the output port in the line below the ***"Cha"*** parameter line (parameter ***Port).*** For more information on this subject, please read the section ***Direct Output Assign-***

ment on page 5-12. However, keep in mind that the direct output assignment can also be replaced by manual cabling to other environment objects.

Activating and Selecting Sub-channels

To select a sub-channel for a specific MIDI channel number, click the appropriately numbered button on the multi-instrument object. The first time you click it, the sub-channel is activated making it available from the Instrument List. To avoid menu clutter, you should only activate as many sub-channels as you need. As with any other Environment objects, you can remove sub-channels from the Instrument List, by unchecking the icon box. The button for that sub-channel will then be shown with a diagonal line through it. If you want to select the entire multi-instrument rather than a particular sub-channel, click on the top edge of the multi-instrument or next to the icon.

In this diagram, sub-channel 1 is selected, sub-channels 1-8 are activated and 9-16. have been removed from the instrument list.



The Sub-channel Parameter Box

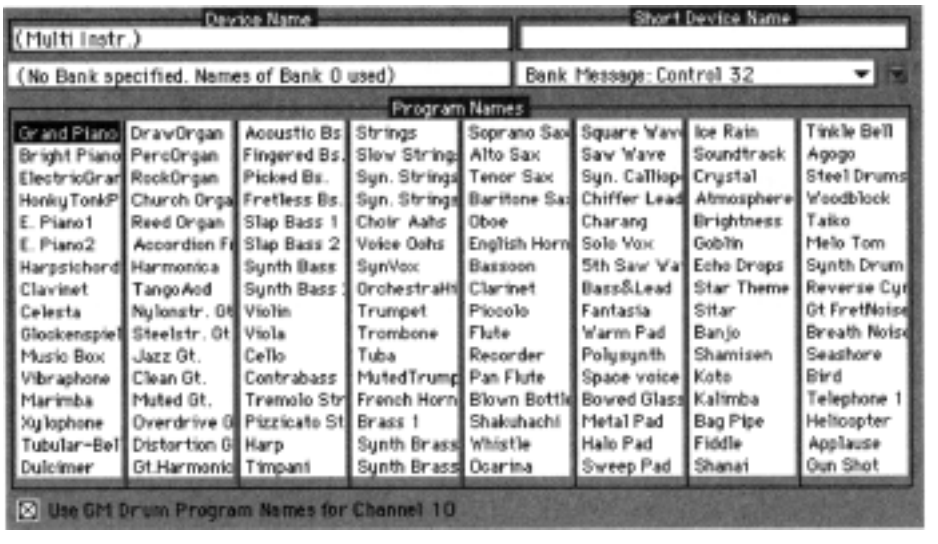
The sub-channel parameters are the same as for Standard Instruments (see page 5-24). The only parameter you can not change is the MIDI channel. (If you change the output port, this will affect the whole multi-instrument and all other sub-channels.) If you try to change the channel from the Parameter Box in the Environment window you will get the following warning: “Channel protected! Please select a sub-channel from the Instrument flip menu.” However, if you change the channel in the Arrange window’s Parameter Box to a sub-channel that is on the instrument selection list (i.e. one without a slash through it on the multi-instrument), the track’s instrument will change to the selected sub-channel.

Cabling

To connect the output of an Environment object directly to the input of a sub-channel, hold down ***Alt***, and connect it the via the Instrument List (see section ***Cabling between Layers*** on page 5-14).

Multi-Instrument Window

Double-clicking on a multi-instrument opens the multi-instrument window:



Name and Short Name

In the top left, under ***Device Name, you*** can enter the full name of the multi-instrument. In the top right, under ***short Device Name, you*** can also enter a short name. This short name is used in the Arrange window track list when the program name is displayed (see below). The short name begins right next to the icon, so leave a space in front if you want uniform spacing in the track list.

Depending on whether or not you have checked the sub-channel's *Prg* box, the following information will appear in the Arrange window track list:

the name of the multi-instrument & channel number (if the box is not checked) or

the short name, channel number and program name (if the box is checked).

ProgramNames

There are 128 program names in the multi-instrument window. A total of 15 banks of 128 program names are available. There are several ways of entering the program names:

by double-clicking on the name (via the text input field),

by copying via the clipboard from a different multi-instrument, or from a word processing program. The clipboard functions for a whole sound bank are available in the text import flip menu (downwards arrow at the top right). First copy the program numbers or General MIDI names to the clipboard, and add them to a word processor document. You can then edit the names and copy the whole section back again.

If you want to use program numbers instead of names, select **Init Names as Numbers** from the text import menu.

if you want to use General MIDI program names, select **Init General MIDI Names.**

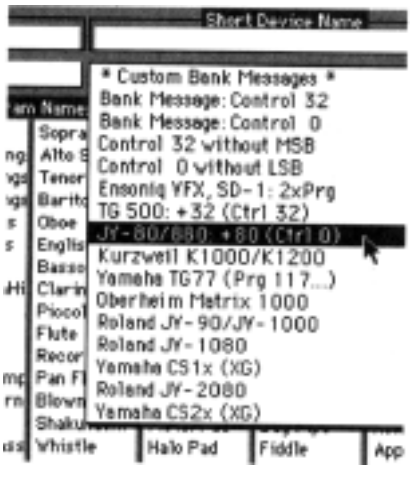
if you have Emagic's SoundDiver editor/librarian software: start SoundDiver then switch back to Logic and select **Update Using Autolink** The current program names for the currently selected bank will then appear in the multi-instrument window automatically.

If the ***Use GM Drum Names for Channel 10*** box at the very bottom is checked, the standard GM names for drum sets will be shown in the multi-instrument window for sub-channel 10.

If the Prg box in the selected sub-channel's parameter box is checked, you can send a program change messages by selecting a program names in the multi-instrument window (either by clicking, or using your computer's arrow keys).

Banks

On the left, above the program names, you can choose one of 15 available sound banks (0-14) via a flip menu. The top item ("No Bank specified. Names of Bank 0 used.") can be used if that particular sound generator does not understand bank select messages, or only has 128 sound programs. Bank 0 is always initialized. The first time you choose one of the banks 1-14 you will be asked whether you want to initialize this bank. Press **OK** or ***Cancel*** if you do not want to initialize the bank. Non-initialized banks do not have their own program names, but use the names of the equivalent program numbers from bank 0. You should only initialize additional banks if you want to enter program names for those banks, because each initialized bank uses more memory.



In the flip menu to the right, labeled ***Bank Message,*** you can define the MIDI events which are sent when you switch banks on this multi-instrument. Different bank select formats are used by different MIDI instrument manufacturers-please consult your instrument's manual to see whether it supports bank select messages and if so, using what format. With modern synths, there is a good chance the format will be one of the top items in the flip menu: either controller #32 or controller #0. There are also presets to accommodate several of the more common types of synth. If your synth does not use one of the formats on the list, don't give up-all is not lost!

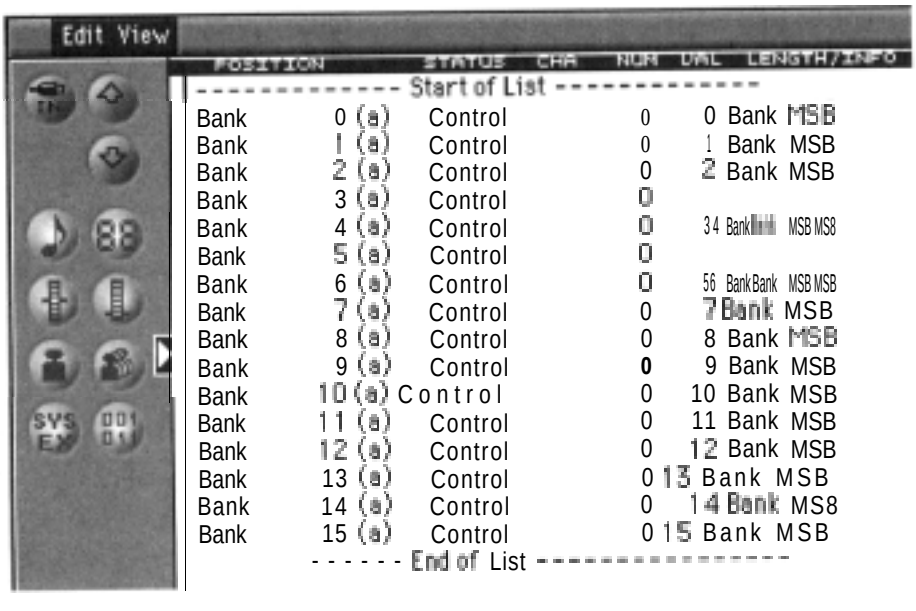
Defining Your Own Bank Select Commands

Logic provides 63 bank numbers (0 thru 62) for each single standard instrument, multi-instrument sub-channel, or mapped instrument. You can create custom lists of as many events of any kind (even Sysex) as you wish - for each of these banks. Whenever you change the bank manually or send a standard bank change from Logic, the entire list for that bank will be transmitted to your synth.

Note: if you need more than 63 banks for a particular MIDI device, use several instrument objects.

To set up custom bank select messages, first select the instrument, then select **Options > Define Custom Bank Messages...** A window similar to the event list will open, with a single,

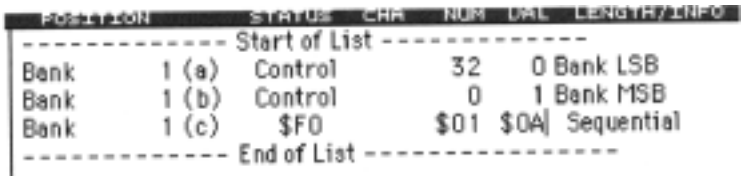
default bank select message for banks 0 thru 15. (The default used is a MIDI controller #0 message with value equal to the bank number.)



You can create MIDI events here in exactly the same way as you do in the Event Editor, by cutting, copying, inserting and editing. The only difference is that you enter the bank number you want, instead of a time position.

The letter in parentheses next to the bank number allows you to control the order of transmission, when a bank uses more than one message.

For example:



Here the Bank Select command “Bank 1" has been set up to send three events: first, (a) controller #32 message with the value 1, then (b) controller message #0 with value 0 and finally, (c) a Sysex message.

If there are no events defined for a particular bank, a standard Bank Select message will be sent for that bank instead.

The custom bank select information becomes part of the instrument, and will be automatically copied along with it.

For bank messages that need a channel (e.g. MIDI controller messages), the channel of the instrument is used. (This feature is especially useful for multi-instruments, since you only have to create one set of bank messages for all 16 sub-channels.) If an instrument's channel is set to *All*, then channel 1 is used.

On multi-instruments, the bank select message (whether it's a standard controller or a string of events you've configured) is also used by Sound Diver, if Autolink is switched on. This enables it to request the program names in the current bank.

Mapped Instruments

To create a new mapped instrument, select **New > Mapped Instrument.**

A mapped instrument is useful for drum programming. It is used just like a standard instrument, but has the following special features:

Each individual input note can be:

named (e.g. snare, hi hat...):
mapped to an output note (e.g. F# and G# to the same hi-hat sound to allow you to play rapid repeats);
given a velocity offset;
assigned its own MIDI channel;
sent to one of up to 16 output cables (this allows you to create a single instrument that addresses multiple sound sources);
given its own notation parameters:

note head shape
relative vertical position in the staff
drum group assignment

Object Parameter Box

The mapped instrument's parameters are a subset of the parameters of a standard instrument (described on page 5-24). The

missing settings are available on a note-by-note basis in the mapped instrument window (see page 5 -37).

Cha Use the ***Cha*** parameter to set the MIDI channel for the instrument's output. All events at the output of the instrument will be sent with the channel that is set here. Exception: If you set this parameter to All, all events will be sent with their original channel settings as they appear in the Event List window.

Port Below the channel parameter, a direct connection to one of the MIDI output ports can be set. Remember that you can also directly cable the instrument to other Environment objects for MIDI processing. If no MIDI output port is defined in the object parameter box for that instrument, the ***Port*** parameter will display off.

Prg, Vol, Pan The ***Prg, Vol*** and ***Pan*** parameters transmit program changes, volume controller (#7) and pan controller (#10) If there is no check in a box, you can edit its value, but it will not be sent until you check the box (by clicking on it). If the box is already checked, any value alterations will be sent immediately, and the values will also be sent whenever the track is selected.

To the left of the program number, directly to the right of the box, is the parameter for bank select. If your MIDI sound sources recognize Bank Select messages (check in their manuals), you will be able to switch between sound banks. If your sound sources respond to the standard Bank Select message (Controller #32), you will be able to use this parameter directly. If not, please read the section section ***Defining Your Own Bank Select Commands*** on page 5-

Delay The ***Delay*** parameter causes all MIDI events to be sent early or late by from -99 to 99 ticks. This allows you to adjust for any differences in reaction time between your various MIDI devices. For delay effects use the sequence parameter of the same name, as this allows longer delay times.

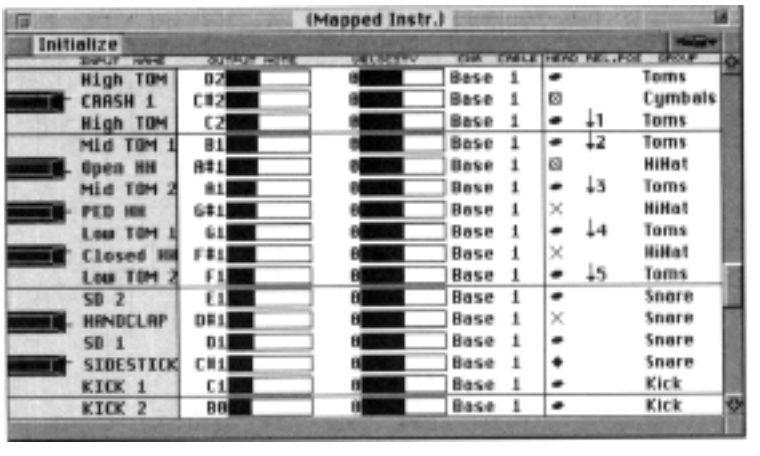
No Seq Trp If the box next to the No ***Sequence Transpose*** parameter is checked, all sequences on any tracks played by this instrument are protected from transposition. In other words, the “transpose” sequence parameter is ignored. This parameter is typically checked for mapped instruments since they generally are used for drum- mode MIDI devices and transposition would cause the sounds rather than the pitches to change.

No Reset If the box next to the No ***Reset*** parameter is checked, no reset messages will be sent to this instrument. This can be useful if controllers are being used for non-musical purposes, for example when an instrument is used for mixer automation. The **Preferences > Reset Messages** window controls what reset messages are normally sent, but these messages are not sent to ***"No Reset”*** instruments.

Default Score Style The ***Default Score Style*** parameter is located at the bottom of the Parameter Box. It displays ***Auto Style*** by default but can be changed to any of the available score styles. Whenever a sequence is created on one of the instrument’s tracks, it will be assigned the score style displayed here. In the case of ***Auto Style*** Logic will pick an appropriate style based on the pitch range of the notes in the sequence.

Mapped Instrument Window

To open the mapped instrument window, double-click on the napped instrument icon in the Environment window. The rows correspond to input notes, and the columns contain the various parameters available for each note.



Monitor Keyboard (Selecting Notes)

The keyboard on the left represents the input notes. It can be played by clicking on it. You can also select individual notes or note ranges, by dragging the mouse over the notes you want. To select more notes, click them while holding ⇧. Any value alterations will apply to all selected notes.

Input Name

In the next column, you can click on the input note and give it a name of up to 12 characters. You can initialize the names of the selected notes to their pitch descriptions (e.g. (C#3) by selecting **Initialize > Names as Notes,** or to the names of the GM Standard drum sounds by selecting **Initialize > Names as General MIDI.**

Output Note

This column is where you set the output note either as text by double-clicking on the note description, or graphically by dragging the beam. When you change the value, MIDI notes are sent out so you can hear what you're doing. If you choose

Initialize > Output Notes, the output notes of the selected pitches are set to be the same as the input notes.

Velocity

This is where you set a velocity offset, which is added to or subtracted from the velocity of the incoming note. You can either grab the number and use the mouse as a slider, or click the beam at the desired point. **Initialize > Output Velocities** resets all velocity offsets to 0 (no offset).

Cha

Here you can set the MIDI channel of individual notes. This allows you to play individual sounds from different drum sets in the same sound generator. Normally you will use the ***Base Setting*** here. This means that the notes are sent on the channel set in the mapped instrument's object parameter box. If you choose ***All,*** the channel information of the incoming notes is used. This is useful if you want to insert the mapped instrument in the signal path after a multi-instrument or single instrument. To set all selected notes to ***Base,*** select **Initialize > Output Channels.** Please also refer to the section ***Setting the MIDI Channel: Multi Instruments*** on page 3-15.

Cable

You can send individual notes from the mapped instrument object to up to 16 different output cables, to enable you to play sounds from different sound sources. If you select a cable that does not exist, then the note will not be sent out. **Initialize > Output Cables** resets all cables to #1 (the top outlet of the mapped instrument).

Notation Parameters

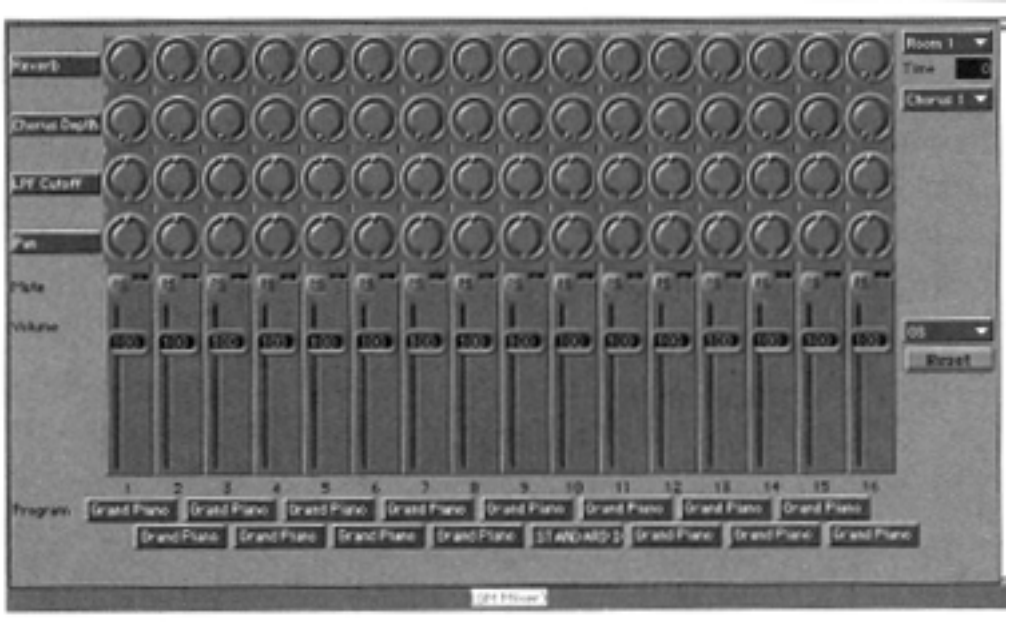
The last three columns define the notation parameters of the individual notes. Selecting **Initialize > Score Parameters** neutralizes the settings of the ***Head, Rel Pos.*** and ***Group*** parameters.

Faders

To create a new fader, select **New > Fader >...** A sub-menu appears, where you select the fader type you want. Faders are such a diverse and important part of the Environment, that we have devoted an entire section just to them. Please see the section *Faders* on page 5-68.

GM Mixer

To create a new GM Mixer, select New > **GM Mixer.**



The GM Mixer is intended for controlling the 16 channels of a MIDI device which conforms to the GM, XG or GS standard. This includes the GM set of 128 program names (with variation banks for XG or GS), MSB/LSB bank select messages, and standard controller names (#1 for mod wheel, #2 for breath, #7 for volume, #10 for pan, #11 for expression, etc.).

Most of the GM Mixer's set up is done in the mixer itself, but there are also a few parameters in its parameter box.

Cha/Port Like an instrument object, the GM Mixer can also have a direct MIDI output connection. This is set in the line below the channel in the Parameter Box (see also page 5-12). The *Cha* para-

meter itself just sets the channel of the lowest GM Mixer module. You can use this, together with resizing the GM Mixer window, to create sub-mixers for any continuous string of channels (e.g. 3 thru 7).

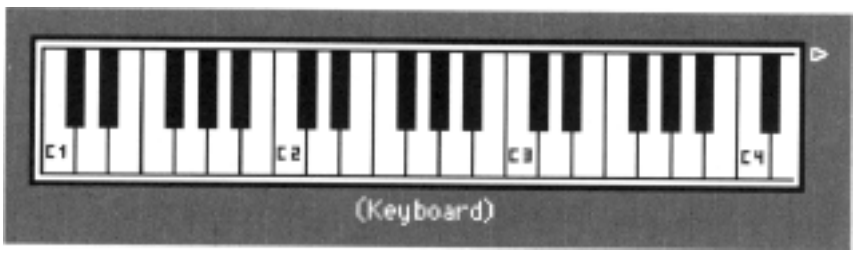
Legend — The *Legend* checkbox toggles the display of the legend along the left edge of the GM Mixer. The Legend not only indicates what the rows of controls do, it is used to set the function of the top four rows of knobs. After the GM Mixer is set up, you can hide the legend to save space.

Bank — The *Bank* checkbox toggles bank MSB/LSB display below the program name display, at the bottom of the GM Mixer. (Bank display is off in the illustration above.) You can save space by hiding bank display, unless you need to select program banks numerically.

Style — The GM Mixer comes in three flavors. Choose the one most pleasing to your eye.

Keyboard

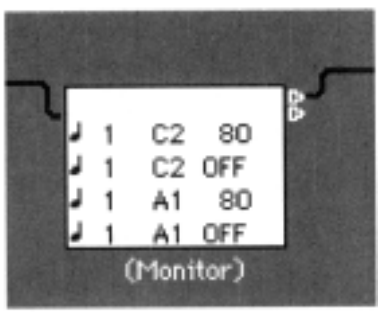
You can create a virtual keyboard by selecting **New > Keyboard.**



The keyboard object generates MIDI note events when clicked with the mouse. It also displays note events passing through it-when a note is being held, the corresponding key will reverse its color. Although you can record the output of the keyboard in Logic, its main purpose is for testing and monitoring in the Environment.

Apart from the standard parameters, you can also set the MIDI channel, a fixed velocity for all notes and the octave of the owest note on the keyboard (which is always a C). You can re-lize the keyboard to control its range above the lowest note. Click the keyboard with ⌂ to select and move it.

Monitor

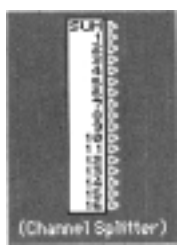


To create monitor, select **New > Monitor.** The monitor shows all events (MIDI and meta) passing through it. It remembers the last 32 events, with the newest events at the bottom. You can resize it to show from 1 to 32 events. (Actually you can make it bigger but the display above the 32nd line is gibberish.)

Monitors are very handy as both testing, and branching devices. In the case of objects that have functionally different outputs [e.g. the channel splitter and the mapped instrument), you can use them to attach more than one output cable with the same function.

Click in the body of the monitor to clear it.

Channel Splitter



To create a new channel splitter, select **New > Channel Splitter.** You can use the channel splitter to reroute MIDI events to different cables, according to their MIDI channels. Every MIDI event received at the input of the channel splitter is first routed to the output corresponding to its MIDI channel. If there is no cable connected there, the event is rerouted to the SUM output (at the top). The SUM output carries the events of those channels which have no cables from their individual outputs.

If you want to route the information for a single channel to more than one object, use a new transformer or a monitor as a junction (see above and page 5-14).

Transformer

You can create a new transformer object by selecting **New > Transformer.** The transformer allows you to both select and alter MIDI events in real time.

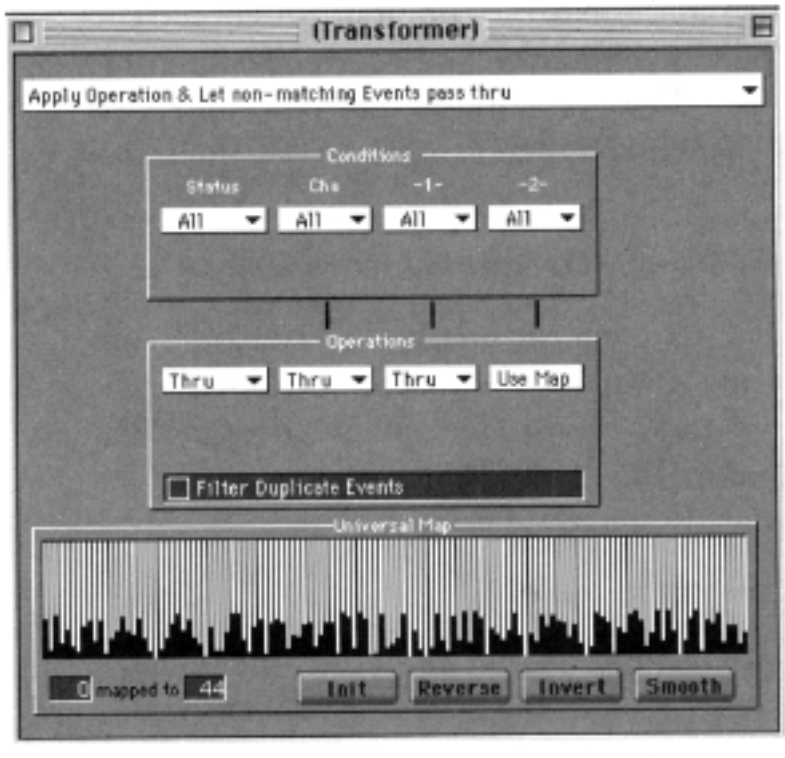
For the transformer to work, it must be in the MIDI signal path. If you use it to alter the output of an instrument, for example, cable it between the instrument and a port object (removing the instruments internal port assignment in the process).

Working Principle

The transformer first checks whether an event corresponds to certain *conditions*. If it does, then certain *operations* are carried out. Events which do not meet the conditions may pass unchanged to the output, or be thrown away. The transformer can also make copies of matching events before altering them, and can split the selected events from the unselected ones, using its two top cables.

Transformer Dialog Box

Double-clicking on a transformer opens a dialog box, where you can set the conditions and operations. You can open this box as a floating window, by double-clicking on the transformer while holding the ***Alt*** key.



Mode of operation

Use the flip menu at the top of the transformer window, to set the transformer's mode of operation.

Apply Operation & Let non-matching Events pass thru

MIDI events which conform to the condition are processed. MIDI events which do not conform to the condition are passed thru. This is the default setting.

Apply Operation & Filter non-matching Events

MIDI events which conform to the condition are processed. MIDI events which do not conform to the condition are not passed through.

Filter matching Events

All MIDI events which match the condition are filtered out. MIDI events which do not conform to the condition are passed through.

Copy matching Events & Apply Operation

All MIDI events which conform to the condition are copied and the copy is processed. The original *and* transformed copy (plus any MIDI events which do *not* conform to the condition) are passed thru. The unchanged original is passed *before* the transformed copy.

Copy matching Events & Apply Operation (rev. order)

This is the same as above except the original is passed *after* the processed copy. You might want to use this for example, when converting note events to pan controllers. This causes the pan message to be sent before the note. (Many synths do not alter the pan position of already sounding notes.)

Condition Splitter (true -› top cab/e)

Events which match the conditions are altered by the operations and sent to the transformer's top output. Events which do not match the conditions are sent, unaltered, to the transformer's second output. Do not use the other outputs-nothing ever appears there.

Alternating Split

Events entering the transformer will be alternated between the top two outlets. No conditions or operations apply.

SysEx Mapper ('1 '+position, '2'-›value)

This transformer mode is used to create and edit Sysex messages. You first enter the structure of the Sysex message in the transformer window-i.e. its length, whether a checksum is required, and the values of bytes you don't want to change in real time. Next you select which kind of MIDI event will affect the data bytes in the Sysex message. (Typically you will use MIDI controller events.) Incoming MIDI events will then change the data bytes in the Sysex message according to the following rules:

The -1- value sets the position of the data byte to change.

The -2- value sets the value of the data byte.

The channel controls the action to be taken:

Channel 1: The changed SysEx message is sent.

Channel 2: The changed SysEx message is not sent.

Channel 3: The unchanged SysEx message is sent.

Channel 4-16: No meaning (reserved for future use).

The purpose of the channel controls is to allow multiple bytes of a Sysex message to be modified before it is sent.

The "Filter non-matching" parameter in the sysex mapper transformer window prevents incoming MIDI events which do not control the Sysex message from being passed through. Typically you will want this feature turned on, to prevent interloping controller data from invading the Sysex data stream.

Filter Duplicate Events

You can use this feature to save MIDI bandwidth. When this checkbox is selected, the transformer will filter out a MIDI event if:

its type matches the transformer's Status condition. (This Status condition must not be set to "All" for this to work.)
the previous event matching the Status condition was identical (i.e. same channel, -1- and -2- value).)

Note that you can use this feature with the Status condition set to "All" to filter out duplicates, but here duplicates will only be recognized if they are consecutive (i.e. there are no other events in between).

Conditions and Operations

The conditions and operations are the same as for the Transform window. The only differences arise from the fact that transformer objects work in real-time and therefore, position and note length have no useful meaning.

Pitch Bend editing

The transformer can process 14-bit pitch bend events: if byte is changed (e.g. by addition or scaling), the change will affect 1

the second byte too. Set the -2- operation to "THRU" to process 14-bit pitch bend data properly.

Remote Controlling the Condition and Operation Values

The value(s) of any transformer conditions or operations can be externally controlled using meta events.

Meta events are internal Logic messages-their form is similar to MIDI controller events, but they have no MIDI meaning and never leave Logic.

Like MIDI controller events, all Meta events have a two data values: the first one indicates the type of Meta event (49-127 but not all are used), and the second one is the event value (0-127).

Meta events number 122 thru 127 affect transformer parameters.

Meta event #127 sets the first (top) operation value,

Meta event #126 sets the second (bottom) operation value (if available).

Meta event #125 sets the first (top) condition value.

Meta event #124 sets the second (bottom) condition value (if available).

Note that *all* numerical condition and operation parameters that are used (i.e. conditions not set to All and operations not set to Thru) will be affected by the same meta event. (The status condition and operation are never affected by meta events.) If you want to set numerical parameters individually, or leave some fixed, use separate transformers in series.

Meta event #123 sets the transformer map value for the current map position. (Use meta event #122 to set the position first.)

Meta event #122 sets the transformer map position.

Note that you can use a transformers *MapSet* operation to create these two meta events simultaneously. When the operation status is set to *MapSet* then the -1- parameter specifies the map position and the -2- parameter specifies the map value at that position. (I.e. a meta event #122 is sent with the -1- value followed by a meta event #123 with the -2- value.)

You can create meta events with a fader; you can transform MIDI events into meta events with a transformer, and you can add meta events to a sequence using the Event List editor. When you use a fader, set the Out definition to *Switch*, then set the -1- value to the meta event number. (Cable switcher type faders and meta events share the same Outdefinition. -Z-values above 49 signify meta events.)

Since you can remote-control any fader by using its In definition, you can use faders as well as other transformers to convert MIDI events to meta events for remote-control of transformers.

Condition: Map

Most of the transformer conditions are numerical relations (=, unequal, inside, etc.), and the condition is met if the incoming MIDI event value satisfies the condition. The last condition, *Map*, works a little differently. In this case, two numerical parameters are specified, and the incoming value is first converted by the map, then the mapped value is compared to the two parameters to see if it is inside them.

Transforming Meta Events

Transformers can also process meta events, as long as they are not the meta events that affect transformers. These particular meta events change the transformer, rather than being processed by it.

Arpeggiator

To create a new arpeggiator, select **New > Arpeggiator.**

The arpeggiator cycles through all held notes (i.e. arpeggiates chords) within its range. You can set the range as wide or narrow as you wish. There are parameters for direction, velocity, speed, note length, start quantize (a.k.a. snap), repeats, octaves, and velocity offset. All the arpeggiator's parameters can be MIDI controlled in real time, which adds immensely to its versatility.

To use the arpeggiator, you must put it in the MIDI signal path and Logic's Transport must be running. (For technical reasons, the arpeggiator resets on cycle jumps.) Typically you would assign the arpeggiator to an Arrange track, and cable its output to an instrument or port object. Of course, you can insert it anywhere else in the MIDI signal path that you like.

You can then use the arpeggiator with live MIDI input, or for sequence playback. You can also record the output of the arpeggiator by cabling it back to the sequencer input object. If you do this, be sure you either record to a no output track, or break the arpeggiators connection to the MIDI output.

Parameters

An arpeggiator has the following parameters (the numbers in parentheses are *CTRL Base* offsets-see below):

Direction (+0)

The direction of the arpeggiated chord:

Up	Lowest note to highest note
Down	Highest note to lowest note
UpDn	Up and down-highest & lowest notes repeat
Auto	Up or down depending on whether the second chord note arrived before or after the first chord note
UpD2	Up and down-highest & lowest notes don't repeat
Rand	Notes play in random order
All	All notes play at once (useful when repeat is on)

Vel (+1)

Velocity values of the arpeggiated notes.

1-127 Fixed velocities,

Orig the velocities of the recorded notes are retained,

Rand random velocities between 1 and the original value.

Lim (Low: +2 High: +3)

This is where you can define the pitch range (between C-2 and G8) within which the chords are arpeggiated. Any chord notes outside this range are passed directly to the output. This is useful when you want to solo above, or play a bass line below an arpeggiated chord.

Res (+4)

This is where you set the rhythmic note value (i.e. the speed in note divisions) of the arpeggio via the flip menu shown here. None = arpeggiator switched off.

Len (+5)

This is where you define the length of the arpeggiated notes. The Orig setting means that the lengths of the recorded notes are retained.

Snap to (+6)

Unless this value is set to None, the arpeggiator will wait for Logic's transport to reach the next indicated note division before starting the arpeggio. This is useful for rhythmically syncing the arpeggio with other MIDI data.

Repeat (+7)

On continues the arpeggio for as long as the chord is held down.
Off finishes the arpeggio after one run.

Octaves (+8)

The arpeggio can be repeated over 1-10 octaves.

Crescendo (+9)

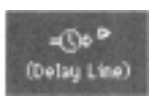
The velocity value set here (-99 to +99 is added every time the arpeggio is repeated (provided *Repeat* is set to On, of course).

Ctrl Base

All 10 parameters of the Arpeggiators can be remote controlled by controller events. With the parameter Ctrl Base, you can determine the controller number for the first parameter (Direction). The other parameters will be controlled by subsequent controller numbers. To find out which controller number will be used for a certain parameter, add the number shown in brackets behind the parameter's name in this documentation, to the Ctrl Base value. When *CTRL Base* is set to Off MIDI control of the arpeggiator is turned off.

Delay Line

To create a new delay object, select **New > Delay line.**



This object repeats any MIDI events, like a delay line. Pitch and velocity offsets can be added to note events. Like the arpeggiator, you will need to put the delay line in the MIDI signal path and you will need to start Logic's Transport for it to work.

The delay line can send up to 99 repeats, at intervals ranging from one tick to 256 whole notes. You can suppress the original note. Repeats leaving the delay line will cycle through all cabled outlets, so that you can play each repeat with a different sound.

Parameters

A delay object has the following parameters:

Thru Original: if this box is checked, the original signal is passed thru, otherwise it is suppressed.

Repeats: number of event repeats (1 to 99). Setting 0 = delay switched off.

Del: the delay time between the individual repeats. The left value is in divisions and the right value in ticks.

Trp: the transposition of note events per repeat (+ 99 semitones).

Vel: the change in the velocity values of note events per repeat (± 99).

Voice Limiter

To create a new voice limiter, select **New > Voice imiter.**

The voice limiter limits the number of MIDI notes (1 to 32) that can be held simultaneously. It does this by "note stealing"- newly arriving notes will cause some currently held notes to be turned off, once the voice limit is reached (see the ***Priority*** parameter below).

Normally you will assign the voice limiter to an Arrange track, and cable its output to the instrument object whose voices you want to limit. Or, you could cable it between an instrument object (breaking its internal port connection) and a port object.

Parameters

Voices: this is where you define the maximum number of simultaneously played voices, from 1-32.

Priority this is where you define which notes get stolen (i.e. turned off) when the number of held notes exceeds the limit. ***Top*** = lowest notes are turned off first. ***Bot*** = the highest notes are turned off first. ***Last*** = the earliest notes are turned off first.

Chord Memorizer

To create a new chord memorizer, select New > **Chord Memorizer.**

The chord memorizer maps individual notes to chords. You can assign one chord to each pitch class (i.e. to C, C#, D, etc.). The octave of the incoming note determines the octave of the resulting chord. A chord can have from 0 to 12 notes in it. (Zero and one note chords can be useful for creating scale-filters and scale-correctors).

The easiest way to use the chord memorizer is to connect its output to the instrument that you want to have play the chords, and assign it to an Arrange window track. You can, of course, put t anywhere else in the MIDI signal path that is convenient.

Parameters

Cha: All chord notes will be output to the channel set here.

Lim: Notes within this range are mapped to chords. Notes outside the range are passed through unaltered.

Trp: The output chords are transposed by the amount set here (+/- 99 semitones).

Example: if you map C to the chord CEG and set transpose to 1, then C will be mapped to C# F G#,

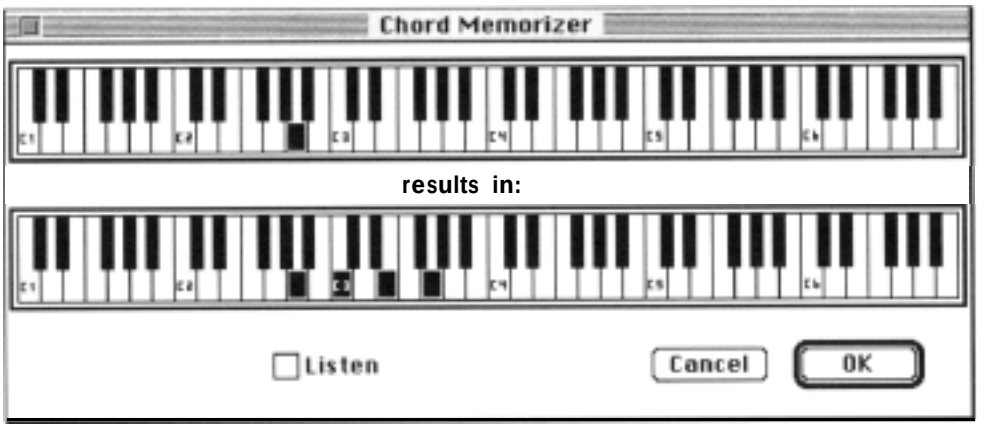
Key: The entire chord map is transposed by the amount set here (+/- 99 semitones).

Example: if you map C to CEG and set key to 1 then C# will be mapped to C# F G#.

Example: if you want to map single notes to triads within a given key, first set the key then define each of the triad chords (see below). When you want notes mapped to the triads in a different key, just change the key parameter.

Chord Memorizer Window

Double-clicking on the chord memorizer opens the chord memorizer window, where you can set up the chord definitions



The top keyboard is for inputting/displaying the incoming note, and the lower keyboard is for inputtingldisplaying the assigned chords. Notes can be input using the mouse, or a MIDI keyboard.

Remember to use notes on the top keyboard within the ***Lim*** range. Also remember that you can only define one chord for each of the 12 pitch classes (i.e. C, C#, D etc.). You *can* of course, use more than one chord memorizer.

Entering Chords with the Mouse

First click the input note on the top keyboard. All notes in the associated chord will become inverted on the bottom keyboard.

Next, click notes on the bottom keyboard until the inverted notes make up the chord you want. You can click notes either on or off on the bottom keyboard. Also, you don't have to stay within the same octave as the input note.

When you have the chord you want, select another input note, or close the chord memorizer window.

Entering Chords from Your MIDI Keyboard

First play the "input note".

Next click the ***Listen*** checkbox. The bottom screen keyboard is inverted to indicate it is ready to record.

Now play the notes in the desired chord. (You can play them one at a time or as a chord.) If you make a mistake, uncheck the ***Listen*** checkbox, then check it again to start over.

When you have the chord you want, uncheck the ***Listen*** checkbox and play a new input note, or close the chord memorizer window.

Touch Tracks

Touch Tracks turns Logic into an interactive ***live*** sequencer. A touch tracks object allows you to trigger MIDI sequences or folders with single notes. (You can not use Touch Tracks to trigger audio.). (In the following section, any references to sequences means folders as well).

To create a Touch Tracks object, drag a sequence or folder from the Arrange window into the Environment window or select **New > Touch Tracks'".** To use a Touch Tracks object, simply assign it to an Arrange window track, or put it anywhere in the MIDI signal path.

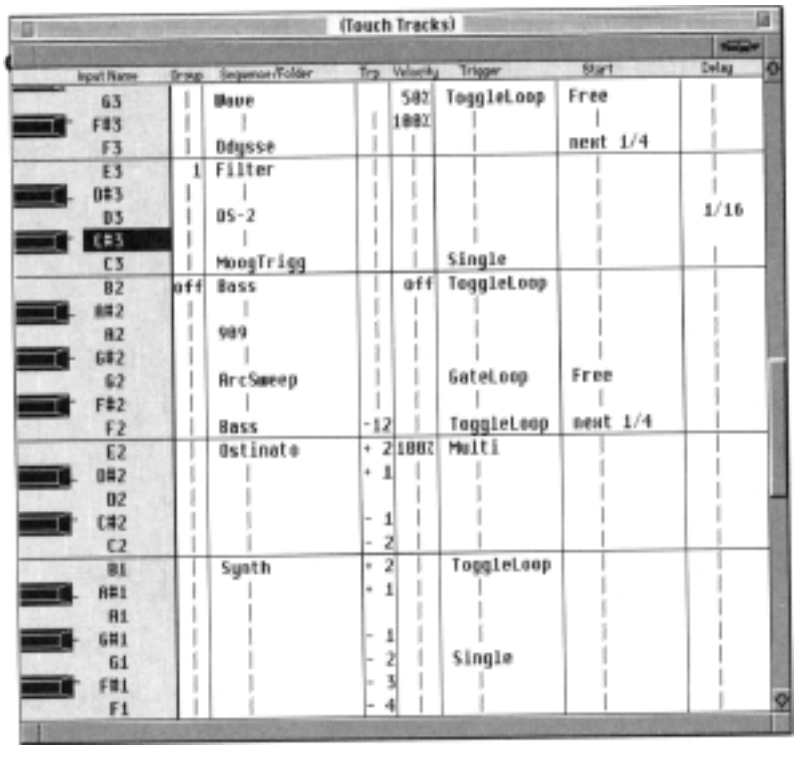
Here are some things to remember:

Only the Touch Tracks input has meaning-the trigger notes must appear here. Although there is an output triangle, it has no meaning-no events ever appear here.

Sequences triggered by Touch Tracks play exactly as they would from the Arrange window. In particular, they play back through the instruments assigned to their tracks.

Logic's Transport must be running for Touch Tracks to work.

The Touch Tracks



You open the Touch Tracks window by double-clicking on the the Touch Tracks object. This window is set up similarly to the mapped instrument window (see page 5-37) - the input note is selected using the keyboard on the left, and the output sequence assignment and parameters are set in the columns of the corresponding row,

Sequence Note Assignment

Dragging a sequence into the Environment initializes a Touch Tracks object, so that initially all notes trigger this sequence. (Middle C plays it at its original pitch, and all other notes transpose it relative to middle C.)

In the ***Input*** column, you can see the input notes, and to the right, in the ***Sequence/Folder*** column, the names of the assigned sequences or folders. On the vertical keyboard to the left, you can select individual notes or whole pitch ranges, by grabbing and dragging. If you then drag a sequence from the Arrange window, it will only be assigned to the selected note(s).

Remember that with the following parameters, all settings also apply to multiple selections.

A **vertical** gray line means that the setting is the same as the line above. If you change a vertical gray line that is above another vertical gray line, the lower one will change to display its previous value. (I.e. it will no longer be the same as the line above.)

Group

To the left of the sequence name is the group column. Groups work the same way as they do in the Hyper Editor-when you trigger a sequence, any other sequences in the same group that were playing will stop.

The ***off*** setting means that the sequence is not assigned to any group.

Transposition

When you drag a sequence to a key in the Touch Tracks window, that key triggers the sequence at its original pitch (i.e. without transposition). If you want the sequence transposed, you can set the amount in the ***Trp*** column (± 99 semitones).

If a key range is selected when you drag the sequence to the Touch Tracks window, then incremental transpositions will automatically be set for adjacent keys.

When you create a Touch Tracks object by dragging a sequence into the Environment, C3 triggers the sequence at its normal pitch, and all other keys trigger it transposed relative to C3.

> If you do not have Display ***Middle C as C3 (Yamaha)*** checked in the **Options › Settings › Display Preferences** window, then C4 rather than C3 will trigger the sequence at its original pitch. This will actually be the same MIDI note event, but Logic will use the Roland naming convention of C4 for middle C, rather than C3.

Velocity Sensitivity

In the ***Velocity*** column, you can set how much the velocity values of the played sequences are affected by the velocity value of the trigger note: by 100% (meaning a lot), 50% (meaning a little) or ***off*** (meaning not at all).

Trigger Modes

The Trigger column sets how sequence playback is handled:

Multi	Playing the trigger note starts the sequence. Playing it again starts the sequence again, without stopping the original version.
Single	Playing the trigger note once starts the sequence. Playing it again stops playback, and starts the sequence over.
Gate	The sequence plays until the trigger note is released (or until the sequence ends).
Gate *Loop*	The sequence loops until the trigger note is released.
Toggle	Playing the trigger note starts the sequence. Playing it again stops playback.
Toggle Loop	Playing the trigger note starts the sequence looping. Playing it again stops playback.

Start

In this column you can quantize how sequences start and stop. ***Free*** means there is no quantization. The ***next*** 1/16, 1/4 or 1/1 settings start or stop the sequence at the next 1/16 or 1/4 note or at the start of the next bar, after a trigger note is input.

Delay

In this column you can assign the sequence a delay. As in the sequence parameter box, the delay is set on the right side of the column in ticks, or on the left side in note values.

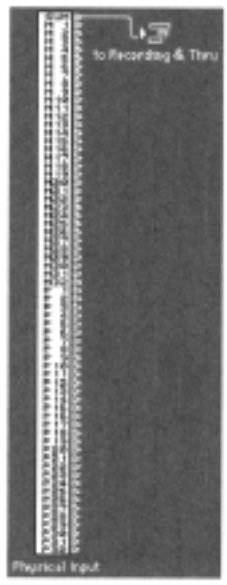
You can use ***Delay*** and ***Start*** together to make sequences start at any position in the measure. E.g. Set ***Start*** to 1/1 and set ***Delay*** to 480 ticks to start playback at the second 8th note in the measure.

Physical Input/Sequencer Input

The Physical Input represents the MIDI inputs on your Macintosh MIDI interface; the Sequencer Input symbolizes Logic's MIDI input. You can (and should) have only one of each of these objects in the Environment.

To create an input object, select **New > Physical Input** or **> New > Sequencer Input.** To make use of an existing object, drag it onto the relevant layer. This will not affect its cabling.

Physical Input



The physical input represents the MIDI inputs on your Macintosh MIDI interface. Each Song can have only one physical input object. You only need to create a physical input object if you have deleted the one that was there when the Song was created. In this case, select **New > Physical Input.**

The physical input object is where the MIDI signals are received from the inputs of the MIDI interface(s) connected to your computer. This object has a total of 65 outputs. The outputs labeled Ml to M32 carry the MIDI data from the first 32 individual inputs of the interface(s) connected to "MIDI connection 1" and "MIDI connection 3", as set up in **Options > Settings > MIDI Communication Preferences.** The bottom 32 outputs on this object are labeled P1 to P32, Those carry the first 32 individual inputs of the interface(s) connected to "MIDI connection 2". Depending on the type (and age) of your computer, these interfaces can be connected to the modem or printer port, or to the ***universal serial bus*** (USB) The connection assignment is set up in the ***communication preference.***

The top output (SUM) carries the MIDI events for all of the individual outputs which are not cabled separately.

MIDI Remote Control

Remote control events are intercepted at the physical input, and are not passed through to its outputs. Therefore, remote control events will not reach the Environment, or be recorded

on Arrange window tracks. If you want to trigger a MIDI Remote Control action from a fader or transformer, cable that object's output back into the physical input object.

Define your own MIDI Remote Commands, using the Key Command window. (section Assigning a Function to a MIDI ***Message*** on page 1-47).

Sequencer Input

MIDI events arrive at Arrange window tracks thru the sequencer input object. If nothing is cabled into the sequencer input object, nothing can be recorded in Logic. Typically, the physical input is cabled directly to the sequencer input but other objects can be inserted between them. Typical candidates are a monitor, a keyboard object or a cable switcher feeding various effects such as arpeggiator, a delay line, etc. (See the *Clicks & Ports* layer of the Logic default Song.) You can also cable the output of complex Environment processes back into the sequencer input object in order to record the processed signal.

Each Song can have only one sequencer input object. You only need to create a sequencer input object if you have deleted the one that was there when the Song was created. If you do need to create one, select **New > Sequencer Input.**

MIDI Metronome Click

The MIDI metronome click object is used to create note events at bar, beat and division intervals. There is a button on the Transport to turn the MIDI metronome click on and off. You can also set the MIDI metronome click to activate during recording and/or playback in the **Options > Settings > Recording Options window.**

Each Song can have only one MIDI metronome click object. You only need to create a MIDI metronome click object if you have deleted the one that was there when the Song was created. If you do need to create one, select **New > MIDI Metronome Click.**

The checkboxes after *Bar, Beat and Division* allow you to generate note events separately for bars, beats and divisions. The *Cha, Note* and *Vel* parameters define the MIDI channel, note # and velocity of the generated notes. The line directly below the *Cha* parameter for *Bar (parameter Port)* allows you to set a direct output port for the metronome.

Internal

"Internal" objects include instruments that are located inside your Macintosh (such as a Nubus card, PCI card and/or software synthesizers). Although communication with sound sources inside the computer does not actually use MIDI, it still conforms to the MIDI communication protocol. Internal objects have an input but no output. They therefore act like extra output ports which are hard-wired to the relevant sound generator.

Using internal objects

If you want to address internal sound sources, you must first create an equivalent internal object (see below). Then create a standard instrument, multi-instrument or mapped instrument for the device. Next connect the output of the instrument to the internal object. Then use the instrument as you normally would.

Use the Which parameter to select which (up to 16) internal device an internal object represents.

Digidesign SampleCell

Digidesign's SampleCell II is a 32-voice sample player on a PCI or Nubus card. To create an internal object for the SampleCell II card, select **New > Internal > SampleCell.**

The options for the SampleCell II card are similar to those of any good stand-alone sampler. Because there are no delays due to MIDI transfer or decoding, the timing is far superior to that of external MIDI sound sources. There should be no need to set any delay for this multi instrument.

Apple QuickTime

The QuickTime system extension is a software GM sound generator. The "QuickTime™ Musical Instruments" file must be present in your System folder for it to work.

To create a virtual representation of the QuickTime synth, choose **New > Internal > QuickTime.**

This software synth is 16-part multi-timbral. As usual, you can switch sounds via program changes; these are arranged according the GM table. Channel 10 is always set to drum sounds, with GM key assignments. The QuickTime synthesizer reacts to notes and program change commands, as well as to pitch bend (+/- 2 semitones) and the main volume controller (#7).

The first note is used to initialize the relevant MIDI channel of the QuickTime synthesizer and therefore is not played.

Notes

Don't expect too much in terms of sound quality from the QuickTime synthesizer. It is not intended for use in professional productions.

The QuickTime synthesizer has a noticeable delay, that will vary according to your hardware, and the number of initialized MIDI channels. You can compensate for the delay by setting negative *delay* values in the object parameter box of the relevant (multi) instrument.

Morning Star MacWavemaker

Logic can control Morning Star's MacWavemaker soundcard directly. You can create an object to address the MacWave maker in the Environment window by selecting ***New > Internal > WaveMaker.*** All other details of setting up a multi instrument to control it are the same as with Sample Cell.

Retro AS-l, Unity DS-1

These objects represent direct outputs to the Retro AS-l software synth and the Unity DS-1 software sampler, if these are installed in your computer. The objects can be regarded similar to a MIDI output port, but supply a direct internal connection to those software sound sources. Therefore, it's not necessary (but still possible) to use OMS if you want to adress the AS-1 or the DS-1 from within Logic.

AutoLink Object

This object is designed to supply a direct internal connection to other MIDI applications which support the AutoLink standard that is used by Logic and SoundDiver.

Modem/Printer Port

Modem or printer port objects represent the MIDI output(s) of a MIDI interface connected to the modem or printer port.
Please note: These objects are not available if you use a newer Macintosh computer with USB connections (see below).

To create a port object choose **New > Modem Port** or **New > Printer Port.**

You can use the lowest line in the parameter box to set the MIDI output (1-99) for the port object. The signals from all the MIDI cables which you plug into this object will then be sent via the MIDI output you've chosen. The All setting means that the incoming MIDI signals are sent simultaneously to all outputs of the MIDI interface,

If you want to address several MIDI outputs separately, you have to use an interface which is compatible with Mark Of The Unicorn's MIDI Time Piece (like Emagic's Unitor8, Mark of the Unicorn's MIDI Express or Opcodeøs Studio 4 or 5). Only this kind of interface will allow you to activate the ***MTP Cable Mode*** checkbox **(Preferences > Communication).** All this,

however, only applies to "old" interfaces which are connected to the modem or printer port of older Macintosh computers.

There is no seperate port object for USB (Universal Serial Bus) connections. However, Emagic recommends (even for modem and printer port connections) not to use these port objects any more, but instead use a standard instrument object as a port object, set its Port parameter according/y (see section Direct Output Assignment on page 5-12) and connect the desired cables to it.

Remember that you can also address any of your MIDI output ports directly from any of the instrument objects (and the GM mixer object as well).

MIDI interface with only one MIDI port

If your MIDI interface has just one MIDI port set the port object to ***MTP port 0*** or ***All. You*** still have the option of connecting a second independent MIDI bus to the other Macintosh port (printer or modem).

Aliases

You can create an alias of any Environment object, and it will behave exactly like the original. In the case of faders, the alias has its own value which can be different than the value of the original. This is particularly useful in the case of text faders which take up a lot of memory-if you need several text faders with the same names, use one original and make the rest aliases.

To create an alias of any object, first select the object, then select **New > Alias.**

Aliases can be reassigned using meta events, and there is a special fader for accomplishing this which works something like the cable switcher. You can create an ***alias assigner*** by selecting New > **Fader > Specials > Alias Assigner.** This will create a meta fader whose -1- definition is set to 46.

To use the alias assigner, cable its top outlet to the alias, and cable subsequent outlets to the various originals you want to assign to the alias. If you set the alias assigner's maximum range to the number of original's, minus 1 (we're counting from 0 here), you can then connect the next cable to a new alias and subsequent cables to originals for that alias. In this way, the alias assigner can be used to assign multiple aliases at the same time.

One situation where you might use an alias assigner, is to switch a delay line alias between different original delay lines with different delay times. This is one way of simulating MIDI control over delay time.

Alias Parameters

Although aliases share the parameters of the original objects from which they are made, they also have their own, special parameters:

Icon & Instrument List Checkbox

As usual, the checkbox next to the word Icon determines whether the alias appears on the Arrange window's Instrument List, and you can assign any of the usual icons to it.

Ref

The Ref menu is used to select the original of which the alias is an alias. It is the same as the Arrange window's Instrument List.

Channel

For everything except faders, the channel parameter causes all events leaving the alias to be channelized to the selected channel. (If the setting is *All*, then their channel will be unchanged.)

For faders, the channel parameter causes the alias to act as if the original fader's In and Out definitions both were set to that channel. (Again a setting of *All* results in no change.)

Share Name/Share Size

These two parameters do just what they say-when checked the alias shares the size and/or name of the original, and if not, it can be resized and given its own name.

Copying Aliases

If you copy an alias by itself, you will get a new alias of the same original. If you select and alias and its original and make a copy, you will get a copy of the original with its own alias.

Ornaments

You can create an ornament by selecting **New > Ornament.** Ornaments are simply solid backgrounds behind other Environment objects. An ornament never "covers" the objects behind it, and you can still select multiple objects by ***rubber-banding.***

Macros

As mentioned above, macros are not actually Environment objects at all, they are collections of other Environment objects, together with their cabling. To create a new macro, first select all the Environment objects to be included, then select ***New > Macro.*** Logic will enclose all the selected objects in a frame with a gray background. The macro will automatically be named after its input and output objects, separated by a dash (see below).

You can create a "protected" macro by holding ctrl while creating the macro. Protected macros ***can not be unpacked*** so be sure that you make a copy of the objects and cabling before creating a protected macro.

Macros have a lot in common with standard Environment objects-you can connect cables to and from them, they can be resized, they have there own parameters and icon, they can be placed on the Arrange window's Instrument List and they can

be copied or dragged between Environment layers (including between different songs).

Macro Input and Output

Since a macro is a collection of objects, individual objects need to be specified as the macro's input and output. This can be done in two ways: by name or by default. If you name one object "Macro-In" and name another object "Macro-Out" these will automatically become the macro's input and output. If there is no object named "Macro-In" then the upper-left-most object will become the macro's input. If there is no object named "Macro-Out" then the lower-rightmost object will become the macro's output. Cables leading into the macro will deliver their events to the macro's input object, and cables leading from the macro's outlets will carry events leaving the macro's output object.

Macro parameters

Macros have four parameters:

Icon & Instrument List Checkbox

As usual, the checkbox next to the word Icon determines whether the macro appears on the Arrange window's Instrument List. If you use a macro as a track instrument, events played on the track, or incoming MIDI events when the track is selected, will be sent to the macro's input object (see above).

If you place a macro on the Instrument List, you can choose any of the available icons for it, using the pop-up menu as usual.

Auto Fader

If you select a group of faders, and create a macro from them, checking this box will make them act as if they were cabled serially.

Show All

This checkbox determines which objects will remain visible when collected in a macro. When it is checked, all objects will be visible (assuming the macro is not resized to hide some of them). When it is unchecked, only faders, monitors ornaments and keyboard objects will be visible. (You can toggle this check-box at any time.)

Special Macro Features

Unpacking

You can turn any macro back into its component objects and cables, by double-clicking in any empty region of the macro object.

Size Limitations

Macros are limited in size-the limit depends on the memory taken by the individual objects within the macro. Typically this limit is between 100 and 200 objects.

Nesting Macros

You can nest macros-i.e. a macro can contain other macros as objects.

Cables Leading Outside of the Macro

If some of the objects you select when creating a macro have cables leading to objects not selected, these cables will be deleted when the macro is created, and a warning dialog will notify you of this. In this case, the macro is made from a copy of the original selection of objects and the original collection of objects remains unchanged. If no cables lead to unselected objects then the macro is made from the original objects.

5.5 Faders

To create a new fader, select **New > Fader > . . .** A sub-menu appears, where you may select the style of fader you want. There's also a sub-menu at the bottom named *Special,* for selecting various special types of faders (e.g. cable switcher, meta-fader, etc.).

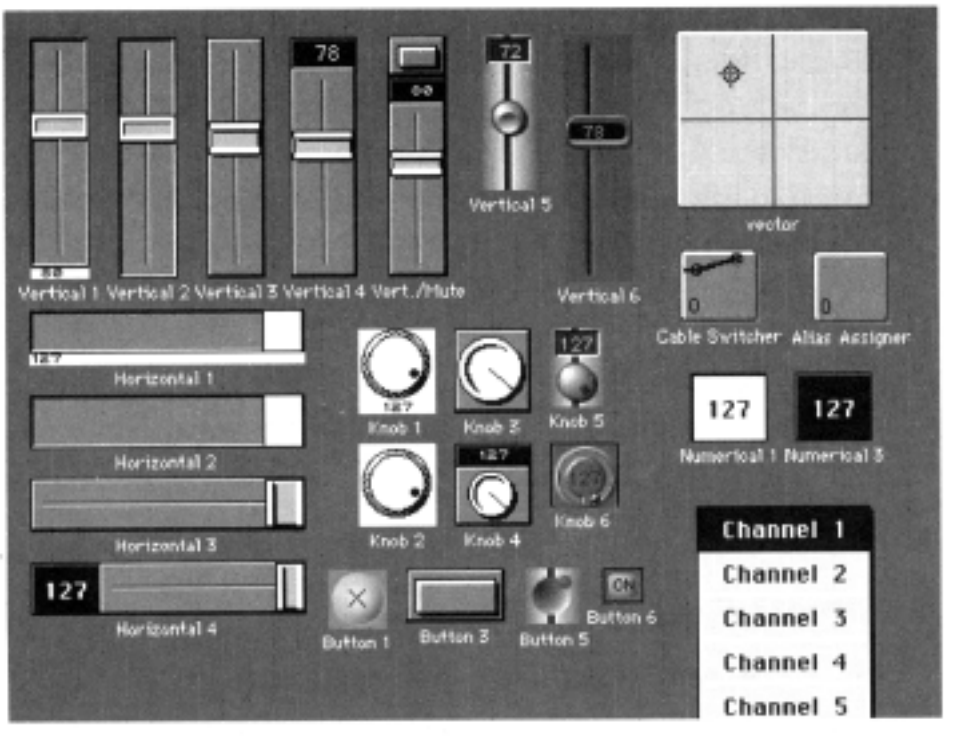
The style of a fader is how it looks on screen (i.e. slider, knob, menu, numerical...) and how it responds to the mouse (i.e. click & drag, double-click & type a number, click & select from a menu...). The *Auto* style fader, which is the default style if you create a fader with a key command, changes styles as you change its shape and size.

The *type* of fader determines what events the fader sends out and responds to. In two cases: the cable switcher and the alias assigner, the fader doesn't send out events at all. The cable switcher routes events to its different outlets, and the alias assigner changes the references of fader aliases.

In general, a fader's style is completely independent of its type. A cable switcher can look like a button, a knob can send out MIDI or meta or Sysex events, etc. The one exception is the Vector style fader-these behave in a special way by sending out two (or in special cases, four) messages at a time, depending on the mouse location within their 2-dimensional, vector window.

Don't worry about selecting the wrong fader style or type from the *New* menu-you have complete control of a fader's style and type from its parameter box.

The illustration below shows examples of each of the fader styles. Notice that some of the styles have numerical displays. When there is a numerical display, you can double-click it to enter exact values by typing. Otherwise, when you scroll any of the sliders or knobs, the series of values that are sent out depends on the fader size and scrolling speed-i.e. not every consecutive value is sent out. (If you click & drag slowly on the numerical you can get all consecutive values.)



The text-style fader on the right, behaves like a pop-up menu. If you double-click it a window will open, where you can type in the names for each of the menu positions. There is also a checkbox so that you can make the names scroll, rather than choosing from the menu.

The vector-style fader (upper right) allows you to scroll in two directions, and it sends out two values, one corresponding to the vertical position and the other to the horizontal position.

The ***cable switcher,*** shown below the vector, routes all incoming events to the selected output. As with most objects, cabling one outlet automatically produces another. Clicking the cable switcher causes it to change to the next output.

Operation

Normally you grab the surface of a fader with the mouse pointer or pencil, and drag it vertically or horizontally, depending on what style it is. Knobs can be dragged either vertically or horizontally.

When there is a numerical display, you can grab this number and use the mouse as a slider. The advantage of this is that if you drag slowly, it will move in increments of 1, regardless of the size of the object. When using the fader itself as a slider, values may jump in larger increments, depending on the size of the fader and speed of dragging. You can also change the values in steps of 1, by dragging on the slider or knob while holding down ctrl.

Buttons send out only two values, the lowest and highest of their range, which you may set in the parameter box. You change a button's state by clicking it.

Working with Object Groups

When building a virtual mixing desk or synthesizer control panel in the Environment, you often have to deal with large groups of fader objects, which have the same size, regular spacing and/or a similar definition. To save time with the definition and alignment of these groups you can define (one or more) objects ***as prototypes (templates)*** by copying them onto the clipboard **(Edit > Copy** or ⌘C). You may then apply certain characteristics of the ***template*** to selected objects.

Size and Alignment

The function **Options > Apply Buffer Template to > Size,** allows you to transfer the size of the prototypes to the selected objects.

To align a group of objects, choose 2 objects in a row, or 2 objects in a column, or 4 objects in 2 rows and 2 columns, whose horizontal and vertical alignment you want to transfer to the group, and copy them into the clipboard. You can then transfer this alignment by selecting the function **Options > Apply Buffer Template to > Position,** and the selected objects will be positioned at the top left corner, in the same way as in the prototype.

Options > Apply Buffer Template to > Position and Size combines both the above functions.

Definition

The function **Options > Apply Buffer Template to > Definition** transfers the parameters of a copied prototype to all selected objects. If several prototypes of the same type are available, the one with the most similar size is used.

...> Definition, channel increment increases the channel number from object to object, beginning with the top left object. It is not necessary for the selected objects to have the same In or Out definition as the prototype.

...> Definition, number increment increases the first data byte of the definition (e.g. controller #).

Cabling serially

Using **Options > Cable serially,** you can connect all selected objects in series, beginning with the object at the top left.

Names with Numbers

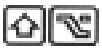
If you select a group of objects then label one of them with a name that ends in a numeral, the remaining objects will be renamed with the same name, but increasing numerals. (E.g. Selecting several objects and naming one of them “Object 1" will result in the names “Object 1", “Object 2", “Object 3", etc.)

Special Functions for Faders

Grouping Faders

If you select several faders and move one of them, all the selected faders will move proportionally. As long as all the faders remain selected, their proportional positions will be retained (even after one or more of the faders has reached its minimum or maximum position).

Moving the fader group with ***Alt*** pressed, changes all values linearly (the absolute value differences are maintained).

Moving the fader group with  pressed, changes all faders to the same value.

Sending Fader Values

You can use the functions **Options > Send All Fader Values** and **...> Send Selected Fader Values** to make all fader objects or all selected fader objects send their current values. Using this function in conjunction with Logic's *Record/Pause* mode, allows you to record a "snapshot" of all current fader positions. This is especially useful for virtual mixing desks, or synthesizer panels. *(Send All Fader Values* only sends values for faders whose outputs are cabled to other objects.) The option *Send All Fader Values after loading* in the **MIDI Options** page of the **Song Settings** causes all fader values to be automatically sent after a song is loaded.

Resetting Fader Values

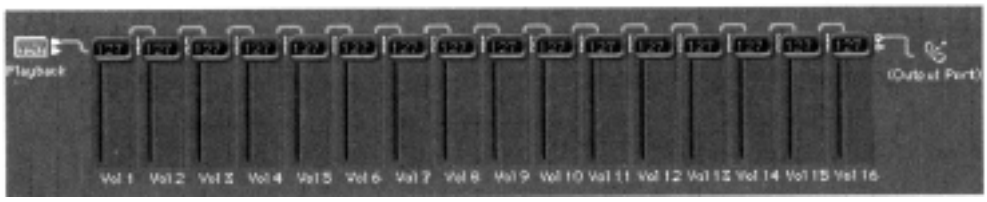
With **Options > Reset Selected Faders** all selected faders are set to zero and transmitted.

Recording Fader Movements

You don't need any special cabling to record the data generated by a fader. All data generated by faders is recorded on the selected track when Logic is in *record or record/pause* mode. The recorded events correspond to the *out definitions* of the faders.

Playing Back Fader Movements

Any fader will react to incoming events matching its *In definition*. The fader must, of course, be in the MIDI signal path. Typically you accomplish this by cabling the track instrument into the fader. For purposes of automated mixing, it is useful to create a new (standard) instrument just for the purpose of automating the faders. Ensure that the instrument (which you might conveniently name *playback*) has no direct MIDI output, that its channel setting is *All* and that it is cabled into the first fader in any chain of faders cabled serially.



Mixer Automation

There are extended automation functions for all faders in the Environment. As with mixing desk automation, there are three modes: Merge, Replace and Update. Using these three entries under **Mixer Rutomation > . . .** , existing fader movements can be altered in real-time.

Merge

Newly-written, data is merged with existing data. (This is the mode used in Logic versions prior to 3.0.)

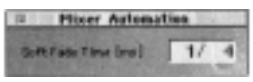
Update

In this mode, only changes in the fader values relative to existing data are recorded.

Replace

In this mode, existing data is replaced by the new data. The data is replaced for as long as the mouse button is held down.

Soft Fade Time



When you release the mouse button after carrying out a replace operation, there is a smooth transition between the current fader value and the recorded value, as determined by the *Soft Fade Time.* You can set this using **Options > Mixer Automation > Soft Fade Time...** This purpose of this is to ensure that there is no sudden jump in level.

Setting the Automation Mode

You can set the automation mode in either the Record Switch menu, or from the Environment **(Options > Mixer Automation >...).**

Correcting the Automation

If you want to change recorded fader values:

Select the sequence(s) containing the recorded fader values. (Other MIDI data in these sequences will not be affected.)

Make sure you have chosen the desired automation mode.

Start Logic recording.

Move the fader(s) whose values you want to change. (Remember that in ***Replace*** mode, the old values will be replaced for as long as the mouse button is held down.

Fader Styles

The fader's style is shown in the line under the icon. It can be changed by clicking on the current style name, and selecting the new style from the resulting flip-menu.

Remember that a fader's style does not usually affect its function-i.e. you can select the most convenient style for the desired application (see page 5-68).

Here are notes on some of the fader styles:

Vertical/Mute

This is exactly like the ***Vertical 4*** fader style, with an added mute button. When you press the mute button, the fader sends an event matching its Out definition with value 0. While the mute button is down, the fader's movements are not sent. When you release the mute button, the current fader value is sent.

Numericals

When a fader has a numerical display, you can double-click the numerical and type in a new value. If you use the numerical for slow scrolling (i.e. click and drag slowly on the numerical), the fader values will change in steps of 1.

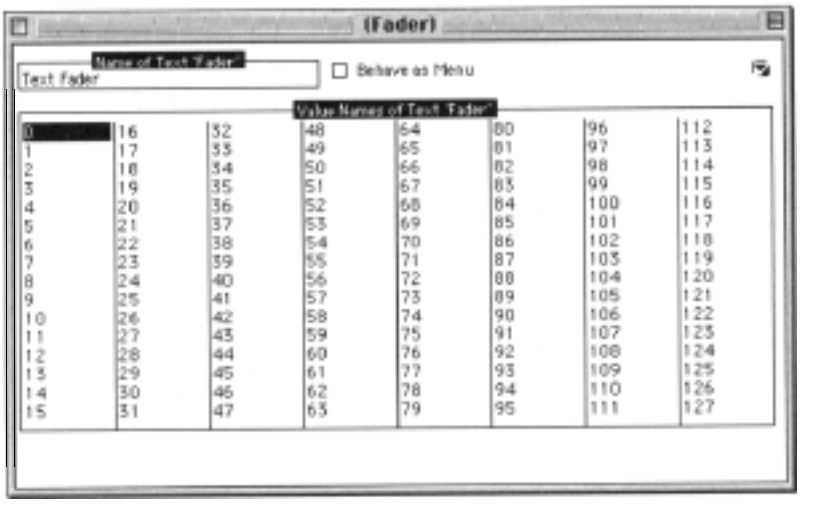
Buttons

Button style faders send only two possible values corresponding to the minimum and maximum of their range. When you click the button on (hollow or X'ed in the illustration), the maximum value is sent, and when you click the button out (filled or not Xed in the illustration) the minimum value is sent. If the minimum and maximum range parameters are set to the same value, then the button sends out this value each time it is clicked.

Text

Text faders function like ***Numetical*** faders, but can display text for each of the 128 possible MIDI values (0 thru 127).

Double-clicking on the surface of a text fader opens the text fader window.



Clicking on a position in this window will cause the corresponding fader value to be sent. (This is similar to selecting programs by name in the multi-instrument window.) Double-click on a position in this window to type in the text for that position. By default (i.e. when you create a text style fader) the text positions contain the numerical values. (You can use this feature to create numerical style faders with colored backgrounds.)

Clipboard Functions

The flip menu located at the top right of the text fader window provides cut, copy, and paste clipboard functions. You can use this to transfer the entire list of text names to a text editor (e.g. a marker text window in Logic), for more convenient editing. Note that empty lines, and lines containing only spaces are ignored, so to insert blank, positions in the list, use  .

Behave as Menu

Check the *Behave as menu* box to cause the text fader act as a flip menu. When this box is not checked, the text fader acts as a scrolling menu.

Text Fader Range

A text fader will only allow you to enter as many names as are in its range. (E.g. if you set a text style fader's range to 0,1 you can only enter two values.) You should always set the minimum range necessary for a text fader, as this saves memory. In any case, remember that the first name corresponds to the lowest value in the range (not necessarily 0) and the last name corresponds to the highest value (not necessarily 127).

If you force a text fader to a value outside its range (i.e. with MIDI input) it will display "---" for values below its range and "+++" for values above. The one exception is when the fader's range is 0,1 in which case all values above 1 will display the name for value 1.

Vector-Style Faders

Vector faders function like joysticks-they can be moved in two dimensions: up/down and right/left. Each dimension generates to its own MIDI events, so that each time you change the position of the crosshair with the mouse, two MIDI events are sent.

Most faders have In and Out definitions (see page 5-78) which determine what MIDI events the fader sends (Out) and reacts to (In). The vector style fader replaces these with *Vert* and *Horz* definitions, which determine what MIDI events correspond to vertical and horizontal motion. (If you cable corresponding MIDI events into the vector fader, it's crosshair display will update accordingly.)

4-Channel vector mode

If you set a vector fader's *Vert* and *Horz* definitions to the same MIDI event (e.g. the same MIDI controller & channel), the vector fader will send out four MIDI events each time the crosshair is moved. These will be the same MIDI event (e.g. controller) on four consecutive MIDI channels, starting with the channel set in the *Vert* definition.

The values of these MIDI events will correspond to the closeness of the crosshair to the vector fader's four corners-upper-left for the lowest channel then upper-right, lower-left and lower-right respectively for the next 3 channels. In the center, all channels receive value 32. At the corners the corresponding channel receives value 127 and all remaining channels receive value 0. (This is with the default range of 0 to 127. If you alter the range, the center and corners will behave differently but mysteriously, the four values will always add up to 12.5).

Fader Functions

MIDI Events

Each fader has an In and an Out definition. The In definition determines the kinds of MIDI events that can remote control the fader (i.e. the types of events it reacts to). The Out definition determines what kinds of MIDI events the fader sends out. (A fader can, therefore, convert one type of MIDI event to another.)

Most MIDI events consist of three bytes. The first one indicates the type and channel of the MIDI event (e.g. a note on channel 3). The second one indicates the first *data value (e.g.* the pitch in the case of note events) and the third one indicates the second *data value (e.g.* velocity in the case of note events). A few MIDI events use only two bytes (program change & aftertouch). Although faders (and transformers) always provide for 3 bytes, the second one is thrown away for these special, 2-byte messages.

Fader parameters are provided for setting the message type, MIDI channel and the first data value. (Note that the message type and MIDI channel are actually combined in the resulting MIDI event.) The second data value is determined by the fader setting, or in the case of remote control, by the incoming MIDI event.

Here are the `In` and `Out` definition parameters:

Out (or `In`): defines the event type (use the flip menu shown in the diagram).

Channel (1-16): defines the MIDI channel of the event.

-1- (0-127): defines the first *data byte* of the event. In some cases like pitch bend this is an actual data value. In other cases like MIDI controllers, this indicates the controller type (e.g. volume, pan, etc.). In still other cases like aftertouch, this byte is unused.

Here is the meaning of the -1- parameter and fader position for each of the In and Out definition choices:

Note On

The -1- parameter sets the pitch and the fader position sets the velocity. This is most useful as an In definition for trapping specific notes and converting them to other MIDI events, or simply monitoring their velocity. (If you move a fader whose Out definition is set to Note On, then a note-off MIDI event will immediately follow the note-on. This might be useful for creating on-screen drum pads from button-style faders, for example.)

P-Press

The -1- parameter sets the pitch and the fader position sets the Poly Pressure amount (i.e. the key pressure).

Control

The -I- parameter sets the MIDI controller number (i.e. the controller type) and the fader position sets the controller value. The controller type can actually be selected by name, using the flip menu that appears when you click and hold the -1- parameter.

Program Change

The -1- parameter is ignored. (In earlier versions of Logic, this sets the program bank.) The fader position determines the program number.

C-Press

The -1- parameter is ignored. The fader position sets the channel pressure (a.k.a. aftertouch) amount.

PitchBd

The -I- parameter sets the pitch bend LSB and the fader position sets the MSB. Typically you would set the -1- parameter to

0 and use the fader to control the “coarse” pitch bend amount. (No pitch bend corresponds to a -1- setting of 0 and a fader position of 64).

The Sysex and *Switcher/Meta* functions are described later (see both the section *Sysex Fader* on page 5-86 and the section *Cable Switchers* on page 5-83).

Range

The range parameter contains two numbers-the left one sets the lowest possible fader value, and the right one sets the highest. (Note that these limits can be exceeded using MIDI remote control.) When the fader style is a button, the range determines the in and out position values of the button.

For text style faders, the first name always corresponds to the low end of the range, and the subsequent names correspond to incrementing range values up to the top of the range. The number of names you can enter into the window is limited by the range.

Val as

This parameter determines the way the fader value is displayed numerically:

Num: the fader value is displayed directly as a number (0-127).

Pan: the fader value 64 is displayed as “0”; smaller values appear as negative numbers and larger values as positive numbers (-64 to 63).

Hz, Oct, dB, ms: These display formats are tailored to various DSP functions.

Bpm: An offset of 50 is added to the fader value. This displays the correct tempo settings for the special *Tempo Control* fader.

If none of the above formats is suitable, consider using a text style fader and entering the desired display values as text. Examples include percentages, note names and program names.

Filter

This parameter provides various filter options for MIDI events reaching the fader:

Off: All incoming MIDI events are allowed to pass through. All events that match the input definition are converted in accordance with the output definition.

Other: All MIDI events which do *not* correspond to the input definition are filtered. All the events that match the input definition are converted in accordance with the output definition, and allowed to pass through.

Match: All MIDI events which match the input definition are filtered, all the others are allowed to pass.

All: All incoming MIDI events are filtered.

Thru: All MIDI events coming from the Physical Input are filtered. This is the same as *off* for all events coming from Logic (i.e. from sequences or the Environment).

Use this filter mode to prevent MIDI feedback by blocking incoming MIDI events from being sent back out.

Shot: When the fader is moved with the mouse, only the final value (the value when the mouse button is released) is sent.

14Bit: Used in conjunction with pitch bend, this allows Z-byte (i.e. fine tuning) pitch bend events. Used with controller messages, this causes the fader to send two MIDI controller messages-one for the MSB and one for the LSB.

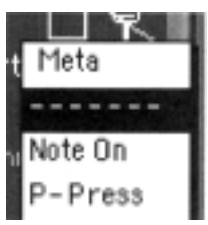
For the 14Bit setting to work, the fader's ***In*** and ***Out*** definitions must be the same. For controllers, the MSB uses the In definition controller number and the LSB uses the controller number 32 higher. This conforms to the MIDI standard for sending 14-bit controller data.

Note that with this filter setting, the fader's range maximum can be set to 16383 and a fader value of 8192 represents no pitch bend.

Feedback

When the ***Feedback*** parameter is unchecked, the fader will automatically prevent feedback loops resulting from circular cabling. (I.e. the fader remembers when a specific MIDI event has passed through it and will not allow it to pass through again.) In some instances, you may want to allow this feedback-for example, to allow a MIDI event to change a ***cable switcher?*** position after it has passed through the switch. Checking the ***Feedback*** checkbox will allow this.

Switchers, Alias Assigners & Meta Faders



These are two types of faders that do not generate MIDI events. They share the same ***Out*** definition type-the one at the top of the definition menu. This selection will either display Switch or ***Meta*** depending on the current -1- value setting:

If the -1- value is 48 then the fader will be a cable switcher.

If the -1- value is 46 then the fader will be an alias assigner

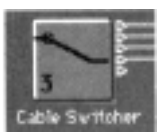
If the -1- value is anything else then the fader will send out meta-messages whose type corresponds to the -1- value.

To create a cable switcher:, either select **New > Fader > Specials > Cable Switcher...** or change an existing fader's ***Out*** definition to ***Switch/Meta,*** and set its -1- value to 48.

To create an ***alias assigner*** either select New > **Fader > Specials > Alias Assigner** or change an existing fader's Out definition to ***Switch/Meta,*** and set its -1- value to 46.

To create a ***Meta*** type fader, either select it from the **New > Fader > Specials >...** menu, or change an existing fader's Out definition to ***Switch/Meta*** and set its -1- value to the desired meta event number (see below).

Cable Switchers



As stated above, cable switchers route events rather than generating them. Any kind of MIDI or meta event can be routed by a cable switcher. The only exception is events which match the cable switcher's In definition. These events will change the switch position (i.e. the routing) rather than passing through the switch themselves.

A cable switcher can have any fader style. In particular, using the text fader style allows you to label the switcher's various routes. Use the Auto-style to actually display the switch routing, as shown here.

A cable switcher can have up to 128 separate outlets-a new outlet is generated each time an existing one is cabled to something. You can click on an auto-style cable switcher to step through the outlets (including the last, uncabled one). Incoming events matching the cable switcher's In definition will change the switch position to match the incoming data value. (If the data value is greater than the number of switch positions, the last, uncabled outlet will be selected.)

Data values of 126 and 127 have a special effect:

An event with a value of 127 increases the output number. If you're on the switcher's last output when this event is received, you jump back to the first output. This is just like clicking on the fader.

An event with a value of 126 decreases the output number. If you're on the switcher's first output when this event is received, you jump forward to the last output.

Meta Event Faders

Meta faders generate special events called ***meta events,*** which are used to control certain Logic functions, but have no MIDI meaning, and are never sent to the MIDI output. In some cases (e.g. ***Goto Screenset, Goto Song,*** etc.), you do not need to cable meta faders to anything for them to work, but even in these cases you can use cabling to process meta events in the Environment, and therefore alter their effect. In the majority of cases (e.g. ***Set fader range minimum, Bang!, Set TransformerOperation Minimum,*** etc.) the meta-fader must be cabled to the object being affected.

Here's a quick summary of the currently implemented meta events that can be generated by faders:

46 Assign Alias

49 Goto Screenset

50 Goto Song

51 Goto marker

52 Stop playback

96 Set fader range minimum

97 Set fader range maximum

98 Set the fader value without sending.

99 Bang! Cause the fader tore-send its current value. The bang events data value can be used to cause the fader to increment-without-rollover (127), decrement-without-rollover (125) increment-with-rollover (123) or decrement-with-rollover (121). Use even values one less than those shown, to have the bang passed through to all connected faders.

100 Tempo Control (see the section section ***The Real-time Tempo Fader*** on page 15-14)

122 Set transformer map value for the currently selected map position.

223 Select transformer map position.

124 Set transformer condition maximum (bottom) parameter, if any. (This applies to all conditions not set to ***ALL)***

125 Set transformer condition minimum (top) parameter, if any. (This applies to all conditions not set to ***All)***.

126 Set transformer operation maximum (bottom) parameter, if any. (This applies to all operations not set to ***Thru).***

127 Set transformer operation minimum (top) parameter, if any. (This applies to all operations not set to ***Thru).***

For more on Meta events 124-127, read the section ***Remote Controlling the Condition and Operation Values*** on page 5-46.

Goto Marker/Screen Set/Song

Selecting **New > Fader > Special > goto Marker** creates a fader which allows you to enter a marker number (Meta event 51), the Song position line then moves instantly to the marker with that number. Markers are numbered sequentially, throughout the song, even if you have renamed them from the default numbers they had when first created.

Selecting **New > Fader > Special > goto Screenset** creates a screenset fader (Meta event 49): you can then switch to the screenset with the number you select using the fader.

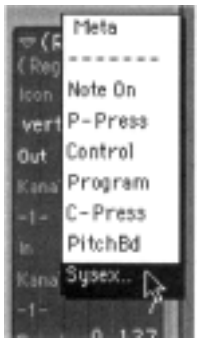
Meta event 50 has no effect within Logic, but you can use it to switch Songs on an external hardware sequencer.

Meta event 52 allows you to interrupt playback at any position.

To make the most efficient use of these faders, we recommend that you limit the value range to suit your particular circum-

stances, for example so that you limit the faders to operate on just the number of screensets, Songs or markers you're actually using.

Sysex Fader



The Sysex fader type (the last selection on the fader definition menu) is slightly different than the other fader types. It allows you to create a list of MIDI events that will be sent whenever the fader is moved or remote controlled. You enter the messages to be sent in a window that resembles Logic's Event editor, except that the time positions serve only to control the order in which the events in the list are sent - they are not sent at specific times, or with any delay between them.

The Sysex fader type was originally intended for sending MIDI System *Exclusive (SysEx)* messages-messages that are exclusive to individual manufacturers' MIDI devices. There are a number of built in features in the Sysex fader's window (see below) that facilitate creating Sysex messages. But, remember that Sysex faders can be used for any kind of MIDI event. and are therefore, very handy for sending ordered batches of messages, like mixer or control panel snapshots with one click of the mouse.

You can also specify Sysex as an In definition, but this has limited usefulness because incoming message (presumably Sysex) must be very short to be recognized. This is an artifact of the fact that Sysex messages can be of any length, and Logic must break them up in to small packets to avoid interrupting other MIDI activities.

Opening the Fader Event Editor window

When you set a fader's Out or In definition to Sysex, its event list window opens up. You can also open this window at any time by double-clicking on the word Sysex in the fader's parameter box.

An important thing to remember about the Sysex fader is that only those events that are selected when the event list window is closed will have their value altered by the fader value. Events which are not selected will be sent exactly as they appear in the window.

Those events that are selected will have the value in the *Val* column of the list altered if they are normal MIDI events (i.e. controllers, program change, aftertouch, etc.). For Sysex messages, you can set both which byte(s) are altered, and the format (e.g. MSB/LSB, BCD, nibbles, etc.) that the value will take. You can also specify a checksum format, if necessary.

Creating a Sysex message

There are two ways of entering Sysex messages in the Sysex fader's event list. One way is to have the fader "learn" the sysex string. To do this, you manually operate your synthesizer to have it send the corresponding Sysex message, using its onboard controls. Here's how:

If you haven't already done so, connect the MIDI output of the synthesizer to the MIDI input of your computer.

Click the MIDI In button in the top left. If the button is red, this means it is ready to receive incoming MIDI data.

Alter the parameter on your device. The corresponding Sysex message will be displayed and the MIDI In button will switch off automatically.

Another way is to type the sysex string into the event list. (Consult your MIDI devices manual for Sysex documentation.) Using this method qualifies you as a confirmed MIDI freak. Here's how:

Click the Sysex button with ⌘ held down. A generic Sysex message appears.

The first data byte in the top line (directly after the word Sysex in the *NUM* column) gives the manufacturer's ID. This may be several data bytes long (since there are more than 128 manufacturers of MIDI devices).

Usually the manufacturer's ID is followed by a device type ID, an individual device ID, a data type ID (which could be several bytes), if necessary the number of the multi-timbral sub channel, the identification number(s) of the sound parameters plus the value of the sound parameter. But there are no standards here-MIDI manufactures can and do adopt arcane schemes, presumably intended to make your life miserable (and protect the keys to the kingdom).

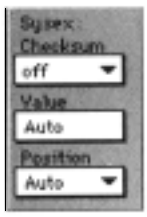
The data bytes are normally input as decimal numbers. To enter them in hex, place a "$" in front of values when you type them in. **View > SysEx in Hex Format** allows you to see all the values in hexadecimal (for more on this, read the section on using Sysex displayed in hex, in Chapter 9).

The last data byte (the *EOX* indicating the end of the Sysex message) is entered automatically by Logic. The number of data bytes in the Sysex message can be decreased or increased by clicking one of the two arrows before and after the word < EOX >.

Although you can have any number of messages of any kind in the Sysex fader3 event list there is only one fader value, and all selected messages will adopt this same value.

You can use a similar method to create any kind of MIDI or meta event in the Sysex fader window. Clicking any of the 8 event type buttons (note, program number, pitch bend, controller, channel pressure, poly pressure, Sysex and meta event) with held down creates a new event of that type. (Meta events are created using the expanded view button with the 0's and 1's on it).

Setting the Sysex Data format



The terms "SUM" for the checksum and "VAL" for the fader value that's being sent are displayed within the Sysex string. The position and format of these bytes are set from the "Checksum", *"Value"* and *"Position"* flip menus. These settings may be found in the lower left-hand corner of the fader event list.

Only *selected Sysex* events are affected by the above parameters, because only these events are affected by the fader value.

Checksum format

A checksum can be created in any of the following formats:

Roland
Yamaha
Regular Checksum
2's complement
l's complement

If you don't know which one works with your MIDI device, try "off" (= no checksum) first or "2's complement".

Value byte position

Position allows you to determine the position of the value byte. This position is specified in bytes counting from the end of the message: "last" refers to the position directly before the *EOX* byte, "Last-1" indicates the byte before that, etc.

"Auto" ensures that the value byte is inserted last in the Sysex string if no checksum has been selected, or -if a checksum value *has* been entered-that the value is inserted as the second to last byte.

Value byte format

Value can be set to the following:

Auto	The value is sent as one byte if the value range maximum is 127 or less. If the maximum is higher than 127, the value is sent as two bytes, the MSB (most significant byte) first.
One Byte	The value is sent as one byte.
MSB/LSB	The value is sent in two bytes, with the MSB (most significant byte) first.
LSB/MSB	The value is sent in two bytes, with the LSB (least significant byte) first.
BCD 4 LSB	The value is sent as “Binary Coded Decimal” in four bytes, in the order 1,10,100,1000.
BCD 4 MSB	The value is sent as “Binary Coded Decimal” in four bytes, in the order 1000,100,10,1.
2 Nibbles L	The value is sent in two nibbles, with the “least significant nibble” first.
3 Nibbles L	The value is sent in three nibbles, with the “least significant nibble” first.
4 Nibbles L	The value is sent in four nibbles, with the “least significant nibble” first.
2 Nibbles M	The value is sent in two nibbles, with the “most significant nibble” first.
3 Nibbles M	The value is sent in three nibbles, with the “most significant nibble” first.
4 Nibbles M	The value is sent in four nibbles, with the “most significant nibble” first.
2 ASCII M	The value is sent in two nibbles, the “most significant nibble” first; the nibbles are sent in ASCII format for the hex value. For example, the value $7F (=127 in decimal) will be transmitted as a “7” and "F".

3 ASCII M same as 2 *ASCII M* but in 3 nibbles.

4 *ASCII M* same as 2 *ASCII M* but in 4 nibbles.

The unused bits of the transfer in nibbles (X in 0XXXNNNN) are sent with the information at the relevant positions of the Sysex strings. If you want to transfer these deleted bits, you will have to enter zeros in the Sysex string.

Transforming Sysex messages

Logic only allows one Sysex string per fader. To transform one Sysex message into another, please use the following method:

Create a fader with the Sysex message you want transformed as the input definition. Select any controller you like as the output definition.

Create a second fader with the same controller as the input definition. Make the transformed Sysex message the output definition.

Connect the first fader to the second.

Remember that the incoming Sysex message must be very short for this to work properly. This is an artifact of the fact that Sysex messages can be of any length and Logic must break them up in to small packets to avoid interrupting other MIDI activities.

Meta Events for Faders

Meta are only used internally by Logic-they are not part of the MIDI signal flow, and have no MIDI meaning. But, they do have a similar form as MIDI events, and they are processed in the Environment.

Meta events serve several purposes in Logic. They are utilized extensively for formatting in the Score editor, and if you look at a scored sequence in the Event editor, you will see many meta events interspersed with the MIDI events that comprise the notes and controller messages.

Meta events are also used to affect the operation of faders and transformers in the Environment (see page 5-84). They can also be generated by faders and processed by transformers. The

only restriction is that meta events intended to affect transformers can not be modified by transformers. (Transformers can, however, convert MIDI events or other meta events in to ones that affect transformers, without being affected themselves.)

5.6 Environment Exchange

One of the main reasons for the Environment is to customize Logic to your MIDI studio. This can present a problem when sharing songs with other musicians, using different studio setups. It also presents a problem when you return to older songs after you've changed your studio. Logic offers several functions to make these changes as easy as possible.

Whenever you want to exchange Environments between songs, there is a source song containing the desired Environment, and a *destination* song whose Environment you want to change. The destination Environment must be in memory, and must be the active song (i.e. one of its windows must be active). The source song can also be in memory or it can be a file on your hard drive (or any other medium like CD, Zip, floppy, etc.).

If there are two songs in memory, Logic will assume the active song is the destination, and the other song is the source. If there are more than two songs in memory, Logic will assume the active song is the destination and the most recently active of the other songs is the source. If there is only one song in memory, Logic will present an *Open* dialog, for you to select the source song when you try to import an Environment.

Here are some tips on how to accomplish specific tasks:

Importing Single-purpose Environments

A single-purpose Environment *patch* might be an editor for a specific piece of MIDI equipment or an Environment patch for single MIDI processing task like a MIDI LFO, or complex

arpeggiator/delay line configuration. If the Environment patch is contained on a single layer, the simplest solution is to select **Options > Import Environment > Layer...,** and then select the desired layer in the dialog that appears. This layer and all of its objects will be inserted in the destination song, at the same layer position (i.e. the same place on the Layer's pop-up menu) as it occupied in the source song. Note that this inserts a new layer into the destination song-its original layers will be shifted as needed.

> You can also move selections of Environment objects together with their cabling between songs by dragging or copy & pasting. This is often even simpler if you first collect the objects into a macro.

If the Environment patch you wish to import is complex and spread across several Environment layers, then you should use the merge feature by selecting **Options > Import Environment > Merge.** In this case, all Environment objects from the source song will be added to the Environment of the destination song. One thing to watch out for here is that the merged objects will be placed on the same layer as they came from, which can create a mess if there are already objects on that layer in the destination song. To avoid this, create blank layers in the destination song at the same positions as the layers to be brought in, before importing.

Something to be careful of in both cases, is the treatment of unique objects in the Environment like the Physical Input and Sequencer Input objects. If you import these objects, they will replace their counterparts in the destination Environment AND ANY CABLES LEADING INTO THEM WILL BE LOST. It is best to temporarily delete these from the source Environment, before importing.

Another thing to be aware of is that when you import an object, all objects cabled from the imported object are also imported.

Replacing an Older Environment with a Current One

If you have only added things to the Environment, but not deleted anything, then you can update older songs by choosing **Options > Import Environment > Update.** Only the new objects will be imported-the older objects along with their cabling and track assignments will remain in place.

Swapping Environments

In the complex situation where you want to swap the Environment for one MIDI setup for the Environment for another, Logic offers three options; ***Replace by Port MIDI/Channel, Replace by Name*** and ***Total Replace.*** But be aware that this complex task involves a lot of guess work on Logic's part and the results almost always require some manual fine tuning on your part.

Replace by Port/MIDI Channel

This option replaces all objects in the destination song for which there are objects in the source song addressing the same port and MIDI channel.

Replace by Name

This option replaces all objects in the destination song for which there are objects with the same name in the source song. This affords you some degree of control, by renaming objects in the source song accordingly.

Total Replace

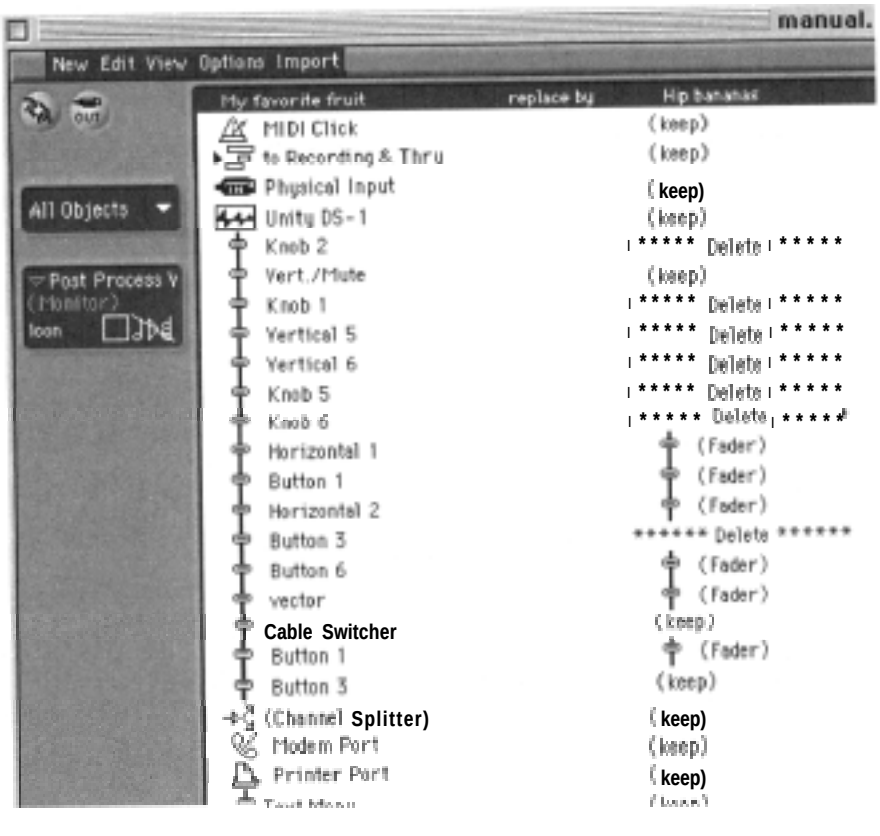
This completely replaces the destination song's Environment with the source song's Environment. If you use this option, you will have plenty of work left reassigning objects to Arrange window tracks, but sometimes it's the only way.

Replacement by Assignment

The most flexible and also most time consuming method of exchanging Environments is to indicate for each Environment

object whether it is kept, deleted or replaced and if it is replaced, what object replaces it. You do this by selecting **Options > Import Environment > Custom...**

When you select custom import, an Environment window appears listing all objects in the destination Environment in a column on the left, and listing the chosen action for each object in a column on the right.



You can assign the replacements one at a time, by clicking on items in the right hand list and selecting the desired replacement object from a pop-up menu. This menu includes all objects on the Instrument List of the source song.

Or, you can select one or more rows of the list, and then choose what to do from the Import menu. In addition to keep and delete, this menu offers several automated selection techniques described below.

The first selection on the Import menu is *Import Environment using current Assignment. No* importing takes place until you

make this selection. The other choices (including the pop-menus in the right column), only set up how things will be imported.

Import Options

The last two selections on the Import menu provide additional options in the importing process:

Import > Copy layer Names: the layer names of the second Environment are transferred to the current one.

Import > Copy selected Objects from 2nd Environment all objects which are selected in the source Environment are copied into the destination Environment, even if they don't appear on the assignment list.

Automatic Assignment Functions

The assignment functions described below are immediately applied to all selected rows in the import list. The assignments appear in the right column and all rows are then deselected.

Import > Assign as 'Keep': these objects are unaltered.

Import > Assign as 'Delete': these objects are deleted.

Import > Assign by Identical: these objects are assigned objects from the source song which match the type, icon, name, port and MIDI channel. Objects with the same Unique ID have priority.

Logic maintains an internal list of all Environment objects. When objects are deleted, their position in this list becomes available for a new object. Whenever an object is added to the Environment, it is put in the first available position on the list. (If there are no empty positions, it is added to the end of the list.) An object's position in the list is its Unique ID - as long as the object is not deleted, its Unique ID never changes.

Import > Assign by Unique ID: objects are assigned objects from the source song with the same Unique ID. (This option is

useful for updating an Environment when objects have only been added.)

Import > Assign by Port/MIDI Channel: objects are replaced by objects with the same port/MIDI channel. If a match can not be found for an object, an object set to port 0 and the same MIDI channel is assigned. If a match still can not be found, an object with the same MIDI channel is assigned.

import > Assign by Name: objects are replaced by objects with the most similar name. (At least 80% of the name must be the same.)

Import > Assign by Icon/Name: same as above except the objects' icons must also be the same.

5.7 The Environment and OMS

What is OMS?

OMS, or the "Open Music System" (or as it used to be known: "Opcode MIDI System"), is a System extension that acts as a logical interface between application programs and the physical MIDI ports of your Macintosh.

To work in this way, OMS lets you define so-called "nodes", which represent the individual items in your MIDI setup. These nodes have a name and information about the MIDI device they represent: where it is connected, what MIDI channel(s) it receives on, and so on. You create the OMS nodes with the OMS setup program, and from there can then edit or erase them. For more information, see your OMS manual (although there are some hints to follow in this manual, in the section ***Creating an OMS Setup for Logic*** on page 5-101).

The OMS manual sometimes refers to nodes as "instruments", but as this word has its own meaning in Logic, we will be referring to the virtual objects that you create in the OMS setup program as nodes-and only nodes.

When you are using OMS as a MIDI driver system, Logic is not aware of your MIDI interfaces, or the physical MIDI ports. All information concerning these is hidden in the OMS system, and cannot be accessed by applications like Logic. Instead, OMS gives you access to the above-mentioned nodes to communicate, and ensures that the flow of data is properly managed and maintained, if for example, strings of data need to be merged.

As with Emagic's Autolink (the connection between Logic and the universal editor/librarian SoundDiver), OMS also lets you run several programs at once.

When Should You Use OMS?

Logic offers very extensive support for OMS (from OMS version 1,2,3 onwards).

Nevertheless, you should only use OMS in Logic if you plan to run other OMS applications at the same time as Logic- for example Opcode's editor/librarian "Galaxy", or when you are using a multi-port MIDI interface which is not able to run in MTP emulation mode.

"MTP" stands for Mark of the Unicorn's MIDI ***Time Piece.*** Please note that you do NOT need to use OMS in Logic to run Opcode's Studio 4 or Studio 5 MIDI interface, Both of them can emulate an MTP (Mark of the Unicorn's MIDI ***Time Piece)*** without compromising on functionality at all. Only if you want to stripe analog tape with SMPTE using one of these interfaces, you will need the OMS setup program.

You can switch between using and not using OMS in Logic's Communication Preferences. It doesn't hurt to keep OMS installed on your system (i.e. as an active extension), whether or not you're using it for Logic-if you're not using it, it remains completely passive.

Please note that you do NOT need OMS to run an Opcode MIDI interface. In fact, you could run either a Studio 4 or Studio 5 in MTP emulation mode without compromising on functionality at all. However, if you want to stripe analog tape with

SMPTE using an Opcode interface, then you WILL need OMS.

Please note that because OMS adds an additional layer of hierarchy between Logic and the MIDI ports:

- Logic's MIDI timing under OMS will not be as good as with its own MIDI driver.
- Logic Environments can be more difficult to design and use under OMS.

Logic under OMS

To use OMS with Logic, you first need to install and set up OMS. Consult your OMS manual for further instructions.

Once OMS is installed on your system, you tell Logic to use it by checking the Use ***OMS if available*** checkbox in the **Preferences > Communication** window. Thereafter, whenever you start Logic, OMS will be used instead of Logic's own MIDI driver (unless you later uncheck the Use ***OMS if available*** checkbox).

You do not have to re-start Logic to activate and de-activate OMS. Each time you uncheck and re-check either the Modem Port or Printer Port checkbox, Logic will check its OMS status, and re-connect accordingly.

Assigning OMS nodes to Logic's inputs

You connect OMS nodes to Logic's inputs on the **Preferences > OMS Input Connections** page.

These connections are necessary if Logic is to receive MIDI events (from external keyboards, for example).

If the option ***Connect all Real OMS Input Nodes with Logic*** is active, all the information from the real, physical OMS nodes will be received by Logic. Normally, this means all the information that is received at the inputs of your MIDI interface.

If this option is switched off, Logic will only receive information from nodes that have been selected via the input flip

menus. If none of these have been assigned, Logic will be unable to receive any MIDI information at all.

You can select eight more OMS nodes from here to input data to Logic's inputs, but normally this will not be necessary if the **Connect all *Real OMS Input Nodes with Logic*** option is active.

The selection of OMS nodes for connection to the input becomes important when you wish to connect two simultaneously active programs via virtual OMS nodes. This is known as IAC-inter-application communication. (You can connect a node to more than one destination-superfluous connections are ignored.)

When you turn Connect all Real OMS ***Input Nodes*** option on or off, you'll have to switch OMS briefly off, then back on again via **Preferences > Communication,** This breaks the previous input connections, and establishes the new ones.

Input connections to OMS slow the computer down. You can speed things up - particularily if you have a slow Macintosh-by only making the connections you are really going to need.

Assigning Instruments to OMS nodes

You can assign any instrument's output directly to any OMS node. When OMS is active, there is a pop-up menu between the icon and the Cha parameter for selecting this assignment. If no OMS node has been assigned ***(OMS Map used)*** will be displayed here.

If you have not made a direct OMS node assignment, you can use the instrument's ***Cha*** parameter as usual to select an output port ***(MU thru M63 PO thru P63).*** But, be aware that these selections have a different meaning under OMS-they assign OMS nodes to the instrument via ***OMS Output Maps,*** which you set up in the Preferences window.

OMS Output Maps

Logic allows you to set up ***OMS Output Maps*** for both the printer and modem ports of your computer. (Of course you need multi-

port MIDI interfaces connected to these ports, for this to be useful.)

You create the OMS Output Maps using the **Preferences > OMS Output Map** pages. There are separate pages for the modem and printer ports, and each one contains pop-up menus for ports 0 thru 42. (The remaining instrument ports, 43 thru 63, are not used under OMS.) The assignments you set up in this table are stored in the Preferences, and are therefore shared by all songs.

There's a good reason for using OMS Output Maps instead of assigning instruments directly to OMS nodes. This allows you to switch between using and not using OMS without any further Environment adjustment. When OMS is in use, the map connects instruments to the correct OMS node and when OMS is not in use, the standard ***Cha*** port assignment selects the same MIDI ports directly.

Creating an OMS Setup for Logic

If you have not done so already, here is one logical way to create and use an OMS setup for Logic. Once you've created the Setup, everything becomes really quite simple:

> This method describes a setup for older Macintosh computers, which still have a modem and printer port. The procedure for newer computers with a USB port should be similar-just use the appropriate names to label the OMS nodes.

1. Start your OMS Setup Program.
2. Create a node for each of the physical outputs your system has and name them in the same way that Logic does - "Ml", "M2", . . . and/or "P1", "P2", etc.
3. Exit the OMS setup program. This OMS setup should now never be changed, unless you add another MIDI interface to your system. Of course, you can add OMS instruments at any time-but for work with Logic, that shouldn't be necessary. Just create virtual representations of new synths in the Environment in the usual way.
4. Boot up Logic, and activate OMS (via **Preferences > Communication),** if you haven't already. OMS must be up

and running as your MIDI driver.

5. Select **Preferences > OMS Output Map** for the desired output port (modem or printer).

6. Assign each OMS node to the corresponding output; for example, Modem 1 should be assigned to the "Ml" node in the flip menu.

7. Quit Logic, so that the Logic Preferences file updates itself (or select **Options > Save Preferences** from the Key Commands window).

That's it! Now you should be able to play any of your old songs and they should sound identical, with or without OMS.

Logic as an OMS node

You can create a virtual OMS node for Logic by checking the ***Create virtual IAC OMS Instrument for Logic*** checkbox in **Preferences > OMS Input Connections.**

Logic creates a virtual node in the OMS ***IAC driver.*** Double-click the driver in the OMS Setup window to assign the node a suggestive name like "Logic". The node will be visible to connect cables to, in OMS applications which support IAC (inter-application communication).

Please remember that Logic can transmit all MIDI data (track events, MTC or MIDI clock, for example) to all OMS nodes, even virtual ones representing other programs which support IAC.

Points to Note when Working Under OMS

MIDI Clock/SPP and MTC Output

You can make any OMS node an output for MIDI Clock/SPP (Song Position Pointers) or MTC. Just make the necessary settings via **Song Settings > MIDI Options.**

Rectangular Output Symbols

Environment objects with direct OMS assignments have rectangular outlet symbols instead of the usual triangles. However, if the OMS Map is used, the outlets remain as triangles.

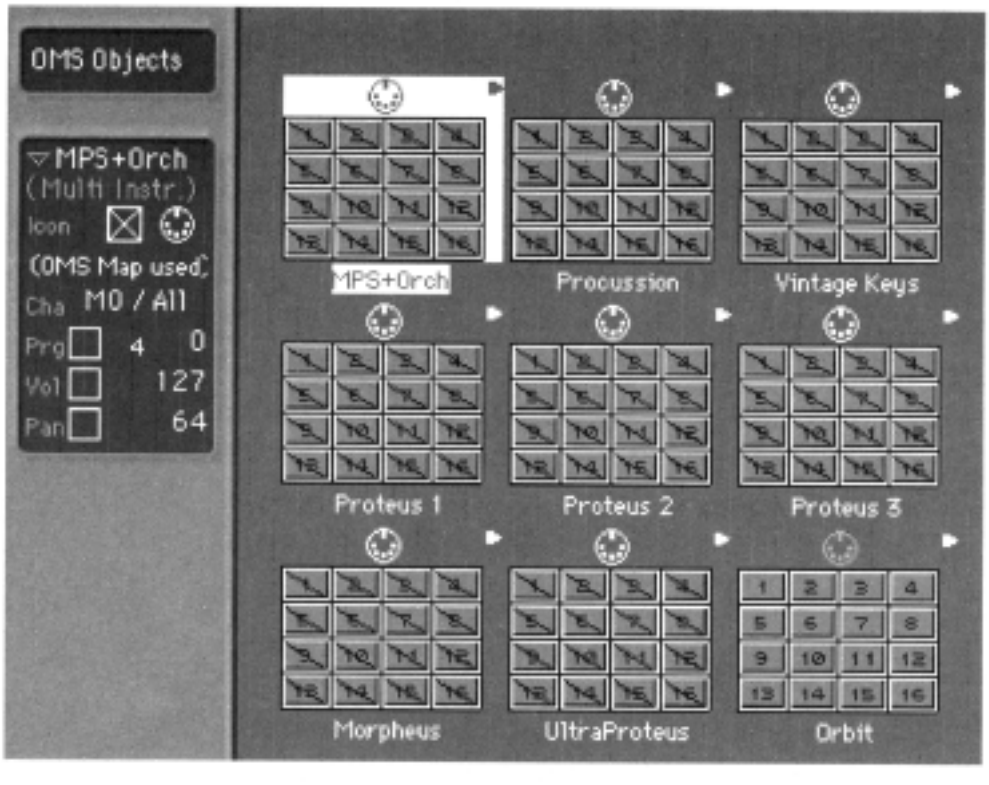
Initializing OMS settings

Select **Options > Initialize > just OMS settings** from the Key Commands window.

This initializes the OMS Output Map, all the input connections, and the settings for MIDI Clock, MTC and the CBX Ports.

Updating Logic to Your OMS Setup

If you already have OMS set up, or you make changes to your OMS setup, you can select **Options > Update > OMS Equivalents,** or use the associated key command to create Environment instruments for each of your OMS nodes. If you selected OMS's multi-timbral option when creating the node, Logic will create a multi-instrument for that node. (So, when making your OMS setup, de-select the multi-timbral option if you want regular instruments created by this process.)



The *OMS Objects* layer might look something like the above picture.

About the Created OMS Environment Objects

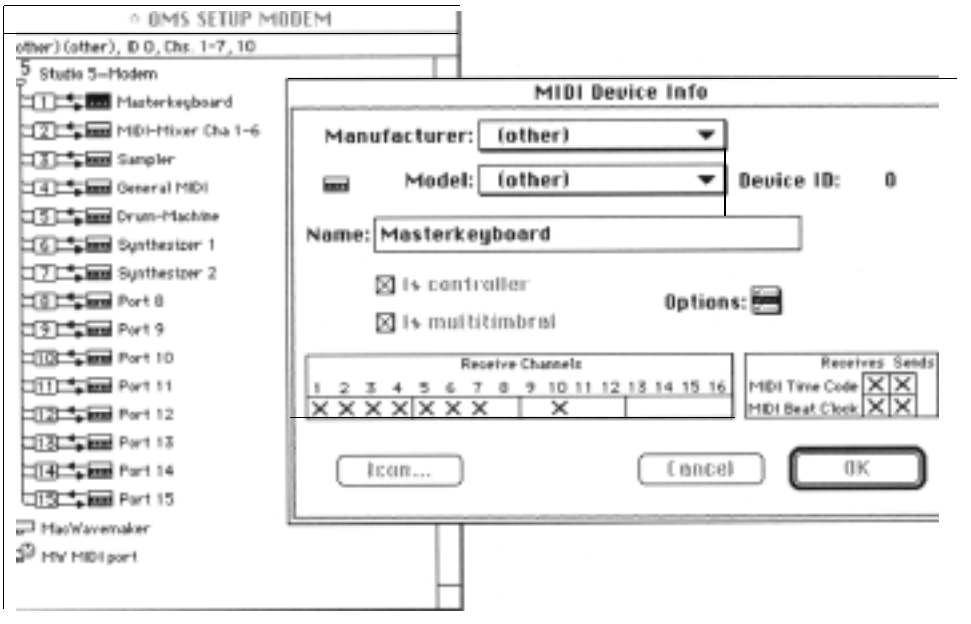
The following settings are taken from the OMS studio setup:

The names of the OMS instruments,
The number and assignment of the ports

In the example shown in the diagram, there are 15 instruments on Opcode Studio 5 ports and one Macintosh MIDI card with OMS driver.

The channels of the multi-mode instruments (as activated sub-channels of multi-instruments).

The OMS settings for the Environment layer shown above are as follows:



On the left is the OMS setup. This is where all the MIDI ports and internal MIDI cards are displayed. All external MIDI devices are connected to the ports by cables.

The right window (MIDI Device Info) is for making settings for the individual MIDI devices. Keep in mind that the receive channels which are active here are “translated” into the active sub-channels of the multi-instrument.

5.8 Tips and Comments

Here are a few helpful tips for creating your own Environments. For more detailed examples of Environment processes and how to create them, see the *Beginner? Guide to the Environment* PDF file included on your Logic CD-ROM.

Working with Notes

One of the most common applications of the Environment, beyond controlling your MIDI studio, is processing MIDI note events. This includes such things as transposition, scale correction &filtering, arpeggiation, echo & delay, voice doubling, etc. Some of these processes have Environment objects dedicated

to them, while others require patching several Environment objects together. In either case, it is helpful to keep in mind how MIDI handles notes, and what information they contain.

How to Avoid Hanging Notes

MIDI notes come in two parts: a note-on event and a note-off event. For live MIDI, there is an *indeterminate* time between these events (i.e. you don't know when the player is going to release the note). For notes recorded in sequences, the note length is known, and for this reason it is not uncommon for MIDI sequencers to combine a note-on and its matching note-off into a single event-as a note and its duration. Logic records notes this way and it is a good idea to keep this in mind when processing notes in the Environment.

With live MIDI input it's not hard to leave hanging notes. This can happen when you change a note-on event without making a similar change to the corresponding note-off event. For notes recorded in sequences this is not a problem, because notes are single events-Logic takes care of making the matching changes when it sends out the note-on and the note-off. The upshot is that processes that work fine when sequences are played through them, might leave hanging notes when live MIDI is processed. Logic tries to prevent this by not allowing some kinds of changes to note events (e.g. random pitch, changing velocity to 0 and changing 0 velocity to something else), but you can work around these limitations, and its up to you to bear the consequences.

What's in a Note

Each MIDI note event contains 3 pieces of information: channel (1-16), note number (0-127) and velocity (0-127). (Note that in many places-transformer, instrument and fader parameters, for example-the MIDI channel is displayed using numerals 1-16.) Channel is often associated with the type of sound the note will produce, because it is typically used to select the MIDI device that will play the note (the MIDI channel in the case of

multi-channel MIDI devices). Note number is most often associated with pitch, but for drum-mode MIDI instruments such as drum synths, it determines what sound is played. When used for pitch, note number 60 typically corresponds to middle C (C3). Velocity is generally associated with accent, as it is often applied to a synth preset's volume or filter cutoff parameters (or both). Do not confuse velocity with volume, however-a preset's overall output volume is typically set by MIDI controller #7.

> Note-on events with velocity 0 are interpreted as note-off events, and this is the only form of note-off event some older synths recognize. (This means that the note-on velocity range is really 1-127). Although Logic will record and play back true note-off events with their correct velocity, you can not change, use or even see note-off velocities in the Environment (or the Event editor). For Environment purposes, consider note-off events to be note-on events with velocity 0.

You have (almost) complete freedom within the Environment to change any of these three pieces of information: channel, note number and velocity. You could for example, use a note's number (pitch) and/or velocity to influence what channel it goes to (i.e. what sound it plays). Conversely, you could use its channel to influence its pitch and/or velocity (e.g. transpose and accent notes on channel 3). The things you are prevented from doing include randomizing pitch, and changing note-on events to note-off events or visa versa. The purpose of these restrictions is to prevent the occurance of hanging notes, as described above. (With a little clever Environment plumbing you can work around both restrictions.)

Where to Insert Environment Processes

The following applies to Environment process that are intended to modify MIDI data passing through them. This includes processes that involve a single Environment object (e.g. an arpeggiator or a delay line), as well as complex processes that combine many objects. For complex processes it is a good idea to start and end the process with a monitor object. This allows you both to see what the process is doing, and to have an easily identifiable entry and exit point. (If you intend to

collect the process into a macro, name these monitors "Macro-In" and "Macro-Out".)

The three most common positions to insert an Environment process are: between the physical input object and the sequencer input object; before an instrument object; or between any instrument object and an output port. In the third case, remember to remove the instrument's direct output assignment (see page 5-13).

The *Clicks & Ports* layer of Logic's default song (see illustration on page 5-3) contains an example where objects are inserted in the cabling between the physical input and the sequencer input. The keyboard object shows note activity; the two monitor objects show the data passing through them before and after events and the cable switcher switches between: no process, an *octave doubler* and a velocity *enhancer*. Processing applied at this point will affect all incoming MIDI data, and will be recorded when Logic is recording. Note that processing of this sort is not easy to undo-you would have to apply a complimentary process to any recorded sequences.

If you want an Environment process to affect data going to a specific MIDI device, but you don't wish to record the process, simply assign the process's input object to an Arrange window track, and cable its output object to the Environment instrument or port object serving the desired MIDI device. Although this is the simplest solution, it has the disadvantage of not making the instrument parameters available when you select the track. If you wish to have these available, use the method below.

If you wish to process data recorded in sequences (and have the instrument parameters available from the Arrange track), then the third location-between an instrument and a port object-is the best choice. In this case the process can be switched in and out at will without making any permanent change to the data. In case you want to make the changes permanent, cable the output of the Environment process into the sequencer

input instead of the port object; record on a track assigned to "No Output", then reassign the track to the port object.

Creating Control Panels

Many Environment processes involve on-screen control panels. Typical examples include simulated mixing desks, synth editors and analog style sequencers. In some cases you may want to automate, or at least take snapshots of the fader positions. For this you need some way of recording the fader positions, and of playing back the results through the controls. Here are two rules of thumb to make this simpler:

Cable the control panel faders in series.
Ensure that each fader's **In** and **Out** definitions match.

With the faders cabled in series, you can either assign the first one to a track, or cable a track instrument into it, so that data played on that track will control the faders. For this to work, a fader's In and Out definition must be the same, because the Out definition is what is automatically recorded by Logic when the fader is moved with the mouse, and the fader will only react to these events if they match its In definition. Keep in mind that each of the faders in the series must be unique in some way - i.e. it must have a different MIDI channel, or MIDI controller number than any of the other faders.

If you want to convert some set of recorded fader positions to a snapshot, open the Event list editor, then copy & paste the MIDI controller messages into a Sysex fader (see page 5-86). Cable the output of the Sysex fader into the first control panel fader in the series. Ensure that none of the events is selected when you close the Sysex fader window. Then clicking the Sysex fader (button styles work best here) will set all the controls to the snapshot values.