

Chapter 8

MIDI Manager and Custom MIDI

Selecting between the MIDI Manager and Custom MIDI

The Macintosh version of HMSL can use either the Apple MIDI Manager and Driver, or a custom MIDI driver. The default is the Apple MIDI Manager. This must be installed as described in chapter one. To use the custom driver instead, enter the following:

```
HMSL . TERM
USE . CUSTOM . MIDI
HMSL . INIT
```

If you would like to make the custom driver the default, do the following:

```
Run HMSL by double clicking on the icon.
Answer 'n' for no when asked to initialize.
Enter: USE . CUSTOM . MIDI
Enter: SAVE-FORTH
```

In general, you will not notice any differences between the two systems, except for those described below.

To get back to using the MIDI Manager, enter:

```
HMSL . TERM
USE . MIDI . MANAGER
HMSL . INIT
```

Comparing the MIDI Manager and Custom MIDI

The Apple MIDI Manager is a tool provided by Apple that allows multiple applications to share the serial ports without interfering with each other. An application called the PatchBay allows you to connect one MIDI manager based application to another or the the MIDI ports. The Apple MIDI Manager provides MIDI Time Code sync and generation for use with SMPTE. The Apple MIDI Manager provides time stamping for output packets, which is used in HMSL as an event buffer. Unfortunately, this buffer is quite small (32 note on/off). The MIDI manager can also be slow. Some users have reported having chords play as arpeggios. Since Apple provides this code, compatibility is "guaranteed" across the Apple family.

The Custom MIDI Driver was adapted from an older version of HMSL. It uses a custom Event Buffer that can also schedule events other than MIDI. It is used, for example, to support the wakeup buffer for the multitasking system, and could be used for DSP control. The custom driver seems to be faster than the Apple code despite being coded primarily in high level Forth! Here is a table that summarizes the differences between the two systems.

Item	Apple	Custom
speed of execution	Slow	Fast

#notes that can be buffered	32	1000+
full source provided	No	Yes
Apple blessed	Yes	No
SMPTE and MIDI Time Code	Yes	No, not yet
patchable between apps	Yes	No
mixed Events in Buffer	No	Yes

Each system has advantages, if one of them does not work for your application, the other probably will.

MIDI Manager Problems

Output Buffer Fills Up - MIDIM.SEND

The MIDI Manager supports time stamping on output. This means that you can send a MIDI Packet to the MIDI Manager with a time for it to be output. The MIDI manager will hold onto it until that time, then output it.

There is a limited space to hold these output events. The Apple MIDI Driver only has a 4K buffer which is enough for about 128 notes. If you schedule too many notes in advance, the buffer can fill up. If this happens, HMSL will wait for a while then try again. If HMSL can't output MIDI, it will eventually abort.

This can occur if you try to schedule too far in advance, like an entire piece worth of notes some time next week. This is usually from a bug in your code. This can also occur if you are outputting notes at a high rate and have TIME-ADVANCE set too high.

The only way to unclog the buffer is to trick the MIDI manager into thinking it is now the distant future by entering:

```
MIDI .UNCLOG
```

You may also want to lower the TIME-ADVANCE to prevent this from happening in the future. (See TIME & Scheduling.)

```
10 TIME-ADVANCE !
```

Serial Ports in Use

There is a bug in Chooser that can cause the phone port to look like it is in use when it is not. This can prevent any use of that port for MIDI. To forcibly free the Phone Port (A), enter:

```
MIDI .FIX .SPCONFIG
```

The printer port is often in use by AppleTalk. You can turn off AppleTalk in the Chooser if you want to use that port for MIDI instead.

Using Both Ports

You can select between the phone and the printer port for MIDI. To switch ports, set the variable MIDI-PORT to 0 or 1. For example:

```
0 MIDI-PORT ! \ use phone port
MIDI .SEQOUT
1 MIDI-PORT ! \ use printer port
MIDI .SEQOUT
```

HMSL Timing and SMPTE

HMSL derives its timing from the TIMER1 hardware clock. This is used by Macintosh local sound. For this reason, local sound cannot be used with HMSL. HMSL will automatically turn down the volume when it runs. HMSL drives the MIDI Drivers clock using the values from this hardware clock. The Application Sync bit must be set using the Patchbay Desk Accessory for the timing to work.

If you want to use the MIDI Manager timing, enter:

```
USE.MIDI.TIMING
```

This will give you a 1000 ticks per second clock. You can then use the PatchBay to select whether to use Internal Timing or to use External MIDI Time Code. Time Code can be generated from SMPTE on tape using specialized hardware.

Further Information

For more information, see the release notes that came with HMSL. If you want to examine the source code, look in:

```
HH:H4th_MIDI+RTC  
HH:H4th_MIDI_IO   = Midi Manager Interface  
HH:H4th_CUSTOM_MIDI  
H:MIDI           = host independant code
```