Chapter 30

MIDI Input Tools

The Humdrum Toolkit provides two tools for inputting MIDI data. In this chapter we briefly introduce the **record** and **encode** commands. These tools provide ways for capturing MIDI input and translating them to representations that conform to the Humdrum syntax. The **record** command translates a live or computer-generated MIDI performance to the `**MIDI` representation. The **encode** command provides an interactive editor that translates MIDI events to any pre-defined or user-defined Humdrum representation.

Note that the Humdrum **record** and **encode** commands are currently available only for the DOS operating system. These commands can be used only with computers that have MIDI capable hardware.

The *record* Command

The **record** command captures a stream of input MIDI data and translates this data into the Humdrum `**MIDI` representation described in Chapter 7. The input data is obtained from a MIDI instrument such as a keyboard synthesizer.

Recording commences as soon as the command is invoked and recording ceases when any key is pressed — with the exception of the space bar. Pressing the space bar causes a `**MIDI` barline token to be output. Measure numbers are incremented automatically beginning with measure 1.

Only MIDI key-press activity (including after-touch) information is recorded. MIDI "system-exclusive" instructions and other non-key-press data are not recorded.

Each MIDI channel is represented using a separate Humdrum spine. New spines are added automatically during the recording in response to MIDI activity appearing in any MIDI channel. Once a MIDI channel becomes active, the corresponding Humdrum spine continues to be output until the recording is terminated.

The recorded output is normally directed to a file as in the following:

```
record > filippa
```

The *encode* Command

The **encode** command provides an interactive editor for capturing Humdrum data from a MIDI input, such as a keyboard synthesizer. MIDI events are mapped to user-defined signifiers so **encode** can be used to enter data directly into a particular representation such as **kern, **fret, **solfg, etc. Since the mapping of MIDI events to Humdrum data tokens is arbitrary, users can enter data using a representation design by the user.

The **encode** command is limited to encoding information one spine at a time. A typical use of **encode** is to encode individual musical parts or voices using a representation like **kern. A full score is generated from the individual parts using the **timebase** and **assemble** commands.

The **encode** command implements a full-screen interactive editor similar to the **vi** text editor. When invoked, the screen is divided into three display regions, including a *status window*, a *command window*, and a larger *text window*. The *status window* displays various items of basic information about the file being edited. The **command window** allows the user to execute general-purpose commands to manipulate the data. The **text window** is used to display, encode and edit the encoded Humdrum text.

When the **encode** command is invoked, its operation is determined by a configuration file (that may be written or edited by the user). This configuration file contains a series of definitions that map MIDI events to output strings. For example, the instruction

```
KEY 60 middle-C
```

assigns the key-on event for MIDI key #60 to the string `middle-C`. Each time key #60 is depressed, the string `middle-C` will appear in the text window.

Such mappings can be made for each individual MIDI key. In addition, the user may define mappings for *key velocity*. For example, the following instruction in the configuration file will map any key-velocities between 90 and 127 MIDI units to the apostrophe character (the `**kern` signifier for a staccato note):

```
VEL 90 127 '
```

A third class of mapping instructions relates to the elapsed time between MIDI key onsets — "delta time" or DEL. Consider, for example, the following configuration instructions:

```
DEL 48 80 8
DEL 81 112 8.
DEL 113 160 4
DEL 161 224 4.
DEL 225 320 2
```

These instructions divide the elapsed time between key onsets into five ranges. When the elapsed time lies between 48 and 80 clock ticks, the string "8" is output. When the elapsed time lies between 81 and 112 clock ticks, the string "8." is output. And so on. This allows the durations to be classified as either an eighth note, a dotted eighth note, a quarter note, a dotted quarter note, or a half note. Once again, the user is free to map events to any arbitrary output string and to arrange

the ranges and number of classes as needed.

The Humdrum Toolkit provides a large selection of predefined configuration files for use with **encode**. Depending on the configuration, the input may be mapped to a particular representation such as **kern. For example, Humdrum provides a configuration file that is optimized for encoding lute tablatures using the **fret representation. Other configuration files are optimized for particular keys. For example, one may select a configuration file that interprets the MIDI events in the key of C# minor; in this case, playing the pitch C will result in a default encoding of B#.

The **encode** command provides many additional features that facilitate encoding Humdrum data from a MIDI input device. These include setting metronome values, assigning the beat, rearranging the order of signifiers, making global and local substitutions, replaying an interpreted input, defining buffers and string constants, and so on. Using the configuration files, users can tailor the **encode** editor to suit specific needs and skills.

Reprise

In this chapter we have briefly identified two tools for capturing MIDI-related input: **encode** and **record.** These tools allow MIDI data to be translated to a Humdrum format. Further information regarding these tools is given in the *Humdrum Toolkit Reference Manual.*