

NAME

midi — convert from ****kern** to Humdrum ****MIDI** format

SYNOPSIS

midi [-Ccu] [-d *n*] [-q *n*] [*inputfile.krn* ...] [> *outputfile.hmd*]

DESCRIPTION

The **mid**i command converts Humdrum ****kern** data into Humdrum ****MIDI** data. Humdrum ****MIDI** data may be played by the **perform** command or converted to a standard MIDI file using the **smf** command. The **mid**i command will translate all ****kern** spines in the input stream into ****MIDI** data, and will echo any non-kern spines.

It is recommended that output files produced using the **mid**i command should be given names with the distinguishing ‘.hmd’ extension (‘Humdrum mid

OPTIONS

The **mid**i command provides the following options:

- c echo all data records as global comments (prior to the data record)
- C echo all data records as global comments (following the data record)
- d *n* assigns a note-duration of *n* ****recip** value to all pitches and rests
- h displays a help screen summarizing the command syntax
- q *n* set number of MIDI clock ticks per quarter-duration to *n*
- u suppress the deletion of duplicate (unison) concurrent note-on instructions

Options are specified in the command line.

If the -c option (‘comment’) is invoked, all data records are echoed as global comments in addition to the usual processing. This option is useful in conjunction with the **perform** command; **perform** echoes all global comments while playing MIDI inputs, and so the comment option allows the user to view the original ****kern** score while the music is performed.

The -d *n* option allows the user to reassign all the durations of all notes and rests in a given input. The value *n* is specified in reciprocal duration notation (****recip**), where ‘4’ represents a quarter-duration, ‘2.’ represents a dotted half-duration, ‘12’ represents a quarter-note triplet, ‘0’ represents a breve, etc. (See ****recip** in Section 2 for details.) With the -d option all pitches and rests in the input will be assigned the specified duration value — whether or not the notes already have encoded durations. This option is useful for such tasks as proof-listening to passages containing especially short notes, or auditing works — such as Gregorian chant — where durations have not been encoded or are suspect for other reasons. Note that the -d option should be used only in the case of monophonic inputs,

or multi-part inputs that contain strictly isochronous chords. Polyphonic inputs containing non-isochronous durations will cause a loss of synchronization between the parts. The resulting output is apt to cause serious problems with commands such as **perform** and **smf**.

In contrast to MIDI performance data, canonical musical scores (such as ****kern**) frequently contain unisons — where two voices share the same pitch for a period of time. MIDI produces a ‘note-off’ instruction at the end of each note; however only a single ‘note-off’ instruction is required to turn-off a note. Unfortunately, the first note-off will terminate a note — even if the same pitch is sustained in another voice. In normal operation, **midi** suppresses the sending of the first of two MIDI ‘note-off’ commands when a unison is encountered in the input. This ensures that the pitch will be sustained for the longer of the unison durations. This feature can be disabled by invoking the **-u** option.

The **midi** command assigns a default duration of 72 MIDI clock-ticks per quarter-note. This default value may be reassigned using the **-q n** option. The value *n* should be chosen so that all canonical durations present in a given score divide evenly into it. For example, the default value of 72 clock-ticks per quarter-note, means that eighth-notes and sixteenth-notes will have durations of 36 and 18 clock-ticks respectively. Eighth-note triplets and sixteenth-note triplets will have durations of 24 and 12 clock-ticks respectively. Thirty-second notes will have a duration of 9 clock-ticks. But sixty-fourth notes can cause synchronization problems since they do not divide evenly into 72. The user may wish to reassign this value depending on the types of canonical durations present in a given input. Changing the default number of clock-ticks will affect the tempo of performance when using the **perform** command. Hence, it may be necessary to revise the tempo indication of a resulting Humdrum midi-format file.

EXAMPLES

The following examples illustrate how **midi** may be invoked.

```
midi chopin > chopin.hmd
```

converts the ****kern** data from the file `chopin` to ****MIDI** data in the file `chopin.hmd`.
The command:

```
midi -c siegfrd.idl > siegfrd.hmd
```

translates the ****kern** data from the file `siegfrd.idl` to ****MIDI** data in the file `siegfrd.hmd` and echoes all ****kern** data as global comments.

SAMPLE OUTPUTS

The operation of **midi** is illustrated in the following inputs and outputs. In the first example, a simple C major scale is encoded in ****kern**.


```

!! midi example #1
**kern
*M2/4
*C:
=1
8c
8r
8d
8e
=2
8f
8g
8a
8b
=3
4cc
4r
====
*-

```

Executing the **midi** command with the default settings results in the ****MIDI** output given below. Notice that a default channel of MIDI channel 1 has been assigned via the tandem interpretation ***Ch1**. Note also that each ****kern** note has resulted in two MIDI instructions: “key-on” and “key-off”. Since the end of one note coincides with the start of the subsequent note, key-on and key-off data are output as Humdrum double-stops (two tokens separated by a single space character).

```

!! midi example #1
**MIDI
*Ch1
*M2/4
*C:
=1
72/60/64
36/-60/64
36/62/64
36/-62/64 36/64/64
=2
36/-64/64 36/65/64
36/-65/64 36/67/64
36/-67/64 36/69/64
36/-69/64 36/71/64
=3
36/-71/64 36/72/64
72/-72/64
====
.
*-

```

****MIDI** data consist of three numbers separated by slashes (/). The first integer represents the number of clock ticks that must elapse from the previous instruction before the current event is initiated. In the above case, a default value of 72 clock ticks per quarter-note has resulted in 36 clock ticks for each eighth note. The second integer represents the MIDI key

number, where middle C is represented as key #60 (negative numbers denote key-off instructions). The third integer represents the MIDI key velocity. The default key velocity is 64 units.

The second example illustrates the handling of input containing multiple-stops.

```
!! midi example #2
**harm    **kern
*          *Ch3
=1         =1
I          4c 4e 4g
IV         4c 4f 4a
V          4d 4g 4b
I          4e 4g 4cc
==         ==
*--        *--
```

The output below is generated by invoking the following command:

```
midi -c input > output

!! midi example #2
**harm    **MIDI
*          *Ch1
!!**harm  **kern
*          *Ch3
!!*       *Ch3
=1         =1
!!=1      =1
I          0/60/64 0/64/64 0/67/64
!!I        4c 4e 4g
IV         96/-60/64 96/-64/64 96/-67/64 96/60/64 96/65/64 96/69/64
!!IV       4c 4f 4a
V          96/-60/64 96/-65/64 96/-69/64 96/62/64 96/67/64 96/71/64
!!V        4d 4g 4b
I          96/-62/64 96/-67/64 96/-71/64 96/64/64 96/67/64 96/72/64
!!I        4e 4g 4cc
==         ==
!!==      ==
.          96/-64/64 96/-67/64 96/-72/64
*--        *--
!!*-      *--
```

Notice that non-kern data (**harm) is echoed in the output. Also, notice that each input record has been reproduced as a global comment (preceded by !!). This feature is useful in conjunction with the **perform** command.

DIAGNOSTICS

The **midi** command echoes tempo indications for the benefit of the **perform** command. If a tempo range is specified (e.g. MM92-98), **midi** calculates the average range and echoes that (MM95). Tempo terms (such as “largo”) are not handled by **midi**.

PORTABILITY

DOS 2.0 and up.

SEE ALSO

humdrum (4), ****kern** (2), **kern** (4), ****MIDI** (2), **perform** (4), **proof** (4), **smf** (4), **tacet** (4)

PROPOSED MODIFICATIONS

Channel assignment tandem interpretations (e.g. *Ch6) should be recognized in the input stream and cause the default channel 1 output to be suppressed.