

```
+ + + + + + + + + + + + + + + + + + + + + +  
+ + + + + + + + + + + + + + + + + + + + + +  
+ Everything you need to know about  
+ The Humdrum  
+ ***kern  
+ Representation  
+ + + + + + + + + + + + + + + + + + + + + +
```

## TABLE OF CONTENTS

- (1) General description
- (2) Table of Signifiers
- (3) Processing Issues
  - 1.0 [Introduction](#)
  - 2.0 [Humdrum](#)
    - 2.1 [Layout](#)
    - 2.2 [Interpretations](#)
    - 2.3 [Tandem Interpretations](#)
    - 2.4 [Comments](#)
  - 3.0 [\\*\\*kern](#)
    - 3.1 [Tokens](#)
    - 3.2 [Note Tokens](#)
      - 3.2.1 [Pitch](#)
      - 3.2.2 [Slurs, Ties, Phrases](#)
      - 3.2.3 [Ornaments](#)
      - 3.2.4 [Articulation Marks](#)
      - 3.2.5 [Up & Down Bows](#)
      - 3.2.6 [Stem Directions](#)
      - 3.2.7 [Duration](#)
      - 3.2.8 [n-Tuplets](#)
      - 3.2.9 [Grace Notes & Groupettos](#)
      - 3.2.10 [Beaming](#)
    - 3.3 [Rests](#)
    - 3.4 [Barlines](#)
    - 3.5 [Editorial Signifiers](#)
    - 3.6 [File Extension](#)
    - 3.7 [Table Of Signifiers](#)
    - 3.8 [Context Dependencies](#)
    - 3.9 [\\*\\*kern Example](#)
    - 3.10 [Pertinent Commands](#)
    - 3.11 [Tandem Interpretations](#)
    - 3.12 [Multiple Stops](#)
  - 4.0 [Humdrum Revisited](#)
    - 4.1 [Spine Paths](#)
    - 4.2 [Repetitions](#)
  - 5.0 [Processing Hints](#)

## 1.0 INTRODUCTION

=====

The \*\*kern representation can be used to represent basic or core information for period-of-common-practice Western music. The \*\*kern scheme allows the encoding of pitch and duration, as well as accidentals, articulation, ornamentation, ties, slurs, phrasing, glissandi, barlines, stem-direction and beaming.

In general, \*\*kern is intended to represent the underlying SYNTACTIC information conveyed by a musical score rather than the visual or ORTHOGRAPHIC information embodied by a given printed rendition. \*\*kern is designed to facilitate analytic applications rather than music printing or sound generation. Other Humdrum representations might be used for these latter purposes.

Note that \*\*kern is just one of many representation schemes that conform to the broad "Humdrum" syntax. There are innumerable other representations within the Humdrum system, and the \*\*kern representation is not intended to cater to every type of analytic need.

Since \*\*kern is a representation that conforms to the HUMDRUM SYNTAX, we will begin by discussing the syntax for Humdrum -- including general layout information. Subsequent sections will focus more specifically on \*\*kern.

## =====

## 2.0 HUMDRUM

## =====

Humdrum is a syntax or grammar within which an unbounded number of representations may be defined. Any representation conforming to this syntax can be manipulated using the Humdrum Toolkit software.

Since no single representation can possibly cater to all possible musical needs, Humdrum avoids defining a single representation. The premise is that the problem of representation ought to be broken-up into more manageable pieces, or that special-purpose representations be defined as needed to cater to unique tasks.

## =====

## 2.1 LAYOUT

## =====

Musical notation might be regarded as a two-dimension table or grid in which time moves from left to right:

Soprano:	S1	S2	S3	S4	S5	etc.
Alto:	A1	A2	A3	A4	A5	etc.
Tenor:	T1	T2	T3	T4	T5	etc.
Bass:	B1	B2	B3	B4	B5	etc.

In typical musical scores, individual musical parts or voices are generally ordered with the higher parts toward the top of the system.

Humdrum data are also encoded in a two-dimensional table or grid -- only ROTATED so that time moves down the page. Notice that the higher parts are toward the right:

	Bass	Tenor	Alto	Soprano
B1	T1	A1	S1	
B2	T2	A2	S2	
B3	T3	A3	S3	
B4	T4	A4	S4	

```

B5  T5  A5  S5
etc. etc. etc. etc.

```

This layout is exactly as though the musical score was turned sideways, and notated on a single system.

Each position in the grid is referred to as a "token." Tokens within each line are separated by tabs (not spaces). In other words, tabs separate data for different voices or parts.

Each line of data is referred to as a "record". Each record encodes a "slice" of time -- and so represents a single musical moment or sonority.

Not all musical parts have a new note with each sonority. For example, some parts or voices may sustain a pitch while other parts are moving. When this occurs, a special "null token" is encoded in that part's grid-position. A "null token" is represented simply by a single "period" character. For example, in the following representation, the soprano part has two notes for each note in the other parts:

```

Bass Tenor Alto Soprano
B1  T1  A1  S1
.   .   .   S2
B2  T2  A2  S3
.   .   .   S4
B3  T3  A3  S5
etc. etc. etc. etc.

```

We have noted that each column of data represents a single part or voice. This is often the case in the Humdrum representations, but is not always the case. Columns of data may be used to represent many different things -- such as text, figured bass, fingerings, or other information.

In the Humdrum syntax, columns of data are called SPINES. This special term is required because a SPINE is not always the same as a COLUMN of data. For example, SPINES of information may change position within a file; they may disappear, or split into several columns, for example.

## ===== 2.2 INTERPRETATIONS =====

Humdrum SPINES must always be labelled in order to indicate the type of information being represented. Such labels are referred to as INTERPRETATIONS (since they indicate how the data should be interpreted). Humdrum interpretations always begin with two asterisks (\*\*). Below we have identified four spines -- each with its own interpretation:

```

**Bass **Tenor **Alto **Soprano
B1      T1      A1      S1
.       .       .       S2
B2      T2      A2      S3
.       .       .       S4
B3      T3      A3      S5
etc.    etc.    etc.    etc.
*_      *_      *_      *_

```

In addition, we've added special tokens (\*-) at the end of each spine in the above example. These special tokens are called SPINE-PATH TERMINATORS. They indicate the end of their respective spines.

(Incidentally, the above sample data now conforms to the Humdrum Syntax.)

Normally, the soprano, alto, tenor, and bass parts would all encode the same TYPE of information -- that is, notes, rests, and such. So we'd normally expect the INTERPRETATIONS to be the same, e.g.:

```

**notes **notes **notes **notes
B1      T1      A1      S1
.        .        .        S2
B2      T2      A2      S3
.        .        .        S4
B3      T3      A3      S5
etc.    etc.    etc.    etc.
*_      *_      *_      *_

```

=====

## 2.3 TANDEM INTERPRETATIONS

=====

However, we'd still like to distinguish the different parts. We can do this using Humdrum TANDEM INTERPRETATIONS. Tandem interpretations are identified by a single leading asterisk (\*). Tandem interpretations encode additional or supplementary information -- in this case, the names of the voices:

```

**notes **notes **notes **notes
*Bass   *Tenor *Alto   *Soprano
B1      T1      A1      S1
.        .        .        S2
B2      T2      A2      S3
.        .        .        S4
B3      T3      A3      S5
etc.    etc.    etc.    etc.
*_      *_      *_      *_

```

Any number of tandem interpretations may appear in a spine. Tandem interpretations are often used to encode various pieces of GENERAL INFORMATION. In the following example, the tandem interpretations encode (1) the names of the voices (soprano, alto, tenor, or bass), (2) a meter signature of 3/8, (3) a key signature consisting of F-sharp, and (4) a metronome marking of 92 quarter-notes per minute:

```

**notes **notes **notes **notes
*bass   *tenor *alto   *soprano
*M3/8   *M3/8  *M3/8   *M3/8
*k[f#]  *k[f#] *k[f#]  *k[f#]
*MM96   *MM96  *MM96   *MM96
B1      T1      A1      S1
.        .        .        S2
B2      T2      A2      S3
.        .        .        S4
B3      T3      A3      S5
etc.    etc.    etc.    etc.
*_      *_      *_      *_

```

Notice that each SPINE in the above example encodes the SAME meter, SAME key signature, and SAME metronome information. Although this is the most likely musical situation, Humdrum does not forbid having parts with different key signatures, meters, or tempi, etc.

In the following example, eight simultaneous spines are encoded. Notice that several tandem interpretations pertain to instrument identification. From left to right, we can identify the following instruments: harpsichord or cembalo (\*Icembra), contrabass or violone (\*Icbass), violoncello (\*Icello), viola (\*Iviola), violin (\*Ivioln), traverse flute (\*Iflt), and horn (\*Icor). In the ensuing line, the INSTRUMENT CLASSES (IC) are identified: keyboard instruments (\*ICKlav), string instruments (\*ICstr), woodwinds (\*ICww), and brass (\*ICbras).

```

**notes **notes **notes **notes **notes **notes **notes **notes
*Icembra *Icbass *Icello *Iviola *Ivioln *Ivioln *Iflt *Icor
*ICKlav *ICstr *ICstr *ICstr *ICstr *ICstr *ICww *ICbras
*IGripn *IGripn *IGripn *IGripn *IGripn *IGconc *IGconc *IGconc
*IGcont *IGcont *      *      *      *      *      *
data      data      data      data      data      data      data      data
data      data      data      data      data      data      data      data
etc.      etc.      etc.      etc.      etc.      etc.      etc.      etc.
*_        *_        *_        *_        *_        *_        *_        *_

```

The third and fourth tandem interpretation records (above) identify the INSTRUMENT GROUPINGS (IG) for this work. There are three groups: the continuo instruments (\*ICcont), the ripieno instruments (\*ICripn), and the concertino instruments (\*ICconc). Notice that the cembalo (harpsichord) and the contrabass both belong to TWO groups concurrently -- they are both part of the "continuo" grouping as well as the "ripieno" grouping. Consequently, an extra record has been encoded in order to identify the continuo grouping (\*IGcont). Since the other instruments require no further grouping information, isolate asterisks appear in the respective spines as place-holders. Single isolated asterisks are referred to as NULL INTERPRETATIONS and they have no meaning apart from their role as place-holders.

## =====

### 2.4 COMMENTS

## =====

In any representation, some information may best be conveyed as COMMENTS rather than as part of the encoded data.

Humdrum comments are records (lines) that begin with an exclamation mark.

Two basic types of comments are distinguished. Comments that pertain to all spines are referred to as GLOBAL COMMENTS and begin with two exclamation marks (!!). Comments that pertain to a single spine are called LOCAL COMMENTS and begin with a single exclamation mark in each spine.

```

!! J.S. Bach, Four-part Fugue
**notes **notes **notes **notes
*bass   *tenor  *alto   *soprano
!       !       !       !subject
rest    rest    rest    S1
.        .        .        S2
!       !answer !       !
.        T1      .        S3
.        T2      .        S4

```

.	T3	.	S5
etc.	etc.	etc.	etc.
*_	*_	*_	*_

The above example illustrates both types of comments. The first record is a global comment identifying the composer and piece. The fourth record encodes a local comment. The comment identifies that the fugal subject begins in the soprano voice. On the same line, notice that the other spines also encode single exclamation marks, but contain no text. These are referred to as NULL LOCAL COMMENTS.

Later, in the seventh record, another local comment appears identifying the fugal answer beginning in the tenor voice.

Notice that LOCAL COMMENTS conform to the prevailing spine structure. Each spine begins with an exclamation mark and tabs continue to demarcate each spine. GLOBAL COMMENTS by contrast completely ignore the spines.

In a later section of this description of Humdrum a further type of comment is distinguished -- the REFERENCE RECORD. Reference records are formal ways of encoding "library-type" information pertaining to a score.

This brings us to the end of our introduction to the Humdrum Syntax. There is more you should know about Humdrum representations, but further information is deferred to a later section. At this point, let's introduce **\*\*kern**.

```
=====
3.0  **KERN
=====
```

Since **"\*\*kern"** begins with two asterisks, this identifies it as a Humdrum exclusive interpretation. "Kern" is German for "core" -- and **\*\*kern** is intended to be ONE WAY of representing the basic or core information conveyed by traditional Western musical scores.

```
=====
3.1  TOKENS
=====
```

As we've seen, Humdrum information is encoded in data "tokens." The **\*\*kern** representation distinguishes three types of data tokens: (1) notes, (2) rests, and (3) barlines. In **\*\*kern**, all non-null data tokens must be one of these three types.

```
=====
3.2  NOTE TOKENS
=====
```

Notes can encode a variety of attributes including absolute pitch, accidental, canonical duration, articulation, ornamentation, ties, slurs, phrasing, stem-direction and beaming.

```
=====
3.2.1 PITCH
=====
```

Pitch information is encoded through a scheme of upper- and lower-case letters. Middle C (C4) is represented using the single lower-case letter "c". Successive octaves are designated by letter repetition, thus C5 is represented by "cc", C6 by "ccc" and so on. The higher the octave, the more letters are repeated.

For pitches below C4, upper-case letters are used: "C" designates C3, "CC" designates C2, and so on. This same scheme is used for other pitch letter-names. Changes of octave are deemed to occur between B and C. Thus the B below middle c is represented as "B"; the B below "CC" is represented as "BBB", and so on. The lower the octave, the more letters are repeated.

Accidentals are encoded using the octothorpe (#) for sharps, the minus sign (-) for flats, and the lower-case letter "n" for naturals. Accidentals are encoded immediately following the diatonic pitch information.

Examples of \*\*kern pitch representations are given below:

```

c      middle C (i.e. C4)
cc     C an octave higher than middle C (C5)
C      C an octave lower than middle C (C3)
CC     C two octaves lower than middle C (C2)
B      B below middle C (B3)
b      B a major seventh above middle C (B4)
d#     D-sharp above middle C (D#4)
d##    D double-sharp above middle C
d###   D triple-sharp above middle C
e-     E-flat above middle C (enharmonically equivalent to d##)
BB--   B double-flat; augmented ninth below middle C
cn     C natural, middle C

```

Double-sharps and double-flats have no special representations in \*\*kern and are simply denoted by repetition (##) and (--). Triple- and quadruple accidentals are similarly encoded by repetition. Sharps, flats, and naturals are mutually exclusive in \*\*kern, so tokens such as "cc#n" and "GG-#" are illegal.

(Note that numbers are NOT used in \*\*kern pitch designations because the numbers are used to represent note durations.)

In the \*\*kern representation, all pitches are encoded as CONTEXTUALLY INDEPENDENT ABSOLUTE VALUES.

The word "independent" here means that pitches are encoded as isolated entities -- without regard for what is going on around them. For example, in \*\*kern, pitches must be encoded with the appropriate accidental, EVEN IF THE ACCIDENTAL IS SPECIFIED IN A KEY-SIGNATURE, OR IS PRESENT EARLIER IN THE SAME MEASURE.

The word "absolute" here means that pitches are encoded without regard to transposition. In \*\*kern, transposing instruments are always represented at (sounding) concert pitch. A special tandem interpretation is provided to indicate the nature of any transposing instrument -- but the pitches themselves appear only at concert pitch.

Pitches in \*\*kern are encoded as "nominally" equally-tempered values. A special tandem interpretation is provided to indicate if the tuning system is other than equal temperament.

In addition to the pitch information, note tokens may include other types of signifiers.

### =====

### 3.2.2 SLURS, TIES, PHRASES

### =====

The \*\*kern representation provides no generic means for representing "curved lines" found in printed

scores. Since \*\*kern is a "syntactic" rather than "orthographic" representation, all lines must be explicitly interpreted as either TIES, SLURS or PHRASES.

The open brace { denotes the beginning of a phrase. The closed brace } denotes the end of a phrase.

The open parenthesis ( and closed parenthesis ) signify the beginning and end of a slur respectively.

The open square bracket [ denotes the first note of a tie.

The closed square bracket ] denotes the last note of a tie.

The underscore character \_ denotes middle notes of a tie.

Slurs and phrase markings can be NESTED (e.g. slurs within slurs) and may also be ELIDED (e.g. overlapping phrases) to a single depth.

Nested markings mean that one slur or phrase is entirely subsumed under another slur or phrase. For example: ( ( ) ) means that a short slur has occurred within a longer slur.

Elisions are overlaps, for example, where an existing phrase fails to end while a new phrase begins. In \*\*kern the ampersand character is used to mark elided slurs or phrases. For example: { & { } & } means that two phrases overlap -- the initial phrase ending after second phrase has begun.

### =====

### 3.2.3 ORNAMENTS

### =====

Additional signifiers are provided for denoting ornaments.

The letters "T" and "t" are used to signify whole-tone and semitone TRILLS, respectively. Whole-tone and semitone MORDENTS are signified by the letters "M" and "m".

INVERTED MORDENTS are signified by "W" (whole-tone) and "w" (semitone).

(Note that trills, mordents, and inverted mordents wider than two semitones in size are also denoted by the upper-case signifier.)

The letter "S" signifies a TURN, whereas the dollar sign ("\$\$") signifies an inverted (or Wagnerian) turn. When a concluding turn is appended to the end of an ornament (such as a trill), the upper-case letter "R" is added to the ornament signifier (as in "tR" and "TR").

In addition to these ornaments, \*\*kern provides a signifier for (multi- note) ARPEGGIATION (":").

The presence of ornaments OTHER than trills, mordents, inverted mordents, and turns can be indicated by the generic ornament symbol ("O").

### =====

### 3.2.4 ARTICULATION MARKS

### =====

Articulation marks include the apostrophe (') for STACCATO, the double- quote (") for PIZZICATO, the grave (`) for ATTACCA, the tilde (~) for TENUTO, and the caret (^) for all note-related accents (including < and >).

The presence of other articulation types can be indicated by the generic articulation symbol ("I").

### =====

### 3.2.5 UP & DOWN BOWS

### =====



\*\*kern provides signifiers for up-bow (v) and down-bow (u).

### 3.2.6 STEM DIRECTIONS

As noted, the \*\*kern scheme is intended for analytic applications rather than as a means for representing visual renderings of notation. Nevertheless, \*\*kern distinguishes up-stems, down-stems, and beamings in order to assist in analytic tasks such as the determination of voicings and in order to facilitate the parsing of note-groupings.

UP-STEMS and DOWN-STEMS are indicated by the slash ("/") and backslash characters ("\") respectively.

### 3.2.7 DURATION

Durations are encoded using a reciprocal number notation. More specifically, durations are encoded as nominal beat proportions using integer numbers and the period character.

With the exception of the value zero, durations are represented by reciprocal numerical values corresponding to the American duration names: "1" for whole note, "8" for eighth, "32" for thirty-second, etc. The number zero (0) is reserved for the breve duration (i.e. a duration of twice the length of a whole note).

0	breve duration
1	whole duration
2	half duration
4	quarter duration
8	eighth duration
16	sixteenth duration
32	thirty-second duration
64	sixty-fourth duration
etc.	

DOTTED DURATIONS are indicated by adding the period character (.) immediately following the numerical value -- hence "8." signifies a dotted-eighth note and "2.." signifies a doubly-dotted half note. Any number of augmentation dots may follow the duration integer.

2.	dotted half duration
8..	doubly-dotted eighth duration

Triplet and other irregular durations are represented in a somewhat more arcane, though no less logical fashion. See below.

The semicolon (;) denotes a pause.

### 3.2.8 N-TUPLETS

Triplet and other irregular durations are represented using the same reciprocal logic.

Consider, for example, the quarter-note triplet duration. Three quarter triplets occur in the time of four quarters or one whole duration. If we divide a whole duration ("1") into three equal parts, each part has a duration of one-third. The corresponding reciprocal integer for  $1/3$  is 3, hence \*\*kern represents a quarter-note triplet as a "third-note" -- 3. Similarly, eighth-note triplets are represented by the integer 6 while sixteenth-note triplets are represented by the integer 12. Eighth-note quintuplets (5 in the time of 4) will be represented by the value 10 (a half duration divided by 5).

In general, the way to determine the \*\*kern equivalent of an arbitrary "tuplet" duration is to multiply the number of tuplets by the total duration which they occupy. If 7 notes of equal duration occupy the duration of a whole-note ("1"), then each septuplet is represented by the value 7 (i.e.  $1 \times 7$ ). A more extreme example is 23 notes in the time of a doubly-dotted quarter. The appropriate \*\*kern duration can be found by multiplying 4 by 23 (equals 92) and adding the appropriate augmentation dots. Thus "92.." is the correct \*\*kern encoding for a note whose duration is 23 notes in the time of a doubly-dotted quarter.

The well-known rhythm in Gustav Holst's "The Planets" ('Mars' movement) is encoded as:

```
!! Gustav Holst
**kern
*M5/4
=1
12
12
12
4
4
8
8
4
=2
*_
```

### 3.2.9 GRACE NOTES & GROUPETTOS

The \*\*kern representation also allows for the encoding of acciaccaturas (grace notes), non-canonical groupettos, and appoggiaturas.

Depending on the expected analytic application, one way to handle these notational devices is to encode the notes according to the manner in which they are typically performed. Alternatively, since the component notes of an expanded ornament are viewed as embellishments that hold potentially less analytic status, a special designation for these notes can be useful for certain types of studies.

ACCIACCATURAS (GRACE NOTES) are visually represented as miniature notes denoted by a slash drawn through the stem. In \*\*kern these notes are treated as "durationless" notes and are designated by the lower-case letter "q". Hence, the token "G#q" denotes a G#3 grace note with an undetermined duration.

Non-canonical GROUPETTOS are miniature (non-cue) notes (typically appearing in groups) whose stems do not contain a slash, and whose notated durations cause the total notated duration for the measure to exceed the prevailing meter. These groupetto notes are encoded as notes retaining their notated durations, but all such notes are also designated by the upper-case letter "Q". Hence, a miniature sixteenth-note middle C would be encoded as "16cQ".

When processed by various tools in the Humdrum Toolkit, these notes may be treated as equivalent to their notated durations. Alternatively, in some types of processing these notes may be discarded. For example, the Humdrum "timebase" command eliminates acciaccaturas and gruppetto notes. Note that data records containing acciaccaturas or gruppetto notes must not include normal notes.

In the case of APPOGGIATURAS, \*\*kern requires that they be encoded as performed. An appropriate duration is assigned to the appoggiatura according to common performance practice. The duration of the subsequent note is reduced by a corresponding amount. The status of the two notes forming the appoggiatura is nevertheless marked. The appoggiatura note itself is designated by the upper-case letter "P", whereas the subsequent note (whose notated duration has been shorted) is designated by the lower-case letter "p".

### =====

#### 3.2.10 BEAMING

### =====

The beginning and ends of beams are signified by the upper-case letters "L" and "J" respectively. For example, the first note of a beam will include the "L" signifier and the last note of a beam will include the "J" signifier.

Where notes are joined by more than one beam, each beam is designated by its own L-J pair. For example, where two sixteenth notes are beamed together (i.e. two beams), the first note will encode "LL" whereas the last note will encode "JJ". For each new beam that starts, the "L" signifier should appear; for each beam that ends, the "J" signifier should appear.

Partial beams may extend to the right (K) or left (k). Again, multiple partial beams are indicated via letter repetition. By way of example, a doubly-dotted sixteenth note beamed to a sixty-fourth note can be represented as:

16..LL  
64JJkk

### =====

#### 3.3 RESTS

### =====

Rests tokens are denoted by the single lower-case letter "r" along with a numerical duration signifier. Rests may also have the attributes of stem-direction, beaming, slur, phrase, and gruppetto, but rests cannot be assigned articulation or ornamentation attributes.

### =====

#### 3.4 BARLINES

### =====

In the \*\*kern representation, barlines are represented primarily as logical entities, with optional signifiers for specifying the precise visual appearance of the barlines. That is, barlines are represented SYNTACTICALLY, with some possibilities for supplementary ORTHOGRAPHIC information.

Barlines are logically signified by the presence of an equals-sign (=) in the first column of a spine. Immediately after the equals sign there may follow an optional integer value indicating the measure number (e.g. =107 -- for measure 107). In addition, a lower-case alphabetic character may be appended to the measure number -- as in: =14b. This convention permits the user to distinguish measure numbers for first and second endings, etc. Measure numbers refer to the measure immediately following the barline, thus the token =23 occurs just prior to the encoded data for measure 23.

Double barlines are indicated by a minimum of two successive equals signs (==). Several consecutive equals signs may be encoded in order to enhance readability (e.g. =====).

An additional attribute for barlines is the pause -- which is represented by the semicolon (;). Thus the token =4; means that the barline starting measure number 4 has a pause written above or below it, while the token =====; means that a double barline contains a pause indication.

Where appropriate, users can specify more precisely the notational appearance of the barline by appending additional ORTHOGRAPHIC signifiers to the basic SYNTACTIC signifiers identified above. Barlines may be normal or heavy in width. Barlines of normal width can be explicitly indicated by the addition of the vertical bar (|).

Heavy barlines can be explicitly signified by the exclamation mark (!). Dotted barlines can be signified by the double quote character ("). Partial barlines (extending between the second and fourth lines) are signified by the single quote character ('). Partial barlines (rendered as a short vertical stroke across the top of the staff) are signified by the single grave character (`). "Invisible" barlines are signified by the minus sign (-).

In addition, barlines may be associated with repetition marks -- pointing left, right, or in both directions. The repeat sign is denoted by the colon character (:). (Note that the colon is used to represent exclusively ORTHOGRAPHIC rather than SYNTACTIC information. It is not used to indicate a logical repetition per se. The representation of repeated material is discussed later.)

All of these orthographic signifiers may be combined to form complex visual representations, such as a triple barline consisting of a normal line, followed by a heavy line, followed by a normal line, followed by a repeat indicator. See EXAMPLES below.

The following table summarizes the mappings of signifiers and signifieds for \*\*kern barlines.

All barlines begin with an equals sign in the first column of the spine.

	0-9 measure numbers
	a-z alternate measures
;	pause
=	barline
==	double barline
	normal width visual rendering
!	heavy width visual rendering
'	partial barline (from second to fourth line)
`	partial barline (short stroke at top of staff)
-	invisible barline
:	repeat sign

## EXAMPLES

Several examples of \*\*kern barlines are given below:

=	unnumbered barline
=29	the beginning of measure 29
=29;	the beginning of measure 29 with pause
=29a	first occurrence of measure 29
=29c	third occurrence of measure 29
=29c;	third occurrence of measure 29 with pause
==	double barline

```

==;    double barline with pause
====;  double barline with pause
=|     unnumbered barline, normal line width
=!     unnumbered barline, heavy line width
==|!   double barline, normal line followed by heavy line
=29|   beginning of measure 29, normal line width
=:|:   barline with left and right repeats, normal line width
=:||:  barline with left and right repeats, two normal-width lines
='     unnumbered barline, rendered with partial barline (mid)
=29`   beginning of measure 29, rendered with partial barline (top)
=29-   beginning of measure 29, no barline drawn
==:|!  double barline with repeat, normal/heavy lines
==|    logical double barline, visually rendered as single normal line
|       not a barline
29|    not a barline

```

### 3.5 EDITORIAL SIGNIFIERS

In representing any work, editorial interpretations are inevitable. It may be necessary to make explicit certain implicit information in a score (such as expanding abbreviations), or it may be necessary to estimate missing or unreadable information.

Although not all Humdrum representations require this, in the case of the \*\*kern representation, it is necessary to interpret the voicings -- that is, the degree of connectedness between successive pitches must be made explicit. This is an important editorial function in \*\*kern representations.

The \*\*kern representation provides several special-purpose signifiers to help chronicle various classes of editorial amendments, interpretations, or commentaries.

Five types of editorial signifiers are made available:

#### (1) SIC

Information is encoded literally, but is questionable. This is signified by the upper-case letter "Y" immediately following the pertinent signifier. For example, the token "4G#Y" means that the sharp is present in the original source, but is questionable. By contrast, the token "4GY#" means that the diatonic pitch "G" is present in the original source, but is questionable (the sharp is not in question).

An entire token can be identified the by "sic" signifier, by repeating the "Y". For example, the token "4G#YY" means that this entire note is present in the original, but is questionable.

#### (2) INVISIBLE SYMBOL

Unprinted note, rest, or barline, but logically implied. This is signified by the lower-case letter "y". For example, the token "4g#y" indicates that the sharp is implied but absent in the original source.

Once again, repeating the "y" signifier indicates that the entire token is implied. For example, an implied (but absent) barline might be encoded as "=yy". An implied half note rest: "2ryy".

#### (3) EDITORIAL INTERPRETATION

A "modest" act of editorial interpretation -- such as the interpretation of accidentals in *musica ficta*. This is signified by the lower-case letter "x". For example, the token "{x4g#" indicates that a curved line has been interpreted as a phrase marking (rather than, say, a slur).

The repeated "xx" is used to designate entire tokens that are interpretations.

#### (4) EDITORIAL INTERVENTION

A "significant" editorial intervention -- such as an added note. This is signified by the upper-case letter "X".

The repeated "XX" is used to designate entire tokens that are "significant" interpretations.

N.B. The difference between an "editorial interpretation" and an "editorial intervention" is a judgement made by the editor.

#### (5) FOOTNOTE

An accompanying local or global comment provides a text commentary pertaining to a specified data token. This is signified by the question-mark (?). For example, the token "4G#?" means that a nearby comment discusses the sharp in this token. The token "4G#??" means that a nearby comment discusses this note.

### 3.6 FILE EXTENSION

Files containing predominantly \*\*kern data are normally distinguished by the presence of the '.kern' extension in the filename.

### 3.7 TABLE OF SIGNIFIERS

The following table summarizes the \*\*kern mappings of signifiers and signifieds.

0	breve duration
1	whole duration
2	half duration
3	half-note triplet duration
4	quarter duration
6	quarter-note triplet duration
8	eighth duration
12	eighth-note triplet duration
16	sixteenth duration
24	sixteenth-note triplet duration
32	thirty-second duration
64	sixty-fourth duration
128	one-hundred and twenty-eighth duration
.	duration augmentation dot (must follow a number)
-	flat sign (minus character)
--	double-flat (two successive minus characters)

a-g	absolute pitches above middle C
A-G	absolute pitches below middle C
#	sharp
##	double sharp
h	end glissando
j	harmonic
k	partial beam extending leftward
kk	two partial beams extending leftward
m	mordent (semitone)
n	natural sign
p	designator of a note subsequent to an appoggiatura
q	acciaccatura (grace note signifier; in lieu of duration)
r	rest
t	trill (semitone)
u	down-bow
v	up-bow
w	inverted mordent (semitone)
x	editorial interpretation; immediately preceding signifier is interpreted
xx	editorial interpretation; entire data token is interpreted
y	editorial mark: invisible symbol; unprinted note-, rest-, or barline-attribute, but logically implied
yy	editorial mark: invisible symbol; unprinted note, rest, or barline, but logically implied
z	sforzando
H	begin glissando
I	generic articulation (unspecified articulation)
J	end beam
JJ	end two beams
K	partial beam extending rightward
KK	two partial beams extending rightward
L	start beam
LL	start two beams
M	mordent (whole tone)
O	generic ornament (unspecified ornament)
P	appoggiatura note designator
Q	groupetto note designator
R	signified ornament ends with a turn
S	turn
\$	Wagnerian turn
T	trill (whole tone)
W	inverted mordent (whole tone)
X	editorial intervention; immediately preceding signifier is an editorial addition
XX	editorial intervention; entire data token is an editorial addition

Y	editorial mark: sic marking; information is encoded literally, but is questionable
YY	editorial mark: sic marking; entire data token is encoded literally, but is questionable
(space character)	multiple-stop conjunction -- indicates joint note-tokens
=	barline; == double barline
[	first note of a tie
]	last note of a tie
_	middle note(s) of a tie (underscore)
(	slur start
)	slur end
{	phrase mark (start)
}	phrase mark (end)
;	pause sign
'	staccato mark
s	spiccato
"	pizzicato mark
`	attacca mark
~	tenuto mark
^	accent mark
:	arpeggiation (of multi-note chord)
,	breath mark
/	up-stem
\	down-stem
&	elision marker (for slurs or phrases)
?	editorial mark: immediately preceding signifier has accompanying editorial footnote
??	editorial mark: entire preceding data token has accompanying editorial footnote

### 3.8 CONTEXT DEPENDENCIES

In general, most signifiers in the \*\*kern representation are treated as context independent. This means, for example, that the data tokens `{(16ff#/'` and `/ff#16'({` are logically equivalent. A few exceptions to this principle are necessary in order to maintain the meaning of multiple-character signifiers, and to assure the predictability of certain types of processing (such as searching for patterns).

Numbers encoding a duration must be contiguous. That is, a sixteenth note may be encoded as `16ff#` or `ff#16` but not as `1ff#6`. Augmentation dots (signified by the period) must follow immediately after the associated duration numerals. Thus `16.ff#` is acceptable, but not `16ff#.` or `.16ff#`. Sharps, flats, and naturals must follow immediately after the corresponding alphabetic pitch signifiers (`16ff#` but not `16#ff`). Signifiers that can be repeated must be contiguous. This include `kk`, `KK`, `LL`, `JJ`, `XX`, `xx`, `??`, `##`, `--`, and `..`

The elision marker (`&`) must immediately precede the associated phrase (`&{ ... &}`) or slur (`&( ... &)`).

Barlines follow a strict contextual syntax. Barlines must begin with one or more equals-signs, followed by an optional measure number, followed by an optional lower-case letter, followed by an optional pause signifier, followed by optional orthographic (visual) signifiers.



In certain applications, it may be necessary to have a canonical ordering of the signifiers within \*\*kern data tokens. For example, when comparing two ostensibly identical \*\*kern files, differences of signifier orderings will cause UNIX commands such as "cmp" and "diff" to declare the files to be "different." In this case, it is useful to adopt a standard order of signifiers so that direct file comparisons may be made. Similarly, differences in signifier orderings can cause problems for pattern-matching tasks. For example, in searching for a sixteenth-note F-sharp, it is convenient to define a simple regular expression -- such as 16f# rather than having to define a regular expression that handles all possible contextual orderings -- such as (16.\*f#)|(f#.\*16).

For this reason, a canonical ordering of the \*\*kern signifiers is given in the following table.

signified	signifier(s)	comments
1. open phrase elision indicator	&	must precede {
2. open phrase mark	{	
3. open slur elision indicator	&	must precede (
4. open slur	(	
5. open tie	[	
6. duration	0123456789	any combination; signifiers may be repeated
7. augmentation dot(s)	.	signifier may be repeated
8. pitch	abcdefgABCDEFG	only one of; signifier may be repeated
9. accidental	- or # or n	- and # may be repeated
10. pause	;	
11. ornament	MmS\$TtWwR or O	O precludes others; no repetition of a given signifier; must appear in order given
12. appoggiatura designator	p or P	
13. acciaccatura designator	q	
14. gruppetto designator	Q	
15. articulation	z ' " ` ~ ^ : or I	I precludes others; no repetition of a given signifier; must appear in order given
16. bowing	u or v	only one of
17. stem-direction	/ or \	only one of
18. beaming	L or J	signifiers may be repeated
19. partial beaming	k or K	signifiers may be repeated
20. user-defined marks	il NUVZ @ % +   < >	one or more of; may be repeated but must be in order given
21. closed or continuing tie	] or _	
22. closed slur elision indicator	&	must precede )
23. closed slur	)	
24. closed phrase elision indicator	&	must precede }
25. closed phrase mark	}	
26. breath mark	,	
27. editorial marks	xx or XX	
28. editorial marks	yy or YY	

## 29. editorial marks

??

Note that the editorial signifiers ?, y, Y, x, and X, (as opposed to xx, XX, yy, YY, and ??) can appear anywhere in a data token, except as the first character.

### 3.9 KERN EXAMPLE

A sample document is given below:

```
!! J.S. Bach, Fugue 2 WTC Book I
!! (3 parts), in c minor; BWV 847b
**kern      **kern      **kern
*M4/4       *M4/4       *M4/4
*MM72      *MM72      *MM72
*k[b-e-a-] *k[b-e-a-] *k[b-e-a-]
*c:         *c:         *c:
=1          =1          =1
1r          8r          1r
.           16cc        .
.           16bn        .
.           8cc         .
.           8g          .
.           8a-         .
.           16cc        .
.           16b         .
.           8cc         .
.           8dd         .
=2          =2          =2
1r          8g          1r
.           16cc        .
.           16bn        .
.           8cc         .
.           8dd         .
.           16f         .
.           16g         .
.           4a-         .
.           16g         .
.           16f         .
=3          =3          =3
1r          16e-        8r
.           16cc        .
.           16bn        16gg
.           16an        16ff#
.           16g         8gg
.           16fn        .
```

.	16e-	8cc
.	16d	.
.	8c	8ee-
.	8ee-	16gg
.	.	16ff#
.	8dd	8gg
.	8cc	8aan
=4	=4	=4
*_	*_	*_

### 3.10 PERTINENT COMMANDS

A large number of Humdrum commands are able to process \*\*kern-encoded data. The following list is not exhaustive:

```

census -k determine general characteristics of a **kern file
cents    translates **kern to cents (**cents)
deg      translates **kern to relative scale degree (**deg)
degree   translates **kern to absolute scale degree (**degree)
freq     translates **kern to frequency (**freq)
hint     calculate harmonic intervals from **kern input
key      estimate the key of a **kern input
mint     calculate melodic intervals from **kern input
pc       translates **kern to pitch-class (**pc)
pitch    translates **kern to **pitch
proof    check for errors in **kern encoded file
semits   translate **kern to numerical **semits
solfa    translate **kern to numerical **solfa
solfg    translate **kern to numerical French **solfg
synco    measure degree of metric syncopation
timebase reformat **kern score with constant timebase
tonh     translate **kern to German **Tonh
trans    transpose **kern score
urrhythm characterize the rhythmic prototypes in a passage
vox      determine active and inactive voices in a Humdrum file
etc.
```

### 3.11 TANDEM INTERPRETATIONS

Some sample tandem interpretations frequently used in conjunction with \*\*kern are listed below:

```

clef *clefG2
instrument *I
instrument class *IC
instrument group *IG
```

```

key signatures *k[f#c#]
key *c#:
meter signatures *M6/8
tempo *MM96.3
timebase *tb32
transposing instrument *ITr

```

### 3.12 MULTIPLE STOPS

In \*\*kern, spines typically represent individual musical parts or voices. We might ask, what happens if a single "part" contains more than one concurrent note? A good example of such a situation occurs when a violin plays a double-stop.

The \*\*kern representation provides two answers. If the notes have the same duration, then they can be encoded as generic Humdrum MULTIPLE STOPS.

Multiple stops are encoded as two or more note tokens -- separated by single spaces. For example, the following \*\*kern data represents a scale played in ascending thirds:

```

**kern
4c 4e
8d 8f
8e 8g
8f 8a
8g 8b
8a 8cc
8b 8dd
4cc 4ee
*_

```

Notice that the representation remains a single spine (i.e. no tabs).

Multiple-stops may occur at any point in a \*\*kern spine. For example, the following \*\*kern data represents a scale that begins and ends with chords:

```

**kern
4c 4e 4g
8d
8e
8f
8g
8a
8b
4cc 4ee 4gg 4ccc
*_

```

An important limitation to the use of multiple-stops in \*\*kern is that each note within the multiple-stop must have the SAME DURATION. If the durations of the concurrent notes differ, then one must use Humdrum spine-path indicators (see below).

## =====

### 4.0 HUMDRUM REVISITED

## =====

Before ending our tutorial, there are some further topics pertaining to the \*\*kern representation -- but topics that properly belong to the Humdrum Syntax in general, rather than the \*\*kern representation specifically.

The first topic continues our discussion about concurrent notes within a single "part".

## =====

### 4.1 SPINE PATHS

## =====

So far we have discussed files that contain a fixed number of spines from the beginning to the end. Such files are referred to as RECTANGULAR FILES, since the grid or table never changes shape.

Humdrum provides general mechanisms by which spines can appear, disappear, or otherwise change position. These mechanisms are known as SPINE-PATH INTERPRETATIONS.

We've already encountered the spine-path TERMINATOR (\*-). This token must appear at the end of each spine. Spines can terminate at any point in a Humdrum data file. For example, in the following illustration, the second spine terminates before the first:

```

**A **B
a    b
a    b
*    *_
a
a
*_

```

Notice that a null interpretation (\*) appears in the first spine at the point where the terminator occurs in the second spine.

The next example is slightly more confusing. Here spine "A" has terminated prior to spines "B" and "C". Notice how the data continue:

```

**A **B **C
a    b    c
a    b    c
*_   *    *
b    c
b    c
*_   *_

```

At the point where spine "A" has been terminated, all spines to the right shift over one column. (Incidentally, this is the reason why we refer to SPINES rather than COLUMNS.) When spine "A" terminates, spine "B" appears in column 1 rather than column 2.

Humdrum also allows the user to ADD a new spine at any time using the SPINE-PATH ADD interpretation (\*+). In the following example, a spine "B" is added:

```

**A
a
a
*+
*   **B
a   b
a   b
*_  *_

```

Notice that `*+` means the following: "Add a new spine (immediately to the right) beginning with the next record." Also notice that when the new spine begins, it must first be properly labelled with an exclusive interpretation.

Another type of Humdrum spine-path interpretation is the SPLIT interpretation (`*^`). Splitting a spine causes a new spine (to the right of the current spine) to be spawned. An important feature of the spine-path split is that the new spine inherits all of the interpretations of the original single spine:

```

**A
a
a
*^
a1  a2
a1  a2
*_  *_

```

The complement to the spine-path SPLIT is the JOIN (`*v`). In order for the join to be legal, both of the joined spines must reside in neighboring columns, and both spines must share the same exclusive interpretation:

```

**A **A
a1  a2
a1  a2
*v  *v
a
a
*_

```

Together, spine-path SPLITS and JOINS are useful mechanisms for encoding multiple lines within a single voice or instrument. The following `**kern` example illustrates how two different lines (alternating thirds) within a single violin part might be encoded:

```

**kern
*violn
4c 4e
*^
4d      8r
.        4f
4e      .
.        4g

```

```

4f      .
.       4a
4g      .
.       4b
4a      .
.       4cc
4b      .
.       8dd
*v      *v
4cc 4ee
*_

```

The final type of spine-path interpretation is the EXCHANGE path (\*x). These interpretations must appear as a single pair. They cause the position of two spines to be swapped:

```

**A **B
a      b
a      b
*x     *x
b      a
b      a
*_     *_

```

## 4.2 REPETITIONS

An important feature of Humdrum is its ability to encode repetitions, Da Capos, strophes, and other features. This is a broad topic of discussion, so our presentation here will be limited to the encoding of repeats.

Humdrum files may be logically divided into segments or passages via Humdrum SECTION LABELS. A section label is a type of tandem interpretation that consists of a single asterisk followed by a greater-than sign, followed by a keyword that labels the section.

The following are examples of section labels:

```

*>Coda
*>1st ending
*>Refrain

```

Sections begin with a section label and end when another section label is specified or when the end of the data is encountered. If there is more than one spine present in a passage, identical section labels must appear concurrently in all spines.

Rather than encode multiple copies of a passage, a single instance may be encoded and labelled as a section. The complete version of the work can be reconstructed by referring to an EXPANSION LIST.

An expansion list is another tandem interpretation that contains an ordered list of section labels. In effect, the expansion list indicates how the abbreviated file should be expanded to a full-length encoding.

Consider the following expansion list:

\*>[verse1,refrain,verse2,refrain]

This list indicates that the abbreviated file contains (at least) three sections, labelled "verse1", "verse2", and "refrain". When the file is expanded, the "refrain" section will be repeated following each verse.

The Humdrum "thru" command expands ABBREVIATED FORMAT representations to a so-called THROUGH-COMPOSED FORMAT in which repeated passages are expanded according to an expansion list.

When the "thru" command is invoked, it eliminates any expansion lists present in the input; in addition, "thru" places a "\*thru" tandem interpretation in all spines immediately following each instance of an exclusive interpretation in the input. This marks the file as being in a through-composed format.

Humdrum provides other mechanisms for the encoding and extracting of different versions, editions, strophic materials, and other organizational mechanisms. These mechanisms are quite extensive. Refer to the Humdrum Toolkit Reference Manual for further information.

## =====

### 5.0 PROCESSING HINTS

## =====

#### PROBLEMS WITH TABS.

- 1. Some text editors are unable to handle tabs properly. In particular some text editors replace tabs by spaces. This circumvents the proper operation of Humdrum data files.

Use a full-feature editor that permits the use of tabs and does not inadvertently introduce control characters.

- 2. Most good editors allow the user to set the tabstops. The default setting is usually every 8 characters. For some files it is useful to set the tabstops farther apart.

#### PROBLEMS WITH LINE-LENGTH

Occasionally, a Humdrum file will contain more spines that can be fit comfortably on the screen. This may cause difficulties reading the data. There are four approaches to alleviating this problem.

- 1. Most computer monitors can be reconfigured from 80-column format to 130-column format. For most large files, increasing the number of columns displayed will eliminate the problem.

For DOS-based machines using VGA graphics, a common command to increase the display to 130 columns is:

```
/vga/vgamode 13225
```

To return your monitor back to 80 column format:

```
/vga/vgamode vga
```

(This assumes that the pertinent software is kept in the directory /vga)

Check your graphics-card documentation for further information.

- 2. Full-feature editors let the text "bleed" over the right-hand margin. This prevents the line from wrapping-around to the left margin -- and so confusing the user.



Using a full-feature editor, such as the Linux "vi" is recommended for files containing long lines.

- 3. Reducing the distance between tabstops may allow more spines to be displayed in a given horizontal space. However, this may not be practical, since data tokens may often contain more than 7 characters. If your application does not require all of the encoded information, you might consider working from copies that eliminate unnecessary information. For example, few analytic applications need stem-direction or beaming information. The following Humdrum command will eliminate this data from a file:

```
humsed 's/[LJ\\V]/g' inputfile > outputfile
```

- 4. A final solution is to use the Humdrum "extract" command to isolate certain groups of parts or instruments for viewing and editing. For example, a score that contains strings and woodwinds might be divided into two files (named "strings" and "winds" say) by executing the following command:

```
extract -i '*ICstr' inputfile > strings  
extract -i '*ICww' inputfile > winds
```

As long as no data records are eliminated, the files can be rejoined as necessary using the Humdrum "assemble" command:

```
assemble strings winds > fullscore
```

Incidentally, most editors will allow you to edit multiple files without having to exit the editor.