Chapter 5

The Humdrum Syntax

In the previous chapters we have seen several examples of pre-defined Humdrum representations, such as `**kern`, `**solfa` and `**MIDI`. These representations exhibit a number of common properties, including the manner in which the data are organized in spines. In this chapter, we provide a complete description of the Humdrum representation syntax. This chapter will help you better understand how Humdrum representations are organized, and will provide essential foundations for designing your own Humdrum representations.

The Humdrum syntax provides a framework within which representation schemes can be defined. Each scheme consists of a mapping between the concepts we wish to represent (called *signifieds*) and how we wish to represent them (called *signifiers*). The signifieds can be any music-related concept determined by the user. The signifiers consist of the text characters commonly available on computers.

Humdrum regards each file as a two-dimensional plane, much like a sheet of paper. *Successions* of events proceed vertically down the page, whereas *concurrent* events extend horizontally across the page. Two signifiers that occupy the same horizontal line represent concurrent (or overlapping) events. The basic organization of Humdrum files may be schematically illustrated as follows:

	successive events ↓			
concurrent events →	A	J	V	etc.
	B	K	W	etc.
	C	L	X	etc.
	D	M	Y	etc.
	etc.	etc.	etc.	

Types of Records

Humdrum encodings consist of a set of one or more lines or *records.* There are three types of Humdrum records:

1. comment records,

2. interpretation records, and
3. data records.

These three record types are mutually exclusive, so it is not possible to mix comments, interpretations, or data records on the same line.

Comment Records

As we noted in Chapter 2, there are two kinds of comments: global comments and local comments. *Global comments* pertain to all concurrent parts, whereas *local comments* pertain to some specific column of data (such as a particular staff, instrument, note, finger, etc.). Comments are lines that contain an exclamation mark (!) at the beginning of the record (in the left-most position); subsequent characters up to and including the occurrence of a carriage return or newline character constitute the comment record. Recall that global comments are denoted by two exclamation marks (!!) at the beginning of the record. Global comments may contain any sequence of printable characters, including 'blank space' such as tabs and spaces. Local comments may contain any sequence of printable ASCII characters, with the important exception of the tab character which is used to separate spines. Comments may be used to insert free-format commentaries in Humdrum encodings.

Interpretation Records

Interpretations are lines that begin with the asterisk character (*). Interpretations are used to identify more precisely the state of the representation — for example, to indicate that an encoded part is for a transposing instrument in E-flat, or to indicate that the representation is for a given Balinese tuning, or that the representation encodes a conductor's physical gestures. Humdrum requires that at least one interpretation must be specified before any data records are encountered. The difference between a comment and an interpretation is that interpretations are formal, potentially *executable* statements; interpretations pass information to programs that process the Humdrum encoding.

As in the case of comments, there are two types of interpretations: *exclusive* interpretations begin with a double asterisk (**) whereas *tandem* interpretations begin with a single asterisk (*). Exclusive interpretations are mutually exclusive: only one such interpretation can be active at a given time for a given string of data. No set of data is complete without the presence of an exclusive interpretation. Tandem interpretations, by contrast, provide supplementary information about how a set of data is to be interpreted. Several tandem interpretations may pertain to a given set of data; unlike exclusive interpretations, tandem interpretations are not necessarily mutually exclusive.

Data Records

Lines that do not contain either an exclamation mark or an asterisk in the first column are *data records.* Blank lines (i.e., lines which are either empty, or contain only blank space, such as tabs and spaces) are forbidden in Humdrum. Thus data records may be formally defined as non-empty lines that do not begin with either an exclamation mark or an asterisk.

In Humdrum, each data record encodes information pertaining either to a particular moment in time or to a particular time window or duration. (Whether a record represents a precise moment or an expanse of time depends on the accompanying interpretation.) Each data record may contain one or more data *tokens.* When more than one token is present, tokens are separated from each other by tabs. When several data records are present, multiple tokens are aligned in columns through the file. As we noted earlier, these columns are referred to as *spines.* By itself, a spine has no particular meaning; it is simply a way of linking together related tokens through time. Spines become meaningful only when they are labelled by adding an interpretation.

By itself, Humdrum recognizes only six ASCII characters. Two of these characters — the exclamation mark (!) and the asterisk (*) — have a special meaning *only* when they appear in the first column of a record (or are preceded by a tab; see below). The remaining special Humdrum characters are the period (.), the space, the tab character, and the carriage return (= newline character). As we have seen, the exclamation mark and the asterisk are used to identify comments and interpretations, respectively. The tab and carriage return characters are used to format the data into *spines* and *records,* respectively.

Data Tokens and Null Tokens

As we noted above, the data in the data records are conceptually divided into tokens. In Humdrum, there are two possible types of tokens:

1. *data* tokens, and
2. *null* tokens (.).

Consider, for example, the following file:

```
X       .       X
X       X       X
.       X       X
X       .       X
```

This file consists of three vertical spines and four horizontal records. The first and third spines begin with data tokens, while the second spine begins with a null token. Without the presence of interpretations, the meaning of this file is indeterminate. The file below contains two spines that have been labelled using Humdrum interpretations:

```
**left  **right
X       .
.       X
X       .
.       X
X       .
*-      *-
```

The user has defined two interpretations: "left" and "right." The intention is to represent the footfalls of a person's left and right feet. The representation simply encodes that the left and right feet have alternating events, such as might be produced by walking or running. Notice that null tokens (.) indicate nothing at all and merely act as place-holders to maintain the format of the two spines. Notice also that interpretations must be defined for each spine, and that each interpretation consists

of some keyword appended to the double asterisks (e.g. `**left`). No intervening spaces are permitted between the interpretation *keyword* (`left`) and the asterisks; however, spaces may appear as part of the keyword itself. In addition, when more than one spine is present, both the data tokens and the associated interpretations must be separated by a tab character; spaces cannot be used to separate spines. Finally, note that each spine is formally terminated by a *spine-path terminator* — an asterisk followed by a minus sign.

Interpretations can be cascaded so that a single spine has more than one interpretation associated with it. This is done through the addition of tandem interpretations. Consider the following example:

```
**foot   **foot   **arm   **arm
*left    *right   *left   *right
X        .        .       X
.        X        X       .
X        .        .       X
.        X        X       .
X        .        .       X
*-       *-       *-      *-
```

In this case the categories "left" and "right" have been transformed to tandem interpretations. The first spine is interpreted both as "left" and as "foot." The exclusive interpretation (double asterisks) takes conceptual precedence over the tandem interpretation (single asterisk). That is, tandem interpretations merely modify or supplement the exclusive interpretation. Hence, given the above representation, we could say that "left" is an attribute of "foot" or "arm," but we could not say that "foot" is an attribute of "left."

Users are free to define as many different exclusive and tandem interpretations as they wish. For example, a user might define the interpretation `**bowing` that would be suitable for encoding detailed bowing information in works for strings. For each exclusive interpretation, the Humdrum user can re-define the meaning of all of the text characters, with the exception of the tab and the carriage return, which always retain their functions as 'token/spine separator' and 'record separator' respectively. The characters ! . * can also be re-defined, although there are some restrictions as to how they can be used. Specifically, the exclamation mark cannot occur in the first column of the record unless it is used to indicate a comment. Similarly, the asterisk cannot occur in the first column of a record unless it is used to indicate a Humdrum interpretation. The period cannot appear in the first column unless it is used to indicate a null data token. In addition, the exclamation mark, asterisk, and period cannot appear following a tab unless they are used to indicate a comment, interpretation, or null token, respectively.

Data Sub-Tokens

Data tokens can be split into sub-tokens via the space character. In the first data record of the following example, the first spine contains two sub-tokens whereas the third spine contains three sub-tokens. Sub-tokens do not have their own spine organization and can appear and disappear as necessary:

```
**spine1    **spine2  **spine3
A B         J         X Y Z
AB          J         XYZ
A B C       .         X Z
*-          *-        *-
```

Data sub-tokens are useful in a variety of circumstances. An appropriate use of sub-tokens might be to encode double- and triple-stops in string parts.

In the Humdrum data records, the space character is reserved solely for use as a sub-token delimiter. Note that consecutive spaces are illegal, and that data tokens cannot begin or end with a space character. Of course spaces can be used freely in comments and in interpretations.

Spine Paths

Humdrum representations often consist of a fixed number of spines that continue throughout the course of an encoded file. As we have seen in the preceding chapters, a typical use of spines is to encode different voices or parts in a musical work. However, there is no reason to equate spines with voices; spines are used for many other purposes as well.

In encoding Humdrum representations it is occasionally useful to be able to vary the number of spines. However, files with varying numbers of spines can pose significant questions of interpretation. Consider, for example, the following sequence of Humdrum-like data records:

```
1       2       3
1       2       3
1       2       3
A       B
A       B
A       B
```

At the point where three spines are reduced to two spines the continuity is ambiguous: Has spine '3' been discontinued? Or is spine 'B' a continuation of spine '3' with spine 'A' a continuation of spine '1' or '2'? For some representations such questions will be of little concern; however, in other circumstances, the manner in which the spines continue will be of critical importance. For example, if all of the above spines encoded pitch information for various musical parts, a study of melodic intervals would need to resolve the specific melodic paths as the representation moves from three to two spines. Failure to clarify the pitch paths would make it difficult to determine or search for specific successions of melodic intervals.

The Humdrum syntax provides special *spine path indicators* that make it possible to resolve such ambiguities and to ensure that the continuity (or lack of continuity) is made clear. Humdrum provides five special path indicators, one of which we have already encountered:

- a new spine may be introduced
- an existing spine may terminate (without continuing further)
- a previous spine may be split into two spines

- two or more spines may be amalgamated into a single spine
- the positions of two spines may be exchanged

Spine path indicators use the following signifiers: the plus sign (add a spine), the minus sign (terminate a spine), the caret (split a spine), the lower-case letter 'v' (join spines), and the lower-case letter 'x' (exchange spines). In addition to these, a *null interpretation* exists whose purpose is merely to act as a place-holder in interpretation records:

`*+`	add a new spine (to the right of the current spine)
`*-`	terminate a current spine
`*^`	split a spine (into two)
`*v`	join (two or more) spines into one
`*x`	exchange the position of two spines
`*`	null interpretation (place holder)

Spine Path Interpretations

Spine paths are types of interpretations, so the spine path indicators are encoded as Humdrum interpretations, using the asterisk signifier (*). The following examples illustrate a few possible path changes:

```
1   2     3
*   *-    *          (elimination of spine #2)
1   3

1   2     3
*   *x    *x         (exchange spines #2 and #3)
1   3     2

1   2     3
*   *^    *          (splitting of spine #2)
1   2a    2b   3

1   2     3
*   *v    *v         (amalgamation of spines #2 and #3)
1   2&3
```

Notice that in cases where two or more spines are amalgamated, the spines must be adjacent neighbors. For example, the arrangement below is forbidden by the Humdrum syntax since it is not clear whether spines #1 and #3 amalgamate into spine 'A' or spine 'B'.

```
1    2   3
*v   *   *v        (syntactically illegal)
A    B
```

In such cases, amalgamating the two outer spines can be accomplished by first using the exchange path signifier. Here we exchange spines #2 and #3 before amalgamating the original first and third spines:

```
1     2    3
*     *x   *x
*v    *v   *
1&3   2
```

In cases where the user wishes to amalgamate several spines, a number of interpretation records may be necessary. In the following example, spines #1 and #2 are first joined together (momentarily defining three spines: 1&2, 3, 4). In the subsequent interpretation record, spine #2 (previous spine #3) and spine #3 (previous spine #4) are then joined:

```
1     2     3    4
*v    *v    *    *
*     *v    *v
1&2   3&4
```

In addition, it is possible to join more than two spines at the same time:

```
1           2       3       4
*v          *v      *v      *v
1&2&3&4
```

In cases where a new spine is introduced, it is essential to indicate the exclusive interpretation that applies to the new data. Thus an 'add spine' indication must be followed by a second interpretation record:

```
1   2    3
*   *+   *         *     (add a new spine.)
*   *    **inter  *     (define exclusive interpretation for the
1   2    new      3      new spine.)
```

Failing to follow the introduction of a new spine by a subsequent exclusive interpretation is illegal.

The following examples illustrate a variety of more complex path redefinitions:

```
1     2    3    4
*v    *v   *^   *^
1&2   3a   3b   4a   4b
```

```
1       2    3    4    5
*       *-   *    *-   *
*v      *v   *v
1&3&5
```

```
1     2    3    4    5
*     *-   *    *^   *+
*     *    *    *    *      **new
*v    *v   *    *    *      *
1&3   4a   4b   5    new
```

```
1    2    3    4
*x   *x   *    *
*    *x   *x   *
*    *    *x   *x
2    3    4    1
```

Note that with judicious planning, the user can completely reconfigure all spines within a Humdrum file.

Syntactically, some path constructions are illegal; here are some examples of illegal constructions:

```
1    2        3
*v   *        *v
```
(The join-spine indication in spine #1 does not adjoin spine #3.)

```
1    2        3
*x   *x       *x
```
(No more than two exchange interpretations at a time.)

```
1    2        3
*x   *        *
```
(Must have two exchange interpretations together.)

```
1    2        3
*v   *        *
```
(Must have two or more join interpretations at a time.)

```
1    2        3
*    *
1    2
```
(Spine eliminated without using a termination interpretation.)

```
1    2        3
*    *        *+
1    2        3
```
(Adding a new spine should result in 4 interpretations.)

```
1    2
*    *        *-
```
(Cannot eliminate non-existent spine.)

```
1    2
*+   *
1    new      2
```
(New spine started without specifying new interpretation.)

```
1    2
*    *+
*    **inter  *
A    B        C
```
(Interpretation labels the wrong spine.)

The Humdrum Syntax: A Formal Definition

With the preceding background it is now possible to define formally a Humdrum representation. First we can define a Humdrum file. A Humdrum file must conform to one of the following:

1. A file containing *comments, data records* and *interpretations* with the restriction that no data record or local comment appears before the first *exclusive interpretation.*
2. A file containing *data records* preceded by at least one *exclusive interpretation.*
3. A file containing only *comments* and *interpretations* with the restriction that no local comments appear before the first interpretation.
4. A file containing only *interpretations* beginning with an exclusive interpretation.
5. A file containing only global *comments.*
6. A totally empty file (i.e. a file containing no records).

In addition, each spine in a Humdrum file must ultimately end with a path terminator (*–). Only global comments (or new exclusive interpretations) may occur following the termination of all spines. A property of Humdrum files is that the concatenation of two or more Humdrum files will always result in a Humdrum file.

Additional interpretations may be added throughout the file. Global comments may appear anywhere in the file. However, local comments are much more restricted: (1) Local comments may not appear until after the first interpretation record, and (2) The number of sub-comments in a local comment record must be equivalent to the number of currently active spines.

Comment	Either a global or local comment. Any record beginning with an exclamation mark.
Global comment	Any record beginning with two exclamation marks (!!).
Local comment	Any record beginning with one and only one exclamation mark (!). Every spine in that record must also begin with an exclamation mark.
Null comment	A comment record containing no commentary; only the appropriate exclamation mark(s) are present.
Interpretation	Either an exclusive or tandem interpretation. Any record beginning with an asterisk (*).
Exclusive interpretation	Any record beginning with one or more asterisks (*), where at least one spine begins with two asterisks.
Tandem interpretation	Any record beginning with a single asterisk (*) where none of the spines begin with two asterisks.
Path indicator	One of five special tandem interpretations *+ *- *v *^ *x found only in tandem interpretation records.
Null interpretation	An interpretation for a given spine or spines consisting of just the interpretation signifier (i.e., a single asterisk).
Data record	Any record that is not a comment or interpretation. Must contain the same number of tokens as the number of current spines.
Null token	The period (.) either alone on a single record or separated from other characters by a tab. Appears only in data records.
Null data record	A data record consisting only of null tokens.
Spine	A column-like "path" of information — including data records, local comments, and interpretations.

Humdrum Terminology

As a supplement to the above "positive" definition of the Humdrum syntax, we can also describe various inputs that do *not* conform to the Humdrum syntax:

An empty record.
A record containing only tabs.
A record beginning with a tab.
A record ending with a tab.
Any record containing two successive tab characters.
Any data record having fewer or more spines than the immediately preceding data record.
A record having only one join-spine indication.
A record having only one exchange-spine indication.
A record having more than two exchange-spine indications.

Some Illegal Humdrum Constructions

The *humdrum* Command

One of the most important commands in the Humdrum Toolkit is the **humdrum** command itself. This command is used to identify whether a file or other input stream conforms to the above Humdrum syntax. Where appropriate, the **humdrum** command issues error messages identifying the type and location of any syntactic transgressions. If no infractions are found, **humdrum** produces no output (i.e., in UNIX parlance "silence is golden"). All of the commands in the Humdrum toolkit assume that the inputs given to them conform to the Humdrum syntax. Whenever you encounter a problem, you should always test the input to assure that it is in the proper Humdrum format.

The examples given below provide further illustrations of Humdrum representations:

```
**form
Introduction
Exposition
Development
Recapitulation
Coda
*-

**American    **British
quarter       crotchet
eighth        quaver
dotted half   dotted minim
*-            *-

**Opus/No     **Year
23/1          1821
23/2          1821
23/3          1822?
24            1822
*-            *-
```

```
**recip   **diaton **accidental**stem-dir **kern
4         c        #           /          4c#/
8         d        .           /          8d/
8         e        .           /          8e/
2         f        #           /          8f#/
*-        *-       *-          *-         *-

**heart-rate
74
73
74
77
78
*-

**foreground
flute
*^
flute          violin1
*-             *
violin1
*^
violin1        bassoon
*              *^
violin1        bassoon  'cello
*              *        *^
violin1        bassoon  'cello trombone
*-             *-       *-     *
trombone
*^
trombone       trumpet
*-             *-
```

Reprise

This chapter has identified the formal structural and organizational features of the Humdrum syntax. The syntax provides a framework within which sequential symbolic data can be represented. Individual representation schemes map the ASCII character set (signifiers) to various music-related concepts (signifieds).

Each representation is designated by an exclusive interpretation. The corresponding data are organized in spines that may meander throughout the file. New spines may be added, spines joined together, exchanged, split, or terminated. Data are organized as tokens, although tokens can consist of multiple subtokens separated by single spaces. Null tokens can appear as place-holders where no specific data exists.

Free-form comments may be interspersed throughout the file. Global comments pertain to all spines whereas local comments pertain to individual spines. Additional interpretive information may be encoded using tandem interpretations. Both local comments and tandem interpretations may occur anywhere, but must be preceded in the spine by some exclusive interpretation.