Beyond MIDI: The Handbook of Musical Codes

MuseData §1: Composite File Organization

Walter B. Hewlett

The organization of *MuseData* files is an integral part of the *MuseData* representation. Each *MuseData* file represents the encoding of one musical part from a movement from a composition. In the following scheme one file would contain the information required for Part 1, Bars 1-11, another Part 2 (Bars 1-11), another Part 3 (Bars 1-11), and so forth.
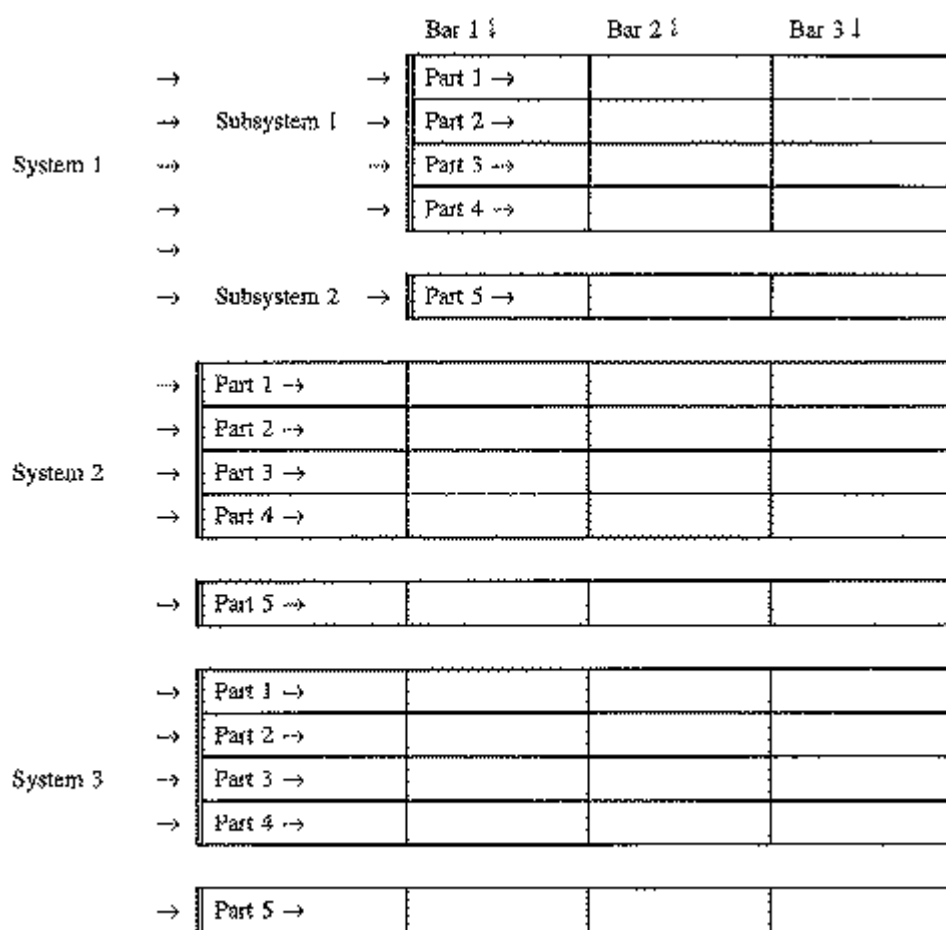

Figure 27.1 A hypothetical score viewed as a two-dimensional array.

The arrangement of parts into systems and subsystems and the layout of system breaks and page breaks is irrelevant to the encoding of *MuseData*. However, a musical part may be notated as one line or more of music. For example, if a movement has two oboe parts, Oboe I and Oboe II, these may be encoded as separate parts, or they may be combined on one staff and encoded as a single musical part, namely Oboes I & II (e.g., Parts 1 and 2 above). In the latter case, verbal cues and directions would be used to differentiate the two parts.

In the *MuseData* system, they may be encoded both ways, since a score might call for both oboes on one staff, but the players might want to play from separate parts. Music on the grand staff may be encoded as one or two parts. If musical notation or symbols cross between the staves of the grand staff, then the music on the grand staff must be treated as one musical part.
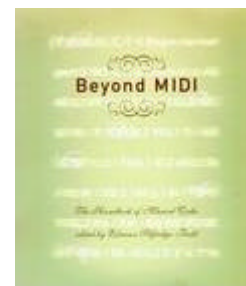
1.1 File Relationships within the Database

Our database currently consists of more than 7,000 *MuseData* files. When complete, the database could exceed 100,000 files. We currently use a hierarchical directory tree structure to organize these files, which is briefly outlined below. In this outline, you must imagine real composer names for *COMPOSER1*, *COMPOSER2*; real source names for *SOURCE1*, *SOURCE2*; and real work titles for *WORK1*, *WORK2*.

Stage-1 files contain data for pitch and duration only (i.e., note and rest records) and support sound applications. Stage-2 files contain a great deal of additional information to support printing and interpretive applications.

Stage-1 files are all of the same type, while Stage-2 files may belong to one or more of the following data-file types:

- *sound:*   used to compile MIDI sound files
- *score:*   used to print scores
- *parts:*   used to print parts
- *short:*   used to print short scores
- *tracks:*   used for analysis
- *midi:*   data specifically used to compile MIDI files (*e.g.*, channel assignments and instrument assignments)
- *data:*   anything and everything

```
Figure 2: Hierarchy of file relationships within database.
   MuseData/
         COMPOSER1/
         COMPOSER2/
         . . .
         COMPOSERn/
               INDEX
               SOURCE1/
               SOURCE2/
               . . .
               SOURCEn/
                     WORK1/
                     WORK2/
                     . . .
                     WORKn/
                           STAGE1/
                                 MVT1/
                                 MVT2/
                                 . . .
                                 MVTn/
                           STAGE2/
                                 MVT1/
                                 MVT2/
                                 . . .
                                 MVTn/
```

1.2 Organization of Single Files

MuseData files consist of a set of time-ordered, variable length ASCII records. The order of the records is essential to the representation. Single *MuseData* files have four essential components header records, a musical attributes record, a series of regular note records, and an end-of-file marker.

All records are organized as a series of 80-character columns. The first character in each record is called the *control key*, and this key determines the nature and function of the record.

A. Header Records

The first 12 lines of a *MuseData* file contain general information about the entire file. The format and contents of the header records are as follows:

Record 1:	free
Record 2:	free
Record 3:	free
Record 4:	*<date> <name of encoder>*
Record 5:	WK*n*:*<work number>* MV*n*:*<movement number>*
Record 6:	*<source>*
Record 7:	*<work title>*
Record 8:	*<movement title>*
Record 9:	*<name of part>*
Record 10:	miscellaneous designations (*e.g.*, *<mode>*, *<movement type>* and *<voice>*)
Record 11:	group memberships: *<name1> <name2>* ...
Record 12:	*<name1>*: part *<x>* of *<number in group>*
Record 13:	...
...	(as needed)

Additional header records may be used, but those shown above are always used or, if not pertinent to the file at hand, reserved.

Files that are distributed contain copyright and other identifying information in Records 1-3.

B. Group Memberships [Record 11]

The concept of group membership for parts facilitates flexibility in the creation of diverse kinds of scores and parts and in other uses of the data. Record 11 contains a list of names, which are the application groups to which this file belongs. The group names can be anything the encoder wishes them to be.

In our data files, we use names that have an obvious connection to their associated application: *e.g.*, score, parts, sound, tracks, shortscore, data, etc. For each name listed in Record 11, there is a record that starts with that name and specifies the order or ranking in the group to which it belongs. The idea is that all files for a particular movement should be placed in the same directory. This way, a program designed to compile a score would simply check every file in the directory and identify for use those belonging to the group called *score*.

The order in the group would determine the top-to-bottom order in the score. A program designed to compile MIDI sound files would identify for use those belonging to the group called *sound*. The same method would apply for other applications such as printing parts, printing short scores, and compiling track data for melodic and harmonic analysis.

C. The Musical Attributes Record [$]

A musical attributes record immediately follows the last header record. In the above

example it would constitute Record 13. This record identifies such attributes, usually pertaining throughout the file, as key signatures, time signatures, clef signs, etc. Musical attribute records always begin with a dollar sign ($) in Column 1. Their format is as follows:

Column 1:	$
Column 2:	level number (optional)
Column 3:	footnote column
Columns 4-80:	attribute fields

The record may contain one or more fields; fields are initiated by the field identifier and terminated by a blank. In the case of clefs and directives, the field identifier may contain a number, which is the staff (1 or 2 at the moment) to which the clef or directive belongs. The absence of a number indicates Staff 1.

Field type	Field identifier	Field data type
key	K:	integer (positive or negative)
divisions per quarter note	Q:	integer (positive only)
time designation	T:	two integers (positive only)
clef	C:	integer (positive only)
clef	Cn:	integer
transposing part	X:	integer (positive or negative)
number of staves for part	S:	integer (positive only; default = 1)
number of instruments represented	I:	integer (positive only; default = 1)
directive (last field on line)	D:	ASCII string
directive (last field on line)	Dn:	ASCII string

Here is one example of a musical attributes record:

```
K:-2  Q:8  T:3/8  C:4  C2:22  D:Allegro ma non troppo
```

Explanations of the parameters for the less self-evident data types represented in this record follow.

1. *Clef Code [C:]* The standard clefs are represented by a positive integer between 1 and 85. The tens digit of the code specifies the clef sign and the ones digit specifies the staff line to which the clef sign refers. The clef-sign codes are as follows:

0	G-clef
1	C-clef
2	F-clef
3	G-clef transposed down
4	C-clef transposed down
5	F-clef transposed down
6	G-clef transposed up
7	C-clef transposed up

8	F-clef transposed up

The line-number designations are as follows:

1	highest line
	...
5	lowest line

Some examples of clef codes:

04	treble clef
13	alto clef
22	bass clef
34	treble clef for tenors

2. ***Transposing Part*** **[X:]** This integer (positive or negative) indicates a transposing interval (if one exists) and/or a doubling of the part an octave lower. The base-40 system is used. 23 means the music sounds a fifth higher than it is written; 23 means the music sounds a fifth lower than it is written. Adding 1000 to the number indicates a doubling of the part an octave lower (e.g., violoncello and bass on the same part or 8' and 16' sound on an organ pedal line). In the base-40 system, the minor second has a value of 5 and the major second has a value of 6. All other interval sizes can be computed from these numbers.

3. ***Number of Instruments Represented*** **[I:]** This integer (1 or more) indicates the number of independent instruments represented in the file. If this number is more than one, these printing conventions will hold:

   A. Notes with the same stem direction will be combined into one chord.
   B. If more than one voice is represented in a measure on a staff, then each voice will follow its own set of accidentals within the measure.