

THE I - FILE FORMAT

=====

- 1.1 I-Files come in two types: linear and page specific. Linear files represent one voice or part in its entirety. Page specific files represent the music on a single page. We call this a music page file.
- 1.2 A music page file is basically a list of glyphs (characters) from a set of music fonts. The musical notation is constructed from this list of glyphs. If you are printing music, the music fonts must first be downloaded to the printer; if you are displaying music, the display utility puts the glyphs on the screen in the same manner that the printer puts them on the page.
- 1.3 To construct a page of music, you need to have a list of glyphs (characters) together with their location on the page. A musical note, for example, consists of a note head, one or more pieces of stem, possibly a flag, possibly an accidental, possibly some leger lines, etc., etc..
- 1.4 Since the HP PCL3 printer language is a full page language, it doesn't matter what order our list of glyphs is in. But for purposes of editing, some choices of ordering are better than others. Let us imagine, for example, that we want to move the musical note described in the previous paragraph to a new (x,y) position on the page. In order for the note to stay together, we must move all parts of it by the same amount. It makes sense, therefore, to think of the note as a single unit, even though it has several parts.
- 1.5 To implement this concept, we define something we call an **object**. The object is the basic unit of musical notation. The object may consist of several glyphs, as in the case of a note. We call each of these glyphs, **sub-objects**, because they are members of the thing we call an object. The position of sub-objects is specified in relation to the object to which they belong. This way, if we want to move a note, we simply move the object, and all the parts of it (the sub-objects) will move together.
- 1.6 All basic units of musical notation (objects) are attached to a **staff line** (in our system of representation). It therefore makes sense to specify the location of an object in relation to the staff line to which it belongs. This way, if we want to move a staff line, all notes and other notation connected with that line will move together.
- 1.7 All staff lines belong to a **system**. A system may have one or more staff lines associated with it. As before, it makes sense to specify the location of a staff line in relation to the system to which it belongs.
- 1.8 Systems represent the highest level in the location hierarchy on the page. The location of a system is therefore specified by absolute (x,y) co-ordinates on the page.
- 1.9 To summarize, let us take the example of a sharp (#) attached

2
to a note. The absolute (x,y) of that sharp will be:

The absolute location of the system (x,y), plus
the (dx,dy) offset to the staff line in the system, plus
the (dx,dy) offset to the note on the staff line, plus
the (dx,dy) offset to the sharp (#) from the note.

1.10 Much of musical notation can be represented in the manner described thus far: notes, rests, text attached to a note, musical ornaments, single articulation marks, dynamics, musical directions, bar lines, time signatures, key signatures, clefs, basically anything that stands by itself on the musical page. There is a class of things, however, that cannot be represented as objects, because the position of these things depends on the position of more than one object. Items in this class include: ties, beams, slurs, endings, long trills, transposition lines, figure continuation lines, tuplets (and their brackets), dynamic wedges, and dashes associated with changes in dynamics, tempo, etc. We call this class of things, **super-objects**, because their position (and shape) depend on more than one object.

1.11 In actual fact, most super-objects are printed using glyphs from the music fonts; however, they are not represented in the music page file in this manner, because, if we move a object which has a super-object associated with it, the position and the shape of that super-object (and therefore the glyphs that comprise it) will change.

2.1 A music page file consists of list of variable length records. The order of the records is an integral part of the representation. There are n types of records. The character in column of a record identifies its type.

Record Type	Identifier (column 1)
Header Record	Z
Page oriented text	X
System (page oriented)	S
Staff Line	L
Objects	J
Sub-objects	K
Text (form of sub-object)	T
Words (form of sub-object)	W
Attribute (form for sub-object)	A
Super-objects	H
End of Music Line	E
System Bar	B

2.2 The basic location hierarchy is: System - Line - Object - Sub-object. The order of records in the music page file reflects this hierarchy. A system record remains active in the list until a new system record is encountered. A staff line record remains active until an End of Music Line record is encountered. Objects on a staff line are listed in the time order (left to right) in which they occur on the staff line. The sub-objects that belong to an object are listed directly

below the object. System bars are listed after the last staff line of a system has been fully represented. Page oriented text should be listed before or between systems. Header records belong at the beginning of the file.

Record Formats

Each record in the file consists of two or more fields. All fields in a record are delimited by a single space. Fields which contain text usually occur at the end of a record. In specifying (x,y) locations and offsets, x increases as you move to the right; y increases as you move down. This is not the normal convention for (x,y) co-ordinates. All distances and offsets are measured in dots. For the moment, all music is represented at 300 dots = 1 inch.

I. Header Information (optional for display and printing)

- Field 1: Identifier = Z
- Field 2: information type (a number from 1 to 7)
- Field 3: actual data

Examples

=====

1. In this example we have the full score of Beethoven's 5th Symphony, sub-divided by movements. This file represents the first page of the first movement. The resolution is 300 dots/inch, and the interval between staff lines is 14 dots.

Field 2	Field 3
-----	-----
1 = Composer:	Ludvig van Beethoven
2 = Work:	Symphony No. 5 in C minor
3 = Sub-division:	Movement 1
4 = Representation:	Full Score
5 = Resolution:	300 dots/inch
6 = Notesize:	14 dots per staff line
7 = Page:	1

2. In this example we have a set of parts for Beethoven's 5th Symphony, sub-divided by instruments. This file represents the first page of the first violin part.

Field 2	Field 3
-----	-----
1 = Composer:	Ludvig van Beethoven
2 = Work:	Symphony No. 5 in C minor
3 = Sub-division:	Violin I
4 = Representation:	Complete Parts
5 = Resolution:	300 dots/inch
6 = Notesize:	14 dots per staff line
7 = Page:	1

4

3. In this example we have the first violin part for Beethoven's 5th Symphony, sub-divided by movements. This file represents the first page of the first movement.

Field 2	Field 3
1 = Composer:	Ludvig van Beethoven
2 = Work:	Symphony No. 5 in C minor
3 = Sub-division:	Movement 1
4 = Representation:	Violin I
5 = Resolution:	300 dots/inch
6 = Notesize:	14 dots per staff line
7 = Page:	1

The point of these examples is to show various ways in which a data set can be sub-divided. At the present time, only one level of sub-division is possible. In the second example above, we cannot sub-divide the Violin I part by movements, because the Violin I is already a sub-division of the category, "Complete Parts." If we want to sub-divide the instrumental parts by movements, then we must have a separate data set for each instrumental part.

(End of Header Information)

II. Page oriented text

Field 1: Identifier = X
Field 2: font number
Field 3: x co-ordinate of beginning of text
Field 4: y co-ordinate of text line
Field 5: ASCII text

III. Systems (page oriented)

Field 1: Identifier = S
Field 2: 0
Field 3: x co-ordinate of the system on the page
Field 4: y co-ordinate of the system on the page
Field 5: horizontal length of the system
Field 6: vertical length of the system
Field 7: number of lines (L) in the system
Field 8: control string in double quotes (null string allowed)

The control string describes how the lines of the system are bound together on the left, and where bar lines stop and start between the staff lines. The control string consists of the following elements:

[= start bracket
] = end bracket
{ = start brace
} = end brace
(= start bar line
) = end bar line
. = part with a single staff line

5

: = part with a grand staff (piano)

The control string for a score of a piano concerto with four woodwinds, timpani, piano and strings might look as follows: "[.....)(.):{...}....]"

(End of Systems)

IV. Staff lines (system oriented)

Field 1: Identifier = L

Field 2: y offset in the system. (y = 0 for top line)

Field 3: text offset(s) from line (up to 10, separated by |)

The text offset is the vertical distance between the top line of the staff and the base line of the text. We are talking here about text underlay, i.e., text that goes with the music. The representation permits up to 10 lines (verses) of text.

Fields 4--7: parameters carried over from this line (musical part) on previous system.

Field 4: dyoff(s)

Field 5: uxstart(s)

Field 6: backloc(s)

Field 7: xbyte(s)

The parameters in fields 4 to 7 relay certain information required for the printing of text underlay. If this part on the previous system of music (which might be on a previous page and therefore not even part of data in this file) ended with text that was being extended by dashes (- - -) or a heavy line (_____), these dashes or the heavy line would need to be continued under the present line of music. xbyte, contains the terminating punctuation for a heavy line.

In the case of fields 4, 5 and 6, the parameters are numeric and if there is more than one line of text, there will be multiple parameters, separated by |. Field 7 is a character, and if there is more than one line of text, there will be multiple characters in this field.

Field 8: y offset to second line of grand staff line (0 = none)

Field 9: notesize (optional; 0 = no information supplied)

(End of Staff lines)

V. Objects

Field 1: Identifier = J

Field 2: type of object (a letter code)

B = bar line

C = clef

K = key signature

T = time signature

6

D = directive (e.g. time word, etc)
 S = other symbol
 N = note
 R = rest
 G = grace note
 Q = cue note
 F = figures
 I = isolated directive or symbol (not associated with a note, rest, or figure)
 M = mark (a dummy object used mostly for starting or ending super-objects such as ties or slurs that extend beyond the line)

Field 3: object code (# = number)

The meaning of this number depends on the type of object.

Object type

Bar line: # = measure number if there be one, otherwise zero.

Clef: # = clef code

4 = treble clef (G-clef on line 4)
 13 = alto clef (C-clef on line 3)
 12 = tenor clef (C-clef on line 2)
 22 = base clef (F-clef on line 2)
 34 = transposed treble clef (tenor voice clef)

Key: # = key code

0 = no flats or sharps
 1 = 1 sharp (etc.)
 -1 = 1 flat (etc.)

Time sig: # = time code (100 * tnum + tden)

tnum = time numerator
 tden = time denominator
 special cases:
 1 1 = common time
 0 0 = alle breve
 2 0 = simple 2
 3 0 = simple 3

Directive: # = print flag

0 = print always (parts and score)
 bit 0 set = print in parts
 bit 1 set = print in score
 bit 2 set = print if top part in score
 bit 3 set = print if bottom part in score
 (bit 1 over rides bits 2 and 3)

Symbol: # = symbol type

0 = not specifically identified
4 = extended rest
6 = whole measure rest

From the standpoint of typesetting for a score the major difference between a directive and a symbol is that a symbol will always be printed whether or not the part stands alone or is part of a score (e.g., letter dynamics), whereas a directive can be "made invisible" in parts that are not at the top or the bottom of a score (e.g., the segno sign).

Notes: # = note type
Rests: # = rest type
Grace note: # = note type
Cue note: # = note type

Note types

11 = longa	5 = sixteenth note
10 = breve	4 = 32nd note
9 = whole note	3 = 64th note
8 = half note	2 = 128th note
7 = quarter note	1 = 256th note
6 = eighth note	0 = undefined

Figure: # = 0 always
Isolated: # = same as for Directives
Mark: # = 0 always

Field 4: horizontal displacement from beginning of line

For objects which are groups of notes (chords), the x co-ordinate is the position of the notes which fall on the proper side of the stem. For stem up, the object's position is that of the note heads to the left of the stem; for stem down, the object's position is that of the note heads to the right of the stem.

Field 5: vertical displacement from the first line of the staff.

For objects which are groups of notes (chords), the y co-ordinate is the position of the note which is farthest from the note-end of the stem (and hence closest to a beam, if there be one). This is important for the proper setting of beams at print time.

For objects to be placed on the lower staff of the grand staff, 1000 will be added to the displacement. This means that the print program must check the

8

range of the displacement before placing the object.

For Bar type objects, which always have a y co-ordinate of 0, this field contains instead the bar line code:

- 1 = single light bar
- 2 = single heavy bar
- 3 = dotted bar
- 5 = double light bar
- 6 = light-heavy double bar
- 9 = heavy-light double bar
- 10 = heavy-heavy double bar

In case the bar object extends to the lower staff of the grand staff, 1000 is added to the bar line code.

Field 6: print code (or if < 32, number of sub-objects)

If this object can be printed with one glyph (character), then this field contains the glyph number (which will be 32 or greater). Otherwise, this field contains the number of sub-objects required to represent this object.

Field 7: space node number

This field is used to line up notes in a score. The measure is divided into 6912 (27 x 256) parts. Divisions run from 1 to 6912. Each object is assigned the division number which most closely approximates its time location in the measure.

Field 8: distance increment flag.

This flag describes the distance from the preceding object, as well as an attribute of the object.

0 = preceding distance should remain fixed at print time

= preceding distance may vary at print time; parameter is the notated duration of the preceding object on this line, measured in units such that 576 = quarter note.

10000 = object will be centered between bar lines (used only with whole measure rests)

Fields 7 and 8 do not effect the actual display and printing of the music, but they play an important role in the automatic editing of music (a possible future application for Dmuse).

Field 9: number of super-objects associated with this object

Fields 9a,b,...,: super-object numbers

A Super-object record will appear in the file after all objects that relate to the super-object have appeared. For example, a group of notes (note objects) may be connected by a beam. After the last of these note objects has appeared as a record in the file, the super-object record representing the beam itself will appear.

(End of Objects)

VI. Sub-objects (these must follow directly after their associated objects)

- Field 1: Identifier = K
- Field 2: horiz. displacement, relative to object co-ordinate
- Field 3: vertical displacement, relative to object co-ordinate
- Field 4: print code of sub-object (glyph number)

VII. Text (a form of sub-object attached to a note or rest)

- Field 1: Identifier = T
- Field 2: horizontal displacement of text, relative to object
- Field 3: vertical code (text line number between 1 and 10)

The vertical code may be followed by the "|" character and a y offset. The interpretation of the offset is a displacement from the normal position of the line. This allows us to move individual words (syllables) up or down relative to the normal height of the line.

Field 4: ASCII string (blanks represented by "_" character)

If this field = "~" (the tilda character), this means that this note is the termination of a forward underline.

Field 5: single byte field indicating presence of a forward hyphen or underline character

Byte code	Meaning
-	forward hyphen
_	forward underline character alone
.	" " " followed by .
,	" " " " "
:	" " " " "
;	" " " " "
!	" " " " "
?	" " " " "
*	no forward hyphen or underline (normal case).

Field 6: space required by ASCII string (in dots)

Strictly speaking, this field is unnecessary, since Dmuse can calculate this space from the ASCII string in field 4. But it saves time to have this information, so it is

10

included.

(End of Text)

VIII. Words (a form of sub-object attached to symbol)

Field 1: Identifier = W

Field 2: horizontal displacement of word(s), relative to
directive or symbol object

Field 3: vertical displacement

Field 4: font number

Field 5: ASCII string (include additional font designations)

IX. Attributes (a form of sub-object attached to notes/rests)

The purpose of attributes is to communication information about an object that is not represented precisely by purely graphical information. An example is the duration of tuplets. The tuplet super-object (H . X) references only the end points of a tuplet; there is no precise way to determine the remaining objects whose duration might be effected by the tuplet. When converting music page files to SCORE pmx files, we need to know the precise duration of all notes. The only foolproof way to convey this information is with an attribute. Attributes are ignored in the printing process.

Field 1: Identifier = A

Field 2: type of attribute

Only one type of attribute currently exist: D = duration

Field 3: numerator of duration

Field 4: denominator of duration

Duration is represented as a proper fraction.

1/4 = quarter note

1/2 = half note

etc.

(End of Attributes)

X. Super-objects (these are things whose shapes depend on the locations of more than one object. They generally follow after all of their associated objects have been described.

Field 1: Identifier = H

Field 2: super-object number

Field 3: type of super-object

There are currently eleven types of super-objects. Since these represent quite different types of musical notation, the description of the remaining fields for each type of super-object will be given separately.

Types of super-objects

B = beam	E = ending
T = tie	R = long trills
S = slur (dotted slur)	V = octave transposition
X = tuple/bracket	F = figure extension
W = wedge	N = null super-object (no action)
D = word plus dashes	

Fields 4--: depend on the super-object type

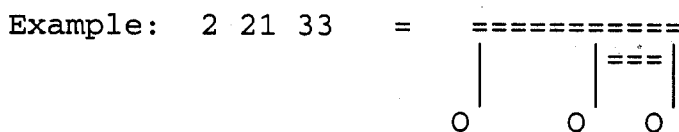
1. Beams (type = B)

Field 4: length of first stem (positive = stem up)
 Field 5: slope of beam
 Field 6: font number (full size vs. que size)
 Field 7: number of associated objects
 Fields 7a, b,... beam codes

beam code = from 1 to 6 digits

0 = no beam
 1 = continue beam
 2 = begin beam
 3 = end beam
 4 = forward hook
 5 = backward hook
 6 = single stem repeater
 7 = begin repeated beam
 8 = end repeated beam

1's digit = eight level beams
 10's digit = 16th level beams
 100's digit = 32nd level beams
 1000's digit = 64th level beams
 10000's digit = 128th level beams
 100000's digit = 256th level beams



2. Ties (type = T)

Field 4: vertical position of tied notes (position measured in dots (300/inch) from top staff line)

For ties on the lower grand staff, 1000 will be added to this value.

Field 5: horizontal displacement of first note head from associated object (non-zero only with chords)

Field 6: horizontal displacement of second note head from associated object (non-zero only with hords)

Field 7: horizontal displacement from note head one

Field 8: vertical displacement from note head one

Field 9: font number

Field 10: situation flag

interference		tips down		tips up	
stem	staff	space	line	space	line
----	----	----	----	----	----
no	yes	1	5	9	13
no	no	2	6	10	14
yes	yes	3	7	11	15
yes	no	4	8	12	16

Field 11: zero

3. Slurs (and dotted slurs) (type = S)

Field 4: situation flag

	bit clear	bit set
	-----	-----
bit 0:	full slur	dotted slur
bit 1:	stock slur	custom slur
bit 2:	first tip down	first tip up
(*) bit 3:	second tip down	second tip up
(+) bit 4:	compute stock slur	hold stock slur

(*) used on custom slurs only

(+) used on stock slurs only

Field 5--8: position fields:

For stock slurs, these are displacements added to the note positions before the slur number is calculated. Displacements are measured in dots (300/inch). Horizontal positive is forward to the right; vertical positive is down.

For custom slurs, these displacements are the actual starting point and ending points of the slur relative to the first and second objects.

Field 5: (extra) horizontal displacement from associated object one

Field 6: (extra) vertical displacement from associated object one

Field 7: (extra) horizontal displacement from associated object two

Field 8: (extra) vertical displacement from associated object two

Field 9--10: parameter fields:

Stock slurs: Field 9: post adjustment to curvature
Field 10: beam flag 1 = set slur above beam
2 = set slur below beam

Custom slurs: Field 9: vertical height parameter

Field 10: length of flat portion as a percent of span between end points

Field 11--12: more parameters

Stock slurs: Field 11: post adjustment to x-position
negative = left; positive = right
Field 12: post adjustment to y-position
negative = up; positive = down

4. Tuplets/brackets (type = X)

Field 4: situation flag

	bit clear -----	bit set -----
bit 0:	no tuplet	tuplet
bit 1:	no bracket	bracket
bit 2:	tips down	tips up
bit 3:	0	tuplet aligns with beam
bit 4:	tuple near notes (bit 4 has meaning only if bit 3 is set)	tuple near beam

Field 5: tuplet number(s) # or #:#

Fields 6--10: parameters

Case I: bit 3 of sitflag = 0 (tuple not aligned with beam)
or bit 3 of sitflag = 1 and bit 4 of sitflag = 0
(tuple aligned with notes of beam)

- Field 6: horizontal displacement from associated object one
- Field 7: vertical displacement from associated object one
- Field 8: horizontal displacement from associated object two
- Field 9: vertical displacement from associated object two
- Field 10: not used

Case II: bit 3 of sitflag = 1 (tuple aligned with beam)

- Field 6: horizontal post adjustment to tuple
- Field 7: vertical post adjustment to tuple
- Field 8: horizontal post adjustment to tuple
- Field 9: vertical post adjustment to tuple
- Field 10: beam super-object number with which tuplet aligns

5. Wedges (type = W)

- Field 4: wedge size at left end (measured in dots)
- Field 5: wedge size at right end
- Field 6: horizontal displacement from associated object one
- Field 7: vertical displacement from top of staff
- Field 8: horizontal displacement from associated object two
- Field 9: vertical displacement from top of staff

6. Word(s) plus dashes (type = D)

- Field 4: horizontal displacement from associated object one
- Field 5: horizontal displacement from associated object two
- Field 6: vertical displacement from staff lines
- Field 7: spacing parameter
- Field 8: font designator

7. Endings (type = E)

- Field 4: situation flag
 - 0 = ending with no number
 - 1 = first ending
 - 2 = second ending
 - 3 = third ending
 - 4 = fourth ending
 - 11 = transposed 8av down
 - 12 = transposed 8av up

- Field 5: horizontal displacement from associated object one
- Field 6: horizontal displacement from associated object two
- Field 7: vertical displacement from staff lines
- Field 8: length of left vertical hook
- Field 9: length of right vertical hook

8. Long trills (type = R)

- Field 4: situation 1 = no trill, 2 = trill
- Field 5: horizontal displacement from associated object one
- Field 6: horizontal displacement from associated object two
- Field 7: vertical displacement from associated object one

9. Octave transposition (type = V)

- Field 4: situation 0 = 8ve up, 1 = 8ve down
2 = 15 up, 3 = 15 down
- Field 5: horizontal displacement from associated object one
- Field 6: horizontal displacement from associated object two
- Field 7: vertical displacement from associated object one
- Field 8: length of vertical hooks

10. Figure extension (type = F)

The figure extension is a solid line indicating that a particular figure carries over to succeeding notes. There can be from 1 to 4 figures aligned vertically.

- Field 4: level 1,2,3 or 4
- Field 5: horizontal displacement from associated object one
- Field 6: horizontal displacement from associated object two

11. Null super-object (type = N)

No additional fields

(End of Super-objects)

XI. End of Music Line

Field 1: Identifier = E

Field 2: xbyte(s) at end of line

xbyte is the punction found at the end of a solid line (____). * = no solid line (or no text).

XII. System Bar

Field 1: Identifier = B

Field 2: bar type	Single bars	Double Bars
	-----	-----
	1 = simple	5 = double light
	2 = heavy	6 = light-heavy
	3 = dotted	9 = heavy-light
		10 = heavy-heavy
		25 = heavy-light with
		repeat dots
	99 = no bar	

Field 3: x off-set from beginning of line

Field 4: number of breaks in line (to avoid interference with text and other designations)

Fields 5--: parameter pairs, beginning and ending of breaks

=====

4.1 Nothing yet has been said about what shapes the glyph numbers represent. The glyph numbers appear mainly as the fourth field of sub-objects. Occasionally one finds a glyph number in field 6 of an object. The table below shows the shapes assigned to the various glyph numbers.

Glyph Numbers for the Music Font

=====

Lower numbers	Higher numbers
=====	=====
1. large clefs	1. small clefs
-----	-----
33 treble (top)	161 treble (top)
34 treble (bottom)	162 treble (bottom)
35 C-clef	163 C-clef
36 bass	164 bass
2. large time signatures	2. small time signatures
-----	-----
37 common time	165 common time

16
38 alle breve time

166 alle breve time

167 duple

168 triple

3. full-size note heads

39 longa

40 breve

41 whole

42 half

43 quarter

3. grace/que-size note heads

169 whole

170 half

171 quarter

4. full-size dot, leger line

44 dot

45 leger line

4. grace/que-size dot, leger line

172 dot

173 leger line

5. full-size rests

46 whole

47 half

48 quarter

49 eighth

50 add-eighth

5. que size rests

174 whole

175 half

176 quarter

177 eighth

178 add-eighth

6. full size flags

51 short up-eighth

52 short down-eighth

53 up-eighth

54 down-eighth

55 up-sixteenth

56 down-sixteenth

57 up-add-flag

58 down-add-flag

6. cue size flags

179 up-eight + slash

180 down-eight + slash

181 up-eighth

182 down-eighth

183 up-sixteenth

184 down-sixteenth

185 up-add-flag

186 down-add-flag

7. full size stems

59 full length up

60 full length down

61 notesize up

62 notesize down

7. cue size stems

187 full length up

188 full length down

189 notesize up

190 notesize down

8. accidentals

63 sharp

64 natural

65 flat

66 double sharp

8. small accidentals

191 sharp

192 natural

193 flat

194 double sharp

9. editorial brackets

67 square left

68 square right

69 round left

70 round right

9. editorial brackets

195 que square left

196 que square right

197 que round left

198 que round right

10. big numbers

10. fingering/figured numbers

71 0
 72 1
 73 2
 74 3
 75 4
 76 5
 77 6
 78 7
 79 8
 80 9

199 0
 200 1
 201 2
 202 3
 203 4
 204 5
 205 6
 206 7
 207 8
 208 9

11. staff lines

81 full-size staff

11. staff lines

209 que-size staff

12. vertical lines

82 full length bar
 83 quarter length bar
 84 full leng. thick bar
 85 qtr. length thick bar
 86 full leng. dotted bar
 87 thick vertical top
 88 thick vertical bottom

12 & 13. other figures

210 small plus for figures (+)
 211 small x for figures (x)
 212 2+
 213 sharp
 214 4+
 215 5+
 216 6/
 217 7\
 218 natural
 219 flat
 220 small place holder
 for figures (-)

13. horizontal lines

89 begin/end hook
 90 solid line (30 dots)
 91 dash line (30 dots)
 92 heavy line (30 dots)

14. tuple numbers (10)

14. articulations (10)

93 horizontal accent
 94 ^ accent
 95 v accent
 96 staccato dot
 97 v stricht
 98 ^ stricht
 99 - legato
 100 , breath
 101 fermata up
 102 fermata down

221 0
 222 1
 223 2
 224 3
 225 4
 226 5
 227 6
 228 7
 229 8
 230 9

15. repetition (5)

103 ./.
 104 solid /
 105 empty /
 106 signet sign
 107 circle + cross sign †

15. octave (5)

231 big upright 8
 232 little upright 8
 233 big italic 8
 234 little italic 8
 235 big italic 15

16. dynamics (6)

108 p
 109 m
 110 f

16. ornaments

236 tr.
 237 ~
 238 mordent

111 s	239 shake
112 z	240 shake from above
113 r	241 shake from below
17. performance indications	242 turn from above
-----	243 turn from below
114 Ped.	244
115 * (end pedal)	245
116 up bow	246
117 down bow	247
118 pedal heel	248
119 pedal toe	249 colon for tuples
120 arpeggiate	250 (blank)
121 repeat notes	251 (editorial piano)
122 harmonic a	252 (editorial mezzo)
123 harmonic b	253 (editorial forte)
124 thumb position	254 (editorial trill)
	255
18. miscellaneous	

125 stem repeater (for quarters, etc.)	
126 stem repeater (for eights, etc.)	
127 stem repeater (for whole notes)	