

# FORMAT FOR CCARH S-FILES (source files)

-----  
Version 4.00  
(05/03/2003)

A CCARH S-file is organized as a set of variable length records. Each S-file describes the music in one musical part of a musical work. A musical part may consist of one or more lines of music. For example, Oboe I and Oboe II may be combined on one staff and therefore be considered as one musical part (Oboes). Music on the grand staff may be considered 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.

An S-file has three sections: a header, the main section containing the primary musical data, and an auxiliary section containing footnote data. The third section of the S-file is optional.

The first nine records of the S-file header provide information on the encoding process (date and encoder), the musical work (work and movement numbers and titles), and the source of the data. Record 10 identifies the part groups to which this part belongs. For example, there may be files for Oboe I, Oboe II and Oboe I,II combined. Oboe I,II would belong to the group called "score," whereas the Oboe I and Oboe II parts by themselves would belong to the group called "parts" (but not to score). In addition, if the Oboe parts are musically independent, they might belong to a group called "tracks." The names and arrangements of the groups and parts is left to the encoder. For each group to which a part belongs, there is a record (starting with record 12) giving the group name and the sequence number of that part in that group. The arrangement of the parts into various groups allows flexibility in printing full scores, short scores, and individual instrumental parts. It makes possible the representation of vocal soloist parts both as separate parts (for analysis) and combined with other soloists (e.g., as in recitatives). It makes possible the representation of keyboard works both as single files (for printing) and as separate tracks (for analysis).

## Header Format

```
record 1:  free
record 2:  free
record 3:  free
record 4:  [date] [encoder]
record 5:  WK#:[work number]    MV#:[movement number]
record 6:  [source]
record 7:  [work title]
record 8:  [movement title]
record 9:  [name of part]
record 10: free (we put in [mode], [movement type] and [voice])
record 11: Group memberships: [name1] [name2] ...
record 12: [name1]: part [x] or [number in group]
record 13: ...
... (as needed)
```

## Source files (page 2)

The main section of an S-file follows directly after the header. The end of the main section is marked by a special end-of-movement record consisting of the five characters, "/END". The order of the records is an integral part of the information contained in the file. The first character in each data record functions as a **control code**, describing the type of information contained in the record. For certain data types, such as notes, the first character also contains data information. This system of describing record types with a control code allows for significant future expansion of the the S-file format. At the moment, there are twenty-one control codes and fourteen data types.

### Data Types

At the moment there are **fourteen** data types:

1. musical attributes
2. musical directions
3. bar lines
4. regular notes and rests
5. extra note in a regular chord
6. grace notes and cue notes
7. extra grace/cue note in a chord
8. figured harmony
9. forward or back space in time
10. continuation line
11. comments
12. end of music/end of file
13. sound directions
14. print suggestions

### Special concepts and requirements for CCARH music printing

In addition to the formal structure of the S-file, there are some extra rules that must be followed in order for the CCARH Autoset program to process the files for display and printing.

#### 1. Concept of division pointer

S-files are a set of ordered records; roughly speaking, the order is chronological. The time keeper is an invisible pointer called the division pointer. The division pointer is set to 1 at the beginning of each measure. Duration values in columns 6-8 (regular notes and rests, irests and backups) advance or backup the division pointer. The division pointer should not be backed up to less than one. You cannot backup over a measure boundary. The time length of a measure is defined by the greatest value attained by the division pointer. **The CCARH Autoset program requires** that the final value of the division pointer be equal to its greatest value.

#### 2. Concept of the figure division pointer

Whereas figured harmonies are normally located on a division pointer boundary, it sometimes happens that we would like figured

harmonies to be offset (forward) from a division pointer. In this case, the offset value is determined by the figure division pointer. For each advance of the division pointer, the figure division pointer is set to zero. The figure division pointer is advanced by figured harmony data by the amount specified in columns 6-8.

The CCARH Autoset program requires that the sum of a division pointer and its associated figure division pointer must always be less than the final value of the division pointer (time length of the measure).

### 3. Concept of the musical direction offset

Musical directions are normally located on a division pointer boundary, but there are occasions when we would like musical directions to be offset (forward) from a division pointer. In this case, the offset value is contained in columns 6-8.

The CCARH Autoset program requires that the sum of a division pointer any musical direction offset associated with it must always be less than the final value of the division pointer (time length of the measure).

### 4. Concept of the cue note pointer

Cue notes do not advance the division pointer; they have their own time pointer, the cue note pointer. Any advance of the division pointer sets the cue note pointer to zero. The cue note pointer is advanced by the note value of the cue note. In this respect, cue notes are not as flexible as regular notes; i.e., the set of possible lengths is limited to what regular notation allows. Autoset converts the cue note duration (e.g., eighth note) into a fixed number of divisions, and this amount is added to the division pointer to determine the location of the cue note.

The CCARH Autoset program requires that the final value of the cue note pointer + the division pointer must always be less than the final value of the division pointer (time length of the measure).

### 5. Grace notes

Grace notes take their logical position from the value of the division pointer. As far as Autoset is concerned, grace notes are "piled on top of" whatever regular musical object comes next in the file. Space is later made for them to precede that object. Grace notes normally precede regular notes; but they may also precede rests or a measure line.

### 6. Order of objects at the end of a measure

When the division pointer is at its greatest (final) value, it is still possible to place objects of the types: musical attributes and musical directions. When both types occur at the end of a measure, The CCARH Autoset program requires that musical attributes (i.e., records starting with \$) be placed first, followed by musical directions (i.e., records starting with \*).

# Source files (page 4)

## Data Format -- control code in column one

1. Musical attributes: control code = \$

column 1: "\$"  
column 2: level number (optional)  
column 3: footnote column  
columns 4--80: attribute fields

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 number one.

field type	field identifier	field data type
key	K:	integer
divisions per quarter note	Q:	pos. int.
time designation	T:	two integers
clef	C:	integer
clef	C#:	integer
transposing part	X:	integer
number of staves for part	S:	integer (def = 1)
number of instruments represented	I:	integer (def = 1)
directive (last field on line)	D:	ASCII string
directive (last field on line)	D#:	ASCII string

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

key code: The numbers -7 to +7 are reserved for standard key signatures. Minus numbers are for flats; Positive numbers are for sharps. Other integer codes may be assigned at the discretion of the encoder.

time designation: Time designation is given by two non-negative integers separated by a slash (/). The first integer is normally the time numerator and the second integer is normally the time denominator. Special codes for time signatures are shown below:

1 1 = common time  
0 0 = alla breve  
2 0 = simple 2  
3 0 = simple 3

## Source files (page 5)

divisions per quarter note: This parameter can only be specified at the beginning of a piece or directly after a controlling bar line. Specifying it at other locations makes it difficult to combine separate parts for printing and for MIDI output.

clef code: 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.

clef sign codes: 0 = G-clef  
1 = C-clef  
2 = F-clef  
3 = G-clef transposed down  
                    (modern clef for tenors)  
4 = C-clef               "       down  
5 = F-clef               "       down  
6 = G-clef               "       up  
7 = C-clef               "       up  
8 = F-clef               "       up

line numbers: 1 = highest line  
5 = lowest line

transposing part: This integer (positive or negative) indicates a transposing interval, if there is one, 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., vc and bass on the same part or 8' and 16' sound on an organ pedal line.)

number of staves: This integer (1 or 2 at the moment) indicates the number of staves the part or parts will be written on. This number can change within a movement. The number of staves will automatically be set to 2 if a "C2:" or a "D2:" is encountered.

number of instruments represented: This integer (1 or more) indicates the number of independent instruments represented by the parts. If this number is more than one, certain printing conventions will hold.

- (1) notes with the same stem direction will be combined into one chord
- (2) 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.

## Source files (page 6)

directive: This ASCII string data group is terminated by the end of the record. For this reason, if a record contains a directive, the directive must be the last field of the record. Directives may contain font designators.

### 2. Musical directions: control code = \*

column 1:        " \* "

columns 2-5:    blank

columns 6-8:    optional forward offset (measured in units of duration and right justified in the field). Use of this field allows the encoder to place a musical direction at a division that does not otherwise contain a musical record.

columns 9-12:   blank

columns 13-15:  footnote and level information

column 13: footnote flag (blank = none)

column 14: level number (optional)

column 15: track number (necessary if two or more wedges, sets of dashes, 8va transpositions, etc.)

column 16:       blank

columns 17-18:  type of direction (one or two letters)

#### 1. rehearsal numbers/letters

A = segno sign

#### 2. directions expressed in words

B = right justified ASCII string

C = centered ASCII string

D = left justified ASCII string

(may be combined with types E,F,G,H,J)

#### 3. wedges

E = begin wedge

F = end wedge

(may be combined with types B,C,D,G)

#### 4. letter dynamics

G = letter dynamics (given in ASCII string)

(may be combined with types B,C,D,E,F,H,J)

#### 5. dashes ( - - - - - )

H = begin dashes (after words)

J = end dashes

(may be combined with types B,C,D,G)

Source files (page 7)

6. pedal (pianoforte)

P = begin pedal: Ped.

Q = release pedal: \*

7. octave shifts (in the printing process)

U = shift notes up (usually by 8va)

V = shift notes down (usually by 8va)

W = stop shift

Notes that are difficult to notate because they are very low (usually in bass clef) are shifted up; notes that are difficult to notate because they are very high (usually in treble clef) are shifted down.

8. tie terminators

X = tie terminator (generates a Mark object)

The pitch of the tie being terminated appears starting in column 25

column 19: location flag (optional)

' ' = indication below line

+ = indication above line

(may be used by types A,B,C,D,E,F,G,H)

column 20: blank

columns 21-23: numerical parameter (optional)

for types E and F: wedge spread

for types U and V: shift size (when not 8va)

Wedge spread is measured in tenths of staff line space. 10 units = space between two staff lines

column 24: staff number (' ' = 1)

Used in the case of music represented on more than one staff.

columns 25...: ASCII word string used in A,B,C,D,G and X

Examples:

1. cresc. - - - - - ff

Starting record: DH cresc.

Ending record: JG ff

2. f <decreasing wedge> p

Starting record: GE 15 f

Ending record: FG 0 p

### 3. Bar line: control code = m

```

column 1:      "m"

columns 2-7:   "easure" = regular bar line
               "dotted" = dotted bar line
               "double" = (light) double bar line
               "heavy1" = heavy bar line
               "heavy2" = light-heavy double bar
               "heavy3" = heavy-light double bar
               "heavy4" = heavy-heavy double bar

column 8:      empty

columns 9-12:  optional bar number for this bar
               (left justified)

columns 13-15: footnote and level information

column 13:     footnote flag (blank = none)
column 14:     level number (optional)
column 15:     blank

column 16:     blank

columns 17-80: flags:  * = non-controlling bar line
                       ~ = continue ~~~ across bar line
                       A = segno sign at bar
                       F = fermata sign over bar line
                       E = fermata sign under bar line
start-end#     = start ending #
stop-end#      = stop ending #
disc-end#      = discontinue ending # line
:|             = repeat backward
|:             = repeat forward
    
```

Bar lines are divided into two types: controlling and non-controlling. Controlling bar lines are lines which run through an entire score. In this respect, they mark the beginning of a new global measure. Non-controlling bar lines need not have this property. Non-controlling bar lines may not serve for line breaks or page breaks. The designation of a bar line as non-controlling is to some extent left to the discretion of the encoder, e.g., in the case of a double bar in the middle of a normal measure, this could be controlling or non controlling. However, in a case such as the Minuet from the Mozart opera "Don Giovanni," where the score uses three different meters simultaneously, the non-aligned bar lines must be designated as non-controlling.



Source files (page 9)

4. Regular note/rest control code = A,B,C,D,E,F,G or r

columns 1-4: pitch or rest, Cff0 to B##9, C4 = middle C

column 5: blank

columns 6-8: duration (right justified)

column 9: tie flag    " " = no tie  
                      "- " = tie

From the data in columns 1 to 9, (i.e., pitch and duration), it is possible to reproduce aural output of the musical part)

Note: Normally a note with a tie must be followed immediately by another note of the same pitch (in that pass). It can happen that a tie goes nowhere (e.g. in first endings or da capos). In this case you must use an "X" type musical direction to terminate the tie.

columns 10-12: blanks

columns 13-15: footnote and level information, track #

column 13: footnote flag (blank = none)

column 14: level number (optional)

column 15: track number (optional)

Where more than one musical line is represented in a part (e.g., Oboe I,II or keyboard music), it is essential for purposes of analysis to know for each note (or chord) the musical line or "track" to which the note belongs. In some cases this is "interpretive" information, provided as a service by the encoder.

column 16: blank

columns 17-22: note description

column 17: note type

L,b,w,h,q,e,s,t,x,y,z (Longa to 256th note)

H,Q,E,S,T,X,Y,Z (Half to 256th note)

B,A,9,8,7,6,5,4,3,2,1 (Longa to 256th cue-size)

When the control code = "r" (rest), column 17 may be blank. This signals the music typesetter that the rest is to be represented as a centered whole rest, regardless of its duration.

For half notes and smaller, both upper and lower case letters will work, but in general the upper case alternative should NOT be used.

## Source files (page 10)

With notes it makes absolutely no difference, but with rests, using an upper case letter will cause the music typesetter to replace the letter with a blank when the part inclusion flag (General print suggestion x) = 1. The part will be omitted from a system when there are only whole rests in that part in that system (See 14. Print Suggestions: section 11).

column 18: dot flag      " " = no dot  
                          "." = single dot  
                          ": " = double dot  
                          "; " = triple dot  
                          "! " = quadruple dot  
column 19: actual accidental flag

# = sharp	X = sharp-sharp
n = natural	& = flat-flat
f = flat	S = natural-sharp
x = double sharp	F = natural-flat

columns 20-22: designation of time modification  
Two digits, separated by a colon (:)  
For standard cases, such as triplets (3:2), the colon and the second digit are usually omitted. The numbers 10--35 are represented by the letters A--Z.

column 23:      stem direction  
                  d = down  
                  u = up  
                  " " = no stem

column 24:      staff number (" " = 1)

Used in the case of music represented on more than one staff.

column 25:      blank

columns 26-31: beams (up to six levels = 256th note)  
                  [ = start beam  
                  = = continue beam  
                  ] = end beam  
                  / = forward hook  
                  \ = backward hook

column 26 is for eighth beams,  
27 for sixteenth,

...

## columns 32-43: other notations

Codes are read from left to right. The character "&", followed by a digit (1..9,A..Z), is used to indicate a specified data level. All codes to the left of the first "&" belong to the base data level.

The following codes are somewhat arbitrary. They have been chosen for representing common musical notation for Western music from the 16th through the 19th centuries. The encoding scheme is not complete but may be augmented and/or altered to meet the special requirements of the music being encoded.

for more on ties and slurs, see type 14, "print suggestions"

## Ties, Slurs, Tuples

```

-----
-   = tie
(   = open slur1
)   = close slur1
[   = open slur2
]   = close slur2
{   = open slur3
}   = close slur3
z   = open slur4
x   = close slur4
*   = start tuplet
!   = stop tuplet

```

## Ornaments

```

-----
t   = tr.
r   = turn
k   = delayed turn
w   = shake
~   = wavy line (trill)
c   = continue wavy line
M   = mordant
j   = slide

```

## Technical Indications

```

-----
v   = up bow
n   = down bow
o   = harmonic
0   = open string
Q   = thumb position (cello)
1,2,3,4,5 = fingering
:   = next fingering is a
      substitution e.g.,
      5:42:1 = 5-4
              2-1

```

## Articulations and Accents

```

-----
A   = vertical accent (/ \)
V   = vertical accent (\ /)
>   = horizontal accent
.   = staccato
_   = legato
=   = line with dot
      under it
i   = spiccato
,   = breath mark

```

## Accidentals on ornaments (must follow directly after ornament)

```

-----
s   = sharp (ss = double sharp)
h   = natural
b   = flat (bb = double flat)
u   = next accidental is below
      rather than above ornament.
      In case accidentals appear
      above and below ornament,
      the accidental above
      should be encoded first.

```

## Other Indications and Codes

```

-----
S   = arpeggiate (chords)
F   = upright fermata
E   = inverted fermata
G   = G. P. (grand pause)
p   = piano (pp, ppp, etc.)
f   = forte (ff, fff, etc., fp)
m   = mezzo (mp, mf)
Z   = sfz (also sf)
Zp  = sfp
R   = rfz
^   = accidental above note
+   = cautionary accidental

```

Source files (page 12)

columns 44-80: text: multiple lines of text set off by |  
example: Deck|See|Fast

Special text codes:

"-" at the end of a word generates hyphons to the next text word on that line.

"-" alone continues hyphons.

"\_" at the end of a word generates an extension line to the next word on that line.

"\_" alone continues the extension line.

"&" can be used as "pseudo-text" for the purpose of ending an extension line in the situation where there is no further text (see "\_" above). Otherwise, the extension line may continue indefinitely. The "&" is silent; i.e., it generates no text output.

",.!?;" punctuation when followed by "\_" will be placed at the end of the extension line.

5. Extra note in a chord control code = ' ' (blank)

column 1: blank  
columns 2-5: pitch (see regular note)  
columns 6-8: blanks  
column 9: tie flag  
columns 10-42: same as for regular note

6. Grace notes, cue-notes control code = g or c

column 1: g = grace note  
c = cue note (grace size, but written in time)  
columns 2-5: pitch/rest (see regular note)  
columns 6-7: blanks  
column 8: note type  
0 = eighth note with slash  
1 = 256th note  
2 = 128th note  
3 = 64th note  
4 = 32nd note  
5 = 16th note  
6 = 8th note  
7 = quarter note  
8 = half note  
9 = whole note  
A = breve  
column 9: blank  
columns 10-80: same as regular notes

Source files (page 13)

7. Extra Grace/Cue note in a chord

column 1: g or c (same as 7 above)  
column 2: blank  
columns 3-6: pitch (see regular note)  
column 7: blank  
column 8: note type (same as 7 above)  
column 9: blank  
columns 10-80: same as other grace and cue notes

8. Figured harmony control code = f

column 1: f = figured harmony  
column 2: number of figure fields  
columns 3-5: blanks  
columns 6-8: advance of figure division pointer

Figures take their position from the first regular note that follows the figure records. In the case where the figures change during the duration of a note, the advancing parameter (columns 6-8) is used to indicate the elapsed time between changes. In the case where a figure appears after a note has sounded, the "blank" figure is used as a placeholder to advance the figure division pointer.

columns 9-12: blanks

columns 13-15: footnote and level information

column 16: blank

columns 17--: figure fields

The figure fields are set off by one or more blanks. Figure numbers may extend from 1 to 19. They may be modified in front by #, n, f, and x. They may be modified afterward by #, n, f, +, \, / and x. The #, n, f and x signs may stand alone as figures. A "b" indicates a blank figure. This is used as a place holder in a list and also to start a continuation line with no figure. The first figure field is for the top of the figure list. For example, in the

figure list #4<sup>6</sup>, the 6 would be represented in the first field.  
2

figures, signs and modifiers:

numbers 1 2 ... etc. to 19

# = sharp	\ = diminish (used with
n = natural	fig. nos. 6 and 7)
f = flat	- = short line in
x = double sharp	_ = long line from previous fig.
+ = augment (used with	b = blank
fig. nos. 2, 4, 5, and 6)	

## Source files (page 14)

It is possible to represent editorial figures by placing them inside parentheses (). No empty space is permitted inside parentheses.

### 9. Forward and Back space in time control code = i,b

columns 1-5: "irst " = forward space (invisible rest)  
(or "irest")  
"back " = backspace

columns 6-8: duration to skip forward or back up

columns 9--12: blanks

columns 13-15: footnote and level information

column 13: footnote flag (blank = none)

column 14: level number (optional)

column 15: blank

column 16: blank

column 17: pass number (optional)

This feature can be used to express parallel action in the same part (e.g., keyboard music).

### 10. Continuation line control code = a

column 1: a = append to previous line

columns 2-16: blanks

columns 17-80: continuation of previous line

### 11. Comments control codes = & and @

column 1: @ = this line is a comment

& = enter comment mode. All subsequent records are comments until another record with a & in column 1 appears. "&" in column one acts like a toggle switch between "data" mode and "comment" mode.

### 12. End of music data or end of file control code = /

column 1: /

columns 2-5: "END" = end of file

### 13. Sound directions

column 1:        S

A Sound Direction record can follow any record that produces a sound or influences time in some way. This includes types 4 to 8: regular notes and rests, extra note in a regular chord, grace notes and cue notes, extra grace/cue note in a chord, and figured harmony.

Sound information can be given on a variety of attributes connected with a note. This is a complicated subject, for which I am not yet ready to attempt a full description. I am inclined to suggest the following mechanism, as a first pass at the problem:

Since sound directions may apply to a wide variety of musical attributes, e.g., (1) the attack and dynamic envelope of a note, (2) the time of attack and length of a note (or rest), and (3) directions for performing ornaments, it makes sense to use a multiple field system, similar to the one used for musical attributes (\$). In this case, each field is introduced by a capital "C", followed immediately by a number and a colon, e.g., "C8:" or "C23:". The meaning is that the data following this designation (all columns up to the next field designation or to the end of the record), will apply to the item in the specified column number of the previous record. For example, if the previous record were a note, and there were a trill indicated by a "t" in column 33 of that record, then C33: would indicate a sound direction field containing data on how that trill should be realized in a sound file. Of course, each ornament, or pitch, or duration, will have a different set of needs regarding its sound specification, and these will have to receive further definition as the specification for this type of record develops. At the moment, our specific need relates to specifying information for the following situations:

#### 1. Onset and length of grace notes

Field designator: C1:

Data elements:

p	= steal time from previous note
f	= steal time from following note (default)
m	= don't steal time; make time (free cadenzas, etc.)

t<#> (stealing time) = percentage (0 to 100) of time to steal for this note. The time is specified as a percentage of the duration of the controlling note, even though the time may not be stolen from this note. The controlling note is the first regular note following the grace note. In the case where the grace note comes just before a bar line, the first note in the next measure is used as the controlling note.

## Source files (page 16)

t<#> (making time) = number of real-time divisions for this note. There is no limit to the number used. All regular notes sounding at this point will play out their sound and then wait while these grace notes "do their thing."

Notes on adding time: If time is added by specifying "m" type grace notes, there are two limitations that must be observed.

1. The total amount of time added must be representable by divisions in all parts. For example, if a particular part has 4 divisions per quarter note and a total of 25 divisions are added to a measure using C1:m records, then this is the equivalent of adding 25 sixteenth notes. In this case, all other parts must be able to represent the time interval of a sixteenth in this measure. If there is a part which has 2 divisions per quarter, then a \$ Q:4 record must be inserted at the beginning of this measure, and all durations in the measure must be doubled. A \$ Q:2 record must be inserted at the beginning of the next measure to restore the original parameter.
2. A part which has extra time added must be assigned to its own midi channel. For example, if there are 2 oboes and the first oboe has a cadenza (as in the first movement of Beethoven's 5th Symphony), then the oboes may not be assigned to the same midi channel.

## 2. Onset and length of trills, turns, shakes, and wavy lines

Field designator: C32: to C43: (depending on where the ornament is indicated)

Data elements:

u = start on upper note (default)  
m = start on main note  
w = whole-step trill (default)  
h = half-step trill  
j = unison trill  
e = include a two note turn at the end of the trill (whole step)  
f = include a two note turn at the end of the trill (half step)  
a = accelerate trill slightly  
n<#> = number of beats (min is 2, default is 4)  
s<#> = percentage point for landing on second beat of trill (default is 25)

t<#> = percentage point for landing on last beat of trill (default is 75)

The default trill (uwn4s25t75) is a four note trill starting on the upper whole step and having four equal beats.



3. Onset and length of (inverted) mordents

Field designator: C32: to C43: (depending on where the ornament is indicated)

Data elements:

m = start on main note (default)  
b = start on note below main note  
w = whole-step mordent (default)  
h = half-step mordent  
a = accelerate mordent slightly  
n<#> = number of beats (minimum is 2, default is 3)  
s<#> = percentage point for landing on second beat of trill (default is 12)  
t<#> = percentage point for landing on last beat of trill (default is 24)

The default mordent (mwn3s12t24) is a three-note snap starting on the main note and going down a whole step.

4. Alternate instrument (pizz. for strings)

Field designator: C2: (to distinguish it from grace note information)

Data elements:

a = pizzicato for this note (this designation must be contained in a sound record which follows directly after the note in question.)  
A = pizzicato for this note and every regular note that follows in the file, until cancelled.  
b = arco (used to cancel A)

5. Da Capo direction and Segno sign and implied |:

Field designator: C0: (because this does not relate to a column number)

Data elements:

d = da capo to beginning of movement or to Segno sign. Normally this record would directly precede the "/END" or "/FINE" record.  
S<#> = segno sign: da capo to this point in the file.  
# = divisions per quarter (for information of sound and MIDI generating programs)  
|> = implied forward repeat dots (usually follows a bar line)

6. Fine signs (written or implied)

Field designator: C8:

Data elements:

F<#> = Fine sign (written or implied) This record should follow any final note or rest in a movement which has a "da capo" direction. # = actual duration of the final note or rest. This is needed because some "fines" are indicated only by a fermata, and these can be over notes of different durations in different parts!

In the case where there is more than one active track in the measure with the "Fine" (i.e., there is a backspace command in the measure), all final notes must have their durations specified by a "Fine" sound record. In the case of chords, the "Fine" sound record should follow the last chord tone record.

7. Tempo changes

Field designator: C0: (because this does not relate to a column number)

Data elements:

W<#> = new tempo in quarter notes per minute. If parameter <#> = 0, this means that the sound generating programs must ask the user for a value At the time of compiling a sound (MIDI) file.

Note: This must be the only data element in this sound record.

8. Changes in dynamics

Field designator: C0: (because this does not relate to a column number)

Data elements:

V<#> = dynamic level (velocity) measured as a percent of the default (which is forte). Normally the programs constructing MIDI performance files assign a flat value of 90 to the velocity byte. Occasionally we may want to change this value either to bring out a part or to suppress a part. MIDI plus compilations (for data transmission) are not affected by this feature.

Note: This must be the only data element in this sound record.

14. Print suggestions

column 1:        P

Print suggestions come in two types: General and Specific.

General print suggestions can occur anywhere in body of the stage2 data and are used to set general print parameters such as minimum note spacing, default fonts, etc.

Specific print suggestions can follow any record that contributes to the printed output of the music. This includes types 2 through 8: musical directions, bar lines, regular notes and rests, extra notes in a regular chord, grace notes and cue notes, extra grace/cue notes in a chord, and figured harmony.

columns 2... until a blank is encountered:

A code string (string of codes containing no blanks) indicating the classes of situations to which this suggestion applies. A space " " in column 2 means that the suggestion is universal; it applies to all printing.

The code string may contain letters and/or numbers. All numbers must be preceded by the "#" sign.

Numbers refer to the notesize to which the print suggestion applies. The codes "<" and ">" may be used to indicate all notesizes "less than" or "greater than" a certain number. For example, a print suggestion beginning thus, P#>12, would apply to all notesizes greater than twelve.

Letter codes indicate the application to which a print suggestion applies.

Codes currently recognized by the software are listed below.

"p" = parts  
"s" = score  
"t" = short score (piano vocal)  
"a" = all applications

Note: If there is a code string present, the designation of notesizes is optional, but the designation of relevant application letter codes is mandatory. The reason for this is that notesize designation is a process of elimination (i.e., don't use this suggestion if the notesize doesn't fit) whereas application designation is a process of inclusion (i.e., use this suggestion only if the relevant letter is present in the code). The code letter "a" can be used to designate all applications (for a particular set of notesizes).

Special case: If column 2 contains a "v", the next four columns will contain the version number of autoset for which the print suggestions are valid and have been checked. If the version of autoset running is different from the version number in the stage2 file, (and the version of autoset is 4.00 or higher), a warning message will be displayed. When this happens, it is a good idea to check the output of each print suggestion to make sure that the outcome is the desired one. It is expected that changes and improvements will be made to autoset from time to time.

Print suggestions, like sound directions, use a multiple field system. Each field is introduced by a capital "C", followed immediately by a number and a colon, e.g., "C8:" or "C23:". The meaning is that the data following this designation (all columns up to the next field designation or to the end of the record), will apply to the item in the specified column number of the previous (non-Sound) record. For example, if the previous record were a note, and there were a slur starting on that note indicated by a "(" in column 33 of that record, then C33: would start a field containing one or more suggestions on how that slur should be printed. At the moment, we can give print suggestions in the following situations:

The type of printing to which a suggestion applies (columns 2..) may also be specified in a "C" field. In this case, the selective number and letter codes(s) described above should follow the column number and precede the colon. For example, "C33sp:" would indicate a print suggestion applied to the datum in column 33 from the previous record and applied only to the printing of a score or a part (and not to a short score). This feature allows greater selectivity for print suggestions than the use of selective codes starting in column 2.

A print suggestion which has a code C0: is a general suggestion and not related to any specific column in a previous record.

#### 1. Position of slurs

Field designator: C32: to C43: (depending on location of slur)

Data elements:     o       = place slur over the note in question  
                    u       = place slur under the note in question

These suggestions are needed only when the standard algorithms fail to place the slur properly, as sometimes happens with multiple parts on a staff or with double stops in the strings.

## 2. Orientation of ties

Field designator: C32: to C43: (depending on location of tie)

Data elements:     o     = over-hand tie (tips down)  
                  u     = under-hand tie (tips up)

These suggestions are needed only when the standard algorithms fail to place the tie properly, as sometimes happens with multiple parts on a stave or with double stops in the strings.

## 3. Tuplet and tuplet bracket (default is no bracket)

Field designator: C32: to C43: (depending on location of tuplet starter)

Data elements:

[     = use continuous square bracket  
(     = use continuous slur bracket  
:     = use square bracket and break in the middle  
;     = use slur bracket and break in the middle  
i     = place tuplet number inside bracket (below, if tips are down; above if tips are up). Default is outside. i has meaning only when combined with [ or (  
x<#> = shift x position of tuplet (units are in tenths of interline distance)  
      # > 0: right -->  
      # < 0: left <--  
y<#> = shift y position of tuplet (units are in tenths of interline distance)  
      # > 0:  down  
      # < 0:  up

## 4. Suggestions modifying the printing of note, rest and figure objects

Field designator: C1:

Data elements:

x<#> = shift default x position (units are in tenths of interline distance)  
      # > 0:  right -->  
      # < 0:  left <--  
X<#> = x position relative to primary horizontal position of notes (where most notes line up) (units are in tenths of interline distance)  
y<#> = shift default y position (units are in tenths of interline distance)  
      # > 0:  down  
      # < 0:  up  
Y<#> = y position relative to staff line. (units are in tenths of interline distance)

## Source files (page 22)

```
p<#> = printout modifier
      # = 0: (default)
no-print suggestions
      # = 1: leave space, don't print note or dot
      # = 2: leave space, print only a dot
suggestions extending note length
      # = 3: print note, no dot
      # = 4: print note, add extension dot
      # = 5: double note length, no dot
      # = 6: double note length, print dot
      # = 7: quadruple note length, no dot
```

Note: In certain special cases, an invisible rest (irest) will require the allocation of physical space in order for mskpage to run properly. This condition is rarely encountered; but when it does, the result is that mskpage cannot compile a score properly because it can't make the various parts line up. Use the print suggestion "P C1:p1" after the offending irest to allowcate the necessary physical space.

### 5. Suggestions for location of notations attached to notes

Field designator: C18: for extension dots  
C19: for accidentals  
C32: to C43: (depending on column location  
of the notation)

#### Data elements:

```
x<#> = shift default x position (units are in tenths of
      interline distance)
      # > 0: right -->
      # < 0: left  <--
X<#> = x position relative to note object (units are in
      tenths of interline distance)
y<#> = shift default y position (units are in tenths of
      interline distance)
      # > 0: down
      # < 0: up
Y<#> = y position relative to note object. (units are in
      tenths of interline distance)
L<#> = (ties only) change end point of a tie (+ or -).
      (units are in tenths of interline distance)
a     = place notation above note object (over-ride default)
b     = place notation below note object (over-ride default)
```

#### Notations for which this works

-----

ornaments: turns, trills, shakes, mordents, wavy lines (trill)

articulations: spiccato, staccato, line over dot, legato,  
horizontal accent, vertical accents

technical: up bow, down bow, string harmonic, thumb position,  
open string, fingerings

dynamics: combinations of letter dynamics

other: fermatas, ties

In the case of ties, the x-shift applies only to the left end of the tie. This can be used to avoid a clash between extension dots and the tie.

## 6. Suggestions for shape and location of slurs

Field designator: C32: to C43: (depending on column of the slur code)

Case I: Start slur "(", "[", "{", "z"

Data elements:

x<#> = (extra) horizontal displacement from associated starting note (units are in tenths of interline distance)

# > 0: right -->

# < 0: left <--

Length and shape of slur are affected.

X<#> = post adjustment to x-position of slur.

negative = left; positive = right. (units are in tenths of interline distance). Length and shape of slur are not affected.

y<#> = (extra) vertical displacement from associated starting note (units are in tenths of interline distance)

# > 0: down

# < 0: up

Length and shape of slur are affected.

Y<#> = post adjustment to y-position of slur negative = up; positive = down (units are in tenths of interline distance). Length and shape of slur are not affected.

Note: The suggestion may contain any combination of the four data elements

Case II: End slur ")", "]", "}", "x"

Data elements:

x<#> = (extra) horizontal displacement from associated ending note (units are in tenths of interline distance)

# > 0: right -->

# < 0: left <--

Length and shape of slur are affected.

y<#> = (extra) vertical displacement from associated ending note (units are in tenths of interline distance)

# > 0: down

# < 0: up

Length and shape of slur are affected.

h<#> = post adjustment to curvature. This works within limits. positive = more curvature; negative = less curvative. Shape is affected, but length should not be (if it is, then the value of # is out-of-bounds).

Source files (page 24)

7. Suggestions for representing beamed notes with repeaters

Field designator: C26: or C27:

Data elements:

- a = use repeater for next beam only.
- A = use repeaters for all beams which follow.
- b = return to normal beaming (used to cancel A)

Note: Print suggestions for beams normally precede the beginning of beams. If the field designator is C26:, this indicates that the maximum use of repeaters is desired; if the field designator is C27:, then the "top" beam should not be represented as a repeater.

8. Suggestion for changing the length of the first stem on beamed notes

Field designator: C26:

Data element:

- y<#> = change length of first stem for a set of beamed notes, as calculated by mskpage or mkpart.  
(units are in tenths of interline distance)
  - # > 0: make stem longer
  - # < 0: make stem shorterSuggestions apply to up and down stems.

9. Suggestions for musical directions

Field designator: C17: or C18: (depending on location of the musical direction)

Data elements:

- x<#> = shift default x position (units are in tenths of interline distance)
  - # > 0: right -->
  - # < 0: left <--
- X<#> = x position relative to staff line. (units are in tenths of interline distance)
- y<#> = shift default y position (units are in tenths of interline distance)
  - # > 0: down
  - # < 0: up
- Y<#> = y position relative to staff line. (units are in tenths of interline distance)

Field designator: C25: and greater (depending on location of the font change). Multiple font changes are allowed in the same line.

Data element:

- f<#> = font number for ASCII text in the specified column and subsequent columns to the right (until a new change is encountered). Used with direction types A,B,C,D,G.



**Note:** If the code **C25:f0** is used, i.e., "set the initial font to zero," this will cause the directive to BLANK, that is, not to print. This feature can be used, for example, to blank directives which are present (and needed) in all parts but which in the full score are needed only at the bottom. The print suggestion in this case would be:

Ps C25:f0 --> Apply this suggestion only when  
typesetting the i-files for a  
score, and don't include the  
directive above this suggestion.

#### 10. Suggestions for treatment of whole measures

Field designator: C1:

Suggestion codes:

n = do not expand the spacings in the following measure  
in the left-edge alignment process.

#### 11. General print suggestions

Field designator: C0:

Suggestion codes:

c<#> = continuo line  
    # = 0: normal operation  
    # = 1: blank out rests  
d<#> = height for time words and other musical designations.  
    # is measured in scale steps above the top line of  
    the staff. Default is 6 scale steps.  
f<#> = default font for musical directions.  
n<#> = numbering measures  
    # = 0: stop numbering measures  
    # > 0: start numbering measures with <#>  
p<#> = minimum distance between notes (expressed as percent  
    of the default).  
q<#> = duration which is assigned the minimum distance  
    0 = recompute default (from this point onward)  
    1 = whole notes  
    ...  
    8 = eighth notes  
    16 = sixteenth notes, etc  
r<#> = placement of rests  
    0 = use default placement  
    1 = always place on middle line  
s<#> = space between grand staves measured in multiples of  
    leger lines times 10 (e.g. 100 = 10 leger lines).  
t<#> = global tuplet placement  
    # = 0: use default  
    # = 1: place tuplet near note heads  
    # = 2: place tuplet near note stems (beams)  
    # = 3: place all tuplets above notes  
    # = 4: place all tuplets below notes

## Source files (page 26)

x<#> = part inclusion flag  
# = 0: include part in every system (default)  
# = 1: leave part out of system when there are  
only whole rests in the part  
y<#> = line control flag (type = Y U)  
# = 0: turn all line control tags off (default)  
# = 1: generate a line control (type=Y U) type-1  
tag (suppresses dominant representation)  
# = 2: generate a line control (type=Y U) type-2  
tag (suppresses non-dominant  
representation)

Note: type-y general print suggestions are written to the output file by AUTOSET as soon as they are encountered. This means that any such suggestion encountered during the processing of a measure of data will be put out at the **beginning** of the measure (i.e., immediately following the previous bar line.

z<#><string> = abbreviated part name flag (type= Y P)  
# = font number. (0 = flag OFF)  
<string> (when # > 0) = the abbreviated part name.  
Use the "\_" character to indicate a blank. Use  
the "/" character to indicate a second  
instrument.

Note: type-z general print suggestions are written to the output file by AUTOSET as soon as they are encountered. This means that any such suggestion encountered during the processing of a measure of data will be put out at the **beginning** of the measure (i.e., immediately following the previous bar line.

### Summary of Control codes

' '	= extra note in chord	@	= single line comment
\$	= controlling musical attributes	a	= append to previous line
&	= comment mode toggle switch	b	= backspace in time
*	= musical directions	c	= cue size note
A	= regular note	f	= figured harmony
B	= " "	g	= grace note
C	= " "	i	= invisible rest
D	= " "	m	= bar line
E	= " "	r	= regular rest
F	= " "	S	= sound directions
G	= " "	P	= print suggestions
/	= end of music or end of file		