Description of ts(.) array in AUTOSET

P 2, action

Purpose: Action is basically a continuation of the music processing program. It is called when the data for a measure is complete.

Inputs: @n = number of elements in data arrays
tv1(.) = element type

type	element
1	regular note
2	extra regular note in chord
3	regular rest
4	cue note
5	extra cue note in chord
6	cue rest
7	grace note or grace rest
8	extra grace note in chord
9	figured harmony
10	bar line
11	musical direction
12	invisable rest
13	backspace
14	clef change
15	time designation or other directive
16	time change
17	change in divspq
18	key change
19	print suggestion

```
tv2(.) = duration for types 1--9, 12,13
       = measure number for type 10
       = track number for type 11 (1 = default)
       = new clef number for type 14
       = 0 for type 15
       = new time flag for type 16
       = new divspq for type 17
       = new key for type 18
       = type of suggestion for type 19

           if between 0 and 7
             0 = force slur 1 over
             1 = force slur 1 under
             2 = force slur 2 over
             3 = force slur 2 under
             4 = force slur 3 over
             5 = force slur 3 under
             6 = force slur 4 over
             7 = force slur 4 under

           if between 8 and 9
```

```
                         8 = overhanded tie (tips down)
                         9 = underhanded tie (tips up)

                       if between 16 and 31 (03-21-97)
                         bit 0: clear = square tuplet
                                set   = round tuplet
                         bit 1: clear = broken tuplet
                                set   = continuous tuplet
                         bit 2: clear = number outside tuplet
                                set   = number inside tuplet

                       if between 128 and 255
                         font = type - 128

                       if between 0x100 and 0x1ff
                         vert and/or horz adj to musical dir

                       if between 0x200 and 0x2ff
                         vert and/or horz adj to sub-obj

                       if between 0x300 and 0x3ff
                         vert and/or horz adj to note/rest/fig
                           objects.

                       if between 0x400 and 0x4ff
                         suggestion for barline or measure

            tv3(.) & 0x00ff = staff number  (0 or 1)

                   For notes,
                   & 0xff00 = value of repeater_flag
                   For rests,
                   & 0xff00 = value of restplace
                   For musical directions
                   & 0xff00 = value of optional forward
                                offset for division counter

            tv5(.) used for flagging $ data that occurs
                     at the beginning of a measure, but
                     is not typeset immediately

          tcode(.) = pitch (rest) for types 1--8
                   = number of figure fields for type 9
                       (figured harmony)
                   = bar type for type 10
                   = musical direction code and position
                       for type 11
                   = "ires" for type 12
                   = "back" for type 13
                   = "0" or "128" for type 14 (clef change)
                   = "" for types 15--18
                   = for type 19 (print suggestions)
                       a 4 byte code
                     byte 1: 0x01: active flag (0 = inactive)
                             0xfe: various meanings
                     (for ties only)
                             length modification (+128)
                               (0 = no data)
                     (for start slurs ([{z only)
                             curvature modification (+128)
                               (0 = no data)

                     byte 2: x-y active flags
                             0x01: active flag (0 = inactive)
                             0x02: 0 = x position is relative
                                   1 = x position is absolute
                             0x04: 0 = y position is relative
                                   1 = y position is absolute
                     byte 3: x position data (+128) (0=no data)
                     byte 4: y position data (+128) (0=no data)

          tdata(.) = additional data for types 1--9, 11, 19


Output:  ts(.,.)
                      Description of ts
                 --------------------------

 Case I:  Notes, Rests, Grace Notes, Cue Notes, Cue Rests
             Extra Regular, Grace, and Cue notes in Chords
                 (types  1--8)

    ts(1) = type:    1 = note
                     2 = extra regular note in chord
                     3 = rest
                     4 = cue note
                     5 = extra cue note in chord
                     6 = cue rest
                     7 = grace note or grace rest
                     8 = extra grace note in chord

    ts(2) = division number (starting with 1)
    ts(3) = clave    <100 = clave number
                      100 = rest
                      101 = movable rest
                      200 = irest

    ts(4) (used initially to store pointer to tcode(.) )

    ts(4) = accidental flag

       bits 0x0f: 0 = none          6 = natural-sharp
                  1 = natural       7 = natural-flat
                  2 = sharp        10 = double sharp
                  3 = flat         15 = double flat

       bit  0x10: 0 = regular    1 = "silent"
```

bits 0xff00: left shift (positioning)

ts(5) = note type

1 = 256th note	7 = quarter note
2 = 128th note	8 = half note
3 = 64th note	9 = whole note
4 = 32nd note	10 = breve
5 = 16th note	11 = longa
6 = 8th note	12 = eighth with slash

ts(6) = dot flag   0 = no dot, 1 = dot, 2 = double dot
ts(7) = tuplet flag   0 = no tuplet, # = tuplet
ts(8) = location on staff
ts(9) = spacing number
ts(10) = stem/chord flag

bit	clear	set
0	no stem	stem
1	step up	stem down
2	single note	chord
3	first note	extra note
4-7	(note number in chord)	

ts(11) = beam flag

0 = no beam	
1 = end beam	
2 = start beam	
3 = continue beam	

ts(12) = beam code (up to six digits)

This is an integer less than 1000000. The one's digit is the code for the eighth beam; the tens digit is the code for the sixteenth beam, etc.

digit	char	meaning
0	blank	no beam
1	=	continued beam
2	[	begin beam
3	]	end beam
4	/	forward hook
5	\	backward hook
6		simple repeater
7		begin repeated beam
8		end repeated beam

ts(13) = local x-offset (for chords)

ts(14) = superflag

bit	set
0	tie

1	begin ~~~~~ without tr.
2	begin ~~~~~ with tr.
3	end ~~~~~
4	begin tuplet
5	end tuplet
6	tuple has a bracket
7	bracket is continuous (0 = broken)
8	number is inside (0 = outside)
9	bracket is round (0 = square)
16	tie is editorial (dotted)
17	~~~ is editorial
18	tuple is editorial

ts(15) = slurflag

bit set	meaning
0	start slur1 (new slur)
1	stop slur1 (from prev. note)
2	start slur2 (etc.)
3	stop slur2
4	start slur3
5	stop slur3
6	start slur4
7	stop slur4

<u>for editorial slurs</u>

16	start slur1 (new slur)
17	stop slur1 (from prev. note)
18	start slur2 (etc.)
19	stop slur2
20	start slur3
21	stop slur3
22	start slur4
23	stop slur4

<u>for both kinds of slurs</u>

8	force slur1
9	0 = up, 1 = down
10	force slur2
11	0 = up, 1 = down
12	force slur3
13	0 = up, 1 = down
14	force slur4
15	0 = up, 1 = down

<u>for ties</u>

24	specify tie orientation

```
                   25      0 = overhand; 1 = underhand

ts(16) = subflag 1  bit     item
                    -----   -------
                    0-3     ornaments
                            ---------
                             0 = none
                             1 = turn
                             2 = trill(tr.)
                             3 = shake
                             4 = mordent
                             5 = delayed turn
                            6-15 (available)

                    4-9     accidental combinations
                               with ornaments
                            -----------------------
                            4-6 accidental above ornament
                            7-9 accidental below ornament

                            Accidental code
                            ---------------
                             0 = none
                             1 = sharp-sharp
                             2 = flat-flat
                             3 = sharp
                             4 = natural
                             5 = flat
                             6 = (not used)
                             7 = (not used)

                   10--13   dynamics
                            --------
                               0 = none
                               1 = p
                               2 = pp
                               3 = ppp
                               4 = pppp
                               5 = f
                               6 = ff
                               7 = fff
                               8 = ffff
                               9 = mp
                              10 = mf
                              11 = fp
                              12 = sfp
                              13 = sf
                              14 = sfz
                              15 = rfz

                      14    upright fermata
                      15    inverted fermata
```

```
                              16    print note in cue size
                              17    editorial accidental
                              18    cautionary accidental


ts(17) = subflag 2          bit     item
                           -----    -------
                 n           0      down bow
                 v           1      up bow
                 i           2      spiccato
                 .           3      staccato
                 =           4      line over dot
                 _           5      legato
                 >           6      horizontal accent
                 A           7      vertical sfortzando accent
                 V           8      vertical sfortzando accent
                 o           9      harmonic
                 Q          10      thumb (*)
                 0          11      open string (0)

                           12-31    fingering (up to 5 numbers)
                                    ----------
                           12-14    first number
                                      0 = no number
                                      1 = finger 1
                                      2 = finger 2
                                      3 = finger 3
                                      4 = finger 4
                                      5 = finger 5
                             15     substitution bit
                                      0 = no substitution
                                      1 = substitution
                           16-19   (second number, see 12 to 15)
                           20-23   (third  number)
                           24-27   (fourth number)
                           28-31   (fifth  number)


ts(18) = used for sorting, later used to indicate position
              of virtual note head (for placing slurs and
              other articulations and signs).  bit 24
              set if modified
ts(19) = used for sorting, later used to indicate global
              x-offset for chord groups
ts(20) = index to ASCII tsdata
ts(21) = pass number
ts(22) = backtie flag (for regular, chord and cue notes)

              0 = this note is not backward tied
              # = this note is backward tied

          Actually the BACKTIE flag has multiple uses.
```

(1) When the ts array is first being constructed, there may be a tie **into** this group of notes from a previous measure. In this case, a tiearr ROW element has already been constructed. The tiearr rows need to be searched and the proper one found. This index (+ INT10000) is then stored as the backtie flag.

(2) For all other row elements of the ts array, it is sufficient to store a back pointer to the ts row that originated the tie.

(3) When it comes time to process the ts array, three cases may be encountered.

(a) There is a non-zero backtie flag, and this flag is greater than INT10000. In this case, the backtie flag (- INT10000) points to a tiearr ROW element, and the tie may be processed.

(b) There is a forward tie from this note. In this case, the backtie flag has already been used to set a tie and the element is now free for other use. We can generate a new row element in tiearr, and place the pointer to this element in the backtie flag (the term "backtie" is now a misnomer).

(c) Now when we encounter a non-zero backtie flag in a new ts ROW, we know this points to a previous ts row, from which we can get the pointer to the relevant tiearr ROW in that ts(,.BACKTIE).

For this method to work properly, it is necessary that backward ties be processed before forward ties. When a backward tie is processed it is important to set the backtie flag to zero.

```
ts(23) = note duration (in divisions)
ts(24) = increment distance flag

         0 -- fixed distance (not to be modified by print)
         # -- variable distance; # = time elaps between
              this  node and next node.
              (576 divisions = quarter note)

ts(25) = virtual end of stem (bit 24 set if modified)
ts(26) = editorial version of ts(16), subflag 1
ts(27) = editorial version of ts(17), subflag 2
ts(28) = staff number
ts(29) = multi-track flag << 2 + mcat flag

         multi-track flag
```

0 = this note lies on a staff that has notes from only one pass (the simplest and most common situation).
1 = this note belongs to one of multiple passes on this staff and all notes on this pass have stems which point up
2 = this note belongs to one of multiple passes on this staff and all notes on this pass have stems which point down
3 = this note belongs to one of multiple passes on this staff and the notes for at least one these passes have stem directions which are both up and down

mcat flag

0 = only one independent instrument represented in this measure (vflag = 1)
1 = more than one independent instrument (vflag > 1) but only one pass and without chords (either unison part, or single part)
2 = more than one independent instrument (vflag > 1) but only one pass but with chords (more than one part, but parts are isorhythmic)
3 = more than one independent instrument (vflag > 1) and more than one pass (two or more musically independent parts)

ts(30) = spacing parameter (1 <= spn <= 6913)
ts(31) = y position of object (saves time in proc. chords)
ts(32) = pointer to extra ts() row element for storing data on slurs. Elements 1-6 of new element are for storing global data on slurs entering and leaving the note. Elements 7-42 are taken in groups of three (expanded from two in **05/06/03** code revision), making a total of 12 such groups. Each group describes a slur entering or leaving this note. The first element in the group contains general information + the x-offset; the second element in the group contains the y-offset. The third element in the group contains the integer equivalent of the 4-byte print suggestion for the slur. See **TS32** for more information.

ts(33) = node shift flag (positive and negative values)

ts(34) = tsr pointer

Case II: Figures

ts(1) = 9
ts(2) = division number (starting with 1)

```
 ts(3) = number of figures in this chord
 ts(4) = space parameter
 ts(5) = first figure -- position one
 ts(6) = first figure -- position two
 ts(7) = first start/stop flag for continuation line
 ts(8) = second figure -- position one
 ts(9) = second figure -- position two
ts(10) = second start/stop flag for continuation line
ts(11) = third figure -- position one
ts(12) = third figure -- position two
ts(13) = third start/stop flag for continuation line
ts(14) = fourth figure -- position one
ts(15) = fourth figure -- position two
ts(16) = fourth start/stop flag for continuation line

      figure field:  0 = blank
                  1-19 = figure
                    20 = +
                    21 = x
                    22 = 2+
                    23 = sharp
                    24 = 4+
                    25 = 5+
                    26 = 6\
                    27 = 7\
                    28 = natural
                    29 = flat
                    30 = short continuation line (-)

      Adding 1000 to figure field (position one) indicates
        small parentheses around the field.
      Adding 2000 to figure field (position one) indicates
        large parentheses this figure and the one below it.
      Adding 3000 to figure field (position one) indicates
        large parentheses this figure and the two below it.
                                             (Added 11/16/03)

      start/stop continuation flag:  0 = none
                                     1 = stop
                                     2 = start
                                     3 = continue

ts(20) = minimum space for figure group
ts(21) = pass number
ts(23) = figure duration in divisions (0 if not given)
ts(24) = increment distance flag (see notes)
ts(28) = staff number
```

Case III: Bar Lines

```
 ts(1) = 10
 ts(2) = division number (starting with 1)
 ts(3) = bar number (0 = none)
 ts(4) = bar type
```

```
                 1 = regular       5 = double regular
                 2 = heavy         6 = regular-heavy
                 3 = dotted        9 = heavy-regular
                                  10 = heavy-heavy

   ts(5) = repeat flag

                 0 = no repeats         1 = forward repeat
                 2 = back repeat        3 = both repeats

   ts(6) = backward ending flag

                 0 = no ending
                 # = ending number: positive = stop ending
                                    negative = discontinue
                                                 ending

   ts(7) = forward ending flag

                 0 = no ending
                 # = ending number

   ts(8) = flags

               bit        set             clear
              -----   ------------      -------
                0     continue ~~~      stop ~~~
                1     segno sign         0
                2     fermata over bar   0
                3     fermata under bar  0

   ts(9) = space parameter (important for non-contr. bars)
  ts(10) = number over previous measure: 0 = none

  ts(20) = index to ASCII tsdata         taken out
  ts(21) = pass number
  ts(28) = number of staves

Case IV:  Signs, Words, Marks

   ts(1) = type   sign = 11, words = 12, mark = 13
   ts(2) = division number (starting with 1)
   ts(3) = vertical position flag:  1 = below line
                                    2 = above line
   ts(4) = sign number

               0 = no sign
               1 = segno
               2 = ped
               3 = *
               4 = other letter dynamics
               5 = D.S or D.C. (right justified string)
               6 = fine (centered string)
```

```
                 7 = words (left justified string)
                 8 = tie terminator       (added **10-12-96**)

      ts(5) = super flag

                 0 = no super-object
                 1 = start wedge
                 2 = stop wedge
                 3 = start dashes (after words)
                 4 = stop dashes
                 5 = start 8ve up
                 6 = stop  8ve up
                 7 = start 8ve down
                 8 = stop  8ve down
                 9 = start 15 up
                10 = stop  15 up
                11 = start 15 down
                12 = stop  15 down
                13 = normal transposition (temporary flag)

      ts(6) = parameter for words:  optional font designation

      ts(7) = wedge offset  (for cases where a wedge begins after
                              or stops at a letter dynamic)

      ts(8) = track number  (useful for multiple wedges, dashes
                              or transpositions of the same type)

      ts(9) = spacing (for case of isolated mark)

     ts(10) = parameter for wedges: wedge spread

     ts(11) = parameter for musical directions which are
                objects: position shift

     ts(12) = special flag for case where this element is
                isolated on a division (possibly with other
                members of this same group).

     ts(13) = parameter for musical directions which are
                super-objects: position shift

     ts(20) = index to ASCII tsdata

     ts(21) = pass number

     ts(22) = backtie flag (for tie terminators) (added 10-12-96)

     ts(28) = staff number

   Case V:  Clef change in middle of a measure

      ts(1) = type = 14
      ts(2) = division number (starting with 1)
```

```
    ts(3) = clef number
    ts(4) = clef font number
    ts(5) = transpostion flag:

                 1 = notes written octave higher than sound
                 0 = notes written at sound
                -1 = notes written octave lower than sound

    ts(6) = position on staff
    ts(9) = space parameter
   ts(20) = index to ASCII tsdata
   ts(21) = pass number
   ts(28) = staff number
```

Case VI: Time designation in middle of a measure

```
    ts(1) = type = 15
    ts(2) = division number (starting with 1)
    ts(9) = space parameter
   ts(20) = index to ASCII tsdata
   ts(21) = pass number
   ts(28) = staff number
```

Case VII: Meter change in middle of a measure

```
    ts(1) = type = 16
    ts(2) = division number (starting with 1)
    ts(3) = time number (100 time numerator + denominator)
    ts(9) = space parameter
   ts(20) = index to ASCII tsdata
   ts(21) = pass number
   ts(28) = number of currently active staves
```

Case VIII: Change in number of divisions per quarter

```
    ts(1) = type = 17
    ts(2) = division number (starting with 1)
    ts(3) = divisions per quarter
    ts(9) = space parameter
   ts(20) = index to ASCII tsdata
   ts(21) = pass number
```

Case IX: Change in key signature

```
    ts(1) = type = 18
    ts(2) = division number (starting with 1)
    ts(3) = new key signature
    ts(4) = old key signature
    ts(9) = space parameter
   ts(20) = index to ASCII tsdata
   ts(21) = pass number
   ts(28) = number of currently active staves
```