The GUIDO Music Notation Format

Additionnal and new Specifications

Supported by the GUIDO Engine version 1.5.1
Grame - April 2013

---

1. Trill

\trill<dx, dy, adx, ady, size, color, tr, anchor>(chords)

Places a trill on the first note of each chord. Each chord should have two voices which specify the main and auxiliary note. Upward and downward trills can be realised by specifying these notes accordingly.

- **Parameters** (in the order of their default positions):
    - **dx, dy, adx, ady** - graphic offsets of the trill (dx/dy for entire trill, adx/ady only for the accidental)
        - standard values:
            - unit parameters
        - default value: 0
    - **size** - graphic size of trill
        - standard values:
            - unit parameter
        - default value: 1
    - **color** - graphic color of the trill
        - standard values:
            - color strings
        - default value: "black"
    - **tr** - possibility to draw the "tr" or not
        - standard values:
            - "false"
            - "0"
            - "true"
        - default value: "true"
    - **anchor** - possibility to anchor the trill line at the note head or above the staff
        - standard values:
            - "note"
            - "above"
        - default value: "above"

- **Range:** obligatory

- **Semantics:**
    - rapid alternation between two adjacent notes
- **Remarks:**
    - The trill line will be drawn above automaticaly tied notes (split by a staff or a system) but not above notes that have been tied explicitly by the user

Examples:

```
{[\meter<"2/4">\tie( \trill({a} {g} {g}) )],
[\meter<"2/4">\trill({a}{g/2})]}
```
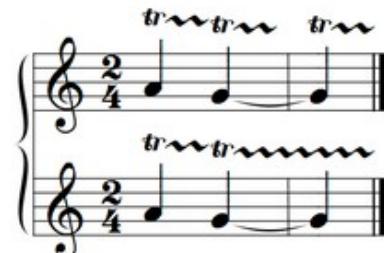


```
[\trill<tr="false", anchor="note">({g}{e/2})]
```

2. Noteheads

2.1 Noteheads

\noteFormat<*type*>

Changes notehead graphical display, where *type* can be one of the following:

- "x" (a cross notehead)
- "diamond" (a diamond notehead)
- "square" (a square notehead)
- "round" (a round notehead)
- "triangle" (a triangle notehead)
- "reversedTriangle" (an upside down triangle notehead)

Examples:

```
[ \noteFormat<"reversedTriangle"> a \noteFormat<"diamond"> c { \noteFormat<"x"> d, \noteFormat<"square"> g } ]
```


2.2 Noteheads brackets

\noteFormat<*style*>

Allows to display brackets around noteheads, where *style* can be one of the following:

- "()" (round brackets)
- "[]" (square brackets)
- "<>" (angled brackets)
- "(S)" (with S being a type, as previously seen)
- "[S]"
- "<S>"

Examples:

```
[ \noteFormat<"()"> a \noteFormat<"[diamond]"> a \noteFormat<"<x>"> a ]
```


---

3. Glissandos

\glissando<*dx1, dy1, dx2, dy2, thickness, lineStyle*>(notes)

Marks glissandos, affects both playback and graphic appearance.

- **Parameters** *(in the order of their default positions):*

    - **dx1, dy1, dx2, dy2** - graphic offsets of the glissando line

        - *standard values:*
            - unit parameters
        - *default value:* 0
    - **fill** - possibility of filling the space between glissandi (for chords or clusters)

        - *standard values:*
            - true
            - false
        - *default value:* false
    - **thickness** - graphic thickness of glissando indication

        - *standard values:*
            - unit parameter
        - *default value:* 0.3
    - **lineStyle** - graphic appearance of glissando indication

        - *standard values:*
            - "line" - mark glissando with straight line
            - "wavy" - marks glissando with wavy line
        - *default value:* "line"

- **Range:** obligatory

- **Semantics:**
    - Indicates a glissando between the outermost notes of its range; if the range contains more than two notes, the inner notes are interpreted as transitory notes in a glissando chain.
- **Remarks:**
    - In the case of chords, the first written note in the chord will be linked to the first note of the next chord, the second one to the second one, etc...

Examples:

```
[\noteFormat<size=1.5, dx=2> \glissando(e g \noteFormat<dx=4, dy=2> f b)]
```

```
[\glissando(e g# d& b& c# b c)]
```



```
[\meter<"2/4"> \glissando<fill="true">({b, g} f/2 {g, c2} {e1, g} {b, f}) \glissando({c/4, f, b} {f, b, c}) \glissando<thickness=1, color = "red">(e2 f1 b)]
```


---

4. Clusters

\cluster<dx, dy, adx2, ady2, size, color>(chords)

Creates clusters, affects graphic appearance.

- *Parameters :*
    - o *dx, dy, adx, ady* - graphic offsets of the cluster (dx/dy for entire cluster, adx/ady only for cluster's head)
        - *standard values:*
            - unit parameters
        - *default value:* 0
    - o *size* - graphic size of cluster (only affects horizontal size)
        - *standard values:*
            - unit parameter
        - *default value:* 1
    - o *color* - graphic color of cluster (only affects cluster's head, use **\noteFormat**-tag to affect all cluster)
        - *standard values:*
            - color strings
        - *default value:* "black"

- *Range:* obligatory

- *Semantics:*
    - o Transforms a chord into a cluster. Only the two first notes are taken into account.

Examples:

```
[ \cluster<color="red">({c,d}{f}{d/2,b}) ]
```


---
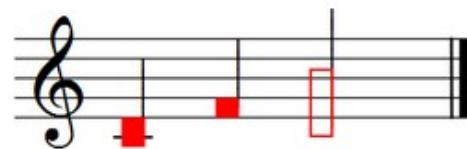
5. staffOff, staffOn

\staffOff
\staffOn

The **\staffOff** tag makes the current staff invisible starting from the position where it occurs until the next \staffOn is encountered. While the staff is invisible, all notational elements fixed to it (notes, rests, dynamic markings, etc.) also become invisible.

- *Parameters:* none
- *Range:* none

- *Remarks:*
    - o Takes into account the changes of staff with \staff<i>
    - o If a \staffOn is placed at the end of the line and you don't want the last bar to be drawn, you have to write it explicitely before the \staffOn.

Examples:

```
[ \meter<"4/4"> \staffOff \clef    g d \staffOn c f c d f \staffOff a c b g \staffOn f ]
```



```
{[ \meter<"4/4"> g d c f c d f a],
 [ \meter<"4/4"> c d f \staff<1> a g \staffOff c \staff<2> g f ]}
```

6. Symbol

\symbol<*file, position, size, w, h, dx, dy*>
\symbol<*file, position, size, w, h, dx, dy*>(notes)

Inserts a symbol in the score, which can be a png/jpg/bmp file and which can modify score's layout.
The symbol gets a duration if some notes are set between brackets, after the **\symbol**-tag.

- *Parameters* :
  - *file* **(required)** - path to the symbol: can be relative from home directory, from current gmn file directory, or absolute.
    - *standard values:*
      - string
  - *position* - graphic position of symbol in relation to the staff
    - *standard values:*
      - "bot"
      - "top"
      - "mid"
    - *default value:* "mid"
  - *size* - graphic size of symbol
    - *standard values:*
      - unit parameter
    - *default value:* 1
  - *w, h* - graphic width/height of the symbol, in pixel
    - *standard values:*
      - unit parameters
    - *default value:* original w/h of the symbol
  - *dx, dy* - graphic offset of the symbol
    - *standard values:*
      - unit parameters
    - *default value:* 0

Examples:

```
[ \meter<"4/4"> a b \symbol<file="test.png", dx=-3> c d ]
```
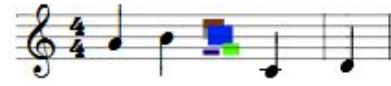


```
[ \meter<"4/4"> a b  \symbol<file="symbols/curve1.png", position="top", dy=-1, h=13, w=50> (c d ) ]
```


---

7. Feathered Beaming

\fBeam<*durations, drawDuration, color*>(notes shorter than a quarter note)

Creates a feathered beam

- *Parameters* :
  - *color* - graphic color of beams
    - *standard values:*
      - color strings
    - *default value:* "black"
  - *durations* - begin- and end- durations of the feathered beam : "beginDuration,endDuration"
    - *standard values:* string values
      - 1/8
      - 1/16
      - 1/32
      - 1/64
    - *default value:* if not specified, the real durations of the first and last notes
    - *remark:* if the specified durations are not standard, we will take the closest standard duration
  - *drawDuration* - possibility to draw the duration of the entire beaming
    - *standard values:* boolean
    - *default value:* false

- *Range:* obligatory

- *Semantics:*
  - Indicates an acceleration or a deceleration within a group of beamed notes
- *Remarks:*
  - The spacing between notes stays the same as usual : it depends on the inner duration explicitly given by the user (and not the graphic appearance given by the number of beams)
  - The beams can be chained (using the \fBeamBegin \fBeamEnd tags), combined, and included in bigger beams.

Examples:

```
[ \fBeam<durations="1/8,1/32">(f/16 e g f c d e a e f) \fBeam<durations="1/32,1/16", drawDuration="true">(b e d a d e)]
```
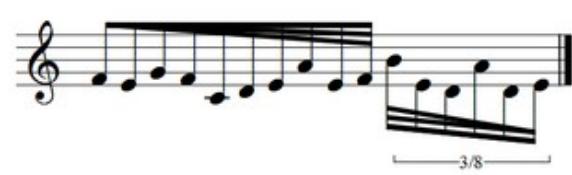
[\fBeamBegin:1<drawDuration = "true"> a/8 e f e/16 d g f/32 \fBeamBegin:2<drawDuration = "true"> a/64 \fBeamEnd:1 a f e f/32 e g a/16 e d/8 \fBeamEnd:2]
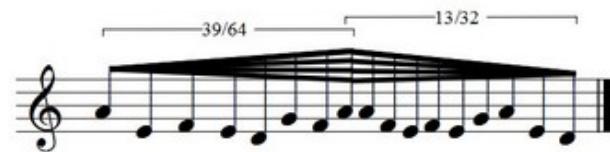
[\beam(\fBeam(a/16 f2 g1 b/32) \fBeam(d/32 c b a g/64) \fBeam(e d c d/8)) \beam(\beam(a/8 f g/32 b) \beam(d c b a g2/64) \beam(e d c a/32))]