

## TECHNICAL NOTE

Subject: MusicXML—the conundrum of the element “duration”

Issue: 1 File: TN-MXML-conundrum\_duration-WP.doc

Date: May 20, 2021

Author: Douglas A. Kerr, P.E. (Ret.)

To: File

### ABSTRACT AND INTRODUCTION

The MusicXML language, used to represent musical scores, includes as part of the encoding for a note the element `<duration>`, for which a seemingly-definition is given by the MusicXML Documentation. But various factors make unclear the exact significance of that definition. The result is that different implementers have adopted different interpretations of that definition, leading to the development of differing, and incompatible, protocols, with the result that the interchange of scores via MusicXML files is often imperfect.

In this note the author gives a succinct review of this conundrum and makes a recommendation for an singular interpretation of the significance of the element `<duration>` upon which to base protocols for note encoding and decoding. This is then followed by a complete recommended protocol for this area.

### 1. NOTATION

In this note, MusicXML elements will be shown thus: `<note>`, which is evocative of the basic form of the opening XML tag for the element.

MusicXML attributes will be identified by being shown underlined, thus: attack.

### 2. THE UNIT OF MUSICAL TIME

The unit of musical times used in various elements discussed herein is the *division*, which is defined as a stated fraction of the time value of a quarter note.

The value of a *division* in a given MusicXML file is declared via the element `<divisions>`. The value of that element is the number of *divisions* in the *time value* of a quarter note.

### 3. THE ELEMENT `<duration>`

#### 3.1 Syntactic requirement

The MusicXML syntactic specification requires that every `<note>` element must contain one and only one `<duration>` element.

#### 3.2 The unit

The value of the element `<division>` is given in *divisions* (see section 2).

#### 3.3 The formal definition

##### 3.3.1 *In the “specification”*

The “specification” part of the MusicXML documentation gives this definition of the element `<duration>`:

Duration is a positive number specified in division units. This is the intended duration vs. notated duration (for instance, swing eighths vs. even eighths, or differences in dotted notes in Baroque-era music). Differences in duration specific to an interpretation or performance should use the note element's attack and release attributes.

##### 3.3.2 *In the tutorial*

This is augmented by this definition given in the tutorial portion of the Documentation:

The duration element is an integer that represents a note's duration in terms of divisions per quarter note. . . .

This may seem to add nothing, but its appearance here is itself of significance (as will be discussed in section 3.5.3).

#### 3.4 Interpretations

Although these two definitions at first seem to be quite clear, it turns out that they do not really pin down the significance of this element. As a consequence, developers have embraced at least three interpretations of the significance of the element `<duration>`

- a. It defines whether the note is a quarter note, half note, or such in terms of its numerical time value. Yes, there is another element, `<type>` (optional), that does this in another way, as an enumerated text value. More on this later
- b. It defines the musical time allocation of the note, by which I mean the amount of musical time that the “domain” of the note occupies.
- c. It defines the intended play duration of the note.

Each of these interpretations has spawned one or more distinct, and mutually incompatible, protocols for the use of this element in MusicXML code.

### 3.5 Credibility of the interpretations

#### 3.5.1 *Interpretation (a)*

Another passage in the Documentation points out that, while a receiving program should be able to discern the note symbol (“quarter note”, etc.), and thus the note’s *time value*, from the value of `<duration>` (read in conjunction with the value of a *division*), its task is simplified if the optional element `<type>` is included in the `<note>` element. That element gives the symbol as an enumerated text value, such as ‘quarter’ or ‘16th’.

This of course gives weight to interpretation (a) as being the intended one.

#### 3.5.2 *Interpretation (b)*

The clause in the basic definition referencing “swing eighths” and the like tends to give weight to interpretation (b) as being the intended one.

#### 3.5.3 *Interpretation (c)*

The MusicXML syntax is said to comprise two aspects, one devoted to the transport of the score as a graphic document and one (called the “MIDI-compatible part”) devoted to how the score should sound if “played” by a receiving program.

In the tutorial, the passage cited in section 3.3.2 defining `<duration>` is in the “MIDI Compatible Part” portion of the tutorial. This lends weight to interpretation (c) as being the intended one.

## 4. HOW TO RESOLVE THIS CONUNDRUM

In resolving this conundrum I have relied not only on intensive analysis of the MusicXML documentation but also on remarks made by Michael Good, the original developer of the MusicXML language and its current *de facto* “keeper”, in various forums.

In one of these comments, he observed that the use of “duration” in the sense of “swings eighths” is essentially obsolete, and this such departure from the “normal” timing implication of the notes is today generally done by way of the attack and release attributes of the `<note>` element, which describe departures of the start and end play times respectively from those implied by the note symbol.

This to me tends to weaken the argument in favor of interpretation (b).

In another comment, he suggests that conveying the intended play duration of the note in MusicXML code should also be done using the attack and release attributes.

This to me tends to weaken the argument in favor of interpretation (c).

That leaves me with interpretation (a).

## 5. THE PROTOCOL IMPLICATION

This then leads to the conclusion that, if a `<note>` element contains, in addition to the mandatory `<duration>` element, a `<type>` element (optional), these two elements should (in their respective fashions) describe the same note symbol (and thus by implication the same note *time value*).

This is of course a rather oddly-redundant conclusion, but I believe this situation has unfolded as the MusicXML language (and its actual usage) has evolved.

## 6. A RECOMMENDED PROTOCOL

### 6.1 Introduction

I recommend the following protocol for use in MusicXML-enabled application programs as to the area of note timing matters:

### 6.2 Encoding into MusicXML

- a. The value of `<divisions>` shall be chosen large enough to accommodate the needed precision of the element `<duration>` and the attributes attack and release, if used.
- b. The `<note>` element shall be provided with the subelements `<type>` and `<duration>`.
- c. The `<type>` element shall describe the symbol of the note (as an enumerative text variable, such as 'quarter' or 'whole'). This also, in accordance with accepted musical practice, defines the *time value* of the note.
- d. The `<duration>` element shall (interpreted in connection with the value of `<divisions>`) define the time value of the note (and thus its symbol).

Comment: Yes, this seems redundant. This situation is a result of the convoluted evolution of this aspect of MusicXML encoding.

- e. If it is the intent that the play duration of the note is different than that implied by the time value of the note, and/or that the start of the sounding of the note is displaced from the time instant implied by the location of the note in the measure, then the attributes (of the `<note>` element) attack and or release shall be included.
- f. If the intent is that the starting play time differs from the "nominal", then the attribute attack shall be provided. It specifies the offset of the starting play time from nominal as a signed value in units of *divisions*.
- g. If the intent is that the ending play time differs from the "nominal", then the attribute release shall be provided. It specifies the offset of the ending play time from nominal as a signed value in units of *divisions*.

Note that if the situation is a simple departure of the play duration from nominal, that is treated here as a displacement of the ending play time. to be conveyed with release.

### 6.3 Response to an imported MusicXML file

- a. If the optional element `<type>` is provided in a `<note>` element. It indicates (as, for example, “quarter”) the symbol of the note and thus its time value.
- b. If the optional element `<type>` is provided, the mandatory element `<duration>` is ignored.

Comment: This oddity is a result of the convoluted evolution of this aspect of MusicXML encoding.

- c. If the optional element `<type>` is not provided, the mandatory element `<duration>` is to be taken as giving (in numerical form, interpreted in connection with the value of `<divisions>`) the time value of the note and thus its symbol.
- d. The “nominal” play duration is to be as implied by the time value of the note. The nominal starting play time is at the instant implied by the position of the note in the measure.
- e. If the optional attribute attack is provided, it defines, in units of *division*, the displacement (in either direction—its value is signed) of the starting play time from its nominal value.

If the optional attribute release is provided, it defines, in units of *division*, the displacement (in either direction—its value is signed) of the ending play time from its nominal value.

## 7. ISSUE RECORD

Issue 1, May 20, 2021 (this issue). Initial issue.