

TECHNICAL NOTE

Subject: MusicXML—the duration element—overview

Issue: 1

File: TN-MXML-duration-overview-WP.doc

Date: May 15, 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 includes several objects, including the element `<duration>`, for which, although the MusicXML documentation gives seemingly thorough definitions, the overall context is not fully defined. The result is that different implementers have adopted different interpretations, with the result that the interchange of score via MusicXML files is imperfect.

In this note the author give a succinct description of this situation and makes a recommendation for a “doctrine” to be adopted by implementers.

An appendix discusses the impact of this matter on transporting a score via MusicXML between the notation programs Overture and MuseScore.

1. ADDITIONAL INFORMATION

This note is intentionally concise. The reader who is interested in a deeper discussion may wish to read the Technical Report “MusicXML—the duration element” by the same author.

2. 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.

3. THE SYNTACTIC INGREDIENTS

The MusicXML syntactic ingredients that are relevant to this issue are:

- Element: **<divisions>**. Defines the unit of numerical musical time definitions, the *division*, in terms of the number of those in the musical time of a quarter note.
- Element: **<note>**. This defines a note (and, as a subcase, a rest).
- Subelement of **<note>**: **<type>**. This is optional, and may or may not appear (once) in a given **<note>** element. This tells in an enumerative text value (*e.g.*, “quarter”) the symbol for the note, and thus the *time value* of the note.
- Subelement of note: **<duration>**. This is mandatory, and must appear (once) in a given **<note>** element. Its significance is a bit unclear (hence this note).
- Attribute of **<note>**: **attack**. This is optional, and may or may not appear (once) in the opening tag of a given **<note>** element. It tells in (musical time units of *divisions*) how the actual play start time of the note is to be displaced from its nominal time (it not always being clear what that nominal time is).
- Attribute of **<note>**: **release**. This is optional, and may or may not appear (once) in the opening tag of a given **<note>** element. It tells (in musical time units of *divisions*) how the actual play end time of the note is to be displaced from its nominal time (it not always being clear what that nominal time is).

4. SYNTACTIC DOCTRINES

4.1 Introduction

Probably because of the inadequacy of description of the overall intended syntactic scheme in this area in the MusicXML Documentation, various interpretations of the significance of the element **<duration>** have led to several overall syntactic doctrines. With some of the esoteric details overlooked, these include (as “fleshed out” by the author):

4.2 Doctrine A

- 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.
- c. If the optional element **<type>** is not provided, the mandatory element **<duration>** is to be taken as giving (in numerical form) the symbol for the note and thus its musical time.
- d. The “base” play duration is to be as implied by the time value of the note. The base play start 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 start play time from its base value.
- f. If the optional attribute release is provided, it defines, in units of *division*, the displacement (in either direction—its value is signed) of the end play time from its base value.

The author considers that this is consistent with the doctrine intended by the MusicXML Documentation and its developer.

4.3 Doctrine B

- a. The optional element `<type>` is always provided in a `<note>` element. It indicates by an enumerative text value (as, for example, 'quarter') the symbol of the note and thus its musical time.
- b. The mandatory element `<duration>` is taken as giving (in units of *divisions*) the actual play duration of the note (it being assumed that the play of the note starts at the nominal time).
- c. If the optional attributes attack or release appear, they are ignored. They are also not included in encoding.

This concept provides a simple way of describing in the MusicXML code the intended play duration of each note.

The author, however, does not believe that this doctrine is consistent with the actual intent of the MusicXML Documentation.

4.4 Doctrine MS

This is the doctrine followed by the notation program, MuseScore (as of version 3.6.2).

- a. It is assumed that 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. The mandatory element `<duration>` is taken as defining (in units of *divisions*) the *musical time allocation* of the note, in the sense of the amount of musical time from the nominal start of this note to the nominal start of the next note.
- c. The play duration of the note is made as implied by its *time value*.
- d. If the optional attributes attack or release appear, they are ignored. They are not used in encoding.

5. PRACTICE IN THE FIELD

5.1 Introduction

The following sections describe, concisely, the practice found in current versions of some notation programs. There are in most cases any number of small details that are overlooked by these descriptions.

5.2 Overture

Overture, as of version 5.6.3-3, essentially follows Doctrine B (see section 4.2) both with respect to the encoding of a MusicXML file for export and as to the response to an imported MusicXML file.

5.3 MuseScore

MuseScore, as of version 3.6.2, follows Doctrine MS (see section 4.4) with respect to the response to an imported MusicXML file.

With respect to the generation of a MusicXML file for export:

- a. The optional element `<type>` is included in each `<note>` element. It indicates (as, for example, 'quarter') the symbol of the note and thus its *time value*.
- b. The mandatory element `<duration>` is included in each `<note element>`, and gives a duration equal to that implied by the time value of the note.
- c. Any departure of the start play time or the play duration of the note from the nominal are not reflected in the MusicXML encoding.
- d. The optional attributes `attack` or `release` are not used.

5.4 Finale

The behavior of Finale in this matter is, unexpectedly, rather bizarre, and its description is beyond the scope of this note. The reader who is interested in the details will find them in the Technical Report cited in section 1.

6. RECOMMENDATION

The author recommends the adoption of Doctrine A (as described in section 4.2) for the use of MusicXML to transport musical scores.

7. ISSUE RECORD

Issue 1, May 14, 2021 (this issue). Initial issue.

Appendix A TRANSPORT OF SCORES FROM OVERTURE TO MUSESORE

Of current interest in the author's circle is the transport of scores via MusicXML from Overture to MuseScore. Overture practices doctrine B (as described in section 4.3), while MuseScore practices Doctrine MS (section 4.4). Thus we can readily imagine that this transport will not produce a satisfactory result in MuseScore.

MuseScore treats the value of the <duration> element as defining the "musical time allocation" for the note. If for example a measure had four quarter notes whose play duration in Overture was 75% of "face", then the <duration> elements of those notes would have a value of 75% of a quarter note (corresponding to the time value of a dotted eighth note).

MuseScore then, in response to the <type> elements ('quarter'), puts on the reconstructed score quarter note symbols, but on the "musical timeline" it (in response to the <duration> elements) spaces them at intervals of a dotted eighth note.

Thus when the score is played, the notes start sounding at intervals of a dotted eighth note, but are sounded for the duration of a quarter note (their "face time").

Also on the time line, MuseScore considers this measure to have a duration of 3 quarter notes. And so the last quarter note sounds well into the next measure. A small hyphen above the end of the measure tells us that there are not enough musical time in the notes in it to fill its time as indicated by the time signature

We can see that here (the horizontal grid divisions correspond to a 16th note):

The screenshot displays a musical score in MuseScore. At the top, it indicates a tempo of ♩ = 96 and a note: "All play durations 75% and start times nominal". The staff shows a 4/4 measure with four quarter notes. Below the staff, the piano roll view shows a grid where each horizontal division represents a 16th note. The notes are labeled with their pitch classes: C, D, E, F, G, A, B, C. The notes are spaced at intervals of a dotted eighth note, and their duration is 75% of a quarter note. A small hyphen above the end of the measure indicates that the notes do not fill the measure's time.

There is a further fly in the ointment. The typical Overture score will have most (maybe all) notes set for a play duration of perhaps 90% of “face value” (the musical time implied by the note symbol). These play durations are reflected by the values of the <duration> elements of the various notes.

But MuseScore treats the values of <duration> as indicating the musical time allotment of the notes (the realm each note presides over in musical time space, regardless of the play duration it is somehow given).

When a MusicXML file is imported into <MuseScore, it is subject to a syntactic validation. A higher-level part of this is to add up the values of the <duration> elements of all the notes in each measure and match that to the musical time length of the measures (based on the time signature encoded in the MusicXML file. If these do not match, an error message is given before the import will be consummated.

But in the case discussed here, these figures will not match. The sum of the values of the <duration> elements in the MusicXML file from Overture for the notes in one measure will typically be perhaps 90% of the musical time length of the measure. Thus indeed this error message is encountered.

An additional complication until the latest version of Overture (5.6.3-3) was that Overture committed a syntactic error in composing a MusicXML file (an issue outside the scope of this note) which led to another error message in the import procedure into MuseScore. But that error has now been corrected.

-#-