makemusic™ CLOUD (smartmusic.)

MusicXML and Repertoire Development

Michael Good
VP of MusicXML Technologies

Agenda

- Introduction to MusicXML
- MusicXML issues: export or import?
- MusicXML resources
- Looking to the future

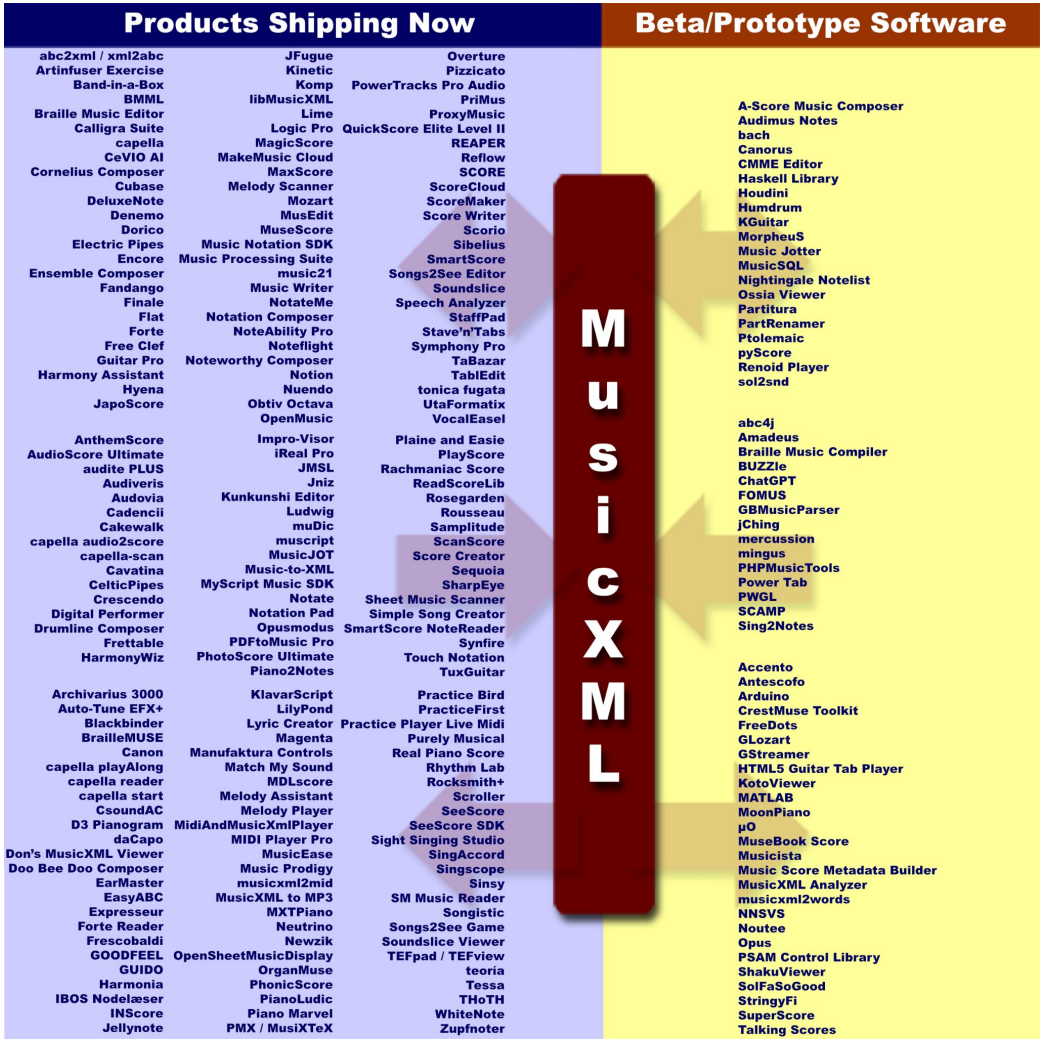
Introduction to MusicXML

What is MusicXML?

- The standard, open format for exchanging digital sheet music between applications
- Invented by Michael Good at Recordare in 2000
- Managed by W3C Music Notation Community Group since 2015
- Developed collaboratively by a community of hundreds of musicians and developers over the past 22 years
- Available under an open, royalty-free license suitable for both open-source and proprietary software
- Supported by 270 applications worldwide

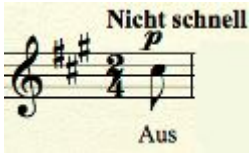
Who Uses MusicXML

Usage map as of December 2022



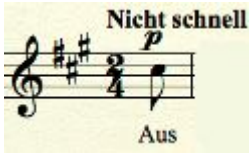
MusicXML Is a Notation Format

- **Not a graphical format like PDF**
- **Music is represented using the semantic concepts behind common Western music notation**
- **Includes both how a score looks and how it plays back**
  - Sometimes there are pairs of elements
  - <tie> for playback, <tied> for appearance
  - <alter> for playback, <accidental> for appearance
  - <duration> for playback, <type> for appearance
- **Includes low-level details of the appearance of a particular engraving, or the nuances of a particular performance**
  - Allows transfer between applications with high visual fidelity
  - Also allows the visual details to be ignored when appropriate



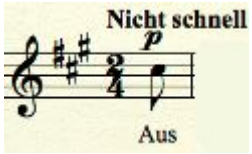
in MusicXML (1 of 3)

```
<part id="P1">
    <measure implicit="yes" number="0" width="179">
      <attributes>
        <divisions>24</divisions>
        <key>
          <fifths>3</fifths>
          <mode>major</mode>
        </key>
        <time>
          <beats>2</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
```



in MusicXML (2 of 3)

```
<direction placement="above" directive="yes">
    <direction-type>
      <words default-y="25" font-size="10.5"
             font-weight="bold">Nicht schnell</words>
    </direction-type>
    <sound tempo="42"/>
  </direction>
  <direction placement="above">
    <direction-type>
      <dynamics default-y="10" relative-x="-5">
        <p/>
      </dynamics>
    </direction-type>
    <sound dynamics="54"/>
  </direction>
```



in MusicXML (3 of 3)

```
  <note default-x="141">
     <pitch>
       <step>C</step>
       <alter>1</alter>
       <octave>5</octave>
      </pitch>
      <duration>12</duration>
      <voice>1</voice>
      <type>eighth</type>
      <stem default-y="-50">down</stem>
      <lyric default-y="-80" number="1">
        <syllabic>single</syllabic>
        <text>Aus</text>
      </lyric>
   </note>
</measure>
```

MusicXML as an Archival Format

- **MusicXML is an XML format, with all its advantages:**
  - Files can be opened in any computer text editor
  - Fully internationalized via Unicode
  - Files are human-readable as well as machine-readable
  - Can use all the standard XML tools developed by larger industries than the music industry
- **Backward compatibility: all valid MusicXML 1.0 files are also valid MusicXML 4.0 files**
- **W3C Music Notation Community Group provides a home in the leading web standards organization**
- **Already implemented by 270 programs**

How Did MusicXML Get So Popular?

- **We made it usable by music notation software developers**
  - Scope was limited to common Western music notation from the 17th century on
  - Element and attribute names use musical terms, not computer terms
  - Clarity chosen over concision
  - MusicXML document structure matched that of existing commercial applications like Finale and Sibelius
  - Selective encoding: Apps do not have to handle everything at once, so they can start small and add more features later
- **We supported a market leader early (Finale 2003 on Windows)**
- **The format was developed together with the software**
- **We marketed to developers and supported them**

If It's a Standard, Why Are Apps So Different?

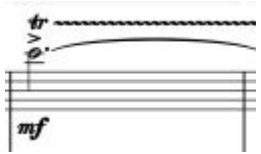
- **Remember this ease-of-use advantage for developers?**
  - "Selective encoding: Apps do not have to handle everything at once, so they can start small and add more features later"
- **That's great for getting more apps to adopt the format, but it also means that different apps support different subsets**
- **We also ran into limitations of a bootstrapped effort in a relatively small industry:**
  - We didn't have a full written specification until MusicXML 4.0
  - Before that, the "spec" was comments in the schema files used to validate MusicXML files in software
  - Thus things weren't as clear as they could have been
  - We still don't have a standard test suite for app developers

MusicXML Issues: Export or Import?

Is It Export or Is It Import?

- **Because MusicXML uses standard musical terms and everything is ordered hierarchically, it's relatively easy to see what is going on**
- **If something doesn't transfer over, look at the measure where there's a problem**
  - Look at the part-list near the top of the file then search for the right part number, e.g. "P17"
  - Then look for the right measure number, e.g. number="24"
  - See if what you're looking for is there and in the right location
  - If yes, it's an import problem
  - If no, it's an export problem

Missing Trill in Flute, Bar 52



- **Look at the part-list:**

```
<part-list>
   <score-part id="P1">
     <part-name>Flute</part-name>
   </score-part>
```

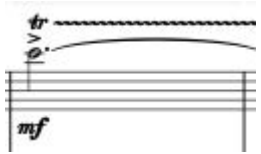
- **Go to P1:**

```
  </part-list>
  <part id="P1">
    <measure number="1">
```

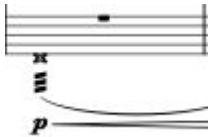
- **Go to measure number="52"**

No <trill> in the MusicXML: Export Issue

```
<measure number="52">
  <attributes>
    <divisions>4</divisions>
  </attributes>
  <direction placement="below">
    <direction-type>
      <dynamics>
        <mf/>
      </dynamics>
    </direction-type>
    <staff>1</staff>
  </direction>
  <note>
    <pitch>
      <step>C</step>
      <octave>6</octave>
    </pitch>
    <duration>12</duration>
    <tie type="start"/>
    <voice>1</voice>
    <type>half</type>
    <dot/>
    <stem>down</stem>
    <staff>1</staff>
    <notations>
      <tied type="start"/>
    </notations>
    <notations>
      <articulations>
        <accent placement="above"/>
      </articulations>
    </notations>
  </note>
</measure>
```



Hairpin Location in Percussion 2, Bar 135



- **Look at the part-list:**

```
<score-part id="P25">
  <part-name>Percussion 2  (Low Tom, Bongos, Gong,
Suspended  Cymbal, Wind Chimes,  Triangle)</part-name>
```

- **Go to P25:**

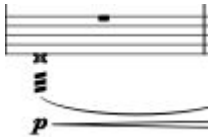
```
</part>
<part id="P25">
  <measure number="1">
```

- **Go to measure number="135"**

Two Types of Issue Here

```
<measure number="135">
  <attributes>
    <divisions>4</divisions>
  </attributes>
  <direction>
    <direction-type>
      <wedge type="crescendo" number="1"/>
    </direction-type>
    <staff>1</staff>
  </direction>
  <direction placement="below">
    <direction-type>
      <dynamics>
        <p/>
      </dynamics>
```

- **Placement is missing, which is an export issue**
- **But hairpins usually go below (unless avoiding lyrics in vocal music), so an import issue too**



MusicXML Resources

W3C Music Notation Community Group

- **This is the home of the MusicXML community and ongoing MusicXML development**
- **Projects include:**
  - MusicXML
  - SMuFL (Standard Music Font Layout)
  - MNX - next generation music notation standard for native web apps
  - Standardized instrument data (not yet started)
- **Three co-chairs:**
  - Adrian Holovaty, Soundslice, MNX editor
  - Daniel Spreadbury, Steinberg, SMuFL editor
  - Michael Cuthbert, MIT, MusicXML editor
- **https://www.w3.org/community/music-notation/**

W3C Music Notation Community Group Membership

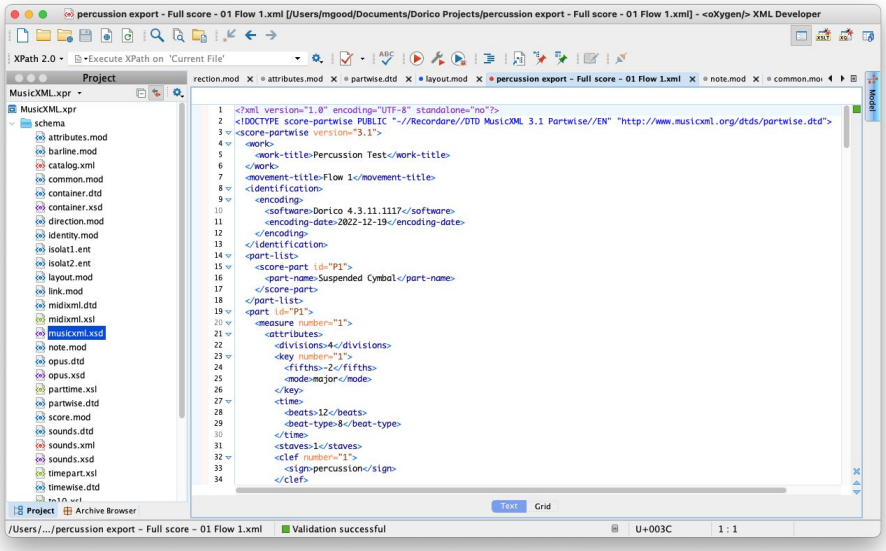
- **Membership is free of charge**
  - Go to the home page and click JOIN OR LEAVE THIS GROUP to start
  - Be sure to do so as an employer representative if you work in this field
- **Membership allows you to contribute to projects**
- **You will also be emailed the co-chair meeting minutes every two weeks, along with other (infrequent) blog posts**
- **Most of the work happens on GitHub**
  - Each project has its own repository
  - MusicXML repository: https://github.com/w3c/musicxml
  - Includes issues (Things to improve in future MusicXML versions)
  - Includes discussions (For general MusicXML Q&A)
  - To track this work, use your GitHub account to follow a repository

The MusicXML 4.0 Specification

- **Available at https://www.w3.org/2021/06/musicxml40/**
- **Includes all the following:**
  - Tutorial
  - Documentation for each element, attribute, and data type
  - Examples for every element, cross-linked to documentation
  - All the schema files (XSD, DTD, and XSLT files)
  - Version history

oXygen XML Developer

- This is the XML editor that I use for my MusicXML work
- Affordable, high quality, and runs on Mac and Windows
- https://www.oxygenxml.com/xml_developer.html

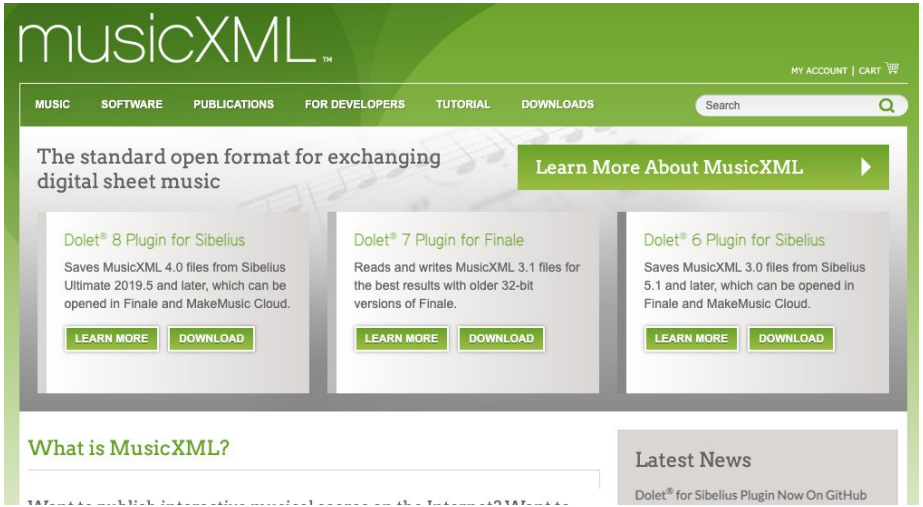


Altova DiffDog

- **This is what I use for regression testing MusicXML export from Finale and Sibelius**
- **Lets you compare individual files and entire directories**
- **Lets you exclude elements and/or attributes that you don't care about**
  - For example, the encoding date and the software version
- **oXygen XML Developer has differencing built-in as well**
  - It's fine for one-off uses
  - But if you do lots of comparisons, the added features in DiffDog are worth it
- **Windows only**
- **https://www.altova.com/diffdog**

The musicxml.com website

- **Contains pointers to lots of resources**
  - All the apps we know about that support MusicXML
  - Places you can find collections of music in MusicXML format
  - Publications that describe how MusicXML is used in academic research



Looking to the Future

The Future of MusicXML

- **Prof. Michael Cuthbert from MIT has taken over as MusicXML spec editor and W3C Music Notation CG co-chair**
  - Inventor and developer of the music21 toolkit for computer-aided musicology
  - Long-time contributor to both the MusicXML and SMuFL specs
  - Long-time implementor of MusicXML within the music21 app
  - His academic background helps balance the industry backgrounds of Daniel and Adrian and should provide some fresh perspectives
- **We expect there to be new MusicXML versions in the future, but we don’t know when**
  - The group’s focus is more on MNX now
  - MusicXML 4.0 is a mature format with lots of features
  - Beyond MakeMusic, many app makers have not taken up version 4.0 yet

Why MNX?

- **MusicXML was designed as a score interchange format**
- **It was deliberately not designed as a native format for notation apps**
  - At the time, the industry did not see the benefit of a music notation standard
  - The only way to get acceptance was to model the printed page
  - That way, nobody could say, “Oh, that’s just what Finale / Sibelius / SCORE does, we can’t / don’t want to do that”
- **Still, people try to use it as a native format for their new apps**
  - And then they discover why this isn’t recommended
  - But by then it can be too late

What Is MNX?

- **MNX is a next-generation standard to handle the increasing demands from app developers for things that MusicXML doesn't do well**
  - Most notably, serve as a native format for music notation apps, especially web-based apps
  - But there are many other use cases too: see https://w3c.github.io/mnx/use-cases/
- **It also learns from our 20+ years of experience with MusicXML to do things better from an application point of view**
  - We can do this now that the music notation industry has experience with the benefits of standards
  - We don't have to model paper scores any more, especially as things move more towards digital

Does That Mean MusicXML Goes Away?

- **That seems very unlikely**
- **MNX is designed to do things where using MusicXML doesn't work very well, but changing MusicXML would break things**
- **These will tend to be different than what MusicXML is used for**
  - MusicXML will still be strong for exchange between applications
  - MNX will be strong for building interactive web apps
- **With 270 apps supporting it and so many music repositories supporting it, MusicXML seems unlikely to go anywhere**
- **The quickest route to obsolescence is to think "we're done"**
  - That's why MNX may serve as a disruptor from inside the community
  - Try to avoid the pain of MIDI 1.0 to MIDI 2.0, HTML 4 to HTML 5, Python 2 to Python 3…

Thank You