



W3C TPAC Music Notation Community Group Meeting

28 October 2021



Agenda

- Introduction to the Music Notation Community Group
- Progress update since last meeting in April 2020
 - MusicXML 4.0
 - SMuFL 1.4
 - Documentation system for MNX and MusicXML
- MNX specification update
- New possibilities for Community Group work
- Planning for in-person meetings
- Questions and answers



W3C Music Notation Community Group

- Founded in July 2015
- Develops and maintains format and language specifications for notated music used by web, mobile, and desktop applications
- Maintains and updates the MusicXML and Standard Music Font Layout (SMuFL) specifications
- Developing new MNX spec to handle new use cases and technologies
- Now the 8th largest W3C Community Group and 14th most active
- <https://www.w3.org/community/music-notation/>



The most popular music notation software standards

- MusicXML
 - Supported by over 260 web, mobile, and desktop applications
 - Notation editors, digital audio workstations, scanning, practice, analysis, ...
 - MusicXML 4.0 released in June 2021
- SMuFL
 - Supported by most major notation editors including Dorico, MuseScore, Finale, capella, and Soundslice
 - Also supported by Logic Pro X, MaxScore for MAX/MSP, and Verovio
 - Most new music notation fonts are now SMuFL compliant
 - SMuFL 1.4 released in March 2021



MusicXML app support as of 27 October 2021

Products Shipping Now			Beta/Prototype Software		
abc2xml / xml2abc	Kinetic	Pizzicato	A-Score Music Composer		
Band-in-a-Box	Komp	PowerTracks Pro Audio	Audimus Notes		
BMML	libMusicXML	Primus	bach		
Braille Music Editor	Lime	ProxyMusic	Canorus		
Calligra Suite	Logic Pro	QuickScore Elite Level II	CMME Editor		
capella	MagicScore	REAPER	Haskell Library		
Cornelius Composer	MaxScore	Reflow	Humdrum		
Cubase	Melody Scanner	SCORE	KGuitar		
DeluxeNote	Mozart	ScoreMaker	Morpheus		
Denemo	MusEdit	Score Writer	Music Jotter		
Dorico	MuseScore	Scorio	MusicSQL		
Electric Pipes	Music Notation SDK	Sibelius	Nightingale Notelist		
Encore	Music Processing Suite	SmartMusic	Ossia Viewer		
Ensemble Composer	music21	SmartScore	Partitura		
Fandango	NotateMe	Songs2See Editor	PartRenamer		
Finale	Notation Composer	Soundslice	Ptolemaic		
Flat	NoteAbility Pro	Speech Analyzer	pyScore		
Forte	Noteflight	StaffPad	RenoID Player		
Free Clef	Noteworthy Composer	Stave'n'Tabs	sol2snd		
Guitar Pro	Notion	Symphony Pro	Zong! Editor		
Harmony Assistant	Nuendo	TaBazar			
Hyena	Oktiv Octava	TabEdit			
JapoScore	OpenMusic	tonica fugata			
JFugue	Overture	VocalEasel			
			abc4j		
AnthemScore	Impro-Visor	Plaine and Easie	Amadeus		
AudioScore Ultimate	iReal Pro	PlayScore	Braille Music Compiler		
audite PLUS	JMSL	Rachmaniac Score	BUZZle		
Audiveris	Jniz	ReadScoreLib	FOMUS		
Audovia	Kunkunshi Editor	Rosegarden	GBMusicParser		
Cadencii	Ludwig	Rousseau	jChing		
Cakewalk	muDic	Samplitude	mercussion		
capella audio2score	muscript	ScanScore	mingus		
capella-scan	MusicJOT	ScoreCloud	PHPMusicTools		
capella wave kit	Music-to-XML	Score Creator	Power Tab		
Cavatina	MyScript Music SDK	Sequoia	PWGL		
CelticPipes	Notate	SharpEye	SCAMP		
Crescendo	Notation Pad	Sheet Music Scanner	Sing2Notes		
Digital Performer	Opusmodus	Simple Song Creator			
Drumline Composer	PDFtoMusic Pro	SmartScore NoteReader			
Frettable	PhotoScore Ultimate	Synfire			
HarmonyWiz	Piano2Notes	Touch Notation			
		TuxGuitar			
Archivarius 3000	KlavarScript	Practice Bird	Accento		
Auto-Tune EFX+	Kooplet	PracticeFirst	Antescofo		
Blackbinder	LilyPond	Practice Player Live Midi	Arduino		
BrailleMUSE	Magenta	Purely Musical	CrestMuse Toolkit		
Canon	Manufaktura Controls	Real Piano Score	FreeDots		
capella playAlong	Match My Sound	Rhythm Lab	GLozart		
capella reader	MDLscore	Scroller	GStreamer		
capella start	Melody Assistant	SeeScore	HTML5 Guitar Tab Player		
CsoundAC	Melody Player	Sight Singing Studio	KotoViewer		
D3 Pianogram	MidiAndMusicXmlPlayer	SingAccord	MATLAB		
daCapo	MIDI Player Pro	Singscope	MoonPiano		
Don's MusicXML Viewer	MuseCloud	Sinsky	μO		
Doo Bee Doo Composer	MusicEase	SM Music Reader	MuseBook Score		
EarMaster	Music Prodigy	Songistic	Musicista		
EasyABC	musicxml2mid	Songs2See Game	Music Score Metadata Builder		
Expresseur	MusicXML to MP3	Soundslice Viewer	MusicXML Analyzer		
Forte Reader	Myriad QuickLook	TEFpad	musicxml2words		
Frescobaldi	Newzik	TEFview	Neutrino		
GOODFEEL	OpenSheetMusicDisplay	OrganMuse	Noutee		
GUIDO	PhonicScore	PhonicScore	Opus		
Harmonia	PianoLudic	Piano Marvel	PSAM Control Library		
IBOS Nodelærer	PMX / MusiXTeX	PMX / MusiXTeX	Rocksmith+		
INScore			ShakuViewer		
Jellynote			SolFaSoGood		
			StringyFi		
			SuperScore		
			Talking Scores		
			Zong! Player		

M
u
s
i
c
X
M
L



MusicXML 4.0

- Released in June 2021
- Major new features since 3.1
 - Concert scores with transposed parts
 - A standard way to combine score and parts in a single .mxl file
 - System-level directions
 - Score following, assessment, and other machine listening applications
 - Swing playback
 - Roman numerals and Nashville numbers
 - XML Catalogs
 - Complete documentation on the W3C site with examples of every element



MusicXML 4.1

- No current plans for a version 4.1
- We likely would not start this until MNX 1.0 is completed
- Please create a GitHub issue for any suggestions for a future version



SMuFL 1.4

- Released in March 2021
- Added more than 150 new glyphs
 - New ranges for scale degrees and techniques noteheads
 - Supplemental ranges for fingering, figured bass, note name noteheads, accidentals
- Enriched font-specific metadata
 - Specification of preferred text fonts
 - Additional engraving defaults for barline separation, multi-bar rest H-bars, etc.



SMuFL 1.5

- Not yet actively being worked on
- Ideas for development topics welcome
 - Please raise an issue on GitHub to start the conversation
- Focus likely to be on font-specific metadata, to enrich data about optional glyphs
 - Define categories of optional glyphs (e.g. optical variants at different sizes, historical period, etc.)
 - Allow grouping of optional glyphs into sets (cf. OpenType stylistic sets)
 - Make it easier for applications to use (or present to the user) optional glyphs as alternates



Documentation Generator

- An open source, database-driven web app using Django
- Manage the documentation within the app, making things editable and viewable on a local web server
- At checkpoints, freeze the database contents into a JSON file, and generate a static HTML site via a script
- Used for both the ongoing work on MNX and for MusicXML 4.0
- Never could have done the complete MusicXML 4.0 documentation without it
- <https://github.com/w3c/mnx/tree/master/docgenerator>



Introduction

Tutorial

"Hello World"

File Structure

MIDI-Compatible Part

Notation Basics

Chord Symbols

Tablature

Percussion

Compressed .MXL Files

Code Generation

MusicXML Reference

Elements

Data Types

Examples

Container Reference

Elements

MusicXML 4.0 > MusicXML reference > Examples > <grace>

<grace>



```
<measure number="54">
  <note default-x="13">
    <grace slash="yes"/>
    <pitch>
      <step>B</step>
      <octave>4</octave>
    </pitch>
    <voice>1</voice>
    <type>eighth</type>
    <stem default-y="3">up</stem>
    <notations>
      <slur number="1" placement="above" type="start"/>
    </notations>
  </note>
  <note default-x="31">
    <pitch>
      <step>A</step>
      <octave>4</octave>
    </pitch>
    <duration>8</duration>
    <voice>1</voice>
    <type>quarter</type>
    <stem default-y="10">up</stem>
    <notations>
      <slur number="1" type="stop"/>
    </notations>
  </note>
</measure>
```

MNX 1.0 draft specification



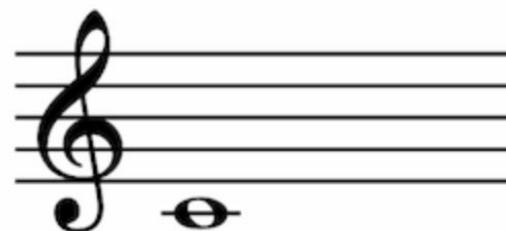
- Editor: [Adrian Holovaty \(Soundslice\)](#)
- Original editor: Joe Berkovitz (Risible LLC)

MNX is a new, open standard for representing music notation as machine-readable data. The aim is to improve [MusicXML](#) in fundamental ways while retaining many of its key concepts, terms and features. A primary goal is to provide a high degree of interoperability and exchange between different applications working with music notation.

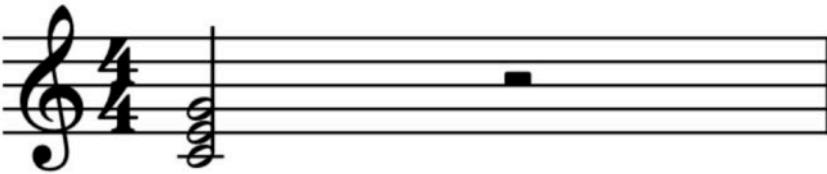
MNX is a work in progress, and it's **not ready for implementation**. But if you work with music notation data in any way, we'd love your input.

A gentle introduction

Conceptually, MNX is a way to represent this image...



Three-note chord and half rest



In MusicXML, notes within chords contain `<chord>` (except for the first note in the chord). MNX uses `<event>` to group notes within chords.

Show: Relevant section Full document

MusicXML

```
...  
<note>  
  <pitch>  
    <step>C</step>  
    <octave>4</octave>  
  </pitch>  
  <duration>2</duration>  
  <type>half</type>  
</note>  
<note>  
  <chord></chord>  
  <pitch>  
    <step>E</step>  
    <octave>4</octave>  
  </pitch>  
  <duration>2</duration>  
  <type>half</type>  
</note>
```

MNX

```
...  
<event value="/2">  
  <note pitch="C4"/>  
  <note pitch="E4"/>  
  <note pitch="G4"/>  
</event>  
<event value="/2">  
  <rest/>  
</event>  
...
```

1 2 3

Flutes

1 2

Oboes

1 2

Piano

This musical score consists of three systems of music. The first system features Flutes (3 parts) and Oboes (2 parts) in 3/4 time. The second system features the same instruments. The third system features the Flutes and Oboes in 3/4 time, while the Piano is in 2/4 time. The piano part includes bass clef and a bass staff.

1

Fl. 2

3

Ob. 1 2

Piano

This musical score consists of four systems of music. The first system features Flutes (3 parts) and Oboes (2 parts) in 3/4 time. The second system features the same instruments. The third system features the Flutes and Oboes in 3/4 time, while the Piano is in 2/4 time. The piano part includes bass clef and a bass staff.

Milestone



Set milestone

Filter milestones

Open Closed

Clear this milestone

V1

No due date

Uncommitted

No due date

V Next

No due date

4 participants



MNX next steps

- Finalize list of V1 features, using V1 GitHub milestone
- Implement each remaining V1 feature with an example-first approach
- Continue implementing mnxconverter open-source utility



MNX: how you can help

- Follow the MNX GitHub issue tracker
- Post your opinion, even if it's just a thumbs up
- Follow the community group blog and mailing list



New possibilities for Community Group work

- Instrument data
- Non-Western music notation, including traditional Asian music notations
- Updating the group charter



Instrument data: problem

- Most music notation applications have data about instruments
 - Basic data, e.g. default clef(s), number of staves, transposition, playable ranges
 - Specific idiomatic data, e.g. tunings for stringed instruments, number of frets for fretted instruments, sounds/playing techniques that can be produced
- This data is not secret, but it can be time-consuming to research and encode
 - Every developer does this research from scratch every time
 - Errors and omissions are different in every data set
 - Massive duplication of effort



Instrument data: solution?

- Music Notation CG could start a new specification for instrument data
- Develop one or more simple data formats (JSON? XML?) to allow easy conversion or consumption of this data by notation applications
- Software developers could contribute their existing data
- Identify and appoint expert volunteers to help advise on errors and omissions, or to arbitrate on disagreements between different sources



Instrument data: value?

- Do group members agree this is an area of need?
- Is there sufficient value to be had in devoting attention to this?
- Would group members who have built data sets actually be willing to contribute data?



Non-Western music notation

- The Music Notation Community Group is not restricted by charter to Western music notation
- However, all current projects focus on Western music notation
- What about non-Western notation, e.g. traditional Asian music notations?
- We would need experts in this area to guide any such projects
 - Not just experts in the notation, but in software that uses these notations
 - Which means there would need to be more software for these notations



Updating the group charter

- If we want to create any new deliverables besides MNX, MusicXML, and SMuFL, we need to add them to the charter
- The charter description is outdated anyway
 - Many of the deliverables happened years ago with new versions afterwards
 - The MNX description refers to the original design that we changed last year
- Is it time to update the charter to clarify our current work, even if we do not add any new projects?



Planning for in-person meetings

- We plan to continue online meetings for their greater inclusiveness
- But we want to resume in-person meetings too
- Here are some of the upcoming events next year:
 - Musikmesse in Frankfurt: 29 April – 1 May 2022 (starts Friday)
 - The NAMM Show in Anaheim: 3 – 5 June 2022 (starts Friday)
 - TENOR conference in Marseille: 9 – 11 May 2022 (starts Monday)
 - Music Encoding Conference in Halifax: 19 – 22 May 2022 (starts Thursday)



Questions and answers

- Any questions for the Community Group?



Staying in touch

- If you are not already a member, join the Music Notation Community Group
 - Membership is free of charge
 - Sign up at the [Community Group home page](#)
 - Click on the JOIN OR LEAVE THIS GROUP button to get started
 - If you work for an organization in the area of music notation, please join as a representative of that organization
- Technical work largely happens in our GitHub repositories for [MNX](#), the [MNX Converter](#), [MusicXML](#), and [SMuFL](#).
- We post co-chair meeting minutes on the blog, usually every 2 weeks