

DAISY Music Braille Project:

Requirements Capture for Music Braille Conversion Tools

11 February 2019, Sarah Morley Wilkins

# Contents

Introduction 2

Vision 2

Mission 2

Out of scope 2

This document 3

Description of intended user 3

Prioritisation 3

How to respond by the deadline: Friday 15 March 4

Reference rule books 4

1. Accessibility and Usability 5

2. File handling 5

3. Formatting/Layouts 7

3a. Layout descriptions (as reference) 7

4. Country Codes 8

5. Options 8

6. Conversion of musical symbols 9

7. Output 14

8. System Requirements 15

9. Business Arrangements 15

10. Note about Funding 16

11. Anything else? 16

Thank you! 16

# Introduction

## Vision

Musicians who read braille enjoy timely and affordable access to increased numbers of accurate music braille scores in hard-copy and digital formats produced by effective and reliable conversion tools.

## Mission

Through collaboration, we wish to achieve greater automation for music braille production, primarily in production houses and in education. Where possible we would also like blind musicians themselves to be able to obtain and convert files for their own use.

We also wish to secure greater opportunity for creating higher-quality MusicXML (and later, MNX) files better suited for conversion.

Through these improvements, we intend to enable greater international file sharing, increasing the speed agencies can deliver music braille to musicians, and reducing the duplication of effort across the sector.

We wish to support the development of a sustainable software tool, or tools, which enable rapid and accurate conversion primarily of digital music scores in MusicXML format, into music braille scores.

The tool(s) will primarily produce scores suitable for embossing. If the resulting file can also be displayed on a refreshable braille display that would be beneficial.

The tool(s) will be supported by a robust business plan with a financial plan in place to ensure their sustainability (e.g. for at least 5 years).

## Out of scope

The project is not (at least for the moment) concerned with tools which enable blind musicians to write, read, listen to, and edit their own scores, and producing output for sighted musicians. These are essential tools for blind musicians, but this project cannot focus on this at this stage.

## This document

This Requirements document tries to capture the high-level functionality requirements raised through the two international surveys and two Round Table Meetings conducted by the DAISY Music Braille Project.

These are issues which agencies, transcribers, and end-users have reported that a tool should do in future, to make the production of embossed music braille accurate, efficient, and reliable so they can get more scores out to more blind musicians. These improvements should also mean that producers can share files effectively.

These requirements also include common issues identified through testing of various conversion tools during 2018 by the UK team, identifying bugs and fixes required. These have already been reported to developers, and some may not yet have been implemented.

## Description of intended user

It is expected that the user of this tool(s) would be familiar with music and the basic terminology and features of music braille used by blind end-users. They may also be proficient music braille transcribers. The users may be sighted, or low-vision or blind using screen-access technology with the tool.

## Prioritisation

You are being invited, as a sector specialist, to review and prioritise these Requirements, and add any comments you wish to make.

Bear in mind that it will not be possible to have everything on this list, and some requirements may not even be possible, so please prioritise your requirements carefully so the essential things become obvious.

Please score your priority against each requirement as:

1 = Essential

2 = Desirable

3 = Nice to Have

4 = Not Needed

0 = No Opinion.

You do not need to add comments unless you wish to. We may use any comments you give as evidence to support the prioritisation of these Requirements.

We will review all responses, and seek to find the majority view, if not consensus. Responses from DAISY members may be weighted if we need a way to make final decisions.

The resulting prioritised Requirements will then be shared with Developers, seeking their comments, and estimates for pricing and timeframe for implementation.

## How to respond

### Deadline:

Friday 15 March

### Submit online responses at:

<https://www.surveymonkey.co.uk/r/MusicBrailleRequirements>

We believe this will be accessible for the majority of you. If for any reason you cannot complete the online survey, please get in touch to arrange an alternative method: [musicbraille@daisy.org](mailto:musicbraille@daisy.org)

## Reference rule books

Throughout the document, reference is made to the two main rule books:

* NIM: New International Manual of Braille Music Notation
* MBC: Music Braille Code, Braille Authority of North America 2015

# 1. Accessibility and Usability

1. Sighted users rate it as easy to learn and to use.  
   Why: A tool is easier to learn and more efficient to use by sighted people when it is similar to other modern tools.
2. Blind people rate it as easy to use with at least one of the main screen-access solutions (e.g. Jaws, NVDA, SuperNova).  
   Why: Blind transcribers and blind end-users need to use the tool to convert materials.
3. The tool can be translated into other languages.  
    Why: To support users in different countries.
4. When navigating through the digital score, it supports: Select, Go To, and Browse Next-Previous, by: note, voice, stave, system, print line, and braille line.  
   Why: Users need to be able to quickly navigate through the score.

# 2. File handling

1. It can handle large files up to 100 MB.  
   Why: Some scores are large and complex, or one book may contain a large number of scores.
2. It accepts the latest MusicXML or MNX input files, as well as older versions.  
   Why: Keeps up to date with industry standards, so digital files can be converted without requiring scanning from the paper score.
3. It can handle the full range of stave notation imports: single line, vocal score (including more than one vocal line), keyboard, score, lead sheet, continuo part, all standard instrument-specific articulation and other techniques.   
   Why: Musicians need such scores for study and performance.
4. It can accept a MusicXML file with an infinite number of parts, and the user should be able to select from any of those parts.  
   Why: To accommodate all kinds of scores, and user needs.
5. It validates MusicXML/MNX input and provides automated or proposed corrections.   
   Why: To check the file is good enough to get a reliable conversion whilst minimising human intervention for common issues.
6. It is future-proof by allowing for the addition of agreed DAISY-defined new signs for special notation.  
   Why: DAISY may wish to add a sign used in their specification.
7. It accepts SMuFL specifications ([www.smufl.org](http://www.smufl.org)) in the MusicXML file.  
   Why: SMUFL-compliant fonts (such as Bravura) can be used to display music notation, or access SMUFL glyph names. New symbols can be defined, and can incorporate updates to MusicXML and MNX. SMuFL is the next-generation music font standard, and most of the notation software (now Finale, Dorico and MuseScore) will use it in the future. MusicXML 3.1 and MNX also make use of such a standard.
8. It allows users to specify the order of pieces in a volume, or over multiple volumes.   
   Why: agencies need to produce large books of many pieces in a particular order, or a large work, which is often large enough to require multiple volumes.
9. It can split a large file into several segments; and combine segments into a complete file/full score.  
   Why: Multiple transcribers need to be able to collaborate to complete one project and these need to be collated when complete.
10. Scores can be given encrypted copy protection.  
    Why: To protect publisher’s materials from being used for back-translation to print scores.
11. It can save different revisions for comparison, or rolling back.  
    Why: Users may wish to compare scores for study or performance, or revert to an older version of the score they are preparing.

# 3. Formatting/Layouts

The tool should produce the most common layout formats (each described below), with default settings in place, together with options for customisation:

3.1 Single Line

3.2 Bar-Over-Bar, (with option for Line-Over Line)

3.3 Line-By-Line (Open Score)

3.4 Section-by-Section (with option for Stave-By-Stave)

## 3a. Layout descriptions (as reference)

* Single Line - There are some variations on how this is transcribed. E.g. in the UK, the print line and bar-number is written in cell 3 with the music starting on the next line in the first cell.
* Bar-Over-Bar - All lines within a parallel are aligned vertically at the beginning of bars. This format is used for keyboard music, chamber music, orchestral scores as well as other instances where a parallel disposition is required. Bar-numbers are transcribed in the left-hand margin for keyboard music, but appear above the music when chamber music and orchestral scores are brailled.
* Line-Over-Line - This is a variation of Bar-Over-Bar. Bars within a parallel are not vertically aligned. It is used far less often in the UK.
* Line-By-Line (Open Score) - Used for words and voice of solo and choral music where separate parts are produced. E.g. in the UK, words are written beginning at the start of the line, the corresponding music is indented starting in cell three. Other countries adopt a different approach.
* Section-By-Section (Continental Style) - Each paragraph has a hand or part sign. A composition is divided into sections by its phrasing. This is not used in the UK but it is used in other countries.
* Stave-By-Stave - Paragraph lengths are determined by the length of print staves. It is a variation of section-by-section, which will base the sections by print systems.
* Bar-By-Bar - The work is divided into sections, each of which is a paragraph with the lowest part written first, intervals read upwards. The paragraphs are numbered or identified with print page and line, as appropriate. There is a variant of this where the right-hand comes first with intervals reading downwards. E.g. this was used in the UK for many years until Bar-Over-Bar was adopted around 60 years ago. This format is not really used, but might be of benefit for single line braille displays.

# 4. Country Codes

1. It can produce at least the main country codes: e.g. UK, USA, European *[to be specified].* Why: Minimises the options available, making it easier to share files, and to build/maintain the tools.
2. It can handle multiple languages in one file.  
   Why: So text can be output in any language which has a braille table.

# 5. Options

1. The tool permits options to show/hide features in particular layouts to suit different user needs, e.g. for learners, or those learning the score in a particular way.  
   Why: Agencies and end-users need settings to produce for different end-user purposes, or for different countries. Proposing default sets reduces tool complexity and variability of scores, but may require some compromises.

Options should include:

5.1.1 Fingering (on/off)

5.1.2 Print line numbers (on/off )

5.1.3 Position of print page number (specified)

5.1.4 Position of braille page number (specified)

5.1.5 Line numbers (on/off)

5.1.6 Bar numbers (on/off)

5.1.7 Bar number location (side/top)

5.1.8 Print page numbers (on/off)

5.1.9 Define interval direction (specified)

5.1.10 Indentation (specified)

5.1.11 Hide empty staves (on/off)

5.1.12 Paper size (specified)

5.1.13 Doubling convention (specified)

5.1.14 Repeat handling (specified)

5.1.15 Transpose a single part/voice (specified)

5.1.16 Rehearsal letters (on/off)

5.1.17 Position of rehearsal letters (specified)

5.1.18 Octave signs at the start of each line (on/off)

5.1.19 Lyrics (contracted/uncontracted)

5.1.20 Braille page number (on/off)

5.1.21 Running header (on/off)

5.1.22 Other (please specify):

1. User profiles (configurations and definitions) can be saved for users with different requirements.  
   Why: Agencies can save settings when producing for multiple customers; educators can do this when producing for multiple students.
2. It can provide dynamic on-screen visual and braille music notation.  
   Why: To easily make corrections viewing either notation, where modifying one updates the other.
3. When editing, it can automatically move objects in a braille staff or system, like modifying texts in MS Word or Duxbury Braille Translator. For big changes to a system or section, a reformatting operation can simply update the modified score to the correct layout.  
   Why: Ease the editing process, avoiding large number of manual adjustments after editing or correction.
4. It can transcribe according to specific instrument types, so that the same symbol can be transcribed to appropriate braille equivalents.   
   Why: So a user gets the right annotation for the parts (e.g., the plus sign in horn scores is a stopped note, while in scores for bowed string instruments is a left hand pizzicato).

# 6. Conversion of Musical Symbols

1. Accents are correctly converted, according to NIM and MBC.  
   Why: The braille music should show the correct translation from the print music notation.
2. Supports both types of breath marks, according to NIM and MBC.  
   Why: In print vocal and wind instrumental music, signs are used for half-breath and full-breath; which both need to be translated into braille scores.
3. Chord symbols are accurately translated according to NIM and MBC.   
   Why: There are a range of signs used for this type of music, but they must be translated correctly into music braille with the various conventions being observed. Examples of errors found include:  
   A. Signs for sharps and flats being shown as S for flat and F for sharp.  
   B. To check that recognised signs for Sharp and Flat are used.  
   C. Special Parenthesis (dots 2,3,5,6) are used rather than Brackets in Unified English Braille.  
   D. To check that braille signs for circles, triangles and other signs are recognised.
4. It presents clefs accurately, according to NIM and MBC.  
   Why: Accurate clef presentation is vital. Examples of errors found include:  
   A. Tenor clef not being recognised by some software.  
   B. Dot-3 needed after tenor clef sign before Sharp.  
   C. Figure 8 following a bass clef sign does not mean the part is transposed down an octave. Advisable to check Figure 8 above does not transpose up an octave.  
   D. More flexibility with interval direction after clef signs.  
   E. Hand alternation on the same stave.
5. It converts dynamics appropriately, whether written as symbols or words, or combinations, following NIM and MBC.   
   Why: So a user gets the correct dynamic marking.
6. It converts fingering for keyboard, string instruments (right and left hands) appropriately, according to NIM and MBC.  
   Why: Fingering for keyboard and string instruments differs, conversion must follow rules for the correct instrument.
7. Markings for strings are correctly translated according to NIM and MBC.   
   Why: Accurate presentation of these kinds of details are essential. Examples of errors found includes:   
   A. Bowing signs.  
   B. String signs.  
   C. Position and fret signs.  
   D. Fingering
8. Grouping/beaming of notes is converted correctly according to NIM and MBC rules.  
   Why: These need to be observed with specific reference to unusual groupings. Example of errors identified include:   
   A. Grouping of notes when 3/8 time-signature is used.  
   B. Can the software use signs to show irregular grouping?.
9. In-accord signs are handled correctly, following NIM and MBC for various situations.  
   Why: To make it clear which are outer and inner parts. Example of errors identified include:   
   A. Outer part needs to come first, rather than inner part, then outer part.  
   B. Use of part-bar division and in-accord signs for part of a bar
10. Instrument names are correctly presented according to language, and where instrument names change within a single part, the translation is correct.   
    Why: so the user can follow the correct part. Example of errors include:  
    A. Instrument names to be defined correctly.  
    B. Instrument names in: English, French, German, Italian and Spanish.  
    C. Instrument change of names within the same part.
11. Long and short-value signs are presented, and whether a bar in 4/4 begins with a semiquaver, following NIM and MBC rules.   
    Why: Braille signs are necessary to avoid ambiguity not present in print music.
12. Vocal music conversions confirm to NIM and MBC. Examples of errors found include: presentation of multi-verse lyrics, syallabic slurs, and slurs in first, second and third languages etc.  
    Why: In braille music, the presentation of vocal music is, by its nature, quite different. As the words and music are not vertically aligned, slurs need to be added in many cases so the singer knows which syllables are sung to the required notes.
13. Unison signs are correctly used, showing two or more parts on the same note.  
    Why: In braille music there are two different situations: 1. That the sign indicating two or more parts on the same note is used; and 2. Where signs for intervals are use. It is important that the correct combination of signs is transcribed i.e. the written note, the octave sign for the specific octave, and the sign indicating an interval of an octave.
14. Tremolos are translated appropriately, as per NIM and MBC. E.g. repetition, alternation and doubling of note repetition signs.  
    Why: Print music notation has ways of displaying tremolos, but braille music requires a range of signs to convey this information. We need to be sure they work in single or doubled usage.
15. Time signatures are correctly translated, even if they are not numeric.  
    Why: In addition to numeric indication, there are also some signs conveying information, such as Common Time, Alla Breve, and the Breve sign, which should be translated.
16. Ties are presented so they are distinguishable from each other: single note, chord tie, accumulating tie/arpeggios.  
    Why: Users must be able to tell these different types of ties apart. This may be an issue for MNX.

1. It can present stem signs.   
   Why: Stems are an obvious feature of printed music notation, and should be correctly translated when users need to know what a print score contains.
2. Different kinds of slurs are presented accurately.   
   Why: As far as possible, the range and use of slurs should be catered for (may be issues for MNX). Examples of errors found include:  
   A. The software needs to be able to use both the two-cell and one-cell slur signs at the same time, and if the one-cell sign is doubled.  
   B. Overlapping slurs.  
   C. Can the software deal with different types of slurs in addition to those mentioned above?.
3. Prefixes for shaped notes are correctly translated according to the most usual symbols.  
   Why: There are many and various print elements concerning contemporary notation. Where it is important to show different types of symbols for interpretation, the most usual should be correctly translated.
4. Ornaments are correctly translated, following NIM and MBC rules.  
   Why: There are many different types of ornaments in printed music notation, and they need to be correctly translated into braille music, as illustrated in NIM and MBC.
5. Octave signs are correctly presented following NIM and MBC rules.  
   Why: Unlike print notation, braille music relies upon use of octave signs to denote pitch, and must be correctly translated. Examples of errors found include:  
   A. Doubling of octave signs.  
   B. Octave sign needed after bar with an in-accord sign(s).
6. Nuances are correctly presented.  
   Why: Print music notation must be correctly transcribed in braille. Examples of errors found include:  
   A. Nuances with interval signs.  
   B. First and second lines of continuation.  
   C. First and second-time bar numbering.
7. It can present Tablature (notation indicating instrument fingering rather than musical pitches) following the agreed specification.   
   Why: To cater for guitar or lute reading conventions.
8. It can present Chord Symbols correctly following NIM and MBC rules.  
   Why: To produce lead sheets, song books and harmony theory books. Examples of errors identified include:   
   A. Signs for sharps and flats being shown as S for flat and F for sharp.  
   B. Recognised signs for Sharp and Flat are used.  
   C. Special Parenthesis (dots 2,3,5,6) are used rather than Brackets in Unified English Braille.  
   D. Braille signs for circles, triangles and other signs are recognised.
9. It can present Figured Bass following NIM rules.  
   Why: To produce correct figured bass of Baroque scores in braille.
10. In ensemble scores, it can control part names flexibly, including defining instrument names and abbreviations, printing out list of instrument names and braille abbreviations before the music, setting instrument changes by either using those inserted in MusicXML or manually added ones.  
    Why: To make instrument labels correctly follow the original score, for clear reading.
11. It combines musical symbols in the right order.  
    Why: So it’s accurate and comprehensible to the user.
12. Multiple Bars' rest are presented in a compact manner, following NIM and MBC rules.  
    Why: These occur in parts for single-line instruments and orchestral scores, and need to be transcribed giving the information in a compact manner. In orchestral scores, to save space in the braille copy, the part can be left out.
13. It handles bars apprropriately - correct bar alignment in bar-over-bar format; offers a variable number of bars per line; identifies bar zero; and does not split bars over a page break.  
    Why: ease of reading

# 7. Output

1. Conversion generates an e.g. >80% error-free success rate when an agreed range of samples is tested *[to be specified]*Why: So greater numbers of scores are automatically converted with little or no human intervention.
2. It either: generates output which can be compiled in a braille editor as part of a production workflow; or can handle mixed text-music files entirely.  
   Why: Agencies need to be able to produce text books containing musical scores and easily integrate the two types of files.
3. It can output the full range of stave notation (single line, vocal (including multi-voice), keyboard, full score, lead sheet, continuo part, all standard instrument-specific articulation and other techniques).   
   Why: Musicians need such scores for study and performance.
4. It can extract parts from, e.g. chamber music or orchestral scores.  
   Why: Users may need to isolate specific parts when learning or assessing scores.
5. Outputs a file suitable for embossing (e.g. BRF, BRL (text tile) format).   
   Why: End-users and agencies wish to emboss the score.
6. It can reformat Bar-Over-Bar material using different braille page sizes [to be defined].[Some adaptation and compromises of some braille rules may need to be made].Why: Embossed output needs to be printable on different sized braille paper.
7. Outputs an eBraille/Text file [in BRL text format] for use on a refreshable braille display.   
   Why: More and more users are learning scores from their braille display.
8. Outputs a printable file of stave notation*.*Why: so a blind user can share a score with a sighted teacher/another musician who may be helping them with learning the score.
9. When using a publisher’s master file, it disables hard-copy printing, only displaying stave notation on-screen.  
   Why: To protect the publisher’s print score so it cannot be printed.
10. It notifies the user of conversion progress, success or failure.  
    Why: So users know how long to wait for their file, or why it failed to convert.

# 8. System Requirements

1. If installed locally, it must work with XX technology, and uses no more than 2GB RAM even for the most complex files. *[minimum specification to be defined].*Why: To run on the typical set-up of the intended agency client.
2. It must be programmed in such a way that it can be further developed.  
   Why: To ensure the tool is future-proof.

# 9. Business Arrangements

1. Annual service level agreement available to secure fixes and support *[define an acceptable response time for support questions]*   
   Why: For business support for agencies and security for long-term business practice for developers.
2. If server-based, provide built-in redundancy for server failure/recovery to minimise downtime *[define acceptable downtime]*  
   Why: To protect agencies from loss of data and unnecessary delays.
3. Offer the service for a minimum term, e.g. 5 years *[to be defined].*Why: So agencies are assured some business continuity with the tool (and their payment secures business continuity for the developers).
4. Operate a system for reporting bugs, issues and requests, and for sharing updates on implementation.  
   Why: So users can request maintenance and development of the tool in a structured way.

# 10. Note about Funding

1. The sector should aim to secure the future of a tool, or tools, which together support both agency production as well as education.  
   Why: To support production now, and to secure future users.

# 11. Anything else?

Please tell us if there is anything you would like to see in these Requirements – ideally written in a similar way please:

Issue:

Why:

Priority:

Any other comments:

# Thank you!

Thank you for sharing your expertise to help prioritise these Requirements for music braille conversion tools.

We will keep you updated with progress ahead of our next meeting in Geneva 27-28 May 2019, where we hope to discuss the prioritised Requirements with Developers, and make a plan to finance the agreed technical implementation.

If you have influence over funds in your organisation, please be ready to engage with the financing process for this development.

Thank you very much for your participation.

Sarah Morley Wilkins and Arne Kyrkjebø