MusicXML Format Preservation Assessment

Document History

Date	Version	Author(s)	Circulation
09/02/2018	1.0	Michael Day, Simon Whibley	Public
26/02/2020	1.1	Robert Lacey, Sally McInnes, Michael Day, Simon Whibley	Public

British Library Digital Preservation Team
digitalpreservation@bl.uk

National Library of Wales Legal Deposit Group

1. Introduction

This document provides a high-level, non-collection specific assessment of the MusicXML file format with regard to preservation risks and the practicalities of preserving data in the format.

This format assessment is one of a series of assessments carried out by the British Library's Digital Preservation Team. An explanation of the criteria used in this assessment is provided in italics below each heading.

This document was revised and updated by the National Library of Wales Legal Deposit Group, 16 July 2019.

1.1 Scope

This file format assessment will cover the MusicXML format as a whole, but will focus in particular on the most recently published versions of the format. MusicXML Version 3.1 was released in December 2017 and includes both a Document Type Definition (DTD) and an XML Schema Definition (XSD) [1]. Since July 2015, MusicXML has become an activity of the W3C (World Wide Web Consortium) Music Notation Community Group [2].

As MusicXML is based on the Extensible Markup Language (XML), this assessment should be read in conjunction with the British Library's file format assessment of XML [3].

Please note that this assessment considers format issues only, and does not explore other factors essential to a preservation planning exercise, e.g. collection specific characteristics that need to be considered prior to the implementation of preservation actions.

1.2 Summary

MusicXML is an XML-based format for the representation of western musical notation. MusicXML is defined in MakeMusic's introduction to the Version 3.1 DTD as the standard open format for exchange of digital sheet music and further [4]:

> *MusicXML was designed from the ground up for sharing sheet music files between applications, and for archiving sheet music files for use in the future. You can count on MusicXML files being readable and usable by a wide range of music notation applications, now and in the future. MusicXML complements the native file formats used by Finale and other programs, which are designed for rapid, interactive use.*
>
> *Just as MP3 files have become synonymous with sharing recorded music, MusicXML files have become the standard for sharing interactive sheet music. With MusicXML you can create music in one program and share your results – back and forth – with people using other programs. Today more than 240 applications include MusicXML support.*

The MusicXML format was, therefore, primarily designed to support the exchange of digital sheet music between the many applications used to create and adapt music notation, e.g. score writer programs like Sibelius and Finale (which use their own proprietary formats for representing music notation). As a widely-accepted exchange format, MusicXML is also in a strong position to establish itself as a format for the distribution of music scores as well as (potentially) as a format suitable for longer-term preservation.

As its name implies, MusicXML uses the Extensible Markup Language (XML), a text-based format developed and maintained by the World Wide Web Consortium (W3C) [5]. As a text-based format, XML-based documents are designed to be human readable, although individual files are often not easy to be understood without some kind of software support [3 p. 3].

MusicXML is a proprietary format, but all documentation (XML DTDs, XML Schemas, etc.) have been made freely available through the MakeMusic website [4] and the format itself can be used under a Public License.

Development of the MusicXML format is now led by the W3C Music Notation Community Group [2], currently (December 2019) chaired by Michael Good, Daniel Spreadbury, and Adrian Holovaty.

2. Assessment

2.1 Development Status

A summary of the development history of the format and an indication of its current status

The MusicXML format was originally developed by Michael Good of Recordare LLC at the end of the 1990s. It was recognised at that time that the emergence of XML provided a good opportunity to create a standard language for music notation interchange [6]. Other XML-based formats for music notation that were in development at that time included the Music Markup Language (MML) [7] and the Music Encoding Initiative (MEI) [8]. While some of these other musical notation formats continue to be developed and used -- e.g., MEI has carved out a useful role for itself within the music research and scholarly music editing communities[1] -- Michael Good has noted that only MusicXML had been widely taken up by commercial music applications [6]. By the end of the 2000s, therefore, MusicXML had become a *de facto* standard for the exchange of music notation.

MusicXML was originally built upon two formats developed for by music researchers, MuseData and Humdrum. The version history followed the following timeline:

- Version 1.0 (January 2004); Version 1.1 (May 2005)
- Version 2.0 (June 2007), supported compressed MusicXML files that can contain images, audio files, and multimedia; Version 2.0 update (July 2007)
- Version 2.0 XSD (September 2008), XML Schema Definition available for partwise, timewise, and opus documents
- Version 3.0 (August 2011),
- Version 3.1 (December 2017), the current version of the format

Selected assets of Recordare LLC (including MusicXML and the Dolet software) were taken over by MakeMusic, Inc. in November 2011. MakeMusic is also the publisher of the popular Finale music notation software; its acquisition of MusicXML was intended, in part, to support better integration between different versions of Finale as well as to improve the MusicXML export from Finale for the SmartMusic education platform [9].

In response to concerns expressed by the MusicXML community, further development of MusicXML moved in July 2015 to the W3C Music Notation Community Group, where development has also been aligned with the Standard Music Font Layout (SMuFL) standard [2] [9]. W3C Community and Business Groups are open for any interested parties to join and have been described as a means to give "developers, designers, and anyone passionate about the Web a place to hold discussions and publish ideas" [10]. Community Groups are not W3C Working Groups -- i.e. those responsible for developing official W3C/IETF standards like CSS, HTML, MathML, RDF, XHTML, XML, etc. -- but are intended to be places "where Web developers and other stakeholders develop specifications, hold discussions, develop test suites, and connect with W3C's international community of Web experts" [10].

The current version of MusicXML is Version 3.1 issued in December 2017 and is available through the GitHub repository. It is licenced under the W3C Community Final Specification Agreement (FSA) [11]. According to the MakeMusic site, it resolved 80 issues entered in the MusicXML GitHub repository [12]. MusicXML 3.1 is compatible with earlier versions of MusicXML. Valid MusicXML 1.0, 1.1, 2.0, and 3.0 files are also valid MusicXML 3.1 files [1].

The main MusicXML web site, which includes official format documentation (including DTDs and XSDs) is maintained by MakeMusic [4].

2.2 Adoption and Usage

An impression of how widely used the file format is, with reference to use in other memory organisations and their practical experiences of working with the format

There is little evidence so far that memory institutions have been collecting large numbers of MusicXML files. It is, however, used by music publishers, composers, and researchers, e.g. for the production of digital critical editions of music scores, and it is used for the online distribution of completed scores.

The Library of Congress *Recommended Formats Statement, 2017-2019* for digital musical compositions (score based representations) supports the use of MusicXML. Amongst its "recommended" formats, the Library of Congress guidance expresses a preference for XML representations of musical notation over page-layout formats like PDF [13]. For musical scores, the guidance specifically recommends XML formats that include "accessible DTD/schema, XSD/XSL presentation stylesheet(s), and explicitly stated character encoding" (the examples of these it provides are MusicXML and MEI). Other musical composition formats are only deemed "acceptable," e.g. XML formats that use proprietary DTDs or schemas, or the native formats generated by proprietary music notation software.

[1] MEI continues to exist as an open-source project, hosted by the Akademie der Wissenschaften und der Literatur in Mainz.

2.3 Software Support

2.3.1 Rendering Software Support

An overall impression of software support for rendering the format with reference to: typical desktop software; and current support on British Library reading room PCs

Being based on XML, MusicXML code would be able to be viewed and edited by a wide-range of generic tools, including text editors or dedicated XML editors [3]. In addition, most of the current generation of Web browsers would support the enhanced viewing of the XML code, including indentation of tags and colour-coding. However, as the British Library's generic assessment of XML has noted, "support for rendering, processing or otherwise working with specific schema defined XML formats depends on software support, the function of the data and the nature of the intended processing" [3 p. 3]. In the case of MusicXML, this would mean using rendering tools that have been specifically designed to be able to open and render MusicXML files. In practice, these will typically be music notation (scorewriter) programs.

In July 2019, the MusicXML website recorded 240 software applications that supported MusicXML -- support being defined as apps that have "at least some MusicXML interchange capability" [14] [15]. The number of applications supporting MusicXML seems to be still gradually increasing; the list on the MusicXML website currently (July 2019) stands at 245 [15]. The list includes many specialist apps, but seems to be dominated by music notation programs, including MakeMusic's Finale [16], Avid's Sibelius [17], Passport Music's Encore [18], PreSonus' Notion [19], the LilyPond music engraving program [20], and the open source MuseScore [21].

Music notation software will typically produce scores in their own formats. For example, Sibelius files will typically be generated and saved using its own proprietary format, which has the extension: `.sib`. However, most of the major music notation programs have now implemented the capability to convert and export their own proprietary formats to-and-from MusicXML -- in the way that most would also be able to generate PDF or MIDI files. In the context of music notation programs, MusicXML is typically used as an exchange format. It does not mean that all features available in the original file (when operating in its original environment) will be represented when converted to MusicXML-- the format would merely provide a means for interchange. Furthermore, it is unlikely that any single music notation software application would be able to support all of the features available in the full MusicXML standard.

The existence of MusicXML as an exchange format has also made it attractive as a means of distributing music score content. For example, the MusicXML webpages list a number of services that enable the download of sheet music in MusicXML or compatible formats [22]. One of the largest of these is the membership (subscription) site Musicalion [23], which has over 37,000 musical compositions available for download, with over 19,000 of these available in MusicXML (alternative formats include PDF, MIDI, and original source formats). MuseScore.com [21] provides access to downloadable free scores in a number of formats, including MusicXML, MuseScore (its own format), PDF, MIDI, and MP3.

2.3.2 Preservation Software Support

An impression of the availability and effectiveness of software for managing and preserving instances of the file format

Format identification

The British Library's preservation assessment of XML noted that text-based formats like XML can cause problems for format identification tools, especially if files contain no XML declaration [3].

Standard MusicXML files have the standard XML extension: `.xml`; compressed MusicXML files have the extension: `.mxl`). DROID (after version 6.3) and other PRONOM-based tools such as Siegfried and FIDO are able to identify files as MusicXML (PUID: fmt/896) [24]. The TrID file identifier software also correctly identifies the MusicXML format [25]. However, none of the identification tools tested were able to determine specific versions.

Xena (current version 6.1.0, 31 July 2013) [26] and JHOVE (current version 1.16, 16 March 2017) [27] identify MusicXML simply as XML. If the XML module is used in JHOVE they show as not well-formed.

Validation, Conformance Checking and Detecting Preservation Risks

The British Library's preservation assessment of XML provides a quick definition of what the validation of XML generally means:

> *XML validation is the process of checking a document is both valid and well-formed* [28]. *The notion of 'valid' and 'well-formed' XML is expressed through two main kinds of constraint documented in the XML specification. These are summarised informally as: "the well-formedness constraints are those imposed by the definition of XML itself (such as the rules for the use of the < and > characters and the rules for proper nesting of elements), while validity constraints are the further constraints on document structure provided by a particular DTD"* [29].

A number of generic XML tools provide the ability to check the validity of XML files. These test whether files follow the specific syntax of an XML file (i.e. are 'well-formed') and whether they adhere to the structure defined by a particular XML DTD or Schema [28] [30].

Since Version 2.0 (2008), MusicXML has been made available as both a XML DTD and a XML Schema Definition (XSD). When MusicXML was first being developed, the only official way to define an XML format was through a DTD [31]. Versions 1.0 and 1.1 of the format were therefore released as XML DTDs. For Version 2.0, however, the developers had recognised the additional benefits that could be achieved through the use of XSDs, given that their use by then had been established and new XML tools had become available [31]. Part of the justification for using XSD for MusicXML was the ability to improve validation. The MusicXML documentation notes that XSDs are able to provide a stricter definition of the format than DTDs, adding that, "MusicXML semantics that were previously only captured in the documentation are now enforced during schema validation" [32] The MusicXML FAQ notes that the format's evolution from an interchange format to a distribution format made it important to be able to support the best-possible tools for automated quality assurance, noting that "validating against an XSD can catch many more errors in XML document creation than validating against a DTD" [31].

> *The XSD puts much more of MusicXML's semantics into the language definition itself, rather than just in the documentation. Thus there are many documents that validate against the DTD that do not validate against the XSD, but these reflect MusicXML file errors that you do want to catch as early and automatically as possible.*

There are also implications for the readability of the format:

> *DTDs remain more readable than XSDs for many people. To learn MusicXML, you may find it easiest to read the DTDs, the XSDs, or both in combination. This will depend on your experience and comfort level with the different technologies. The XSDs will generally offer more precise definitions than the DTDs since the format is now much more strongly typed.*

Metadata Extraction

The British Library's preservation assessment of XML noted that metadata in XML will be encapsulated in the (text-based) XML content itself [3]. Extraction would involve the parsing a DTD or schema to extract metadata content from relevant elements.

The MusicXML DTDs and XSDs include an identification section that contains basic descriptive metadata about the score. The metadata elements included are: creator, rights, encoding, source, relation, and miscellaneous. The creator, rights, source, and relation elements are broadly based on the Dublin Core metadata element set. The creator and rights elements can be qualified by a type attribute (e.g., creator attributes could include composer, lyricist, or arranger; rights attributes could specify whether they apply to the music, words or arrangement) The encoding element has additional sub elements and is intended to contain "information about who did the digital encoding, when, with what software, and in what aspects" [33]. The miscellaneous element is intended for any metadata not yet supported by MusicXML.

Metadata can apply a range of different levels, e.g. score-wide, movement-wide, and part-wide.

Migration

The British Library's preservation assessment of XML noted that XML is designed to be easily transformable from one format or standard to another, e.g. using something like XSL, the Extensible Stylesheet Language [3].

As an interchange standard, the ability to generate or read MusicXML is an integrated feature of most of the current generation of scorewriter programs, e.g. Sibelius, Finale, and Notion

A number of other tools are also able to convert content in other formats into MusicXML. For example:

- The Sibelius software has allowed MusicXML export since version 7 and, prior to that, MusicXML generation was possible using the Dolet plugin [34].

- The MuseScore scorewriter software includes a plugin which allows the ABC notation to be converted to MusicXML [35]. This uses the abc2xml utility which can also be accessed directly in a browser or via command line.
- PDFtoMusic Pro allows you to rebuild a score in PDF format and export as MusicXML [36].

Scorewriter programs would also enable imported MusicXML files to be converted into a range of other formats. For example, Sibelius 7 would support export to formats like PDF, MIDI, WAV or a Scorch Web Page.

Other tools support the conversion of MusicXML to other formats. For example:

- The LilyPond notation software is supplied with a program called musixxm121y which allows the conversion of MusicXML to a Lilypond `.ly` file [37].
- Verovio is a library for engraving MEI music scores into SVG. It supports the conversion from MusicXML to MEI in your web browser though some elements are not fully supported [38].

2.4 Documentation and Guidance

An indication of the availability of practical documentation or guidance with specific reference to the facilitation of any recommended actions

MakeMusic currently maintains the main MusicXML web site, which includes all official format documentation, including DTDs and XSDs [4].

The W3C Music Notation Community Group [2] is the focus of the on-going development of the MusicXML format; it makes versions of the format (currently Version 3.1) available through the GitHub repository [39].

2.5 Complexity

An impression of the complexity of the format with respect to the impact this is likely to have on the British Library managing or working with content in this format. What level of expertise in the format is required to have confidence in management and preservation?

As an XML-based format, MusicXML is ostensibly human-readable [3], although files are most likely in practice to be parsed and manipulated by software.

The complexity of defining musical notation in XML is such that the textual representation can be verbose and this can generate large file sizes. The MusicXML FAQ notes that large file sizes can be a problem for the distribution of digital sheet music. In order to overcome this, from version 2.0 onwards, MusicXML has defined a compressed form of MusicXML that uses the `.mxl` extension that can make files approximately 20 times smaller [31]. MXL is based on the ZIP format [40].

2.6 Embedded or Attached Content

The potential for embedding or attaching files of similar or different formats, and the likely implications of this

While XML enables the embedding of content (including binary content) using the CDATA (Character Data) encoding, MusicXML does not typically seem to be used to embed or attach content in different formats.

2.7 External Dependencies

An indication of the possibility of content external to an instance of the file format that is complimentary or even essential to the intellectual content of the instance

The British Library's preservation assessment of XML noted that externally-hosted content is easy to reference from XML documents, including links to remotely-hosted DTDs or schemas [3]. The MusicXML DTDs do frequently include such references, e.g. to the MMA MIDI DTD or to W3C standards like XLink. Over time, these external resources may be subject to deletion, change or temporary unavailability. Knight has suggested that organisations undertaking preservation should produce a local XML Catalogue to mitigate the potential loss of access to critical DTDs and schemas [41].

2.8 Legal Issues

Legal impediments to the use, management or preservation of instances of the file format

The MusicXML DTDs and XSDs are licensed by MakeMusic based on the World Wide Web Consortium's license. Provided the terms are adhered to then no payment is required. There are no patents issued or pending for the MusicXML DTDs and XSDs [31].

The full license (W3C Community Final Specification Agreement) is available from the MusicXML web pages [11].

2.9 Technical Protection Mechanisms

Encryption, Digital Rights Management and any other technical mechanisms that might restrict usage, management or preservation of instances of the file format

None known.

2.10 Other Preservation Risks

Other evidence based preservation risks, noting that many known preservation risks are format specific and do not easily fit under any of the sustainability factors above

None known.

2.11 Preservation Risk Summary

A summary of preservation risks and recommended actions (where possible).

MusicXML is a standard that is used an exchange and distribution format for western music notation. The standard uses XML and is defined in both an XML DTD and XSD (schema). As with all XML formats, MusicXML files are text-based and human readable, although a full understanding of the content would always be dependent on:

- The retention of the relevant DTDs or schemas, and preferably the format documentation
- Sgnificant knowledge of western music notation and its representations

MusicXML is a mature standard that is supported as an import/export format by a large number of music notation programs and is also used for the online distribution of sheet music. Since 2015, development of the MusicXML standard has been led by the W3C Music Notation Community Group. Documentation, including the MusicXML DTDs and schemas, is available from the MusicXML web site.

The main preservation risks include the following (these are all based on risks listed in the British Library's generic assessment of XML [3]):

- **Missing or unavailable external references**
  - In XML, DTDs and schemas are typically referenced externally; these will need to be retained in order to support processing (e.g. testing of conformance)
- **Invalid, badly-formed or insufficiently specified XML**
  - Poor quality XML could impact on transformation and long-term preservation. MusicXML tends to be generated from the formats natively used by other programs, e.g. scorewriting programs, so its quality will be dependent on how well these transformations perform
- **Interpretability and Interoperability**
  - MusicXML can generate large and complex files, which may be difficult to interpret and understand, at least without appropriate software

3. Recommendations for Action

Recommended actions in usage and handling of the format. Recommend actions in the support or development of software applications that provide, or have the potential to provide, significant risk mitigation for the format. Note that these recommendations do not take into account other requirements such as those driven by specific British Library collections, or non-preservation issues such as resourcing.

Handling Recommendations

- Investigate how generic XML tools may be able to show that MusicXML files are well-formed and valid XML, e.g. as part of an ingest workflow
- Investigate in more detail what additional validation / quality assurance steps might be necessary in order to handle MusicXML as part of an ingest workflow

Knowledge Recommendations

- Ensure that documented versions of all relevant MusicXML DTDs and schemas are preserved in an appropriate technical registry (or local XML Catalogue)
- Develop and maintain knowledge and expertise of the MusicXML format

Software Recommendations

- Collect examples of music notation software that can demonstrably import and export MusicXML files, e.g. the open source MuseScore program, Sibelius, Finale
- Conduct an assessment of programs that are able to convert MusicXML into other formats, e.g. PDF

Monitoring Recommendations

- MusicXML continues to be developed with new features being added as new versions are released; the format should, therefore, be monitored on a biennial basis with a medium priority

4. References

1. **MakeMusic.** MusicXML: For Developers. [Online] [Cited: 7 August 2017.] http://www.musicxml.com/for-developers/.
2. **World Wide Web Consortium.** W3C Music Notation Community Group. *W3C.* [Online] [Cited: 7 August 2017.] https://www.w3.org/community/music-notation/.
3. **British Library.** XML Format Preservation Assessment. *DPC Wiki.* [Online] 4 September 2015. [Cited: 2017 August 7.] wiki.dpconline.org/images/0/0e/XML_Assessment_v1.3.pdf.
4. **MakeMusic.** Welcome to MusicXML 3.0. [Online] [Cited: 9 August 2017.] http://usermanuals.musicxml.com/MusicXML/MusicXML.htm.
5. **World Wide Web Consortium.** Extensible Markup Language (XML). [Online] 11 October 2016. [Cited: 9 August 2017.] http://www.w3.org/XML/.
6. **Good, Michael D.** Using MusicXML 2.0 for music editorial applications. [book auth.] Peter Stadler and Joachim Veit. *Digital Edition zwischen Experiment und Standardisierung.* Tübingen : Max Niemeyer Verlag, 2009.
7. **Steyn, Jacques.** Music Markup Language. [Online] [Cited: 9 August 2017.] http://www.steyn.pro/mml/index.html.
8. **Music Encoding Initiative.** Music Encoding Initiative. [Online] [Cited: 9 August 2017.] http://music-encoding.org/.
9. **Good, Michael.** Five Years at MakeMusic. *Songs and Schemas blog.* [Online] 28 November 2016. [Cited: 9 August 2017.] http://michaelgood.info/2016/11/five-years-at-makemusic/.
10. **World Wide Web Consortium.** W3C Community and Business Groups. *W3C.* [Online] [Cited: 23 August 2017.] https://www.w3.org/community/.
11. —. W3C Community Final Specification Agreement. *W3C.* [Online] [Cited: 16 December 2019.] https://www.w3.org/community/about/agreements/final/.
12. **GitHub.** w3c/musicxml Issues=closed+milestone. *GitHub.* [Online] [Cited: 16 December 2019.] https://github.com/w3c/musicxml/issues?q=is%3Aclosed+milestone%3AV3.1.
13. **Library of Congress.** Recommended Formats Statement: I. Textual Works and Musical Compositions. *Library of Congress.* [Online] [Cited: 7 August 2017.] https://www.loc.gov/preservation/resources/rfs/textmus.html.

14. **Good, Michael.** MusicXML Now Supported by More Than 220 Apps. *MusicXML.* [Online] 9 September 2016. [Cited: 24 August 2017.] http://www.musicxml.com/musicxml-now-supported-220-apps/.
15. **MakeMusic.** MusicXML Software. *MusicXML.* [Online] [Cited: 23 August 2017.] http://www.musicxml.com/software/.
16. —. Finale . *MakeMusic.* [Online] [Cited: 23 August 2017.] https://www.finalemusic.com/.
17. **Avid.** Sibelius. [Online] [Cited: 23 August 2017.] http://www.avid.com/sibelius.
18. **Encore Music Notation Software.** Passport Music. *Encore Music Notation Software.* [Online] [Cited: 3 November 2017.] http://www.passportmusic.com/products/encore/.
19. **PresSonus.** PreSonus Notion. *PresSonus.* [Online] 2019. [Cited: 16 December 2019.] https://www.presonus.com/products/notion.
20. **LilyPond.** LilyPond. *LilyPond.* [Online] [Cited: 3 November 2017.] http://lilypond.org/.
21. **MuseScore.** MuseScore. *MuseScore.* [Online] [Cited: 24 August 2017.] https://musescore.com/.
22. **MakeMusic.** Music in MusicXML. *MusicXML.* [Online] [Cited: 24 August 2017.] http://www.musicxml.com/music-in-musicxml/.
23. **Musicalion.** *Musicalion.* [Online] [Cited: 24 August 2017.] http://www.musicalion.com/.
24. **National Archives.** Details for: Music XML. *PRONOM the technical registry.* [Online] 16 June 2016. [Cited: 25 August 2017.] https://www.nationalarchives.gov.uk/PRONOM/Format/proFormatSearch.aspx?status=detailReport&id=1700.
25. **Pontello, Marco.** TrID - File Identifier. *Marco Pontello's Home Page.* [Online] 24 April 2017. [Cited: 25 April 2017.] http://mark0.net/soft-trid-e.html.
26. **Sourceforge.** Xena. *Sourceforge.* [Online] 31 July 2013. [Cited: 24 October 2017.] http://xena.sourceforge.net/.
27. **Open Preservation Foundation.** JHOVE. *Open Preservation Foundation.* [Online] 16 March 2017. [Cited: 24 October 2017.] http://jhove.openpreservation.org/.
28. **Wikipedia.** XML validation. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/XML_validation.
29. **W3C.** XML Schema Part 1: Structures (Second Edition). *W3C.* [Online] 28 October 2004. [Cited: 27 April 2015.] http://www.w3.org/TR/xmlschema-1/.
30. **Wikipedia.** Well-formed document. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Well-formed_document.
31. **MakeMusic.** MusicXML FAQ. *MusicXML.* [Online] 2017. [Cited: 29 August 2017.] http://www.musicxml.com/tutorial/faq/.
32. —. MusicXML XSD. *MusicXML.* [Online] [Cited: 25 September 2017.] http://www.musicxml.com/for-developers/musicxml-xsd/.
33. —. MusicXML DRT: Identity Elements. *MusicXML.com.* [Online] [Cited: 23 October 2017.] http://www.musicxml.com/for-developers/musicxml-dtd/identity-elements/.
34. —. Dolet 6 Plugin for Sibelius 5.1 and later. *MusicXML.* [Online] March 2016. [Cited: 30 August 2017.] http://www.musicxml.com/dolet-plugin/dolet-6-plugin-for-sibelius-5-1-and-later/.
35. **MuseScore.** ABC Import. *MuseScore.* [Online] 2017. [Cited: 30 August 2017.] https://musescore.org/en/project/abc-import.
36. **Myriad Software.** PDFtoMusic Pro. *Myriad Software.* [Online] March 2017. [Cited: 30 August 2017.] http://www.myriad-online.com/en/products/pdftomusicpro.htm.
37. **Bjuhr, Peter.** Importing Music into LilyPond. *LilyPond Blog.* [Online] 16 November 2016. [Cited: 30 August 2017.] http://lilypondblog.org/2013/11/importing-music-into-lilypond/.
38. **Verovio.** MusicXML Converter. *Verovio.* [Online] 2017. [Cited: 30 August 2017.] http://www.verovio.org/musicxml.html.
39. **GitHub.** w3c/musicxml . *GitHub.* [Online] [Cited: 23 August 2017.] https://github.com/w3c/musicxml.
40. **Library of Congress.** MusicXML, Version 3. *Sustainability of Digital Formats: Planning for Library of Congress Collections.* [Online] 9 October 2012. [Cited: 23 October 2017.] http://www.loc.gov/preservation/digital/formats/fdd/fdd000358.shtml.
41. **Knight, Steve.** A short story about XML Schemas, digital preservation and format libraries. *International Journal of Digital Curation.* 2012, Vol. 7, 1.
42. **abc2xml appspot.** ABC to MusicXML converter. *abc2xml appspot.* [Online] [Cited: 30 August 2017.] http://abc2xml.appspot.com/.

43. **ibiblio.** Effective XML: Stay with XML 1.0. *ibiblio.* [Online] [Cited: 27 April 2015.] http://www.ibiblio.org/xml/books/effectivexml/chapters/03.html.

< - end - >