# *XML Format Preservation Assessment*

*Document History*

| Date | Version | Author(s) | Circulation |
|---|---|---|---|
| 27/04/2015 | 1.1 | Peter May, Paul Wheatley, Akiko Kimura | Internal |
| 01/05/2015 | 1.2 | Peter May, Maureen Pennock | External |
| | | | |

**British Library Digital Preservation Team**
digitalpreservation@bl.uk

# 1. Introduction

This document provides a high level, non-collection specific assessment of the Extensible Markup Language (XML) with regard to preservation risks and the practicalities of preserving data in this format.

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

## 1.1 Scope

The focus of this document is not strictly speaking a format but effectively a markup language – a syntax imposed on the contents of a text file that enables annotation of that document - and so differs in some ways to many of the other assessments. Specific applications of XML are defined by specific schemas, for example, the JATS (Journal Article Tag Suite) schema; each of these will be considered in their own assessment. This document should therefore be seen as a higher level assessment that addresses generic risks with preserving XML data.

Note that this assessment considers format issues only, and does not explore other factors essential to a preservation planning exercise, such as collection specific characteristics, that should always be considered before implementing preservation actions.

## 1.2 XML Summary

XML is a standard markup language [1] created by the World Wide Web Consortium (W3C), and used extensively by the library and archival community and beyond. Data formats defined by XML schemas [2] range from widely used internet standards such as RSS [3], SOAP [4] and XHTML [5], to document formats such Office Open XML [6] and JATS/NLM [7], and to metadata standards such as METS [8], MODS [9] and PREMIS [10]. It is extended from Standard Generalised Markup Language (SGML [11]) and is an open, text-based, human readable language designed to be machine interpretable also.

XML was created with the following aims [12][1]:

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

# 2. Assessment

## 2.1 Development Status

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

XML version 1.0 First Edition [12] was released as a W3C recommendation in February 1998 and is currently in its fifth edition, released November 2008 [13]. XML version 1.1 was initially published in 2004 [14] and is currently in its second edition [15]. Changes introduced in v1.1 are minimal and mainly focused on liberalising the characters allowed in element and attribute names (i.e., it lists characters you can't use, rather than XML 1.0 which only listed the characters you could). XML 1.0 ed. 5 has adopted some of these character set changes. According to Wikipedia, "there has been discussion of an XML 2.0, although no organization has announced plans for work on such a project" [16].

[1] See Section 1.1: Origin and Goals.

## 2.3 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*

XML 1.0 is very widely used, and underpins a number of key internet standards, such as XHTML[2]. XML 1.1 is less well used, and recommended only for when you need its limited benefits (primarily for creating XML elements in languages not in Unicode 2.0, for example, Mongolian, Yi, Amharic) [17].

Within digital preservation fields, XML is often used to structure metadata and “package” digital objects. For example, METS files are XML structured documents (adhering to the METS schema) that are often used to package digital objects for transport and preservation purposes (e.g., as SIPs, DIPs or AIPs [18]).

## 2.4 Software Support

### 2.4.1 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*

XML handling is well supported by an array of software tools. Text editors typically have XML viewing and editing modes (such as Notepad++ [19]). Dedicated XML editing applications (such as XML Spy [20]) provide comprehensive support for modelling and creating XML and related technologies. Most web browsers, such as Internet Explorer, provide enhanced viewing modes for XML that indent the XML and wrap elements as collapsible/expandable tree nodes, facilitating reading of XML files. Finally, a variety of tools and related technologies are available for processing, interpreting or displaying XML (such as XMLStarlet [21], or the Schematron [22] rule based validation language).

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. The value of data is not necessarily easily realised just because it is encoded as XML, whose notional human readability may be insufficient without software support. This may explain, at least in part, why academic XML based preservation formats, such as XCDL [23], have not taken off.

### 2.4.2 Preservation Software Support

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

***Format identification***

Identification of XML (and related technologies) is supported by DROID [24] and Apache Tika™ [25], amongst others, but only in XML documents that contain an XML declaration. As noted by Van der Knijff, “the problem is that not all XML files actually contain an XML declaration” [26]. This is allowed by the XML 1.0 specification, which states that “XML documents SHOULD begin with an XML declaration which specifies the version of XML being used” [13]. XML documents that do not include this may still be valid and may still be well-formed, they will just not be identifiable as XML by tools relying on the presence of the declaration. This is not the case for XML 1.1 which “MUST begin with an XML declaration” according to the specification [15].

Text based formats such as XML typically cause problems for format identification tools, with signature based approaches rarely sufficient. Van der Knijff demonstrates the application of a more thorough approach based on the ability of an XML parser to successfully process a possible XML file [26].

Identification of a file as XML may not, of course, provide sufficient information about the file or its purpose. In this case the detection and capture of a DTD (Document Type Definition) or schema declaration may be necessary.

***Validation, Conformance Checking and Detecting Preservation Risks***

XML validation is the process of checking a document is both valid and well-formed [27]. 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” [28].

[2] Although it is possible to declare use of XML 1.1 in XHTML documents.

Many XML tools provide functionality to check an XML file's validity (i.e., that it adheres to the structure defined in a DTD or schema [27]) and that it is well-formed (i.e., that it adheres to the XML specification syntax [29]) of an XML file. These tools tend to focus on XML 1.0 and,  according to Harold, do not "work with XML 1.1", instead reporting "well-formedness errors when presented with an XML 1.1 document" [17].

Morrissey notes issues encountered by Portico in validating XML [30]. These include issues that directly, or in some cases quite subtly, make a document invalid or not well-formed, such as:

- "Document type declarations with incorrect public or unresolvable system identifiers
- Documents with white space or comments before the XML declaration
- Documents that omit encoding declarations where the default (UTF-8) encoding is not the one employed by the document
- Documents that incorrectly declare one encoding and employ another
- Documents that declare they are instances of one version of a publisher DTD, but employ elements from a later version of that DTD
- Documents that incorrectly declare that they are "standalone"
- Syntactically invalid Document Type Definitions"

More challenging cases encountered that are more difficult to identify automatically include:

- "Documents that come in fragments, but which do not use the standard XML parsed entity mechanism for connecting the fragments into a single document
- Documents that contain HTML fragments buried in CDATA sections
- Document type definitions that include character entity definition files with names that are the same as standard character entity files, but which in fact contain publisher-specific, non-standard, non-Unicode private characters" [30]

Morrissey also notes that it then becomes necessary to "'reverse-engineer' the publisher's processes in order to render this content truly interoperable—and must often accomplish this absent [of] any explicit documentation—or even indication of variance—from the publisher".

Even if an XML document is syntactically well-formed and valid, the flexibility offered by XML - and especially through schemas trying to enable flexible data models - means the semantic interpretation may still be open to debate (see the section on Complexity). Validation, in this case, offers little help towards the interpretation or rendering of the underlying data.

***Metadata Extraction***

Metadata in XML will be encapsulated in the (text-based) XML content itself. Extraction of this will involve a need to parse and understand the DTD/Schema, and extract the associated text from relevant XML entities. XPath [31] expressions and XSL Transformations [32] could be used to easily perform this.

***Migration***

XML is designed to be easily transformable from one format or standard to another, typically using the Extensible Stylesheet Language (XSL) [33]. Application of such transformation approaches is of course subject to the potential variation in interpretation of XML implementations (as noted above).

## 2.5 Documentation and Guidance

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

XML and related technologies are well documented on the web. Specific applications of XML are typically described by an XML Schema (or DTD); however the ability to understand these (and therefore the XML) is dependent on the Schema's documentation. Both XML Schema and DTD can have embedded documentation.

## 2.6 Complexity

*An impression of the complexity of the format with respect to the impact this is likely to have on the BL 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 a text based language XML is ostensibly human readable, although in the case of particularly sizeable and complex XML formats, the ability to read or understand them becomes more challenging (for example Office Open XML [34]). Comprehension may also be hindered further in XML 1.1 documents, whose less strict rules on character names could potentially make them harder to interpret.

XML can be compacted through representation as Binary XML [35]. This reduces the verbosity and the cost of parsing the XML, but makes it harder to read in standard editors. There are several competing formats, although EXI [36] is the W3C recommendation. The Open Geospatial Consortium provides a Binary XML Encoding specification optimized for geo-related data (e.g., GML) [37].

Binary compression aside, interpreting even reasonable-sized, well-formed and valid XML documents can be difficult. Morrissey notes that the "promise of XML was that it would enable seamless, automated interchange of content, using standard tools, technologies, and shared XML vocabularies. The experience of many cultural memory institutions, however, makes it clear that there are limits to the interoperability of even standards-compliant XML content" [30]. In particular, consider the variations in XML encodings made possible through schemas designed to promote flexibility. If a schema is defined with highly abstract structural elements, freedom is given to document encoding; different syntactic XML implementations could be realised to represent the same underlying data model, for example, "should the Lord of Rings film trilogy be encoded as a single METS file? Three METS files? Three METS files for the individual films and a fourth representing the abstract notion of the Trilogy?" [38]. Such variations - and the resulting confusion in understanding – may impact on the interoperability of XML documents as a means to exchange information between different organisations, as evidenced from practical interoperability experiments [39].

This problem relates more generally to the development of the schemas coupled with the potential technological benefits of using XML - namely flexibility, extensibility and modularity. The desire for harnessing these traits led schema designers within the digital library community to create abstract and extensible schemas that enabled flexibility in encoding the underlying data model and ensured the schema's applicability across a variety of domains; to this end, schema design efforts (within this community) tended to focus on reducing the dependency on other schemas. As a consequence of this approach, each schema (MODS, PREMIS, etc.) contains an overlap of information also contained within other schemas; this increases the potential ways to encode the data model and hinders interoperability. This concept is captured succinctly by McDonough: "Like a rope, it [XML] is extraordinarily flexible; unfortunately, just as with a rope, that flexibility makes it all too easy to hang yourself" [38].

Ultimately, as McDonough argues, the problems surrounding interoperability from XML are "the result of the interplay of technical *and* social factors" [38]. Approaches to solving these issues must also recognise this. McDonough suggests some approaches with respect to interoperability, such as acknowledging that perhaps (within the libraries and archives domain at least) interoperability is of less importance than originally thought, or for those who wish to promote interoperability, taking steps to shift the balance between "internal control" over encoding practices to favour "external connection" with others. Strategies to help could include defining and using schemas which restrict ways of encoding the data model and restrict inclusion of arbitrary schemas, mandating use of particular vocabularies and ontologies, and developing ways to translate between schemas.

Expertise requirements are therefore likely to focus more on specific XML standards. Developing and maintaining expertise in particularly important or frequently utilised XML standards is necessary, such as METS for example. Efforts should be made towards restricting or unifying encoding practices and, especially if external interoperability is of importance, working with other organisations to achieve a common and well-understood approach.

## 2.7 Embedded or Attached Content

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

Content (including binary content) can be embedded within an XML file as a CDATA section, although there are problems around ensuring the CDATA termination string ("]]>") is not used (unless escaped) [40], nor the Nul character (escaped or not) which is invalid anywhere in XML 1.0 and 1.1 documents [41]. An alternative would be to do a binary-to-text encoding (e.g. Base64 encode) of the data, which could then be placed within any String-type XML element (not just CDATA), however there is a file size overhead in this approach (33% with Base64 [42]) and the added complication of knowing whether an XML element is (Base64) encoded or not [43].

In either case, appropriate rendering of this content is dependent on knowledge of approach taken, knowledge of the embedded data itself (e.g., what does the binary data represent) and implementation support in the rendering application.

## 2.8 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*

External content can easily be referenced from an XML document. For example, remotely hosted DTDs or schemas are often referenced from XML documents. These are referenced by URI, meaning that they do not have to point to a resource that exists, and opening up the possibility that an XML document's schema may no longer exist or may have been moved. Even if the schema does exist however, the dynamic nature of the internet could result in temporary unavailability of the resource, as happened to the METS and PREMIS schemas when the Library of Congress was closed during the 2013 United States federal government shutdown [44]. In such circumstances, tools reliant on the existence of these schemas for validation purposes will fail to work. Knight suggests a local XML catalogue [45] as mitigation to the loss (temporary or otherwise) of the critical schemas [46].

## 2.9 Legal Issues

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

XML is an open standard with no known legal restrictions.

## 2.10 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*

XML-Enc [47] is W3C specification for encrypting part or all of the contents of an XML document [48]. Security concerns have been raised about this specification [49].

CDATA sections may also include encrypted data (i.e., binary encrypted data is placed in a CDATA section). In this case, it may not be clear what encryption has been applied, or even that encryption has been used at all.

## 2.11 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*

The W3C recommends encoding XML as UTF8 and defining this encoding in the declaration statement [50]. Characters could be interpreted incorrectly if the encoding is unknown and this could lead to errors in rendering or interpreting XML data.

## 2.12 Preservation Risk Summary

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

XML is an open and very widely used markup language. It is human readable and in principle provides a method of specifying data in an easily interchangeable form. In practice this might not necessarily be enough to make the use, reuse or indeed preservation of data stored in an XML format an easy task. Experiences from the digital preservation community working with metadata standards and XML based alternatives to binary file formats (e.g. Word doc files) have been mixed and so some caution is required.

- **Missing or unavailable external references**
  - DTDs and schemas are typically referenced externally; even temporary loss of these will affect XML processing tools.
  - Other relevant data may also be externally referenced and cause interpretation difficulties if lost or unavailable.

- **Invalid, badly formed or insufficiently specified XML**
  - Poor quality XML could impact on transformation and long term preservation.

- **Interpretability and Interoperability**
  - Documents with unspecified character encodings may cause interpretation problems.
  - Large and complex XML formats are likely to be difficult to read and understand.
  - Schema flexibility allows different approaches to encoding the same information, and may hinder interoperability, especially between organisations.

- **Identification of XML files**
  - Format identification may be challenging even for valid and well-formed XML documents.

- **Embedded content**
  - Content (including binary data) may be embedded in CDATA sections and may not be easily rendered without appropriate knowledge of that content and/or software support.
- **XML data may be encrypted**
  - All or parts of the XML document may be encrypted using XML-Enc.
  - CDATA content may be encrypted without any indication of being so.

## 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.*

Specific recommendations are likely to be more useful when focused on particular schema defined XML formats, but Portico recommend conservative, high level principles when working with XML, focussed around being explicit (e.g. avoid relying on defaults), transforming defensively, testing/validating frequently, and documenting everything [51].

### *Handling Recommendations*

- Ensure created XML documents are well-formed and valid
  - In particular, ensure created XML documents contain an XML declaration to facilitate identification.
  - Ensure the declaration specifies the character encoding.
- Detect and capture relevant DTDs or schemas to aid processing and/or understanding of the content.
- Ensure DTDs/Schemas and profiles of these are documented and that this documentation is preserved.
- As much as possible, take a common approach to encoding data in XML; consider appropriate vocabularies and ontologies to enable consistency.
- Ensure capture of appropriate representation information about content embedded in XML documents.
- Be aware of the possibility of encoded binary data or potentially encrypted data in CDATA sections.
- Use XML 1.0, unless the additional features offered by 1.1 are needed (primarily the need to write XML markup – the <tags> – in one of the languages not covered by Unicode 2.0)

### *Knowledge Recommendations*

- Develop and maintain expertise in particularly important or frequently used XML standards.
- Work with other organisations to achieve a common, well-understood and interoperable approach to use of relevant XML standards.
- Consider implementing a local XML catalogue as mitigation to the loss (temporary or otherwise) of critical DTDs/Schema.

### *Software Recommendations*

- Support enhancements to format identification tools to more accurately detect XML documents (e.g., through parsing techniques which enable identification without XML declaration)

### *Monitoring Recommendations*

XML (1.0 at least) is widely used with a long and stable development history. When used, it often forms a core component in a system's architecture, and can therefore be seen to require development stability (i.e., a slow and minimal change process) to ensure stability of the overall system; changes to XML are likely to be small and infrequent. XML should be monitored on a biennial basis with a low priority.

## 4. References

1. Markup language. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Markup_language.

2. XML schema. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/XML_schema.

3. RSS (Rich Site Summary). *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/RSS.

4. SOAP - Simple Object Access Protocol. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/SOAP.

5. XHTML. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/XHTML.

6. Office Open XML. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Office_Open_XML.

7. Journal Article Tag Suite. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/JATS.

8. Metadata Encoding and Transmission Standard. *Library of Congress.* [Online] [Cited: 27 April 2015.] http://www.loc.gov/standards/mets/.

9. Metadata Object Description Schema. *Library of Congress.* [Online] [Cited: 27 April 2015.] http://www.loc.gov/standards/mods/.

10. PREMIS. *Library of Congress.* [Online] [Cited: 27 April 2015.] http://www.loc.gov/standards/premis/.

11. Standard Generalized Markup Language. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Standard_Generalized_Markup_Language.

12. Extensible Markup Language (XML) 1.0 (First Edition). *W3C.* [Online] 10 February 1998. [Cited: 27 April 2015.] http://www.w3.org/TR/1998/REC-xml-19980210.html.

13. Extensible Markup Language (XML) 1.0 (Fifth Edition). *W3C.* [Online] 26 November 2008. [Cited: 27 April 2015.] http://www.w3.org/TR/2008/REC-xml-20081126/.

14. Extensible Markup Language (XML) 1.1 (First Edition). *W3C.* [Online] 4 February 2004. [Cited: 27 April 2015.] http://www.w3.org/TR/2004/REC-xml11-20040204/.

15. Extensible Markup Language (XML) 1.1 (Second Edition). *W3C.* [Online] 29 September 2006. [Cited: 27 April 2015.] http://www.w3.org/TR/xml11.

16. XML: Versions. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/XML#Versions.

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

18. Open Archival Information System. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Open_Archival_Information_System.

19. *Notepad++.* [Online] [Cited: 27 April 2015.] http://notepad-plus-plus.org/.

20. XMLSpy XML Editor. *Altova.* [Online] [Cited: 27 April 2015.] http://www.altova.com/xmlspy.html.

21. XMLStarlet Command Line XML Toolkit. [Online] [Cited: 27 April 2015.] http://xmlstar.sourceforge.net/.

22. Schematron. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Schematron.

23. **Christoph Becker, Andreas Rauber et al.** A Generic XML Language for Characterising Objects to Support Digital Preservation. *Planets.* [Online] 17 April 2008. [Cited: 27 April 2015.] http://www.planets-project.eu/events/copenhagen-2009/pre-reading/docs/Characterisation%20Languages_Christoph%20Becker_Volker%20Heydegger_Jan%20Schasse_Manfred%20Thaller.pdf.

24. DROID. *Github.* [Online] [Cited: 27 April 2015.] https://digital-preservation.github.io/droid/.

25. Apache Tika. *Apache.* [Online] [Cited: 27 April 2015.] http://tika.apache.org/.

26. **Knijff, Johan van der.** Improved identification of XML: a Python experiment. *The Open Preservation Foundation.* [Online] 11 July 2011. [Cited: 27 April 2015.] http://openpreservation.org/knowledge/blogs/2011/07/11/improved-identification-xml-python-experiment/.

27. XML validation. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/XML_validation.

28. XML Schema Part 1: Structures (Second Edition). *W3C.* [Online] 28 October 2004. [Cited: 27 April 2015.] http://www.w3.org/TR/xmlschema-1/.

29. Well-formed document. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Well-formed_document.

30. **Morrissey, Sheila M.** "More What You'd Call 'Guidelines' Than Actual Rules": Variation in the Use of Standards. *Journal of Electronic Publishing, Volume 14, Issue 1.* [Online] Summer 2011. [Cited: 27 April 2015.] http://dx.doi.org/10.3998/3336451.0014.104.

31. XPath. *Wikidedia.* [Online] [Cited: 01 05 2015.] http://en.wikipedia.org/wiki/XPath.

32. XSLT - Extensible Stylesheet Language Transformations. *Wikipedia.* [Online] [Cited: 01 05 2015.] http://en.wikipedia.org/wiki/XSLT.

33. XLS. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/XSL.

34. **Reimer, Jeremy.** Microsoft Office XML gets fast-tracked to ISO standard. *Ars Technica.* [Online] 13 March 2007. [Cited: 27 April 2015.] http://arstechnica.com/uncategorized/2007/03/microsoft-office-xml-gets-fast-tracked-to-iso-standard/.

35. Binary XML. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Binary_XML.

36. Efficient XML Interchange (EXI) Format 1.0 (Second Edition). *W3C.* [Online] 11 February 2014. [Cited: 27 April 2015.] http://www.w3.org/TR/exi/.

37. Binary Extensible Markup Language (BXML) Encoding Specification (Version: 0.0.8). *Open Geospatial Consortium.* [Online] 13 January 2006. [Cited: 27 April 2015.] http://portal.opengeospatial.org/files/?artifact_id=13636.

38. **McDonough, Jerome.** XML, Interoperability and the Social Construction of Markup Languages: The Library Example. *Digital Humanities Quarterly, Volume 3, Number 3.* [Online] 2009. [Cited: 27 April 2015.] http://digitalhumanities.org/dhq/vol/3/3/000064/000064.html#p25.

39. The Archive Ingest and Handling Test, The Johns Hopkins University Report. *D-Lib Magazine, Volume 11, Number 12.* [Online] December 2005. [Cited: 27 April 2015.] http://www.dlib.org/dlib/december05/choudhury/12choudhury.html.

40. Dealing with data in XML. *IBM DeveloperWorks.* [Online] [Cited: 27 April 2015.] http://www.ibm.com/developerworks/library/x-cdata/.

41. Valid Characters in XML. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/Valid_characters_in_XML.

42. Encoding binary data within XML : alternatives to base64. *StackOverflow.* [Online] [Cited: 27 April 2015.] http://stackoverflow.com/questions/17301940/encoding-binary-data-within-xml-alternatives-to-base64/17354584#17354584.

43. **Rein, Lisa.** Handling Binary Data in XML Documents. *O'Reilly XML.com.* [Online] 24 July 1998. [Cited: 27 April 2015.] http://www.xml.com/pub/a/98/07/binary/binary.html.

44. News: Federal Government Shutdown (revised). *The Library of Congress.* [Online] 3 October 2013. [Cited: 27 April 2015.] http://www.loc.gov/today/pr/2013/13-A07.html.

45. XML Catalog. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/XML_Catalog.

46. *A Short Story about XML Schemas, Digital Preservation and Format Libraries.* **Knight, Steve.** 1, 2012, International Journal of Digital Curation, Vol. 7, pp. 72-80. http://dx.doi.org/10.2218/ijdc.v7i1.215.

47. XML Encryption Syntax and Processing. *W3C.* [Online] 10 December 2002. [Cited: 27 April 2015.] http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/Overview.html.

48. XML Encryption. *Wikipedia.* [Online] [Cited: 27 April 2015.] http://en.wikipedia.org/wiki/XML_Encryption.

49. RUB Researchers break W3C standard-XML Encryption is insecure: Large companies affected. *Ruhr-Universität Bochum.* [Online] 19 October 2011. [Cited: 27 April 2015.] http://aktuell.ruhr-uni-bochum.de/pm2011/pm00330.html.en.

50. XML Encoding. *W3Schools.* [Online] [Cited: 27 April 2015.] http://www.w3schools.com/xml/xml_encoding.asp.

51. *Portico: A Case Study in the Use of XML for the Long-Term Preservation of Digital Artifacts.* **Morrissey, Sheila M, et al., et al.** 2010. Proceedings of the International Symposium on XML for the Long Haul: Issues in the Long-term Preservation of XML. Vol. 6. http://www.balisage.net/Proceedings/vol6/html/Morrissey01/BalisageVol6-Morrissey01.html. doi:10.4242/BalisageVol6.Morrissey01.

52. Portico: A Case Study in the Use of XML for the Long-Term Preservation of Digital Artifacts. *In Proceedings of the International Symposium on XML for the Long Haul: Issues in the Long-term Preservation of XML, Balisage Series on Markup Technologies, volume 6.* [Online] 2 August 2010. [Cited: 27 April 2015.] http://www.balisage.net/Proceedings/vol6/html/Morrissey01/BalisageVol6-Morrissey01.html.