Office Open XML (OOXML) Formats Binding Profile

1. Introduction

The term labelling is the process of determining the appropriate metadata for a given data object, creating the metadata label and binding the metadata label to the data object. A binding is a relationship between a data object and a metadata label. A binding is realized by applying a binding mechanism. If a metadata label must be bound to a data object, both the metadata label and the data object are input to the binding mechanism. The output of the binding mechanism is the binding of a data object and metadata label (see Figure 1) which says that the data object and the metadata label belong together. The binding can be recorded as a structured data object, known as a Binding Data Object (BDO).

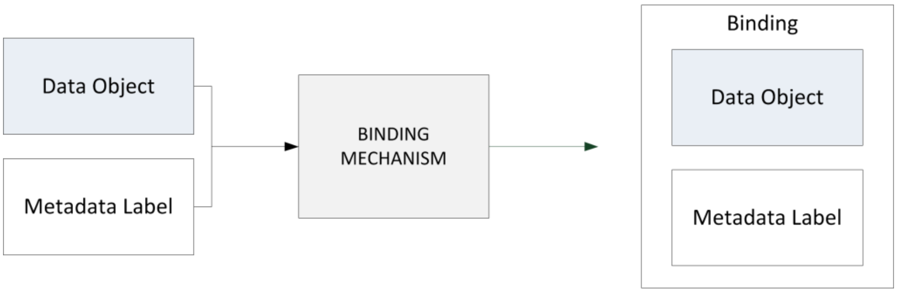


Figure 1 Creation of a binding

STANAG 4778 (Reference [3]) standardizes the binding of a data object and metadata label by specifying a common binding mechanism and a syntax for representing the BDO. However, to support information management and information sharing requirements it is necessary to further profile the application of STANAG 4778 to facilitate locating a BDO and embedding a BDO in data objects.

2. OOXML Introduction

The Office Open XML Formats (OOXML) are defined ISO/IEC 29500 (Reference [1]) and offer standards for representing office documents, including spreadsheets, presentations and word processing documents.

OOXML adopts a structured format which consists of a number of XML-based files packaged into an archive file according to the Open Packaging Conventions (OPC), which is defined in Part 2 of ISO/IEC 29500 (Reference [1]).

OOXML allows for custom XML files to be included within the package without impacting the underlying application. This provides a mechanism for a metadata to be bound to the OOXML document and maintained within the package.

This profile for the OOXML describes how metadata can be maintained.

3. Identification

The profile for OOXML is uniquely identified by the Canonical Identifier shown in Table 1.

Table 1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:ooxml
Version Identifier	urn:nato:stanag:4778:profile:ooxml:1:0

It is recognized that this profile may evolve during its review cycle. For example, a review might identify:

- changes to the base OOXML standard e.g.
    - introduction of new package parts
- additional profiles for OOXML e.g.
    - different combinations of package parts
    - bindings to elements with a package part (e.g. binding metadata to paragraphs within a document)
- improvements to the existing profiles based upon operational feedback

Therefore this version of the profile is uniquely identified by the Version Identifier shown in Table 1.

Subsequent versions of this profile will maintain the same Canonical Identifier, but define a new Version Identifier.

4. Standards

[1] ISO/IEC 29500-2 “Office Open XML File Formats - Part 2: Open Packaging Conventions”, at http://standards.iso.org/ittf/PubliclyAvailableStandards/c061796_ISO_IEC_29500-2_2012.zip, August 2012

[2] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium

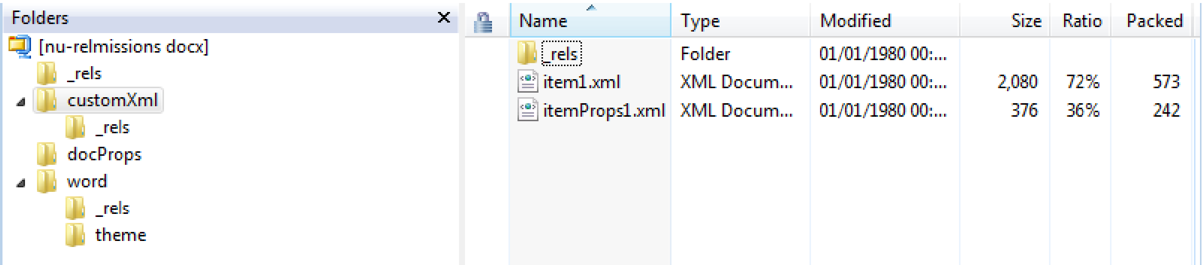
[3] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium

5. Notational Conventions

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in *italics* indicate terms derived from Reference [3].

6. Structure

The structure of an OOXML package consists of a number of folders which contain different components of the document.



```
Folders
[nu-relmissions docx]
    _rels
    customXml
        _rels
    docProps
    word
        _rels
        theme
```

Name	Type	Modified	Size	Ratio	Packed
_rels	Folder	01/01/1980 00:...			
item1.xml	XML Docum...	01/01/1980 00:...	2,080	72%	573
itemProps1.xml	XML Docum...	01/01/1980 00:...	376	36%	242

Figure 2: General Structure of an OPC Package

The structure, as shown in **Figure 2**, generally consists of:

- An application specific folder, for example “word”, “ppt” or “xl”.
- A “customXml” folder in which arbitrary XML files can be stored.
- A “docProps” folder in which core and custom document properties are held.
- Multiple “_rels” folder which contains details of the parts within a folder.

This structure is then packaged into an archive file with an application specific extension (for example, .docx).

The document that is displayed to a user is generally split over a number of different XML files contained with the package. This does not present a problem when applying granular metadata to different parts of the document.

However care must be taken when the intention is to bind metadata to the complete document (refer to Microsoft Office File Types section below for normative text related to binding metadata to a whole document). For example, the XML file /word/document.xml within a Microsoft Word OPC package does not contain the headers or footers of the document (these are contained in the separate files /word/header1.xml and /word/footer1.xml.)

7. Custom XML

In order to support metadata binding within an OPC package, a single CustomXML file SHALL be maintained within the OPC package with the Metadata Binding Container namespace, “urn:nato:stanag:4778:bindinginformation:1:0”.

DataReference elements SHALL be used to reference the files within the OPC package.

Data elements SHALL NOT be used.

When referring to files, or portions of files, within the OPC package, absolute URIs from the package root SHALL be used with the *DataReference* element. For example,

<DataReference URI=”/word/document.xml”/>

Microsoft Office File Types

Microsoft Office has used the OOXML standard, since Microsoft Office 2007, for a number of its document types, including Microsoft Word, Microsoft Excel and Microsoft PowerPoint.

When binding metadata to a complete document (as opposed to a specific part of a document), all of the files (when they are present within the package) listed in the “Package File” for the particular document type SHALL be referenced in the binding (see in **Table 2**).

Table 2: Packages Files to be Referenced in a Binding to a Complete Document[1]

Application	Package File	Description
Microsoft Word	/word/document.xml	The document.
	/word/styles.xml	The styles within the document.
	/word/header<N>.xml	The headers for sections within the document.
	/word/footer<N>.xml	The footers for sections within the document.
	/word/media/*	The media (e.g. pictures) embedded in the document.
	/word/footnotes.xml	The footnotes.
	/word/endnotes.xml	The endnotes.
	/word/comments.xml	The review comments.
	/word/commentsExtended.xml	The review comments.
Microsoft Excel	/xl/workbook.xml	The workbook.
	/xl/styles.xml	The styles within the workbook.
	/xl/sharedStrings.xml	The strings shared between worksheets.
	/xl/worksheets/sheet<N>.xml	The worksheets within the workbook.
	/xl/charts/chart<N>.xml	The charts on a worksheet.
	/xl/charts/colors<N>.xml	The colors of a chart on a worksheet.
	/xl/charts/styles<N>.xml	The style of a chart on a worksheet.
	/xl/pivotTables/pivotTable<N>.xml	The pivotTables on a worksheet.
	/xl/comments<N>.xml	The comments on a worksheet.
	/xl/media/*	The media (e.g.) pictures embedded on the worksheets.
Microsoft PowerPoint	/ppt/presentation.xml	The presentation.
	/ppt/slides/slide<N>.xml	The slides within the presentation.
	/ppt/slideLayouts/slideLayout<N>.xml	The slide layouts.
	/ppt/slideMaster/slideMaster<N>.xml	The slide masters.
	/ppt/comments/comment<N>.xml	The comments on a slide.
	/ppt/media/*	The media (e.g. pictures) embedded on the slides.
	/ppt/presProps.xml	The additional presentation-wide properties.
	/ppt/viewProps.xml	The additional presentation-wide properties.

The common document properties package files (where they are present within the package) SHALL also be referenced in the binding (see Table 3).

Additional package files, beyond those listed in Table 2 and Table 3, MAY be referenced in the binding (e.g. packages files created by COI-specific Office Add-Ins).

Table 3: Common Packages Files to be Referenced in a Binding to a Complete Document

Package File	Description
/docProps/core.xml	The common document properties.
/docProps/app.xml	The application-specific document properties.
/docProps/custom.xml	The custom (e.g. user defined) document properties.

[1] The notation “<N>” in the “Package File” column indicate an increasing integer. For example, “/word/header<N>.xml” would indicate the package files “/word/header1.xml” and “/word/header2.xml” in a document with two headers.

Figure 3 shows the contents of a CustomXML file, stored in /customXml/item1.xml, for a simple Microsoft Word document containing an embedded image. Its uses Confidentiality Metadata Labels (Reference [2]) as example metadata.

```
<mb:BindingInformation
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>2016-11-10T12:30:00Z</slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="/word/document.xml"/>
    <mb:DataReference URI="/word/styles.xml"/>
    <mb:DataReference URI="/word/header1.xml"/>
    <mb:DataReference URI="/word/footer1.xml"/>
    <mb:DataReference URI="/word/media/image.jpeg" xmime:contentType="image/jpeg"/>
    <mb:DataReference URI="/word/footnotes.xml"/>
    <mb:DataReference URI="/word/endnotes.xml"/>
    <mb:DataReference URI="/docProps/app.xml"/>
    <mb:DataReference URI="/docProps/core.xml"/>
    <mb:DataReference URI="/docProps/custom.xml"/>
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
```

Figure 3: Example CustomXML file for a Simple Microsoft Word Document