# Extensible Metadata Platform (XMP) 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).

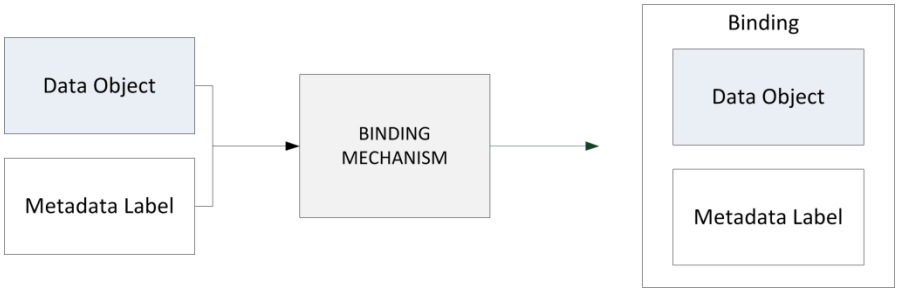


**Figure 1 Creation of a binding**

STANAG 4778 (Reference [5]) 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. XMP Introduction

The Extensible Metadata Platform (XMP) specifications are defined in ISO 16684-1:2012 (Reference [1]) and offer standards for the creation, processing and interchange of standardized and custom metadata for specific finite data formats.

XMP is an XML-based format modelled after the World Wide Web Consortium (W3C) Resource Description Framework (RDF) (Reference [2]) that standardizes a data model, serialization of the data model in XML, core metadata properties, definition and processing of customized metadata and a mechanism for embedding XMP information into documents, such as JPEG and PDF.

XMP offers an alternative for storing metadata in side car files whereby the XMP metadata is associated with a file format by embedding the metadata in that file format. The file formats that are supported by XMP and the locations for embedding the XMP metadata within those file formats is documented in XMP Part 3, Storage in Files (Reference [3]).

An instance of the XMP data model is called an XMP packet. An XMP packet is a set of XMP metadata properties each of which has a name and value. A value can take the form of a simple value, a structured value or an array value. This Binding Profile for XMP describes how metadata should be incorporated within an XMP packet as a structured value.

## 3. Identification

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

**Table 1: Profile Identifiers**

| Type | Identifier |
|---|---|
| Canonical Identifier | urn:nato:stanag:4778:profile:xmp |
| Version Identifier | urn:nato:stanag:4778:profile:xmp:1:0 |

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

- changes to the base XMP standard
- 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 16684-1:2012 “Graphic Technology – Extensible metadata platform (XMP) specification – Part 1: Data model, serialization and core properties”, February 2012 (publically available at: http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/xmp/pdfs/XMP%20SDK%20Release%20cc-2016-08/XMPSpecificationPart1.pdf)

[2] W3C Recommendation, “RDF Primer”, at https://www.w3.org/TR/2004/REC-rdf-primer-20040210/, February 2004.

[3] Adobe XMP, “XMP Specification Part 3, Storage in Files”, at http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/xmp/pdfs/XMP%20SDK%20Release%20cc-2016-08/XMPSpecificationPart3.pdf, August 2016.

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

[5] 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 [5].
- `Courier font` indicates syntax derived from XMP (Reference [1]) and RDF (Reference [2]).

## 6. Structure

An `XMP packet` property of `structure value` form is a container for zero or more named fields, where each field in that structure has a unique name. Each unique name needs to be an XML expanded name. There are no limitations on the field name namespace or the field name value form. **Figure *2*** below (Reference [1]) illustrates a single structured property with three fields.

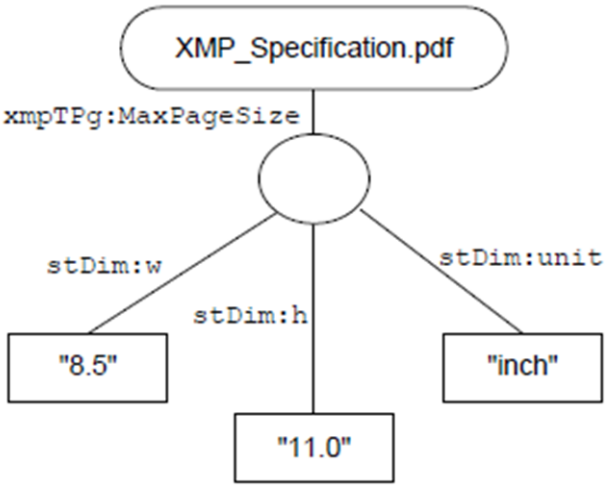


**Figure 2 XMP Structured Property**

For the serialization of a `XMP packet` a single `rdf:RDF` XML element is used. The element content for an unqualified XMP property with a `structure value` is a nested `rdf:Description` element. **Figure 3** below illustrates a serialized XMP property with a `structure value` with three fields as depicted in **Figure 2**.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:xmpTPg="http://ns.adobe.com/xap/1.0/t/pg/"
 xmlns:stDim="http://ns.adobe.com/xap/1.0/sType/Dimensions#">
  <rdf:Description rdf:about="" >
   <xmpTPg:MaxPageSize>
    <rdf:Description>
     <stDim:h>11.0</stDim:h>
     <stDim:w>8.5</stDim:w>
     <stDim:unit>inch</stDim:unit>
    </rdf:Description>
   </xmpTPg:MaxPageSize>
  </rdf:Description>
</rdf:RDF>
```

**Figure 3 Serialized XMP Structured Property**

This profile uses the same serialization (illustrated in **Figure *3***) of a structure value for **storing** the *BindingInformation* in an XMP packet.

Specifically:

- RDF provides for XML content as a literal value. Therefore, the *BindingInformation* SHALL be converted to an XML literal string value compliant with Section 5.3 Reference [5].
- The *BindingInformation* SHALL be a stored as a value within a 'bindingInformation' XML element qualified by the namespace *urn:nato:stanag:4778:bindinginformation:1:0:xmp#*.
- The 'bindingInformation' XML element SHALL contain a `RDF:datatype` attribute with a value '`http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral`'.
- The 'bindingInformation' XML element (containing the *BindingInformation*) SHALL be stored as a child XML element of an `rdf:Description` element that is wrapped in a single `rdf:RDF` XML element.

- The serialized `rdf:RDF` XML element (containing the *BindingInformation*) is known as the XMP binding packet.
- The *BindingInformation* MUST contain at least one *MetadataBinding* that contains a null *DataReference URI* attribute value (refer to Same-Document References Section of Reference [5]) that semantically indicates a binding relationship of the metadata to the data object.
- A relationship is defined between the data object and the *BindingInformation* by embedding the XMP binding packet in the data object (of a supported XMP file format).
- The supported XMP file formats are listed in Reference [3].
- Depending on the file format the XMP binding packet SHALL be embedded in the data object as specified in Reference [3].

**Figure 4** shows the structure of an XMP binding packet. This example uses Confidentiality Metadata Labels (Reference [4]) as example metadata.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mbxmp="urn:nato:stanag:4778:bindinginformation:1:0:xmp#" >
  <rdf:Description rdf:about="" >
    <mbxmp:bindingInformation
      rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral" >
      &lt;mb:BindingInformation
      xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;
      xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;
      xmlns:xsd=&quot;http://www.w3.org/2001/XMLSchema&quot;
      xmlns:mb=&quot;urn:nato:stanag:4778:bindinginformation:1:0&quot;
      xmlns:ds=&quot;http://www.w3.org/2000/09/xmldsig#&quot;
      xmlns:xmime=&quot;http://www.w3.org/2005/05/xmlmime&quot;&gt;
      &lt;mb:MetadataBindingContainer&gt;
        &lt;mb:MetadataBinding&gt;
          &lt;mb:Metadata&gt;
            &lt;slab:originatorConfidentialityLabel

xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;&gt;
              &lt;slab:ConfidentialityInformation&gt;
                &lt;slab:PolicyIdentifier&gt;ACME&lt;/slab:PolicyIdentifier&gt;
                &lt;slab:Classification&gt;UNCLASSIFIED&lt;/slab:Classification&gt;
              &lt;/slab:ConfidentialityInformation&gt;
              &lt;slab:CreationDateTime&gt;
                2015-09-30T12:30:00Z
              &lt;/slab:CreationDateTime&gt;
            &lt;/slab:originatorConfidentialityLabel&gt;
          &lt;/mb:Metadata&gt;
          &lt;mb:DataReference URI=&quot;&quot;  /&gt;
        &lt;/mb:MetadataBinding&gt;
      &lt;/mb:MetadataBindingContainer&gt;
    &lt;/mb:BindingInformation&gt;
    </mbxmp:bindingInformation>
  </rdf:Description>
</rdf:RDF>
```

**Figure 4: Example XMP binding packet**

## 7. XMP Sidecar Files

If a data object file format is not supported by XMP (refer to Reference [3] to determine XMP supported file formats), XMP offers a simple naming convention to relate the Binding Data Object (BDO) with the data object. As the BDO is stored separately from the data object, there is a risk that the association between the metadata and the data object may get lost. XMP-aware applications that support this profile are REQUIRED to conform with the following rules:

1) The XMP binding packet (refer to the Extensible Metadata Platform (XMP) Binding Profile) SHALL be written as a complete and well-formed XML document, including the leading XML declaration.
2) The file extension for the XMP binding packet file SHALL be '.xmp'.
3) If a MIME type is required 'application/rdf+xml' SHALL be used.
4) Values used in *DataReference* URI with the BDO SHALL use relative paths and assume that the data object resides at the same location as the BDO.

As an example, distinct metadata may be associated with a text file, "example.txt", by creating a *BindingInformation* element and storing it as "example.txt.xmp" in the same folder as the original file.

**Figure 5** shows an example XMP sidecar file for "example.txt". This example uses Confidentiality Metadata Labels (Reference [4]) as example metadata.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mbxmp="urn:nato:stanag:4778:bindinginformation:1:0:xmp#" >
  <rdf:Description rdf:about="" >
    <mbxmp:bindingInformation
      rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral" >
      &lt;mb:BindingInformation
      xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;
      xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;
      xmlns:xsd=&quot;http://www.w3.org/2001/XMLSchema&quot;
      xmlns:mb=&quot;urn:nato:stanag:4778:bindinginformation:1:0&quot;
      xmlns:ds=&quot;http://www.w3.org/2000/09/xmldsig#&quot;
      xmlns:xmime=&quot;http://www.w3.org/2005/05/xmlmime&quot;&gt;
      &lt;mb:MetadataBindingContainer&gt;
        &lt;mb:MetadataBinding&gt;
          &lt;mb:Metadata&gt;
            &lt;slab:originatorConfidentialityLabel

xmlns:slab=&quot;urn:nato:stanag:4774:confidentialitymetadatalabel:1:0&quot;&gt;
              &lt;slab:ConfidentialityInformation&gt;
                &lt;slab:PolicyIdentifier&gt;ACME&lt;/slab:PolicyIdentifier&gt;
                &lt;slab:Classification&gt;UNCLASSIFIED&lt;/slab:Classification&gt;
              &lt;/slab:ConfidentialityInformation&gt;
              &lt;slab:CreationDateTime&gt;
                2015-09-30T12:30:00Z
              &lt;/slab:CreationDateTime&gt;
            &lt;/slab:originatorConfidentialityLabel&gt;
          &lt;/mb:Metadata&gt;
          &lt;mb:DataReference URI=&quot;./example.txt&quot;
xmime:contentType=&quot;text/plain&quot; /&gt;
        &lt;/mb:MetadataBinding&gt;
      &lt;/mb:MetadataBindingContainer&gt;
    &lt;/mb:BindingInformation&gt;
    </mbxmp:bindingInformation>
  </rdf:Description>
</rdf:RDF>
```

**Figure 5: Example XMP Sidecar file**