XMP SDK Release notes – Version 2.9

Introduction

This document contains the release notes for version 2.9 of the XMP Toolkit and includes changes that have taken place since version 2.8 was released. The following sections cover new features, bug fixes, and known problems in the Toolkit.

What's New

No new major features but some minor new functions have been added in this release

- MetaXAP::GetNamespacePrefix, pass in the full URI for a registered namespace and get back the prefix.
- MetaXAP::GetNamespaceURL, pass in the prefix for a registered namespace and get back the full URI.
- UtilityXAP::GetLocalizedText, handles many of the complexities of writing an alt-text property. See UtilityXAP.h for details.
- UtilityXAP::SetLocalizedText, handles many of the complexities of reading an alt-text property. See UtilityXAP.h for details.
- UtilityXAP::MakeLocalTime, converts a XAPDateTime value to contain the local time and UTC offset.
- UtilityXAP::MakeUTCTime, converts a XAPDateTime value to contain the UTC time.
- UtilityXAP::SetTimeZone, determines the local time zone offset from UTC, using only POSIX functions.
- Much of the documentation is changed to use xmp in namespace prefixes instead of XAP but the toolkit has not yet changed.

Changes and Bug fixes

- Fix init/term functions to avoid memory leaks.
- Allow periods as NCName characters, for example in the property name parameter to MetaXAP::set.
- Fix several XML namespace issues:
  - Add rdf: namespace prefix to the about attribute of the rdf:Description element. Note that from a formal XML namespace point of view this was implicit because attributes inherit their namespace from the element and not the global default namespace. This was done simply to comply with RDF recommended practice.
  - Change to a single global namespace registry. The namespace URIs and prefixes are both unique in this registry. If a URI is already registered,

attempts to register it with a different prefix do nothing. If a prefix for a new URI is already in use, a unique prefix is computed by adding a numeric suffix.

  - Namespaces encountered when parsing are automatically registered when first encountered if necessary. Parsed input can of course have distinct prefixes that resolve to the same URI. Serialized output will use the unique prefix from the global namespace registry.
  - Remove the default namespace declaration from the rdf:Description elements. The serialized output always uses explicit prefixes.
- Always write the x:xmpmeta element in serialized output. Allow either xmpmeta or xapmeta on input if the client requires this bracketing element.
- Add UNIX make files, with specific cases for Solaris, HP-UX, AIX, and Linux. Note that no promise is made that any particular compiler or STL implementation on these platforms can build XMP. Problems with templates and STL seem to be common on UNIX and vary from one compiler to another, even on the same platform.
- Define NDEBUG in the Mac and UNIX release builds. This was already defined in Windows. This disables the ANSI C assert macro. Also change XMPAssert to call a function instead of throwing inline; this saves a huge amount of code.
- Fix a bug in gen_convert_text.cpp where convert_utf is broken if wchar_t is not the same size as unsigned short. It was casting pointers from one to the other type.
- Fix a bug in MetaXAP::remove where simple aliases were left behind. For example, pdf:Author is an alias of dc:creator/*[1]. The bug was that removing all of dc:creator would leave a value behind for pdf:Author.

Known problems

- Aliasing is mishandled for some cases of qualified properties. If the items in an RDF container have qualifiers and the container has an alias, the qualifiers get copied as actual items in the alias container. For example, the xap:Keywords property is an alias of dc:subject. Suppose you parse the following RDF:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about=""
                   xmlns:dc="http://purl.org/dc/elements/1.1/">

    <dc:subject>
      <rdf:Bag>
        <rdf:li rdf:parseType="Resource">
          <rdf:value>keyword</rdf:value>
          <dc:qualifier>info</dc:qualifier>
        </rdf:li>
      </rdf:Bag>
    </dc:subject>

  </rdf:Description>
</rdf:RDF>
```

  The alias expansion will be as though you parsed this RDF:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```
  <rdf:Description rdf:about=""
                   xmlns:dc="http://purl.org/dc/elements/1.1/"
                   xmlns:xap="http://ns.adobe.com/xap/1.0/">

    <dc:subject>
      <rdf:Bag>
        <rdf:li rdf:parseType="Resource">
          <rdf:value>keyword</rdf:value>
          <dc:qualifier>info</dc:qualifier>
        </rdf:li>
      </rdf:Bag>
    </dc:subject>

    <xap:Keywords>
      <rdf:Bag>
        <rdf:li>keyword</rdf:li>
        <rdf:li>info</rdf:li>
      </rdf:Bag>
    </xap:Keywords>

  </rdf:Description>
</rdf:RDF>
```

- Note: the above problem and the fixed problem in MetaXAP::remove might seem perplexing. Many would assume that aliases are just alternate names, like Mac OS file aliases or Windows file shortcuts. However, the current XMP toolkit implements aliases by copy. The problems arise from bugs in managing the separate copies.
- The packet scanner fails if the end of a packet is exactly at the end of a file.
- An empty structure or container in the RDF gets turned into a simple property with an empty string value when parsed. RDF allows empty structures (inner rdf:Description elements) and empty rdf:Bag or rdf:Seq containers.
- Qualified structured values don't work: If the value (rdf:value part) of a qualified property is itself a structure, the XMP toolkit drops the entire value leaving an empty string simple value. The following RDF is an example:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="" xmlns:ns="ns:test/">

    <ns:prop rdf:parseType="Resource">
      <rdf:value rdf:parseType="Resource">
        <ns:field>value</ns:field>
      </rdf:value>
      <ns:qualifier>info</ns:qualifier>
    </ns:prop>

  </rdf:Description>
</rdf:RDF>
```