

Exiv2 - Feature #640

method is missing in Exiv2 to get list of known XMP namespaces

18 Jul 2009 10:06 - mikolaj -

| | | | |
|---|-------------|------------------------|--------------------|
| Status: | Closed | Start date: | 18 Jul 2009 |
| Priority: | Normal | Due date: | |
| Assignee: | Robin Mills | % Done: | 100% |
| Category: | xmp | Estimated time: | 10.00 hours |
| Target version: | 0.26 | | |
| Description | | | |
| If application query for an XMP namespace that exiv2 don't know (for ex. IPTC Ext with Exiv2 0.18.2), we have an assertion and application (digiKam in this case) crash. | | | |
| <hr/> | | | |
| 1. I have latest exiv2 (svn ver. 1862) with latest additions of IPTC Ext and PLUS namespaces and I don't see them in list of XMP tags - recompiled kexiv2 and digikam also. Leftovers from older compilations? Note: generally I am on 4.3 branch but switched only kdegraphics/libs to trunk and compiles OK (with including of new feature in digiKam). | | | |
| Because, only with XMP, the list of namespace are not provided by EXIV2. I need to know which namespace i want to list exactly. | | | |
| http://lxr.kde.org/source/KDE/kdegraphics/libs/libkexiv2/libkexiv2/kexiv2xmp.cpp#936 | | | |
| Note : if i query for an XMP namespace that exiv2 don't know (for ex. IPTC Ext with Exiv2 0.18.2), we have an assertion and digiKam crash... | | | |
| In fact, a method is missing in Exiv2 to get list of know XMP namespace... | | | |
| ----- | | | |
| Related issues: | | | |
| Related to Exiv2 - Bug #11113: Crash in Exiv2 0.25 | | Closed | 31 Aug 2015 |
| Related to Exiv2 - Bug #11116: Issues with namespace 'video' | | Closed | 08 Sep 2015 |

Associated revisions

Revision 3891 - 24 Aug 2015 09:45 - Robin Mills

exiv2 -vV lists registered Namespaces. #640 and topic 2169 <http://dev.exiv2.org/boards/3/topics/2169>

Revision 3914 - 29 Aug 2015 16:43 - Robin Mills

#640 #751 Fix to test suit to prevent crash in exiv2json -x test/data/BlueSquare.xmp (see #751 for discussion)

Revision 3916 - 29 Aug 2015 20:00 - Robin Mills

#640 and # 751. Fixing linux build-breaker.

Revision 3925 - 03 Sep 2015 20:51 - Robin Mills

#640 Modified XmpParser::getRegisteredNamespaces to use Exiv2::Dictionary.

Revision 3931 - 08 Sep 2015 13:01 - Robin Mills

#640 Public API to reveal all namespaces known to Exiv2 and XMPsdk is XmpProperties::registeredNamespaces(Exiv2::Dictionary&)

Revision 3932 - 08 Sep 2015 13:15 - Robin Mills

#640. Correction to r3931.

History

#1 - 23 Aug 2015 16:16 - Robin Mills

- Target version set to 53

#2 - 24 Aug 2015 09:45 - Robin Mills

- Category set to xmp
- Status changed from New to Closed
- Assignee set to Robin Mills
- Target version changed from 53 to 0.26

[r3891](#) I've updated the output of `exiv2 --verbose --version` to include a list of registered namespaces. The code in `version.cpp/namespaceDumper()` is a callback from XMPsdk to dump the registered namespace. A variant of that code may be used to dynamically enumerate the XMPsdk namespace registry.

#3 - 28 Aug 2015 19:57 - Robin Mills

- Status changed from Closed to Feedback
- Assignee changed from Robin Mills to Andreas Huggel

I have changed my mind about this. I revisited `samples/exiv2json.cpp` and discovered that it's not possible to determine the registered namespaces from an application.

[r3912](#) added the API:

```
static void Exiv2::XMPParser::getRegisteredNamespaces(std::map<std::string, std::string>&);
```

I've modified `samples/exiv2json.cpp` to report the namespaces in use. I've modified `src/version.cpp` to use `getRegisteredNamespaces`.

I'm assigning this to Andreas for comment and feedback about the API.

TODO:

Just before checking this in, I discovered an exception in test 1054 "Caught Exiv2 exception 'XMP Toolkit error 9: Fatal namespace map problem'". Infuriatingly, it's only in the release version with the file `./test/data/BlueSquare.xmp`. Before I spend time to fix this, I'd appreciate feedback on the API.

#4 - 29 Aug 2015 20:28 - Robin Mills

I've found and fixed the issue with test 1054. I've discussed the fix in [#751](#).

#5 - 04 Sep 2015 08:12 - Andreas Huggel

Sorry, this took me longer than it should have.

We have a static class that provides access to metadata reference information for each type of metadata. These classes are [Exiv2::ExifTags](#), [Exiv2::IptcDataSets](#) and [Exiv2::XmpProperties](#). (The high-level class-design is outlined in the [API doc intro](#).)

A new library API to return a list of known XMP namespaces and prefixes is such reference data, so I'd add it to `Exiv2::XmpProperties`, re-using as many of the existing types from there as possible, rather than re-inventing new ones, following the guideline that a smaller API is a better API.

In this class, there is already a (public) registry `nsRegistry_`, but we currently only use it for custom namespaces. I'm wondering if we should (could?) add the built-in namespaces to this registry as well (initializing it in `XmpParser::initialize()`)? The registry is public, so that would solve this issue.

#6 - 04 Sep 2015 10:46 - Robin Mills

I've tried to implement this and it doesn't work out very well. The obstacle is calling `XmpParser::initialize()` which requires a lock. By defining a new API `Exiv2::XmpParser::getRegisteredNamespace()`, we call `initialize()` correctly and invisibly.

My other concern is how cumbersome things become. Iterating the `nsRegistry_` is troublesome. Accessing the prefix/uri involves code something like `(*it).second.ns_` (everything I tried crashed). It's so much easier to iterate an `Exiv2::Dictionary`.

I realise that, in principle, there is more information potentially available via `XmpNsInfo.xmpPropertyInfo_`, however I don't believe these structures are populated. My feeling is that your proposal, although elegant, adds complexity for no purpose. `XmpParser::initialize` call `SXMPMeta::DumpNamespaces` to update `nsRegistry_`. So, I believe we can do this your way and mine!

However my primary concern is how to invisibly and correctly call `XmpParser:: initialize()`.

#7 - 06 Sep 2015 10:33 - Andreas Huggel

Shouldn't have mentioned `nsRegistry_`, that was really just 'wondering'. The feedback was supposed to just be:

1. API - I feel this belongs in class `XmpProperties`, for the reasons given.
2. Implementation - the simpler the better, I'm not really worried about that.

So I played a bit, trying to implement a trivial version - and opened a can of worms:

```
Index: include/exiv2/properties.hpp
=====
--- include/exiv2/properties.hpp      (revision 3927)
+++ include/exiv2/properties.hpp      (working copy)
@@ -224,6 +224,9 @@
     // DATA
     static NsRegistry nsRegistry_;          //!< Namespace registry

+     typedef std::vector<std::pair<const char*, const char*> > NsList;
+     static void registeredNamespaces(NsList& nsList);
+
     }; // class XmpProperties

     /*!
Index: src/properties.cpp
=====
--- src/properties.cpp                (revision 3927)
+++ src/properties.cpp                (working copy)
@@ -2391,6 +2391,14 @@
     return xn;
 }

+ void XmpProperties::registeredNamespaces(NsList& nsList)
+ {
+     nsList.clear();
+     for (unsigned int i = 0; i < EXV_COUNTOF(xmpNsInfo); ++i) {
+         nsList.push_back(std::make_pair(xmpNsInfo[i].prefix_, xmpNsInfo[i].ns_));
+     }
+ }
+
+ void XmpProperties::printProperties(std::ostream& os, const std::string& prefix)
+ {
+     const XmpPropertyInfo* pl = propertyList(prefix);
```

Simple, straightforward, and works well at first glance. But compared to `$ exiv2 -vV | grep xmlns`, it produces less output. At the same time, it contains several entries that `$ exiv2 -vV | grep xmlns` doesn't have!

So, besides the discussion of 1. and 2. above, we've uncovered that the XMP namespaces known by Exiv2 and its internal XMPSDK are not in sync.

To fix that, we'd have to register namespaces that Exiv2 knows and XMPSDK doesn't in XMPSDK, and vice versa, probably, add namespaces that XMPSDK knows but Exiv2 doesn't to our lists in properties.cpp.

Here's the difference between the output of this simple function (<) and `exiv2 -vV` (>):

```
2c2,3
< xmlns=audio:http://www.audio
---
> xmlns=album:http://ns.adobe.com/album/1.0/
> xmlns=asf:http://ns.adobe.com/asf/1.0/
3a5,6
> xmlns=bmsp:http://ns.adobe.com/StockPhoto/1.0/
> xmlns=creatorAtom:http://ns.adobe.com/creatorAtom/1.0/
6a10
> xmlns=DICOM:http://ns.adobe.com/DICOM/
14d17
< xmlns=Iptc4xmpExt:http://iptc.org/std/Iptc4xmpExt/2008-02-29/
16c19,21
< xmlns=iptc:http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/
---
> xmlns=iX:http://ns.adobe.com/iX/1.0/
> xmlns=jp2k:http://ns.adobe.com/jp2k/1.0/
> xmlns=jpeg:http://ns.adobe.com/jpeg/1.0/
25a31,36
> xmlns=pdfaExtension:http://www.aiim.org/pdfa/ns/extension/
> xmlns=pdfaField:http://www.aiim.org/pdfa/ns/field#
> xmlns=pdfaid:http://www.aiim.org/pdfa/ns/id/
> xmlns=pdfaProperty:http://www.aiim.org/pdfa/ns/property#
> xmlns=pdfaSchema:http://www.aiim.org/pdfa/ns/schema#
> xmlns=pdfaType:http://www.aiim.org/pdfa/ns/type#
26a38,39
> xmlns=pdfx:http://ns.adobe.com/pdfx/1.3/
> xmlns=pdfxid:http://www.npes.org/pdfx/ns/id/
28a42,43
```

```

> xmlns=png:http://ns.adobe.com/png/1.0/
> xmlns=rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#
33a49
> xmlns=stMfs:http://ns.adobe.com/xap/1.0/sType/ManifestItem#
37,38c53,54
< xmlns=video:http://www.video
< xmlns=xapG:http://ns.adobe.com/xap/1.0/g/
---
> xmlns=wav:http://ns.adobe.com/xmp/wav/1.0/
> xmlns=xml:http://www.w3.org/XML/1998/namespace
40a57
> xmlns=xmpG:http://ns.adobe.com/xap/1.0/g/
44a62
> xmlns=xmpNote:http://ns.adobe.com/xmp/note/
45a64
> xmlns=xmpT:http://ns.adobe.com/xap/1.0/t/

```

#8 - 08 Sep 2015 13:14 - Robin Mills

- % Done changed from 0 to 90

- Estimated time set to 10.00 h

As always, Andreas, you have very good ideas. I've put the public API into XmpProperties::registeredNamespaces(Exiv2::Dictionary&).

I have "demoted" the API XmpParser::registeredNamespaces() to be private and made XmpProperties a *friend class* of XmpParser. This enables XmpProperties::registeredNamespaces to call XmpParser::registerNs before calling XmpParser::registeredNamespaces(). So the sync between Exiv2 and XMPsdk is achieved seamlessly. The output of exiv2 correctly shows libexiv2 namespaces such as video and audio. XMPsdk namespaces such as png are also listed.

```

510 rmills@rmillsmbp:~/gnu/exiv2/trunk $ exiv2 -vV -g audio -g video -g png
exiv2 0.25 001900 (64 bit build)
enable_video=0
xmlns=audio:http://www.audio
xmlns=png:http://ns.adobe.com/png/1.0/
xmlns=video:http://www.video
511 rmills@rmillsmbp:~/gnu/exiv2/trunk $

```

I have update the code in src/version.cpp and samples/exiv2json.cpp to use XmpProperties::registeredNamespace. exiv2json now correctly lists all namespaces in use.

```

516 rmills@rmillsmbp:~/gnu/exiv2/trunk $ bin/exiv2json -x http://dev.exiv2.org/attachments/download/805/DSC_71
54.jpg
{
  "Xmp": {
    "xmp": {
      "Rating": "0",
      "ModifyDate": "2015-07-16T20:25:28+01:00"
    },
    "dc": {
      "description": {
        "lang": {
          "x-default": "Classic View"
        }
      }
    },
    "xmlns": {
      "dc": "http://purl.org/dc/elements/1.1/",
      "xmp": "http://ns.adobe.com/xap/1.0/"
    }
  }
}
517 rmills@rmillsmbp:~/gnu/exiv2/trunk $

```

#9 - 08 Sep 2015 14:38 - Alan Pater

With these changes we are also listing XMP SDK namespaces that are not supported by exiv2.

Is that not a separate feature then listing namespaces known to exiv2?

#10 - 08 Sep 2015 17:07 - Robin Mills

Is there such a thing as an XMP SDK namespace that is not supported by exiv2?

If we register the namespace (in this case cm), we can use it. If XMP SDK knows it (xmpBJ), we can use it.

```

$ curl --silent -O http://dev.exiv2.org/attachments/download/805/DSC_7154.jpg
$ exiv2 -M"reg cm http://clanmills.com/exiv2/1.0" -M"set Xmp.cm.title robin" -M"set Xmp.xmpBJ.bubble jet" DSC_7154.jpg
$ bin/exiv2json -x DSC_7154.jpg
{
  "Xmp": {
    "xmp": {
      "Rating": "0",
      "ModifyDate": "2015-07-16T20:25:28+01:00"
    },
    "cm": {
      "title": "robin"
    },
    "xmpBJ": {
      "bubble": "jet"
    },
    "dc": {
      "description": {
        "lang": {
          "x-default": "Classic View"
        }
      }
    },
    "xmlns": {
      "cm": "http://clanmills.com/exiv2/1.0/",
      "dc": "http://purl.org/dc/elements/1.1/",
      "xmp": "http://ns.adobe.com/xap/1.0/",
      "xmpBJ": "http://ns.adobe.com/xap/1.0/bj/"
    }
  }
}
$

```

If neither XMPsdk nor Exiv2 knows the namespace, we throw an exception.

```

540 rmills@rmillsmbp:~/gnu/exiv2 $ exiv2 -M"set Xmp.alan.title Pater" DSC_7154.jpg
-M option 1: Invalid key `Xmp.alan.title'
exiv2: Error parsing -M option arguments
Usage: exiv2 [ options ] [ action ] file ...

```

Manipulate the Exif metadata of images.
\$

If we register, the namespace, all is good.

```

$ exiv2 -M"reg alan http://clanmills.com/alan/1.0" -M"set Xmp.alan.title Pater" DSC_7154.jpg
$

```

Everything has been stored nicely in DSC_7154.jpg.

```

$ bin/exiv2json -x DSC_7154.jpg
{
  "Xmp": {
    "xmp": {
      "Rating": "0",
      "ModifyDate": "2015-07-16T20:25:28+01:00"
    },
    "cm": {
      "title": "robin"
    },
    "xmpBJ": {
      "bubble": "jet"
    },
    "alan": {
      "title": "Pater"
    },
    "dc": {
      "description": {
        "lang": {
          "x-default": "Classic View"
        }
      }
    },
    "xmlns": {
      "alan": "http://clanmills.com/alan/1.0/",

```

```

    "cm": "http://clanmills.com/exiv2/1.0/",
    "dc": "http://purl.org/dc/elements/1.1/",
    "xmp": "http://ns.adobe.com/xap/1.0/",
    "xmpBJ": "http://ns.adobe.com/xap/1.0/bj/"
  }
}
}
$

```

The code in `SXMPMeta::DumpNamespaces` (which is called by `XmpProperties::registeredNamespaces`) guarantees every ns/uri pair we define is either unique to Exiv2 or has the same definition as XMPsdk. This is discussed in [#751](#).

However, we seem to have namespaces in exiv2 which are not correctly supported and I suspect this is because exiv2 has not correctly registered all its namespaces!

```

544 rmills@rmillsmbp:~/gnu/exiv2 $ exiv2 -M"set Xmp.video.def abc" DSC_7154.jpg
Error: XMP Toolkit error 101: Unregistered schema namespace URI
Error: Failed to encode XMP metadata.
545 rmills@rmillsmbp:~/gnu/exiv2 $ exiv2 -vV -g video
exiv2 0.25 001900 (64 bit build)
enable_video=0
xmlns=video:http://www.video
546 rmills@rmillsmbp:~/gnu/exiv2 $

```

Should work, but doesn't.

```

$ 548 rmills@rmillsmbp:~/gnu/exiv2 $ trunk/bin/exiv2json DSC_7154.jpg
Caught Exiv2 exception 'XMP Toolkit error 9: Fatal namespace map problem'
$

```

Let's register and use it:

```

$ exiv2 -M"reg video http://www.video" -M"set Xmp.video.def abc" DSC_7154.jpg
$

```

He says he's happy. However he's not.

```

$ bin/exiv2json -x DSC_7154.jpg
Caught Exiv2 exception 'XMP Toolkit error 9: Fatal namespace map problem'
$

```

The XMP is good:

```

$ exiv2 -px DSC_7154.jpg
Xmp.xmp.Rating                XmpText      1  0
Xmp.xmp.ModifyDate           XmpText      25 2015-07-16T20:25:28+01:00
Xmp.video.def                 XmpText      3  abc
Xmp.dc.description           LangAlt      1  lang="x-default" Classic View
$

```

I don't think this is serious. The library isn't registering namespaces with XMPsdk correctly. I think we need to open a different issue to investigate this finding concerning 'video' as we've wandered way off topic.

#11 - 08 Sep 2015 17:48 - Robin Mills

I've opened a new issue [#1116](#) to investigate my discoveries about namespace 'video' above.

#12 - 08 Sep 2015 20:22 - Alan Pater

Robin Mills wrote:

Is there such a thing as an XMPsdk namespace that is not supported by exiv2?

If we register the namespace (in this case cm), we can use it.

Okay, bad wording on my part. I should have said that we are listing namespaces that are not **known** to exiv2.

I think the original request is for a way to get a list of namespaces that exiv2 already knows about.

#13 - 09 Sep 2015 04:11 - Andreas Huggel

I think the namespaces known by Exiv2 and its own internal XMPSDK should be in sync.

Copied from <http://dev.exiv2.org/issues/1116#note-3>: I wanted to fix that: those namespaces that Exiv2 knows and XMP-SDK doesn't just need to be

registered there (I think), but those that XMP-SDK has and Exiv2 doesn't, will require a bit more thought and work. For the current implementation of `XmpProperties::registerNamespaces()`, registering the missing namespaces in XMP-SDK in `XmpParser::initialize()` should suffice.

#14 - 09 Sep 2015 10:12 - Robin Mills

- *Status changed from Feedback to Resolved*

- *Assignee changed from Andreas Huggel to Robin Mills*

- *% Done changed from 90 to 100*

Done. I'll mark this Resolved/100%. If nothing else raises, it will be marked 'Closed' prior to shipping v0.26.

#15 - 08 Nov 2015 18:40 - Robin Mills

- *Subject changed from method is missing in Exiv2 to get list of know XMP namespace to method is missing in Exiv2 to get list of known XMP namespaces*

- *Status changed from Resolved to Closed*

I'm going to set the status of this to closed. It has been 100% resolved for some time without further incident.