Is there life after libXSLT 1?

Tony Graham
Mentea
13 Kelly’s Bay Beach
Skerries, Co Dublin, Ireland
info@mentea.net
http://www.mentea.net

Version 1.0. – 18 September 2013
© 2013 Mentea
Is there life after libXSLT 1?
What is libXSLT?

- XSLT 1.0 processor
- Coupled with libXML2 parser
- Written in C
- By Daniel Veillard of RedHat
- Bindings to other languages
- Used everywhere except Windows and Java systems

Binaries and bindings

- Binaries
  - Linux
  - Solaris
  - MacOS
  - Windows
- Bindings
  - Perl
  - PHP
  - Python
  - Ruby
  - Tcl

What’s the problem?

- libXSLT and libXML2 are everywhere
- Stuck at XSLT 1.0
- Life would be better if usable XSLT 2.0 processor replacement
  - Even if subset of XSLT 2.0

What do users want?

“What I am looking for is not full blown XSLT 2.0, but rather XSLT 1.5: various string functions support, regular expressions, xsl:attribute/@select, xsl:for-each-group, xpath1 except xpath2, xpath1=(value1, value2), tunnelled params, etc. I do not use XSLT 2.0 sequence generators, collations or strong types in my code”

https://mail.gnome.org/archives/xslt/2012-February/msg00002.html

What do we need?

- XSLT 2.0
- Full or subset
- Compatible with libXML2/libXSLT APIs
Why do we need compatibility?

- Current code “just works”
- Easy transition to XSLT 2.0
- Multiple ways to be compatible
  - Read and write libXML2 trees
  - Implement high-level libXSLT API for running a transform
  - Implement full libXSLT API

What’s the problem?

- APIs expose internals

```c
struct _xmlNode {
    void * _private : application data
    xmlElementType type : type number, must be second!
    const xmlChar * name : the name of the node, or the entity
    struct _xmlNode * children : parent->childs link
    struct _xmlNode * last : last child link
    struct _xmlNode * parent : child->parent link
    struct _xmlNode * next : next sibling link
    struct _xmlNode * prev : previous sibling link
    struct _xmlDoc * doc : the containing document End of common p
    xmlNs * ns : pointer to the associated namespace
    xmlChar * content : the content
    struct _xmlAttr * properties : properties list
    xmlNs * nsDef : namespace definitions on this node
    void * psvi : for type/PSVI informations
    unsigned short line : line number
    unsigned short extra : extra data for XPath/XSLT
};
```

If not libXSLT 1.0, then what?

- DIY libXSLT 2.0
- libx
- Qt XSLT processor
- Saxon-C
- EXSLT

Do-It-Yourself libXSLT 2.0

- A "lot" of work
libx


- Steve Ball, Explain
- Commercial/open source XSLT 2.0 based on libXSLT
- Pay for early access to XSLT 2.0 code
- Code to be released to open source after 6 months
- Not much take up
- Stalled since 2011

libx features

- Functions operating on QNames and Sequences
- if then else
- for
- Quantified expressions
- value and general comparisons
- Sequences
- xsl:function
- xsl:for-each-group

libx risks

- libXSLT not designed for XSLT 2.0
- No analysis/optimisation phase when compiling stylesheets

Qt XSLT processor

- C++, open source
- Offshoot of Qt XQuery processor
- XPath 2.0 + some XSLT 2.0
- 42% conformance in W3C test suite
**Qt XSLT conformance**

<table>
<thead>
<tr>
<th>XSL Feature</th>
<th>Support Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPath Conformance</td>
<td>Since XPath is a subset of XSLT, its issues are in affect too.</td>
</tr>
<tr>
<td>xsl:copy</td>
<td>The copy-namespaces and inherit-namespaces attributes have no effect. For copied comments, attributes and processing instructions, the copy has the same node identity as the original.</td>
</tr>
<tr>
<td>Patterns</td>
<td>Complex patterns or patterns with predicates have issues.</td>
</tr>
<tr>
<td>2.0 Compatibility Mode</td>
<td>Stylesheets are interpreted as XSLT 2.0 stylesheets, even if the version attribute is in the XSLT source is 1.0. In other words, the version attribute is ignored.</td>
</tr>
<tr>
<td>Grouping</td>
<td>fn:current-group(), fn:grouping-key() and xsl:for-each-group.</td>
</tr>
<tr>
<td>Date &amp; Time</td>
<td>fn:format-dateTime(), fn:format-date() and fn:format-time().</td>
</tr>
<tr>
<td>xsl:copy-of</td>
<td>The copy-namespaces attribute has no effect.</td>
</tr>
</tbody>
</table>


**Qt XSLT processor risks**

- Unknown, unused?
  - Question on Qt forum about linking libXSLT with Qt program
- No progress between Qt 4.5 and Qt 5.0?
- Future of Qt when Microsoft owns Nokia?

**Saxon-C**

- Saxon-CE = Saxon compiled to JavaScript
- Saxon.Net = Saxon compiled to .Net
- Saxon-C = Saxon-CE compiled to C/native code?

**Saxon-C risks**

- Saxonica mightn't like it?
- Unknown amount of work
**Exelt**

- F#, XSLT 3.0
- Closed source, free for non-commercial use
- Abrasoft
- In private beta
- Windows
- Said to be available for Linux with Mono libraries

**Exelt risks**

- Really unknown
- Closed source and license would limit acceptance
- Bindings from F#/Mono may be harder?

**Summary**

- Upgrade from libXSLT 1.0 hindering XSLT 2.0 use
- No obvious successor
- XSLT 2.0 subset would be okay?
- Anything not based on libXSLT 1.0 needs to fake being libXML/libXSLT