A DIFFERENT VIEW ON XML

SML – XML MADE SIMPLE

Version 2018-02-10

WHO I AM

- Jean-François Larvoire, aka. JF, or Jeff
- Now a technical leader at HPE, in Grenoble, France
- System software development since 1985 at HP then HPE, in Grenoble, and Roseville and Sunnyvale in California.
- Worked on speech recognition; PC BIOS; Windows drivers; Boot loaders; HPC file systems; Performance testing; Storage management.
- Many contributions to open source software.
- Contact: jf.larvoire@hpe.com

Hewlett Packard Enterprise

AGENDA

- Who I am
- XML readability issue
- Alternatives to XML, and why I don't want them
- Searching for a simpler way to present XML itself
- The SML proposal
- Comparisons with alternatives
- The sml.tcl conversion script
- SML support in libxml2. The x2s.exe program.
- Effect on data size; On mixed data
- Questions?

XML IS SIMPLE, ISN'T IT?

- Goal #6 in the XML specification was: XML documents should be human-legible and reasonably clear.
- The XML specification is several thousand lines long.
- Writing a full-featured XML I/O library is a huge work…
- In practice, small pieces of XML limited to a few lines are indeed simple to read by humans.
- In practice, large XML files are obscured by lots of markup.
  ⇒ Very difficult to read by humans

XML FILES THAT DROVE ME CRAZY (1/2)

```
<?xml version="1.0" ?>
<cib admin_epoch="0" epoch="0" num_updates="0">
  <configuration>
    <crm_config>
      <cluster_property_set id="cib-bootstrap-options">
        <attributes>
          <nvpair id="cib-bootstrap-options-symmetric-cluster" name="symmetric-cluster" value="true"/>
          <nvpair id="cib-bootstrap-options-no-quorum-policy" name="no-quorum-policy" value="stop"/>
          <nvpair id="cib-bootstrap-options-default-resource-stickiness" name="default-resource-stickiness" value="0"/>
          <nvpair id="cib-bootstrap-options-default-resource-failure-stickiness" name="default-resource-failure-stickiness" value="0"/>
          <nvpair id="cib-bootstrap-options-stonith-enabled" name="stonith-enabled" value="true"/>
          <nvpair id="cib-bootstrap-options-stonith-action" name="stonith-action" value="reboot"/>
          <nvpair id="cib-bootstrap-options-startup-fencing" name="startup-fencing" value="true"/>
          <nvpair id="cib-bootstrap-options-stop-orphan-resources" name="stop-orphan-resources" value="true"/>
          <nvpair id="cib-bootstrap-options-stop-orphan-actions" name="stop-orphan-actions" value="true"/>
          <nvpair id="cib-bootstrap-options-remove-after-stop" name="remove-after-stop" value="false"/>
          <nvpair id="cib-bootstrap-options-short-resource-names" name="short-resource-names" value="true"/>
          <nvpair id="cib-bootstrap-options-transition-idle-timeout" name="transition-idle-timeout" value="5min"/>
          <nvpair id="cib-bootstrap-options-default-action-timeout" name="default-action-timeout" value="600s"/>
          <nvpair id="cib-bootstrap-options-is-managed-default" name="is-managed-default" value="true"/>
          <nvpair id="cib-bootstrap-options-cluster-delay" name="cluster-delay" value="60s"/>
          <nvpair id="cib-bootstrap-options-pe-error-series-max" name="pe-error-series-max" value="-1"/>
          <nvpair id="cib-bootstrap-options-pe-warn-series-max" name="pe-warn-series-max" value="-1"/>
          <nvpair id="cib-bootstrap-options-pe-input-series-max" name="pe-input-series-max" value="-1"/>
        </attributes>
      </cluster_property_set>
    </crm_config>
    <nodes/>
    <resources>
      <primitive class="ocf" id="ost1" provider="heartbeat" type="Filesystem">
        <operations>
          <op id="ost1_mon" interval="120s" name="monitor" timeout="60s"/>
        </operations>
        <instance_attributes id="ost1_inst_attr">
          <attributes>
            <nvpair id="ost1_attr_0" name="device" value="/etc/sfs/luns/lun8"/>
            <nvpair id="ost1_attr_1" name="directory" value="/mnt/ost1"/>
            <nvpair id="ost1_attr_2" name="fstype" value="lustre"/>
          </attributes>
        </instance_attributes>
      </primitive>
      <primitive class="ocf" id="ost2" provider="heartbeat" type="Filesystem">
        <operations>
          <op id="ost2_mon" interval="120s" name="monitor" timeout="60s"/>
        </operations>
        <instance_attributes id="ost2_inst_attr">
          <attributes>
            <nvpair id="ost2_attr_0" name="device" value="/etc/sfs/luns/lun10"/>
            <nvpair id="ost2_attr_1" name="directory" value="/mnt/ost2"/>
            <nvpair id="ost2_attr_2" name="fstype" value="lustre"/>
          </attributes>
        </instance_attributes>
      </primitive>
```

XML FILES THAT DROVE ME CRAZY (2/2)

```
    <clone id="stonith_quincy3">
      <instance_attributes id="stonith_quincy3_inst_attr">
        <attributes>
          <nvpair id="stonith_quincy3_attr_1" name="clone_max" value="2"/>
          <nvpair id="stonith_quincy3_attr_2" name="clone_node_max" value="1"/>
        </attributes>
      </instance_attributes>
      <primitive class="stonith" id="stonith_hb_quincy3" provider="heartbeat" type="external/riloe">
        <operations>
          <op id="stonith_hb_quincy3_mon" interval="30s" name="monitor" prereq="nothing" timeout="20s"/>
          <op id="stonith_hb_quincy3_start" name="start" prereq="nothing" timeout="20s"/>
        </operations>
        <instance_attributes id="stonith_hb_quincy3_inst_attr">
          <attributes>
            <nvpair id="stonith_hb_quincy3_attr_2" name="hostlist" value="quincy3"/>
            <nvpair id="stonith_hb_quincy3_attr_3" name="ilo_hostname" value="192.168.16.153"/>
            <nvpair id="stonith_hb_quincy3_attr_4" name="ilo_user" value="jimi"/>
            <nvpair id="stonith_hb_quincy3_attr_5" name="ilo_password" value="secret:-)"/>
            <nvpair id="stonith_hb_quincy3_attr_6" name="ilo_can_reset" value="1"/>
            <nvpair id="stonith_hb_quincy3_attr_7" name="ilo_protocol" value="2.0"/>
            <nvpair id="stonith_hb_quincy3_attr_8" name="ilo_powerdown_method" value="off"/>
          </attributes>
        </instance_attributes>
      </primitive>
    </clone>
    <clone id="stonith_quincy4">
      <instance_attributes id="stonith_quincy4_inst_attr">
        <attributes>
          <nvpair id="stonith_quincy4_attr_1" name="clone_max" value="2"/>
          <nvpair id="stonith_quincy4_attr_2" name="clone_node_max" value="1"/>
        </attributes>
      </instance_attributes>
      <primitive class="stonith" id="stonith_hb_quincy4" provider="heartbeat" type="external/riloe">
        <operations>
          <op id="stonith_hb_quincy4_mon" interval="30s" name="monitor" prereq="nothing" timeout="20s"/>
          <op id="stonith_hb_quincy4_start" name="start" prereq="nothing" timeout="20s"/>
        </operations>
        <instance_attributes id="stonith_hb_quincy4_inst_attr">
          <attributes>
            <nvpair id="stonith_hb_quincy4_attr_2" name="hostlist" value="quincy4"/>
            <nvpair id="stonith_hb_quincy4_attr_3" name="ilo_hostname" value="192.168.16.154"/>
            <nvpair id="stonith_hb_quincy4_attr_4" name="ilo_user" value="jimi"/>
            <nvpair id="stonith_hb_quincy4_attr_5" name="ilo_password" value="secret:-)"/>
            <nvpair id="stonith_hb_quincy4_attr_6" name="ilo_can_reset" value="1"/>
            <nvpair id="stonith_hb_quincy4_attr_7" name="ilo_protocol" value="2.0"/>
            <nvpair id="stonith_hb_quincy4_attr_8" name="ilo_powerdown_method" value="off"/>
          </attributes>
        </instance_attributes>
      </primitive>
    </clone>
  </resources>
  <constraints>
    <rsc_location id="rsc_location_ost1" rsc="ost1">
      <rule id="prefered_location_ost1" score="100">
        <expression attribute="#uname" id="prefered_location_ost1_expr" operation="eq" value="quincy3"/>
      </rule>
    </rsc_location>
    <rsc_location id="rsc_location_ost2" rsc="ost2">
      <rule id="prefered_location_ost2" score="100">
        <expression attribute="#uname" id="prefered_location_ost2_expr" operation="eq" value="quincy4"/>
      </rule>
    </rsc_location>
  </constraints>
</configuration>
<status/>
</cib>
```

XML ALTERNATIVES

Facing the same difficulty, many others have tried alternative data formats, or proposed new ones.

- **Old ones. Ex: SGML, ASN.1, YAML**
- **JSON**
  Pros: Considerably more legible; Wide support
  Cons: Mixed data? Xpath?
- **Google Protocol Buffers**
- **New ones keep popping up regularly. Ex a few weeks ago:**
  Mark – A superset of JSON, with attributes à-la XML
  https://mark.js.org/

But I don't want an alternative, I want to access my XML data.

CONVERTING FROM/TO XML

- **ASN.1 XML XER - ASN.1 → XML Encoding Rules**
- **MicroXML - XML presented using a JSON syntax.**
  Pro: The closest in spirit to SML.
  Con: Longer than both SML and XML.
- **XSLT json-to-xml and xml-to-json - JSON presented using an XML syntax. (The inverse of MicroXML)**

But perfect compatibility with different data languages is impossible.

- **Missing features get lost, or must be emulated via complex encodings… which add dead weight even when not needed.**

LOOKING FOR A SIMPLER REPRESENTATION OF XML

- EXI - Efficient XML Interchange: A binary representation of XML… And proof that reversible conversion is possible.
- DOM
- Fundamental equivalence of all structured text trees, starting with XML data files and C/Java/PHP/Tcl programs.
- Is it possible to represent XML's DOM using a C-like syntax?
  - In a way 100% compatible with XML.
    (Allowing 100% fidelity back and forth conversions)
  - With as few changes as possible.
- Tcl has the simplest syntax in the {C-style languages}

THE SML SOLUTION

Basic principle:

- **XML: <tag attr="value" … >content</tag>**
- **SML: tag attr="value" … {content}**

+ Several rules for getting the simplest reversible content:

- **C-style "content text strings" with \\, \", etc, escaping; \ line continuation; ‘;’ for separating elements.**
- **Tcl-style simplifications, with multi-line strings; Implicit ‘;’ at the end of each line; Optional "" when no space nor any SML punctuation character in string. Optional {} when no sub-element in content.**

For the complete list with details, see the paper.

COMPARISON WITH XML

XML (from a Google Earth .kml file)	SML (generated by the sml script)

<?xml version="1.0" encoding="UTF-8"?>
<kml>
<Folder>
<name>Take off zones in the Alps</name>
<open>1</open>
<Folder>
<name>Drome</name>
<visibility>0</visibility>
<Placemark>
<description>Take off</description>
<name>Mont Rachas</name>
<LookAt>
<longitude>5.0116666667</longitude>
<latitude>44.8355</latitude>
<range>4000</range>
<tilt>45</tilt>
<heading>0</heading>
</LookAt>
</Placemark>
</Folder>
</Folder>
</kml>
	?xml version="1.0" encoding="UTF-8"
	kml {
	Folder {
	name "Take off zones in the Alps"
	open 1
	Folder {
	name Drome
	visibility 0
	Placemark {
	description "Take off"
	name "Mont Rachas"
	LookAt {
	longitude 5.0116666667
	latitude 44.8355
	range 4000
	tilt 45
	heading 0
	}
	}
	}
	}
	}

COMPARISON WITH MICROXML

SML (Converted from a G.E. .kml file)	MicroXML (hand made)

?xml version="1.0" encoding="UTF-8"
kml {
  Folder {
  name "Take off zones in the Alps"
  open 1
  Folder {
  name Drome
  visibility 0
  Placemark {
  description "Take off"
  name "Mont Rachas"
  LookAt {
  longitude 5.0116666667
  latitude 44.8355
  range 4000
  tilt 45
  heading 0
  }
  }
  }
  }
}
	["kml", {}, [
	  ["Folder", {}, [
	  ["name", {}, ["Take off zones in the Alps"]],
	  ["open", {}, ["1"]],
	  ["Folder", {}, [
	  ["name", {}, ["Drome"]],
	  ["visibility", {}, ["0"]],
	  ["Placemark", {}, [
	  ["description", {}, ["Take off"]],
	  ["name", {}, ["Mont Rachas"]],
	  ["LookAt", {}, [
	  ["longitude", {}, ["5.0116666667"]],
	  ["latitude", {}, ["44.8355"]],
	  ["range", {}, ["4000"]],
	  ["tilt", {}, ["45"]],
	  ["heading", {}, ["0"]]
	  ]]
	  ]]
	  ]]
	  ]]
	]]

COMPARISON WITH {MARK}

SML (Converted from a G.E. .kml file)	{mark} (hand made)

?xml version="1.0" encoding="UTF-8“
kml {
Folder {
name "Take off zones in the Alps"
open 1
Folder {
name Drome
visibility 0
Placemark {
description "Take off"
name "Mont Rachas"
LookAt {
longitude 5.0116666667
latitude 44.8355
range 4000
tilt 45
heading 0
}
}
}
}
}
	{kml
	{Folder
	{name "Take off zones in the Alps"}
	{open 1}
	{Folder
	{name "Drome"}
	{visibility 0}
	{Placemark
	{description "Take off"}
	{name "Mont Rachas"}
	{LookAt
	{longitude 5.0116666667}
	{latitude 44.8355}
	{range 4000}
	{tilt 45}
	{heading 0}
	}
	}
	}
	}
	}

THE SML.TCL CONVERSION SCRIPT

- Available in https://github.com/JFLarvoire/SysToolsLib: Enter the folder /Tcl and download the file sml.tcl.
- The README.md file in that /Tcl folder explains how to setup Tcl in Windows.
- Converts XML to SML, then SML back to XML without any change.
- I've been using it in many work projects for years.
- Tested successfully with all libxml2 test files.
- Limited to ASCII-based encodings and UTF-8.

Demo

SML SUPPORT IN THE LIBXML2 LIBRARY

- Available in https://github.com/JFLarvoire/libxml2
- Features still limited:
  - It can save DOM trees as SML. (Works well)
  - It can parse well formed SML. (Incomplete)
  - Heuristic to auto-detect if parser input is XML or SML.
  - All XML parsing and generation unchanged.
- No API changes, but a few new constants.
- A tiny program called x2s.c reads either XML or SML, and outputs the other kind.
- Conversion at least 20 times faster than with sml.tcl.
- BUT not binary reversible ☹ due to libxml2 limitations.

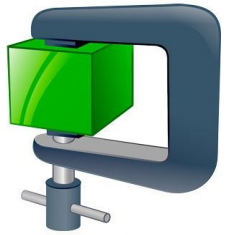
WHY NOT BINARY REVERSIBLE

- Libxml2 does not record non-significant white spaces.
- For example, starting with:
  <tag␣␣␣attr="value"␣␣␣>␣␣␣data␣␣␣</tag␣␣␣>
- sml.tcl generates:
  tag␣␣␣attr="value"␣␣␣{"␣␣␣data␣␣␣"}{␣␣␣}
  Then the second run generates:
  <tag␣␣␣attr="value"␣␣␣>␣␣␣data␣␣␣</tag␣␣␣>
- x2s.exe generates canonic spaces:
  tag␣attr="value"␣ "␣␣␣data␣␣␣"
  Then the second run generates:
  <tag␣attr="value">␣␣␣data␣␣␣</tag>
- Idea: Create a new optional node type for preserving these blanks?

ISSUES WITH THE XMLWRITER APIS

- Cannot optimize SML by removing {} and/or "".
- Ex: A program calls routines xmlTextWriterStartElement(), xmlTextWriterWriteString(), xmlTextWriterEndElement(). xmlTextWriterStartElement() cannot predict whether the { can be optimized-out.
- Idea: Add a new function xmlTextWriterWriteStringElement(). Problem: This resolves just one case. We'd need MANY new functions to resolve all cases.
- Idea: Cache the partially built elements in a temporary DOM tree, then write that tree. This will work with no API change. But we lose the performance edge of the writer APIs.

SML FILES SIZES



- On a 1 MB set of XML test files, the total size of the converted files is 12% smaller than the original files.
- Among big files, that reduction goes from 4% for a file with lots of large CDATA elements, to 17% for a file with deeply nested elements.
- Even after zipping the two full sets of samples, the SML files archive is 2% smaller than the XML files archive.
  ⇒ Microsoft would gain 2% file size by using SML for MS Office .docx or .pptx documents. ☺

USING SML FOR DATA TRANSFERS

- The savings potential is much better for XML-based network protocols, such as SOAP. (Data trees with deeply nested elements)
- Adapting existing XML-based protocols to use SML instead is very easy, and increases bandwidth considerably.
- Last but not least, packet analysis is much easier!
  I've done it a lot in HPE cluster management scripts!

NEXT STEPS

- Experiment with the tools on your XML data, and give feedback about the syntax, and the possible alternatives.
  sml.tcl available in https://github.com/JFLarvoire/SysToolsLib
- I challenge you to finds holes in the SML definition, preventing a valid XML file to be converted back identically. If you find one, send me (jf.larvoire@hpe.com) a sample file to prove it.
  Bugs in the tools don't count. Report them on GitHub.
- If interest grows, work with interested people to formalize a real specification.
- Continue work to improve SML parsing and generation in libxml2. Anybody interested in participating? Or in any other library?
- Consider using for editing complex XML files.
- Consider using for efficiently storing and transmitting XML data.

THANK YOU.
QUESTIONS?

SPARE SLIDES

SML ELEMENTS

- Elements normally end at the end of the line.
- They continue on the next line if there's a trailing '\'.
- They also continue if there's an unmatched "quotes" or {braces} block.
- Multiple elements on the same line must be separated by a ';'.

Pros: A natural match for canonical XML files, which have one XML terminal element per line; Same as Tcl instructions.

Cons: ?

SML ATTRIBUTES

- **The syntax for attributes is the same as for XML. Including the rules for using quotes, avoiding '<' and '&', and escaping using entity chars.**

```
bird species='eagle' note="Found here &amp; there"
```

Pros: Conversion straightforward; Readable as it is.

Cons: Slightly different from Tcl's string quoting and escaping rules used for content data, which may be confusing?

SML CONTENT DATA

- **The content data are normally inside a {curly braces} block.**
  ```
  tag {content}
  ```
- **The content text is between "quotes".**
  Escape '\' and '"' with a '\'.
  ```
  tag {"Some text with an \" and";{an inner element} }
  ```
- **If there are no further child elements embedded in contents (i.e. it's only text), the braces can be omitted.**
  ```
  tag "Some text with an \" but no inner element"
  ```
- **Furthermore, if the text does not contain blanks, '"', '=', ';', '#', '{', '}', '<', '>', nor a trailing '\', the quotes around the text can be omitted too.**
  (ie. It cannot be confused with an attribute or a comment or any kind of SML markup.)
  ```
  tag 3.1415926535
  ```

Pros: Removes as much syntactic clutter as possible, just like Tcl.

OTHER TYPES OF SML MARKUP

- This is a `?Processing instruction` .
  (The final '?' in XML is removed in SML.)
- This is a `!Declaration` . (Ex: a `!doctype definition`)
- This is a `#-- Comment block, ending with two dashes --` .
- Simplified case for a `# One-line comment .`
- This is a `<[[ Cdata section ]]>`. Initial \n discarded.

Pros: Conversion straightforward; Minimizes clutter.

Cons: I wish I had found something with just {}. Ex:

```
{
  An indented Cdata section
}
```

SML SCRIPT STATUS

- About 3000 lines of Tcl, half of which are an independent debugging library.
- Performance: It converts about 10 KB/s of data on a 2 GHz machine. => Should be rewritten in C if high perf needed.
- Known limitations:
  - The converted files use the local operating system line endings.
  - Only supports ASCII-compatible encodings, including UTF-8. No UTF-16 nor EBDCIC.

THE SHOW SCRIPT

- Displays a file tree as SML.
- Each file or directory is an SML element.
- Directories contain inner elements that represent files and subdirectories.
- File contents are displayed as text if possible, else are dumped in hexadecimal.
- Several modes of operation, with standard or experimental SML, and a simplified output mode that's most readable.

Demo

THE SPATH VIRTUAL SCRIPT

- A thought experiment that gives some insight on the power of the SML concept.
- The xpath script for extracting XML data using XPATH
- Writing a full-featured script able to use XPATH to extract data from SML files would be very difficult.
- Yet this can be done in a single line of code:
  ```
  sml | xpath %*
  ```
- Powerful in combination with the show script

Demo

COMPARISON WITH YAML

SML (Converted from a G.E. .kml file)	YAML (Without lists)

?xml version="1.0" encoding="UTF-8"
kml {
  Folder {
  name "Take off zones in the Alps"
  open 1
  Folder {
  name Drome
  visibility 0
  Placemark {
  description "Take off"
  name "Mont Rachas"
  LookAt {
  longitude 5.0116666667
  latitude 44.8355
  range 4000
  tilt 45
  heading 0
  }
  }
  }
  }
}
	--- 	
	kml:
	  Folder:
	  name: Take off zones in the Alps
	  open: 1
	  Folder:
	  name: Drome
	  visibility: 0
	  Placemark:
	  description: Take off
	  name: Mont Rachas
	  LookAt:
	  longitude: 5.0116666667
	  latitude: 44.8355
	  range: 4000
	  tilt: 45
	  heading: 0

COMPARISON WITH YAML

SML (Converted from a G.E. .kml file)	YAML (Converted from MicroXML)

```
?xml version="1.0" encoding="UTF-8"
kml {
  Folder {
    name "Take off zones in the Alps"
    open 1
    Folder {
      name Drome
      visibility 0
      Placemark {
        description "Take off"
        name "Mont Rachas"
        LookAt {
          longitude 5.0116666667
          latitude 44.8355
          range 4000
          tilt 45
          heading 0
        }
      }
    }
  }
}
```

```
---
- kml
- {}
- - - Folder
    - {}
    - - - name
        - {}
        - - Take off zones in the Alps
      - - open
        - {}
        - - '1'
      - - Folder
        - {}
        - - - name
            - {}
            - - Drome
          - - visibility
            - {}
            - - '0'
          - - Placemark
            - {}
            - - - description
[…]
```