Network Working Group S. Perreault
Internet-Draft
Viagenie
Intended status: Standards Track March 8, 2010
Expires: September 9, 2010

vCard XML Representation
draft-ietf-vcarddav-vcardxml-02

Abstract

This document defines the XML schema of the vCard data format.

Status of This Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 9, 2010.

Copyright Notice

Table of Contents

1. Introduction

vCard [I-D.ietf-vcarddav-vcardrev] is a data format for representing and exchanging information about individuals. It is a text-based format (as opposed to a binary format). This document defines an XML representation for vCard. The underlying data structure is exactly the same, enabling a 1-to-1 mapping between the original vCard format and the XML representation. The XML formatting may be preferred in some contexts where an XML engine is readily available and may be reused instead of writing a stand-alone vCard parser.

Earlier work on an XML format for vCard was started in 1998 by Frank Dawson [I-D.dawson-vcard-xml-dtd]. Sadly it did not take over the world.

2. The Schema

The schema is expressed in the RELAX NG language [relaxng][relaxng-compact] and is found in Appendix A.

3. Example: Author's XML vCard

```
<?xml version="1.0" encoding="UTF-8"?>
<vcards xmlns="urn:ietf:params:xml:ns:vcard-4.0">
  <vcard>
    <fn><text>Simon Perreault</text></fn>
    <n>
      <surname><text>Perreault</text></surname>
      <given><text>Simon</text></given>
      <prefix><text/></prefix>
      <suffix>
        <text>ing. jr.</text>
        <text>M.Sc.</text>
      </suffix>
    </n>
    <bday><date>--0203</date></bday>
    <anniversary>
      <date-time>20090808T1430-0500</date-time>
    </anniversary>
    <sex>1</sex>
    <lang>
      <parameters><pref>1</pref></parameters>
      <language-tag>fr</language-tag>
    </lang>
    <lang>
      <parameters><pref>2</pref></parameters>
      <language-tag>en</language-tag>
    </lang>
```

```
      <org>
        <parameters><type>work</type></parameters>
        <text>Viagenie</text>
      </org>
      <adr>
        <parameters><type>work</type></parameters>
        <pobox><text/></pobox>
        <ext><text>Suite 625</text></ext>
        <street><text>2600 boul. Laurier</text></street>
        <locality><text>Quebec</text></locality>
        <region><text>QC</text></region>
        <code><text>G1V 4W1</text></code>
        <country><text>Canada</text></country>
      </adr>
      <tel>
        <parameters>
          <type>work</type>
          <type>voice</type>
        </parameters>
        <uri>tel:+1-418-656-9254;ext=102</uri>
      </tel>
      <tel>
        <parameters>
          <type>work</type>
          <type>text</type>
          <type>voice</type>
          <type>cell</type>
          <type>video</type>
        </parameters>
        <uri>tel:+1-418-262-6501</uri>
      </tel>
      <tel>
        <parameters>
          <type>work</type>
          <type>fax</type>
        </parameters>
        <uri>tel:+1-418-656-9257</uri>
      </tel>
      <email>
        <parameters><type>work</type></parameters>
        <text>simon.perreault@viagenie.ca</text>
      </email>
      <geo>
        <parameters><type>work</type></parameters>
        <uri>geo:46.772673,-71.282945</uri>
      </geo>
      <key>
        <parameters><type><work/></type></parameters>
```

```
        <uri>http://www.viagenie.ca/simon.perreault/simon.asc</uri>
      </key>
      <tz><text>America/Montreal</text></tz>
      <class>PUBLIC</class>
    </vcard>
  </vcards>
```

4. Design Considerations

The general idea is to map vCard parameters, properties, and value types to XML elements. For example, the "FN" property is mapped to the "fn" element. That element in turn contains a text element whose content corresponds to the vCard property's value.

vCard parameters are also mapped to XML elements. They are contained in the <parameters> element, which is contained in property elements. For example, the "TYPE" parameter applied to the "TEL" property would look like the following in XML:

```
  <tel>
    <parameters>
      <type>voice</type>
      <type>video</type>
    </parameters>
    <uri>tel:+1-555-555-555</uri>
  </tel>
```

Parameters taking a list of values are simply repeated multiple times, once for each value in the list.

Properties having structured values (e.g. the "N" property) are expressed by XML element trees. Element names in that tree (e.g. "surname", "given", etc.) do not have a vCard equivalent since they are identified by position in plain vCard.

Line folding is a non-issue in XML. Therefore, the mapping from vCard to XML is done after the unfolding procedure is carried out. Conversely, the mapping from XML to vCard is done before the folding procedure is carried out.

4.1. Extensibility

The original vCard format is extensible. New properties, parameters, data types and values (collectively known as vCard objects) can be registered from IANA. It is expected that these vCard extensions will also specify extensions to the XML format described in this

document. This is not a requirement: a separate document may be used

instead.

Unregistered extensions (i.e. those starting with "X-" and "VND-...-") can be expressed in XML by making use of XML namespaces. They simply correspond to elements outside of the core namespace. For example:

```
  <ext:my-prop
      ext:xmlns="http://example.com/extensions/my-vcard">
    <parameters>
      <pref>1</pref>
    </parameters>                <!-- Core vCard elements  -->
    <text>value goes here</text>  <!-- are still accessible -->
  </ext:my-prop>
```

Note that extension elements do not need the "X- or "VND-" prefix in XML. The XML namespace mechanism is sufficient.

A vCard XML parser MUST ignore XML elements and attributes for which it doesn't recognize the expanded name. The normal behaviour of ignoring XML processing instructions whose target is not recognized MUST also be followed.

In the original vCard format, the "VERSION" property was mandatory and played a role in extensibility. In XML, this property is absent.
Its role is played by the vCard core namespace identifier, which includes the version number. vCard revisions will use a different namespace.

Parameters containing a list of values are expressed using a list of elements in XML (e.g. the <type> element).

4.2. Limitations

The schema does not validate the cardinality of properties. This is a limitation of the schema definition language. Cardinalities of the
original vCard format [I-D.ietf-vcarddav-vcardrev] MUST still be respected.

Some constructs (e.g. value enumerations in type parameters) have additional ordering constraints in XML. This is a result of limitations of the schema definition language and the order is arbitrary. The order MUST be respected in XML for the vCard to be valid. However, reordering as part of conversion to or from plain vCard MAY happen.

5. Security Considerations

All the security considerations applicable to plain vCard [I-D.ietf-vcarddav-vcardrev] are applicable to this document as well.

6. IANA Considerations

6.1. Registration of the XML Namespace

URI: urn:ietf:params:xml:ns:vcard-4.0

Registrant Contact: Simon Perreault <simon.perreault@viagenie.ca>

XML: None. Namespace URIs do not represent an XML specification.

7. Acknowledgements

Thanks to the following people for their input:

Alexey Melnikov, Barry Leiba, Cyrus Daboo, Joe Hildebrand, Joseph Smarr, Marc Blanchet, Peter Saint-Andre, Robins George, Zahhar Kirillov, Zoltan Ordogh.

8. References

8.1. Normative References

[I-D.ietf-vcarddav-vcardrev] Perreault, S. and P. Resnick, "vCard Format Specification", draft-ietf-vcarddav-vcardrev-09 (work in progress), October 2009.

[relaxng] Clark, J., "RELAX NG Specification", December 2001.

[relaxng-compact] Clark, J., "RELAX NG Compact Syntax", November 2002, <http://www.relaxng.org/compact-20021121.html>.

8.2. Informative References

[I-D.dawson-vcard-xml-dtd] Dawson, F., "The vCard v3.0 XML DTD", draft-dawson-vcard-xml-dtd-03 (work in progress), June 1998.

Appendix A. Relax NG Schema

```
default namespace = "urn:ietf:params:xml:ns:vcard-4.0"
```

```
# Value types
value-text = element text { text }
value-text-list = value-text+
value-uri = element uri { xsd:anyURI }
value-date = element date {
    xsd:string { pattern = "\d{8}|\d{4}-\d\d|--\d\d(\d\d)?|---\d\d" }
  }
value-time = element time {
    xsd:string { pattern = "(\d\d(\d\d(\d\d)?)?|-\d\d(\d\d?)|--\d\d)"
                         ~ "(Z|[+\-]\d\d(\d\d)?)?" }
  }
value-date-time = element date-time {
    xsd:string { pattern = "(\d{8}|--\d{4}|---\d\d)T\d\d(\d\d(\d\d)?)?"
                         ~ "(Z|[+\-]\d\d(\d\d)?)?" }
  }
value-date-and-or-time = value-date | value-date-time | value-time
value-timestamp = element timestamp {
    xsd:string { pattern = "\d{8}T\d{6}(Z|[+\-]\d\d(\d\d)?)?" }
  }
value-boolean = element boolean { xsd:boolean }
value-integer = element integer { xsd:integer }
value-float = element float { xsd:float }
value-binary = element binary { xsd:base64Binary }
value-language-tag = element language-tag {
    xsd:string { pattern = "([a-z]{2,3}((-[a-z]{3}){0,3})?|[a-z]{4,8})"
                         ~ "(-[a-z]{4})?(-([a-z]{2}|\d{3}))?"
                         ~ "(-([0-9a-z]{5,8}|\d[0-9a-z]{3}))*"
                         ~ "(-[0-9a-wyz](-[0-9a-z]{2,8})+)*"
                         ~ "(-x(-[0-9a-z]{1,8})+)?|x(-[0-9a-z]{1,8})+|"
                         ~ "[a-z]{1,3}(-[0-9a-z]{2,8}){1,2}" }
  }

# Parameters
param-language = element language { value-language-tag }?
param-pref = element pref {
    xsd:integer { minInclusive = "1" maxInclusive = "100" }
  }?
param-pid = element pid {
    xsd:string { pattern = "\d+(\.\d+)?" }
  }?
param-type = element type { "work" | "home" }*
param-calscale = element calscale { "gregorian" }?

# Properties
property-source = element source {
    element parameters { param-pid, param-pref },
    value-uri
  }
```

```
property-name = element name { value-text }
property-kind = element kind { "individual" | "group" | "org"
                             | "location" | "thing" }*
property-fn = element fn {
    element parameters { param-language, param-pid,
                         param-pref, param-type }?,
    value-text
  }
property-n = element n {
    element parameters { param-language }?,
    element surname { value-text-list },
    element given { value-text-list },
    element prefix { value-text-list },
    element suffix { value-text-list }
  }
property-nickname = element nickname {
    element parameters { param-language, param-pid,
                         param-pref, param-type }?,
    value-text-list
  }
property-photo = element photo {
    element parameters {
      param-pid,
      param-pref,
      param-type,
      element type { element media { value-text } }?
    }?,
    (value-binary | value-uri)
  }
property-bday = element bday {
    element parameters { param-calscale }?,
    (value-date-and-or-time | value-text)
  }
property-dday = element dday {
    element parameters { param-calscale }?,
    (value-date-and-or-time | value-text)
  }
property-birth = element birth {
    element parameters { param-language }?,
    value-text
  }
property-death = element death {
    element parameters { param-language }?,
    value-text
  }
property-anniversary = element anniversary {
    element parameters { param-calscale }?,
    (value-date-and-or-time | value-text)
```

```
  }
property-sex = element sex { "0" | "1" | "2" | "9" }
property-adr = element adr {
    element parameters {
      param-language,
      param-pid,
      param-pref,
      param-type,
      element geo { value-uri }?,
      element tz { value-text | value-uri }?
    }?,
    element pobox { value-text-list },
    element ext { value-text-list },
    element street { value-text-list },
    element locality { value-text-list },
    element region { value-text-list },
    element code { value-text-list },
    element country { value-text-list }
  }
property-label = element label {
    element parameters { param-language, param-pid,
                         param-pref, param-type }?,
    value-text
  }
property-tel = element tel {
    element parameters {
      param-pid,
      param-pref,
      element type { "work" | "home" | "text" | "voice"
                   | "fax" | "cell" | "video" | "pager" }*
    },
    value-uri
  }
property-email = element email {
    element parameters { param-pid, param-pref, param-type }?,
    value-text
  }
property-impp = element impp {
    element parameters { param-pid, param-pref, param-type }?,
    value-uri
  }
property-lang = element lang {
    element parameters { param-pid, param-pref, param-type }?,
    value-language-tag
  }
property-tz = element tz {
    element parameters { param-pid, param-pref, param-type }?,
    (value-text | value-uri)
```

```
  }
property-geo = element geo {
    element parameters { param-pid, param-pref, param-type }?,
    value-uri
  }
property-title = element title {
    element parameters { param-language, param-pid,
                         param-pref, param-type }?,
    value-text
  }
property-role = element role {
    element parameters { param-language, param-pid,
                         param-pref, param-type }?,
    value-text
  }
property-logo = element logo {
    element parameters {
      param-language,
      param-pid,
      param-pref,
      param-type,
      element type { element media { value-text } }?
    }?,
    (value-binary | value-uri)
  }
property-org = element org {
    element parameters { param-language, param-pid,
                         param-pref, param-type }?,
    value-text-list
  }
property-member = element member { param-pid, param-pref, value-uri }
property-related = element related {
    element parameters {
      param-pid,
      param-pref,
      element type { "work" | "home" | "parent" | "child"
        | "sibling" | "spouse" | "family" | "friend"
        | "supervisor" | "supervisee" | "assistant"
        | "colleague" | "agent" | "emergency" }*
    }?,
    (value-uri | value-text)
  }
property-categories = element categories {
    element parameters { param-pid, param-pref, param-type }?,
    value-text
  }
property-note = element note {
    element parameters { param-language, param-pid,
```

```
                        param-pref, param-type }?,
    value-text
  }
property-prodid = element prodid { value-text }
property-rev = element rev { value-timestamp }
property-sort-string = element sort-string {
    element surname { value-text }+,
    element given { value-text }*
  }
property-sound = element sound {
    element parameters {
      param-language,
      param-pid,
      param-pref,
      element type {
        element work { empty }?,
        element home { empty }?,
        element media { value-text }?
      }
    }?,
    (value-binary | value-uri)
  }
property-uid = element uid { value-uri }
property-clientpidmap = element clientpidmap {
    element sourceid { xsd:positiveInteger },
    value-uri
  }
property-url = element url {
    element parameters { param-pid, param-pref, param-type }?,
    value-uri
  }
property-class = element class { "PUBLIC" | "PRIVATE" |
"CONFIDENTIAL" }
property-key = element key {
    element parameters {
      param-pid,
      param-pref,
      element type {
        element work { empty }?,
        element home { empty }?,
        element media { value-text }?
      }
    }?,
    (value-binary | value-uri)
  }
property-fburl = element fburl {
    element parameters { param-pid, param-pref, param-type }?,
    value-uri
  }
```

```
property-caladruri = element caladruri {
    element parameters { param-pid, param-pref, param-type }?,
    value-uri
  }
property-caluri = element caluri {
    element parameters { param-pid, param-pref, param-type }?,
    value-uri
  }

# Top-level grammar
property = property-adr | property-anniversary | property-bday
         | property-birth | property-caladruri | property-caluri
         | property-categories | property-class | property-clientpidmap
         | property-dday | property-death | property-email
         | property-fburl | property-fn | property-geo | property-impp
         | property-key | property-kind | property-label | property-
lang
         | property-logo | property-member | property-n | property-name
         | property-nickname | property-note | property-org
         | property-photo | property-prodid | property-related
         | property-rev | property-role | property-sex
         | property-sort-string | property-sound | property-source
         | property-tel | property-title | property-tz | property-uid
         | property-url
start = element vcards {
    element vcard {
      (property
       | element group {
           attribute name { text },
           property*
         })+
    }+
  }
```

Appendix B. Change Log (to be removed by RFC Editor prior to publication)

B.1. Changes in -02

- Synchronized with draft-ietf-vcarddav-vcardrev-10.
- Turned <type> parameter values into plain text.
- Moved the "XML" property to vCard base.
- Changed title to avoid confusion with XML Schema.

- Added prefixes "value-", "param-", and "property-" in schema.
- Better language for specifying what a parser must ignore.

B.2. Changes in -01

- Synchronized with draft-ietf-vcarddav-vcardrev-09.
- Added the <vcards> element to allow multiple vCards in a single XML file.
- Created the <parameters> container element.
- Use text value for enumeration in <class> element.
- Created the "XML" vCard property.
- Added IANA considerations section.
- Added security considerations section.

B.3. Changes in -00

- Same as draft-perreault-vcarddav-vcardxml-02.

Author's Address

Simon Perreault
Viagenie
2600 boul. Laurier, suite 625
Quebec, QC G1V 4W1
Canada

Phone: +1 418 656 9254
EMail: simon.perreault@viagenie.ca
URI: http://www.viagenie.ca