Internet Engineering Task Force (IETF)
Request for Comments: 6351
Category: Standards Track
ISSN: 2070-1721

S. Perreault
Viagenie

August 2011

# xCard: vCard XML Representation

## Abstract

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

## Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6351.

Table of Contents



## 1. Introduction

vCard [RFC6350] is a data format for representing and exchanging information about individuals and other entities. It is a text-based format (as opposed to a binary format). This document defines xCard, an XML [W3C.REC-xml-20081126] 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 standalone vCard parser.

Earlier work on an XML format for vCard was started in 1998 by Frank Dawson [VCARD-DTD]. Sadly, it did not take over the world.

## 2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. The Schema

The schema is expressed in the RELAX NG language [ISO.19757-2.2008] and is found in Appendix A.

## 4. 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>Perreault</surname>
      <given>Simon</given>
      <additional/>
      <prefix/>
      <suffix>ing. jr</suffix>
      <suffix>M.Sc.</suffix>
    </n>
    <bday><date>--0203</date></bday>
    <anniversary>
      <date-time>20090808T1430-0500</date-time>
    </anniversary>
    <gender><sex>M</sex></gender>
    <lang>
      <parameters><pref><integer>1</integer></pref></parameters>
      <language-tag>fr</language-tag>
    </lang>
    <lang>
      <parameters><pref><integer>2</integer></pref></parameters>
      <language-tag>en</language-tag>
    </lang>
    <org>
      <parameters><type><text>work</text></type></parameters>
      <text>Viagenie</text>
    </org>
    <adr>
      <parameters>
        <type><text>work</text></type>
        <label><text>Simon Perreault
2875 boul. Laurier, suite D2-630
Quebec, QC, Canada
G1V 2M2</text></label>
      </parameters>
      <pobox/>
      <ext/>
      <street>2875 boul. Laurier, suite D2-630</street>
      <locality>Quebec</locality>
      <region>QC</region>
      <code>G1V 2M2</code>
      <country>Canada</country>
    </adr>
    <tel>
```

```
        <parameters>
          <type>
            <text>work</text>
            <text>voice</text>
          </type>
        </parameters>
        <uri>tel:+1-418-656-9254;ext=102</uri>
      </tel>
      <tel>
        <parameters>
          <type>
            <text>work</text>
            <text>text</text>
            <text>voice</text>
            <text>cell</text>
            <text>video</text>
          </type>
        </parameters>
        <uri>tel:+1-418-262-6501</uri>
      </tel>
      <email>
        <parameters><type><text>work</text></type></parameters>
        <text>simon.perreault@viagenie.ca</text>
      </email>
      <geo>
        <parameters><type><text>work</text></type></parameters>
        <uri>geo:46.766336,-71.28955</uri>
      </geo>
      <key>
        <parameters><type><text>work</text></type></parameters>
        <uri>http://www.viagenie.ca/simon.perreault/simon.asc</uri>
      </key>
      <tz><text>America/Montreal</text></tz>
      <url>
        <parameters><type><text>home</text></type></parameters>
        <uri>http://nomis80.org</uri>
      </url>
    </vcard>
  </vcards>
```

## 5. 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. In turn, that element 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>
      <text>voice</text>
      <text>video</text>
    </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.

A top-level <vcards> element is used as root. It contains one or more <vcard> elements, each representing a complete vCard. The <vcards> element MUST be present even when only a single vCard is present in an XML document.

The group construct (Section 3.2 in [RFC6350]) is represented with the <group> element. The "name" attribute contains the group's name. For example:

```
<vcards>
  <vcard>
    <group name="contact">
      <fn>...</fn>
      <email>...</email>
    </group>
    <group name="media">
      <photo>...</photo>
    </group>
    <categories>...</categories>
  </vcard>
</vcards>
```

is equivalent to:

```
BEGIN:VCARD
VERSION:4.0
contact.FN=...
contact.EMAIL=...
media.PHOTO=...
CATEGORIES=...
END:VCARD
```

### 5.1. Extensibility

The original vCard format is extensible. New properties, parameters, data types and values (collectively known as vCard elements, not to be confused with XML elements) can be registered with IANA (see [RFC6350], Section 10.2). It is expected that these vCard extensions will also specify extensions to the XML format described in this document.

New XML vCard property and parameter element names MUST be lower-case. This is necessary to ensure that round-tripping between XML and plain-text vCard works correctly.

Unregistered extensions (i.e., those starting with "X-" and "VND-...-") are expressed in XML by using elements starting with "x-" and "vnd-...-". Usage of XML namespaces [W3C.REC-xml-names-20091208] for extensibility is RECOMMENDED for extensions that have no equivalent in plain-text vCard. Refer to Section 6 for the implications when converting between plain-text vCard and XML.

Examples:

```
<x-my-prop>
  <parameters>
    <pref><integer>1</integer></pref>
  </parameters>
  <text>value goes here</text>
</x-my-prop>

<ext:my-prop
    ext:xmlns="http://example.com/extensions/my-vcard">
  <parameters>
    <pref><integer>1</integer></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 behavior 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).

## 5.2. 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 [RFC6350] 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.

## 6. Format Conversions

When new properties or "X-" properties are used, a vCard<->xCard converter might not recognize them or know what the appropriate default value types are, yet they need to be able to preserve the values. A similar issue arises for unrecognized property parameters. As a result, the following rules are applied when dealing with unrecognized properties and property parameters:

- When converting from vCard to xCard:
  - Any property that does not include a "VALUE" parameter and whose default value type is not known MUST be converted using the value type XML element <unknown>. The content of that element is the unprocessed value text.
  - Any unrecognized property parameter MUST be converted using the value type XML element <unknown>, with its content set to the parameter value text, treated as if it were a text value, or list of text values.
  - The content of "XML" properties is copied as is to XML.
  - Property and parameter XML element names are converted to lower-case.
  - Property value escaping is undone. For example, "\n" becomes a NEWLINE character (ASCII decimal 10).
  - Double-quoting of parameter values, as well as backslash escaping in parameter values, is undone. For example, PARAM="\"foo\",\"bar\"" becomes <param>"foo","bar"</param>.
- When converting xCard to vCard:
  - Properties in the vCard 4 namespace:
    - If the converter knows of a specific plain-text representation for this property, it uses it. For example, the <adr> element corresponds to the "ADR" property, which is encoded using comma-separated lists separated by semicolons.
    - Otherwise, the property name is taken from the element name, property parameters are taken from the <parameters> element, and the content of the property is taken from the content of the value element. If the property element has attributes or contains other XML elements, they are dropped.

+ If a standard property's XML element contains XML elements and attributes for which the converter doesn't recognize the expanded name, they are dropped. Therefore, it is RECOMMENDED to limit extensions to the property level to ensure that all data is preserved intact in round-trip conversions.

* Properties in other namespaces are wrapped as is inside an "XML" property.

* Any <unknown> property value XML elements are converted directly into vCard values. The containing property MUST NOT have a "VALUE" parameter.

* Any <unknown> parameter value XML elements are converted as if they were <text> value type XML elements.

* Property and parameter names are converted to upper-case.

* Property value escaping (Section 3.3 of [RFC6350]) is carried out. For example, a NEWLINE character (ASCII decimal 10) becomes "\n".

* Double-quoting of parameter values, as well as backslash escaping in parameter values, is carried out. For example, <param>"foo","bar"</param> becomes PARAM="\"foo\",\"bar\"".

For example, these two vCards are equivalent:

```
<?xml version="1.0"?>
<vcards xmlns="urn:ietf:params:xml:ns:vcard-4.0">
  <vcard>
    <fn><text>J. Doe</text></fn>
    <n>
      <surname>Doe</surname>
      <given>J.</given>
      <additional/>
      <prefix/>
      <suffix/>
    </n>
    <x-file>
      <parameters>
        <mediatype><text>image/jpeg</text></mediatype>
      </parameters>
      <unknown>alien.jpg</unknown>
    </x-file>
    <a xmlns="http://www.w3.org/1999/xhtml"
       href="http://www.example.com">My web page!</a>
  </vcard>
</vcards>
```

```
BEGIN:VCARD
VERSION:4.0
FN:J. Doe
N:Doe;J.;;
X-FILE;MEDIATYPE=image/jpeg:alien.jpg
XML:<a xmlns="http://www.w3.org/1999/xhtml"\n
    href="http://www.example.com">My web page!</a>
END:VCARD
```

## 7. Security Considerations

All the security considerations applicable to plain vCard [RFC6350] are applicable to this document as well.

XML Signature [W3C.CR-xmldsig-core1-20110303] and XML Encryption [W3C.CR-xmlenc-core1-20110303] can be used with xCard to provide authentication and confidentiality.

# 8. IANA Considerations

## 8.1. Registration of the XML Namespace

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

Registrant Contact: The IESG <iesg@ietf.org>

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

## 8.2. Media Type

This section defines the MIME media type [RFC4288] for use with vCard-in-XML data.

To: ietf-types@iana.org

Subject: Registration of media type application/vcard+xml

Type name: application

Subtype name: vcard+xml

Required parameters: none

Optional parameters: charset as defined for application/xml in [RFC3023]; per [RFC3023], use of the charset parameter with the value "utf-8" is "STRONGLY RECOMMENDED".

Encoding considerations: Same as encoding considerations of application/xml as specified in [RFC3023].

Security considerations: This media type has all of the security considerations described in [RFC3023], plus those listed in Section 7.

Interoperability considerations: This media type provides an alternative syntax to vCard data [RFC6350] based on XML.

Published specification: This specification.

Applications that use this media type: Applications that currently make use of the text/vcard media type can use this as an alternative. In general, applications that maintain or process contact information can use this media type.

Additional information:

Magic number(s): none

File extension(s): XML data should use ".xml" as the file extension.

Macintosh file type code(s): none

Person & email address to contact for further information: Simon Perreault <simon.perreault@viagenie.ca>

Intended usage: COMMON

Restrictions on usage: none

Author: Simon Perreault

Change controller: IETF

## 9. Acknowledgments

Thanks to the following people for their input:

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

## 10. References

### 10.1. Normative References

[ISO.19757-2.2008] International Organization for Standardization, "Information technology -- Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG", ISO International Standard 19757-2, October 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, August 2011.

[W3C.REC-xml-20081126]
Paoli, J., Yergeau, F., Maler, E., Bray, T., and C. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126>.

[W3C.REC-xml-names-20091208]
Bray, T., Hollander, D., Layman, A., Tobin, R., and H. Thompson, "Namespaces in XML 1.0 (Third Edition)", World Wide Web Consortium Recommendation REC-xml-names-20091208, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208>.

## 10.2. Informative References

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[VCARD-DTD]
Dawson, F., "The vCard v3.0 XML DTD", Work in Progress, June 1998.

[W3C.CR-xmldsig-core1-20110303]
Roessler, T., Solo, D., Yiu, K., Reagle, J., Hirsch, F., Eastlake, D., and M. Nystroem, "XML Signature Syntax and Processing Version 1.1", World Wide Web Consortium CR CR-xmldsig-core1-20110303, March 2011, <http://www.w3.org/TR/2011/CR-xmldsig-core1-20110303>.

[W3C.CR-xmlenc-core1-20110303]
Eastlake, D., Reagle, J., Roessler, T., and F. Hirsch, "XML Encryption Syntax and Processing Version 1.1", World Wide Web Consortium CR CR-xmlenc-core1-20110303, March 2011, <http://www.w3.org/TR/2011/CR-xmlenc-core1-20110303>.

## Appendix A. Relax NG Schema

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

### Section 3.3: vCard Format Specification
#
# 3.3
iana-token = xsd:string { pattern = "[a-zA-Z0-9-]+" }
x-name = xsd:string { pattern = "x-[a-zA-Z0-9-]+" }

### Section 4: Value types
#
# 4.1
value-text = element text { text }
value-text-list = value-text+

# 4.2
value-uri = element uri { xsd:anyURI }

# 4.3.1
value-date = element date {
    xsd:string { pattern = "\d{8}|\d{4}-\d\d|--\d\d(\d\d)?|---\d\d" }
  }

# 4.3.2
value-time = element time {
    xsd:string { pattern = "(\d\d(\d\d(\d\d)?)?|-\d\d(\d\d?)|--\d\d)"
                         ~ "(Z|[+\-]\d\d(\d\d)?)?" }
  }

# 4.3.3
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)?)?" }
  }

# 4.3.4
value-date-and-or-time = value-date | value-date-time | value-time

# 4.3.5
value-timestamp = element timestamp {
    xsd:string { pattern = "\d{8}T\d{6}(Z|[+\-]\d\d(\d\d)?)?" }
  }

# 4.4
value-boolean = element boolean { xsd:boolean }
```

```
# 4.5
value-integer = element integer { xsd:integer }

# 4.6
value-float = element float { xsd:float }

# 4.7
value-utc-offset = element utc-offset {
    xsd:string { pattern = "[+\-]\d\d(\d\d)?" }
  }

# 4.8
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}" }
  }
```

### Section 5: Parameters

```
#
# 5.1
param-language = element language { value-language-tag }?

# 5.2
param-pref = element pref {
    element integer {
      xsd:integer { minInclusive = "1" maxInclusive = "100" }
    }
  }?

# 5.4
param-altid = element altid { value-text }?

# 5.5
param-pid = element pid {
    element text { xsd:string { pattern = "\d+(\.\d+)?" } }+
  }?

# 5.6
param-type = element type { element text { "work" | "home" }+ }?

# 5.7
param-mediatype = element mediatype { value-text }?
```

```
# 5.8
param-calscale = element calscale { element text { "gregorian" } }?

# 5.9
param-sort-as = element sort-as { value-text+ }?

# 5.10
param-geo = element geo { value-uri }?

# 5.11
param-tz = element tz { value-text | value-uri }?

### Section 6: Properties
#
# 6.1.3
property-source = element source {
    element parameters { param-altid, param-pid, param-pref,
                         param-mediatype },
    value-uri
  }

# 6.1.4
property-kind = element kind {
    element text { "individual" | "group" | "org" | "location" |
                   x-name | iana-token }*
  }

# 6.2.1
property-fn = element fn {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type }?,
    value-text
  }

# 6.2.2
property-n = element n {
    element parameters { param-language, param-sort-as, param-altid }?,
    element surname { text }+,
    element given { text }+,
    element additional { text }+,
    element prefix { text }+,
    element suffix { text }+
  }
```

```
# 6.2.3
property-nickname = element nickname {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type }?,
    value-text-list
  }

# 6.2.4
property-photo = element photo {
    element parameters { param-altid, param-pid, param-pref, param-type,
                         param-mediatype }?,
    value-uri
  }

# 6.2.5
property-bday = element bday {
    element parameters { param-altid, param-calscale }?,
    (value-date-and-or-time | value-text)
  }

# 6.2.6
property-anniversary = element anniversary {
    element parameters { param-altid, param-calscale }?,
    (value-date-and-or-time | value-text)
  }

# 6.2.7
property-gender = element gender {
    element sex { "" | "M" | "F" | "O" | "N" | "U" },
    element identity { text }?
  }

# 6.3.1
param-label = element label { value-text }?
property-adr = element adr {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type, param-geo, param-tz,
                         param-label }?,
    element pobox { text }+,
    element ext { text }+,
    element street { text }+,
    element locality { text }+,
    element region { text }+,
    element code { text }+,
    element country { text }+
  }
```

```
# 6.4.1
property-tel = element tel {
    element parameters {
      param-altid,
      param-pid,
      param-pref,
      element type {
        element text { "work" | "home" | "text" | "voice"
                     | "fax" | "cell" | "video" | "pager"
                     | "textphone" }+
      }?,
      param-mediatype
    }?,
    (value-text | value-uri)
  }

# 6.4.2
property-email = element email {
    element parameters { param-altid, param-pid, param-pref,
                         param-type }?,
    value-text
  }

# 6.4.3
property-impp = element impp {
    element parameters { param-altid, param-pid, param-pref,
                         param-type, param-mediatype }?,
    value-uri
  }

# 6.4.4
property-lang = element lang {
    element parameters { param-altid, param-pid, param-pref,
                         param-type }?,
    value-language-tag
  }

# 6.5.1
property-tz = element tz {
    element parameters { param-altid, param-pid, param-pref,
                         param-type, param-mediatype }?,
    (value-text | value-uri | value-utc-offset)
  }
```

```
# 6.5.2
property-geo = element geo {
    element parameters { param-altid, param-pid, param-pref,
                         param-type, param-mediatype }?,
    value-uri
  }

# 6.6.1
property-title = element title {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type }?,
    value-text
  }

# 6.6.2
property-role = element role {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type }?,
    value-text
  }

# 6.6.3
property-logo = element logo {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type, param-mediatype }?,
    value-uri
  }

# 6.6.4
property-org = element org {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type, param-sort-as }?,
    value-text-list
  }

# 6.6.5
property-member = element member {
    element parameters { param-altid, param-pid, param-pref,
                         param-mediatype }?,
    value-uri
  }
```

```
# 6.6.6
property-related = element related {
    element parameters {
      param-altid,
      param-pid,
      param-pref,
      element type {
        element text {
          "work" | "home" | "contact" | "acquaintance" |
          "friend" | "met" | "co-worker" | "colleague" | "co-resident" |
          "neighbor" | "child" | "parent" | "sibling" | "spouse" |
          "kin" | "muse" | "crush" | "date" | "sweetheart" | "me" |
          "agent" | "emergency"
        }+
      }?,
      param-mediatype
    }?,
    (value-uri | value-text)
  }

# 6.7.1
property-categories = element categories {
    element parameters { param-altid, param-pid, param-pref,
                         param-type }?,
    value-text-list
  }

# 6.7.2
property-note = element note {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type }?,
    value-text
  }

# 6.7.3
property-prodid = element prodid { value-text }

# 6.7.4
property-rev = element rev { value-timestamp }

# 6.7.5
property-sound = element sound {
    element parameters { param-language, param-altid, param-pid,
                         param-pref, param-type, param-mediatype }?,
    value-uri
  }
```

```
# 6.7.6
property-uid = element uid { value-uri }

# 6.7.7
property-clientpidmap = element clientpidmap {
    element sourceid { xsd:positiveInteger },
    value-uri
  }

# 6.7.8
property-url = element url {
    element parameters { param-altid, param-pid, param-pref,
                         param-type, param-mediatype }?,
    value-uri
  }

# 6.8.1
property-key = element key {
    element parameters { param-altid, param-pid, param-pref,
                         param-type, param-mediatype }?,
    (value-uri | value-text)
  }

# 6.9.1
property-fburl = element fburl {
    element parameters { param-altid, param-pid, param-pref,
                         param-type, param-mediatype }?,
    value-uri
  }

# 6.9.2
property-caladruri = element caladruri {
    element parameters { param-altid, param-pid, param-pref,
                         param-type, param-mediatype }?,
    value-uri
  }

# 6.9.3
property-caluri = element caluri {
    element parameters { param-altid, param-pid, param-pref,
                         param-type, param-mediatype }?,
    value-uri
  }
```

```
# Top-level grammar
property = property-adr | property-anniversary | property-bday
         | property-caladruri | property-caluri | property-categories
         | property-clientpidmap | property-email | property-fburl
         | property-fn | property-geo | property-impp | property-key
         | property-kind | property-lang | property-logo
         | property-member | property-n | property-nickname
         | property-note | property-org | property-photo
         | property-prodid | property-related | property-rev
         | property-role | property-gender | 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*
         })+
    }+
  }
```

## 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