This document specifies an XMPP extension for use of the vCard4 XML format in XMPP systems, with the intent of obsoleting the vcard-temp format.
Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 by the XMPP Standards Foundation (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

## NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. ##

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF’s Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy> or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).
11.4 Properties Defined Similarly in vcard-temp, vCard3, and vCard4 .......................... 16
11.5 Properties Defined in vCard3 but Removed from vCard4 ................................. 17

12 Migration Tools ............................................................................................................. 17
   12.1 Extensible Stylesheet Language Transformation (XSLT) from vcard-temp to vCard4 ............................................................................................................ 18
   12.2 Example of vcard-temp Data .............................................................................. 29
   12.3 Example of vCard4 XML Data ............................................................................. 34

13 Acknowledgements ......................................................................................................... 41
1 Introduction

Since 1999, the Jabber/XMPP community has used an interim, unofficial XML representation of vCard data for personal contacts, called vcard-temp (XEP-0054). Recently, the IETF has upgraded vCard from vCard3 to vCard 4 (RFC 6350), and at the same time has defined an official XML format for vCard4 (RFC 6351). This document specifies an XMPP extension for use of the vCard4 XML format in XMPP systems, with the intent of obsoleting the vcard-temp format. Primarily this document defines the encapsulation method itself; secondarily it also defines transport methods and a mapping to the vcard-temp format for migration by clients and servers.

2 Requirements

This specification was designed with the following requirements in mind.

1. Reuse vCard4 as defined in RFC 6350.
2. Reuse the vCard4 XML format as defined in RFC 6351.
3. Ensure that clients and servers can easily migrate from vcard-temp to the new encapsulation format.
5. Support vCards for non-human entities such as XMPP servers and Multi-User Chat (XEP-0045) rooms.

3 Reuse of vCard4

Because there is now an XML namespace for the official vCard format, we can simply re-use that namespace: “urn:ietf:params:xml:ns:vcard-4.0”.

The vCard XML format defined at the IETF specifies that the root element is <vcards/>, where the only defined child element is <vcard/>. For use in XMPP, we specify that the root element shall be <vcard/>, not <vcards/>.

---

4 Self vCards

This section describes the use of the vCard format for self-publication and retrieval of publicly-accessible information about any entity on an XMPP network, thus fulfilling all the use cases of the old vcard-temp format.

4.1 PEP Publication and Retrieval

The primary method for publishing and retrieving vCards is via Best Practices for Persistent Storage of Public Data via Publish-Subscribe (XEP-0222) using the node name 'urn:xmpp:vcard4'.

4.1.1 Retrieval

An XMPP entity retrieves the vCard of another entity (or itself) by sending a PEP-get to the target entity for the 'urn:xmpp:vcard4' node.

Listing 1: vCard Retrieval Request

```xml
<iq from='samizzi@cisco.com/foo' id='bx81v356' to='stpeter@jabber.org' type='get'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='urn:xmpp:vcard4'/>
  </pubsub>
</iq>
```

If a vCard exists for the target entity, the responsible entity (e.g., the XMPP server that hosts the account for a bare JID) MUST return the data in an IQ-result:

Listing 2: Server Returns vCard

```xml
<iq from='stpeter@jabber.org' id='bx81v356' to='samizzi@cisco.com/foo' type='result'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='urn:xmpp:vcard4'>
      <item id='e0bf7714-a8dd-4749-8a18-e1979525b0d5'>
        <vcard xmlns='urn:ietf:params:xml:ns:vcard-4.0'>
          <fn><text>Peter Saint-Andre</text></fn>
        </vcard>
      </item>
    </items>
  </pubsub>
</iq>
```

---

<vcard>
  <tel>tel:+1-303-555-1212</tel>
  <geo>geo:39.59,-105.01</geo>
  <title>Executive Director</title>
  <role>Patron Saint</role>
  <org>
    <parameters>
      <type><text>work</text></type>
      <text>XMPP Standards Foundation</text>
    </parameters>
    <url>https://stpeter.im/</url>
    <note>
      More information about me is located on my personal website: https://stpeter.im/
    </note>
    <gender><sex>M</sex></gender>
    <language-tag>en</language-tag>
    <email>
      <parameters>
        <type><text>work</text></type>
        <text>psaintan@cisco.com</text>
      </parameters>
      <email>stpeter@jabber.org</email>
    </email>
    <impp>
      <parameters>
        <type><text>work</text></type>
        <uri>xmpp:psaintan@cisco.com</uri>
      </impp>
      <impp>
        <parameters>
          <type><text>home</text></type>
          <uri>xmpp:stpeter@jabber.org</uri>
        </impp>
        <key>
          <uri>https://stpeter.im/stpeter.asc</uri>
        </key>
      </impp>
    </impp>
  </org>
  <email>
    <parameters>
      <type><text>home</text></type>
      <text>stpeter@jabber.org</text>
    </parameters>
  </email>
  <email>
    <parameters>
      <type><text>home</text></type>
      <text>stpeter@jabber.org</text>
    </parameters>
  </email>
  <email>
    <parameters>
      <type><text>home</text></type>
      <text>psaintan@cisco.com</text>
    </parameters>
  </email>
  <email>
    <parameters>
      <type><text>home</text></type>
      <text>stpeter@jabber.org</text>
    </parameters>
  </email>
  <email>
    <parameters>
      <type><text>home</text></type>
      <text>psaintan@cisco.com</text>
    </parameters>
  </email>
</vcard>
4.1.2 Publication

An XMPP entity publishes or updates its vCard by doing a PEP publish to its own 'urn:xmpp:vcard4' node. The publication request needs to include the entire vCard, not a "diff" against the prior data (if any).

Listing 3: vCard Publication Request

```xml
<iq from='stpeter@jabber.org/squire'
     id='h3vz319m'
     type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:vcard4'>
      <item>
        <vcard xmlns='urn:ietf:params:xml:ns:vcard-4.0'>
          [...]
        </vcard>
      </item>
    </publish>
  </pubsub>
</iq>
```

If no error occurs, the responsible entity returns an IQ-result.

Listing 4: Server Acknowledges Publication

```xml
<iq from='stpeter@jabber.org'
     id='bx81v356'
     to='stpeter@jabber.org/squire'
     type='result'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:vcard4'>
      <item id='1e3fdd90-f542-41d3-82ae-33d183467a7a'/>
    </publish>
  </pubsub>
</iq>
```

Note: An entity MAY have authorization to update the vCard of another entity (e.g., a server administrator might have authorization to modify the server’s vCard).

4.2 Event Notifications

Personal Eventing Protocol (XEP-0163)\(^7\) also provides a way to subscribe to events. It can be used to automatically generate event notifications when a user's vCard is modified.

4.2.1 Subscribing to vCard Notifications

Let us imagine that Juliet wishes to receive the updates that Romeo publishes to his vCard. She has two options:

1. Implicitly subscribe by advertising support for "urn:xmpp:vcard4+notify" in her Entity Capabilities (XEP-0115) data. Romeo’s PEP service then automatically sends vCard updates to her when it receives presence from her, until and unless she sends presence of type unavailable or stops advertising an interest in vCard updates. This is in accordance with Publish-Subscribe (XEP-0060), section 6.1.

2. Explicitly subscribe by sending a formal subscription request to the "urn:xmpp:vcard4" node at Romeo’s JabberID. Romeo’s PEP service might send her all vCard updates even if she is offline at the time (depending on service policies regarding presence integration).

4.2.2 Receiving a vCard Notification

Because Juliet has sent presence to Romeo including Entity Capabilities data that includes the "urn:xmpp:vcard4+notify" feature, Romeo’s XMPP server will send a PEP notification to Juliet.

Listing 5: Receiving a vCard publication/update

```
<message from='romeo@montague.lit'
         to='juliet@capulet.lit'
         type='headline'>
   <event xmlns='http://jabber.org/protocol/pubsub#event'>
     <items node='urn:xmpp:vcard4'>
       <item id='1e3fdd90-f542-41d3-82ae-33d183467a7a'/>
     </items>
   </event>
</message>
```

Note: There is no payload, because this is a pure notification (the receiver needs to retrieve the vCard as described earlier).

5 Contact vCards

In addition to enabling the publication and retrieval of vCards about any entity on an XMPP network, the vCard format can also be used to store information about an entity’s contacts.

---

5.1 Format

A contact is simply a vCard about someone else (or something else, in the case of automated entities). If the other person or entity is in the user’s roster RFC 6121, the vCard SHOULD contain the Jabber ID of the person or entity. This enables a user to store information about the contact outside of the roster, thus obviating the need for changes or extensions to the roster namespace itself (as in Annotations (XEP-0145)).

Listing 6: Contact

```xml
<vcard xmlns="urn:ietf:params:xml:ns:vcard-4.0">
  <fn><text>Samantha Mizzi</text></fn>
  <n>
    <given>Samantha</given>
    <surname>Mizzi</surname>
    <additional></additional>
  </n>
  <nickname><text>Sam</text></nickname>
  <nickname><text>samizzi</text></nickname>
  <geo><uri>geo:39.59,-105.01</uri></geo>
  <org>
    <parameters><type><text>work</text></type></parameters>
    <text>Cisco</text>
  </org>
  <note>
    My co-author on XEP-0292. She’s cool!
  </note>
</vcard>
```

5.2 Storage

Because contact vCards are private information, they are best stored using Best Practices for Persistent Storage of Private Data via Publish-Subscribe (XEP-0223). The canonical location is a well-known pubsub node "urn:xmpp:contacts". In accordance with XEP-0223, this node MUST have an access type of "whitelist" by default. When a client stores items at this node, it

---

MUST include an ItemID set to the bare JID of the contact.

Listing 7: Storing a Contact vCard

```xml
<iq from='stpeter@stpeter.im/squire' type='set' id='h3vs7163'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:contacts'>
      <item id='samizzi@cisco.com'>
        <vcard xmlns='urn:ietf:params:xml:ns:vcard-4.0'>
          <fn><text>Samantha Mizzi</text></fn>
          <n>
            <given>Samantha</given>
            <additional></additional>
          </n>
          <nickname><text>Sam</text></nickname>
          <nickname><text>samizzi</text></nickname>
          <geo><uri>geo:39.59,-105.01</uri></geo>
          <org><parameters><type><text>work</text></type></parameters>
          <text>Cisco</text></org>
        </vcard>
      </item>
    </publish>
  </pubsub>
</iq>
```

My co-author on XEP-0292. She's cool!
When a contact’s vCard is stored in a private node, it is pushed out to all of the user’s resources that have included in their entity capabilities (XEP-0115) data a service discovery feature of "urn:xmpp:contacts+notify" (in the following example those resources are "squire" and "roundabout").

Listing 8: Publisher resources receive event notification

```xml
<message from='stpeter@stpeter.im'
         to='stpeter@stpeter.im/squire'
         type='headline'
         id='ka92g1b5'>
   <event xmlns='http://jabber.org/protocol/pubsub#event'>
     <items node='urn:xmpp:contacts'>
       <item id='samizzi@cisco.com'>
         <vcard xmlns='urn:ietf:params:xml:ns:vcard-4.0'>
             <fn><text>Samantha Mizzi</text></fn>
             <n>
              <surname>Mizzi</surname>
              <given>Samantha</given>
              <additional></additional>
             </n>
             <nickname><text>Sam</text></nickname>
             <nickname><text>samizzi</text></nickname>
             <geo><uri>geo:39.59,-105.01</uri></geo>
             <org><parameters><type><text>work</text></type><text>Cisco</text> </parameters></org>
             <note>
             <text>My co-author on XEP-0292. She's cool!</text>
             </note>
             <impp><parameters><type><text>work</text></type><text>xmpp:samizzi@cisco.com</text> </parameters></impp>
         </vcard>
       </item>
     </items>
   </event>
</message>

<message from='stpeter@stpeter.im'
         to='stpeter@stpeter.im/roundabout'
         type='headline'
         id='pty14x69'>
```
6 vCards of Automated Entities

Traditionally, vCards have been used on the XMPP network for entities other than human users, e.g. by XMPP servers and chatrooms. When such automated entities use vCards, it is RECOMMENDED to specify a value of "application" for the vCard4 KIND property RFC 6473, as illustrated in the following example:

Listing 9: vCard for a Thing

```
<iq from='jabber.org' id='yhx51c35' to='samizzi@cisco.com/fo0' type='result'>
    <fn>Sam Mizzi</fn>
    <given>Samantha</given>
    <additional>Mizz</additional>
    <nickname>Sam</nickname>
    <nickname>samizzi</nickname>
    <geo uri='geo:39.59,-105.01'/>
    <org parameters='type'><text>Cisco</text> </org>
    <note><text>My co-author on XEP-0292...She's cool!</text></note>
  </vcard>
</iq>
```
7 Determining Support

If an XMPP client or server supports the vCard4 namespace, it MUST advertise that fact in its responses to Service Discovery (XEP-0030) information (“disco#info”) requests by returning a feature of “urn:ietf:params:xml:ns:vcard-4.0”:

Listing 10: A disco#info query

```xml
<iq type='get'>
  from='stpeter@jabber.org/squire'
  to='samizzi@cisco.com/foo'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 11: A disco#info response

```xml
<iq type='result'>
  from='samizzi@cisco.com/foo'
  to='stpeter@jabber.org/squire'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...<feature var='urn:ietf:params:xml:ns:vcard-4.0'/>
  </query>
</iq>
```

In order for an application to determine whether an entity supports this protocol, where possible it SHOULD use the dynamic, presence-based profile of service discovery defined in XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.

---

in Entity Capabilities (XEP-0115) \(^{15}\). However, if an application has not received entity capabilities information from an entity, it SHOULD use explicit service discovery instead.

### 8 Security Considerations

The vCard information published to one’s XMPP server is world-readable; therefore, users should exercise due caution when determining what information to include (e.g., street addresses, personal telephone numbers, or email addresses).

### 9 IANA Considerations

This document does not require interaction with the Internet Assigned Numbers Authority (IANA) \(^{16}\).

### 10 XMPP Registrar Considerations

10.1 Well-Known Service Discovery Nodes

The XMPP Registrar \(^{17}\) shall include ‘urn:xmpp:contact’ and ‘urn:xmpp:vcard4’ in its registry of Nodes for Service Discovery and Publish-Subscribe at \(<https://xmpp.org/registrar/nodes.html>\).

### 11 Mapping from vcard-temp to vCard4

This section provides a more detailed description of mapping vcard-temp properties to vcard4 properties.

---

\(^{15}\)XEP-0115: Entity Capabilities \(<https://xmpp.org/extensions/xep-0115.html>\).

\(^{16}\)The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see \(<http://www.iana.org/>\).

\(^{17}\)The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see \(<https://xmpp.org/registrar/>\).
11.1 Properties Defined in vcard-temp but not in vCard3 or vCard4

11.1.1 DESC

The vcard-temp specification defined a <DESC/> element. This element was not part of the vCard3 schema. Mapping the vcard-temp <DESC/> element to the vCard4 NOTE property is appropriate.

Listing 12: Deprecated DESC element

```
<DESC>
  More information about me is located on my personal website: https://stpeter.im/
</DESC>
```

Listing 13: NOTE property

```
<note>
  <text>
    More information about me is located on my personal website: https://stpeter.im/
  </text>
</note>
```

11.1.2 JABBERID

The vcard-temp specification defined a <JABBERID/> element:

Listing 14: Deprecated JABBERID element

```
<JABBERID>stpeter@jabber.org</JABBERID>
```

Although the JABBERID field was not part of the vCard3 schema and was simply hacked into vcard-temp, RFC 4770 18 defined an IMPP property for instant messaging and presence addresses, which was ported to vCard4. In the vCard4 XML format, the IMPP property for a JabberID would be as follows.

Listing 15: IMPP property

```
<impp>
  <uri>xmpp:stpeter@jabber.org</uri>
</impp>
```

11.1.3 MIDDLE

The vcard-temp specification defined a `<MIDDLE/>` element as the third allowable element within the `<N/>` ("name") element. This element was not part of the vCard3 schema, although the Dawson drafts did contain an `<other/>` element in the third position of child elements within the `<n/>` element. It is appropriate to map the vcard-temp `<MIDDLE/>` element to the vCard4 "Additional Name" part of the "N" structured property value, which in xCard is the `<additional/>` child of the `<n/>` element.

11.2 Properties Defined Incorrectly in vcard-temp

Several of the properties in vcard-temp are defined differently in vCard3. In fact, the definitions even differ from those provisionally made in the so-called "Dawson drafts" from which vcard-temp was supposedly derived (for reference, the last of these is archived at <http://www.watersprings.org/pub/id/draft-dawson-vcard-xml-dtd-03.txt>). The reasons for these discrepancies are unknown. However, care must be taken in correctly mapping these properties from vcard-temp to vCard4.

11.2.1 KEY

The DTD in XEP-0054 provided this definition for the KEY element:

```xml
<!ELEMENT KEY ( TYPE?, CRED )>
```

However, the DTD in the final Dawson draft provided the following definition:

```xml
<!ELEMENT key ( extref | b64bin )>
```

The relevant RelaxNG definition in vCard4 XML is as follows:

```xml
property-key = element key {
  ( value-uri | value-text )
}
```

The source of the spurious `<TYPE/>` and `<CRED/>` elements is unknown. The vcard-temp `<CRED/>` element is mapped to the vCard4 value-text construction.

11.2.2 SOUND

The DTD in XEP-0054 provided this definition for the SOUND element:

```xml
<!ELEMENT SOUND ( PHONETIC | BINVAL | EXTVAL )>
```
However, the DTD in the final Dawson draft provided the following definition:

```xml
<!ELEMENT sound (extref | b64bin)>
```

The source of the spurious vcard-temp `<PHONETIC/>` element is unknown. However, it does not exist in vCard4 and therefore is simply discarded when mapping. The vcard-temp `<BIN-VAL/>` element is mapped to the vCard4 b64bin construction and the vcard-temp `<EXTVAL/>` element is mapped to the vCard4 extref construction.

### 11.2.3 VERSION

As explained in XEP-0054, the `<VERSION/>` element from the final Dawson draft was not used in vcard-temp; instead, the vcard-temp protocol used a 'version' attribute (in fact the Dawson drafts were inconsistent, since the DTD defined a `<VERSION/>` element and the body of the specification used a 'version' attribute).

### 11.3 Properties Defined Differently in vcard-temp, vCard3, and vCard4

The following properties are defined differently in vcard-temp and vCard4. As a result, the mappings are workable but might not preserve all information that could have been contained in vcard-temp data.

#### 11.3.1 ADR

The following address type values allowed in vCard3 were removed from vCard4:

- DOM
- INTL
- PARCEL
- POSTAL

#### 11.3.2 AGENT

In vCard3 and vcard-temp, the AGENT property was allowed to contain the inline vCard of someone who could act as an agent for the primary owner of the referenced vCard. In vCard4, inline vCards are disallowed. Therefore only pointers to external vCard objects are now allowed, by means of a URI.
11.3.3 ORG

The ORGUNIT property was removed from vCard4, with the result that the ORGNAME property becomes the only child of ORG.

11.3.4 SORT-STRING

The SORT-STRING property from vCard3 was renamed to SORT-AS in vCard4.

11.3.5 TEL

The following telephony type values allowed in vCard3 were removed from vCard4:

- BBS
- ISDN
- MODEM
- MSG
- PCS

In addition, in vCard4 the telephone number is represented as a tel: URI, not by means of a NUMBER property.

11.4 Properties Defined Similarly in vcard-temp, vCard3, and vCard4

The following properties are defined similarly in vcard-temp, vCard3, and vCard4. The mappings are fairly straightforward (a future version of this document might provide more detailed narrative descriptions of the mappings).

- BDAY
- CATEGORIES
- EMAIL
- FN
- GEO
- LOGO
- N
12 Migration Tools

This section contains three tools that are intended to help developers in migrating from vcard-temp to vCard4 XML:

1. An Extensible Stylesheet Language Transformation (XSLT) script for automatically translating the vcard-temp XML format into the vCard4 XML format.

2. An example of vcard-temp data that uses most of the elements defined in XEP-0054 that can be mapped to vCard4 properties (note that some of these elements were never used in practice).

11.5 Properties Defined in vCard3 but Removed from vCard4

The following properties were defined in vCard3 but were removed from vCard4:

- CLASS
- LABEL
- MAILER

There is no mapping from these properties to vCard4.

3. An example of vCard4 XML data showing the transformation of the vcard-temp example using the XSLT stylesheet.

The tools are purely informational and are not a normative part of this specification.

12.1 Extensible Stylesheet Language Transformation (XSLT) from vcard-temp to vCard4

```xml
<?xml version='1.0' encoding='UTF-8'?>
<!--
Copyright (c) 1999 - 2021 XMPP Standards Foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
-->

<-- Author: stpeter@jabber.org -->
</xsl:stylesheet
xmlns='urn:ietf:params:xml:ns:vcard-4.0'
```
<!-- END VARIABLES -->

<!-- BEGIN VCARD4 OUTPUT -->

<!-- FN is required -->

<fn><text><xsl:value-of select='/vCard/FN'/></text></fn>

<!-- N is required -->

<xsl:apply-templates select='/vCard'/>

<!-- NICKNAME can be included one or more times -->

<xsl:for-each select='/vCard/NICKNAME'>
 <nickname><text><xsl:value-of select='.'/></text></nickname>
</xsl:for-each>

<!-- PHOTO can be included one or more times -->

<xsl:for-each select='/vCard/PHOTO'>
 <xsl:variable name='PHOTO.ext' select='count(EXTVAL)'/>
 <xsl:variable name='PHOTO.type' select='TYPE'/>
 <xsl:choose>
  <xsl:when test='$PHOTO.ext=1'>
   <photo><uri><xsl:value-of select='EXTVAL'/></uri></photo>
  </xsl:when>
  <xsl:otherwise>
   <photo>
    <uri data:><xsl:text><xsl:value-of select='TYPE'/></xsl:text>;
     <xsl:value-of select='BINVAL'/></xsl:text></uri>
   </photo>
  </xsl:otherwise>
 </xsl:choose>
</xsl:for-each>

<!-- BDAY can be included once -->

<xsl:if test='$BDAY.count=1'>
 <bday><date><xsl:value-of select='/vCard/BDAY'/></date></bday>
</xsl:if>

<!-- ADR can be included one or more times -->

<xsl:for-each select='/vCard/BDAY'>
 <adr>
  <xsl:value-of select='.'/>
 </adr>
</xsl:for-each>
<xsl:apply-templates select='/vCard/ADR'/>

<!-- NOTE: vcard-temp allowed a LABEL element, but it was removed from vCard4 -->

<!-- one or more TEL elements can be included -->
<xsl:apply-templates select='/vCard/TEL'/>

<!-- one or more EMAIL elements can be included -->
<xsl:apply-templates select='/vCard/EMAIL'/>

<!-- JABBERID can be included one or more times -->
<!-- we map this to the vCard4 IMPP property -->
<xsl:for-each select='/vCard/JABBERID'>
  <impp><uri><xsl:apply-template select='.'/></uri></impp>
</xsl:for-each>

<!-- NOTE: vcard-temp allowed a MAILER element, but it was removed from vCard4 -->

<!-- one or more TZ elements can be included -->
<xsl:for-each select='/vCard/TZ'>
  <tz><text><xsl:value-of select='.'/></text></tz>
</xsl:for-each>

<!-- one or more GEO elements can be included -->
<xsl:for-each select='/vCard/GEO'>
  <geo><uri><xsl:apply-template select='.'/></uri></geo>
</xsl:for-each>

<!-- one or more TITLE elements can be included -->
<xsl:for-each select='/vCard/TITLE'>
  <title><text><xsl:value-of select='.'/></text></title>
</xsl:for-each>

<!-- one or more ROLE elements can be included -->
<xsl:for-each select='/vCard/ROLE'>
  <role><text><xsl:value-of select='.'/></text></role>
</xsl:for-each>
<xsl:for-each select='vCard/LOGO'>
  <xsl:choose>
    <xsl:when test='$LOGO.ext=1'>
      <logo>
        <uri><xsl:value-of select='EXTVAL'/></uri>
      </logo>
    </xsl:when>
    <xsl:otherwise>
      <logo>
        <uri><xsl:text data=''>
          <xsl:value-of select='TYPE'/>
        </xsl:text>base64</uri>
        <xsl:value-of select='BINVAL'/>
      </logo>
    </xsl:otherwise>
  </xsl:choose>
</xsl:for-each>

<xsl:for-each select='vCard/AGENT'>
  <xsl:if test='$AGENT.ext=1'>
    <agent>
      <uri><xsl:value-of select='EXTVAL'/></uri>
    </agent>
  </xsl:if>
</xsl:for-each>

<xsl:for-each select='vCard/ORG'>
  <xsl:if test='$ORGNAME.count=1'>
    <text><xsl:value-of select='ORGNAME'/></text>
  </xsl:if>
</xsl:for-each>

<xsl:for-each select='vCard/CATEGORIES'>
  <xsl:choose>
    <xsl:when test='$CATEGORIESTYPES.count=1'>
      <xsl:value-of select='CATEGORIESTYPES'/>
    </xsl:when>
    <xsl:otherwise>
      
    </xsl:otherwise>
  </xsl:choose>
</xsl:for-each>
<!-- NOTE: this element was probably unused in vcard-temp -->
<xsl:for-each select='/vCard/CATEGORIES'>
  <categories><text><xsl:value-of select='.'/></text></categories>
</xsl:for-each>

<!-- one or more NOTE elements can be included -->
<!-- NOTE: this element was probably unused in vcard-temp -->
<xsl:for-each select='/vCard/NOTE'>
  <note><text><xsl:value-of select='.'/></text></note>
</xsl:for-each>

<!-- PRODID can be included exactly once -->
<!-- NOTE: this element was probably unused in vcard-temp -->
<xsl:if test='$PRODID.count=1'>
  <prodid><text><xsl:value-of select='.'/></text></prodid>
</xsl:if>

<!-- REV can be included exactly once -->
<!-- NOTE: this element was probably unused in vcard-temp -->
<xsl:if test='$REV.count=1'>
  <rev><timestamp><xsl:value-of select='.'/></timestamp></rev>
</xsl:if>

<!-- one or more SORT-STRING elements can be included -->
<!-- this element maps to SORT-AS in vCard4 -->
<xsl:for-each select='/vCard/SORT-STRING'>
  <sort-as><xsl:value-of select='.'/></sort-as>
</xsl:for-each>

<!-- one or more SOUND elements can be included -->
<!-- NOTE: for some reason, vcard-temp allowed a <PHONETIC/> child element, but that was not documented in the original Dawson I-Ds and is not supported in vCard4 -->
<xsl:for-each select='/vCard/SOUND'>
  <xsl:variable name='SOUND.ext' select='count(EXTVAL)'/>
  <xsl:variable name='SOUND.bin' select='count(BINVAL)'/>
  <xsl:choose>
    <xsl:when test='$SOUND.ext=1'>
      <sound><uri><xsl:value-of select='EXTVAL'/></uri></sound>
    </xsl:when>
    <xsl:when test='$SOUND.ext=1'>
      <sound>
        ...
      </sound>
    </xsl:when>
  </xsl:choose>
</xsl:for-each>
<uri><xsl:text data:audio/basic;base64, </xsl:text><xsl:value-of select='BINVAL'/></uri>
</sound>
</xsl:when>
<xsl:otherwise>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>

<!-- UID can be included exactly once -->
<xsl:if test='UID.count=1'>
<uid><uri><xsl:value-of select='/vCard/UID'/></uri></uid>
</xsl:if>

<!-- URL can be included one or more times -->
<xsl:for-each select='/vCard/URL'>
?url><uri><xsl:value-of select='.'/></uri></url>
</xsl:for-each>

<!-- NOTE: vcard-temp allowed a CLASS element, but it was removed from vCard4 -->

<!-- KEY can be included one or more times -->
<xsl:for-each select='/vCard/KEY'>
?key><text><xsl:value-of select='CRED'/></text></key>
</xsl:for-each>

<!-- DESC can be included one or more times -->
<!-- this existed in vcard-temp but not vCard3 -->
<!-- mapped to the NOTE element -->
<xsl:for-each select='/vCard/DESC'>
<note><text><xsl:value-of select='.'/></text></note>
</xsl:for-each>

</vcard>
</xsl:template>

<xsl:template match='N'>
<n>
<xsl:variable name='FAMILY.count' select='count(FAMILY)'/>
<xsl:variable name='GIVEN.count' select='count(GIVEN)'/>
<xsl:variable name='MIDDLE.count' select='count(MIDDLE)'/>
<xsl:variable name='PREFIX.count' select='count(PREFIX)'/>

24
<xsl:variable name='SUFFIX.count' select='count(SUFFIX)'/>
<xsl:if test='$FAMILY.count=1'>
  <surname><xsl:value-of select='FAMILY'/></surname>
</xsl:if>
<xsl:if test='$GIVEN.count=1'>
  <given><xsl:value-of select='GIVEN'/></given>
</xsl:if>
<xsl:if test='$MIDDLE.count=1'>
  <additional><xsl:value-of select='MIDDLE'/></additional>
</xsl:if>
<xsl:if test='$PREFIX.count=1'>
  <prefix><xsl:value-of select='PREFIX'/></prefix>
</xsl:if>
<xsl:if test='$SUFFIX.count=1'>
  <suffix><xsl:value-of select='SUFFIX'/></suffix>
</xsl:if>

<xsl:if test='$FAMILY.count=1'>
  <surname><xsl:value-of select='FAMILY'/></surname>
</xsl:if>
<xsl:if test='$GIVEN.count=1'>
  <given><xsl:value-of select='GIVEN'/></given>
</xsl:if>
<xsl:if test='$MIDDLE.count=1'>
  <additional><xsl:value-of select='MIDDLE'/></additional>
</xsl:if>
<xsl:if test='$PREFIX.count=1'>
  <prefix><xsl:value-of select='PREFIX'/></prefix>
</xsl:if>
<xsl:if test='$SUFFIX.count=1'>
  <suffix><xsl:value-of select='SUFFIX'/></suffix>
</xsl:if>

<xsl:variable name='HOME.count' select='count(HOME)'/>
<xsl:variable name='WORK.count' select='count(WORK)'/>

<!-- NOTE: vcard-temp allowed address types of POSTAL, PARCEL, DOM, and INTL, but they were removed from vCard4 -->
<xsl:variable name='PREF.count' select='count(PREF)'/>
<xsl:variable name='POBOX.count' select='count(POBOX)'/>
<xsl:variable name='EXTADD.count' select='count(EXTADD)'/>
<xsl:variable name='STREET.count' select='count(STREET)'/>
<xsl:variable name='LOCALITY.count' select='count(LOCALITY)'/>
<xsl:variable name='REGION.count' select='count(REGION)'/>
<xsl:variable name='PCODE.count' select='count(PCODE)'/>

<!-- NOTE: yes, vcard-temp has CTRY, not COUNTRY -->
<xsl:variable name='CTRY.count' select='count(CTRY)'/>

<!-- first we count the number of vCard TYPE parameters -->
<xsl:variable name='TYPE.count' select='$HOME.count+_+$WORK.count'/>

<!-- now we output all the parameters -->
<xsl:if test='$TYPE.count &gt; 0'>
  <parameters>
    <type>
xsl:if test='$HOME.count=1' 
  <text>home</text>
</xsl:if>
<xsl:if test='$WORK.count=1' 
  <text>work</text>
</xsl:if>
</type>
<xsl:if test='$PREF.count=1' 
  <pref><integer>1</integer></pref>
</xsl:if>
</parameters>
<xsl:if test='$POBOX.count=1' 
  <pobox><xsl:value-of select='POBOX'/></pobox>
</xsl:if>
<xsl:if test='$EXTADD.count=1' 
  <ext><xsl:value-of select='EXTADD'/></ext>
</xsl:if>
<xsl:if test='$STREET.count=1' 
  <street><xsl:value-of select='STREET'/></street>
</xsl:if>
<xsl:if test='$LOCALITY.count=1' 
  <locality><xsl:value-of select='LOCALITY'/></locality>
</xsl:if>
<xsl:if test='$REGION.count=1' 
  <region><xsl:value-of select='REGION'/></region>
</xsl:if>
<xsl:if test='$PCODE.count=1' 
  <code><xsl:value-of select='PCODE'/></code>
</xsl:if>
<xsl:if test='$CTRY.count=1' 
  <country><xsl:value-of select='CTRY'/></country>
</xsl:if>
</adr>
<xsl:template match='TEL'>
 <tel>
  <xsl:variable name='HOME.count' select='count(HOME)'/>
  <xsl:variable name='WORK.count' select='count(WORK)'/>
  <xsl:variable name='TEXT.count' select='count(TEXT)'/>
  <xsl:variable name='VOICE.count' select='count(VOICE)'/>
  <xsl:variable name='FAX.count' select='count(FAX)'/>
  <xsl:variable name='CELL.count' select='count(CELL)'/>
  <xsl:variable name='VIDEO.count' select='count(VIDEO)'/>
  <xsl:variable name='PAGER.count' select='count(PAGER)'/>
  <xsl:variable name='TEXTPHONE.count' select='count(TEXTPHONE)'/>
  <text>NOTE: vcard-temp allowed telephony types of MSG, BBS,
MODEM, ISDN, and PCS but they were removed from vCard4 -->

```xml
<xsl:variable name='PREF.count' select='count(PREF)'/>
<xsl:variable name='NUMBER.count' select='count(NUMBER)'/>
<!-- first we count the number of vCard TYPE parameters -->
<xsl:variable name='TYPE.count'
    select='HOME.count + WORK.count + TEXT.count + VOICE.count + FAX.count + CELL.count + VIDEO.count + PAGER.count + TEXTPHONE.count'/>

<!-- now we output all the parameters -->
<xsl:if test='$TYPE.count &gt; 0'>
  <parameters>
    <type>
      <xsl:if test='$HOME.count=1'>
        <text>home</text>
      </xsl:if>
      <xsl:if test='$WORK.count=1'>
        <text>work</text>
      </xsl:if>
      <xsl:if test='$TEXT.count=1'>
        <text>text</text>
      </xsl:if>
      <xsl:if test='$VOICE.count=1'>
        <text>voice</text>
      </xsl:if>
      <xsl:if test='$FAX.count=1'>
        <text>fax</text>
      </xsl:if>
      <xsl:if test='$CELL.count=1'>
        <text>cell</text>
      </xsl:if>
      <xsl:if test='$VIDEO.count=1'>
        <text>video</text>
      </xsl:if>
      <xsl:if test='$PAGER.count=1'>
        <text>pager</text>
      </xsl:if>
      <xsl:if test='$TEXTPHONE.count=1'>
        <text>texphone</text>
      </xsl:if>
    </type>
  </parameters>
</xsl:if>
```
<xsl:template match='EMAIL'>
  <email>
    <xsl:variable name='HOME.count' select='count(HOME)'/>
    <xsl:variable name='WORK.count' select='count(WORK)'/>
    <!-- NOTE: vcard-temp allowed email types of INTERNET and X400, but they were never in vCard3 -->
    <xsl:variable name='PREF.count' select='count(PREF)'/>
    <xsl:variable name='USERID.count' select='count(USERID)'/>
    <!-- first we count the number of vCard TYPE parameters -->
    <xsl:variable name='TYPE.count' select='HOME.count + WORK.count'/>
    <!-- now we output all the parameters -->
    <xsl:if test='TYPE.count &gt; 0'>
      <parameters>
        <type>
          <xsl:if test='HOME.count=1'>
            <text>home</text>
          </xsl:if>
          <xsl:if test='WORK.count=1'>
            <text>work</text>
          </xsl:if>
        </type>
        <xsl:if test='PREF.count=1'>
          <pref><integer>1</integer></pref>
        </xsl:if>
      </parameters>
    </xsl:if>
    <xsl:if test='USERID.count'>
      <text><xsl:value-of select='USERID/></text>
    </xsl:if>
  </email>
</xsl:template>
12.2 Example of vcard-temp Data

```xml
<vCard>
  <FN>Peter Saint-Andre</FN>
  <N>
    <FAMILY>Saint-Andre</FAMILY>
    <GIVEN>Peter</GIVEN>
    <MIDDLE/>
  </N>
  <NICKNAME>stpeter</NICKNAME>
  <NICKNAME>psa</NICKNAME>
  <PHOTO><EXTVAL>http://stpeter.im/images/stpeter_oscon.jpg</EXTVAL></PHOTO>
  <PHOTO><EXTVAL>http://stpeter.im/images/stpeter_hell.jpg</EXTVAL></PHOTO>
  <BDAY>1966-08-06</BDAY>
  <ADR/>
    <PREF/>
      <EXTADD>Suite 600</EXTADD>
      <STREET>1899 Wynkoop Street</STREET>
      <LOCALITY>Denver</LOCALITY>
      <REGION>CO</REGION>
      <PCODE>80202</PCODE>
      <CTRY>USA</CTRY>
    </EXTADD>
  </ADR>
  <ADR/>
    <HOME/>
    <EXTADD/>
      <STREET/>
        <LOCALITY>Parker</LOCALITY>
        <REGION>CO</REGION>
        <PCODE>80138</PCODE>
        <CTRY>USA</CTRY>
      </EXTADD>
  </ADR>
  <TEL><PREF/><WORK>/</WORK><VOICE>/<NUMBER>+1-303-308-3282</NUMBER>/</VOICE></TEL>
  <TEL><WORK>/</WORK><FAX/><NUMBER>+1-303-308-3219</NUMBER>/</FAX></TEL>
  <TEL><CELL>/</CELL><VOICE><TEXT>/</TEXT><NUMBER>+1-720-256-6756</NUMBER>/</VOICE></TEL>
  <TEL><HOME>/</HOME><VOICE/><NUMBER>+1-303-555-1212</NUMBER>/</VOICE></TEL>
  <EMAIL><INTERNET>/</INTERNET><PREF/></EMAIL>
    <USERID>stpeter@jabber.org</USERID/>
  </EMAIL>
  <EMAIL><WORK/><USERID>psaintan@cisco.com</USERID>></EMAIL>
  <JABBERID>stpeter@jabber.org</JABBERID>
  <TZ>America/Denver</TZ>
  <GEO><LAT>39.59</LAT><LON>-105.01</LON></GEO>
</vCard>
```
12 MIGRATION TOOLS

<TITLE>Executive Director</TITLE>

<ROLE>Patron Saint</ROLE>

<TYPE>image/jpeg</TYPE>

<BINVAL>/9j/4AAQSkZJRgABAQEASABIAAD // gAXQ3Jl1XZR1ZCB3aXR0IFRoZSBHSU1Q/9

sAQwIBgYHBgUI

BwcHCQkJlCgwUDQwLCwwZEhMPFB0aHx4dGhwcICQuJyAiLCMhcG3KSwMTQ0NB8nOT4MjwuMzQy

/9

sAQwEJCQkMCwwYDQ0YMiEcITIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy

MjIyMjIyMjIyMjIyY/8AAEQgAgQB7AwEiAAIRAQMRAF/

EABoAAAMAAAAAAABAAAAAQIDAARIAQAQRBIhMUEGE1EiYXGBkRQyobEHFSmNEJScsEk

YthW8jVDC8Lh/8QAQGEAAEWAQAAAAAAABAAAAAQIEAwAF/8

QAJBEAgICAgMAAAAAAA

AAECEQQmHeJEEQSJReZJCYXH/2g AMAwEAAhEDEQA/

AOFC2c0KGSYz2pDFJiinYMsCMfhj8aya6gdvZj9

ryg6A8eYuU0RXQkJrzdShyZGQy0GXAPv/

MV6ulUp8qgntRyhAfHJ3Y5wOoxxXncbdia9zp/6xQx

KaF4hK208MwX89amOeMDjT3VLi1EaeVGlE14WRCQVwAD7PY5/

CQYbfcdfUPvbJT52FAGHGele

1nmXg0w6/IMV4LnHup/4c8L3/1WfbbioFoP3tD/dH/kfkd00kootUK/ALWes8S3JEPZGA4/z2oM

T7++dY7azuU1A0PS8Jfb98PAmu1/HvOz+HtPjVPy7YX846yXayM+5

e1L1tDa2sKxxwRxoBgkigD6U

nJGn4a7OB/G/0Y67rlxuiJRptojdcXS5c/2R9SfoFXY/DvgzqVciLIZW/

m3Uavbgb5f9vZ9P1

9TVCBaw5RfiODQ93Fdi3b9XyxixT5iE59wNPGSEnja6N5VcyZhYhSLOcuvuNe0y0JDdT0/8ATG8

1Hahc6g8x7xX0ku9TyzhB4dKEA6AZx1prJz2n7K5W83L5W1s3H9yXvSfGl13NdNvmld8/wBRoMHH

NbM5Hw5oAoIjalHLPaF5909KL218NS3byXtje9w+

GdijvBj8MUyk5ixAAPZ61rtreDUba05kiTeV2

HJ7r7P8AiUG5OX2ydzBE1zInAKFR6k9a+XFrH7qWczjGVUKASVPHu+yVLftckzPu+62

UE256Y7

1S+fkj1DxpDaXEe5YHYldc9d0GAdfnFQLE+

VDdC22XDaQM7GDEC6H1752zgjimcv6PffmvXMcy

6SttFtB8+4dIEIz0PPfsUe107eEMVoyzsfPs28Sxdev3Rh5h5jP52LyzSSnqS7E/

vnKiczerf1j

+ibSrlCaXWNBf9dAeaa9d93P+6

Qjp8AksNmt4L0HgtoOiXhVj6CldrsSLOMt7qNhuIdJyPSSc

k23RFg14yqphkcQICmCiQues9TsoAKDP1osX0T1oWO0x2J0eu2rPiki3G0/Hxr0L5F/

nFGwNB2q

2Unw2+MAXCc8v/g/GopkaNijqVdWlYHqDVSt9uDyt2pRrKrJJDgycyrh/7h/wDlawL2Lh78Q86
12 MIGRATION TOOLS

1jXzHDqP5ZBnW7i1FrL5yQ0mfLZAw+
  fNN7e3jV17DrxS@qoha2aNv0DUokEMqX5IoXIDHibIHHu
qQvrK5sJRHPEVcaA4PZvgasb2g8XIVFDD+YcOIoZGC
  ga2yuJlG3C46qT3geFvYa0TOKQXDKnw10T0y
F93f51To5ihYQAWJ5B102ipPgWiaQnLsnNicdAj6mqaalDuG2/wAqjqewAqFp0zRHKNR/
eCte1/x
ArKyvZmEah+9Wvp/gz8Kysp10ciz/3Buvi6/dwv1S0x4maF/6wvz/ADN0tb/
eF8xjv1Uzc
aPYXq3BEmUbcrFxYFZWUMQch0LTP4SH+2X/n4v1ZXqx/VEz7EF019/50
bfp8SpGzrKyuA0P43
0v//+ogw7DD0gysrrzcv7joro/9k=
</BINVAL>
</LOGO>
</ORG>
</ORGNAME>XMPP Standards Foundation</ORGNAME>
</ORGUNIT>
</ORG>
</URL>https://stpeter.im/</URL>
</URL>http://www.saint-andre.com/</URL>
</KEY>
</CRED>

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG/MacGPG2 v2.0.18 (Darwin)
mQINBFETdzsBEAC8FOFv1N3JZjIIxN6cKD475KS9CHDpYepegc0IPnL5eY1DCHeh/
/IwS1S7CePtmvIbNoV9FSI4P0UKnsx0AX6L4VEdAr/LUIhgJKJq+gspbLqBEILh
g3cHe666Wl59arbOKC47T7Kl8miPBFC6E3A4Lq1L+eeu06UcLhKgoYkm0xjdi
WrMgKTNvphcydLKpm/z0ZQ08zRqg1PulTeCrxZYNjhxLkVFN2xy0Uoz0S7p555KV
fx5Z7uQisr8pxytdl6SpTzo6SHgKbV15uzqRxhsJojiTOxFFWznAjaS5FU0U0Rq
CK1G5c5NUA1T8NFtv0ktstaxWDL1ELDVQPy1m7mtzo+VREG+0xmu6AJoMo/GHb1W1U
U7MIy9CiuMLsp/hLRfUiosQLVZ85wuLQJunjPe3tK815UcIXAcq01VqIaODQfe
uL0XTJF8YHpdHyT/ZmIl127BKGbA0y8y7uf07wJ9D3gUaizw29fdJFjVW7iak7ATTw
1flf1zmWdn+MzybHOGUGM5xShsa8bDSieirQ2QoLdn2Fip7KMBTJ2+GISrFsJTN/
QVvmj0DXXadHumU24QgmZdkge35n129yzXn9NczqzGRlo6V01L3LX6cSbhi5H7
GgWY6CABp1MogV6475n9FV05DrIQ4S50yqKiA30P5aKrIPr2TNa4k1wARAQAB
tCQZXR1cBTYW1udC1bBrRyZ5A8C3RwXZr1ckBzdHB1dGVyLmlPtokCQQ7TAQIA
IwUCURMPWobInbAwcLCQgHAWIBHBUlAqKCCWQAAbMABh4BAbeAAAOJeEOGjoErxa2p
6bgQAKpxu07cMD0Lc4+EGBH19NWXlyvbQeVfuHUYZaLkKPhrMzWjIowpyIS5R
9QQZx1eLCEvaojoeVX6kX50MGeSzFkFjZy3j7jBWi+Ybr7F7FxyY2BAXxk49b1n6
Ci9LmBrmYfmaeXadtDNPITZ9NoU9bV50n0R036EvteEHAvQvQdJ07h1z6+Cq1Z
qBq9jInmmknF4cQ53nSQQaKL91av4fQx17SFpKnxqhuX3JNLGnVo3wl+j/FcoK
i0trmHxCI3kc/bx6g32pRjHEPX0ALMBHmzU2uc0+TE0ZC9c6g8yXUACWdnCFW
ybeIeb6z651ML3kAAVaQ0H/GqncnMGN0M0BatnwlTdz/vkLojly7qPcP01UFX
v5491xPJrH0HDwtxP6Wux88fcqhTM6H3PZVtusj2ornKVv+Y0GLsMMTCrXJRg/
0A1y0V77plrpzFg0SnAx0lDIZ6B+76jr1HtUHiWgo/4nF+DN6BII1CZ6j6xxjx
462cu02zuhI1LK2pazA0uFtBmx0uJhZk/KPK2Fay/41x7pvVowCR4uIKsNJKL
PS7Ea4BUuENfd/9LqOpW11BBSe98RLMI8sXkcigc3UXMvda9110Yh+0a1bP1
NasznmbHbuiCsgPGBImJurszEgckPWp/dNeyr6M1FMyfaeuQ1INBFETdzsBEADB
z0sEHPumhKrUjH9TeK87dn5P/Yh/L/HptGCgkr0TL/C+Kyd3HyteMEF061Pnms
12 MIGRATION TOOLS

More information about me is located on my personal website: https://stpeter.im/
<nickname/>
<photo>
  <uri>http://stpeter.im/images/stpeter_oscon.jpg</uri>
</photo>
<photo>
  <uri>http://stpeter.im/images/stpeter_hell.jpg</uri>
</photo>
<bday>
  <date>1966-08-06</date>
</bday>
<adr>
  <parameters>
    <type><text>work</text></type>
    <pref><integer>1</integer></pref>
  </parameters>
  <ext>Suite 600</ext>
  <street>1899 Wynkoop Street</street>
  <locality>Denver</locality>
  <region>CO</region>
  <code>80202</code>
  <country>USA</country>
</adr>
<adr>
  <parameters>
    <type><text>home</text></type>
    <ext></ext>
  </parameters>
  <street></street>
  <locality>Parker</locality>
  <region>CO</region>
  <code>80138</code>
  <country>USA</country>
</adr>
<tel>
  <parameters>
    <type><text>work</text><text>voice</text></type>
    <pref><integer>1</integer></pref>
  </parameters>
  <uri>tel:+1-303-308-3282</uri>
</tel>
<tel>
  <parameters>
    <type><text>work</text><text>fax</text></type>
  </parameters>
  <uri>tel:+1-303-308-3219</uri>
</tel>
<tel>
  <parameters>
    <type><text>cell</text><text>voice</text><text>text</text></type>
  </parameters>
</tel>
13 Acknowledgements

Thanks to Dave Cridland, Todd Herman, Joe Hildebrand, Waqas Hussain, and Matt Miller for their feedback.