XEP-0054: vcard-temp

Peter Saint-Andre
mailto:xsf@stpeter.im
xmpp:peter@jabber.org
http://stpeter.im/

2008-07-16
Version 1.2

Status	Type	Short Name
Active	Historical	vcard-temp

This specification provides canonical documentation of the vCard-XML format currently in use within the Jabber community.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2018 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).

Contents

1 Introduction

This specification documents the vCard-XML format currently in use within the Jabber community. A future specification will recommend a standards-track protocol to supersede this informational document.

The basic functionality is for a user to store and retrieve an XML representation of his or her vCard using the data storage capabilities native to all existing Jabber server implementations. This is done by by sending an <iq/> of type "set" (storage) or "get" (retrieval) to one's Jabber server containing a <vCard/> child scoped by the 'vcard-temp' namespace, with the <vCard/> element containing the actual vCard-XML elements as defined by the vCard-XML DTD. Other users may then view one's vCard information.

2 History

vCards are an existing and widely-used standard for personal user information storage, somewhat like an electronic business card. The vCard format is defined in RFC 2426 [1].

In 1998 and 1999, Frank Dawson submitted four revisions of an Internet-Draft proposing to represent the standard vCard format in XML. When the jabberd open-source project was originally looking for a method to store personal user information, the most recent revision consulted by the jabberd developers was draft-dawson-vcard-xml-dtd-01 [2]. He also submitted a -02 revision on November 15, 1998 [3] and a -03 revision on June 22, 1999 [4].

Unfortunately, Dawson's proposal did not move forward within the IETF's standards process. For reasons now lost in the mists of time, the Jabber project continued to use the DTD from draft-dawson-vcard-xml-dtd-01, making two small modifications to adapt it for use within the Jabber community (adding the JABBERID and DESC elements) but also specifying element names in all caps rather than lowercase as defined in draft-dawson-vcard-xml-dtd-01. In addition, the Jabber community followed the usage (but not DTD) in that draft regarding version information, including it as an attribute of the vCard element rather than as a child element. This format was implemented within the Jabber community under the 'vcard-temp' namespace.

---

[1] RFC 2426: vCard MIME Directory Profile <http://tools.ietf.org/html/rfc2426>.

[2] This document is archived at <http://www.watersprings.org/pub/id/draft-dawson-vcard-xml-dtd-01.txt>.

[3] This document is archived at <http://www.watersprings.org/pub/id/draft-dawson-vcard-xml-dtd-02.txt>.

[4] This document is archived at <http://www.watersprings.org/pub/id/draft-dawson-vcard-xml-dtd-03.txt>.

3   Use Cases

3.1   Retrieving One's vCard

A user retrieves his or her own vCard by sending an IQ-get with no 'to' address and containing a <vCard/> child element qualified by the 'vcard-temp' namespace.

Listing 1: vCard Retrieval Request

```
<iq from='stpeter@jabber.org/roundabout'
    id='v1'
    type='get'>
  <vCard xmlns='vcard-temp'/>
</iq>
```

If a vCard exists for the user, the server MUST return in an IQ-result:

Listing 2: Server Returns vCard

```
<iq id='v1'
    to='stpeter@jabber.org/roundabout'
    type='result'>
  <vCard xmlns='vcard-temp'>
    <FN>Peter Saint-Andre</FN>
    <N>
      <FAMILY>Saint-Andre</FAMILY>
      <GIVEN>Peter</GIVEN>
      <MIDDLE/>
    </N>
    <NICKNAME>stpeter</NICKNAME>
    <URL>http://www.xmpp.org/xsf/people/stpeter.shtml</URL>
    <BDAY>1966-08-06</BDAY>
    <ORG>
      <ORGNAME>XMPP Standards Foundation</ORGNAME>
      <ORGUNIT/>
    </ORG>
    <TITLE>Executive Director</TITLE>
    <ROLE>Patron Saint</ROLE>
    <TEL><WORK/><VOICE/><NUMBER>303-308-3282</NUMBER></TEL>
    <TEL><WORK/><FAX/><NUMBER/></TEL>
    <TEL><WORK/><MSG/><NUMBER/></TEL>
    <ADR>
      <WORK/>
      <EXTADD>Suite 600</EXTADD>
      <STREET>1899 Wynkoop Street</STREET>
      <LOCALITY>Denver</LOCALITY>
      <REGION>CO</REGION>
      <PCODE>80202</PCODE>
      <CTRY>USA</CTRY>
```

```
        </ADR>
        <TEL><HOME/><VOICE/><NUMBER>303-555-1212</NUMBER></TEL>
        <TEL><HOME/><FAX/><NUMBER/></TEL>
        <TEL><HOME/><MSG/><NUMBER/></TEL>
        <ADR>
          <HOME/>
          <EXTADD/>
          <STREET/>
          <LOCALITY>Denver</LOCALITY>
          <REGION>CO</REGION>
          <PCODE>80209</PCODE>
          <CTRY>USA</CTRY>
        </ADR>
        <EMAIL><INTERNET/><PREF/><USERID>stpeter@jabber.org</USERID></
            EMAIL>
        <JABBERID>stpeter@jabber.org</JABBERID>
        <DESC>
          More information about me is located on my
          personal website: http://www.saint-andre.com/
        </DESC>
    </vCard>
</iq>
```

If no vCard exists, the server MUST return a stanza error (which SHOULD be <item-not-found/>) or an IQ-result containing an empty <vCard/> element.

Listing 3: No vCard (item-not-found)

```
<iq id='v1'
    to='stpeter@jabber.org/roundabout'
    type='error'>
  <vCard xmlns='vcard-temp'/>
  <error type='cancel'>
    <item-not-found xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

Listing 4: No vCard (empty element)

```
<iq id='v1'
    to='stpeter@jabber.org/roundabout'
    type='result'>
  <vCard xmlns='vcard-temp'/>
</iq>
```

3.2 Updating One's vCard

A user may publish or update his or her vCard by sending an IQ of type "set" with no 'to' address, following the format in the previous use case.

Listing 5: vCard Update Request

```xml
<iq id='v2' type='set'>
  <vCard xmlns='vcard-temp'>
    <FN>Peter Saint-Andre</FN>
    <N>
      <FAMILY>Saint-Andre</FAMILY>
      <GIVEN>Peter</GIVEN>
      <MIDDLE/>
    </N>
    <NICKNAME>stpeter</NICKNAME>
    <URL>http://www.xmpp.org/xsf/people/stpeter.shtml</URL>
    <BDAY>1966-08-06</BDAY>
    <ORG>
      <ORGNAME>XMPP Standards Foundation</ORGNAME>
      <ORGUNIT/>
    </ORG>
    <TITLE>Executive Director</TITLE>
    <ROLE>Patron Saint</ROLE>
    <TEL><WORK/><VOICE/><NUMBER>303-308-3282</NUMBER></TEL>
    <TEL><WORK/><FAX/><NUMBER/></TEL>
    <TEL><WORK/><MSG/><NUMBER/></TEL>
    <ADR>
      <WORK/>
      <EXTADD>Suite 600</EXTADD>
      <STREET>1899 Wynkoop Street</STREET>
      <LOCALITY>Denver</LOCALITY>
      <REGION>CO</REGION>
      <PCODE>80202</PCODE>
      <CTRY>USA</CTRY>
    </ADR>
    <TEL><HOME/><VOICE/><NUMBER>303-555-1212</NUMBER></TEL>
    <TEL><HOME/><FAX/><NUMBER/></TEL>
    <TEL><HOME/><MSG/><NUMBER/></TEL>
    <ADR>
      <HOME/>
      <EXTADD/>
      <STREET/>
      <LOCALITY>Denver</LOCALITY>
      <REGION>CO</REGION>
      <PCODE>80209</PCODE>
      <CTRY>USA</CTRY>
    </ADR>
    <EMAIL><INTERNET/><PREF/><USERID>stpeter@jabber.org</USERID></
        EMAIL>
    <JABBERID>stpeter@jabber.org</JABBERID>
    <DESC>
      Check out my blog at https://stpeter.im/
```

```
      </DESC>
   </vCard>
</iq>
```

The server then returns an IQ-result (or an IQ-error).

Listing 6: Server Returns Success

```
<iq id='v2'
    to='stpeter@jabber.org/roundabout'
    type='result'/>
```

Notice that the previous IQ-set included only one changed element (the <DESC/> element). Currently there is no method for partial updates of a vCard, and the entire vCard must be sent to the server in order to update any part of the vCard.

If a user attempts to perform an IQ set on another user's vCard (i.e., by setting a 'to' address to a JID other than the sending user's bare JID), the server MUST return a stanza error, which SHOULD be <forbidden/> or <not-allowed/>.

Listing 7: Entity Attempts to Modify Another Entity's vCard

```
<iq id='v2'
    to='stpeter@jabber.org/roundabout'
    type='error'>
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

3.3 Viewing Another User's vCard

A user may view another user's vCard by sending an IQ of type "get" to the other user's bare JID.

Listing 8: Requesting Another User's vCard

```
<iq from='stpeter@jabber.org/roundabout'
    id='v3'
    to='jer@jabber.org'
    type='get'>
  <vCard xmlns='vcard-temp'/>
</iq>
```

In accordance with XMPP Core [5], a compliant server MUST respond on behalf of the requestor and not forward the IQ to the requestee's connected resource.

---

[5]RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

Listing 9: Receiving Another User's vCard

```
<iq from='jer@jabber.org'
    to='stpeter@jabber.org/roundabout'
    type='result'
    id='v3'>
  <vCard xmlns='vcard-temp'>
    <FN>JeremieMiller</FN>
    <N>
      <GIVEN>Jeremie</GIVEN>
      <FAMILY>Miller</FAMILY>
      <MIDDLE/>
    </N>
    <NICKNAME>jer</NICKNAME>
    <EMAIL><INTERNET/><PREF/><USERID>jeremie@jabber.org</USERID></
        EMAIL>
    <JABBERID>jer@jabber.org</JABBERID>
  </vCard>
</iq>
```

If no vCard exists or the user does not exist, the server MUST return a stanza error, which SHOULD be either <service-unavailable/> or <item-not-found/> (but the server MUST return the same error condition in both cases to help prevent directory harvesting attacks).

Listing 10: No vCard

```
<iq id='v3'
    to='stpeter@jabber.org/roundabout'
    type='error'>
  <vCard xmlns='vcard-temp'/>
  <error type='cancel'>
    <service-unavailable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

Note: The use of vCards is not limited to accounts associated with human users. For example, an XMPP server could itself have a vCard that defines the server's hosting organization, physical location, and relevant contact addresses.

4 Determining Support

If an entity supports the vcard-temp protocol, it MUST report that by including a service discovery feature of "vcard-temp" (see Protocol Namespaces regarding issuance of one or more permanent namespaces) in response to a Service Discovery (XEP-0030) [6] information request:

---

[6]XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.

Listing 11: Service Discovery Information Request

```
<iq type='get'
    from='juliet@capulet.lit/balcony'
    to='capulet.lit'
    id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 12: Service Discovery Information Response

```
<iq type='result'
    from='capulet.lit'
    to='juliet@capulet.lit/balcony'
    id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='vcard-temp'/>
    ...
  </query>
</iq>
```

This information can also be encapsulated via Entity Capabilities (XEP-0115) [7] for entities who share presence.

5 Security Considerations

The vCard information published to one's Jabber 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).

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [8].

---

[7] XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.

[8] 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/>.

7 XMPP Registrar Considerations

7.1 Protocol Namespaces

The XMPP Registrar [9] includes the 'vcard-temp' namespace in its registry of official namespaces (see <https://xmpp.org/registrar/namespaces.html>).

7.2 URI Query Types

As authorized by XMPP URI Query Components (XEP-0147) [10], the XMPP Registrar maintains a registry of queries and key-value pairs for use in XMPP URIs (see <https://xmpp.org/registrar/querytypes.html>).
The "vcard" querytype is registered as a vCard-related action.

Listing 13: vCard Action: IRI/URI

```
xmpp:romeo@montague.net?vcard
```

Listing 14: vCard Action: Resulting Stanza

```xml
<iq to='romeo@montague.net' type='get'>
  <vCard xmlns='vcard-temp'/>
</iq>
```

The following submission registers the "vcard" querytype.

```xml
<querytype>
  <name>vcard</name>
  <proto>vcard-temp</proto>
  <desc>enables retrieval of an entity's vCard data</desc>
  <doc>XEP-0054</doc>
</querytype>
```

8 Implementation Notes

Note the following:

- The correct capitalization of the wrapper element is <vCard/> (and XML element names are case-sensitive).

---

[9]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/>.

[10]XEP-0147: XMPP URI Query Components <https://xmpp.org/extensions/xep-0147.html>.

- All elements within the <vCard/> element MUST be in ALL CAPS (even though this is at odds with draft-dawson-vcard-xml-dtd-01).

- The country abbreviation is contained in a <CTRY/> element, *not* a <COUNTRY/> element (even though this is at odds with draft-dawson-vcard-xml-dtd-01).

- Phone numbers MUST be contained in a <NUMBER> element, not included as CDATA within the <TEL/> element.

- If no telephone number is included in a <TEL/> element, an empty <NUMBER/> child MUST be included.

- Email addresses MUST be contained in a <USERID> element, not included as CDATA within the <EMAIL/> element.

- Some Jabber implementations add a 'version' attribute to the <vCard/> element, with the value set at "2.0" or "3.0". The DTD is incorrect, and the examples in draft-dawson-vcard-xml-dtd-01 clearly show that version information is to be included by means of a 'version' attribute, not the <VERSION/> element as defined in the DTD. However, to conform to draft-dawson-vcard-xml-dtd-01, the value should be "3.0", not "2.0".

9 DTD

The following DTD is a slightly modified version of that contained in draft-dawson-vcard-xml-dtd-01. The only modifications were to add the JABBERID and DESC elements.

```xml
<?xml version="1.0" encoding="UTF-8"?>

<!--
Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and
furnished to others, and derivative works that comment
on or otherwise explain it or assist in its implmentation
may be prepared, copied, published and distributed, in whole
or in part, without restriction of any kind, provided that
the above copyright notice and this paragraph are included
on all such copies and derivative works.

However, this document itself may not be modified in any
way, such as by removing the copyright notice or references
to the Internet Society or other Internet organizations,
except as needed for the purpose of developing Internet
standards in which case the procedures for copyrights
defined in the Internet Standards process MUST be followed,
or as required to translate it into languages other than English.
```

```
<!-- ==== -->

<!-- NOTE: the following root element is not used in the
           modified vcard-temp DTD published by the Jabber
           project (now XMPP Standards Foundation) and is
           included here only for historical purposes;
           implementations that comply with vcard-temp must
           specify the root element as vCard, not xCard. -->

<!-- Root element and container for one
  or more vCard objects -->
<!ELEMENT xCard (vCard)+>

<!-- Individual vCard container -->
<!ELEMENT vCard (
  (VERSION, FN, N),
  (NICKNAME?,
  PHOTO?,
  BDAY?,
  ADR?,
  LABEL?,
  TEL?,
  EMAIL?,
  JABBERID?,
  MAILER?,
  TZ?,
  GEO?,
  TITLE?,
  ROLE?,
  LOGO?,
  AGENT?,
  ORG?,
  CATEGORIES?,
  NOTE?,
  PRODID?,
  REV?,
```

```
  SORT-STRING?,
  SOUND?,
  UID?,
  URL?,
  CLASS?,
  KEY?,
  DESC?
)*)>

  <!-- vCard specification version property.
    This MUST be 2.0, if the document conforms to RFC 2426. -->
  <!ELEMENT VERSION (#PCDATA)>

  <!-- Formatted or display name property. -->
  <!ELEMENT FN (#PCDATA)>

  <!-- Structured name property. Name components with multiple
    values must be specified as a comma separated
        list of values. -->
  <!ELEMENT N ( FAMILY?, GIVEN?, MIDDLE?, PREFIX?, SUFFIX?)>

    <!ELEMENT FAMILY (#PCDATA)>
    <!ELEMENT GIVEN (#PCDATA)>
    <!ELEMENT MIDDLE (#PCDATA)>
    <!ELEMENT PREFIX (#PCDATA)>
    <!ELEMENT SUFFIX (#PCDATA)>

  <!-- Nickname property. Multiple nicknames must be
    specified as a comma separated list value. -->
  <!ELEMENT NICKNAME (#PCDATA)>

  <!-- Photograph property. Value is either a BASE64 encoded
    binary value or a URI to the external content. -->
  <!ELEMENT PHOTO ((TYPE, BINVAL) | EXTVAL)>

  <!-- Birthday property. Value must be an ISO 8601 formatted
    date or date/time value. -->
  <!ELEMENT BDAY (#PCDATA)>

  <!-- Structured address property. Address components with
    multiple values must be specified as a comma separated list
    of values. -->
  <!ELEMENT ADR (
    HOME?,
    WORK?,
    POSTAL?,
    PARCEL?,
    (DOM | INTL)?,
    PREF?,
```

```
    POBOX?,
    EXTADD?,
    STREET?,
    LOCALITY?,
    REGION?,
    PCODE?,
    CTRY?
)>

  <!ELEMENT POBOX (#PCDATA)>
  <!ELEMENT EXTADD (#PCDATA)>
  <!ELEMENT STREET (#PCDATA)>
  <!ELEMENT LOCALITY (#PCDATA)>
  <!ELEMENT REGION (#PCDATA)>
  <!ELEMENT PCODE (#PCDATA)>
  <!ELEMENT CTRY (#PCDATA)>

<!-- Address label property. -->
<!ELEMENT LABEL (
  HOME?,
  WORK?,
  POSTAL?,
  PARCEL?,
  (DOM | INTL)?,
  PREF?,
  LINE+
)>

  <!-- Individual label lines. -->
  <!ELEMENT LINE (#PCDATA)>

<!-- Telephone number property. -->
<!ELEMENT TEL (
  HOME?,
  WORK?,
  VOICE?,
  FAX?,
  PAGER?,
  MSG?,
  CELL?,
  VIDEO?,
  BBS?,
  MODEM?,
  ISDN?,
  PCS?,
  PREF?,
  NUMBER
)>
```

```
   <!-- Phone number value. -->
   <!ELEMENT NUMBER (#PCDATA)>

 <!-- Email address property. Default type is INTERNET. -->
 <!ELEMENT EMAIL (
   HOME?,
   WORK?,
   INTERNET?,
   PREF?,
   X400?,
   USERID
 )>

   <!ELEMENT USERID (#PCDATA)>

 <!-- NOTE: the following element was added by the Jabber
           project (now XMPP Standards Foundation) to
           handle Jabber IDs; the value must be in the
           form of user@host -->

 <!ELEMENT JABBERID (#PCDATA)>

 <!-- Mailer (e.g., Mail User Agent Type) property. -->
 <!ELEMENT MAILER (#PCDATA)>

 <!-- Time zone's Standard Time UTC offset. Value must be an
     ISO 8601 formatted UTC offset. -->
  <!ELEMENT TZ (#PCDATA)>

  <!-- Geographical position. Values are the decimal degress of
     LATitude and LONgitude. The value should be specified to
     six decimal places. -->
  <!ELEMENT GEO (LAT, LON)>

    <!-- Latitude value. -->
    <!ELEMENT LAT (#PCDATA)>

    <!-- Longitude value. -->
    <!ELEMENT LON (#PCDATA)>

  <!-- Title property. -->
  <!ELEMENT TITLE (#PCDATA)>

  <!-- Role property. -->
  <!ELEMENT ROLE (#PCDATA)>

  <!-- Organization logo property. -->
  <!ELEMENT LOGO ((TYPE, BINVAL) | EXTVAL)>
```

```
  <!--  Administrative  agent  property.  -->
  <!ELEMENT  AGENT  (vCard  |  EXTVAL)>

  <!--  Organizational  name  and  units  property.  -->
  <!ELEMENT  ORG  (ORGNAME ,  ORGUNIT*)>

    <!ELEMENT  ORGNAME  (#PCDATA)>

    <!ELEMENT  ORGUNIT  (#PCDATA)>

  <!--  Application  specific  categories  property.  -->
  <!ELEMENT  CATEGORIES  (KEYWORD+)>

    <!ELEMENT  KEYWORD  (#PCDATA)>

  <!--  Commentary  note  property.  -->
  <!ELEMENT  NOTE  (#PCDATA)>

  <!--  Identifier  of  product  that  generated  the  vCard  property.  -->
  <!ELEMENT  PRODID  (#PCDATA)>

  <!--  Last  revised  property.  The  value  must  be  an
        ISO  8601  formatted  UTC  date/time.  -->
  <!ELEMENT  REV  (#PCDATA)>

  <!--  Sort  string  property.  -->
  <!ELEMENT  SORT-STRING  (#PCDATA)>

  <!--  Formatted  name  pronunciation  property.  The  value  is
        either  a  textual  phonetic  pronunciation ,  a  BASE64
        encoded  binary  digital  audio  pronunciation  or  a  URI  to
        an  external  binary  digital  audio  pronunciation. -->
  <!ELEMENT  SOUND  (PHONETIC  |  BINVAL  |  EXTVAL)>

    <!--  Textual  phonetic  pronunciation.  -->
    <!ELEMENT  PHONETIC  (#PCDATA)>

  <!--  Unique  identifier  property.  -->
  <!ELEMENT  UID  (#PCDATA)>

  <!--  Directory  URL  property.  -->
  <!ELEMENT  URL  (#PCDATA)>

  <!--  NOTE:  the  following  element  was  added  by  the  Jabber
              project  (now  XMPP  Standards  Foundation)  to
              handle  free-form  descriptive  text.  -->

  <!ELEMENT  DESC  (#PCDATA)>
```

```
  <!--␣Privacy␣classification␣property.␣-->
  <!ELEMENT␣CLASS␣(PUBLIC␣|␣PRIVATE␣|␣CONFIDENTIAL)>

    <!ELEMENT␣PUBLIC␣EMPTY>

    <!ELEMENT␣PRIVATE␣EMPTY>

    <!ELEMENT␣CONFIDENTIAL␣EMPTY>

  <!--␣Authentication␣credential␣or␣encryption␣␣key␣property.␣-->
  <!ELEMENT␣KEY␣(TYPE?,␣CRED)>

    <!ELEMENT␣CRED␣(#PCDATA)>

<!--␣====␣-->

<!--␣Common␣elements.␣-->

<!--␣Addressing␣type␣indicators.␣-->
<!ELEMENT␣HOME␣EMPTY>
<!ELEMENT␣WORK␣EMPTY>
<!ELEMENT␣POSTAL␣EMPTY>
<!ELEMENT␣PARCEL␣EMPTY>
<!ELEMENT␣DOM␣EMPTY>
<!ELEMENT␣INTL␣EMPTY>
<!ELEMENT␣PREF␣EMPTY>
<!ELEMENT␣VOICE␣EMPTY>
<!ELEMENT␣FAX␣EMPTY>
<!ELEMENT␣PAGER␣EMPTY>
<!ELEMENT␣MSG␣EMPTY>
<!ELEMENT␣CELL␣EMPTY>
<!ELEMENT␣VIDEO␣EMPTY>
<!ELEMENT␣BBS␣EMPTY>
<!ELEMENT␣MODEM␣EMPTY>
<!ELEMENT␣ISDN␣EMPTY>
<!ELEMENT␣PCS␣EMPTY>
<!ELEMENT␣INTERNET␣EMPTY>
<!ELEMENT␣X400␣EMPTY>

<!--␣Format␣type␣parameter.␣-->
<!ELEMENT␣TYPE␣(#PCDATA)>

<!--␣Base64␣encoded␣binary␣value.␣-->
<!ELEMENT␣BINVAL␣(#PCDATA)>

<!--␣URI␣to␣external␣binary␣value␣-->
<!ELEMENT␣EXTVAL␣(#PCDATA)>

<!--␣====␣-->
```