XEP-0068: Field Standardization for Data Forms

Joe Hildebrand
mailto:jhildebr@cisco.com
xmpp:hildjj@jabber.org

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

2012-05-28
Version 1.2

Status	Type	Short Name
Active	Informational	formtypes

This document specifies how to standardize field variables used in the context of jabber:x:data forms.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

XMPP extensions that reuse Data Forms (XEP-0004) [1], such as Multi-User Chat (XEP-0045) [2] and Ad-Hoc Commands (XEP-0050) [3], typically need a way to gather data from both humans (using a GUI format) and computer processes (using a pre-defined but flexible format). The 'jabber:x:data' namespace provides an adequate mechanism for both of these uses, as long as computer processes can rely on the var="" names on a particular type of form. This document defines a mechanism for the XMPP Registrar [4] to standardize the field names in such forms, thus enabling XMPP clients to process forms as they have to this point while giving protocol authors a way to specify a mechanism for non-GUI processors to determine the semantic meanings of forms and their constituent fields.

2 Requirements

1. Forms must continue to be presentable to humans for data entry.
2. XMPP clients that know how to process generic jabber:x:data messages must be supported; the basic format of jabber:x:data must not change.
3. If a form type is used in the context of a standards-track protocol, it should be standardized and registered; however, there is no requirement that all form types must be registered (e.g., form types used in custom applications).
4. Forms that are not directed *to* an entity must be able to traverse the entity (e.g., a form sent to a MUC room, intended for the participants of the room, and not the room itself).
5. Forms must continue to be flexible for implementations; unknown future expansion fields must not be limited.
6. The chosen approach must work for forms embedded in <message/> stanzas as well as in <iq/> stanzas.

3 Approach

3.1 Overview

Within XMPP, namespaces are used to scope data that conforms to a schema (often data that extends the core protocol in some fashion). In addition, namespaces can also provide context for the field variable names used in jabber:x:data forms and reports. This proposal makes

[1] XEP-0004: Data Forms <https://xmpp.org/extensions/xep-0004.html>.
[2] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[3] XEP-0050: Ad-Hoc Commands <https://xmpp.org/extensions/xep-0050.html>.
[4] 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/>.

that association explicit by defining a mechanism for linking a namespace name with a form along with the field names and types used in that form. Specifically, the namespace name is specified in the form as the value of a hidden variable called "FORM_TYPE".

3.2 Whether to Register

The first decision-point is whether a FORM_TYPE needs to be registered with the XMPP Registrar. The following rules apply:

1. If the FORM_TYPE is used in the context of a form defined in a XEP published by the XMPP Standards Foundation (XSF) [5], it MUST be registered.
2. If the FORM_TYPE is used in the context of some other XMPP protocol but the form is not defined in a XEP, it MAY be registered.
3. If the FORM_TYPE is used in the context of a custom protocol, it MAY be registered.

3.3 FORM_TYPE Names

While the value of the FORM_TYPE attribute SHOULD be considered an opaque string from the application perspective, the following rules apply:

1. For custom protocols, the name SHOULD be an HTTP URI that is managed by the namespace owner (e.g., "http://example.com/foo").
2. For all new protocols approved by the XSF, the name MUST be a "urn:xmpp:*" URN in accordance with RFC 4854 [6] and Section 4 of XMPP Registrar Function (XEP-0053) [7].
3. For "legacy" protocols managed by the XSF, the name SHOULD use the old-style "jabber:*:*" or "http://jabber.org/protocol/*" format.

3.4 Field Names

For FORM_TYPEs that are registered with the XMPP Registrar, the following rules apply:

1. If the field is defined by the XSF (i.e., in a XEP), the field name SHALL be determined in accordance with the usual XSF consensus process and the field MUST be registered with the XMPP Registrar.

[5] The XMPP Standards Foundation (XSF) is an independent, non-profit membership organization that develops open extensions to the IETF's Extensible Messaging and Presence Protocol (XMPP). For further information, see <https://xmpp.org/about/xmpp-standards-foundation>.

[6] RFC 4854: A Uniform Resource Name (URN) Namespace for Extensions to the Extensible Messaging and Presence Protocol (XMPP) <http://tools.ietf.org/html/rfc4854>.

[7] XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.

2. If the field is defined outside the XSF, the field name SHALL follow the extension rules described below and the field MAY be registered with the XMPP Registrar.

For FORM_TYPEs that are not registered with the XMPP Registrar, the field name SHALL follow the extension rules described below and the field typically will not be registered with the XMPP Registrar.
The ”namespace” of a field is assumed to be inherited from the FORM_TYPE. When an organization or project defines a field that is used in the context of a FORM_TYPE it does not manage (e.g., a non-XSF field contained in a form whose FORM_TYPE is managed by the XSF, or a third-party field contained in a form whose FORM_TYPE is managed by some other organization), the name of the field MUST be namespaced using Clark Notation [8], where the universal name portion SHOULD be a URI controlled by the extending organization or project, e.g., a field name of ”
pathhttp://example.com/pubsub}time_restrictions”.
For reasons that are lost in the mists of time, some XMPP extension protocols produced by the XSF, such as Multi-User Chat (XEP-0045) [9] and Publish-Subscribe (XEP-0060) [10], prefix their field names with strings like ”muc#” and ”pubsub#”. There is no good reason to apply that convention to new XSF extensions. Indeed, there is even no good reason to apply that convention to the names of new fields defined by the XSF for those existing XSF extensions; however, the practice is harmless for those existing extensions (since a string such as ”
pathhttp://jabber.org/protocol/pubsub#subscribe_authorization}pubsub#subscriber_jid”
can be considered equivalent to a string such as ”pubsub#subscriber_jid”), and this document does not actively recommend deprecating the convention.
Note: Older versions of this specification mandated that unregistered field names had to begin with the prefix ”x-”. In accordance with RFC 6648 [11], that mandate has been removed. However, existing ”x-” field names are acceptable and can be registered with the XMPP Registrar as described above.

3.5 Field Values

Field values MAY also be registered; refer to the XMPP Registrar section of this document.

3.6 Uniqueness and Comparison

FORM_TYPE names, field names, and field values MUST be compared as strings. The use of URIs in FORM_TYPE names and field names is simply a recommended method for insuring uniqueness, and other such methods are acceptable (e.g., Java-like reverse domain names

[8] Clark Notation, a syntax to allow universal names written as a URI in curly brackets followed by the local name; developed by James Clark. <http://www.jclark.com/xml/xmlns.htm>.
[9] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[10] XEP-0060: Publish-Subscribe <https://xmpp.org/extensions/xep-0060.html>.
[11] RFC 6648: Deprecating the X- Prefix and Similar Constructs in Application Protocols <http://tools.ietf.org/html/rfc6648>.

such as ”com.example.foo”).

4 Use Cases

4.1 Unspecified Form

These are forms that do not have a hidden field of name FORM_TYPE. Existing processing rules still apply.

Listing 1: Message with no FORM_TYPE

```
<message
    from='juliet@capulet.com/balcony'
    to='romeo@montague.net/garden'>
  <thread>vote-thread-reatmon-134</thread>
  <x xmlns='jabber:x:data' type='form'>
    <title>Vote #134</title>
    <instructions>
      This is the vote to pick a new mascot.
      Thanks for your time!
    </instructions>
    <field var='mascot' type='list-single'>
      <required/>
      <option label='Light␣Bulb'><value>light_bulb</value></option>
      <option label='Penguin'><value>penguin</value></option>
      <option label='Moose'><value>moose</value></option>
      <option label='Triangle␣Man'><value>triangle_man</value></option
          >
      <option label='Other'><value>other</value></option>
    </field>
  </x>
</message>
```

4.2 Correctly Specified FORM_TYPE

In the following example, the FORM_TYPE is 'http://jabber.org/protocol/pubsub', all of the fields whose names start with ”pubsub#” are registered with the XMPP Registrar (see Publish-Subscribe (XEP-0060) [12]), and the custom ”time_restrictions” field defined by the organization at example.com uses Clark Notation in the field name.

Listing 2: Message with FORM_TYPE

```
<message to=”node-owner” from=”pubsub.jabber.org”>
    <x xmlns=”jabber:x:data” type=”form”>
```

[12] XEP-0060: Publish-Subscribe <https://xmpp.org/extensions/xep-0060.html>.

```
        <title>PubSub subscriber request</title>
        <instructions>To approve this entity's␣subscription␣request,
␣␣␣␣␣␣␣␣␣␣␣␣click␣the␣OK␣button.␣To␣deny␣the␣request,␣click␣the
␣␣␣␣␣␣␣␣␣␣␣␣cancel␣button.</instructions>
␣␣␣␣␣␣␣<field␣var="FORM_TYPE"␣type="hidden">
␣␣␣␣␣␣␣␣␣␣␣<value>http://jabber.org/protocol/pubsub#
    subscribe_authorization</value>
␣␣␣␣␣␣␣</field>
␣␣␣␣␣␣␣<field␣var="pubsub#node"␣type="hidden">
␣␣␣␣␣␣␣␣␣␣␣<value>generic/pgm-mp3-player</value>
␣␣␣␣␣␣␣</field>
␣␣␣␣␣␣␣<field␣var="pubsub#subscriber_jid"␣type="jid-single"
␣␣␣␣␣␣␣␣␣␣␣␣␣␣label="Jabber␣ID␣of␣Subscriber">
␣␣␣␣␣␣␣␣␣␣␣<value>sub1@foo.com</value>
␣␣␣␣␣␣␣</field>
␣␣␣␣␣␣␣<field␣var="\path{{http://example.com/pubsub}time_restrictions
    }"␣type="text-multi"
␣␣␣␣␣␣␣␣␣␣␣␣␣␣label="Limit␣to␣these␣time␣ranges">
␣␣␣␣␣␣␣␣␣␣␣<value>09:00-12:00</value>
␣␣␣␣␣␣␣␣␣␣␣<value>13:00-17:00</value>
␣␣␣␣␣␣␣</field>
␣␣␣␣</x>
</message>
```

4.3 Incorrectly Specified FORM_TYPE

If the FORM_TYPE field is not hidden, it MUST be ignored as a context indicator.

Listing 3: Message with bad FORM_TYPE

```
<message to="juliet@capulet.com" from="romeo@montague.net/garden">
    <x xmlns="jabber:x:data" type="form">
       <title>Balcony Scene (Act 2, Scene 2)</title>
       <instructions>But soft! What light through yonder window breaks
          ?</instructions>
       <!-{}- Not hidden.  Treated as any other text-single field.
          -{}->
       <field var="FORM_TYPE" type="text-single">
         <value>http://jabber.org/protocol/shakespeare</value>
       </field>
       <field var="light" type="list-multi">
           <option label="Juliet">Sun</option>
           <option lable="Maid">Moon</option>
           <option label="Eyes">Stars</option>
       </field>
    </x>
</message>
```

4.4 IQ Example

The following example shows a user's interaction with a Multi-User Chat room in order to register with the room.

Listing 4: User Requests Registration Requirements

```
<iq
    from='hag66@shakespeare.lit/pda'
    to='darkcave@macbeth.shakespeare.lit'
    type='get'
    id='reg1'>
  <query xmlns='jabber:iq:register'/>
</iq>
```

Listing 5: Service Returns Registration Form

```
<iq
    type='result'
    from='darkcave@macbeth.shakespeare.lit'
    to='hag66@shakespeare.lit/pda'
    id='reg1'>
  <query xmlns='jabber:iq:register'>
    <instructions>
      To register on the web, visit http://shakespeare.lit/
    </instructions>
    <x xmlns='jabber:x:data' type='form'>
      <title>Dark Cave Registration</title>
      <instructions>
        Please provide the following information
        to register with this room.
      </instructions>
      <field
          type='hidden'
          var='FORM_TYPE'>
        <value>http://jabber.org/protocol/muc#user</value>
      </field>
      <field
          type='text-single'
          label='First Name'
          var='muc#user_first'>
        <required/>
      </field>
      <field
          type='text-single'
          label='Last Name'
          var='muc#user_last'>
        <required/>
      </field>
```

```
      <field
          type='text-single'
          label='Desired␣Nickname'
          var='muc#user_roomnick'>
        <required/>
      </field>
      <field
          type='text-single'
          label='Your␣URL'
          var='muc#user_url'/>
      <field
          type='text-single'
          label='Email␣Address'
          var='muc#user_email'/>
      <field
          type='text-multi'
          label='FAQ␣Entry'
          var='muc#user_faqentry'/>
    </x>
  </query>
</iq>
```

Listing 6: User Submits Registration Form

```
<iq
    type='set'
    from='hag66@shakespeare.lit/pda'
    to='darkcave@macbeth.shakespeare.lit'
    id='reg2'>
  <query xmlns='jabber:iq:register'>
    <x xmlns='jabber:x:data' type='submit'>
      <field var='FORM_TYPE'>
        <value>http://jabber.org/protocol/muc#user</value>
      </field>
      <field var='muc#user_first'>
        <value>Brunhilde</value>
      </field>
      <field var='muc#user_last'>
        <value>Entwhistle-Throckmorton</value>
      </field>
      <field var='muc#user_roomnick'>
        <value>thirdwitch</value>
      </field>
      <field var='muc#user_url'>
        <value>http://witchesonline/~hag66/</value>
      </field>
      <field var='muc#user_email'>
        <value>hag66@witchesonline</value>
      </field>
      <field var='muc#user_faqentry'>
```

```
        <value>Just another witch.</value>
      </field>
    </x>
  </query>
</iq>
```

5 Implementation Notes

If the FORM_TYPE field is not type="hidden", it does not have the special meaning defined herein.
If the form is used in an IQ, the namespace of the <query/> element SHOULD match the base namespace of the FORM_TYPE. (One possible way of solving this problem would have been to reuse the <query/> tag from the IQ form of jabber:x:data within messages, but that would have meant that existing clients would not have been able to participate in these exchanges.)

6 Security Considerations

Security-conscious programs that are using this approach should be careful to process only agreed-upon fields with agreed-upon types.

7 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [13].

8 XMPP Registrar Considerations

8.1 Registries

8.1.1 FORM_TYPEs Registry

The XMPP Registrar shall maintain a registry of information about submitted FORM_TYPEs. In order to submit new values to this registry, the registrant shall define an XML fragment of the following form and either include it in the relevant XMPP Extension Protocol or send it to the email address <registrar@xmpp.org>:

[13] 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/>.

```
<form_type>
  <name>FORM_TYPE namespace or namespace derivative</name>
  <doc>associated specification</doc>
  <desc>natural-language description of form type</desc>
  <field
      var='the_field_name'
      type='the_field_type'
      label='natural-language description of field'/>
</form_type>
```

The registrant MAY register more than one FORM_TYPE at a time, each contained in a separate <form_type/> element. The registrant MAY also register more than one field at a time, each contained in a separate <field/> child element. Registrations of new fields within an existing FORM_TYPE MUST include the full XML snippet but SHOULD NOT include the FORM_TYPE description (only the name and the XEP number or other document identifier). Note that for ease of use the format for the <field/> element in the registry submission is the same as that defined in XEP-0004; in addition, the value of the 'type' attribute MUST be one of those defined in XEP-0004.
In addition, a registrant MAY also register particular field option values for fields of type 'list-single' and 'list-multi'. The format for such submissions is as follows:

```
<form_type>
  <name>FORM_TYPE namespace or namespace derivative</name>
  <doc>associated XEP or other document</doc>
  <desc>natural-language description of form type</desc>
  <field
      var='the_field_name'
      type='the_field_type'
      label='natural-language description of field'>
    <option label='natural-language description of option'>
      <value>the_value</value>
    </option>
  </field>
</form_type>
```