XEP-0122: Data Forms Validation

Matthew Miller
mailto:linuxwolf@outer-planes.net
xmpp:linuxwolf@outer-planes.net

2018-03-21
Version 1.0.2

Status	Type	Short Name
Draft	Standards Track	xdata-validate

This specification defines a backwards-compatible extension to the XMPP Data Forms protocol that enables applications to specify additional validation guidelines related to a form, such as validation of standard XML datatypes, application-specific datatypes, value ranges, and regular expressions.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

Data Forms (XEP-0004) [1] ("x:data") provides a simple and interoperable way to request and present information for both applications and humans. However, the simple nature of "x:data" requires the form interpreter at times to guess as to exactly what type of information is being requested or provided. This document builds upon "x:data" to provide this additional validation.

2 Requirements

The requirements for this document are:

- Backwards compatible with the existing "x:data" protocol.
- Provide extended datatypes (such as dates, times, and numbers).
- Allow for multiple validation methods.
- Allow for user-defined datatypes.
- Accommodate value ranges.
- Accommodate regular expression matching.

3 Use Cases

This document defines a new namespace, "http://jabber.org/protocol/xdata-validate". The root element for this namespace is <validate/>, and MUST be contained within a <field/> element (qualified by the 'jabber:x:data' namespace) for each Data Forms field that possesses additional validation information.

3.1 Datatype Validation

The simplest usage is to provide a more-granular datatype for a <field/> element used in Data Forms. To provide this datatype information, a <validate/> element is included whose 'datatype' attribute specifies the data type of any <value/> contained within the <field/> element:

[1] XEP-0004: Data Forms <https://xmpp.org/extensions/xep-0004.html>.

Listing 1: Field with extended datatype

```
<field var='evt.date' type='text-single' label='Event Date/Time'>
  <validate xmlns='http://jabber.org/protocol/xdata-validate'
            datatype='xs:dateTime'/>
  <value>2003-10-06T11:22:00-07:00</value>
</field>
```

The preceding example demonstrates a field that is expected to contain a date/time value. The 'datatype' attribute specifies the datatype. This attribute is OPTIONAL, and defaults to "xs:string". It MUST meet one of the following conditions:

- Start with "xs:", and be one of the "built-in" datatypes defined in XML Schema Part 2 [2]
- Start with a prefix registered with the XMPP Registrar [3]
- Start with "x:", and specify a user-defined datatype.

Note that while "x:" allows for ad-hoc definitions, its use is NOT RECOMMENDED.

3.2 Validation Methods

In addition to datatypes, the validation method can also be provided. The method is specified via a child element. The validation methods defined in this document are:

- <basic/> for validation only against the datatype itself
- <open/> for open-ended validation against the datatype
- <range/> for validation against a given min/max and the datatype
- <regex/> for validation against a given regular expression and the datatype

If no validation method is specified, form processors MUST assume <basic/> validation. The <validate/> element SHOULD include one of the above validation method elements, and MUST NOT include more than one.
Any validation method applied to a field of type "list-multi", "list-single", or "text-multi" (other than <basic/>) MUST imply the same behavior as <open/>, with the additional constraints defined by that method.

[2] XML Schema Part 2: Datatypes <http://www.w3.org/TR/xmlschema11-2/>.
[3] 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/>.

3.2.1 <basic/> Validation

Building upon the earlier example, to indicate that the value(s) should simply match the field type and datatype constraints, the <validate/> element shall contain a <basic/> child element.

Listing 2: Basic validation

```
<field var='evt.date' type='text-single' label='Event␣Date/Time'>
  <validate xmlns='http://jabber.org/protocol/xdata-validate'
            datatype='xs:dateTime'>
    <basic/>
  </validate>
  <value>2003-10-06T11:22:00-07:00</value>
</field>
```

Using <basic/> validation, the form interpreter MUST follow the validation rules of the datatype (if understood) and the field type.

3.2.2 <open/> Validation

For "list-single" or "list-multi", to indicate that the user may enter a custom value (matching the datatype constraints) or choose from the predefined values, the <validate/> element shall contain an <open/> child element:

Listing 3: Open validation

```
<field var='evt.category'
       type='list-single'
       label='Event␣Category'>
  <validate xmlns='http://jabber.org/protocol/xdata-validate'
            datatype='xs:string'>
    <open/>
  </validate>
  <option><value>holiday</value></option>
  <option><value>reminder</value></option>
  <option><value>appointment</value></option>
</field>
```

The <open/> validation method applies to "text-multi" differently; it hints that each value for a "text-multi" field shall be validated separately. This effectively turns "text-multi" fields into an open-ended "list-multi", with no options and all values automatically selected.

3.2.3 <range/> Validation

To indicate that the value should fall within a certain range, the <validate/> element shall contain a <range/> child element:

Listing 4: Range validation

```
<field var='evt.date' type='text-single' label='Event Date/Time'>
  <validate xmlns='http://jabber.org/protocol/xdata-validate'
            datatype='xs:dateTime'>
    <range min='2003-10-05T00:00:00-07:00'
           max='2003-10-24T23:59:59-07:00'/>
  </validate>
  <value>2003-10-06T11:22:00-07:00</value>
</field>
```

The 'min' and 'max' attributes of the <range/> element specify the minimum and maximum values allowed, respectively.
The 'max' attribute specifies the maximum allowable value. This attribute is OPTIONAL. The value depends on the datatype in use.
The 'min' attribute specifies the minimum allowable value. This attribute is OPTIONAL. The value depends on the datatype in use.
The <range/> element SHOULD possess either a 'min' or 'max' attribute, and MAY possess both. If neither attribute is included, the processor MUST assume that there are no range constraints.

3.2.4 <regex/> Validation

To indicate that the value should be restricted to a regular expression, the <validate/> element shall contain a <regex/> child element:

Listing 5: Regular expression validation

```
<field var='ssn' type='text-single' label='Social Security Number'>
  <desc>This field should be your SSN, including '-' (e.g.
     123-12-1234)</desc>
  <validate xmlns='http://jabber.org/protocol/xdata-validate'
            datatype='xs:string'>
    <regex>([0-9]{3})-([0-9]{2})-([0-9]{4})</regex>
  </validate>
</field>
```

The XML character data of this element is the pattern to apply. The syntax of this content MUST be that defined for POSIX extended regular expressions[4], including support for Unicode[5].
The <regex/> element MUST contain character data only.

[4] The "best" definition of this syntax can be found in the re_format(7) man page
[5] Guidelines for adapting regular expressions to support Unicode is defined at http://www.unicode.org/reports/tr18/

3.3 Selection Ranges in ”list-multi”

For ”list-multi”, validation can indicate (via the <list-range/> element) that a minimum and maximum number of options should be selected and/or entered. This selection range MAY be combined with the other methods to provide more flexibility.

Listing 6: Selection Range validation

```
<field var='evt.notify-methods'
       type='list-multi'
       label='Notify␣me␣by'>
  <validate xmlns='http://jabber.org/protocol/xdata-validate'
            datatype='xs:string'>
    <basic/>
    <list-range min='1' max='3'/>
  </validate>
  <option><value>e-mail</value></option>
  <option><value>jabber/xmpp</value></option>
  <option><value>work phone</value></option>
  <option><value>home phone</value></option>
  <option><value>cell phone</value></option>
</field>
```

The <list-range/> element SHOULD be included only when the <field/> is of type ”list-multi” and SHOULD be ignored otherwise.
The ’max’ attribute specifies the maximum allowable number of selected/entered values. This attribute is OPTIONAL. The value MUST be a positive integer.
The ’min’ attribute specifies the minimum allowable number of selected/entered values. This attribute is OPTIONAL. The value MUST be a positive integer.
The <list-range/> element SHOULD possess either a ’min’ or ’max’ attribute, and MAY possess both. If neither attribute is included, the processor MUST assume that there are no selection constraints.

4 Implementation Notes

4.1 Required to Support

At a minimum, implementations MUST support the following:

- Datatype validation
- The <basic/> validation method

If an implementation does not understand the specified datatype, it MUST validate according to the default ”xs:string” datatype. If an implementation does not understand the specified

method, it MUST validate according to the <basic/> method.

4.2 Namespacing

While all elements associated with this document MUST be qualified by the 'http://jabber.org/protocol/xdata-validate' namespace, explicitly declaring the default namespace for each instance can be overly verbose. However, Jabber/XMPP implementations have historically been very lax regarding namespacing, thus requiring some careful use of prefixes.
The use of namespace prefixes is RECOMMENDED for large forms, to reduce the data size. To maintain the highest level of compatibility, implementations sending the form using prefixes SHOULD use the namespace prefix "xdv", and SHOULD declare the namespace prefix mapping in the ancestor <x xmlns='jabber:x:data'/> element:

Listing 7: Example of recommended namespace prefixing

```
<x xmlns='jabber:x:data'
   xmlns:xdv='http://jabber.org/protocol/xdata-validate'
   type='form'>
  <title>Sample Form</title>
  <instructions>
    Please provide information for the following fields...
  </instructions>
  <field type='text-single' var='name' label='Event␣Name'/>
  <field type='text-single' var='date/start' label='Starting␣Date'>
    <xdv:validate datatype='xs:date'>
      <basic/>
    </xdv:validate>
  </field>
  <field type='text-single' var='date/end' label='Ending␣Date'>
    <xdv:validate datatype='xs:date'>
      <basic/>
    </xdv:validate>
  </field>
</x>
```

4.3 Internationalization/Localization

This document relies on the internationalization/localization mechanisms provided by XMPP Core [6]. As much as possible, all datatype formats MUST be locale-independent.

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

4.4 Form Submissions

Form processors MUST NOT assume that a form with validation has actually been validated when submitted. There is no realistic expectation that form interpreters honor validation.

4.5 Existing Protocols

While this document is compatible with the existing ”x:data” definition, form providers SHOULD first determine support for it, using either Entity Capabilities (XEP-0115) [7] if presence-aware or Service Discovery (XEP-0030) [8]. This is especially important for limited-connection and/or limited-capabilities devices, such as cell phones.

4.6 Display Considerations

Although primarily intended for validating form submission, validation MAY have an impact on display, and MAY be applied to data forms that are not submitted (e.g. ’result’ type forms). The following table outlines which field types a particular validation method is or is not appropriate for, and how a display SHOULD interpret the validation methods if considered[9]:

Validation Method	SHOULD be Allowed	SHOULD NOT be Allowed	Display Suggestions
basic	fixed list-multi list-single text-multi text-single	hidden jid-multi jid-single	Display the datatype appropriate to the locale
open	jid-multi jid-single list-multi list-single text-multi text-single	hidden	Display the datatype appropriate to the locale. For ”text-multi” treat each value as a discrete entry (e.g. a user-entered list). For ”list-multi” or ”list-single”, allow user to add/remove entries to select.

[7] XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.
[8] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[9] If a particular field type is not listed, the display MAY include validation support, but is not expected to do so.

Validation Method	SHOULD be Allowed	SHOULD NOT be Allowed	Display Suggestions
range	text-single	hidden jid-multi list-multi text-multi	Display the datatype appropriate to the locale. For ”text-single”, allow user to increment/decrement through possible values. For ”text-multi” treat each value as a discrete entry (e.g. a user-entered list). For ”list-multi” or ”list-single”, allow user to add/remove entries to select.
regex	text-single	hidden jid-multi list-multi text-multi	Display the datatype appropriate to the locale. If possible, display a valid example. For ”text-multi” treat each value as a discrete entry (e.g. a user-entered list). For ”list-multi” or ”list-single”, allow user to add/remove entries to select.

4.7 Validation Ranges

The <range/> validation method MUST be used only with datatypes that have finite quantities. Within the standard datatype set, it MUST NOT be used with ”xs:string”.

5 Security Considerations

This document introduces no security concerns above and beyond those specified in XEP-0004: Data Forms.

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [10].

7 XMPP Registrar Considerations

7.1 Protocol Namespaces

The XMPP Registrar includes 'http://jabber.org/protocol/xdata-validate' in its registry of protocol namespaces.

7.2 Registries

7.2.1 Datatype Prefixes Registry

The XMPP Registrar maintains a registry of datatype prefixes used in the context of Data Forms Validation (see <https://xmpp.org/registrar/xdv-prefixes.html>), where each prefix denotes a group of related datatypes.
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>:

```
<datatype-prefix>
  <prefix>the prefix token (e.g., "xs")</prefix>
  <desc>a natural-language description of the datatype family</desc>
  <doc>the document in which datatype family is specified</doc>
</datatype-prefix>
```

The registrant may register more than one prefix at a time, each contained in a separate <datatype-prefix/> element.
As part of this document, the following datatype prefixes shall be registered:

```
<datatype-prefix>
  <prefix>x</prefix>
  <desc>An ad-hoc datatype</desc>
  <doc>XEP-0122</doc>
</datatype-prefix>
<datatype-prefix>
  <prefix>xs</prefix>
```

[10] 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/>.

```
  <desc>A "standard" datatype as defined in XML Schema Part 2</desc>
  <doc>XML Schema Part 2</doc>
</datatype-prefix>
```

7.2.2 Datatypes Registry

The XMPP Registrar maintains a registry of datatypes used in the context of Data Forms Validation (see <https://xmpp.org/registrar/xdv-datatypes.html>), where each datatype name includes the relevant prefix (e.g., "xs:anyURI").
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>:

```
<datatype>
  <name>the full datatype name (e.g., "xs:string")</name>
  <desc>a natural-language description of the datatype</desc>
  <methods>the validation methods that may apply to the datatype</
     methods>
  <min>the minimum value for the datatype (if any)</min>
  <max>the maximum value for the datatype (if any)</max>
</datatype>
```

The registrant may register more than one datatype at a time, each contained in a separate <datatype/> element.
The following submission contains the built-in datatypes defined in XML Schema Part 2 that are deemed mostly like to be useful in the context of the Data Forms protocol; additional datatypes defined therein, as well as other datatypes not defined in XML Schema Part 2, may be registered via separate submissions in the future.

```
<datatype>
  <name>xs:anyURI</name>
  <desc>a Uniform Resource Identifier Reference (URI)</desc>
  <methods>basic regex</methods>
  <min>N/A</min>
  <max>N/A</max>
</datatype>
<datatype>
  <name>xs:byte</name>
  <desc>an integer with the specified min/max</desc>
  <methods>basic range</methods>
  <min>-128</min>
  <max>127</max>
</datatype>
<datatype>
  <name>xs:date</name>
  <desc>a calendar date</desc>
```

```
  <methods>basic range regex</methods>
  <min>N/A</min>
  <max>N/A</max>
</datatype>
<datatype>
  <name>xs:dateTime</name>
  <desc>a specific instant of time</desc>
  <methods>basic range regex</methods>
  <min>N/A</min>
  <max>N/A</max>
</datatype>
<datatype>
  <name>xs:decimal</name>
  <desc>an arbitrary-precision decimal number</desc>
  <methods>basic range</methods>
  <min>none</min>
  <max>none</max>
</datatype>
<datatype>
  <name>xs:double</name>
  <desc>an IEEE double-precision 64-bit floating point type</desc>
  <methods>basic range</methods>
  <min>none</min>
  <max>none</max>
</datatype>
<datatype>
  <name>xs:int</name>
  <desc>an integer with the specified min/max</desc>
  <methods>basic range</methods>
  <min>-2147483648</min>
  <max>2147483647</max>
</datatype>
<datatype>
  <name>xs:integer</name>
  <desc>a decimal number with no fraction digits</desc>
  <methods>basic range</methods>
  <min>none</min>
  <max>none</max>
</datatype>
<datatype>
  <name>xs:language</name>
  <desc>a language identifier as defined by RFC 1766</desc>
  <methods>basic regex</methods>
  <min>N/A</min>
  <max>N/A</max>
</datatype>
<datatype>
  <name>xs:long</name>
  <desc>an integer with the specified min/max</desc>
```

```
  <methods>basic range</methods>
  <min>-9223372036854775808</min>
  <max>9223372036854775807</max>
</datatype>
<datatype>
  <name>xs:short</name>
  <desc>an integer with the specified min/max</desc>
  <methods>basic range</methods>
  <min>-32768</min>
  <max>32767</max>
</datatype>
<datatype>
  <name>xs:string</name>
  <desc>a character strings in XML</desc>
  <methods>basic regex</methods>
  <min>N/A</min>
  <max>N/A</max>
</datatype>
<datatype>
  <name>xs:time</name>
  <desc>an instant of time that recurs every day</desc>
  <methods>basic range regex</methods>
  <min>N/A</min>
  <max>N/A</max>
</datatype>
```

8 XML Schema

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

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='http://jabber.org/protocol/xdata-validate'
    xmlns='http://jabber.org/protocol/xdata-validate'
    elementFormDefault='qualified'>

  <xs:element name='validate'>
    <xs:complexType>
      <xs:sequence>
        <xs:choice minOccurs='0' maxOccurs='1'>
          <xs:element ref='basic'/>
          <xs:element ref='open'/>
          <xs:element ref='range'/>
          <xs:element ref='regex'/>
        </xs:choice>
        <xs:element ref='list-range' minOccurs='0' maxOccurs='1'/>
      </xs:sequence>
```

```
      <xs:attribute name='datatype'
                    type='xs:string'
                    use='optional'
                    default='xs:string'/>
    </xs:complexType>
  </xs:element>

  <xs:element name='basic' type='empty'/>

  <xs:element name='open' type='empty'/>

  <xs:element name='range'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='min' use='optional'/>
          <xs:attribute name='max' use='optional'/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name='regex' type='xs:string'/>

  <xs:element name='list-range'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='min'
                        type='xs:unsignedInt'
                        use='optional'/>
          <xs:attribute name='max'
                        type='xs:unsignedInt'
                        use='optional'/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value=''/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```