



XMPP

XEP-0171: Language Translation

Boyd Fletcher

<mailto:boyd.fletcher@us.army.mil>

Keith Lirette

<mailto:keith.lirette@tridsys.com>

Daniel LaPrade

<mailto:dlaprade@echostorm.net>

Brian Raymond

<mailto:braymond@echostorm.net>

2015-10-15

Version 1.1.1

| Status | Type | Short Name |
|--------|-----------------|------------|
| Draft | Standards Track | langtrans |

This specification defines an XMPP protocol extension for providing language translation facilities over XMPP. It supports human, machine, client-based, and server-based translations.

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 | 1 |
| 2 | Glossary | 1 |
| 3 | Requirements | 2 |
| 4 | Use Cases | 2 |
| 4.1 | Message Delivery | 3 |
| 4.1.1 | Direct Translation | 3 |
| 4.1.2 | Translation With Pivot | 3 |
| 4.1.3 | Translation With Pivot Specifying Details | 4 |
| 4.2 | Discovering Translation Providers | 4 |
| 4.2.1 | Discovering Translation Providers On a Server | 4 |
| 4.2.2 | Discovering Identity of Providers | 5 |
| 4.2.3 | Discovering Language Support | 5 |
| 4.3 | Requesting a Translation from a Service | 7 |
| 4.3.1 | Requesting a Basic Translation | 7 |
| 4.3.2 | Requesting a Translation With Multiple Destination Languages | 7 |
| 4.3.3 | Requesting a Translation With a Specific Dictionary | 8 |
| 5 | Implementation Notes | 9 |
| 6 | Internationalization Considerations | 10 |
| 7 | Security Considerations | 10 |
| 8 | IANA Considerations | 10 |
| 9 | XMPP Registrar Considerations | 10 |
| 9.1 | Protocol Namespaces | 10 |
| 9.2 | Service Discovery Identities | 11 |
| 10 | XML Schema | 11 |
| 10.1 | langtrans | 11 |
| 10.2 | langtrans:items | 12 |

1 Introduction

There currently exists no standard for describing language translations over a text chat protocol. While numerous products and services exist to provide translation of text, there exists no standardized protocol extension for requesting a translation and expressing the details of the translation over XMPP (see [XMPP Core ¹](#)). This document describes how to express a translation and its components in an XMPP message as well as a method to request translation.

Direct translation can be realized by either client-side translation before sending or transparent components translating messages on the fly. Discovering XMPP entities capable of translation allows for clients to request translation from them based on their capabilities. The remote XMPP entity could be either an automated translation service or a human providing translation.

2 Glossary

Original Text This is the message text that was originally created by the sender. This is the text that is translated.

Translated Text This is the message text that has been translated by the language translation engines. This also called the destination text. For any given message there can be multiple destination text message bodies.

Pivot Language Pivoting is the process of using one or more intermediate languages to translate from a given source language to a specific destination language. For example, if you needed to translate from English to Russian but only had translators that went from English to French and French to Russian then you could use French as a pivot language.

Pivot Text This is the translated text of the original message in a pivot language. For any given destination language, there can be zero or more pivot text bodies. The ordering of pivoting is required to be specified for the destination language.

Language Translation Engine Since not all language translation engines are the same quality it is important to some classes of users that they know what translation engine was used. It is equally important to also be able to select a specific translation engine for a given language pairing if more than one engine is available.

Language Translation Character Set Some language translation engines can only translate text between languages if certain character sets (or code pages) are used.

Language Translation Dictionary In order to enhance the accuracy of translation engines most support the concept of mission specific dictionaries.

¹RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

3 Requirements

The protocol defined herein addresses the following requirements:

1. Enable an XMPP entity to request a translation from a remote XMPP entity.
2. Enable an XMPP entity to express the following mandatory elements of a translation for any receiving entities.
 - a) Identification of Original Text
 - b) Identification of Translated Text
 - c) Identification of any Pivot Language(s) and Text
 - d) Identification of the method and order of translation
3. Enable an XMPP entity to express the following optional elements of a translation for any receiving entities.
 - a) Identification of Language Translation Engines used.
 - b) Identification of Location of the translation.
 - c) Identification of Sender and Destination language of choice.
 - d) Discovery of Server support for translation including which language pair, dictionaries, and engines are available.
 - e) Identification of desire to text to be translated or not to be translated.

The following methods of translation are supported:

1. Manual or Human
2. Machine or Automated
3. Machine with Human Review

4 Use Cases

The following use cases illustrate simple scenarios for translation of expressions as well as requesting a translation from remote entities.

4.1 Message Delivery

A message directly translated by the originating XMPP entity or a transparent XMPP entity delivered to a remote entity with only the required elements of source and destination language; this is the simplest case for a translation from one language to another. The source language is known because there is no <translation/> tag describing it. Three translation methods are supported by doing the following:

1. If no 'engine' attribute is present, then manual (or human) translation was performed.
2. If an 'engine' attribute is present then machine (or automated) translation was performed, where the translation engine is identified by the value of the 'engine' attribute. If the 'engine' attribute is present its value is an empty string, then the name of the translation engine was not available.
3. If the 'engine' attribute and the 'reviewed' attribute are present, then machine translation was performed but the message text was reviewed and possibly modified by a human.

4.1.1 Direct Translation

Listing 1: Entity sends a message translated from English to French

```
<message xml:lang='en' from='bard@shakespeare.lit/globe' to='
  playwright@marlowe.lit/theatre'>
  <subject xml:lang='en'>Hello</subject>
  <subject xml:lang='fr'>Bonjour</subject>
  <body xml:lang='en'>How are you?</body>
  <body xml:lang='fr'>comment allez-vous?</body>
  <x xmlns='urn:xmpp:langtrans'>
    <translation destination_lang='fr' source_lang='en' />
  </x>
</message>
```

4.1.2 Translation With Pivot

A message translated by the originating XMPP entity or a transparent XMPP entity delivered to a remote entity with the pivot languages used to accomplish the translation. The source language is known because there is no <x/> translation tag describing it. When a translation is done via a pivot language, the pivot languages and their order of use MUST be specified.

Listing 2: Entity sends a message translated from French to Russian via English using human translators

```
<message xml:lang='fr' from='bard@shakespeare.lit/globe' to='
  playwright@marlowe.lit/theatre'>
```

```

<subject xml:lang='fr'>Bonjour</subject>
<subject xml:lang='en'>Hello</subject>
<subject xml:lang='ru'>x443;&#x43B;&#x442;&#x435;</subject>
<body xml:lang='fr'>comment allez-vous?</body>
<body xml:lang='en'>How are you?</body>
<body xml:lang='ru'>&#x41A;&#x430;&#x43A; &#x432;&#x44B;?</body>
<x xmlns='urn:xmpp:langtrans'>
  <translation destination_lang='en' source_lang='fr' />
  <translation destination_lang='ru' source_lang='en' />
</x>
</message>

```

4.1.3 Translation With Pivot Specifying Details

A message translated by the originating XMPP entity or a transparent XMPP entity delivered to a remote entity using pivot languages and machine translation. The source language is known because there is no <x/> translation tag describing it.

Listing 3: Entity sends a message translated from French to Russian via English using a machine translation engine.

```

<message xml:lang='fr' from='bard@shakespeare.lit/globe' to='
  playwright@marlowe.lit/theatre'>
  <subject xml:lang='fr'>Bonjour</subject>
  <subject xml:lang='en'>Hello</subject>
  <subject xml:lang='ru'>x443;&#x43B;&#x442;&#x435;</subject>
  <body xml:lang='fr'>comment allez-vous?</body>
  <body xml:lang='en'>How are you?</body>
  <body xml:lang='ru'>&#x41A;&#x430;&#x43A; &#x432;&#x44B;?</body>
  <x xmlns='urn:xmpp:langtrans'>
    <translation destination_lang='en' engine='SYSTRANS' source_lang='
      fr' />
    <translation destination_lang='ru' engine='SYSTRANS' source_lang='
      en' />
  </x>
</message>

```

4.2 Discovering Translation Providers

4.2.1 Discovering Translation Providers On a Server

When connected to a server, a XMPP entity can locate translation providers by asking a server which translation providers are attached to the server; this MUST be done using [Service Discovery \(XEP-0030\)](#)². The server SHOULD return the availability of of translation providers and language pairings for which the user has rights to use.

²XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

Listing 4: Entity sends discovery request to server

```
<iq type='get' id='disco1' to='shakespeare.lit'>
  <query xmlns='http://jabber.org/protocol/disco#items' />
</iq>
```

Listing 5: Server returns items, including translation providers

```
<iq type='result' id='disco1' from='shakespeare.lit' to='
  bard@shakespeare.lit/globe'>
  <query xmlns='http://jabber.org/protocol/disco#items'>
    ...
    <item jid='towerofbabel@shakespeare.lit'
      name='Tower_of_Babel_Translation_Bot' />
    <item jid='translation.shakespeare.lit'
      name='Translation_Provider_Service' />
    ...
  </query>
</iq>
```

4.2.2 Discovering Identity of Providers

Service Discovery is used to determine if a JID provides translation services. The JID can also be a bot (e.g., <towerofbabel@shakespeare.lit>) or a server component (e.g., <translation.shakespeare.lit>).

Listing 6: Entity queries service regarding identity

```
<iq type='get' to='translation.shakespeare.lit' from='bard@shakespeare
  .lit/globe'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 7: Service reports identity

```
<iq type='result' to='bard@shakespeare.lit/globe' from='translation.
  shakespeare.lit'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <identity category='automation' type='translation' />
    <feature var='urn:xmpp:langtrans' />
    ...
  </query>
</iq>
```

4.2.3 Discovering Language Support

The supported languages and other details for the service must be known to use it. It is permissible for a translation service to provide multiple translation engines for the same

language pairing -- if this is done, then a separate <item/> tag MUST be used for each pairing. A 'dictionary' attribute MAY be used to specify the dictionary for a specific <item/>. In order to specify more than one dictionary for a given language pairing then a separate <item/> tag MUST be used for each dictionary specification for that language pairing.

Listing 8: Entity queries service for further information

```
<iq type='get' to='translation.shakespeare.lit' from='bard@shakespeare
.lit/globe'>
  <query xmlns='urn:xmpp:langtrans:items' />
</iq>
```

Listing 9: Service replies with language details

```
<iq type='result' to='bard@shakespeare.lit/globe' from='_translation.
shakespeare.lit'>
  <query xmlns='urn:xmpp:langtrans:items'>
    <item source_lang='en' jid='translation.shakespeare.lit'
      destination_lang='fr'
      engine='SYSTRANS_2005_Release_2' pivotable='true' />
    <item source_lang='en' jid='translation.shakespeare.lit'
      destination_lang='ko'
      engine='SYSTRANS_2005_Release_2' pivotable='true' />
    <item source_lang='en' jid='translation.shakespeare.lit'
      destination_lang='ru'
      engine='SYSTRANS_2005_Release_2' pivotable='true' />
    <item source_lang='en' jid='translation.shakespeare.lit'
      destination_lang='ru'
      engine='SYSTRANS_2005_Release_2' pivotable='true' dictionary
        ='medical' />
    <item source_lang='fr' jid='translation.shakespeare.lit'
      destination_lang='en'
      engine='SYSTRANS_2005_Release_2' pivotable='true' dictionary
        ='standard' />
    <item source_lang='ru' jid='translation.shakespeare.lit'
      destination_lang='en'
      engine='SYSTRANS_2005_Release_2' pivotable='true' dictionary
        ='Medical_1.0' />
    <item source_lang='ko' jid='translation.shakespeare.lit'
      destination_lang='en'
      engine='SYSTRANS_2005_Release_2' pivotable='true' />
  </query>
</iq>
```

4.3 Requesting a Translation from a Service

4.3.1 Requesting a Basic Translation

To request service from a translation provider you can send a message to a provider requesting translations. The lack of a 'source_lang' attribute in the <translation/> element indicates a request for a translation.

Listing 10: Entity requests a translation from English to French

```
<iq from='bard@shakespeare.lt/globe' id='translationReq_2' to='
  translation.shakespeare.lit' type='get'>
  <x xmlns='urn:xmpp:langtrans'>
    <source xml:lang='en'>How are you?</source>
    <translation destination_lang='fr' />
  </x>
</iq>
```

Listing 11: Translation is returned from translation provider

```
<iq type='result' id='translationReq_2' from='translation.shakespeare.
  lit' to='bard@shakespeare.lt/globe'>
  <x xmlns='urn:xmpp:langtrans'>
    <source xml:lang='en'>hello, how are you?</source>
    <translation destination_lang='fr' source_lang='en' engine='
      default'>comment allez-vous?</translation>
  </x>
</iq>
```

4.3.2 Requesting a Translation With Multiple Destination Languages

Listing 12: bard requests a translation from English to Italian and German

```
<iq from='bard@shakespeare.lt/globe' id='translationReq_4' to='
  translation.shakespeare.lit' type='get'>
  <x xmlns='urn:xmpp:langtrans'>
    <source xml:lang='en'>How are you?</source>
    <translation destination_lang='it' />
    <translation destination_lang='de' />
  </x>
</iq>
```

Listing 13: Translation is returned from translation provider

```
<iq type='result' id='translationReq_4' from='translation.shakespeare.
  lit' to='bard@shakespeare.lt/globe'>
  <x xmlns='urn:xmpp:langtrans'>
    <source xml:lang='en'>How are you?</source>
```

```

<translation destination_lang='it' source_lang='en' engine='
  default'>Come siete?</translation>
<translation destination_lang='de' source_lang='en' engine='
  default'>Wie geht es Ihnen?</translation>
</x>
</iq>

```

4.3.3 Requesting a Translation With a Specific Dictionary

If a specific dictionary is required you MAY request a dictionary. This SHOULD have been returned when discoing the server although a dictionary MAY be requested which was not. The dictionaries are translation engine specific and are free form text.

Listing 14: Requests a translation from English to French using the 'medical' dictionary

```

<iq from='bard@shakespeare.lt/globe' id='translationReq_6' to='
  translation.shakespeare.lit' type='get'>
  <x xmlns='urn:xmpp:langtrans'>
    <source xml:lang='en'>How are you?</source>
    <translation destination_lang='fr' dictionary='medical' />
  </x>
</iq>

```

Listing 15: Translation provider returns translation with dictionary details

```

<iq type='result' id='translationReq_6' from='translation.shakespeare.
  lit' to='bard@shakespeare.lt/globe'>
  <x xmlns='urn:xmpp:langtrans'>
    <source xml:lang='en'>hello, how are you?</source>
    <translation
      destination_lang='fr'
      dictionary='medical'
      engine='default'
      source_lang='en'>comment allez-vous?</translation>
    </x>
</iq>

```

If the translation service cannot complete the translation it SHOULD return a <item-not-found/> error indicating some part of the translation request was problematic, unless doing so would violate the privacy and security considerations in XMPP Core and XMPP IM, or local security and privacy policies.

Listing 16: Translation could not be completed

```

<iq type='error' id='translationReq_7' from='translation.shakespeare.
  lit' to='bard@shakespeare.lt/globe'>
  <x xmlns='urn:xmpp:langtrans'>

```

```

    <source xml:lang='en'>How are you?</source>
    <translation destination_lang='dy' />
  </x>
  <error type='modify'>
    <bad-request xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

If privacy or security considerations make returning an `<item-not-found/>` error not feasible it SHOULD return a `<service-unavailable/>` error.

Listing 17: Service unavailable

```

<iq type='error' id='translationReq_7' from='translation.shakespeare.
  lit' to='bard@shakespeare.lt/globe'>
  <x xmlns='urn:xmpp:langtrans'>
    <source xml:lang='en'>How are you?</source>
    <translation destination_lang='dy' />
  </x>
  <error type='cancel'>
    <service-unavailable xmlns='urn:ietf:xml:params:ns:xmpp-stanzas' />
  </error>
</iq>

```

5 Implementation Notes

In order to reduce user confusion and misunderstanding of a translated message body, it is RECOMMENDED that implementations of langtran implement the following user interface features.

1. Translated messages should be clearly identified as being a translation.
2. The display of translated message should clearly show how (automated, manual, automated with human review) a message was translated.
3. The display of a message should clearly show if the translation is the destination, original or pivot language.
4. If pivoting is used, the destination message text should be marked in such a way as to indicate that it was translated on one or more pivot languages, what those language are, in what order they were used, and the actual pivot language text should be accessible to the user.
5. It is recommended that only one level of pivoting be used as quality of the destination translation degrades significantly after each pivot.

Note: The 'reviewed' and 'pivotable' attributes are of type "boolean" and MUST be handled accordingly.³

6 Internationalization Considerations

In order to properly process multi-language messages, clients MUST implement support for multiple message bodies differentiated by the 'xml:lang' attribute as described in RFC 6120.

7 Security Considerations

Potential attacks may be easier against services that implement translation because of the potential disclosure of information regarding language pairings, engines, and dictionaries used however no specific vulnerabilities are introduced.

This possible weakness can be mitigated by not returning specifics to requesting entities and the responding entity MAY perform authorization checks in order to determine how to respond.

8 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](https://www.iana.org/)⁴.

9 XMPP Registrar Considerations

9.1 Protocol Namespaces

The [XMPP Registrar](https://xmpp.org/registrars/)⁵ includes 'urn:xmpp:langtrans' and 'urn:xmpp:langtrans:items' within its registry of protocol namespaces (see [<https://xmpp.org/registrars/namespaces.html>](https://xmpp.org/registrars/namespaces.html)). Note: Before version 1.1 of this specification, the name of the items namespace was urn:xmpp:langtrans#items, however the '#' character is not recommended in URN syntax (see

³In accordance with Section 3.2.2.1 of XML Schema Part 2: Datatypes, the allowable lexical representations for the xs:boolean datatype are the strings "0" and "false" for the concept 'false' and the strings "1" and "true" for the concept 'true'; implementations MUST support both styles of lexical representation.

⁴The 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/>](http://www.iana.org/).

⁵The 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/registrars/>](https://xmpp.org/registrars/).

RFC 2141⁶) so the name was changed to urn:xmpp:langtrans:items.

9.2 Service Discovery Identities

The XMPP Registrar includes type of "translation" in the "automation" category within its registry of service discovery identities (see <<https://xmpp.org/registrar/disco-categories.html>>).

10 XML Schema

10.1 langtrans

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns='urn:xmpp:langtrans'
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:langtrans'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0171: http://www.xmpp.org/extensions/xep-0171.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='x'>
    <xs:complexType>
      <xs:element ref='source' use='optional' />
      <xs:sequence>
        <xs:element ref='translation' use='required' minOccurs='1' />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name='source'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='xs:string'>
          <xs:attribute ref='xml:lang' use='required' />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
```

⁶RFC 2141: URN Syntax <<http://tools.ietf.org/html/rfc2141>>.

```

<xs:element name='translation'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='charset' type='xs:string' use='optional'
          />
        <xs:attribute name='source_lang' type='xs:language' use='
          optional' />
        <xs:attribute name='destination_lang' type='xs:language' use='
          required' />
        <xs:attribute name='dictionary' type='xs:string' use='
          optional' />
        <xs:attribute name='engine' type='xs:string' use='optional'
          />
        <xs:attribute name='reviewed' type='xs:boolean' use='
          optional' default='false' />
      </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>

```

10.2 langtrans:items

```

<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns='urn:xmpp:langtrans:items'
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:langtrans:items'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0171: http://www.xmpp.org/extensions/xep-0171.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='query'>
    <xs:complexType>

```

```
<xs:sequence>
  <xs:element ref='item' minOccurs='0' maxOccurs='unbounded' />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name='item'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='dictionary' type='xs:string' />
        <xs:attribute name='destination_lang' type='xs:language' />
        <xs:attribute name='engine' type='xs:string' use='optional' />
      >
        <xs:attribute name='jid' type='xs:string' use='required' />
        <xs:attribute name='name' type='xs:string' use='optional' />
        <xs:attribute name='pivotable' type='xs:boolean' use='
          optional' default='false' />
        <xs:attribute name='source_lang' type='xs:language' use='
          required' />
      </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>
```