



XMPP

XEP-0328: JID Preparation and Validation Service

Lance Stout
<mailto:lance@lance.im>
<xmpp:lance@lance.im>

Florian Schmaus
<mailto:flo@geekplace.eu>
<xmpp:flo@geekplace.eu>

2019-12-16
Version 0.2.1

Status	Type	Short Name
Deferred	Standards Track	jid-prep

This specification defines a way for an XMPP entity to request another entity to prepare and validate a given JID.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 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	Discovering support	1
3	Use Cases	1
4	Optional Base64 Encoding	3
5	Implementation Notes	3
6	Security Considerations	4
7	IANA Considerations	4
8	XMPP Registrar Considerations	4
	8.1 Protocol Namespaces	4
	8.2 Protocol Versioning	4
9	XML Schema	4

1 Introduction

To accurately compare and match JIDs, a normalization step is required by [RFC 7622](#)¹. This normalization is split into two steps called 'Preparation' and 'Enforcement' by [RFC 8264](#)² and maybe resource-intensive. The protocol defined herein, called "JID Prep" can be used to ask XMPP entities to perform such a normalization. Amongst other use-cases, this protocol can be used for testing purposes, where a test suite checks the conformance of an normalization implementation of, e.g., an XMPP server implementation.

Furthermore, in some environments, especially ones like IoT where devices with constraint resources are used, a client may not have access to the various Unicode and internationalization libraries necessary to properly normalize a JID. For those situations, this protocol can be used by a client to ask a server to normalize a JID on its behalf.

2 Discovering support

If a server supports JID Prep queries, it MUST specify the 'urn:xmpp:jidprep:1' feature in its service discovery information features as specified in [Service Discovery \(XEP-0030\)](#)³.

Listing 1: Server advertises support for JID prep queries

```
<iq type="result"
  to="juliet@capulet.lit/balcony"
  from="capulet.lit"
  id="info1">
  <query xmlns="http://jabber.org/protocol/disco#info">
    <identity category="component" type="jidprep" />
    ...
    <feature var="urn:xmpp:jidprep:1" />
  </query>
</iq>
```

3 Use Cases

To request preparation and enforcement, and thus normalization and validation, of a string to a JID, the client sends a JID Prep request to the service. This request is an <iq/> of type 'get', containing a <jid-validate-request/> element qualified by the 'urn:xmpp:jidprep:1' namespace. This element contains a <maybe-jid/> child element whose textual content is the

¹RFC 7622: Extensible Messaging and Presence Protocol (XMPP): Address Format <<http://tools.ietf.org/html/rfc7622>>.

²RFC 8264: PRECIS Framework: Preparation, Enforcement, and Comparison of Internationalized Strings in Application Protocols <<http://tools.ietf.org/html/rfc8264>>.

³XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

string to normalize to and validate as JID.

Listing 2: JID prep request (valid JID)

```
<iq type='get'
  from='user@example.org/resource'
  to='example.org'
  id='request1'>
  <jid-validate-request xmlns='urn:xmpp:jidprep:1'>
    <maybe-jid> @example.com/resource</maybe-jid>
  </jid-validate-request>
</iq>
```

If the given string can be enforced to a valid JID, then a <jid-validate-result/> with a <valid-jid/> child element is returned. This child element contains the normalized JID parts. The <valid-jid> element MUST contain <domainpart/> and may contain <localpart/> and/or <resourcepart/> elements. If the JID does not contain a local- or resourcepart, then the corresponding element is omitted. Those elements contain the normalized, i.e., PRECIS enforced, strings of the own JID parts.

Listing 3: Valid JID result

```
<iq type='result'
  from='example.org'
  to='user@example.org/resource'
  id='request1'>
  <jid-validate-result xmlns='urn:xmpp:jidprep:1'>
    <valid-jid>
      <localpart> </localpart>
      <domainpart>example.com</domainpart>
      <resourcepart>resource</resourcepart>
    <valid-jid>
  </jid-validate-result>
</iq>
```

If the service is given an invalid JID, a <jid-validate-result/> IQ result response with an <invalid-jid/> element is returned.

Listing 4: JID Prep request (invalid JID)

```
<iq type='get'
  from='user@example.org/resource'
  to='example.org'
  id='request2'>
  <jid-validate-request xmlns='urn:xmpp:jidprep:1'>
    <maybe-jid> henry@example.com</maybe-jid>
  </jid-validate-request>
</iq>
```

Listing 5: Invalid JID result

```

<iq type='result'
  from='example.org'
  to='user@example.org/resource'
  id='request2'>
  <jid-validate-result xmlns='urn:xmpp:jidprep:1'>
    <invalid-jid>
      <reason>Invalid codepoint ' ' in localpart.</reason>
    </invalid-jid>
  </jid-validate-result>
</iq>

```

4 Optional Base64 Encoding

To be able to feed arbitrary strings into the validator, not being limited by XML 1.0, the protocol supports an optional Base64 encoding of the string to validate. The requestor first encodes the string to UTF-8, then encodes the UTF-8 byte-sequence using Base64 (RFC 4648⁴ § 4) and places the result into an `<base64-maybe-jid/>` element. This element is a put under the `<jid-validate-base64-request/>` IQ child element qualified by the `'urn:xmpp:jidprep:1'` namespace.

The IQ responses of the service are identical to the ones of the non-Base64 case.

If supported, service announces `'urn:xmpp:jidprep:base64:1'` feature.

Listing 6: JID Prep request using Base64

```

<iq type='get'
  from='user@example.org/resource'
  to='example.org'
  id='request1'>
  <jid-validate-base64-request xmlns='urn:xmpp:jidprep:1'>
    <base64-maybe-jid>zqNAZXhhbXBsZS5jb20vcmVzb3VyY2U=</base64-maybe-jid>
  </jid-validate-base64-request>
</iq>

```

5 Implementation Notes

If a client has the ability to perform the normalization process itself, it SHOULD NOT make use of the protocol defined herein.

Upon a successful response, it is RECOMMENDED that the client caches the result, mapping the original JID to the normalized version.

⁴RFC 4648: The Base16, Base32, and Base64 Data Encodings <<http://tools.ietf.org/html/rfc4648>>.

6 Security Considerations

As the process for normalizing and validating a JID can be resource intensive, there is a possibility for denial of service attacks. A server MAY rate limit the number of requests to prevent such attacks. Likewise, the server MAY restrict access to the service to requests from users that are local to the server or otherwise trusted.

7 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁵.

8 XMPP Registrar Considerations

8.1 Protocol Namespaces

The [XMPP Registrar](#)⁶ includes 'urn:xmpp:jidprep:1' in its registry of protocol namespaces (see <https://xmpp.org/registrar/namespaces.html>).

- urn:xmpp:jidprep:1

8.2 Protocol Versioning

If the protocol defined in this specification undergoes a revision that is not fully backwards-compatible with an older version, the XMPP Registrar shall increment the protocol version number found at the end of the XML namespaces defined herein, as described in Section 4 of XEP-0053.

9 XML Schema

TODO: Add once the XEP leaves the experimental state.

⁵The 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/>.

⁶The 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/>.