



XMPP

XEP-0049: Private XML Storage

Peter Saint-Andre

<mailto:stpeter@stpeter.im>

<xmpp:stpeter@jabber.org>

<https://stpeter.im/>

Russell Davis

<mailto:ukscone@burninghorse.com>

<xmpp:ukscone@burninghorse.com>

2004-03-01

Version 1.2

Status	Type	Short Name
Active	Historical	iq-private

This specification provides canonical documentation of the 'jabber:iq:private' namespace currently in common usage.

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	jabber:iq:private Namespace	1
2.1	Description	1
2.2	Methods	1
2.3	Elements	1
3	Error Conditions	4
4	Security Considerations	4
5	IANA Considerations	4
6	XMPP Registrar Considerations	5
7	XML Schema	5

1 Introduction

The 'jabber:iq:private' namespace has previously been documented in the Jabber Programmers Guide, but not in a canonical form such as the Internet-Drafts or a XEP. This specification documents the existing usage of jabber:iq:private.

2 jabber:iq:private Namespace

2.1 Description

A Jabber client can store any arbitrary XML on the server side by sending an <iq/> stanza of type "set" to the server with a <query/> child scoped by the 'jabber:iq:private' namespace. The <query/> element MAY contain any arbitrary XML fragment as long as the root element of that fragment is scoped by its own namespace. The data can then be retrieved by sending an <iq/> stanza of type "get" with a <query/> child scoped by the 'jabber:iq:private' namespace, which in turn contains a child element scoped by the namespace used for storage of that fragment. Using this method, Jabber entities can store private data on the server and retrieve it whenever necessary. The data stored might be anything, as long as it is valid XML. One typical usage for this namespace is the server-side storage of client-specific preferences; another is [Bookmark Storage \(XEP-0048\)](#) ¹.

2.2 Methods

get	Sent with a blank query to retrieve the private data from the server.
set	Sent with the private XML data contained inside of a query.
result	Returns the private data from the server.
error	There was an error processing the request. The exact error can be found in the child error element.

2.3 Elements

The root element of this namespace is query. At least one child element with a proper namespace MUST be included; otherwise the server MUST respond with a "Not Acceptable" error (see [Error Condition Mappings \(XEP-0086\)](#) ² for information about error conditions). A client MUST NOT query for more than one namespace in a single IQ get request. However, an IQ set or result MAY contain multiple elements qualified by the same namespace.

¹XEP-0048: Bookmark Storage <<https://xmpp.org/extensions/xep-0048.html>>.

²XEP-0086: Error Condition Mappings <<https://xmpp.org/extensions/xep-0086.html>>.

Listing 1: Client Stores Private Data

```
CLIENT:
<iq type="set" id="1001">
  <query xmlns="jabber:iq:private">
    <exodus xmlns="exodus:prefs">
      <defaultnick>Hamlet</defaultnick>
    </exodus>
  </query>
</iq>

SERVER:
<iq type="result"
  from="hamlet@shakespeare.lit/denmark"
  to="hamlet@shakespeare.lit/denmark"
  id="1001"/>
```

Listing 2: Client Retrieves Private Data

```
CLIENT:
<iq type="get" id="1002">
  <query xmlns="jabber:iq:private">
    <exodus xmlns="exodus:prefs"/>
  </query>
</iq>

SERVER:
<iq type="result"
  from="hamlet@shakespeare.lit/denmark"
  to="hamlet@shakespeare.lit/denmark"
  id="1002">
  <query xmlns="jabber:iq:private">
    <exodus xmlns="exodus:prefs">
      <defaultnick>Hamlet</defaultnick>
    </exodus>
  </query>
</iq>
```

If a user attempts to get or set jabber:iq:private data that belongs to another user, the server MUST return a "Forbidden" or "Service Unavailable" error to the sender (the latter condition is in common use by existing implementations, although the former is preferable).

Listing 3: User Attempts to Get or Set Data for Another User

```
CLIENT:
<iq type="set" to="hamlet@shakespeare.lit" id="1003">
  <query xmlns="jabber:iq:private">
    <exodus xmlns="exodus:prefs">
      <defaultnick>Macbeth</defaultnick>
    </exodus>
  </query>
</iq>
```

```

    </query>
</iq>

SERVER:
<iq type="error"
  from="hamlet@shakespeare.lit"
  to="macbeth@shakespeare.lit"
  id="1003">
  <query xmlns="jabber:iq:private">
    <exodus xmlns="exodus:prefs">
      <defaultnick>Macbeth</defaultnick>
    </exodus>
  </query>
  <error code="403" type="cancel">
    <forbidden
      xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

If a user attempts to perform an IQ get without providing a child element, the server SHOULD return a "Bad Format" error (however, some existing implementations return a "Not Acceptable" error in such circumstances):

Listing 4: User Attempts to Get Data Without Specifying Child Element/Namespace

```

CLIENT:
<iq type="set" id="1004">
  <query xmlns="jabber:iq:private"/>
</iq>

SERVER:
<iq type="error" id="1004">
  <query xmlns="jabber:iq:private"/>
  <error code="406" type="modify">
    <bad-format
      xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

Certain namespaces are reserved in Jabber (namespaces beginning with 'jabber:' or 'http://jabber.org/', as well as 'vcard-temp'). If a user attempts to get or set jabber:iq:private data in a reserved namespace, historically some server implementations have chosen to return an error (commonly "Not Acceptable") to the sender. Such behavior is OPTIONAL, but may be encountered by clients when interacting with some existing server implementations.

Listing 5: User Attempts to Get or Set Data in a Reserved Namespace

```
CLIENT:
```

```

<iq type="set" id="1005">
  <query xmlns="jabber:iq:private">
    <vCard xmlns="vcard-temp">
      <FN>Hamlet</FN>
    </vCard>
  </query>
</iq>

SERVER (optional error):
<iq type="error" id="1005">
  <query xmlns="jabber:iq:private">
    <vCard xmlns="vcard-temp">
      <FN>Hamlet</FN>
    </vCard>
  </query>
  <error code="406" type="modify">
    <not-acceptable
      xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

3 Error Conditions

Condition	Description
Bad Format	The IQ get does not contain a child element.
Forbidden	The IQ get or set is sent to a JID other than that of the sender.
Not Acceptable	The IQ get or set is qualified by a reserved namespace.

4 Security Considerations

A server MUST NOT allow any entity other than an authorized resource of the user to create, update, or delete private XML stored on behalf of that user.

5 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)³.

³The 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/>.

6 XMPP Registrar Considerations

No action on the part of the [XMPP Registrar](#)⁴ is necessary as a result of this document, since 'jabber:iq:private' is already a registered namespace.

7 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='jabber:iq:private'
  xmlns='jabber:iq:private'
  elementFormDefault='qualified'>
  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0049: http://www.xmpp.org/extensions/xep-0049.html
    </xs:documentation>
  </xs:annotation>
  <xs:element name='query'>
    <xs:complexType>
      <xs:sequence minOccurs='0'>
        <xs:any namespace='##other' />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

⁴The 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/>.