XEP-0402: PEP Native Bookmarks

Dave Cridland
mailto:dave@hellopando.com
xmpp:dwd@dave.cridland.net

JC Brand
mailto:jc@opkode.com
xmpp:jc@opkode.com

2024-08-15
Version 1.2.0

Status	Type	Short Name
Draft	Standards Track	bookmarks2

This specification defines a syntax and storage profile for keeping a list of chatroom bookmarks on the server.

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

The original Bookmarks specification (Bookmark Storage (XEP-0048) [1]) used the widely available Private XML Storage (Private XML Storage (XEP-0049) [2]), but stored all bookmarks in a single element. When the specification was moved to the Standards Track and Draft, it was also updated to use the user's Pubsub service (Best Practices for Persistent Storage of Private Data via Publish-Subscribe (XEP-0223) [3]), but kept this single element containing all bookmarks inside a single Pubsub item.

Most implementations have kept to the original, Private XML Storage based solution, and while some newer implementations have used Pubsub, these are limited in capability by the use of a single item, which prevents safe atomic updates of individual bookmarks.

Finally, while some clients used custom XML elements to store additional private metadata about bookmarks, this was usually stripped when any bookmark was edited by another client. This specification resolves all three issues by providing a new Bookmarks specification to migrate to, and takes the opportunity to update the XML namespace in use as well. The URL storage is dropped, since it is rarely used. Storage of URL bookmarks is therefore out of scope. This specification was originally entitled "Bookmarks 2: This Time It's Serious". Any implication of a sense of humour has been removed with the change in title.

2 Outline of use

Clients store each bookmarked chatroom as a Pubsub item within the 'urn:xmpp:bookmarks:1' node. Each item SHALL have, as item id, the Room JID of the chatroom (eg, coven@chat.shakespeare.lit). While a client can typically assume a chatroom based on Multi-User Chat (XEP-0045) [4], clients are free to store chatrooms based on any particular groupchat protocol.

The payload of the item SHALL be a conference element qualified by the 'urn:xmpp:bookmarks:1' namespace, with the following syntax:

---

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

[2] XEP-0049: Private XML Storage <https://xmpp.org/extensions/xep-0049.html>.

[3] XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <https://xmpp.org/extensions/xep-0223.html>.

[4] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

Element or Attribute	Definition	Datatype	Inclusion
'autojoin' attribute	Whether the client should automatically join the conference room on login.	boolean defaulting to false In 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.	OPTIONAL
'name' attribute	A friendly name for the bookmark, specified by the user. Clients SHOULD NOT attempt to auto-generate this from the JID.	string	OPTIONAL

Element or Attribute	Definition	Datatype	Inclusion
<nick/> element	The user's preferred roomnick for the chatroom, if different to that specified by User Nickname (XEP-0172) XEP-0172: User Nickname <https://xmpp.org/extensions/xep-0172.html>.. In the absence of this element being present, the nickname from User Nickname (XEP-0172) XEP-0172: User Nickname <https://xmpp.org/extensions/xep-0172.html>. SHOULD be used if present.	string	OPTIONAL
<password/> element	A password used to access the chatroom. Note this is not intended to be a secure storage.	string	OPTIONAL
<extensions/> element	A set of child elements (of potentially any namespace). Clients MUST preserve these (particularly preserving unknown elements) when editing items.	XML Elements	OPTIONAL

Note: The datatypes are as defined in XML Schema Part 2 [5].

Listing 1: An example of the conference element

---

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

```
<conference xmlns='urn:xmpp:bookmarks:1'
            name='Council␣of␣Oberon'
            autojoin='true'>
  <nick>Puck</nick>
</conference>
```

This bookmark would be displayed as 'Council of Oberon' and, if activated, would attempt to join the conference room 'council@conference.underhill.org' with nickname 'Puck'.

Note that a bookmark item MUST contain only one conference room.

Note also that a conference element has no truly mandatory attributes or child elements, thus the following is legal:

Listing 2: Minimalist conference element

```
<conference xmlns='urn:xmpp:bookmarks:1'/>
```

3   Workflow

3.1   Registering to receive notifications

A client interested in bookmarks SHOULD include the 'urn:xmpp:bookmarks:1+notify' feature in its Entity Capabilities (XEP-0115) [6], as per Personal Eventing Protocol (XEP-0163) [7], so that it receives notifications for updates done by other clients of the user, and reacts accordingly. The actual notifications are explained in the Bookmark Notifications section of this specification.

Listing 3: Client replies with +notify to a XEP-0030 query from its server

```
<iq from='juliet@capulet.lit/balcony' to='capulet.lit' type='result'
   id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <identity category='client' type='pc' name='poezio'/>
    <feature var='http://jabber.org/protocol/disco#info'/>
    <feature var='urn:xmpp:bookmarks:1+notify'/>
    <!-{}- … -{}->
  </query>
</iq>
```

3.2   Retrieving all bookmarks

Once connected, a client first retrieves the current list of bookmarks. It then SHOULD join every MUC identified by the items' 'id' attribute that have an 'autojoin' attribute that is set to

---

[6]XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.
[7]XEP-0163: Personal Eventing Protocol <https://xmpp.org/extensions/xep-0163.html>.

"true" or "1".

NOTE: A future version of this specification might refer to PubSub Since (XEP-0312) [8] or a similar protocol to reduce the need for full synchronisation on each connection.

Listing 4: Client retrieves all bookmarks

```
<iq from='juliet@capulet.lit/balcony' type='get' id='retrieve1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='urn:xmpp:bookmarks:1'/>
  </pubsub>
</iq>
```

Listing 5: Server returns the bookmarks

```
<iq type='result'
    to='juliet@capulet.lit/balcony'
    id='retrieve1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='urn:xmpp:bookmarks:1'>
      <item id='theplay@conference.shakespeare.lit'>
        <conference xmlns='urn:xmpp:bookmarks:1'
                    name='The␣Play&apos;s␣the␣Thing'
                    autojoin='true'>
          <nick>JC</nick>
        </conference>
      </item>
      <item id='orchard@conference.shakespeare.lit'>
        <conference xmlns='urn:xmpp:bookmarks:1'
                    name='The␣Orcard'
                    autojoin='1'>
          <nick>JC</nick>
          <extensions>
            <state xmlns='http://myclient.example/bookmark/state'
                minimized='true'/>
          </extensions>
        </conference>
      </item>
    </items>
  </pubsub>
</iq>
```

3.3 Adding a bookmark

Adding a bookmark means publishing a new item, with the bookmark JID as id, to the 'urn:xmpp:bookmarks:1' node.

publish-options (as defined in XEP-0060) MUST be supported by the server in order to check

---

[8]XEP-0312: PubSub Since <https://xmpp.org/extensions/xep-0312.html>.

that the node is correctly configured before publishing a new conference. This is especially important to avoid leaking your bookmarks to your contacts for instance.

Listing 6: Client adds a new bookmark

```
<iq from='juliet@capulet.lit/balcony' type='set' id='pip1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:bookmarks:1'>
      <item id='theplay@conference.shakespeare.lit'>
        <conference xmlns='urn:xmpp:bookmarks:1'
                    name='The Play&apos;s the Thing'
                    autojoin='true'>
          <nick>JC</nick>
        </conference>
      </item>
    </publish>
    <publish-options>
      <x xmlns='jabber:x:data' type='submit'>
        <field var='FORM_TYPE' type='hidden'>
          <value>http://jabber.org/protocol/pubsub#publish-options</
              value>
        </field>
        <field var='pubsub#persist_items'>
          <value>true</value>
        </field>
        <field var='pubsub#max_items'>
          <value>max</value>
        </field>
        <field var='pubsub#send_last_published_item'>
          <value>never</value>
        </field>
        <field var='pubsub#access_model'>
          <value>whitelist</value>
        </field>
      </x>
    </publish-options>
  </pubsub>
</iq>
```

Listing 7: Server acknowledges successful storage

```
<iq to='juliet@capulet.lit/balcony' type='result' id='pip1'/>
```

3.4   Editing a bookmark

Editing a bookmark means republishing the item, with the same bookmark JID as id, to the 'urn:xmpp:bookmarks:1' node.
Note that clients MUST preserve any XML elements they do not understand, particularly

including unknown elements, within the <extensions/> element of the bookmark.
publish-options (as defined in XEP-0060) MUST be supported by the server in order to check
that the node is correctly configured before publishing a new conference. This is especially
important to avoid leaking your bookmarks to your contacts for instance.

Listing 8: Client corrects typo in name of bookmark

```
<iq from='juliet@capulet.lit/balcony' type='set' id='pip2'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:bookmarks:1'>
      <item id='orchard@conference.shakespeare.lit'>
        <conference xmlns='urn:xmpp:bookmarks:1'
                    name='The_Orchard'
                    autojoin='true'>
          <nick>JC</nick>
          <extensions>
            <state xmlns='http://myclient.example/bookmark/state'
                minimized='true'/>
          </extensions>
        </conference>
      </item>
    </publish>
    <publish-options>
      <x xmlns='jabber:x:data' type='submit'>
        <field var='FORM_TYPE' type='hidden'>
          <value>http://jabber.org/protocol/pubsub#publish-options</
              value>
        </field>
        <field var='pubsub#persist_items'>
          <value>true</value>
        </field>
        <field var='pubsub#max_items'>
          <value>max</value>
        </field>
        <field var='pubsub#send_last_published_item'>
          <value>never</value>
        </field>
        <field var='pubsub#access_model'>
          <value>whitelist</value>
        </field>
      </x>
    </publish-options>
  </pubsub>
</iq>
```

Listing 9: Server acknowledges successful storage

```
<iq to='juliet@capulet.lit/balcony' type='result' id='pip2'/>
```

3.5 Removing a bookmark

Removing a bookmark means retracting an existing item, identified by the bookmark's JID, form the 'urn:xmpp:bookmarks:1' node.
This implies that server support for the "delete-items" pubsub feature is REQUIRED.
A 'notify' attribute SHOULD be included on the <retract/> element in order to inform other online clients of the deletion.

Listing 10: Client removes a new bookmark

```
<iq from='juliet@capulet.lit/balcony' type='set' id='remove-bookmark1'
    >
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <retract node='urn:xmpp:bookmarks:1' notify='true'>
      <item id='theplay@conference.shakespeare.lit'/>
    </retract>
  </pubsub>
</iq>
```

Listing 11: Server acknowledges successful retraction

```
<iq to='juliet@capulet.lit/balcony' type='result' id='remove-bookmark1
    '/>
```

4 Bookmark Notifications

When a client is sent an event from the Pubsub service for the 'urn:xmpp:bookmarks:1' node, it SHOULD join the room immediately if the 'autojoin' attribute is both present and true.

Listing 12: Client receives a new bookmark notification

```
<message from='juliet@capulet.lit' to='juliet@capulet.lit/balcony'
    type='headline' id='new-room1'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='urn:xmpp:bookmarks:1'>
      <item id='theplay@conference.shakespeare.lit'>
        <conference xmlns='urn:xmpp:bookmarks:1'
                    name='The␣Play&apos;s␣the␣Thing'
                    autojoin='1'>
          <nick>JC</nick>
        </conference>
      </item>
    </items>
  </event>
</message>
```

If the bookmark notification's 'autojoin' attribute is set to false, the client SHOULD leave the room immediately.

Listing 13: Client receives a bookmark notification with autojoin='false' (its default value)

```xml
<message from='juliet@capulet.lit' to='juliet@capulet.lit/balcony'
    type='headline' id='unjoined-room1'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='urn:xmpp:bookmarks:1'>
      <item id='theplay@conference.shakespeare.lit'>
        <conference xmlns='urn:xmpp:bookmarks:1'
                    name='The Play&apos;s the Thing'>
          <nick>JC</nick>
        </conference>
      </item>
    </items>
  </event>
</message>
```

On the other hand, if the event is a retract notification, the client SHOULD leave the room immediately.

Listing 14: Client receives a bookmark retraction notification

```xml
<message from='juliet@capulet.lit' to='juliet@capulet.lit/balcony'
    type='headline' id='removed-room1'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='urn:xmpp:bookmarks:1'>
      <retract id='theplay@conference.shakespeare.lit'/>
    </items>
  </event>
</message>
```

5   Implementation Notes

5.1   Differences to XEP-0048

- The conference element does not contain the jid - this is present only in the item id.

- Each conference element is contained within an item.

- The storage MUST be Best Practices for Persistent Storage of Private Data via Publish-Subscribe (XEP-0223) [9]

---

[9]XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <https://xmpp.org/extensions/xep-0223.html>.

5.2 Storage

Publish-Subscribe (XEP-0060) [10] is used for data storage, specifically through the use of private, personal pubsub nodes (described in Best Practices for Persistent Storage of Private Data via Publish-Subscribe (XEP-0223) [11]) hosted at the user's virtual pubsub service (see Personal Eventing Protocol (XEP-0163) [12]).

5.3 Compatibility

A server MAY choose to unify the bookmarks from both Private XML Storage (XEP-0049) [13] based and the current Bookmark Storage (XEP-0048) [14].

It is encouraged to at least support unification between Private XML Storage because as of 2019 this is still the storage backend that is implemented in the majority of clients.

A server that supports unifying bookmarks from Private XML Storage (XEP-0049) [15] and PEP Native Bookmarks (XEP-0402) [16] SHOULD announce the "urn:xmpp:bookmarks:1#compat" feature on the account. Clients may use that feature as an indication that it is safe to store bookmarks using only PEP Native Bookmarks (XEP-0402) [17] without losing backward compatibility to clients that are only using Private XML Storage (XEP-0049) [18].

A server that supports unifying bookmarks between Best Practices for Persistent Storage of Private Data via Publish-Subscribe (XEP-0223) [19] and PEP Native Bookmarks (XEP-0402) [20] SHOULD announce the "urn:xmpp:bookmarks:1#compat-pep" feature on the account.

5.3.1 Publishing via this specification

When a client publishes a new item, the server MAY collate all items, casting them into the 'storage:bookmarks' namespace and setting the jid attribute to the item id in each case. When contained within a storage element qualified by the 'storage:bookmarks' namespace, this will be the correct format for both current and previous variants of Bookmark Storage (XEP-0048) [21]

---

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

[11]XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <https://xmpp.org/extensions/xep-0223.html>.

[12]XEP-0163: Personal Eventing Protocol <https://xmpp.org/extensions/xep-0163.html>.

[13]XEP-0049: Private XML Storage <https://xmpp.org/extensions/xep-0049.html>.

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

[15]XEP-0049: Private XML Storage <https://xmpp.org/extensions/xep-0049.html>.

[16]XEP-0402: PEP Native Bookmarks <https://xmpp.org/extensions/xep-0402.html>.

[17]XEP-0402: PEP Native Bookmarks <https://xmpp.org/extensions/xep-0402.html>.

[18]XEP-0049: Private XML Storage <https://xmpp.org/extensions/xep-0049.html>.

[19]XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <https://xmpp.org/extensions/xep-0223.html>.

[20]XEP-0402: PEP Native Bookmarks <https://xmpp.org/extensions/xep-0402.html>.

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

5.3.2 Publishing via the old specification

If a client publishes a replacement list of bookmarks via the older specifications, a server MAY examine the list and update the individual items as required, sending updates or retraction notifications as needed. Servers electing to perform this OPTIONAL behaviour SHOULD NOT send notifications for unchanged items.

6 Determining Support

This specification relies fully on a number of others. Most particularly, support for this protocol is available if Best Practices for Persistent Storage of Private Data via Publish-Subscribe (XEP-0223) [22] is supported.
Server side unification between Private XML Storage (XEP-0049) [23] bookmarks and PEP Native Bookmarks is announced with the feature "urn:xmpp:bookmarks:1#compat" on the account. Server side unification between the current use of XEP-0048 bookmarks (PEP) is annouced with the feature "urn:xmpp:bookmarks:1#compat-pep" on the account.

7 Acknowledgements

The authors would like to note that much of the syntax description was copied exactly from Bookmark Storage (XEP-0048) [24] by Rachel Blackman, Peter Millard, and Peter Saint-Andre. Much of the remainder of this specification is based closely on their work.

8 Security Considerations

Security considerations related to object persistence via publish-subscribe are described in Publish-Subscribe (XEP-0060) [25] and XEP-0223.
The client needs to make sure that the server actually supports the "http://jabber.org/protocol/pubsub#publish-options" feature, before relying on it. If it's not supported, the client should configure the 'urn:xmpp:bookmarks:1' node first (see xep-0060), before adding any bookmarks.

9 XML Schema

---

[22] XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <https://xmpp.org/extensions/xep-0223.html>.

[23] XEP-0049: Private XML Storage <https://xmpp.org/extensions/xep-0049.html>.

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

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

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

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='urn:xmpp:bookmarks:1'
    xmlns='urn:xmpp:bookmarks:1'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0402: http://www.xmpp.org/extensions/xep-0402.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='conference'>
    <xs:complexType>
      <xs:sequence>
        <xs:element name='nick' type='xs:string' minOccurs='0'/>
        <xs:element name='password' type='xs:string' minOccurs='0'/>
        <xs:element ref='extensions' minOccurs='0' />
      </xs:sequence>
      <xs:attribute name='autojoin' type='xs:boolean' use='optional'
          default='false'/>
      <xs:attribute name='name' type='xs:string' use='optional'/>
    </xs:complexType>
  </xs:element>

  <xs:element name='extensions'>
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace='##other'
                minOccurs='0'
                maxOccurs='unbounded'
                processContents='lax'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

12