



XMPP

XEP-0402: Bookmarks 2 (This Time it's Serious)

Dave Cridland

<mailto:dave.c@threadsstyling.com>

<xmpp:dwd@dave.cridland.net>

Jan-Carel Brand

<mailto:jc@opkode.com>

<xmpp:jc@opkode.com>

2018-07-22

Version 0.2.1

Status	Type	Short Name
Experimental	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 – 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	Outline of use	1
3	Examples	3
3.1	Retrieving all bookmarks	3
3.2	Adding a bookmark	3
3.3	Removing a bookmark	4
4	Bookmark Notifications	5
5	Implementation Notes	5
5.1	Differences to XEP-0048	5
5.2	Storage	5
5.3	Compatibility	5
5.3.1	Publishing via this specification	6
5.3.2	Publishing via the old specification	6
6	Determining Support	6
7	Acknowledgements	6
8	Security Considerations	6

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.

This specification resolves both 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.

2 Outline of use

Clients store each bookmarked chatroom as a Pubsub item within the 'urn:xmpp:bookmarks:0' node. Each item SHALL have, as item id, the Room JID of the chatroom (eg, coven@chat.shakespeare.lit).

The payload of the item SHALL be a conference element qualified by the 'urn:xmpp:bookmarks:0' namespace, with the following syntax:

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

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

³XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <<https://xmpp.org/extensions/xep-0223.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.	string	RECOMMENDED
<nick/> element	The user's preferred roomnick for the chatroom.	string	OPTIONAL

Note: The datatypes are as defined in [XML Schema Part 2](#)⁴.

Listing 1: An example of the conference element

```
<conference xmlns='urn:xmpp:bookmarks:0'
  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, though a name SHOULD be given. Thus the following is legal:

⁴XML Schema Part 2: Datatypes <<http://www.w3.org/TR/xmlschema11-2/>>.

Listing 2: Minimalist conference element

```
<conference xmlns='urn:xmpp:bookmarks:0' />
```

3 Examples

3.1 Retrieving all bookmarks

Listing 3: 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:0' />
  </pubsub>
</iq>
```

Listing 4: 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:0'>
      <item id='theplay@conference.shakespeare.lit'>
        <conference name='The_Play's_the_Thing'
          autojoin='true'>
          <nick>JC</nick>
        </conference>
      </item>
      <item id='orchard@conference.shakespeare.lit'>
        <conference name='The_Orchard'
          autojoin='true'>
          <nick>JC</nick>
        </conference>
      </item>
    </items>
  </pubsub>
</iq>
```

3.2 Adding a bookmark

Adding a bookmark means publishing a new item, with the bookmark JID as id, to the 'urn:xmpp:bookmarks:0' node.

Listing 5: 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:0'>
    <item id='theplay@conference.shakespeare.lit'>
      <conference 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#access_model'>
        <value>whitelist</value>
      </field>
    </x>
  </publish-options>
</pubsub>
</iq>

```

Listing 6: Server acknowledges successful storage

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

3.3 Removing a bookmark

Removing a bookmark means retracting an existing item, identified by the bookmark's JID, from the 'urn:xmpp:bookmarks:0' node.

This implies that server support for the "delete-items" pubsub feature is REQUIRED.

Listing 7: 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:0'>
      <item id='theplay@conference.shakespeare.lit' />
    </retract>
  </pubsub>
</iq>

```

Listing 8: 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:0' node, it MUST check the 'autojoin' attribute if present, and join the room immediately if the attribute is both present and true.

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 \(XEP-0223\)](#) ⁵

5.2 Storage

[Publish-Subscribe \(XEP-0060\)](#) ⁶ 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 \(XEP-0223\)](#) ⁷) hosted at the user's virtual pubsub service (see [Personal Eventing Protocol \(XEP-0163\)](#) ⁸).

5.3 Compatibility

A server MAY choose to unify the bookmarks from both [Private XML Storage \(XEP-0049\)](#) ⁹ based and the current [Bookmark Storage \(XEP-0048\)](#) ¹⁰.

⁵XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <<https://xmpp.org/extensions/xep-0223.html>>.

⁶XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

⁷XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <<https://xmpp.org/extensions/xep-0223.html>>.

⁸XEP-0163: Personal Eventing Protocol <<https://xmpp.org/extensions/xep-0163.html>>.

⁹XEP-0049: Private XML Storage <<https://xmpp.org/extensions/xep-0049.html>>.

¹⁰XEP-0048: Bookmark Storage <<https://xmpp.org/extensions/xep-0048.html>>.

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 \(XEP-0048\)](#)¹¹

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 \(XEP-0223\)](#)¹² is supported.

7 Acknowledgements

The authors would like to note that much of the syntax description was copied exactly from [Bookmark Storage \(XEP-0048\)](#)¹³ 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 XEP-0060 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:0' node first (see

¹¹XEP-0048: Bookmark Storage <<https://xmpp.org/extensions/xep-0048.html>>.

¹²XEP-0223: Best Practices for Persistent Storage of Private Data via Publish-Subscribe <<https://xmpp.org/extensions/xep-0223.html>>.

¹³XEP-0048: Bookmark Storage <<https://xmpp.org/extensions/xep-0048.html>>.

xep-0060), before adding any bookmarks.