



XMPP

XEP-0413: Order-By

Jérôme Poisson

<mailto:goffi@goffi.org>

<xmpp:goffi@jabber.fr>

2019-08-20

Version 0.1.1

Status	Type	Short Name
Experimental	Standards Track	NOT_YET_ASSIGNED

This specification allows to change order of items retrieval in a Pubsub or MAM query

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 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	Requirements	1
3	Glossary	1
4	Use Cases	1
4.1	Retrieve Items By Date of Creation	1
4.2	Retrieve Items By Date of Modification	3
4.3	Use with MAM	4
4.4	Reversing the Order	4
4.5	Extending this Specification	5
5	Discovering Support	5
6	Business Rules	5
7	Implementation Notes	6
8	Security Considerations	6
9	IANA Considerations	6
10	XMPP Registrar Considerations	6
10.1	Protocol Namespaces	6
10.2	Protocol Versioning	6
11	XML Schema	7

1 Introduction

[Publish-Subscribe \(XEP-0060\)](#)¹ §6.5.7 allows to retrieve the "most recent items" and [Message Archive Management \(XEP-0313\)](#)² state in §3.1 that archives are ordered in "chronological order". While this order is straightforward in general use cases, it is sometimes desirable to use a different order, for instance while using [Microblogging Over XMPP \(XEP-0277\)](#)³: a spelling mistake correction should not bring an old blog post to the top of retrieved items. This specification allows to explicitly change business logic to retrieve the items in a different order, in a similar way as the "ORDER BY" clause in SQL.

2 Requirements

- an entity should be able to retrieve items by date of creation or by date of last modification (see below for definitions)
- the specification should be extensible to allow new ordering
- in case of conflicts, a 2nd, 3rd, etc. level of ordering should be possible

3 Glossary

In XEP-0060, there is no such thing as "updated item". This XEP changes the business logic as follow:

- **Date of creation** — date when the item has been published **ONLY if the item has a new id** (i.e. an id which was not already present in the node at the time of publication). If an item reuses an existing id, it overwrites the original item **and the date of creation stays the date of creation of the original item.**
- **Date of modification** — date when the item has been overwritten by a new item of the same id. If the item has never been overwritten, it is equal to the date of creation defined above.

4 Use Cases

4.1 Retrieve Items By Date of Creation

Juliet wants to retrieve plays of her favorite writer, William Shakespeare. She wants to retrieve the 3 most recent ones by date of creation.

¹XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

²XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

³XEP-0277: Microblogging over XMPP <<https://xmpp.org/extensions/xep-0277.html>>.

To do so, her client do a regular Pubsub request, but adds the <order> element as a children of the <pubsub> element with the "urn:xmpp:order-by:0" namespace and with a 'by' attribute equal to "creation".

Listing 1: Retrieving items ordered by date of creation

```
<iq type='get'
  from='juliet@capulet.lit/balcony'
  to='pubsub.shakespeare.lit'
  id='{}'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='plays' max_items='3' />
    <order xmlns='urn:xmpp:order-by:0' by='creation' />
  </pubsub>
</iq>
```

The Pubsub service then returns the 3 most recently created plays, first one being the most recent.

Listing 2: Service returns all items

```
<iq type='result'
  from='pubsub.shakespeare.lit'
  to='juliet@capulet.lit/balcony'
  id='{}'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='plays'>
      <item id='153214214'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>Henry VIII</title>
        </entry>
      </item>
      <item id='623423544'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>Tempest</title>
        </entry>
      </item>
      <item id='452432423'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>Wintter's_Tale</title>
        </entry>
      </item>
    </items>
  </pubsub>
</iq>
```

4.2 Retrieve Items By Date of Modification

Juliet realizes that there is a spelling mistake, it's "Winter's Tale" and not "Wintter's Tale". She fixes it by overwriting the item:

Listing 3: Juliet Overwrite the Item to Fix It

```
<iq type='set'
  from='juliet@capulet.lit/balcony'
  to='pubsub.shakespeare.lit'
  id='orderby2'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='plays'>
      <item id='452432423'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>Winter's_Tale</title>
        </entry>
      </item>
    </publish>
  </pubsub>
</iq>
```

To check that everything is alright, she requests again the last 3 items, but this time by date of modification. To do so, the client proceeds the same way as for date of creation, except that it uses the value "modification" for the 'by' attribute.

Listing 4: Retrieving items ordered by date of modification

```
<iq type='get'
  from='juliet@capulet.lit/balcony'
  to='pubsub.shakespeare.lit'
  id='orderby3'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='plays' max_items='3' />
    <order xmlns='urn:xmpp:order-by:0' by='modification' />
  </pubsub>
</iq>
```

The Pubsub service returns again the 3 plays but the "Winter Tales" item has been overwritten recently, while the 2 others have never been overwritten, so it returns the items in the following order, with the most recently modified item on top:

Listing 5: Service returns all items

```
<iq type='result'
  from='pubsub.shakespeare.lit'
  to='juliet@capulet.lit/balcony'
  id='orderby3'>
```

```

<pubsub xmlns='http://jabber.org/protocol/pubsub'>
  <items node='plays'>
    <item id='452432423'>
      <entry xmlns='http://www.w3.org/2005/Atom'>
        <title>Winter's Tale</title>
      </entry>
    </item>
    <item_id='153214214'>
      <entry xmlns='http://www.w3.org/2005/Atom'>
        <title>Henry VIII</title>
      </entry>
    </item>
    <item_id='623423544'>
      <entry xmlns='http://www.w3.org/2005/Atom'>
        <title>Tempest</title>
      </entry>
    </item>
  </items>
</pubsub>
</iq>

```

4.3 Use with MAM

With [Message Archive Management \(XEP-0313\)](#)⁴ the logic is the same, but the <order> element is added as a child of the <query> element:

Listing 6: MAM Pubsub Query with Ordering

```

<iq to='pubsub.shakespeare.lit' type='set' id='orderby4'>
  <query xmlns='urn:xmpp:mam:2' queryid='123' node='plays'>
    <order xmlns='urn:xmpp:order-by:0' by='creation' />
  </query>
</iq>

```

This way, filters can be used with a specific ordering.

4.4 Reversing the Order

The ordering can be reversed by using the mechanisms already provided by [Result Set Management \(XEP-0059\)](#)⁵.

⁴XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

⁵XEP-0059: Result Set Management <<https://xmpp.org/extensions/xep-0059.html>>.

4.5 Extending this Specification

This specification can be extended by further XEPs, proposing other kind of ordering in the 'by' attribute (e.g. ordering by filename for a file sharing service). But this is beyond the scope of this XEP, and a client should not assume that other ordering than "creation" and "modification" are available without negotiation.

In a similar way, the semantic described here could be reused in other use cases as for Pubsub or MAM, but this would need to be detailed in a separate specification.

5 Discovering Support

If a server supports the "order by" protocol, it MUST advertize it including the "urn:xmpp:order-by:0" discovery feature (see Protocol Namespaces regarding issuance of one or more permanent namespaces) in response to a [Service Discovery \(XEP-0030\)](#)⁶ information request:

Listing 7: Service Discovery information request

```
<iq from='example.org'
  id='disco1'
  to='example.com'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 8: Service Discovery information response

```
<iq from='example.com'
  id='disco1'
  to='example.org'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>...
    <feature var='urn:xmpp:order-by:0' />...
  </query>
</iq>
```

6 Business Rules

Several ordering elements may be used, this allows to solve next levels of ordering in case of conflicts. In this case, the first ordering (i.e. the top most <order> element) is the main one,

⁶XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

the second <order> element is used in case of conflicts, then the next one if a new conflict happens and so on.

In case of conflicts, if no new <order> element is specified, the item order is not guaranteed and is up to the implementation.

7 Implementation Notes

For implementations based on SQL databases, the "ORDER BY" clause can be used to easily implement this specification. Other kind of databases should have similar mechanisms.

8 Security Considerations

This document introduces no additional security considerations above and beyond those defined in the documents on which it depends.

9 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁷.

10 XMPP Registrar Considerations

10.1 Protocol Namespaces

This specification defines the following XML namespace:

- 'urn:xmpp:order-by:0'

10.2 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.

⁷The 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/>.

11 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:order-by:0'
  xmlns='urn:xmpp:order-by:0'
  elementFormDefault='qualified'>
  <xs:element name='order' maxOccurs='unbounded'>
    <xs:complexType>
      <xs:attribute name='by' type='xs:string' use='required' />
    </xs:complexType>
  </xs:element>
</xs:schema>
```