XEP-0057: Extended Roster

Alexey Shchepin
mailto:alexey@sevcom.net
xmpp:aleksey@jabber.ru

2003-04-28
Version 0.2

Status	Type	Short Name
Retracted	Standards Track	None

This document defines a way to handle extended roster items.

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 What we need to store **1**

3 Where to store **1**

4 How to store **1**

4.1 General information 1

4.2 Server-side information 2

4.3 Client-side information 2

4.3.1 Generic JIDs 3

4.3.2 Client JIDs 3

4.3.3 Conference JIDs 3

4.4 Complete example 3

1 Introduction

The main purpose of this document is to make roster not only a ”contact list”, but also a ”list of useful items”. This means that the user has the ability to store not only users' JIDs, but any JID that he wants to quickly access with more information than just the name, subscription and roster groups.

2 What we need to store

All information, that can be stored for each item can be divided into three categories:

1. Information necessary only for server-side. E.g. we can have a server module that can process ”visibility” to each user in roster (by sending custom presence to this user automatically, without special client-side support of this).
2. Information, neccessary only for client-side. E.g. user description of this item.
3. Information for both sides. This is at least the JID, category and type.

3 Where to store

Using jabber:iq:private as in Bookmark Storage (XEP-0048) [1] for storing this information has one big problem: it is hard to mantain roster data in two separate places. When a client is online, then the client application can handle jabber:iq:roster changes and make similar changes in private storage, but when the user is online with a few different resources, or when he is offline, then making the information consistent is very hard task (a roster can be changed when user offline, e.g. if someone is making an account transfer).
But we have a place where this problem does not exist: jabber:iq:roster. We can store it in <item/> subtags. Existing server implementation MUST NOT remove <x/> tags from it. In this case all information always relates to its JID and disappears when this JID removed.

4 How to store

4.1 General information

JID, category and type are stored as attributes of <item/> tag. Categories and types are the same as in disco. Official categories and types associated with disco are administered by the

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

XMPP Registrar [2] see <https://xmpp.org/registrar/disco-categories.html>.

```
<item jid="jdev@conference.jabber.org"
      category="conference"
      type="text">
```

4.2 Server-side information

This information is implementation-dependent, so to provide flexibility for it, the jabber:x:data namespace defined in Data Forms (XEP-0004) [3] must be used. The client can set these parameters by setting this item with this form with type='submit'.

```
<item jid="romeo@montague.net"
      name="Romeo"
      subscription="both">
  <x xmlns='jabber:x:data' type='form'>
    <field type='list-single' label='Visibility' var='visibility'>
      <value>visible</value>
      <option label='Normal'>
        <value>normal</value>
      </option>
      <option label='Always visible'>
        <value>visible</value>
      </option>
      <option label='Always invisible'>
        <value>invisible</value>
      </option>
    </field>
  </x>
  ...
</item>
```

4.3 Client-side information

This information stored in <x/> tag with namespace jabber:x:roster:item. Following subtags can be used for diferent types of JIDs, however client applications can make this set bigger, to implement more functions.

[2] The 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/>.

[3] XEP-0004: Data Forms <https://xmpp.org/extensions/xep-0004.html>.

4.3.1 Generic JIDs

For all categories and types of JID allowed following subtag:

- **desc**: A description or note describing the JID.

4.3.2 Client JIDs

For all JIDs with category=client allowed following subtags:

- **always-visible**, **always-invisible**: The client should send custom presence to this JID to be always visible or invisible to it.

4.3.3 Conference JIDs

For all JIDs with category=conference allowed following subtags:

- **nick**: The nickname to be used when joining the room. If few such tags in one item, then first is used by default, and others used if first not available.
- **password**: The password to be used to joing the room.
- **auto-join**: The client should automatically join this room once the user has been authenticated, and the roster have been fetched.

```
<x xmlns="jabber:x:roster:item">
  <always-visible/>
  <desc>My old good friend</desc>
</x>
```

4.4 Complete example

```
<iq id="roster_1" type="result">
  <query xmlns="jabber:iq:roster">
    <item jid="romeo@montague.net"
        name="Romeo"
        category="user"
        type="client"
        subscription="both">
      <x xmlns="jabber:x:roster:item">
        <always-visible/>
        <desc>bla bla bla</desc>
      </x>
      <x xmlns='jabber:x:data' type='form'>
```

```
        <field type='list-single' label='Visibility' var='visibility'>
          <value>visible</value>
          <option label='Normal'>
            <value>normal</value>
          </option>
          <option label='Always visible'>
            <value>visible</value>
          </option>
          <option label='Always invisible'>
            <value>invisible</value>
          </option>
        </field>
      </x>
    </item>
    <item jid="jdev@conference.jabber.org"
        name="Developers Zone"
          category="conference"
        type="text"
        subscription="none">
      <x xmlns="jabber:x:roster:item">
        <password>bigsecret</password>
        <auto-join/>
        <nick>aleksey</nick>
        <nick>alexey</nick>
        <desc>Jabber developers talks</desc>
      </x>
    </item>
  </query>
</iq>
```