XEP-0162: Best Practices for Roster and Subscription Management

Lucas Nussbaum
mailto:lucas@lucas-nussbaum.net
xmpp:lucas@nussbaum.fr

2020-06-25
Version 0.2.1

Status	Type	Short Name
Deferred	Informational	N/A

This document specifies best practices for roster and subscription management in Jabber/XMPP clients.

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**
1.1 Motivation 1
1.2 Overview of subscription states 1

2 Removal of Contacts **1**

3 Display of Contacts in the Roster **2**
3.1 Name of the Hidden group 3

1 Introduction

1.1 Motivation

RFC 3921 [1] explains how subscriptions and rosters integrate. However, several points are left to the client author's discretion, and this can lead to some confusion among client developers. This document specifies best practices that enable all Jabber clients to manage subscriptions and roster in a coherent way, thus making sure that such clients will not surprise end users with unexpected behavior.

1.2 Overview of subscription states

As defined in RFC 3921 (sections 6, 7, 8), a roster item can have one of the following subscription states:

- *subscription='none'*: you aren't interested in the item's presence, and neither is the item interested in yours.
- *subscription='from'*: the item is interested in your presence information, but you don't care about the contact. (You must be somebody important! ;)
- *subscription='to'*: You are interested in the item's presence information, but the contact is not interested in yours.
- *subscription='both'*: You and the contact are interested in each other's presence information.

Additionally, if *subscription='from'* or *subscription='none'*, you can have *ask='subscribe'*, which means you sent a subscription request to the item but haven't received an answer yet.

2 Removal of Contacts

It is necessary to distinguish two things:

- Removal from the user's roster (does the user's roster include an entry for this JID?) ;
- Removal of presence subscription (cancellation of subscription).

Three problems have to be taken into consideration here:

[1] RFC 3921: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc3921>.

- Removing an item from your roster (i.e., performing a roster set with *subscription='remove'*) cancels all subscriptions (both from the item and to the item).
- The fact that a user is no longer interested in the contact's presence doesn't mean that the contact is no longer interested in the user's presence.
- This is complex, and should be presented in a simple way to the user.

Therefore, the following behavior is RECOMMENDED for Jabber clients:

1. When the user asks for the removal of an item with *subscription='both'*, the Jabber client SHOULD ask the user whether he wants to revoke the contact's authorization to see the user's presence too. This action SHOULD be called "Remove" since this is the convention used by other IM systems.

2. The Jabber client SHOULD NOT remove the contact from the roster. There are two exceptions:

   a) The Jabber client MAY remove the contact from the roster if the user explicitely asked (so the user has to be informed he might remove both presence subscriptions).

   b) The jabber client MAY transparently remove the contact from the roster if the user asked to, and if *subscription='none'* or *subscription='to'*.

In addition to the "Remove" action described above, the client MAY provide a way to revoke the contact's subscription to the user's presence. This action, if provided, SHOULD be called "Block" since this is the convention used by other IM networks.

3 Display of Contacts in the Roster

Defining a sensible roster policy is difficult, because many users have different, special needs. We have to ensure that:

- Most contacts that everybody wants to hide are hidden.
- No contact that someone wants to see is hidden.

The user's roster contains both contacts of interest for the user (contacts with *subscription='both'* or *subscription='to'*) and contacts which are only interested in receiving the user's presence information. Also, the user might be interested in having some contacts even with

subscription='none', because he often sends messages to them but isn't interested in their presence.
Therefore, the following types of contacts SHOULD be displayed by clients:

- items with *subscription='both'* or *subscription='to'* ;
- items with *subscription='none'* or *subscription='from'* and *ask='subscribe'*. It is ((subscription='none' or subscription='from') and ask='subscribe') ;
- items with *subscription='none'* or *subscription='from'* which have a 'name' attribute or a 'group' child set. It is ((subscription='none' or subscription='from') and (name attribute or group child)).

The client MAY display contacts with subscription='from' which don't match one of the above cases in an additional 'Observers' group. If no 'Observers' group is used, the client SHOULD NOT display contacts which don't match one of the above rules.
Some users give nicknames to contacts they do not want to see, to be able to remember who they are when cleaning the roster. An additional 'Hidden' group MAY be used to handle such contacts. The client SHOULD NOT display contacts if one of their groups is the 'Hidden' group. (This mean that if contact 'C' is in groups 'G' and 'Hidden', 'C' should never be displayed, even in group 'G'.)

3.1 Name of the Hidden group

The name of the 'Hidden' group can be managed in two different ways :

- It could always be named 'Hidden' but be displayed translated by the client ;
- It could be named as the user wants. Its name could then be stored using jabber:iq:private, like the delimiter for nested groups in Nested Roster Groups (XEP-0083) [2].

This is left as an open issue since no clients (to the author's knowledge) implement this 'Hidden' group. But the preference should go to the first solution, which avoids relying on Private XML Storage (XEP-0049) [3].

[2] XEP-0083: Nested Roster Groups <https://xmpp.org/extensions/xep-0083.html>.
[3] XEP-0049: Private XML Storage <https://xmpp.org/extensions/xep-0049.html>.