XEP-0100: Gateway Interaction

Peter Saint-Andre
mailto:peter@andyet.net
xmpp:stpeter@stpeter.im
https://stpeter.im/

Dave Smith
mailto:dizzyd@jabber.org
xmpp:dizzyd@jabber.org

Status	Type	Short Name
Active	Informational	gateway

This document specifies best practices for interactions between Jabber clients and client proxy gateways to legacy IM services.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 - 2014 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 <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

Contents

1 Introduction

One distinguishing characteristic of Jabber technologies from their earliest days has been the existence of gateways (also called "transports") between the Jabber network and legacy instant messaging services such as AOL Instant Messenger (AIM), ICQ, Windows Live Messenger, and Yahoo! Messenger. Surprisingly, the recommended behavior of such gateways, including the protocol elements used by a client to interact with a gateway, has never been fully documented. This document attempts to fill that void by codifying best practices for gateway interaction.

Note well that this document defines protocol usage with regard to client proxy gateways, i.e., gateways that "masquerade" as a client on a non-Jabber IM service. Gateways that perform direct protocol translation without proxying for an account on a non-Jabber service are not addressed in this document. Furthermore, this document does not define any interaction between a gateway and the non-Jabber service, only interactions between a Jabber client and the gateway. Although what happens on the other side of the gateway is highly dependent on the nature of the legacy service, gateways should at least provide a common interface on the Jabber side of the gateway so that Jabber clients can be written in a consistent fashion.

2 Glossary

Gateway A service on the Jabber network that translates between the Jabber/XMPP protocols and the protocol used by a Legacy Service; in the context of this document, by "gateway" we mean a "client proxy service" that acts as a client with regard to a Legacy Service and thereby "masquerades" as a user on such a service.

Jabber User A human user who has registered an account with a Jabber server; a Jabber User who wants to use a Gateway must first have also registered an account with a Legacy Service.

Legacy Service A non-XMPP instant messaging service.

Legacy User A human user who has registered an account with a Legacy Service.

Server An instant messaging server as defined in RFC 6121.

3 Requirements

The requirements defined by this document are captured in two sets of use cases: one set from the perspective of the Jabber User, and a smaller set from the perspective of the Legacy User who wants to interact with the Jabber User.

The Jabber User use cases are:

1. Register

2. Edit Registration

3. Unregister

4. Log In

5. Log Out

6. Add Contact

7. Delete Contact

8. Send Message

The Legacy User use cases are:

1. Add Contact

2. Delete Contact

3. Send Message

While more advanced use cases (e.g., sending files and joining chat rooms) are of inherent interest, they are not covered in this document because registration, contact list management, and message exchange define the baseline functionality included in all gateway implementations; future specifications may address the more advanced use cases.

4 Jabber User Use Cases

4.1 Register

All existing client proxy gateways require a Jabber User to register with the Gateway before sending messages or presence through the gateway. Although strictly speaking registration is not required (e.g., a Gateway could prompt the Jabber User for credentials every time the user attempted to communicate through the gateway, or once per "session"), in practice this step is required.

4.1.1 Primary Flow

1. Jabber User sends IQ-get qualified by the Service Discovery (XEP-0030) [1] information namespace to the Gateway, and/or IQ-get qualified by the Agent Information (XEP-0094) [2] namespace to the Gateway's parent (the latter method is deprecated but still in use).

---

[1]XEP-0030: Service Discovery <http://xmpp.org/extensions/xep-0030.html>.
[2]XEP-0094: Agent Information <http://xmpp.org/extensions/xep-0094.html>.

Listing 1: User Queries Gateway Regarding Service Discovery Identity

```
<iq type='get'
    from='romeo@montague.lit/orchard'
    to='aim.shakespeare.lit'
    id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 2: User Queries Gateway's Parent Regarding Agent Information

```
<iq type='get'
    from='romeo@montague.lit/orchard'
    to='shakespeare.lit'
    id='agent1'>
  <query xmlns='jabber:iq:agents'/>
</iq>
```

Note: Although many existing gateway implementations support only the older Agent Information protocol, it is RECOMMENDED that gateways support the Service Discovery protocol, since the former protocol is deprecated in favor of the latter. Until existing gateways are upgraded, clients SHOULD support both.

2. Gateway and/or parent returns identity information to Jabber User's Client.

Listing 3: Gateway Returns Service Discovery Identity

```
<iq type='result'
    from='aim.shakespeare.lit'
    to='romeo@montague.lit/orchard'
    id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <identity category='gateway'
              type='aim'
              name='AIM␣Gateway'/>
    <feature var='http://jabber.org/protocol/disco#info'/>
    <feature var='jabber:iq:register'/>
    <feature var='jabber:iq:time'/>
    <feature var='jabber:iq:version'/>
  </query>
</iq>
```

Listing 4: Gateway's Parent Returns Agent Information

```
<iq type='result'
    from='romeo@montague.lit/orchard'
    to='shakespeare.lit'
    id='agent1'>
```

```
  <query xmlns='jabber:iq:agents'>
    <agent jid='aim.shakespeare.lit'>
      <name>AIM Gateway</name>
      <service>aim</service>
      <transport/>
      <register/>
    </agent>
  </query>
</iq>
```

Note: Given the foregoing, a client can determine the identity of the gateway, specifically (1) that it is a gateway and (2) to which legacy service it provides a gateway.

3. Jabber User sends IQ-get qualified by the In-Band Registration (XEP-0077) [3] (jabber:iq:register) namespace to Gateway.

Listing 5: User Queries Gateway Regarding Registration Requirements

```
<iq type='get'
    from='romeo@montague.lit/orchard'
    to='aim.shakespeare.lit'
    id='reg1'>
  <query xmlns='jabber:iq:register'/>
</iq>
```

4. Gateway returns IQ-result to Jabber User, specifying information that is required in order to register.

Listing 6: Gateway Returns Registration Requirements

```
<iq type='result'
    from='aim.shakespeare.lit'
    to='romeo@montague.lit/orchard'
    id='reg1'>
  <query xmlns='jabber:iq:register'>
    <instructions>
       Please provide your AIM screen name and password.
    </instructions>
    <username/>
    <password/>
  </query>
</iq>
```

---

[3]XEP-0077: In-Band Registration <http://xmpp.org/extensions/xep-0077.html>.

5. Jabber User sends IQ-set qualified by the 'jabber:iq:register' namespace to Gateway, containing information required to register.

Listing 7: User Provides Registration Information

```
<iq type='set'
    from='romeo@montague.lit/orchard'
    to='aim.shakespeare.lit'
    id='reg2'>
  <query xmlns='jabber:iq:register'>
    <username>RomeoMyRomeo</username>
    <password>ILoveJuliet</password>
  </query>
</iq>
```

Note: The XML character data of the <username/> element SHOULD be the Jabber User's LegacyUserAddress as described under Addressing, such as an AOL screen name, ICQ number, Windows Live Messenger (formerly MSN Messenger) address, or Yahoo! ID.

6. Gateway verifies that registration information provided by Jabber User is valid (using whatever means appropriate for the Legacy Service) and informs Jabber User of success [A1].

Listing 8: Gateway Informs Jabber User of Success

```
<iq type='result'
    from='aim.shakespeare.lit'
    to='romeo@montague.lit/orchard'
    id='reg2'/>
```

7. If Gateway logged into Legacy Service in preceding step, Gateway buffers any translatable events (e.g., messages and presence) queued up for Jabber User on Legacy Service.

8. Optionally, Jabber User sends IQ-set qualified by the 'jabber:iq:roster' namespace to its server (see XMPP Core [4]), containing a roster item for Gateway.

Listing 9: User Creates Roster Entry

```
<iq type='set'
    from='romeo@montague.lit/orchard'
    id='roster1'>
  <query xmlns='jabber:iq:roster'>
    <item jid='aim.shakespeare.lit' name='AIM_Gateway'/>
```

---

[4]RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

```
    </query>
</iq>
```

Listing 10: Server Response

```
<iq type='result'
    to='romeo@montague.lit/orchard'
    id='roster1'/>
```

9. Gateway sends subscription request to Jabber User (i.e., by sending a presence stanza of type "subscribe" to Jabber User's bare JID).

Listing 11: Gateway Subscribes to User's Presence

```
<presence type='subscribe'
          from='aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

10. Jabber User's client SHOULD approve the subscription request (i.e., by sending a presence stanza of type "subscribed" to Gateway).

Listing 12: Jabber User Approves Subscription Request

```
<presence type='subscribed'
          from='romeo@montague.lit'
          to='aim.shakespeare.lit'/>
```

Note: As specified in RFC 6121, Jabber User's server will generate a "roster push" at this point if client did not previously perform a roster set to add Gateway to user's roster (as mentioned above).

11. Jabber User sends subscription request to Gateway (i.e., by sending a presence stanza of type "subscribe" to Gateway).

Listing 13: Jabber User Subscribes to Gateway's Presence

```
<presence type='subscribe'
          from='romeo@montague.lit'
          to='aim.shakespeare.lit'/>
```

12. Gateway sends approves subscription request (i.e., by sending a presence stanza of type "subscribed" to Jabber User's bare JID).

Listing 14: Gateway Approves Subscription Request

```
<presence type='subscribed'
          from='aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

13. Execute "Log In" use case.

14. Gateway sends any buffered messages to Jabber User.

15. Use Case Ends.

4.1.2 Alternate Flows

1. User information not verified:

   a) Gateway returns <not-acceptable/> error to Jabber User. (For detailed information regarding error conditions, refer to Error Condition Mappings (XEP-0086) [5].)

   Listing 15: Gateway Informs Jabber User of Registration Error

```
<iq type='error'
    from='aim.shakespeare.lit'
    to='romeo@montague.lit/orchard'
    id='reg2'>
  <query xmlns='jabber:iq:register'>
    <username>RomeoMyRomeo</username>
    <password>ILoveJuliet</password>
  </query>
  <error code='406' type='modify'>
    <not-acceptable
        xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

   b) Use Case Ends unsuccessfully.

4.2 Edit Registration

After a Jabber User has registered with a Gateway, the user may wish to modify his or her existing registration information (e.g., because the user has changed his or her password on the legacy IM service).

---

[5]XEP-0086: Error Condition Mappings <http://xmpp.org/extensions/xep-0086.html>.

4.2.1 Primary Flow

1. Jabber User sends IQ-get qualified by the 'jabber:iq:register' namespace to Gateway.

Listing 16: User Queries Gateway Regarding Registration Requirements

```
<iq type='get'
    from='romeo@montague.lit/orchard'
    to='aim.shakespeare.lit'
    id='edit1'>
  <query xmlns='jabber:iq:register'/>
</iq>
```

2. Gateway returns IQ-result to Jabber User, specifying registration information on record and including empty <registered/> element to signify that user is already registered. [6]

Listing 17: Gateway Returns Registration Information of Record

```
<iq type='result'
    from='aim.shakespeare.lit'
    to='romeo@montague.lit/orchard'
    id='edit1'>
  <query xmlns='jabber:iq:register'>
    <registered/>
    <username>RomeoMyRomeo</username>
    <password>ILoveJuliet</password>
  </query>
</iq>
```

3. Jabber User sends IQ-set qualified by the 'jabber:iq:register' namespace to Gateway, containing all information (i.e., not just the "delta").

Listing 18: User Provides Registration Information

```
<iq type='set'
    from='romeo@montague.lit/orchard'
    to='aim.shakespeare.lit'
    id='edit2'>
  <query xmlns='jabber:iq:register'>
    <username>RomeoMyRomeo</username>
    <password>B4lc0ny</password>
  </query>
</iq>
```

---

[6]The fact that the Gateway can determine the Jabber User's legacy username based on the JID of the 'from' address indicates that the client proxy model assumes one registration per Jabber User.

4. Gateway verifies that, if changed, information provided by Jabber User is still valid (using whatever means appropriate for the Legacy Service) and informs Jabber User of success [A1].

Listing 19: Gateway Informs Jabber User of Success

```
<iq type='result'
    from='aim.shakespeare.lit'
    to='romeo@montague.lit/orchard'
    id='edit2'/>
```

4.2.2 Alternate Flows

1. Edit unsuccessful:

   a) Gateway returns <not-acceptable/> error to Jabber User.

   Listing 20: Gateway Informs Jabber User of Registration Error

```
<iq type='error'
    from='aim.shakespeare.lit'
    to='romeo@montague.lit/orchard'
    id='edit2'>
  <query xmlns='jabber:iq:register'>
    <username>RomeoMyRomeo</username>
    <password>B4lc0ny</password>
  </query>
  <error code='406' type='modify'>
    <not-acceptable
        xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

   b) Use Case Ends unsuccessfully.

4.3 Unregister

After a Jabber User has registered with a Gateway, the user may choose to unregister with the Gateway, effectively ending his or her relationship with the Gateway (e.g., the user will no longer be allowed to communicate through the gateway with legacy users).

4.3.1 Primary Flow

1. Jabber User sends IQ-set in 'jabber:iq:register' namespace to Gateway, containing empty <remove/> element.

Listing 21: User Unregisters

```
<iq type='set'
    from='romeo@montague.lit/orchard'
    to='aim.shakespeare.lit'
    id='unreg1'>
  <query xmlns='jabber:iq:register'>
    <remove/>
  </query>
</iq>
```

2. Gateway sends unavailable presence from Jabber User to Legacy Users and logs Jabber User out of Legacy Service.

3. Gateway deletes Jabber User's information.

4. Gateway sends IQ-result to Jabber User.

Listing 22: Gateway Informs Jabber User of Success

```
<iq type='result'
    from='aim.shakespeare.lit'
    to='romeo@montague.lit/orchard'
    id='unreg1'/>
```

5. Gateway cancels subscriptions.

Listing 23: Gateway Cancels Subscriptions

```
<presence type='unsubscribe'
          from='aim.shakespeare.lit'
          to='romeo@montague.lit'/>

<presence type='unsubscribed'
          from='aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

6. Gateway sends unavailable presence to Jabber User.

Listing 24: Gateway Logs User Out

```
<presence type='unavailable'
          from='aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

7. Jabber User's client SHOULD delete from the user's roster (1) the gateway itself, and (2) all legacy Contacts associated with the gateway.

8. Use Case Ends.

4.3.2   Alternate Flows

None.

4.4   Log In

After a Jabber User has registered with a Gateway, the Jabber User may subsequently log in to the Gateway, effectively creating a "session" with the Gateway and enabling the Gateway to log into the Legacy Service on behalf of the user by sending the user's legacy credentials to the Legacy Service.

4.4.1   Primary Flow

1. Jabber User sends available presence broadcast to Server or sends directed presence to Gateway or a Legacy User.

Listing 25: Jabber User Sends Available Presence

```
<presence/>
```

Listing 26: Jabber User's Server Broadcasts Available Presence

```
<presence from='romeo@montague.lit/orchard'
          to='juliet@aim.shakespeare.lit'/>
<presence from='romeo@montague.lit/orchard'
          to='aim.shakespeare.lit'/>
...
```

2. Upon receiving the first presence notification stanza from Jabber User to Gateway or Legacy User, Gateway logs Jabber User into Legacy Service [A1].

3. Gateway sends presence stanza to Jabber User expressing availability.

Listing 27: Gateway Sends Presence to Jabber User

```
<presence from='aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

4. Optionally, Gateway handles Legacy Service contact list; see the Contact Lists section of this document.

5. Gateway forwards current presence information from Legacy Users to Jabber User, if possible mapping availability status (e.g., "away").

Listing 28: Gateway Sends Presence from Legacy Users to Jabber User

```
<presence from='juliet@aim.shakespeare.lit'
          to='romeo@montague.lit'>
  <show>away</show>
</presence>
```

Note: If the Legacy Service to which the Gateway connects does not support the concept of "resources", the 'from' address of presence notification stanzas generated by a gateway SHOULD NOT include a resource identifier (i.e., they SHOULD be of the form <user@host> rather than <user@host/resource>). However, the 'from' address MAY include a resource if the Gateway determines that this is appropriate in the context of its communications with the Legacy Service.

6. Gateway forwards all subsequent presence stanzas to Legacy Users (except those of type "probe" and those addressed to the Gateway itself).

Listing 29: Jabber User Modifies Presence

```
<presence from='romeo@montague.lit/orchard'
          to='juliet@aim.shakespeare.lit'>
  <show>dnd</show>
  <status>Wooing Juliet</status>
</presence>
```

7. Use Case Ends.

4.4.2 Alternate Flows

1. Login fails:

   a) Gateway sends appropriate presence error to Jabber User (<not-authorized/> if password is bad, <remote-server-timeout/> if Legacy Service is down, etc.).

Listing 30: Gateway Informs Jabber User of Failed Login

```
<presence to='aim.shakespeare.lit'
          from='romeo@shakespeare.lit'
          type='error'>
  <error code='504' type='wait'>
    <remote-server-timeout
        xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</presence>
```

   b) Use Case Ends unsuccessfully.

4.5 Log Out

At any time after logging in to the Gateway, the Jabber User may log out of the Gateway and thereby end his or her session on the Legacy Service. This may happen automatically when the Jabber User terminates his or her session with a Jabber server, or independently of any session on the Jabber network by manually logging out of the Gateway.

4.5.1 Primary Flow

1. Jabber User sends unavailable presence broadcast to Server or sends directed presence stanza of type "unavailable" to Gateway or (if Gateway does not support directed presence) Legacy User.

Listing 31: Jabber User Sends Unavailable Presence

```
<presence type='unavailable'/>
```

Listing 32: Jabber User's Server Broadcasts Unavailable Presence

```
<presence type='unavailable'
          from='romeo@montague.lit/orchard'
          to='aim.shakespeare.lit'/>
```

2. Gateway transforms unavailable presence stanzas received from the Jabber User's server and routes them to all of the Jabber User's contacts on Legacy Service.

3. Gateway logs Jabber User out of Legacy Service [A1].

4. Gateway sends presence stanza of type "unavailable" to Jabber User.

Listing 33: Gateway Logs User Out

```
<presence type='unavailable'
          from='aim.shakespeare.lit'
          to='romeo@montague.lit/orchard'/>
```

5. Use Case Ends.

4.5.2 Alternate Flows

1. Legacy Service supports directed presence and Gateway receives presence stanza of type "unavailable" directed to a Legacy User:

   a) Gateway passes through directed unavailable presence to Legacy User.

Listing 34: Jabber User Becomes Unavailable

```
<presence type='unavailable'
          from='romeo@montague.lit/orchard'
          to='juliet@aim.shakespeare.lit'/>
```

   b) Use Case Ends.

4.6 Add Contact

After registering with the Gateway, the Jabber User may want to add Legacy Users to his or her Jabber roster.

4.6.1 Primary Flow

1. Jabber User sends presence stanza of type "subscribe" to Legacy User.

14

Listing 35: Jabber User Sends Subscription Request to Legacy User

```
<presence type='subscribe'
          from='romeo@montague.lit'
          to='CapuletNurse@aim.shakespeare.lit'/>
```

Note: As specified in RFC 6121, sending this packet will result in a "roster push" from the Server to all of the Jabber User's available resources.

2. Gateway transforms subscription request and routes it to Legacy User.

3. If Legacy User approves subscription request, Gateway sends presence stanza of type "subscribed" to Jabber User on behalf of Legacy User. [A1]

Listing 36: Gateway Approves Subscription Request on Behalf of Legacy User

```
<presence type='subscribed'
          from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

4. Gateway sends available presence stanza to Jabber User on behalf of Legacy User.

Listing 37: Gateway Sends Legacy User's Current Presence Information to Jabber User

```
<presence from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit/orchard'/>
```

5. Gateway sends presence stanza of type "subscribe" to Jabber User on behalf of Legacy User.

Listing 38: Gateway Sends Subscription Request to Jabber User on Behalf of Legacy User

```
<presence type='subscribe'
          from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

6. Jabber User sends presence stanza of type "subscribed" to Legacy User.

Listing 39: Jabber User Approves Subscription Request

```
<presence type='subscribed'
          from='romeo@montague.lit'
          to='CapuletNurse@aim.shakespeare.lit'/>
```

7. Use Case Ends.

4.6.2 Alternate Flows

1. Legacy User denies subscription request:

    a) Gateway transforms subscription denial and routes it to Jabber User.

    Listing 40: Legacy User Denies Subscription Request

    ```
    <presence type='unsubscribed'
              from='juliet@aim.shakespeare.lit'
              to='romeo@montague.lit'/>
    ```

    b) Use Case Ends unsuccessfully.

4.7 Delete Contact

After adding a Legacy User to his or her Jabber roster, the Jabber User may want to delete that contact.

4.7.1 Primary Flow

1. Jabber User sends IQ-set qualified by the 'jabber:iq:roster' namespace, containing subscription attribute with value of "remove".

    Listing 41: User Removes Roster Entry for Legacy User

    ```
    <iq type='set'
        from='romeo@montague.lit/orchard'
        id='remove1'>
      <query xmlns='jabber:iq:roster'>
        <item jid='CapuletNurse@aim.shakespeare.lit'
              subscription='remove'/>
      </query>
    </iq>
    ```

2. Server sends normal "roster push" to Jabber User (see RFC 6121) and sends presence stanzas of type "unsubscribe", "unsubscribed", and "unavailable" to Legacy User.

    Listing 42: Server Sends Presence Changes to Legacy User

    ```
    <presence type='unsubscribe'
              from='romeo@montague.lit'
              to='CapuletNurse@aim.shakespeare.lit'/>
    ```

```
<presence type='unsubscribed'
          from='romeo@montague.lit'
          to='CapuletNurse@aim.shakespeare.lit'/>

<presence type='unavailable'
          from='romeo@montague.lit/orchard'
          to='CapuletNurse@aim.shakespeare.lit'/>
```

3. Gateway cleans up subscription state, informs Legacy User that Jabber User is unavailable, and MUST NOT send future changes in Jabber User's presence to Legacy User.

4. Use Case Ends.

4.7.2 Alternate Flows

None.

4.8 Send Message

Naturally, the Jabber User may want to exchange messages with a Legacy User. For the purposes of this document, we discuss one-to-one messaging only (i.e., groupchat messages, such as those defined in Multi-User Chat (XEP-0045) [7], are out of scope).

4.8.1 Primary Flow

1. Jabber User sends message stanza to Legacy User.

Listing 43: Jabber User Sends Message to Legacy User

```
<message from='romeo@montague.lit/orchard'
         to='juliet@aim.shakespeare.lit'
         type='chat'>
  <body>Neither, fair saint, if either thee dislike.</body>
</message>
```

2. Gateway transforms message to legacy protocol and sends to Legacy User [A1].

3. Use Case Ends.

---

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

4.8.2 Alternate Flows

1. Legacy Service reports error.

2. Gateway sends appropriate error to Jabber User:

    • <item-not-found/> -- Legacy User address is not valid.

    • <registration-required/> -- Jabber User is not registered with Gateway.

    • <service-unavailable/> -- Legacy User is offline and Legacy Service (or Gateway) does not provide offline message storage.

    • <remote-server-timeout/> -- Legacy Service cannot be reached.

3. Use Case Ends unsuccessfully.

5 Legacy User Use Cases

5.1 Add Contact

The Legacy User may want to add the Jabber User to his or her contact list on the Legacy Service. Because the Jabber User has an account on the Legacy Service by definition, the Legacy User will actually add the Jabber User's legacy address to his or her contact list, not the Jabber User's address on the Jabber/XMPP network.

5.1.1 Primary Flow

1. Legacy User requests subscription to Jabber User's legacy address (using legacy protocol).

2. Gateway sends presence stanza of type "subscribe" to Jabber User on behalf of Legacy User. (Note: Gateway MUST NOT send presence stanza of type "subscribed".)

Listing 44: Gateway Sends Subscription Request on Behalf of Legacy User

```
<presence type='subscribe'
          from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

3. Jabber User approves subscription request by sending presence stanza of type "subscribed" to Legacy User [A1].

Listing 45: Jabber User Approves Subscription Request

```
<presence type='subscribed'
          from='romeo@montague.lit'
          to='CapuletNurse@aim.shakespeare.lit'/>
```

4. Gateway sends Jabber User's presence information to Legacy User.

5. Jabber User's Client sends presence stanza of type "subscribe" to Legacy User.

Listing 46: Jabber User Sends Subscription Request to Legacy User

```
<presence type='subscribe'
          from='romeo@montague.lit'
          to='CapuletNurse@aim.shakespeare.lit'/>
```

6. Gateway sends presence stanza of type "subscribed" to Jabber User on behalf of Legacy User.

Listing 47: Gateway Approves Subscription Request on Behalf of Legacy User

```
<presence type='subscribed'
          from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit'/>
```

7. Gateway sends Legacy User's presence information to Jabber User.

Listing 48: Gateway Sends Legacy User's Current Presence Information to Jabber User

```
<presence from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit/orchard'/>
```

8. Use Case Ends.

5.1.2 Alternate Flows

1. Jabber User denies subscription request:

   a) Jabber User sends presence stanza of type "unsubscribed" to Legacy User.

Listing 49: Jabber User Denies Subscription Request

```
<presence type='unsubscribed'
          from='romeo@montague.lit'
          to='CapuletNurse@aim.shakespeare.lit'/>
```

b) Gateway cleans up subscription state and MUST NOT send Jabber User's presence to Legacy User.

c) Use Case Ends unsuccessfully.

5.2 Delete Contact

After adding the Jabber User to his or her legacy contact list, the Legacy User may want to delete the Jabber User.

5.2.1 Primary Flow

1. Legacy User deletes Jabber User's legacy address (using legacy protocol).

2. Gateway sends presence stanzas of type "unsubscribe", "unsubscribed", and "unavailable" to Jabber User on behalf of Legacy User.

Listing 50: Gateway Cleans Up Subscription on Behalf of Legacy User

```
<presence type='unsubscribe'
          from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit'/>

<presence type='unsubscribed'
          from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit'/>

<presence type='unavailable'
          from='CapuletNurse@aim.shakespeare.lit'
          to='romeo@montague.lit/orchard'/>
```

3. Jabber User's server performs defined functionality for handling presence stanzas of type "unsubscribe" and "unsubscribed" (see RFC 6121).

4. Use Case Ends.

5.2.2 Alternate Flows

None.

5.3 Send Message

Naturally, the Legacy User may want to exchange messages with the Jabber User. (Here again, groupchat messages are out of scope.)

5.3.1 Primary Flow

1. Legacy User sends message to Jabber User using legacy protocol.

2. Gateway transforms message and routes to Jabber User.

Listing 51: Legacy User Sends Message to Jabber User

```
<message from='juliet@aim.shakespeare.lit'
         to='romeo@montague.lit'>
  <body>Art thou not Romeo, and a Montague?</body>
</message>
```

Note: If the Legacy Service to which the Gateway connects does not support a concept equivalent to that of Jabber "resources" as described in RFC 6120 [8], the 'from' address of message stanzas generated by a gateway SHOULD NOT include a resource identifier (i.e., they SHOULD be of the form <user@host> rather than <user@host/resource>). However, the 'from' address MAY include a resource if the Gateway determines that this is appropriate in the context of its communications with the Legacy Service.

3. Jabber User's Server delivers message or (optionally) stores it for later retrieval.

4. Use Case Ends.

5.3.2 Alternate Flows

None.

---

[8]RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

6 Addressing

6.1 Gateways

The address of a gateway itself SHOULD be a hostname only, and that hostname SHOULD NOT be supplemented with a resource identifier when referring to the gateway's address (e.g., when storing the gateway in a roster).

6.2 Users

The Jabber Identifier corresponding to a Legacy User's address is typically of the form <LegacyUserAddress@gateway.example.com>, where LegacyUserAddress is the Legacy User's address on the Legacy Service and where gateway.example.com is the Jabber address of the gateway.

Unfortunately, usernames on some Legacy Services may allow characters that are disallowed in Jabber usernames as specified by the Nodeprep profile of stringprep defined in RFC 3920. For example, the usernames for a Legacy Service may be of the form <user@domain>, which would result in an illegal JID such as <user@domain@gateway.example.com>.

There are two possible ways to solve this problem:

1. Use JID Escaping (XEP-0106) [9].

2. Use the older 'jabber:iq:gateway' protocol (as documented in the following section).

Gateways and clients SHOULD implement at least one of these protocols; at a minimum, it is RECOMMENDED for gateways and clients to implement the 'jabber:iq:gateway' protocol.

6.3 The jabber:iq:gateway Protocol

The 'jabber:iq:gateway' protocol performs two functions:

1. It enables a client to determine the text for the "prompt" to show to a Jabber User when the user wants to add a legacy contact to the user's roster (e.g., "Please enter the AOL Screen Name of the person you would like to contact"), as well as the preferred name for the prompted item (e.g., "Screen Name"). To do so, the client sends an empty <query/> element and the gateway returns a <prompt/> element (the name for the prompted item) and optionally a <desc/> element (the text of the prompt itself).

2. It enables a client to send a legacy username to the gateway and receive a properly-formatted JID in return. To do so, the client sends the legacy address to the gateway as

---

[9]XEP-0106: JID Escaping <http://xmpp.org/extensions/xep-0106.html>.

the character data of the <prompt/> element and the gateway returns a valid JID as the character data of the <jid/> element.

Both uses are illustrated below.

Listing 52: Client Requests Prompt

```
<iq type='get' to='aim.jabber.org' from='stpeter@jabber.org/roundabout
    ' id='gate1'>
  <query xmlns='jabber:iq:gateway'/>
</iq>
```

Listing 53: Gateway Returns Prompt

```
<iq type='result' from='aim.jabber.org' to='stpeter@jabber.org/
    roundabout' id='gate1'>
    <query xmlns='jabber:iq:gateway'>
      <desc>
        Please enter the AOL Screen Name of the
        person you would like to contact.
      </desc>
      <prompt>Contact ID</prompt>
    </query>
  </iq>
```

The following table is intended to assist implementors with mapping of gateway identities to English-language prompt names and text.

Legacy Service	Service Discovery Identity	Prompt Name	Prompt Text
AOL Instant Messenger	gateway/aim	Contact ID	Please enter the AOL Screen Name of the person you would like to contact.
ICQ	gateway/icq	Contact ID	Please enter the ICQ Number of the person you would like to contact.

Legacy Service	Service Discovery Identity	Prompt Name	Prompt Text
Windows Live Messenger	gateway/msn The service discovery type was originally defined as "msn" to reflect the name of the service as "MSN Messenger"; this type is retained even though the service has been renamed to "Windows Live Messenger".	Contact ID	Please enter the Windows Live Messenger address of the person you would like to contact.
Yahoo! Messenger	gateway/yahoo	Contact ID	Please enter the Yahoo ID of the person you would like to contact.

If the client provides an 'xml:lang' attribute with the IQ-get, the gateway SHOULD return localized prompt names and text if available, or default to English if not available.

Once the user enters a legacy username or address, the client MUST send it to the gateway as the character data of the <prompt/> element in an IQ-set; the gateway MUST then return a properly-formed JID based on the provided by the client.

Listing 54: Client Provides Legacy Username

```
<iq type='set' to='aim.jabber.org' from='stpeter@jabber.org/roundabout
    ' id='gate1'>
  <query xmlns='jabber:iq:gateway'>
      <prompt>Foo Bar</prompt>
  </query>
</iq>
```

Listing 55: Gateway Returns JID

```
<iq type='result' from='aim.jabber.org' to='stpeter@jabber.org/
    roundabout' id='gate1'>
  <query xmlns='jabber:iq:gateway'>
    <jid>FooBar@aim.jabber.org</jid>
  </query>
</iq>
```

7 Contact Lists

Some legacy services maintain server-side contact lists, which are sent to the gateway when it logs in to the legacy service on behalf of the user. The gateway MAY initiate adding of the legacy contact list items to the user's Jabber roster. Some existing gateways do this by sending a presence stanza of type "subscribed" from the legacy contact's JID (e.g., <LegacyUser@gateway.jabberserver.com>) to the Jabber user; unfortunately, this behavior violates the presence stanza handling rules specified in RFC 6121. Therefore, a gateway SHOULD instead send the legacy contact list items to the Jabber User via the Roster Item Exchange (XEP-0144) [10] protocol.

8 Business Rules

The following business rules apply:

1. A client SHOULD send a Service Discovery request to the gateway (and/or an Agent Information request to the gateway's parent) before requesting registration information.

2. A gateway SHOULD support the Service Discovery protocol.

3. A gateway SHOULD support the Agent Information protocol, although it is deprecated.

4. A gateway SHOULD map, as best it can, the legacy registration fields onto the fields defined for the 'jabber:iq:register' namespace.

5. A gateway SHOULD NOT attempt to emulate offline message storage functionality for legacy services that lack such functionality.

6. Existing gateway implementations do not strictly adhere to the bi-directional nature of Jabber presence notifications, since they do not broadcast presence from the gateway itself to registered users of the gateway, but rather wait for a registered user to send presence to the gateway before sending presence to the user. This sidesteps scalability challenges but may be sub-optimal; while this document does not require existing gateways to change their current behavior, it does RECOMMEND that they broadcast presence notifications to registered users in accordance with the standard Jabber presence model. Specifically:

---

[10]XEP-0144: Roster Item Exchange <http://xmpp.org/extensions/xep-0144.html>.

- On startup, a gateway (1) SHOULD send presence to all registered users of that gateway but (2) MAY wait to receive presence changes from each registered user.

- On shutdown, a gateway SHOULD send unavailable presence to all registered users of the gateway.

9 Security Considerations

As defined herein, a gateway is a *client proxy*, since it "masquerades" as a user on a legacy instant messaging service. In order to act as a client proxy, the gateway logs into the user's account on the legacy service. This implies two things:

- The gateway must gather the legacy credentials from the user, and perhaps store them on the user's behalf.

- The gateway must provide the user's credentials to the legacy service.

There are obvious security concerns with this approach. The concerns include:

1. The user's credentials on the legacy service may be sent in the clear from the gateway to the legacy service if the legacy service does not support channel encryption or strong authentication.

2. When the user informs the gateway of the user's legacy credentials, the credentials may be sent in the clear between the user's Jabber client and the user's Jabber server (if client-to-server channel encryption is not enabled) or between the user's Jabber server and the gateway (if the gateway is not in the user's "home" domain and server-to-server channel encryption is not enabled).

3. If the gateway stores the user's legacy credentials after registration (this is the default behavior of most or all existing gateway implementations), the user's credentials could be acquired by a malicious user if the server hosting the gateway is compromised.

There is no foreseeable solution to these concerns, since they are instrinsic to the client proxy model. Some assurance regarding the second and third concerns can be achieved if the user runs his or her own Jabber server and gateways. However, the only true solution is to move beyond the client proxy model, either by using Jabber for all IM communications or to convince legacy IM services to allow federated server-to-server communications using open protocols such as Jabber/XMPP, thus obviating the need for client proxy gateways entirely.

10 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [11].

11 XMPP Registrar Considerations

11.1 Protocol Namespaces

The XMPP Registrar [12] includes 'jabber:iq:gateway' in its registry of protocol namespaces.

12 XML Schema

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

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='jabber:iq:gateway'
    xmlns='jabber:iq:gateway'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0100: http://www.xmpp.org/extensions/xep-0100.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='query'>
    <xs:complexType>
      <xs:choice>
        <xs:sequence>
          <xs:element name='desc' minOccurs='0' type='xs:string'/>
          <xs:element name='prompt' type='xs:string'/>
        </xs:sequence>
        <xs:element name='jid' type='xs:string'/>
      </xs:choice>
    </xs:complexType>
```

---

[11]The 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/>.

[12]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 <http://xmpp.org/registrar/>.

```
    </xs:element>

</xs:schema>
```