XEP-0133: Service Administration

Peter Saint-Andre
mailto:xsf@stpeter.im
xmpp:peter@jabber.org
http://stpeter.im/

2017-07-15
Version 1.2

Status	Type	Short Name
Active	Informational	admin

This document defines recommended best practices for service-level administration of servers and components using Ad-Hoc Commands.

Legal

Copyright

Permissions

Warranty

Liability

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

There exists a set of common service-level tasks that administrators often need to perform in relation to Jabber/XMPP servers and components. Examples include creating users, disabling accounts, and blacklisting domains for inbound or outbound communications. Because such tasks can be performed with respect to a server or with respect to many kinds of add-on components (e.g., a text conferencing component that conforms to Multi-User Chat (XEP-0045) [1]), it makes sense to define a generic protocol for such interactions. This document describes such a protocol by specifying a profile of Ad-Hoc Commands (XEP-0050) [2] and associated Data Forms (XEP-0004) [3] fields, rather than by defining a specialized and distinct protocol.

2 Requirements

This document addresses the following requirements:

- Enable users with appropriate privileges to perform common administrative tasks with respect to Jabber/XMPP servers and components.
- Re-use existing XMPP and Jabber protocols wherever possible.

3 Discovery

A server or component MUST advertise any administrative commands it supports via Service Discovery (XEP-0030) [4] (as described in XEP-0050: Ad-Hoc Commands); such commands exist as well-defined discovery nodes associated with the service in question.
In order to interact with a particular component attached to a server, an administrator needs to first discover that component and the commands it support, then send the appropriate command to the component itself. A server SHOULD NOT process commands on behalf of associated components, just as it does not handle service discovery requests on behalf of such components.

4 Use Cases

This document defines a profile of XEP-0050: Ad-Hoc Commands that enables a service-level administrator to complete the following use cases:

[1] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[2] XEP-0050: Ad-Hoc Commands <https://xmpp.org/extensions/xep-0050.html>.
[3] XEP-0004: Data Forms <https://xmpp.org/extensions/xep-0004.html>.
[4] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.

1. Add User
2. Delete User
3. Disable User
4. Re-Enable User
5. End User Session
6. Get User Password
7. Change User Password
8. Get User Roster
9. Get User Last Login Time
10. Get User Statistics
11. Edit Blacklist
12. Edit Whitelist
13. Get Number of Registered Users
14. Get Number of Disabled Users
15. Get Number of Online Users
16. Get Number of Active Users
17. Get Number of Idle Users
18. Get List of Registered Users
19. Get List of Disabled Users
20. Get List of Online Users
21. Get List of Active Users
22. Get List of Idle Users
23. Send Announcement to Active Users
24. Set Message of the Day
25. Edit Message of the Day
26. Delete Message of the Day
27. Set Welcome Message

28. Delete Welcome Message
29. Edit Admin List
30. Restart Service
31. Shut Down Service

Naturally, not all of these use cases apply to all service types (e.g., adding a user may not apply to a multi-user chat service). An implementation or deployment MAY support any subset of the use cases defined herein. In addition, although this document aims to define common use cases, an implementation or deployment MAY support additional commands not defined herein, which may or may not be publicly registered.
Note: The text that follows assumes that implementors have read and understood XEP-0050: Ad-Hoc Commands and XEP-0004: Data Forms.

4.1 Add User

A user is defined as any entity that has a persistent relationship with a service (most commonly through the creation a registered account with the service) and whose account is in some sense hosted by the service. Adding a user MUST result in the creation of an account, along with any implementation-specific data for such an account (e.g., database entries or a roster file). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#add-user". A sample protocol flow for this use case is shown below.

Listing 1: Admin Requests to Add a User

```
<iq from='bard@shakespeare.lit/globe'
    id='add-user-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#add-user'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 2: Service Returns Add User Form to Admin

```
<iq from='shakespeare.lit'
    id='add-user-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
```

```
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#add-user'
           sessionid='add-user:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Adding a User</title>
      <instructions>Fill out this form to add a user.</instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣Jabber␣ID␣for␣the␣account␣to␣be␣added'
             type='jid-single'
             var='accountjid'>
        <required/>
      </field>
      <field label='The␣password␣for␣this␣account'
             type='text-private'
             var='password'/>
      <field label='Retype␣password'
             type='text-private'
             var='password-verify'/>
      <field label='Email␣address'
             type='text-single'
             var='email'/>
      <field label='Given␣name'
             type='text-single'
             var='given_name'/>
      <field label='Family␣name'
             type='text-single'
             var='surname'/>
    </x>
  </command>
</iq>
```

Listing 3: Admin Submits Add User Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='add-user-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#add-user'
           sessionid='add-user:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjid'>
        <value>juliet@shakespeare.lit</value>
```

```
      </field>
      <field var='password'>
        <value>R0m30</value>
      </field>
      <field var='password-verify'>
        <value>R0m30</value>
      </field>
      <field var='email'>
        <value>juliet@capulet.com</value>
      </field>
      <field var='given_name'>
        <value>Juliet</value>
      </field>
      <field var='surname'>
        <value>Capulet</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 4: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='add-user-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#add-user'
           sessionid='add-user:20040408T0337Z'
           status='completed'/>
</iq>
```

Notification of completion MAY include the processed data in a data form of type "result".

4.2 Delete User

An administrator may need to permanently delete a user account. Deleting a user SHOULD result in the termination of any active sessions for the user and in the destruction of any implementation-specific data for the account (e.g., database entries or a roster file). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#delete-user". A sample protocol flow for this use case is shown below.

Listing 5: Admin Requests to Delete a User

```
<iq from='bard@shakespeare.lit/globe'
    id='delete-user-1'
```

```
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#delete-user'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 6: Service Returns Delete User Form to Admin

```
<iq from='shakespeare.lit'
    id='delete-user-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#delete-user'
           sessionid='delete-user:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Deleting a User</title>
      <instructions>Fill out this form to delete a user.</instructions
          >
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣Jabber␣ID(s)␣to␣delete'
             type='jid-multi'
             var='accountjids'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Note: If the entity is an end user, the JID SHOULD be of the form <user@host>, not <user@host/resource>.

Listing 7: Admin Submits Delete User Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='delete-user-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
```

```
            node='http://jabber.org/protocol/admin#delete-user'
            sessionid='delete-user:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjids'>
        <value>juliet@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 8: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='delete-user-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#delete-user'
           sessionid='delete-user:20040408T0337Z'
           status='completed'/>
</iq>
```

4.3 Disable User

An administrator may need to temporarily disable a user account. Disabling a user MUST result in the termination of any active sessions for the user and in the prevention of further user logins until the account is re-enabled (this can be thought of as "banning" the user). However, it MUST NOT result in the destruction of any implementation-specific data for the account (e.g., database entries or a roster file). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#disable-user".
A sample protocol flow for this use case is shown below.

Listing 9: Admin Requests to Disable a User

```
<iq from='bard@shakespeare.lit/globe'
    id='disable-user-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#disable-user'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 10: Service Returns Disable User Form to Admin

```
<iq from='shakespeare.lit'
    id='disable-user-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#disable-user'
           sessionid='disable-user:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Disabling a User</title>
      <instructions>Fill out this form to disable a user.</
          instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣Jabber␣ID(s)␣to␣disable'
             type='jid-multi'
             var='accountjids'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Note: If the entity is an end user, the JID SHOULD be of the form <user@host>, not <user@host/resource>.

Listing 11: Admin Submits Disable User Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='disable-user-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#disable-user'
           sessionid='disable-user:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjids'>
        <value>juliet@shakespeare.lit</value>
```

```
      </field>
    </x>
  </command>
</iq>
```

Listing 12: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='disable-user-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#disable-user'
           sessionid='disable-user:20040408T0337Z'
           status='completed'/>
</iq>
```

4.4 Re-Enable User

An administrator may need to re-enable a user account that had been temporarily disabled. Re-enabling a user MUST result in granting the user the ability to access the service again. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#reenable-user".
A sample protocol flow for this use case is shown below.

Listing 13: Admin Requests to Re-Enable a User

```
<iq from='bard@shakespeare.lit/globe'
    id='reenable-user-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#reenable-user'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 14: Service Returns Re-Enable User Form to Admin

```
<iq from='shakespeare.lit'
    id='reenable-user-1'
    to='bard@shakespeare.lit/globe'
    type='result'
```

```
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#reenable-user'
           sessionid='reenable-user:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Re-Enable a User</title>
      <instructions>Fill out this form to re-enable a user.</
          instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣Jabber␣ID(s)␣to␣re-enable'
             type='jid-multi'
             var='accountjids'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Note: If the entity is an end user, the JID SHOULD be of the form <user@host>, not <user@host/resource>.

Listing 15: Admin Submits Re-Enable User Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='reenable-user-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#reenable-user'
           sessionid='reenable-user:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjids'>
        <value>juliet@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 16: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='reenable-user-2'
```

```
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#reenable-user'
           sessionid='reenable-user:20040408T0337Z'
           status='completed'/>
</iq>
```

4.5 End User Session

An administrator may need to terminate one or all of the user's current sessions, but allow future logins (this can be thought of as "kicking" rather than "banning" the user). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#end-user-session".
A sample protocol flow for this use case is shown below.

Listing 17: Admin Requests to End a User's Session

```
<iq from='bard@shakespeare.lit/globe'
    id='end-user-session-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#end-user-session'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 18: Service Returns End User Session Form to Admin

```
<iq from='shakespeare.lit'
    id='end-user-session-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#end-user-session'
           sessionid='end-user-session:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Ending a User Session</title>
      <instructions>Fill out this form to end a user&apos;s session.</
          instructions>
      <field type='hidden' var='FORM_TYPE'>
```

```
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣Jabber␣ID(s)␣for␣which␣to␣end␣sessions'
             type='jid-multi'
             var='accountjids'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Note: If the JID is of the form <user@host>, the service MUST end all of the user's sessions; if the JID is of the form <user@host/resource>, the service MUST end only the session associated with that resource.

Listing 19: Admin Submits End User Session Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='end-user-session-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#end-user-session'
           sessionid='end-user-session:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjids'>
        <value>juliet@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 20: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='end-user-session-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#end-user-session'
           sessionid='end-user-session:20040408T0337Z'
           status='completed'/>
</iq>
```

4.6 Get User Password

An administrator may need to retrieve a user's password. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-user-password".
A sample protocol flow for this use case is shown below.

Listing 21: Admin Requests to Get a User's Password

```
<iq from='bard@shakespeare.lit/globe'
    id='get-user-password-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-user-password'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 22: Service Returns Get User Password Form to Admin

```
<iq from='shakespeare.lit'
    id='get-user-password-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-password'
           sessionid='get-user-password:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Getting a User's Password</title>
      <instructions>Fill out this form to get a user&apos;s password
    .</instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The Jabber ID for which to retrieve the password'
             type='jid-single'
             var='accountjid'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Note: If the entity is an end user, the JID SHOULD be of the form <user@host>, not <user@host/resource>.

Listing 23: Admin Submits Get User Password Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='get-user-password-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-password'
           sessionid='get-user-password:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjid'>
        <value>juliet@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

Naturally, the data form included in the IQ result will include the user's password.

Listing 24: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-user-password-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-password'
           sessionid='get-user-password:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjid'>
        <value>juliet@shakespeare.lit</value>
      </field>
      <field var='password'>
        <value>R0m30</value>
      </field>
    </x>
  </command>
```

```
</iq>
```

4.7 Change User Password

An administrator may need to change a user's password. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#change-user-password".
A sample protocol flow for this use case is shown below.

Listing 25: Admin Requests to Change a User's Password

```
<iq from='bard@shakespeare.lit/globe'
    id='change-user-password-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#change-user-password
               '/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 26: Service Returns Change User Password Form to Admin

```
<iq from='shakespeare.lit'
    id='change-user-password-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#change-user-password
               '
           sessionid='change-user-password:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Changing a User Password</title>
      <instructions>Fill out this form to change a user&apos;s
          password.</instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣Jabber␣ID␣for␣this␣account'
             type='jid-single'
             var='accountjid'>
        <required/>
      </field>
```

```
      <field label='The password for this account'
             type='text-private'
             var='password'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Note: If the entity is an end user, the JID SHOULD be of the form <user@host>, not <user@host/resource>.

Listing 27: Admin Submits Change User Password Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='change-user-password-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#change-user-password
               '
           sessionid='change-user-password:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjid'>
        <value>juliet@shakespeare.lit</value>
      </field>
      <field var='password'>
        <value>V3r0n4</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 28: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='change-user-password-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#change-user-password
               '
           sessionid='change-user-password:20040408T0337Z'
           status='completed'/>
```

```
</iq>
```

4.8 Get User Roster

An administrator may need to retrieve a user's roster (e.g., to help verify the user's ownership of the account before reminding the user of the password). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-user-roster".
A sample protocol flow for this use case is shown below.

Listing 29: Admin Requests to Get a User's Roster

```
<iq from='bard@shakespeare.lit/globe'
    id='get-user-roster-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-user-roster'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 30: Service Returns Get User Roster Form to Admin

```
<iq from='shakespeare.lit'
    id='get-user-roster-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-roster'
           sessionid='get-user-roster:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Getting a User's Roster</title>
      <instructions>Fill out this form to get a user&apos;s roster.</
    instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The Jabber ID(s) for which to retrieve the roster'
             type='jid-multi'
             var='accountjids'>
        <required/>
      </field>
```

```
␣␣␣␣</x>
␣␣</command>
</iq>
```

Note: If the entity is an end user, the JID SHOULD be of the form <user@host>, not <user@host/resource>.

Listing 31: Admin Submits Get User Roster Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='get-user-roster-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-roster'
           sessionid='get-user-roster:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjids'>
        <value>juliet@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

The data form included in the IQ result will include the user's roster, formatted according to the 'jabber:iq:roster' protocol defined in XMPP IM [5].

Listing 32: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-user-roster-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-roster'
           sessionid='get-user-roster:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
```

[5] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.

```
      <field var='accountjids'>
        <value>juliet@shakespeare.lit</value>
      </field>
      <query xmlns='jabber:iq:roster'>
        <item jid='romeo@example.net'
              name='Romeo'
              subscription='both'>
          <group>Friends</group>
          <group>Lovers</group>
        </item>
        <item jid='mercutio@example.org'
              name='Mercutio'
              subscription='from'>
          <group>Friends</group>
        </item>
        <item jid='benvolio@example.org'
              name='Benvolio'
              subscription='both'>
          <group>Friends</group>
        </item>
      </query>
    </x>
  </command>
</iq>
```

4.9 Get User Last Login Time

An administrator may need to retrieve a user's last login time (e.g., to help verify the user's ownership of the account before reminding the user of the password). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-user-lastlogin".
A sample protocol flow for this use case is shown below.

Listing 33: Admin Requests to Get a User's Roster

```
<iq from='bard@shakespeare.lit/globe'
    id='get-user-lastlogin-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-user-lastlogin'/
               >
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 34: Service Returns Get User Last Login Form to Admin

```
<iq from='shakespeare.lit'
    id='get-user-lastlogin-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-lastlogin'
           sessionid='get-user-lastlogin:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Getting a User's Last Login Time</title>
      <instructions>Fill out this form to get a user&apos;s last login
    time.</instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The Jabber ID(s) for which to retrieve the last
    login time'
             type='jid-multi'
             var='accountjids'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Note: If the entity is an end user, the JID SHOULD be of the form <user@host>, not <user@host/resource>.

Listing 35: Admin Submits Get User Last Login Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='get-user-lastlogin-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-lastlogin'
           sessionid='get-user-lastlogin:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjids'>
        <value>juliet@shakespeare.lit</value>
      </field>
    </x>
  </command>
```

```
</iq>
```

The data form included in the IQ result will include the user's last login time (which SHOULD conform to the DateTime profile specified in XMPP Date and Time Profiles (XEP-0082) [6]).

Listing 36: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-user-lastlogin-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-user-lastlogin'
           sessionid='get-user-lastlogin:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjids'>
        <value>juliet@shakespeare.lit</value>
      </field>
      <field var='lastlogin'>
        <value>2003-12-19T17:58:35Z</value>
      </field>
    </x>
  </command>
</iq>
```

4.10 Get User Statistics

An administrator may want to gather statistics about a particular user's interaction with the service (roster size, bandwidth usage, logins, IP address, etc.). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#user-stats".
A sample protocol flow for this use case is shown below.

Listing 37: Admin Requests User Statistics

```
<iq from='bard@shakespeare.lit/globe'
    id='user-stats-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
```

[6] XEP-0082: XMPP Date and Time Profiles <https://xmpp.org/extensions/xep-0082.html>.

```
            node='http://jabber.org/protocol/admin#user-stats'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 38: Service Returns User Statistics Form to Admin

```
<iq from='shakespeare.lit'
    id='user-stats-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#user-stats'
           sessionid='user-stats:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Get User Statistics</title>
      <instructions>Fill out this form to gather user statistics.</
          instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣Jabber␣ID␣for␣statistics'
             type='jid-single'
             var='accountjid'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Listing 39: Admin Submits User Statistics Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='user-stats-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#user-stats'
           sessionid='user-stats:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='accountjid'>
        <value>iago@shakespeare.lit</value>
```

```
      </field>
    </x>
  </command>
</iq>
```

Listing 40: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='user-stats-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#user-stats'
           sessionid='user-stats:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='ipaddresses'>
        <value>127.0.0.1</value>
      </field>
      <field var='rostersize'>
        <value>123</value>
      </field>
      <field var='onlineresources'>
        <value>work</value>
        <value>home</value>
      </field>
      <field var='stanzaspersecond'>
        <value>3</value>
      </field>
      <field var='loginsperminute'>
        <value>0.1</value>
      </field>
    </x>
  </command>
</iq>
```

4.11 Edit Blacklist

The service may enable an administrator to define one or more service-wide blacklists (lists of entities that are blocked from communications to or from the service). For example, a multi-user chat service may forbid a certain user from joining any room on the service, or may block entire domains from accessing the service. An entity specified on the blacklist MAY be a JID of any form as specified in RFC 6120 [7]; the order of JID matching SHOULD be that

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

specified for privacy lists in Privacy Lists (XEP-0016) [8].
A blacklist may prevent inbound communications, outbound communications, or both; whether to offer only bidirectional blocking or a more granular choice of inbound or outbound blocking is a matter of implementation or deployment policy. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#edit-blacklist" if blocking is bidirectional as shown below; "http://jabber.org/protocol/admin#add-to-blacklist-in" for inbound blocking only; and "http://jabber.org/protocol/admin#add-to-blacklist-out" for outbound blocking only.
A sample protocol flow for this use case is shown below.

Listing 41: Admin Requests Editing of Blacklist

```
<iq from='bard@shakespeare.lit/globe'
    id='edit-blacklist-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#edit-blacklist'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 42: Service Returns Edit Blacklist Form to Admin

```
<iq from='shakespeare.lit'
    id='edit-blacklist-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-blacklist'
           sessionid='edit-blacklist:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Editing the Blacklist</title>
      <instructions>
          Fill out this form to edit the list of entities with whom
          communications are disallowed.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣blacklist'
```

[8] XEP-0016: Privacy Lists <https://xmpp.org/extensions/xep-0016.html>.

```
            var='blacklistjids'>
        <value>marlowe.lit</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 43: Admin Submits Edit Blacklist Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='edit-blacklist-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-blacklist'
           sessionid='edit-blacklist:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='blacklistjids'>
        <value>denmark.lit</value>
        <value>france.lit</value>
        <value>marlowe.lit</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 44: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='edit-blacklist-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-blacklist'
           sessionid='edit-blacklist:20040408T0337Z'
           status='completed'/>
</iq>
```

4.12 Edit Whitelist

The service may enable an administrator to define one or more service-wide whitelists (lists of entities that are allowed to communicate the service). For example, a publish-subscribe may allow only a select list of users to publish or subscribe to nodes hosted on the service. An

entity added to a whitelist MAY be a JID of any form as specified in RFC 6120; the order of JID matching SHOULD be that specified for privacy lists in Privacy Lists (XEP-0016) [9].
As with blacklists, a whitelist may prevent inbound communications, outbound communications, or both; whether to offer only bidirectional blocking or a more granular choice of inbound or outbound blocking is a matter of implementation or deployment policy. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#add-to-whitelist" if blocking is bidirectional; "http://jabber.org/protocol/admin#add-to-whitelist-in" for inbound blocking only; and "http://jabber.org/protocol/admin#add-to-whitelist-out" for outbound blocking only.
A sample protocol flow for this use case is shown below.

Listing 45: Admin Requests Editing of Whitelist

```
<iq from='bard@shakespeare.lit/globe'
    id='edit-whitelist-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#edit-whitelist'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 46: Service Returns Edit Whitelist Form to Admin

```
<iq from='shakespeare.lit'
    id='edit-whitelist-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-whitelist'
           sessionid='edit-whitelist:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Editing the Whitelist</title>
      <instructions>
          Fill out this form to edit the list of entities with whom
          communications are allowed.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
```

[9] XEP-0016: Privacy Lists <https://xmpp.org/extensions/xep-0016.html>.

```
      <field label='The whitelist'
             var='whitelistjids'>
        <value>capulet.com</value>
        <value>denmark.lit</value>
        <value>england.lit</value>
        <value>montague.net</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 47: Admin Submits Edit Whitelist Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='edit-whitelist-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-whitelist'
           sessionid='edit-whitelist:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='whitelistjids'>
        <value>capulet.com</value>
        <value>england.lit</value>
        <value>montague.net</value>
        <value>verona.it</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 48: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='edit-whitelist-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-whitelist'
           sessionid='edit-whitelist:20040408T0337Z'
           status='completed'/>
</iq>
```

4.13 Get Number of Registered Users

It may be helpful to enable an administrator to retrieve the number of registered users. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-registered-users-num".
A sample protocol flow for this use case is shown below.

Listing 49: Admin Requests Number of Registered Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-registered-users-num-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-registered-users
               -num'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD simply return the number of registered users.

Listing 50: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-registered-users-num-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-registered-users
               -num'
           sessionid='get-registered-users-num:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='form'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣number␣of␣registered␣users'
             var='registeredusersnum'>
        <value>123</value>
      </field>
    </x>
  </command>
</iq>
```

4.14 Get Number of Disabled Users

Given that admins may be able to disable user accounts, it may be helpful to enable an administrator to retrieve the number of disabled users. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-disabled-users-num".
A sample protocol flow for this use case is shown below.

Listing 51: Admin Requests Number of Disabled Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-disabled-users-num-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-disabled-users-
               num'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD simply return the number of disabled users.

Listing 52: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-disabled-users-num-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-disabled-users-
               num'
           sessionid='get-disabled-users-num:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='form'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣number␣of␣disabled␣users'
             var='disabledusersnum'>
        <value>123</value>
      </field>
    </x>
  </command>
</iq>
```

4.15 Get Number of Online Users

It may be helpful to enable an administrator to retrieve the number of registered users who are online at any one moment. By "online user" is meant any user or account that currently has at least one connected or available resource as specified in RFC 6120 and RFC 6121, whether that user is actively sending XML stanzas or is idle. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-online-users-num".
A sample protocol flow for this use case is shown below.

Listing 53: Admin Requests Number of Online Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-online-users-num-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-online-users-num
               '/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD simply return the number of online users.

Listing 54: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-online-users-num-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-online-users-num
               '
           sessionid='get-online-users-num:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣number␣of␣online␣users'
             var='onlineusersnum'>
        <value>79</value>
      </field>
    </x>
  </command>
</iq>
```

4.16 Get Number of Active Users

Some services may distinguish users who are online and actively using the service from users who are online but idle. Therefore it may be helpful to enable an administrator to retrieve the number of online users who are active at any one moment. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-active-users-num".
A sample protocol flow for this use case is shown below.

Listing 55: Admin Requests Number of Active Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-active-users-num-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-active-users-num
               '/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD simply return the number of active users.

Listing 56: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-active-users-num-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-active-users-num
               '
           sessionid='get-online-users-num:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣number␣of␣active␣users'
             var='activeusersnum'>
        <value>66</value>
      </field>
    </x>
  </command>
</iq>
```

4.17 Get Number of Idle Users

Some services may distinguish users who are online and actively using the service from users who are online but idle. Therefore it may be helpful to enable an administrator to retrieve the number of online users who are idle at any one moment. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-idle-users-num".
A sample protocol flow for this use case is shown below.

Listing 57: Admin Requests Number of Idle Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-idle-users-num-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-idle-users-num'/
              >
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD simply return the number of idle users.

Listing 58: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-idle-users-num-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-idle-users-num'
           sessionid='get-online-users-num:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣number␣of␣idle␣users'
             var='idleusersnum'>
        <value>13</value>
      </field>
    </x>
  </command>
</iq>
```

4.18 Get List of Registered Users

On a server or service without many registered users, it may be helpful to enable an administrator to retrieve a list of all registered users. The service may need to truncate the result-set, since it could be quite large (however, any ability to limit or page through the result-set is outside the scope of this document). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-registered-users-list".
A sample protocol flow for this use case is shown below.

Listing 59: Admin Requests List of Registered Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-registered-users-list-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-registered-users
               -list'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD do one of the following:

1. If there are not many registered users, the service MAY simply return the list of registered users.
2. However, the service MAY also return a form so that the admin can specify more detailed information about the search (e.g., number of users to show).

Listing 60: Service Returns Get Registered Users Form to Admin

```
<iq from='shakespeare.lit'
    id='get-registered-users-list-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-registered-users
               -list'
           sessionid='get-registered-users-list:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Requesting List of Registered Users</title>
      <instructions>
          Fill out this form to request the registered users
          of this service.
```

```
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Maximum number of items to show'
             type='list-single'
             var='max_items'>
        <option label='25'><value>25</value></option>
        <option label='50'><value>50</value></option>
        <option label='75'><value>75</value></option>
        <option label='100'><value>100</value></option>
        <option label='150'><value>150</value></option>
        <option label='200'><value>200</value></option>
        <option label='None'><value>none</value></option>
      </field>
    </x>
  </command>
</iq>
```

Listing 61: Admin Submits Get Registered Users Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='get-registered-users-list-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-registered-users
              -list'
           sessionid='get-registered-users-list:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='max_items'>
        <value>100</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 62: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-registered-users-list-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
```

```
          node='http://jabber.org/protocol/admin#get-registered-users
             -list'
          sessionid='get-registered-users:20040408T0337Z'
          status='completed'>
    <x xmlns='jabber:x:data' type='form'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The list of all users'
             var='registereduserjids'>
        <value>bernardo@shakespeare.lit</value>
        <value>bard@shakespeare.lit</value>
        <value>cordelia@shakespeare.lit</value>
        <value>crone1@shakespeare.lit</value>
        <value>emilia@shakespeare.lit</value>
        <value>francisco@shakespeare.lit</value>
        <value>goneril@shakespeare.lit</value>
        <value>hag66@shakespeare.lit</value>
        <value>hecate@shakespeare.lit</value>
        <value>iago@shakespeare.lit</value>
        <value>kingclaudius@shakespeare.lit</value>
        <value>kinglear@shakespeare.lit</value>
        <value>laertes@shakespeare.lit</value>
        <value>macbeth@shakespeare.li</value>
        <value>mercutio@shakespeare.lit</value>
        <value>nestor@shakespeare.lit</value>
        <value>northumberland@shakespeare.lit</value>
        <value>painter@shakespeare.lit</value>
        <value>regan@shakespeare.lit</value>
        <value>timon@shakespeare.lit</value>
        <value>wiccarocks@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

The service MAY return an error (rather than a list) if the number of items is excessive or the max_items value is unnacceptable.
The service MAY specify additional fields that restrict the scope of the user list (e.g., regular expression matching for Jabber IDs), and such fields MAY be registered in the future with the XMPP Registrar; however, such fields are not defined herein.

4.19 Get List of Disabled Users

It may be helpful to enable an administrator to retrieve a list of all disabled users. The service may need to truncate the result-set, since it could be quite large (however, any ability to limit or page through the result-set is outside the scope of this document). The command node for

this use case SHOULD be "http://jabber.org/protocol/admin#get-disabled-users-list".
A sample protocol flow for this use case is shown below.

Listing 63: Admin Requests List of Disabled Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-disabled-users-list-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-disabled-users-
               list'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD do one of the following:

1. If there are not many disabled users, the service MAY simply return the list of disabled users.
2. However, the service MAY also return a form so that the admin can specify more detailed information about the search (e.g., number of users to show).

Listing 64: Service Returns Get Disabled Users Form to Admin

```
<iq from='shakespeare.lit'
    id='get-disabled-users-list-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-disabled-users-
               list'
           sessionid='get-disabled-users-list:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Requesting List of Disabled Users</title>
      <instructions>
          Fill out this form to request the disabled users
          of this service.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Maximum␣number␣of␣items␣to␣show'
             type='list-single'
```

```
            var='max_items'>
        <option label='25'><value>25</value></option>
        <option label='50'><value>50</value></option>
        <option label='75'><value>75</value></option>
        <option label='100'><value>100</value></option>
        <option label='150'><value>150</value></option>
        <option label='200'><value>200</value></option>
        <option label='None'><value>none</value></option>
      </field>
    </x>
  </command>
</iq>
```

Listing 65: Admin Submits Get Disabled Users Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='get-disabled-users-list-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-disabled-users-
               list'
           sessionid='get-disabled-users-list:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='max_items'>
        <value>100</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 66: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-disabled-users-list-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-disabled-users-
               list'
           sessionid='get-disabled-users:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
```

```
      </field>
      <field label='The list of all disabled users'
             var='disableduserjids'>
        <value>bernardo@shakespeare.lit</value>
        <value>iago@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

The service MAY return an error (rather than a list) if the number of items is excessive or the max_items value is unnacceptable.
The service MAY specify additional fields that restrict the scope of the user list (e.g., regular expression matching for Jabber IDs), and such fields MAY be registered in the future with the XMPP Registrar; however, such fields are not defined herein.

4.20 Get List of Online Users

It may be helpful to enable an administrator to retrieve a list of all online users. Because the number of online users may be quite large, the service may need to truncate the result-set, since it could be quite large (however, any ability to limit or page through the result-set is outside the scope of this document). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-online-users-list".
A sample protocol flow for this use case is shown below.

Listing 67: Admin Requests List of Online Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-online-users-list-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-online-users-
               list'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD do one of the following:

1. If there are not many online users, the service MAY simply return the list of online users.
2. However, the service MAY also return a form so that the admin can specify more detailed information about the search (e.g., number of users to show).

Listing 68: Service Returns Get Online Users Form to Admin

```
<iq from='shakespeare.lit'
    id='get-online-users-list-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-online-users-
              list'
           sessionid='get-online-users-list:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Requesting List of Online Users</title>
      <instructions>
          Fill out this form to request the online users
          of this service.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Maximum␣number␣of␣items␣to␣show'
             type='list-single'
             var='max_items'>
        <option label='25'><value>25</value></option>
        <option label='50'><value>50</value></option>
        <option label='75'><value>75</value></option>
        <option label='100'><value>100</value></option>
        <option label='150'><value>150</value></option>
        <option label='200'><value>200</value></option>
        <option label='None'><value>none</value></option>
      </field>
    </x>
  </command>
</iq>
```

Listing 69: Admin Submits Get Online Users Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='get-online-users-list-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-online-users-
              list'
           sessionid='get-online-users-list:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
```

```
      </field>
      <field var='max_items'>
        <value>100</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 70: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-online-users-list-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-online-users-
               list'
           sessionid='get-online-users:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The list of all online users'
             var='onlineuserjids'>
        <value>bard@shakespeare.lit</value>
        <value>cordelia@shakespeare.lit</value>
        <value>crone1@shakespeare.lit</value>
        <value>goneril@shakespeare.lit</value>
        <value>hag66@shakespeare.lit</value>
        <value>hecate@shakespeare.lit</value>
        <value>kinglear@shakespeare.lit</value>
        <value>macbeth@shakespeare.li</value>
        <value>mercutio@shakespeare.lit</value>
        <value>northumberland@shakespeare.lit</value>
        <value>painter@shakespeare.lit</value>
        <value>wiccarocks@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

The service MAY return an error (rather than a list) if the number of items is excessive or the max_items value is unnacceptable.
The service MAY specify additional fields that restrict the scope of the user list (e.g., regular expression matching for Jabber IDs), and such fields MAY be registered in the future with the XMPP Registrar; however, such fields are not defined herein.

4.21 Get List of Active Users

It may be helpful to enable an administrator to retrieve a list of all active users. Because the number of active users may be quite large, the service may need to truncate the result-set, since it could be quite large (however, any ability to limit or page through the result-set is outside the scope of this document). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-active-users".
A sample protocol flow for this use case is shown below.

Listing 71: Admin Requests List of Active Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-active-users-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-active-users'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD do one of the following:

1. If there are not many active users, the service MAY simply return the list of active users.
2. However, the service MAY also return a form so that the admin can specify more detailed information about the search (e.g., number of users to show).

Listing 72: Service Returns Get Active Users Form to Admin

```
<iq from='shakespeare.lit'
    id='get-active-users-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-active-users'
           sessionid='get-active-users:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Requesting List of Active Users</title>
      <instructions>
          Fill out this form to request the active users
          of this service.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
```

```
      </field>
      <field label='Maximum number of items to show'
             type='list-single'
             var='max_items'>
        <option label='25'><value>25</value></option>
        <option label='50'><value>50</value></option>
        <option label='75'><value>75</value></option>
        <option label='100'><value>100</value></option>
        <option label='150'><value>150</value></option>
        <option label='200'><value>200</value></option>
        <option label='None'><value>none</value></option>
      </field>
    </x>
  </command>
</iq>
```

Listing 73: Admin Submits Get Active Users Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='get-active-users-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-active-users'
           sessionid='get-active-users:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='max_items'>
        <value>100</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 74: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-active-users-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-active-users'
           sessionid='get-active-users:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
```

```
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The␣list␣of␣active␣users'
             var='activeuserjids'>
        <value>bard@shakespeare.lit</value>
        <value>crone1@shakespeare.lit</value>
        <value>hag66@shakespeare.lit</value>
        <value>hecate@shakespeare.lit</value>
        <value>wiccarocks@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

The service MAY return an error (rather than a list) if the number of items is excessive or the max_items value is unnacceptable.
The service MAY specify additional fields that restrict the scope of the user list (e.g., regular expression matching for Jabber IDs), and such fields MAY be registered in the future with the XMPP Registrar; however, such fields are not defined herein.

4.22 Get List of Idle Users

It may be helpful to enable an administrator to retrieve a list of all idle users. Because the number of idle users may be quite large, the service may need to truncate the result-set, since it could be quite large (however, any ability to limit or page through the result-set is outside the scope of this document). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#get-idle-users".
A sample protocol flow for this use case is shown below.

Listing 75: Admin Requests List of Active Users

```
<iq from='bard@shakespeare.lit/globe'
    id='get-idle-users-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#get-idle-users'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD do one of the following:

1. If there are not many idle users, the service MAY simply return the list of idle users.

2. However, the service MAY also return a form so that the admin can specify more detailed information about the search (e.g., number of users to show).

Listing 76: Service Returns Get Idle Users Form to Admin

```
<iq from='shakespeare.lit'
    id='get-idle-users-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-idle-users'
           sessionid='get-idle-users:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Requesting List of Active Users</title>
      <instructions>
          Fill out this form to request the idle users
          of this service.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Maximum number of items to show'
             type='list-single'
             var='max_items'>
        <option label='25'><value>25</value></option>
        <option label='50'><value>50</value></option>
        <option label='75'><value>75</value></option>
        <option label='100'><value>100</value></option>
        <option label='150'><value>150</value></option>
        <option label='200'><value>200</value></option>
        <option label='None'><value>none</value></option>
      </field>
    </x>
  </command>
</iq>
```

Listing 77: Admin Submits Get Idle Users Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='get-idle-users-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-idle-users'
           sessionid='get-idle-users:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
```

```
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='max_items'>
        <value>100</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 78: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='get-idle-users-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#get-idle-users'
           sessionid='get-active-users:20040408T0337Z'
           status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The list of idle users'
             var='activeuserjids'>
        <value>cordelia@shakespeare.lit</value>
        <value>goneril@shakespeare.lit</value>
        <value>kinglear@shakespeare.lit</value>
        <value>macbeth@shakespeare.li</value>
        <value>mercutio@shakespeare.lit</value>
        <value>northumberland@shakespeare.lit</value>
        <value>painter@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

The service MAY return an error (rather than a list) if the number of items is excessive or the max_items value is unnacceptable.
The service MAY specify additional fields that restrict the scope of the user list (e.g., regular expression matching for Jabber IDs), and such fields MAY be registered in the future with the XMPP Registrar; however, such fields are not defined herein.

4.23 Send Announcement to Online Users

Administrators of some existing Jabber servers have found it useful to be able to send an announcement to all online users of the server (e.g., to announce a server shutdown); this

concept can be extended to any service (such as a multi-user chat service or a gateway to a foreign IM service). The message shall be sent only to users who currently have a ”session” with the service. Obviously there may be latency in sending the message if the number of active users is extremely large. The command node for this use case SHOULD be ”http://jabber.org/protocol/admin#announce”.
A sample protocol flow for this use case is shown below.

Listing 79: Admin Requests Announcement

```
<iq from='bard@shakespeare.lit/globe'
    id='announce-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#announce'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 80: Service Returns Announce Form to Admin

```
<iq from='shakespeare.lit'
    id='announce-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#announce'
           sessionid='announce:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Making an Announcement</title>
      <instructions>
          Fill out this form to make an announcement to all
          active users of this service.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Announcement'
             type='text-multi'
             var='announcement'>
        <required/>
      </field>
    </x>
```

```
  </command>
</iq>
```

Listing 81: Admin Submits Announce Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='announce-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#announce'
           sessionid='announce:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='announcement'>
        <value>Attention! This service will be going down for</value>
        <value>maintenance in 2 minutes. Please log off now!</value>
        <value>We apologize for the inconvenience.</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 82: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='announce-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#announce'
           sessionid='announce:20040408T0337Z'
           status='completed'/>
</iq>
```

4.24 Set Message of the Day

Administrators of some existing Jabber servers have found it useful to be able to send a "message of the day" that is delivered to any user who logs in to the server that day (e.g., to announce service changes); [10] this concept can be extended to any service (such as a

[10] Typically, a "message of the day" is an announcement that is sent once to all users of a server or a service until and unless the message is deleted; it can be thought of as a "standing announcement" as opposed to the "one-time announcement" sent to all online users in the previous use cases. The announcement is sent immediately to users who are online when the message is set, or after the next session initiation for other users (e.g., on server login or chatroom join).

multi-user chat service or a gateway to a foreign IM service). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#set-motd".
A sample protocol flow for this use case is shown below.

Listing 83: Admin Sets Message of the Day

```
<iq from='bard@shakespeare.lit/globe'
    id='set-motd-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#set-motd'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 84: Service Returns MOTD Form to Admin

```
<iq from='shakespeare.lit'
    id='set-motd-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#set-motd'
           sessionid='set-motd:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Setting the Message of the Day</title>
      <instructions>
          Fill out this form to set the message of the day.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Message of the Day'
             type='text-multi'
             var='motd'>
        <required/>
      </field>
    </x>
  </command>
</iq>
```

Listing 85: Admin Submits MOTD Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='set-motd-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#set-motd'
           sessionid='set-motd:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='motd'>
        <value>Don&apos;t forget: the grand re-opening of the</value>
        <value>Globe Theatre will occur tomorrow night.</value>
        <value>The festivities will begin right after tea</value>
        <value>and extend far into the night. Don&apos;t miss it!</
            value>
        <value>-{}-The Globe Staff</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 86: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='set-motd-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#set-motd'
           sessionid='set-motd:20040408T0337Z'
           status='completed'/>
</iq>
```

4.25 Edit Message of the Day

After setting a message of the day, an administrator may want to edit that message (e.g., in order to correct an error). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#edit-motd".
A sample protocol flow for this use case is shown below.

Listing 87: Admin Edits Message of the Day

```
<iq from='bard@shakespeare.lit/globe'
```

```
    id='edit-motd-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#edit-motd'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form, which SHOULD include the current message of the day if one has already been set.

Listing 88: Service Returns MOTD Form to Admin

```
<iq from='shakespeare.lit'
    id='edit-motd-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-motd'
           sessionid='edit-motd:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Editing the Message of the Day</title>
      <instructions>
          Fill out this form to edit the message of the day.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Message of the Day'
             type='text-multi'
             var='motd'>
        <required/>
        <value>Don&apos;t forget: the grand re-opening of the</value>
        <value>Globe Theatre will occur tomorrow night.</value>
        <value>The festivities will begin right after tea</value>
        <value>and extend far into the night. Don&apos;t miss it!</
           value>
        <value>-{}-The Globe Stuff</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 89: Admin Submits MOTD Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='edit-motd-2'
    to='shakespeare.lit'
    type='edit'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-motd'
           sessionid='edit-motd:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='motd'>
        <value>Don&apos;t forget: the grand re-opening of the</value>
        <value>Globe Theatre will occur tomorrow night.</value>
        <value>The festivities will begin right after tea</value>
        <value>and extend far into the night. Don&apos;t miss it!</
            value>
        <value>-{}-The Globe Staff</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 90: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='edit-motd-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-motd'
           sessionid='edit-motd:20040408T0337Z'
           status='completed'/>
</iq>
```

4.26 Delete Message of the Day

Sometimes a previously-set "message of the day" is no longer appropriate and needs to be deleted. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#delete-motd".
A sample protocol flow for this use case is shown below.

Listing 91: Admin Deletes Message of the Day

```
<iq from='bard@shakespeare.lit/globe'
    id='delete-motd-1'
```

```
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#delete-motd'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD simply delete the message of the day.

Listing 92: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='delete-motd-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#delete-motd'
           sessionid='delete-motd:20040408T0337Z'
           status='completed'/>
</iq>
```

4.27 Set Welcome Message

Some existing Jabber servers send an informative "welcome message" to newly registered users of the server when they first log in; this concept can be extended to any service (such as a multi-user chat service or a gateway to a foreign IM service). The command node for this use case SHOULD be "http://jabber.org/protocol/admin#set-welcome".
A sample protocol flow for this use case is shown below.

Listing 93: Admin Sets Welcome Message

```
<iq from='bard@shakespeare.lit/globe'
    id='set-welcome-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#set-welcome'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form, which SHOULD include the current welcome message if one has already been set.

Listing 94: Service Returns Welcome Message Form to Admin

```
<iq from='shakespeare.lit'
    id='set-welcome-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#set-welcome'
           sessionid='set-welcome:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Setting Welcome Message</title>
      <instructions>
          Fill out this form to set the welcome message
          for this service.
      </instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Welcome Message'
             type='text-multi'
             var='welcome'>
        <required/>
        <value>Welcome to Shakespeare.lit, your home for</value>
        <value>instant messaging with a literary touch.</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 95: Admin Submits Welcome Message Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='set-welcome-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#set-welcome'
           sessionid='set-welcome:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='welcome'>
        <value>Welcome to Shakespeare.lit, your home for</value>
        <value>instant messaging with a literary touch.</value>
        <value>For helpful information about this service,</value>
        <value>hie thee to http://www.shakespeare.lit/</value>
```

```
      </field>
    </x>
  </command>
</iq>
```

Listing 96: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='set-welcome-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#set-welcome'
           sessionid='set-welcome:20040408T0337Z'
           status='completed'/>
</iq>
```

4.28 Delete Welcome Message

Sometimes a previously-set "welcome message" is no longer appropriate and needs to be deleted. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#delete-welcome".
A sample protocol flow for this use case is shown below.

Listing 97: Admin Deletes Welcome Message

```
<iq from='bard@shakespeare.lit/globe'
    id='delete-welcome-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#delete-welcome'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD simply delete the welcome message.

Listing 98: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='delete-welcome-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
```

```
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#delete-welcome'
           sessionid='delete-welcome:20040408T0337Z'
           status='completed'/>
</iq>
```

4.29 Edit Admin List

An administrator may want to directly edit the list of users who have administrative privileges. Whether there are distinctions between service-level administrators (e.g., owner, admin, moderator), and thus in what types of administrators are allowed to edit administrative privileges, is a matter for the implementation or local service policy and is not specified herein. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#edit-admin". A sample protocol flow for this use case is shown below.

Listing 99: Admin Requests Editing of Admin List

```
<iq from='bard@shakespeare.lit/globe'
    id='edit-admin-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#edit-admin'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 100: Service Returns Edit Admin List Form to Admin

```
<iq from='shakespeare.lit'
    id='edit-admin-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-admin'
           sessionid='edit-admin:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Editing the Admin List</title>
      <instructions>
          Fill out this form to edit the list of entities who
          have administrative privileges.
      </instructions>
```

```
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='The admin list'
             type='jid-multi'
             var='adminjids'>
        <value>bard@shakespeare.lit</value>
        <value>chris@marlowe.lit</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 101: Admin Submits Edit Admin List Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='edit-admin-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-admin'
           sessionid='edit-admin:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='adminjids'>
        <value>bard@shakespeare.lit</value>
        <value>hecate@shakespeare.lit</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 102: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='edit-admin-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#edit-admin'
           sessionid='edit-admin:20040408T0337Z'
           status='completed'/>
</iq>
```

4.30 Restart Service

A service may allow an administrator to restart the service. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#restart".
A sample protocol flow for this use case is shown below.

Listing 103: Admin Requests Restart of Service

```
<iq from='bard@shakespeare.lit/globe'
    id='restart-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#restart'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 104: Service Returns Restart Form to Admin

```
<iq from='shakespeare.lit'
    id='restart-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#restart'
           sessionid='restart:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Restarting the Service</title>
      <instructions>Fill out this form to restart the service.</
          instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Time␣delay␣before␣restarting'
             type='list-single'
             var='delay'>
        <option label='30␣seconds'><value>30</value></option>
        <option label='60␣seconds'><value>60</value></option>
        <option label='90␣seconds'><value>90</value></option>
        <option label='2␣minutes'><value>120</value></option>
        <option label='3␣minutes'><value>180</value></option>
        <option label='4␣minutes'><value>240</value></option>
        <option label='5␣minutes'><value>300</value></option>
```

```
      </field>
      <field label='Announcement'
             type='text-multi'
             var='announcement'/>
    </x>
  </command>
</iq>
```

Listing 105: Admin Submits Restart Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='restart-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#restart'
           sessionid='restart:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='delay'>
        <value>120</value>
      </field>
      <field var='announcement'>
        <value>The service will be restarted in 2 minutes!</value>
        <value>Please log off now. -{}-The Admins</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 106: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='restart-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#restart'
           sessionid='restart:20040408T0337Z'
           status='completed'/>
</iq>
```

4.31 Shut Down Service

A service may allow an administrator to shut down the service. The command node for this use case SHOULD be "http://jabber.org/protocol/admin#shutdown".
A sample protocol flow for this use case is shown below.

Listing 107: Admin Requests Shut Down of Service

```
<iq from='bard@shakespeare.lit/globe'
    id='shutdown-1'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           action='execute'
           node='http://jabber.org/protocol/admin#shutdown'/>
</iq>
```

Unless an error occurs (see the Error Handling section below), the service SHOULD return the appropriate form.

Listing 108: Service Returns Shut Down Form to Admin

```
<iq from='shakespeare.lit'
    id='shutdown-1'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#shutdown'
           sessionid='shutdown:20040408T0337Z'
           status='executing'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Shutting Down the Service</title>
      <instructions>Fill out this form to shut down the service.</
          instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field label='Time␣delay␣before␣shutting␣down'
             type='list-single'
             var='delay'>
        <option label='30␣seconds'><value>30</value></option>
        <option label='60␣seconds'><value>60</value></option>
        <option label='90␣seconds'><value>90</value></option>
        <option label='2␣minutes'><value>120</value></option>
        <option label='3␣minutes'><value>180</value></option>
        <option label='4␣minutes'><value>240</value></option>
        <option label='5␣minutes'><value>300</value></option>
```

```
      </field>
      <field label='Announcement'
             type='text-multi'
             var='announcement'/>
    </x>
  </command>
</iq>
```

Listing 109: Admin Submits Shut Down Form to Service

```
<iq from='bard@shakespeare.lit/globe'
    id='shutdown-2'
    to='shakespeare.lit'
    type='set'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#shutdown'
           sessionid='shutdown:20040408T0337Z'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>http://jabber.org/protocol/admin</value>
      </field>
      <field var='delay'>
        <value>120</value>
      </field>
      <field var='announcement'>
        <value>The service will be shut down in 2 minutes!</value>
        <value>Please log off now. -{}-The Admins</value>
      </field>
    </x>
  </command>
</iq>
```

Listing 110: Service Informs Admin of Completion

```
<iq from='shakespeare.lit'
    id='shutdown-2'
    to='bard@shakespeare.lit/globe'
    type='result'
    xml:lang='en'>
  <command xmlns='http://jabber.org/protocol/commands'
           node='http://jabber.org/protocol/admin#shutdown'
           sessionid='shutdown:20040408T0337Z'
           status='completed'/>
</iq>
```

5 Error Handling

Several error conditions are possible when an entity sends a command request to the service, as defined in the following table. If one of these error conditions occurs, the service MUST return an error stanza to the requesting entity.

Condition	Cause
<conflict/>	The command cannot be completed because of a data or system conflict (e.g., a user already exists with that username).
<feature-not-implemented/>	The specific command is not supported (even though the ad-hoc commands protocol is).
<forbidden/>	The requesting entity does not have sufficient privileges to perform the command.
<not-allowed/>	No entity is allowed to perform the command (e.g., retrieve the CEO's roster).
<service-unavailable/>	The ad-hoc commands protocol is not supported.

For the syntax of these errors, see Error Condition Mappings (XEP-0086) [11]. Naturally, other errors may be returned as well (e.g., <internal-server-error/> if the service cannot be shut down).

6 Security Considerations

The ability to complete the administrative tasks specified herein MUST NOT be granted to users who lack service-level administrative privileges.

7 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [12].

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

[12] 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/>.

8 XMPP Registrar Considerations

The XMPP Registrar [13] shall include the following information in its registries.

8.1 Protocol Namespaces

The XMPP Registrar includes "http://jabber.org/protocol/admin" in its registry of protocol namespaces.

8.2 Field Standardization

Field Standardization for Data Forms (XEP-0068) [14] defines a process for standardizing the fields used within Data Forms scoped by a particular namespace. The reserved fields for the 'http://jabber.org/protocol/admin' namespace are specified below.

```
<form_type>
  <name>http://jabber.org/protocol/admin</name>
  <doc>XEP-0133</doc>
  <desc>Forms used for administration of servers and components.</desc
     >
  <field var='accountjid'
         type='jid-single'
         label='The Jabber ID of a single entity to which an operation
            applies'/>
  <field var='accountjids'
         type='jid-multi'
         label='The Jabber ID of one or more entities to which an
            operation applies'/>
  <field var='activeuserjids'
         type='jid-multi'
         label='The Jabber IDs associated with active sessions'/>
  <field var='activeusersnum'
         type='text-single'
         label='The number of online entities that are active'/>
  <field var='adminjids'
         type='jid-multi'
         label='A list of entities with administrative privileges'/>
  <field var='announcement'
         type='text-multi'
         label='The text of an announcement to be sent to active users
            or all users'/>
```

[13] 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/>.

[14] XEP-0068: Field Data Standardization for Data Forms <https://xmpp.org/extensions/xep-0068.html>.

```
<field var='blacklistjids'
       type='jid-multi'
       label='A list of entities with whom communication is blocked'
          />
<field var='delay'
       type='list-multi'
       label='The number of seconds to delay before applying a
          change'/>
<field var='disableduserjids'
       type='jid-multi'
       label='The Jabber IDs that have been disabled'/>
<field var='disabledusersnum'
       type='text-single'
       label='The number of disabled entities'/>
<field var='email'
       type='text-single'
       label='The email address for a user'/>
<field var='given_name'
       type='text-single'
       label='The given (first) name of a user'/>
<field var='idleusersnum'
       type='text-single'
       label='The number of online entities that are idle'/>
<field var='ipaddresses'
       type='list-multi'
       label='The IP addresses of an account&apos;s online sessions'
          />
<field var='lastlogin'
       type='text-single'
       label='The last login time (per XEP-0082) of a user'/>
<field var='loginsperminute'
       type='text-single'
       label='The number of logins per minute for an account'/>
<field var='max_items'
       type='list-single'
       label='The maximum number of items associated with a search
          or list'/>
<field var='motd'
       type='text-multi'
       label='The text of a message of the day'/>
<field var='onlineresources'
       type='text-single'
       label='The names of an account&apos;s online sessions'/>
<field var='onlineuserjids'
       type='jid-multi'
       label='The Jabber IDs associated with online users'/>
<field var='onlineusersnum'
       type='text-single'
       label='The number of online entities'/>
```

```
  <field var='password'
         type='text-private'
         label='The password for an account'/>
  <field var='password-verify'
         type='text-private'
         label='Password verification'/>
  <field var='registereduserjids'
         type='jid-multi'
         label='A list of registered entities'/>
  <field var='registeredusersnum'
         type='text-single'
         label='The number of registered entities'/>
  <field var='rostersize'
         type='text-single'
         label='Number of roster items for an account'/>
  <field var='stanzaspersecond'
         type='text-single'
         label='The number of stanzas being sent per second by an
            account'/>
  <field var='surname'
         type='text-single'
         label='The family (last) name of a user'/>
  <field var='welcome'
         type='text-multi'
         label='The text of a welcome message'/>
  <field var='whitelistjids'
         type='jid-multi'
         label='A list of entities with whom communication is allowed'
            />
</form_type>
```

9 XML Schema

Because the protocol defined here is a profile of XEP-0050: Ad-Hoc Commands, no schema definition is needed.