



XMPP

XEP-0013: Flexible Offline Message Retrieval

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

Craig Kaes
<mailto:ckaes@jabber.com>
<xmpp:ckaes@corp.jabber.com>

2005-07-14
Version 1.2

Status	Type	Short Name
Draft	Standards Track	offline

This specification defines an XMPP protocol extension for flexible, POP3-like handling of offline messages. The protocol enables a connecting client to retrieve its offline messages on login in a controlled fashion, without receiving a flood of messages. Messages can also be left on the server for later retrieval.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2018 by the [XMPP Standards Foundation](#) (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1	Introduction	1
2	Use Cases	2
2.1	Discovering Server Support	2
2.2	Requesting Number of Messages	2
2.3	Requesting Message Headers	3
2.4	Retrieving Specific Messages	5
2.5	Removing Specific Messages	6
2.6	Retrieving All Messages	6
2.7	Removing All Messages	7
3	Protocol Flow	8
4	Security Considerations	9
5	IANA Considerations	9
6	XMPP Registrar Considerations	9
6.1	Protocol Namespaces	9
6.2	Service Discovery Identities	9
6.3	Well-Known Service Discovery Nodes	10
6.4	Field Standardization	10
7	XML Schema	11

1 Introduction

Although not required to do so by [XMPP Core](#)¹ and [XMPP IM](#)², many existing Jabber/XMPP instant messaging servers will store messages received while a user is offline and deliver them when the user is next online. Such messages are commonly called "offline messages". The current means of retrieving one's offline messages is simple: one sends available presence to the server and, as a consequence, the server sends a one-time "flood" of all the messages that have been stored while one was offline. This simplification has the following deficiencies:

1. It can be overwhelming, which is undesirable for the vacationer or heavy user. Many individuals upon returning to work from a weeklong vacation spend the first few hours wading through the dozens, even hundreds, of emails that they received during their absence. Unlucky, however, is this user who then logs onto their Jabber server and is bombarded by hundreds of instant messages, possibly in scores of popup dialogs, simultaneously. Should their client crash, they have lost all communication that occurred while they were away.
2. It can be difficult to integrate with web-based email clients, which is undesirable for some portals. Several large portals are currently trying to blur the distinction between IM and email -- providing both through one web interface. With offline retrieval semantics so vastly different between the two, this is quite difficult.

What is needed is a flexible semantic for offline message handling, similar to POP3 in the email world (see [RFC 1939](#)³). This would enable the wireless user to view header information for all offline messages and select only those from their boss and important clients for viewing. It would enable the vacationer to read and delete their messages one at a time, minimizing the possibility of losing all correspondence. And it would provide for seamless integration with existing web-based email clients.

In particular, such a protocol should support the following use cases:

1. Client determines server support for this protocol.
2. Client requests number of messages.
3. Client requests message "header" information (thereby choosing flexible offline message retrieval as opposed to old-fashioned "flood" mode).
4. Client retrieves specific messages.

¹RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

²RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

³RFC 1939: Post Office Protocol - Version 3 <<http://tools.ietf.org/html/rfc1939>>.

5. Client removes specific messages.
6. Client retrieves all messages.
7. Client removes all messages.

2 Use Cases

2.1 Discovering Server Support

In order to discover whether one's server supports this protocol, one uses [Service Discovery \(XEP-0030\)](#)⁴.

Listing 1: User Requests Service Discovery Information

```
<iq type='get' to='montague.net'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 2: Server Reply to Discovery Request

```
<iq type='result'
  from='montague.net'
  to='romeo@montague.net/orchard'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <feature var='http://jabber.org/protocol/offline' />
  </query>
</iq>
```

If the server supports this protocol, it **MUST** return a `<feature/>` element in the disco result with the 'var' attribute set to the namespace name for this protocol: 'http://jabber.org/protocol/offline'.

2.2 Requesting Number of Messages

RFC 1939 includes a feature (the "STAT" command) that enables a user to determine how many messages are waiting to be retrieved (without retrieving all of the headers). Such a feature would be helpful in Jabber/XMPP as well, especially if the client is constrained with regard to storage capacity or available bandwidth.

In order to determine the number of messages in the offline message queue, the user sends a disco#info request without a 'to' address (i.e., implicitly to the user himself) and with the disco node specified as 'http://jabber.org/protocol/offline':

⁴XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

Listing 3: User Requests Information About Offline Message Node

```
<iq type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info'
        node='http://jabber.org/protocol/offline' />
</iq>
```

If the server supports retrieval of the number of messages, it MUST include [Service Discovery Extensions \(XEP-0128\)](#)⁵ data specifying the number of messages:

Listing 4: Server Returns Information About Offline Message Node, Including Number of Messages

```
<iq type='result' to='romeo@montague.net/orchard'>
  <query xmlns='http://jabber.org/protocol/disco#info'
        node='http://jabber.org/protocol/offline'>
    <identity
      category='automation'
      type='message-list' />
    <feature var='http://jabber.org/protocol/offline' />
    <x xmlns='jabber:x:data' type='result'>
      <field var='FORM_TYPE' type='hidden'>
        <value>http://jabber.org/protocol/offline</value>
      </field>
      <field var='number_of_messages'>
        <value>66</value>
      </field>
    </x>
  </query>
</iq>
```

Upon receiving a service discovery request addressed to a node of "http://jabber.org/protocol/offline" (either a disco#info request as in this use case or a disco#items request as in the next use case), the server MUST NOT send a flood of offline messages if the user subsequently sends initial presence to the server during this session. Thus the user is now free to send initial presence (if desired) and to engage in normal IM activities while continuing to read through offline messages. However, once the user sends presence, the user's server MUST deliver in-session messages as usual; this document applies to offline messages only. In addition, if the user authenticates and provides presence for another resource while the first (non-flood) resource still has an active session, the server MUST NOT flood the second resource with the offline message queue.

2.3 Requesting Message Headers

In order to retrieve headers for all of the messages in the queue, the user sends a disco#items request without a 'to' address (i.e., implicitly to the user himself) and with the disco node

⁵XEP-0128: Service Discovery Extensions <<https://xmpp.org/extensions/xep-0128.html>>.

specified as 'http://jabber.org/protocol/offline'.

Listing 5: User Requests Offline Message Headers

```
<iq type='get'>
  <query xmlns='http://jabber.org/protocol/disco#items'
        node='http://jabber.org/protocol/offline' />
</iq>
```

The server now MUST return headers for all of the user's offline messages. So that the user may determine whether to view a full message, the header information provided MUST include the full Jabber ID of the sender (encoded in the 'name' attribute) and a unique identifier for the message within the user's "inbox" (encoded in the 'node' attribute), so that the user may appropriately manage (view or remove) the message.

Listing 6: Server Provides Offline Message Headers

```
<iq type='result' to='romeo@montague.net/orchard'>
  <query xmlns='http://jabber.org/protocol/disco#items'
        node='http://jabber.org/protocol/offline'>
    <item
      jid='romeo@montague.net'
      node='2003-02-27T22:49:17.008Z'
      name='mercutio@shakespeare.lit/pda' />
    <item
      jid='romeo@montague.net'
      node='2003-02-27T22:52:37.225Z'
      name='juliet@capulet.com/balcony' />
    <item
      jid='romeo@montague.net'
      node='2003-02-27T22:52:51.270Z'
      name='juliet@capulet.com/balcony' />
    <item
      jid='romeo@montague.net'
      node='2003-02-27T22:53:03.421Z'
      name='juliet@capulet.com/balcony' />
    <item
      jid='romeo@montague.net'
      node='2003-02-27T22:53:13.925Z'
      name='juliet@capulet.com/balcony' />
  </query>
</iq>
```

If the requester is a JID other than an authorized resource of the user (i.e., the user@host of the requester does not match the user@host of the user), the server MUST return a <forbidden/> error. If the requester is authorized but the node does not exist, the server MUST return an <item-not-found/> error. If there are no offline messages for this user, the server MUST return an empty query as defined in XEP-0030. (For information about the syntax of error

conditions, refer to [Error Condition Mappings \(XEP-0086\)](#) ⁶.)

The syntax and semantics of the attributes are as follows:

- The 'jid' attribute is the Jabber ID with which the item nodes are associated, i.e., the user himself.
- The 'name' attribute is the full JID of the sender as received in the 'from' address of the message itself.
- The 'node' attribute is a unique identifier for the message. The value SHOULD be considered opaque, but applications MAY perform character-by-character dictionary ordering on the values. This enables applications to implement ordering on messages, such as that shown above, wherein the node values are UTC timestamps (if timestamps are used, they MUST conform to the 'Datetime' profile defined in [XMPP Date and Time Profiles \(XEP-0082\)](#) ⁷).

2.4 Retrieving Specific Messages

Messages are viewed based on the value of the 'node' attribute as provided for each item returned by the server in the header information. A user MAY request one or more messages in the same IQ get.

Listing 7: User Requests Offline Messages

```
<iq type='get' id='view1'>
  <offline xmlns='http://jabber.org/protocol/offline'>
    <item action='view'
          node='2003-02-27T22:52:37.225Z' />
  </offline>
</iq>
```

If the requester is a JID other than an authorized resource of the user, the server MUST return a <forbidden/> error. If the requester is authorized but the node does not exist, the server MUST return an <item-not-found/> error. Otherwise, the server MUST send the requested message(s) plus an IQ result:

Listing 8: Server Provides Offline Messages

```
<message to='romeo@montague.net' from='juliet@capulet.com/balcony'>
  <body>O Romeo, Romeo! wherefore art thou Romeo?</body>
  <offline xmlns='http://jabber.org/protocol/offline'>
    <item node='2003-02-27T22:52:37.225Z' />
  </offline>
</message>
```

⁶XEP-0086: Error Condition Mappings <<https://xmpp.org/extensions/xep-0086.html>>.

⁷XEP-0082: XMPP Date and Time Profiles <<https://xmpp.org/extensions/xep-0082.html>>.

```
<iq type='result' to='user@domain/resource' id='view1' />
```

In order to distinguish incoming messages, each message MUST contain the node value. It is RECOMMENDED for the server to include [Legacy Delayed Delivery \(XEP-0091\)](#)⁸ information in the message.

2.5 Removing Specific Messages

A server MUST NOT remove a message simply because it has been requested by and delivered to the user; instead, the user must specifically request to remove a message. This further implies that the user's offline message queue SHOULD NOT be automatically cleared out by the server if there are offline messages remaining when the user's session ends. However, an implementation or deployment MAY remove messages according to its own algorithms (e.g., storage timeouts based on a "first in first out" rule) or policies (e.g., message queue size limits) if desired.

As with viewing, messages are removed based on the value of the 'node' attribute as provided for each item returned by the server in the header information. The user MAY request the removal of one or more messages in the same IQ set.

Listing 9: User Requests Removal of Offline Messages

```
<iq type='set' id='remove1'>
  <offline xmlns='http://jabber.org/protocol/offline'>
    <item action='remove'
      node='2003-02-27T22:49:17.008Z' />
    <item action='remove'
      node='2003-02-27T22:52:37.225Z' />
  </offline>
</iq>
```

If the requester is a JID other than an authorized resource of the user, the server MUST return a <forbidden/> error. If the requester is authorized but the node does not exist, the server MUST return a <item-not-found/> error. Otherwise, the server MUST remove the messages and inform the user:

Listing 10: Server Informs User of Removal

```
<iq type='result' to='romeo@montague.net/orchard' id='remove1' />
```

2.6 Retrieving All Messages

The user retrieves all message by sending the "fetch" command:

⁸XEP-0091: Legacy Delayed Delivery <<https://xmpp.org/extensions/xep-0091.html>>.

Listing 11: User Retrieval of All Offline Messages

```
<iq type='get' id='fetch1'>
  <offline xmlns='http://jabber.org/protocol/offline'>
    <fetch/>
  </offline>
</iq>
```

If the requester is a JID other than an authorized resource of the user, the server MUST return a <forbidden/> error. If the requester is authorized but the node does not exist, the server MUST return a <item-not-found/> error. Otherwise, the server MUST retrieve all messages and inform the user:

Listing 12: Server Sends All Messages and Informs User of Successful Fetch

```
<message to='romeo@montague.net' from='juliet@capulet.com/balcony'>
  <body>O Romeo, Romeo! wherefore art thou Romeo?</body>
  <offline xmlns='http://jabber.org/protocol/offline'>
    <item node='2003-02-27T22:52:37.225Z' />
  </offline>
</message>

<iq type='result' to='romeo@montague.net/orchard' id='fetch1' />
```

A client MAY retrieve all messages without first requesting message headers. In this case, the server MUST return all of the user's offline messages and also MUST NOT send a flood of offline messages if the user subsequently sends initial presence to the server during this session. That is, the semantics here are the same as for requesting message headers.

2.7 Removing All Messages

The user removes all message by sending the "purge" command:

Listing 13: User Requests Removal of Offline Messages

```
<iq type='set' id='purge1'>
  <offline xmlns='http://jabber.org/protocol/offline'>
    <purge/>
  </offline>
</iq>
```

If the requester is a JID other than an authorized resource of the user, the server MUST return a <forbidden/> error. If the requester is authorized but the node does not exist, the server MUST return a <item-not-found/> error. Otherwise, the server MUST remove all messages and inform the user:

Listing 14: Server Informs User of Successful Purge

```
<iq type='result' to='romeo@montague.net/orchard' id='purge1' />
```

3 Protocol Flow

This section shows the flow of protocol between client (C:) and server (S:) for the existing (flood) scenario and the improved (POP3-like) scenario.

```
C: <stream:stream ...>
S: <stream:stream ...>
C: authentication (SASL in XMPP, non-SASL in older systems)
S: acknowledge successful authentication
C: <presence/>
S: send message flood to Client
C: receive flood, send and receive messages, etc.
... and so on
```

The semantics change with POP-like offline message handling, and server behavior changes as well...

```
C: <stream:stream ...>
S: <stream:stream ...>
C: authentication (SASL in XMPP, non-SASL in older systems)
S: acknowledge successful authentication
C: request message headers
S: send message headers to Client
NOTE: Server now MUST NOT flood Client with offline messages.
C: <presence/>
NOTE: Server does not flood Client with offline messages, but
      sends in-session messages as usual.
```

C: request and remove offline messages, send and receive messages, etc
 .
 ... and so on

4 Security Considerations

A server MUST NOT deliver a user's offline messages to any JID except one of the user's authorized resources.

5 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](http://www.iana.org/)⁹.

6 XMPP Registrar Considerations

6.1 Protocol Namespaces

The [XMPP Registrar](http://xmpp.org/protocol-namespaces/)¹⁰ includes 'http://jabber.org/protocol/offline' in its registry of protocol namespaces.

6.2 Service Discovery Identities

The XMPP Registrar includes "automation" in its registry of Service Discovery categories for use for any entities and nodes that provide automated or programmed interaction. This document specifies the following new types for the "automation" category:

Type	Description
message-list	The node for the offline message queue; valid only for the node "http://jabber.org/protocol/offline".
message-node	A node for a specific offline message if service discovery is provided for messages.

⁹The 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/>.

¹⁰The 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/>.

The registry submission is as follows:

```
<type>
  <name>message-list</name>
  <desc>
    The node for the offline message queue; valid only for the
    node
    "http://jabber.org/protocol/offline"
  </desc>
  <doc>XEP-0013</doc>
</type>
<type>
  <name>message-node</name>
  <desc>
    A node for a specific offline message if service discovery is
    provided for messages
  </desc>
  <doc>XEP-0013</doc>
</type>
</category>
```

6.3 Well-Known Service Discovery Nodes

The XMPP Registrar includes "http://jabber.org/protocol/offline" in its registry of well-known Service Discovery nodes.

6.4 Field Standardization

[Field Standardization for Data Forms \(XEP-0068\)](#) ¹¹ defines a process for standardizing the fields used within Data Forms qualified by a particular namespace. There is one uses of such forms in offline message retrieval as described in the [Requesting Number of Messages](#) section of this XEP. The registry submission is as follows:

```
<form_type>
  <name>http://jabber.org/protocol/offline</name>
  <doc>XEP-0013</doc>
  <desc>
    Service Discovery extension for number of messages
    in an offline message queue.
  </desc>
  <field
    var='number_of_messages'
```

¹¹XEP-0068: Field Data Standardization for Data Forms <<https://xmpp.org/extensions/xep-0068.html>>.

```
    type='text-single'  
    label='N/A'/>  
</form_type>
```

7 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>  
  
<xs:schema  
  xmlns:xs='http://www.w3.org/2001/XMLSchema'  
  targetNamespace='http://jabber.org/protocol/offline'  
  xmlns='http://jabber.org/protocol/offline'  
  elementFormDefault='qualified'>  
  
  <xs:annotation>  
    <xs:documentation>  
      The protocol documented by this schema is defined in  
      XEP-0013: http://www.xmpp.org/extensions/xep-0013.html  
    </xs:documentation>  
  </xs:annotation>  
  
  <xs:element name='offline'>  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element ref='item' minOccurs='1' maxOccurs='unbounded' />  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>  
  
  <xs:element name='item'>  
    <xs:complexType>  
      <xs:attribute name='action' use='optional'>  
        <xs:simpleType>  
          <xs:restriction base='xs:NCName'>  
            <xs:enumeration value='remove' />  
            <xs:enumeration value='view' />  
          </xs:restriction>  
        </xs:simpleType>  
      </xs:attribute>  
      <xs:attribute name='jid' type='xs:string' use='optional' />  
      <xs:attribute name='node' type='xs:string' use='required' />  
    </xs:complexType>  
  </xs:element>  
  
</xs:schema>
```