



XMPP

XEP-0011: Jabber Browsing

Jeremie Miller
<mailto:jer@jabber.org>
<xmpp:jer@jabber.org>

Julian Missig
<mailto:julian@jabber.org>
<xmpp:julian@jabber.org>

Thomas Muldowney
<mailto:temas@jabber.org>
<xmpp:temas@jabber.org>

2009-06-03
Version 1.3

Status	Type	Short Name
Obsolete	Historical	iq-browse

This document defines a way to describe information about Jabber entities and the relationships between entities. Note: This document is superseded by XEP-0030: Service Discovery.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 - 2014 by the [XMPP Standards Foundation](#) (XSF).

Permissions

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

Warranty

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

Liability

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

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

Contents

1	Introduction	1
2	JID-Types	1
3	The jabber:iq:browse Namespace	3
3.1	Browsing to a JabberID	4
3.2	Generic Attributes for Browse Results	4
3.3	Expressing Relationships	4
3.4	Namespace Advertising	5
3.5	Empty Results	6
4	Supplanting jabber:iq:agents	6
5	Implementation Notes	7
6	Security Considerations	7
7	IANA Considerations	7
8	XMPP Registrar Considerations	8
9	XML Schema	8
10	Future Considerations	9

1 Introduction

The Jabber world is a diverse place, with lots of services, transports, software agents, users, groupchat rooms, translators, headline tickers, and just about anything that might interact on a real-time basis using conversational messages or presence. Every JabberID (JID) is a node that can be interacted with via messages, presence, and special purpose IQ namespaces. Some JIDs are parents (such as transports), and often many JIDs have relationships with other JIDs (such as a user to their resources, a server to its services, etc.). We need a better way to structure and manage this culture of multi-namespace JID stew. The answer: Jabber Browsing. *Note well that implementors are encouraged to implement [Service Discovery \(XEP-0030\)](#) ¹ instead of Jabber Browsing.*

2 JID-Types

One of the concepts in browsing which helps to extend the interaction between JIDs is a "JID-Type", a simple heirarchy for identifying the role of any JabberID that is similar to the mime-type format. Many programmers are comfortable with the concept of identifying file types by mime-types, which use the format "category/type". A JID-Type, once discovered, is to be used in the same way that a mime-type would be for a file, to alter the user interface representing that JID or provide alternative functionality for interacting with it (either automatically or driven by user interaction). The following categories and types are proposed as the canonical list for the purpose of JID-Types:

Category	Type	Description
application/		Specific applications running as a resource on a user@host
bot	Automated conversations	
calendar	Calendaring and scheduling service	
editor	Collaborative editor	
filesaver	Available files	
game	Multi-player game	
whiteboard	Whiteboard tool	
conference/		Nodes of this category provide multi-user chat facilities (a.k.a. conference rooms).
irc	IRC rooms (note: this enables Jabber users to connect to Internet Relay Chat rooms)	
list	Mailing-list-style conferences	
private	Private, dynamically-generated conference rooms	

¹XEP-0030: Service Discovery <<http://xmpp.org/extensions/xep-0030.html>>.

Category	Type	Description
public	Public, permanent conference rooms	Recognize different sources of headlines, GUI hints
topic	Topic-based conferences	
url	Website-hosted conferences	
headline/		
logger	Log messages (usually presented in a scrolling GUI)	Keyword-based lookup services (search engines, etc.)
notice	Alerts and warnings (usually presented as popup messages)	
rss	Rich Site Summary syndication	
stock	Stock market information by symbol (ticker)	
keyword/		Automated translation services
dictionary	Dictionary lookup service	
dns	DNS resolver	
software	Software search	
thesaurus	Thesaurus lookup service	Nodes of this category provide a link to another Instant Messaging network or messaging gateway. The 'jabber:iq:register' namespace can be used to gain access to such networks, and the 'jabber:iq:search' namespace may also be available.
web	Web search	
whois	Whois query service	
render/		
en2fr	English to French	
2*	Other language to language (using standard language codes)	
tts	Text to Speech	
service/		
aim	AIM transport	A Jabber server which conforms to the specification for the 'jabber:client' namespace
icq	ICQ transport	
irc	IRC gateway (note: this enables IRC users to connect to Jabber)	
jabber		
jud	Jabber User Directory	MSN transport
msn	MSN transport	
pager	Pager gateway	

Category	Type	Description
serverlist	A list of servers. It is assumed that this node has service/* children	
sms	SMS gateway	
smtp	SMTP gateway	
yahoo	Yahoo! transport	
user/		Nodes of this category are Jabber users, typically implementing enough of the 'jabber:client' namespace to be compliant.
client	A standard or fully-featured Jabber client compliant with the 'jabber:client' namespace	
forward	A forward alias	
inbox	An alternate inbox	
portable	A portable device implementing some of the 'jabber:client' namespace	
voice	A node providing phone or voice access	
validate/		Validation services
grammar	Grammar-checking tool	
spell	Spell-checking tool	
xml	XML validator	

Historically each category was used as the name of an element, and the type was an attribute, such as `<service type="aim"/>`. The proper expression for all new implementations supporting this specification is to express the type information as attributes on a generic item element: `<item category="service" type="aim"/>`. When processing returned browse information this new syntax should always be handled first, and the old syntax only used if it is important to be able to access older implementations.

Additional unofficial categories or types may be specified by prefixing their name with an "x-", such as "service/x-virgeim" or "x-location/gps". Changes to the official categories and subtypes may be defined either by revising this document or by activating another specification. Removal of a category or subtype must be noted in this document.

3 The jabber:iq:browse Namespace

The namespace containing the Jabber Browsing data is jabber:iq:browse. The primary element within this namespace is 'item' (again, historically every category listed above would also be an element).

3.1 Browsing to a JabberID

The common way to browse to a JabberID using IQ is:

Listing 1: Browsing to a JabberID

```
<iq type="get" to="jer@jabber.org" id="browse1">
  <query xmlns="jabber:iq:browse"/>
</iq>
```

3.2 Generic Attributes for Browse Results

The item element has these attributes in a browse result:

- `jid` [required] -- The full JabberID of the entity described.
- `category` [optional] -- One of the categories from the list above, or a non-standard category prefixed with the string "x-".
- `type` [optional] -- One of the official types from the specified category, or a non-standard type prefixed with the string "x-".
- `name` [optional] -- A friendly name that may be used in a user interface.
- `version` [optional] -- A string containing the version of the node, equivalent to the response provided to a query in the 'jabber:iq:version' namespace. This is useful for servers, especially for lists of services (see the 'service/serverlist' category/type above).

3.3 Expressing Relationships

Any item may contain any number of additional items as a child, which describes the hierarchical relationship between the parent and the child items. This relationship could be represented as a "link" in a wizard or page-based user interface, or as a branch in a tree as it is expanded. Browse results usually only contain the direct children of a node, not the grandchildren. Browsing to a user, but not a resource, will return results from the server (still with the user's JID) containing the list of resources.

For example, this could be the result of browsing to `jer@jabber.org`:

Listing 2: Result of Browsing to a User

```
<iq type="result" from="jer@jabber.org" id="browse1">
  <query
    xmlns="jabber:iq:browse"
    category="user"
    jid="jer@jabber.org"
    name="jer">
```

```

<item
  category="user"
  jid="jer@jabber.org/foxy"
  type="client"/>
<item
  category="application"
  jid="jer@jabber.org/chess"
  name="XChess"
  type="game"/>
<item
  category="user"
  jid="jer@jabber.org/palm"
  type="client"/>
</query>
</iq>

```

More definitively, throughout all of browsing, a parent describes the children, and the children when browsed to fully describe themselves. The browse data received from the child takes precedence.

Parents should list children only if they are available. This means that if for a user a child client goes offline, the parent should remove it from its browse result.

3.4 Namespace Advertising

On top of the browsing framework, a simple form of "feature advertisement" can be built. This enables any entity to advertise which features it supports, based on the namespaces associated with those features. The `<ns/>` element is allowed as a subelement of the item. This element contains a single namespace that the entity supports, and multiple `<ns/>` elements can be included in any item. For a connected client this might be `<ns>jabber:iq:oob</ns>`, or for a service `<ns>jabber:iq:search</ns>`. This list of namespaces should be used to present available options for a user or to automatically locate functionality for an application.

The children of a browse result may proactively contain a few `<ns/>` elements (such as the result of the service request to the home server), which advertises the features that the particular service supports. This list may not be complete (it is only for first-pass filtering by simpler clients), and the JID should be browsed if a complete list is required.

Clients should answer incoming browsing requests to advertise the namespaces they support.

Listing 3: Result of Browsing to a Resource

```

<iq type="result" from="jer@jabber.org/foxy" id="browse2">
  <query
    xmlns="jabber:iq:browse"
    category="user"
    jid="jer@jabber.org/foxy"
    name="laptop"
    type="client">

```



```

<ns>jabber:client</ns>
<ns>jabber:iq:browse</ns>
<ns>jabber:iq:conference</ns>
<ns>jabber:iq:time</ns>
<ns>jabber:iq:version</ns>
<ns>jabber:x:roster</ns>
<ns>jabber:x:signed</ns>
<ns>jabber:x:encrypted</ns>
</query>
</iq>

```

3.5 Empty Results

When a JabberID is browsed, the result may contain children or it may be empty. An empty result means there are no further relationships or links under that JID, which could be represented as a page containing a list of functions available for the JID, such as vCard, message, register, etc. When the result contains children, they may also be empty (as in the first result from `jer@jabber.org` above). An empty child does not mean anything, and to determine the namespaces supported or if there are more children, it must be browsed to directly.

4 Supplanting jabber:iq:agents

The first important use of `jabber:iq:browse` is to replace the `jabber:iq:agents` namespace. When a client connects, it may optionally browse to the server to which it connected in order to retrieve a list of available services. The resulting iq might look like the following example:

Listing 4: Result of Browsing to a Server

```

<iq type="result" from="jabber.org" id="browse3">
  <query
    xmlns="jabber:iq:browse"
    category="service"
    type="jabber"
    jid="jabber.org"
    name="Jabber.org_Public_Server">
    <ns>jabber:client</ns>
    <ns>jabber:iq:browse</ns>
    <ns>jabber:iq:conference</ns>
    <ns>jabber:iq:time</ns>
    <ns>jabber:iq:version</ns>
    <item
      category="service"
      jid="icq.jabber.org"
      name="ICQ_Transport"
      type="icq">

```

```
<ns>jabber:iq:register</ns>
<ns>jabber:iq:search</ns>
<ns>jabber:iq:gateway</ns>
</item>
<item
  category="conference"
  type="private"
  jid="conference.jabber.org"
  name="Private_Chatrooms"/>
<item
  category="application"
  jid="jabber.org/help"
  name="Assistance_Agent"
  type="bot"/>
</query>
</iq>
```

To determine any further details from this list, each child would have to be browsed. The elements within the icq service are only hints to a client for building user interface elements. The icq.jabber.org service would still need to be browsed in order to determine any relationships or additional namespaces. This top-level list is the master "services" list available from the server, and should be used for any default functionality when available. This list could also serve as the "home page" for a page-based browsing user interface.

5 Implementation Notes

A client should not just blindly request browse information every time the user requests it, rather, a client should cache the browse results based on JabberID. Any display or use of the browse data should then be returned from the cache. This model is similar to that of presence.

6 Security Considerations

There are no security features or concerns related to this proposal.

7 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)².

²The 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

No action on the part of the [XMPP Registrar](#)³ is necessary as a result of this document, since 'jabber:iq:browse' is already a registered protocol namespace.

9 XML Schema

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

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

  <xs:element name='query'>
    <xs:complexType>
      <xs:choice minOccurs='0' maxOccurs='unbounded'>
        <xs:element ref='item' />
        <xs:element ref='ns' />
      </xs:choice>
      <xs:attribute name='category' type='xs:string' use='optional' />
      <xs:attribute name='jid' type='xs:string' use='optional' />
      <xs:attribute name='name' type='xs:string' use='optional' />
      <xs:attribute name='type' type='xs:string' use='optional' />
    </xs:complexType>
  </xs:element>

  <xs:element name='item'>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='ns' minOccurs='0' maxOccurs='unbounded' />
      </xs:sequence>
      <xs:attribute name='category' type='xs:string' use='optional' />
      <xs:attribute name='jid' type='xs:string' use='optional' />
      <xs:attribute name='name' type='xs:string' use='optional' />
      <xs:attribute name='type' type='xs:string' use='optional' />
    </xs:complexType>
  </xs:element>

  <xs:element name='ns' type='xs:string' />

</xs:schema>
```

³The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <http://xmpp.org/registrar/>.

10 Future Considerations

The 'jabber:iq:browse' namespace has been in use for quite some time. However, live browsing still needs to be better defined by a generic publication/subscription system. It is assumed that when such a system is defined, updates to this document will be made. It is, however, possible that no further changes to jabber:iq:browse itself may be needed.