XEP-0120: Infobits

Peter Saint-Andre
mailto:stpeter@stpeter.im
xmpp:stpeter@jabber.org
https://stpeter.im/

Matthew Miller
mailto:linuxwolf@outer-planes.net
xmpp:linuxwolf@outer-planes.net

2004-01-22
Version 0.5

Status	Type	Short Name
Retracted	Standards Track	infobits

NOTE: This proposal was retracted by the author on 2004-02-19.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 by the XMPP Standards Foundation (XSF).

Permissions

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

Warranty

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

Liability

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

Conformance

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

Contents

1 Introduction

Many Jabber protocols need to define metadata about "things" on the Jabber network; such things are not limited to entities that are addressable as JIDs (e.g., users, servers, services, and chatrooms) nor even to things that are addressable as Publish-Subscribe (XEP-0060) [1] or Service Discovery (XEP-0030) [2] nodes (e.g., Ad-Hoc Commands (XEP-0050) [3]), but may include anything that Jabber entities can communicate about, including files sent via Jabber file transfer protocols (e.g., Out-of-Band Data (XEP-0066) [4] and SI File Transfer (XEP-0096) [5]), information published via pubsub protocols (e.g., User Tune (XEP-0118) [6]), and other Jabber phenomena. To date, Jabber protocols have used disparate, specialized formats for capturing such meta-information. This document provides a generic information format that can be included whenever a protocol needs to define metadata about something on the network.

2 Background

The format defined herein uses a simple "key-value" structure. Although this may seem contrary to the XML basis of Jabber technologies, there are at least two good reasons for pursuing this approach:

1. Using namespaced XML elements would force implementations to maintain a record of all namespaces and to differentiate those that define actionable protocols (e.g., 'http://jabber.org/protocol/si') from those that define informational metadata formats (e.g., 'http://purl.org/dc/elements/1.1'); the only way to do this would be to maintain an internal list of all possible namespaces that might be encountered by an application on the network.

2. Not all metadata formats that the Jabber community may want to use exist in stable XML representations (e.g., this is true of the vCard format) or in representations for which namespaces exist. In addition, some metadata formats (e.g., Friend of a Friend (FOAF) [7]) exist only in Resource Description Framework [8], whose syntax is represented in XML but whose semantics impose a more complex structure that requires a specialized (non-XML) parser. As long as a clear mapping can be defined between such metadata formats and Jabber infobits, consistent information representation and exchange can

---

[1] XEP-0060: Publish-Subscribe <https://xmpp.org/extensions/xep-0060.html>.

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

[3] XEP-0050: Ad-Hoc Commands <https://xmpp.org/extensions/xep-0050.html>.

[4] XEP-0066: Out of Band Data <https://xmpp.org/extensions/xep-0066.html>.

[5] XEP-0096: SI File Transfer <https://xmpp.org/extensions/xep-0096.html>.

[6] XEP-0118: User Tune <https://xmpp.org/extensions/xep-0118.html>.

[7] Friend of a Friend (FOAF) <http://xmlns.com/foaf/0.1/>.

[8] Resource Description Framework (RDF) <http://www.w3.org/RDF/>.

be preserved.

3 Protocol

3.1 Protocol Basics

The "infobits" protocol defined herein provides a data format only. The container element is <info/>, which is qualified by the 'http://jabber.org/protocol/infobits' namespace. There is one allowable child of the <info/> element -- <bundle/> -- and one allowable child of the <bundle/> element -- <bit/>. In order to provide the relevant metadata, the <info/> element MAY contain an unbounded number of <bundle/> elements, each of which MAY contain an unbounded number of <bit/> elements.

Each <bundle/> element MUST possess a 'type' attribute, whose value specifies the aspect of reality to which the enclosed bits apply (e.g., geographical location). A <bundle/> element MAY also possess a 'context' attribute, whose value provides further specifying information about the kind of entities described by this bundle (e.g., a home address as opposed to a work address).

Each <bit/> element MUST possess a 'key' attribute, whose value specifies the name of the key (this MUST be an NMTOKEN as defined in XML 1.0 [9]). A <bit/> element MAY also possess a 'datatype' attribute, whose value specifies the datatype of the key (which SHOULD be a datatype specified in XML Schema Part 2 [10] or in a registry of values maintained by the XMPP Registrar, such as those described in Data Forms Validation (XEP-0122) [11]). The <bit/> element SHOULD contain XML character data that specifies the relevant value of the 'key'. A <bundle/> element MAY contain more than one <bit/> element with the same value for the 'key' attribute when necessary (e.g., two instances of 'weblog' if the person has multiple weblogs), but obviously SHOULD NOT do so if a collision would occur (e.g., two instances of 'lat' and 'lon' to define a geographical location).

Note well: *no keys are defined in this document. All such keys MUST be defined in separate specifications. Keys and associated values shown in this document are provided for explanatory purposes only.*

3.2 Discovering Support

When an entity is queried via Service Discovery (XEP-0030) [12] regarding the features it supports, it SHOULD include the 'http://jabber.org/protocol/infobits' namespace.

---

[9]Extensible Markup Language (XML) 1.0 (Fourth Edition) <http://www.w3.org/TR/REC-xml/>.

[10]XML Schema Part 2: Datatypes <http://www.w3.org/TR/xmlschema11-2/>.

[11]XEP-0122: Data Forms Validation <https://xmpp.org/extensions/xep-0122.html>.

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

4 Examples

4.1 Requesting Geographical Location Information From Another Entity

This set of examples is borrowed from User Geolocation (XEP-0080) [13].

Listing 1: Requestor Requests Geolocation

```
<iq type='get'
    from='linuxwolf@outer-planes.net/gabber2'
    to='stpeter@jabber.org'
    id='geoloc1'>
  <info xmlns='http://jabber.org/protocol/infobits'>
    <bundle type='http://jabber.org/protocol/infobits/bundles/geoloc'/
        >
  </info>
</iq>
```

Listing 2: Entity Returns Geolocation Result

```
<iq type='result'
    from='stpeter@jabber.org'
    to='linuxwolf@outer-planes.net/gabber2'
    id='geoloc1'>
  <info xmlns='http://jabber.org/protocol/infobits'>
    <bundle type='http://jabber.org/protocol/infobits/bundles/geoloc'>
      <bit key='geo.lat'>39.75477</bit>
      <bit key='geo.lon'>-104.99768</bit>
      <bit key='geo.alt'>1609</bit>
      <bit key='geo.gpserror'>10</bit>
      <bit key='DC.created'>20040116T17:15:25</bit>
      <bit key='DC.description'>Jabber, Inc.</bit>
    </bundle>
  </info>
</iq>
```

4.2 Requesting All Infobits From Another Entity

Listing 3: Requestor Requests Information

```
<iq type='get'
    from='linuxwolf@outer-planes.net/gabber2'
    to='stpeter@jabber.org'
    id='all1'>
  <info xmlns='http://jabber.org/protocol/infobits'/>
</iq>
```

---

[13]XEP-0080: User Geolocation <https://xmpp.org/extensions/xep-0080.html>.

Listing 4: Entity Returns Information Result

```
<iq type='result'
    from='stpeter@jabber.org'
    to='linuxwolf@outer-planes.net/gabber2'
    id='all1'>
  <info xmlns='http://jabber.org/protocol/infobits'>
    <bundle type='http://jabber.org/protocol/infobits/bundles/geoloc'>
      <bit key='geo.lat'>39.75477</bit>
      <bit key='geo.lon'>-104.99768</bit>
      <bit key='geo.alt'>1609</bit>
      <bit key='geo.gpserror'>10</bit>
      <bit key='DC.created'>20040116T17:15:25</bit>
      <bit key='DC.description'>Jabber, Inc.</bit>
    </bundle>
    <bundle type='http://jabber.org/protocol/infobits/bundles/name'>
      <bit key='vcard.prefix'>Mr.</bit>
      <bit key='vcard.givenname'>Peter</bit>
      <bit key='vcard.familyname'>Saint-Andre</bit>
      <bit key='vcard.nick'>stpeter</bit>
    </bundle>
    <bundle type='http://jabber.org/protocol/infobits/bundles/address'
            context='work'>
      <bit key='addr.floor'>6</bit>
      <bit key='vcard.street'>1899 Wynkoop Street</bit>
      <bit key='vcard.city'>Denver</bit>
      <bit key='vcard.region'>Colorado</bit>
      <bit key='vcard.country'>USA</bit>
      <bit key='vcard.pcode'>80202</bit>
    </bundle>
    <bundle type='http://jabber.org/protocol/infobits/bundles/address'
            context='home'>
      <bit key='vcard.city'>Denver</bit>
      <bit key='vcard.region'>Colorado</bit>
      <bit key='vcard.country'>USA</bit>
      <bit key='vcard.pcode'>80209</bit>
    </bundle>
    <bundle type='http://jabber.org/protocol/infobits/bundles/tel'
            context='work'>
      <bit key='vcard.phone'>+1 303-308-3282</bit>
      <bit key='vcard.fax'>+1 303-308-3215</bit>
    </bundle>
    <bundle type='http://jabber.org/protocol/infobits/bundles/tel'
            context='home'>
      <bit key='vcard.phone'>+1 303-555-1212</bit>
    </bundle>
    <bundle type='http://jabber.org/protocol/infobits/bundles/activity
        '>
      <bit key='rpid.activity'>busy</bit>
      <bit key='jsf.action'>writing</bit>
```

```
      <bit key='DC.created'>20040116T16:56:37</bit>
    </bundle>
    <bundle type='http://jabber.org/protocol/infobits/bundles/mood'>
      <bit key='jsf.mood'>grumpy</bit>
      <bit key='DC.created'>20040116T16:56:37</bit>
    </bundle>
  </info>
</iq>
```

4.3 Publishing Tune Information

This set of examples is borrowed from User Tune (XEP-0118) [14].

Listing 5: User Publishes Tune Information

```
<iq type='set'
    from='stpeter@jabber.org/work'
    to='pubsub.jabber.org'
    id='tunes123'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='generic/tunes/stpeter@jabber.org'>
      <item id='current'>
        <info xmlns='http://jabber.org/protocol/infobits'>
          <bundle type='http://jabber.org/protocol/infobits/bundles/
              tunes'>
            <bit key='DC.creator'>Yes</bit>
            <bit key='DC.title'>Heart of the Sunrise</bit>
            <bit key='DC.source'>Yessongs</bit>
            <bit key='DC.identifier'>3</bit>
            <bit key='DC.extent'>P11M26S</bit>
          </bundle>
        </info>
      </item>
    </publish>
  </pubsub>
</iq>
```

Listing 6: Tune Information is Delivered to All Subscribers

```
<message
    from='pubsub.jabber.org'
    to='maineboy@jabber.org/laptop'>
  <x xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='generic/tunes/stpeter@jabber.org'>
      <item id='current'>
        <info xmlns='http://jabber.org/protocol/infobits'>
```

---

[14]XEP-0118: User Tune <https://xmpp.org/extensions/xep-0118.html>.

```
      <bundle type='http://jabber.org/protocol/infobits/bundles/
         tunes'>
        <bit key='DC.creator'>Yes</bit>
        <bit key='DC.title'>Heart of the Sunrise</bit>
        <bit key='DC.source'>Yessongs</bit>
        <bit key='DC.identifier'>3</bit>
        <bit key='DC.extent'>P11M26S</bit>
      </bundle>
      </info>
    </item>
  </items>
  </x>
</message>
```

5 Security Considerations

Data provided via the infobits protocol MAY be world-readable. Access control considerations MUST be defined by any protocol that makes use of infobits.

6 Internationalization Considerations

Info key names registered with the XMPP Registrar MUST be considered as identifiers, not English-language words. For purposes of internationalization, an identifier SHOULD be rendered as a word or phrase that is appropriate to the end user's preferred language.

7 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [15].

---

[15]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

8.1 Namespaces

Upon advancement of this proposal to a status of Draft, the XMPP Registrar [16] shall add the 'http://jabber.org/protocol/infobits' namespace to its registry of official namespaces.

8.2 Registries

The XMPP Registrar shall maintain a registry of infobit keynames and associated information. All keynames MUST begin with a short prefix string (letters and numbers only), followed by the '.' character used as a separator, followed by the name of the key as determined by the particular specification or organization that is identified with the prefix. Arbitrary keynames SHOULD begin with a prefix consisting of the capital 'X' character.

The XMPP Registrar shall at its discretion reserve certain keyname prefixes for use in specifying particular classes of information. One example is the prefix 'DC', which is reserved for use by infobits specified by the Dublin Core Metadata Initiative (DCMI) [17] (for details, see Dublin Core Infobits Mapping (XEP-0121) [18]). Furthermore, the XMPP Registrar shall reserve the "XMPP" prefix for infobits related to documents created by the XMPP Working Group [19] or its successors, and shall reserve the upper-case versions of all protocol "shortnames" specified in XMPP Extension Protocol specifications (e.g., a prefix of "MUC" for infobits related to Multi-User Chat (XEP-0045) [20]).

In order to prevent naming collisions, infobits that will be used in public protocols that may interoperate with other protocols on the network SHOULD be registered with the XMPP Registrar, and MUST be so registered if they are defined in XMPP Extension Protocol specifications (however, registration of private keys is NOT REQUIRED). Registration with the XMPP Registrar shall be considered to entail reservation of that infobit on the network, and a registered bit MUST NOT be re-used by other protocols and applications for purposes other than those implied by the registry entry.

In addition to the key name, the following data may be provided (but is not required) for each bit:

1. datatype -- provide only if well-defined; otherwise assume string

---

[16] 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/>.

[17] The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to promoting the widespread adoption of interoperable metadata standards. For further information, see <http://www.dublincore.org/>.

[18] XEP-0121: Dublin Core Infobits Mapping <https://xmpp.org/extensions/xep-0121.html>.

[19] The XMPP Working Group was created by the Internet Engineering Task Force to define an adaptation of the base Jabber protocols that could be considered an IETF-approved instant messaging and presence technology. For further information, see <https://datatracker.ietf.org/wg/xmpp/charter/>.

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

2. doc -- the public identifier (e.g., XEP number or RFC number) of the document in which
the info name is defined

3. valid values -- a series of <value/> elements, encapsulated within a <values/> parent
whose 'type' attribute is set to "exhaustive" (these are the only valid values) or "nonex-
haustive" (these are some of the valid values, but this list is not exhaustive and other
values may be provided)

The registry format is as follows:

```xml
<bit key='thename'>
  <datatype>some datatype</datatype>
  <doc>some document identifier</doc>
  <values type='[exhaustive|nonexhaustive]'>
    <value>first value</value>
    <value>second value</value>
  </values>
</bit>
```

9 XML Schema

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

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

  <xs:element name='info'>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='bundle' minOccurs='1' maxOccurs='unbounded'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name='bundle'>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='bit' minOccurs='1' maxOccurs='unbounded'/>
      </xs:sequence>
      <xs:attribute name='type' type='xs:string' use='required'/>
      <xs:attribute name='context' type='xs:string' use='required'/>
    </xs:complexType>
  </xs:element>
```

```
  <xs:element name='bit'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='xs:string'>
          <xs:attribute name='key' type='xs:NMTOKEN' use='required'/>
          <xs:attribute name='datatype' type='xs:NMTOKEN' use='
              optional'/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

10 Historical Note

Before defining a new protocol for metadata, the primary author of this proposal investigated several promising technologies that could be used to meet the above requirements -- in particular, the Friend of a Friend (FOAF) vocabulary, which is a subset of the Resource Description Framework (RDF). Ultimately, the primary author concluded that, while FOAF and RDF have many merits, they are not ideal for use on the Jabber network. In particular:

1. Although RDF uses XML syntax, processing of RDF requires an RDF parser, not an XML parser. The benefits of using RDF were outweighed by the costs of requiring an RDF parser in every application that would use the metadata format.

2. By design, FOAF and RDF allow an entity to define information about any other entity; this gives rise to significant concerns about data integrity, which have not been solved (and perhaps even cannot be solved) within the context of an RDF application.

3. The processes for proposing, changing, and approving RDF-related namespaces are not well defined and are outside the control or influence of the Jabber community.

Given these and other concerns, the primary author concluded that the best course would be to define an extensible XML protocol that can be processed using tools that are standard within existing Jabber/XMPP implementations, that allows only the entity itself to define metadata about itself, and that is under the control of the Jabber/XMPP community.