



XMPP

XEP-0006: Profiles

Adam Theo

<mailto:adamtheo@theoretic.com>

<xmpp:adamtheo@theoretic.com>

Michael Hearn

<mailto:mhearn@subdimension.com>

<xmpp:tweedledee@jabber.org>

Eric Murphy

<mailto:ericmurphy@jabber.org>

<xmpp:ericmurphy@jabber.org>

2002-05-08

Version 1.1

Status	Type	Short Name
Obsolete	SIG Formation	N/A

A proposal for a more powerful and extensible protocol for the management of personal information within Jabber.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 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	Specification	1
2.1	Interface	1
2.2	Structure	1
2.3	Storage	2
2.4	Gate	2
3	Road-map	2
4	Benefits	3
5	Conclusion	3
6	History	4

1 Introduction

Although popular, the vCard spec is too inflexible and limited. These problems become apparent when a User wants to set information outside the scope of the vCard in a standard format others will know is there. The so called 'IQ-Set' in custom namespaces is not sufficient because set information is often not in a standard format that others recognize or support. A short list of problems with vCard:

- It is inflexible, as you can only store information that is defined by a subset of the vCard format.
- It is inefficient, as the whole vCard must be retrieved even if all you want to know is the name of a user.
- It is impersonal, as the vCard format was designed for electronic business cards, and as such is focused on things like business contact details. We believe that people should be able to describe themselves in much greater detail than this.
- It is unreliable, as the data is not controlled or verified by any program. I have had many problems in the past with corrupted vCards.

But instead of bringing down the entire customization of the 'IQ-Set' to only pre-defined schemas, it is important to keep the flexibility and power of it.

Therefore, we propose that Jabber needs a very flexible, extensible, and powerful information structure and retrieval system. There exist four aspects to solve for when designing an improved information storage and retrieval system for any platform, especially Jabber: Interface, Structure, Storage, and Gate.

2 Specification

2.1 Interface

Interface is the process, steps, and protocol a Service has to go through to access the information with the User's approval, for the User to approve or reject the access, and the User's Server to manage and secure it all.

Although it will be part of the Jabber protocol, it is still undecided how heavily the Profiles specification will rely on and use the Jabber system. This is the first thing that will have to be decided on, but thankfully there are some plans we have already outlined as possibilities.

2.2 Structure

Structure is the format that the information will be stored in and accessed by. It will have to be very flexible, likely eventually allowing the User to specify the info-sets they want to use

instead of having them all set by this project.

The personal information will be stored in XML documents or info-sets, called 'Profiles', possibly making use of RDF, and usually following Schema specifications, although that should not be a requirement.

2.3 Storage

Storage is describing and accessing the physical location where the Profiles are kept. It will likely start off being only the Jabber Server, but will eventually allow for remote (specialized network hard drive services) and local (on the User's local machine or a portable floppy) storage. It also describes how this data will be transferred between computers and networks in an efficient and secure manner.

This will likely be the most other-SIG dependent part of the system, since we will have to make heavy use of encryption in Jabber, and likely file transfers. So we may want to help out those related SIGs when we get to this point to make sure their results can be used by this system.

2.4 Gate

Gate gives the User complete control at all times of others' access to and power over their Profiles. Since this system is designed to hold the bulk, if not all, of a User's personal information, they *must* have a powerful way to prevent unwanted others to access their data. So there must exist a powerful access regulation framework to precisely control which, when, and how other parties can get this information.

Since we want this kept flexible and very secure, the techniques used in this system will likely be a new Jabber server module that receives special 'jabber:profiles' namespaces, compares the sending JID to a User-stored list of permission-granted JIDs, and acts upon the message accordingly, either following through or rejecting the requested transfer of information.

3 Road-map

To make sure this project progresses smoothly and orderly, it has been decided it will be split up into steps, or 'Phases'. Each of the above four aspects will be split into two to four Phases, and the Road-map for the entire project will follow along these Phases. Version 1.0 of the Profiles system will be little more than an expanded vCard schema with simple rules and permissions to regulate access to it. It will progress up to Version 4.0 or 5.0, adding in advanced verification to persuade Services to use a User's Jabber Profiles instead of forcing them to set up a local account, and also adding write permissions so Services can set receipts of purchases and similar information in a User's Account.

This Road-map will be produced soon after the SIG is set up, and will give a good feature list

and time-line to follow.

4 Benefits

Such an improved system would obviously provide for more types of information storage and management, but there are some other side-benefits that can be conceived of.

- Since the User will have all personal information in their one account, and Services can retrieve this information with permission, this could mean no more having to fill out forms, especially account sign-up forms with Services. The information can be automatically retrieved and filled in for you. This is similar to client-side form-filler applications, except it can be used from any computer, no matter what software it has installed on it.
- With write permission allowed to Services by the User, the Service can store their own Profiles about you in your Jabber Account. Information such as your Amazon wish list, receipts of purchases, calendar events, etc. This is similar to what Services do now, except you have control over where and how this information is physically stored and accessed.
- Also, with an advanced authentication system (which will be the focus of a future SIG), a Service could use your Jabber Account to verify you are who you say you are, instead of requiring you set up a new account just with them. This is often called 'Universal Accounts', and similar to Microsoft's Passport and AOL's new Magic Carpet services.
- With cookies and other auto-detection methods allowed by the User, Services can automatically detect their JID from the web browser or other tool. This eliminates the step of the User having to type in their JID, and can make this system all the more seamless to them. Combined with the above universal account benefit, this is similar to single sign-on systems such as Microsoft's Passport or AOL's new Magic Carpet.

5 Conclusion

Eventually we would like this Profiles specification to completely replace the strict vCard schema that is 'hard-coded' into the protocol. We do not expect vCard to disappear from Jabber at all, simply be one possible Profile among many in a User's Account. At the end of this SIGs existence, we would like to see it integrate the Profiles and special 'jabber:profiles' namespaces fully into the rest of the Jabber protocol, having it become the method by which all User information (such as Roster, client-side preferences, and filters) is stored and retrieved.

It is important to note that this SIG will not be a stand-alone SIG. It will draw upon many other SIGs (that currently exist and that have yet to be created). It will need encryption from the Security SIG for safe transfer of the information, a versatile forms format from the Forms

SIG for Profiles administration, and advanced authentication from a future SIG for Services to authenticate the User against their Jabber account.

6 History

The concept of Jabber Profiles was started by Eric Murphy and Mike Hearn. They both had begun to come up with similar ideas, but did not meet and exchange ideas until around early 2001. Adam Theo also came across them soon after that, and after some discussion, the three authors of this document agreed to start a serious effort on developing it. We started it as a project at Theoretic Solutions, although at that time it was as a full-fledged identity specification, complete with Profiles, Authentication, and Trust. It was not until we have now moved to set it up as an official SIG that we have split the individual parts up, to make it easier to develop.