

IMPP WG
Internet-Draft
Expires: April 27, 2003

D. Crocker
Brandenburg
A. Diacakis
F. Mazzoldi
Net Proj
C. Huitema
Microsoft
G. Klyne
Baltimore
J. Rosenberg
R. Sparks
dynamicsoft
H. Sugano
Fujitsu
J. Peterson
NeuStar
October 27, 2002

Common Profile: Presence
draft-ietf-imp-pres-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 27, 2003.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

Presence is defined in [RFC2778](#) [[12](#)]. Today, numerous presence protocols are in use (largely as components of commercial instant messaging services), and little interoperability between services based on these protocols has been achieved. This specification defines common semantics and data formats for presence to facilitate the creation of gateways between presence services.

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Abstract Presence Service	5
3.1	Overview of the Presence Service	5
3.2	Identification of PRESENTITIES and WATCHERS	7
3.3	Format of Presence Information	7
3.4	The Presence Service	7
3.4.1	The Subscribe Operation	7
3.4.2	The Notify Operation	8
3.4.3	Subscribe Operation (with Zero Duration)	8
4.	Security Considerations	10
5.	IANA Considerations	11
5.1	The PRES URI Scheme	11
	Authors' Addresses	13
A.	PRES URL IANA Registration Template	16
A.1	URL scheme name	16
A.2	URL scheme syntax	16
A.3	Character encoding considerations	16
A.4	Intended usage	16
A.5	Applications and/or protocols which use this URL scheme name	16
A.6	Interoperability considerations	16
A.7	Security considerations	17
A.8	Relevant publications	17
A.9	Person & email address to contact for further information .	17
A.10	Author/Change controller	17
A.11	Applications and/or protocols which use this URL scheme name	17
B.	Issues of Interest	18
B.1	Address Mapping	18
B.2	Source-Route Mapping	18
	References	12
C.	Acknowledgments	19
	Full Copyright Statement	20

1. Introduction

Presence is defined in [RFC2778](#) [12]. Today, numerous presence protocols are in use (largely as components of commercial instant messaging services, and little interoperability between services based on these protocols has been achieved. This specification defines semantics and data formats for common services of presence to facilitate the creation of gateways between presence services.

Service behavior is described abstractly in terms of operations invoked between the consumer and provider of a service. Accordingly, each presence service must specify how this behavior is mapped onto its own protocol interactions. The choice of strategy is a local matter, providing that there is a clear relation between the abstract behaviors of the service (as specified in this memo) and how it is faithfully realized by a particular presence service.

The parameters for each operation are defined using an abstract syntax. Although the syntax specifies the range of possible data values, each Presence and IM service must specify how well-formed instances of the abstract representation are encoded as a concrete series of bits.

For example, one strategy might transmit presence information as key/value pairs, another might use a compact binary representation, and a third might use nested containers. The choice of strategy is a local matter, providing that there is a clear relation between the abstract syntax (as specified in this memo) and how it is faithfully encoded by an particular presence service.

In order to provide a means for the preservation of end-to-end features (especially security) to pass through presence interoperability gateways, this specification also provides recommendations for presence document formats that could be employed by presence protocols.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119](#) [1] and indicate requirement levels for compliant implementations.

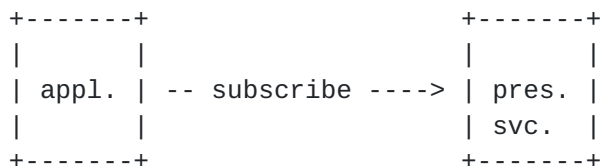
This memo makes use of the vocabulary defined in [RFC 2778](#)[9]. Terms such as CLOSED, INSTANT INBOX, PRESENCE, and OPEN are used in the same meaning as defined therein.

This document defines operations and attributes of a presence service. In order for a protocol to interface with a presence gateway, it must support all of the operations described in this document (i.e. the presence protocol must have some message or capability that provides the function described by this operation). Similarly, the attributes defined for these operations must correspond to information available in the presence protocol in order for the protocol to interface with gateways defined by this specification. Note that these attributes provide only the minimum possible information that needs to be specified for interoperability - the functions in a presence protocol that correspond to the operations described in this document can contain additional information that will not be mapped by CPIM.

3. Abstract Presence Service

3.1 Overview of the Presence Service

When an application wants to (periodically) receive the presence information associated with a PRESENTITY, it invokes the subscribe operation, e.g.,



The subscribe operation has the following attributes: watcher, target, duration, SubscriptID and TransID. The 'watcher' and 'target' identify the WATCHER and PRESENTITY, respectively, using the identifiers described in [Section 3.2](#). The duration specifies the maximum number of seconds that the SUBSCRIPTION should be active (which may be zero, in which case this is a one-time request for presence information). The SubscriptID creates a reference to the SUBSCRIPTION that is used when unsubscribing. The TransID is a unique identifier used to correlate the subscribe operation with a response operation.

Upon receiving a subscribe operation, the service immediately responds by invoking the response operation containing the same transaction- identifier, e.g.,



The response operation has the following attributes: status, TransID, and duration. 'status' indicates whether the subscribe operation has succeeded or failed. The TransID of the response operation corresponds to the TransID of the subscription operation to which it is responding. The 'duration' attribute specifies the number of seconds for which the subscription will be active (which may differ from the value requested in the subscribe operation).

If the response operation indicates success, the service immediate

invokes the notify operation to communicate the presence information to the WATCHER, e.g.,

```

+-----+               +-----+
|       |               |       |
| appl. | <----- notify -- | pres. |
|       |               | svc.  |
+-----+               +-----+

```

The notify operation has the following attributes: watcher, target, and TransID. The values of 'watcher' and 'target' are identical to those given in the subscribe operation that triggered this notify operation. The TransID is a unique identifier for this notification.

The notify operation also has content, namely PRESENCE INFORMATION. Some further information on notify content is given in [Section 3.3](#).

If the duration parameter is non-zero, then for up to the specified duration, the service invokes the notify operation whenever there are any changes to the PRESENTITY's presence information. Otherwise, exactly one notify operation is invoked, achieving a one-time poll of the presence information. Regardless, there is no application response to the notify operation (i.e., the application does not invoke a response operation when a notify operation occurs) defined in CPP.

The application may prematurely cancel a subscription by re-invoking the subscribe operation (as described above) with a duration of 0, e.g.,

```

+-----+               +-----+
|       |               |       |
| appl. | -- subscribe 0 --> | pres. |
|       |               | svc.  |
+-----+               +-----+

```

The service immediately responds by invoking the response operation containing the same transaction- identifier, e.g.,

```

+-----+               +-----+
|       |               |       |
| appl. | <----- response -- | pres. |
|       |               | svc.  |
+-----+               +-----+

```


3.2 Identification of PRESENTITIES and WATCHERS

A PRESENTITY is specified using the PRES URI scheme, which is further described in [Appendix A](#). An example would be:
"pres:fred@example.com"

To resolve identifiers associated with the Presence A client determines the address of an appropriate system running a server by resolving the destination domain name that is part of the identifier to either an intermediate relay system or a final target system.

Compliant implementations SHOULD follow the guidelines for dereferencing URIs given in [\[2\]](#).

3.3 Format of Presence Information

This specification defines an abstract interoperability mechanism for presence protocols; the message content definition given here pertains to semantics rather than syntax. However, some important properties for interoperability can only be provided if a common end-to-end format for presence is employed by the interoperating presence protocols. Implementations therefore SHOULD support the format defined in PIDF [\[10\]](#).

3.4 The Presence Service

An implementation of the service must maintain information about both presence information and in- progress operations in persistent storage.

Note that the transaction-identifier parameter used by the service is potentially long-lived. Accordingly, the values generated for this parameter should be unique across a significant duration of time.

3.4.1 The Subscribe Operation

When an application wants to (periodically) receive the presence information associated with a PRESENTITY, it invokes the subscribe operation.

When the service is informed of the subscribe operation, it performs these steps:

1. If the watcher or target parameter does not refer to a valid PRESENTITY, a response operation having status "failure" is invoked.
2. If access control does not permit the application to request this

operation, a response operation having status "failure" is invoked.

3. If the duration parameter is non-zero, and if the watcher and target parameters refer to an in-progress subscribe operation for the application, a response operation having status "failure" is invoked.
4. Otherwise, if the service is able to successfully deliver the message:

A response operation having status "success" is immediately invoked. (If the service chooses a different duration for the subscription then it conveys this information in the response operation.)

A notify operation, corresponding to the target's presence information, is immediately invoked for the watcher.

For up to the amount of time indicated by the duration parameter, if the target's presence information changes, and if access control allows, a notify operation is invoked for the watcher.

Note that if the duration parameter is zero-valued, then the subscribe operation is making a one-time poll of the presence information. Accordingly, the final step above (continued notifications for the duration of the subscription) does not occur.

When the service invokes a response operation as a result of this processing, the transID parameter is identical to the value found in the subscribe operation invoked by the application.

3.4.2 The Notify Operation

The service invokes the notify operation whenever the presence information associated with a PRESENTITY changes and there are subscribers to that information.

There is no application response to the notify operation.

3.4.3 Subscribe Operation (with Zero Duration)

When an application wants to terminate a subscription, it issues a SUBSCRIBE 0 with the transID of an existing subscription.

There is no explicit UNSUBSCRIBE command.

When an application wants to directly request presence information to be supplied immediately without initiating any persistent subscription, it issues a SUBSCRIBE 0 with a new transID.

There is no explicit FETCH command.

4. Security Considerations

Detailed security considerations for presence protocols given in [RFC2779](#) (in particular, requirements are given in sections [5.1](#) through 5.3 and some motivating discussion in 8.2).

CPP defines an interoperability function that is employed by gateways between presence protocols. CPP gateways **MUST** be compliant with the minimum security requirements of the presence protocols with which they interface.

Note that end-to-end security properties (especially confidentiality and integrity) between presentities and watchers that interface through a CPIM gateway can only be provided if a common presence format (such as the format described in [[10](#)]) is supported by the protocols interfacing with the CPIM gateway.

5. IANA Considerations

The IANA assigns the "pres" URL scheme.

5.1 The PRES URI Scheme

The Presence (PRES) URI scheme designates an Internet resource, namely a PRESENTITY or WATCHER.

The syntax of a PRES URL is given in [Appendix A](#).

References

- [1] Bradner, S., "Key words for use in RFCs to indicate requirement levels", [RFC 2119](#), March 1997.
- [2] Crocker, D., Diacakis, A., Mazzoldi, F., Huitema, C., Klyne, G., Rosenberg, J., Sparks, R., Sugano, H. and J. Peterson, "Address resolution for Instant Messaging and Presence", [draft-ietf-impv-srv-00](#) (work in progress), October 2002.
- [3] Crocker, D., "Standard for the format of ARPA Internet text Messages", [RFC 822](#), STD 11, August 1982.
- [4] Resnick, P., "Internet Message Format", [RFC 2822](#), STD 11, April 2001.
- [5] Mockapetris, P., "Domain Names - Concepts and Facilities", [RFC 1034](#), STD 13, November 1987.
- [6] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", [RFC 2045](#), November 1996.
- [7] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer, "OpenPGP Message Format", [RFC 2440](#), November 1998.
- [8] Klyne, G., "XML Coding of [RFC822](#) Messages", [draft-klyne-message-rfc822-xml-00](#) (work in progress), November 2001.
- [9] Atkins, D. and G. Klyne, "Common Presence and Instant Messaging: Message Format", [draft-ietf-impv-cpim-msgfmt-05](#) (work in progress), December 2001.
- [10] Sugano, H., "CPIM Presence Information Data Format", [draft-ietf-impv-cpim-pidf-00](#) (work in progress), August 2001.
- [11] Ramsdell, B., "S/MIME Version 3 Certificate Handling", [RFC 2632](#), June 1999.
- [12] Day, M., Rosenberg, J. and H. Sugano, "A Model for Presence and Instant Messaging", [RFC 2778](#), February 2000.
- [13] Day, M., Aggarwal, S. and J. Vincent, "Instant Messaging / Presence Protocol Requirements", [RFC 2779](#), February 2000.
- [14] Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for Specifying the Location of Services (SRV)", [RFC 2782](#), February 2000.

- [15] Allocchio, C., "GSTN Address Element Extensions in Email Services", [RFC 2846](#), June 2000.

Authors' Addresses

Dave Crocker
Brandenburg InternetWorking
675 Spruce Drive
Sunnyvale, CA 94086
US

Phone: +1 408/246-8253
EMail: dcrocker@brandenburg.com

Athanassios Diacakis
Network Projects Inc.
4516 Henry Street
Suite 113
Pittsburgh, PA 15213
US

Phone: +1 412/681-6950 x202
EMail: thanos@networkprojects.com

Florencio Mazzoldi
Network Projects Inc.
4516 Henry Street
Suite 113
Pittsburgh, PA 15213
US

Phone: +1 412/681-6950
EMail: flo@networkprojects.com

Christian Huitema
Microsoft Corporation
One Microsoft Way
Redmund, WA 98052-6399
US

EMail: huitema@microsoft.com

Graham Klyne
Baltimore Technologies
1310 Waterside
Arlington Business Park
Theale, Reading RG7 4SA
UK

Phone: +44 118 903 8000
EMail: gk@acm.org

Jonathan Rosenberg
dynamicsoft
200 Executive Drive
Suite 120
West Orange, NJ 07052
US

EMail: jdrosen@dynamicsoft.com

Robert Sparks
dynamicsoft
200 Executive Drive
Suite 120
West Orange, NJ 07052
US

EMail: rsparks@dynamicsoft.com

Hiroyasu Sugano
Fujitsu Laboratories Ltd.
200 Executive Drive
64 Nishiwaki, Ohkubo-cho
Akashi 674-8555
JP

EMail: suga@flab.fujitsu.co.jp

Jon Peterson
NeuStar, Inc.
1800 Sutter St
Suite 570
Concord, CA 94520
US

Phone: +1 925/363-8720

EMail: jon.peterson@neustar.biz

[Appendix A](#). PRES URL IANA Registration Template

This section provides the information to register the pres: presence URL .

[A.1](#) URL scheme name

pres

[A.2](#) URL scheme syntax

The syntax follows the existing mailto: URL syntax specified in [RFC2368](#). The ABNF is:

```
PRES-URL      = "pres:" [ to ] [ headers ]
to             = #mailbox
headers        = "?" header *( "&" header )
header         = hname "=" hvalue
hname          = *urlc
hvalue         = *urlc
```

[A.3](#) Character encoding considerations

Representation of non-ASCII character sets in local-part strings is limited to the standard methods provided as extensions to [RFC 2822](#)[1]

[A.4](#) Intended usage

Use of the pres: URL follows closely usage of the mailto: URL. That is, invocation of an PRES URL will cause the user's instant messaging application to start, with destination address and message headers fill-in according to the information supplied in the URL.

[A.5](#) Applications and/or protocols which use this URL scheme name

It is anticipated that protocols compliant with [RFC2779](#), and meeting the interoperability requirements specified here, will make use of this URL scheme name.

[A.6](#) Interoperability considerations

The underlying exchange protocol used to send an instant message may vary from service to service. Therefore complete, Internet-scale interoperability cannot be guaranteed. However, a service conforming to this specification permits gateways to achieve interoperability sufficient to the requirements of [RFC2779](#).

A.7 Security considerations

When PRES URLs are placed in presence protocols, they convey the identity of the sender and/or the recipient. In some cases, anonymous messaging may be desired. Such a capability is beyond the scope of this specification.

A.8 Relevant publications

[RFC2779](#), [RFC2778](#)

A.9 Person & email address to contact for further information

Jon Peterson [mailto:jon.peterson@neustar.biz]

A.10 Author/Change controller

This scheme is registered under the IETF tree. As such, IETF maintains change control.

A.11 Applications and/or protocols which use this URL scheme name

Instant messaging service; presence service

Appendix B. Issues of Interest

This appendix briefly discusses issues that may be of interest when designing an interoperation gateway.

B.1 Address Mapping

When mapping the service described in this memo, mappings that place special information into the im: address local-part MUST use the meta-syntax defined in [RFC 2846](#)[12].

B.2 Source-Route Mapping

The easiest mapping technique is a form of source- routing and usually is the least friendly to humans having to type the string. Source-routing also has a history of operational problems.

Use of source-routing for exchanges between different services is by a transformation that places the entire, original address string into the im: address local part and names the gateway in the domain part.

For example, if the destination INSTANT INBOX is "pepp://example.com/fred", then, after performing the necessary character conversions, the resulting mapping is:

```
im:pepp=example.com/fred@relay-domain
```

where "relay-domain" is derived from local configuration information.

Experience shows that it is vastly preferable to hide this mapping from end-users - if possible, the underlying software should perform the mapping automatically.

[Appendix C](#). Acknowledgments

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

