INTERNET-DRAFT
Expires: October, 1998

Gordon Mohr
Activerse, Inc.

WhoDP: Widely Hosted Object Data Protocol
draft-mohr-whodp-00.txt (03/02/98)

1. Status of this Memo

2. Abstract

The Widely Hosted Object Data Protocol (WhoDP) exists to communicate the current location and state of large numbers of dynamic, relocatable objects. Initially and most commonly, these objects are people, and the groupings or spaces that people use to enable communication.

Loosely patterned after HTTP, WhoDP implements a publish-and-subscribe model, with objects' location and identity specified via URLs. A WhoDP program "subscribes" to locate and receive information about an object; a WhoDP program "publishes" to control the location and visible state of an object.

3. Contents

4. Introduction

4.1 The Need for a "Presence" Standard

At any moment, millions of people worldwide are interacting with the Internet. But can they interact with each other?

Often they cannot, because the Internet lacks a standard, widely deployed mechanism for people to advertise dynamic information about themselves -- information such as their current

availability, capabilities, or interest in communication or collaboration.

Sharing such information gives Internet users a useful sense of "presence", creating opportunities for interaction analogous to those offered by being in the same location.

A burgeoning category of applications known as "buddy lists", "people browsers" or "colleague awareness tools" seek to provide the benefits of presence, but so far have used proprietary schemes: undocumented protocols, closed namespaces, and centrally-administered service centers.

WhoDP was designed to serve as an open presence protocol, which can bring the benefits of standardization, interoperability, and ubiquity to the presence application category.

4.2 Architectural Goals

In the spirit of the widely successful internet protocols for host-name-resolution, email, and the web, four qualities are essential for a standard, open presence protocol:

1. Scalability. Presence and instant-messaging functionality will become a standard utility available on every internet desktop and perhaps other internet-connected devices. Consequently, any protocol solution must scale to worldwide usage.
2. Openness. Since this functionality will spread across platforms, and become embedded in other applications, a standard is inevitable and desirable. Any protocol must thus enable the easy development of interoperable software by independent groups, across various implementation environments. Well-understood existing standards should be used and/or mimicked whenever appropriate.
3. Precision. An effective sense of "real-time presence" requires instantaneous, reciprocal, and reliable interaction. Thus, source and destination endpoints must be explicitly identified, and notification or communication between endpoints must be asynchronous, relatively lagless, and optionally cryptographically protected.
4. Flexibility. This level of real-time interaction between people (and other agents) opens up new application and platform possibilities, only some of which can be foreseen today. (For example, novelty or notification robots, shared gathering places, hardware device or mobile software agent monitoring and communication; etc.) Therefore, any architecture should be able to accommodate new entities and new modes of communication as they arise.

WhoDP was designed to meet these goals, allowing presence

networks to gracefully "grow like the web", without central coordination, as new servers, users, and applications come online.

4.3 Overview of Operation

WhoDP aims to allow remote observation of WhoDP objects with interesting dynamic state, regardless of from where that object is currently being served. WhoDP achieves this by giving any interesting object an "authoritative" URL, which serves not only as its unique identity but also as a pointer to the location where that object resides "by default." For example, "whodp://activerse.com/jsmith".

"Home servers" are thus the authoritative publishers of persons' presence information. A person comes "online" anywhere on the net by running a WhoDP user-agent on a capable internet-connected machine. She then contacts her "home server" to assume publishing-control over the WhoDP object representing herself. This control may involve changing the information published by the home server, causing subscriptions to be redirected elsewhere (including directly to her current machine), or some combination thereof. Control continues for as long as the user asserts it, according to minimum-activity parameters dictated by the home server.

Having established control over her own "presence", the user would then attempt to "subscribe" in turn to each of people of interest to her. These initial subscription-requests would first be directed at each person's authoritative location, from where they would either be fulfilled or redirected to a current location. A successful subscription -- whether with the authoritative server or a temporary location -- begins with a complete report of object state, and continues with a small-level of followup traffic, such as publisher-generated updates regarding the object's state, or subscriber-generated "still-interested" traffic, according to refresh parameters dictated by the publisher.

The user can go "offline" by canceling her subscriptions, and ending publishing-control over her identity WhoDP object. Notably, when someone is offline, it does not mean you cannot subscribe to her -- instead it means that you are subscribed to the relatively static version of her that is served by the "home server".

All attempts to observe a WhoDP object should be done on behalf of another WhoDP object, allowing for user-configurable limitations on who can remotely observe what.

All communication is UDP-based with a simple retry policy.

Successful attempts to subscribe to or assume publishing-control over an object result in the creation of a "session", which affects the addressing of subsequent messages regarding that

ongoing relationship. In any session (or attempt to begin a session), the side that initially requests the session is considered the 'client', whereas the side that responds to that request is considered the 'server'. A session only ends when

either side cancels it, or when communication fails for other reasons, such as network unreliability or the abnormal exit of client or server software.

Notably, any one WhoDP application often serves 'client' and 'server' roles simultaneously -- even doing both with the same remote peer application. In any session, the WhoDP location initially addressed is considered the 'target' of that session, while the WhoDP identity sought is considered the 'subject' of that session.

4.4 Notable Features

4.4.1 URL Namespace

WhoDP features a decentralized, URL-based namespace. New servers, coming online anywhere, can immediately assign names, and publish objects to existing clients, without plugging into an official server-network.

4.4.2 Lightweight Servers

WhoDP allows authoritative servers to be extremely lightweight, simple in implementation and resource requirements. In particular, if clients are encouraged to assume publishing-control by "self-publishing", causing all subscriptions to be redirected straight to the active client itself, servers paradoxically have less to do when more of their users are online, because each client effectively donates its own CPU and bandwidth to the system. Especially if the clients have wide state, rapidly changing state, or different viewable states for different subscribers, the improved scalability with "self-publishing" can be enormous.

Such an arrangement also allows a high level of robustness against server failures. If a home server becomes unavailable, the direct relationships that are already established with other self-published entities are unaffected: basic presence awareness and communication is still possible. Only new attempts to subscribe to entities at the unavailable home server will fail.

4.4.3 Subscriber Selectivity

Since WhoDP subscriptions are entered on behalf of another WhoDP identity, publishers have fine-grained control over what their subscribers see. A publishing application knows all current subscribers, and may selectively reject or publish different information to each.

5. Protocol Parameters

5.1 Protocol Version

WhoDP follows the version-numbering scheme described in Section 3.1 of the HTTP/1.1 specification [RFC2068]. WhoDP is

abbreviated "W" in version-fields. The version of WhoDP described in this document is "W/0.9".

5.2 Message Format

WhoDP messages follow the syntax of HTTP/1.1 messages [RFC2068].

5.3 Message Transport

WhoDP messages -- both requests and responses -- are sent via UDP, which places special demands on the protocol.
Since UDP is connectionless, the protocol needs facilities for matching responses to requests, and associating related series of messages. Message headers, including the headers "To", "Reply-To", "Request-ID" and "Session-ID", provide these facilities.

Since UDP is unreliable, the protocol needs simple policies for handling packet loss, duplication, or out-of-order arrival. The headers "Request-ID", "Session-ID", and "Sequence-Number" assist in handling these situations, and section 10.4 describes minimal acceptable retry policies.

Notably, WhoDP responses serve both as confirmation that the matching request was received, and as a substantive report of what effect, if any, that request had.

Since UDP reliability is inversely proportional to the size of transmitted packets, in number of maximum-transmission-units (MTUs), WhoDP is best suited for applications where individual messages can be kept small, and resend policies may choose to take message size into account.

Future versions of WhoDP may include optional, conditional, or fallback use of TCP connections for message transport. (See Section 16.2.)

5.4 URLs

WhoDP identities and locations are both described as URLs, as specified in RFC1738. The scheme of a WhoDP URL is "whodp", and the syntax follows the "Common Internet Scheme Syntax" described in Section 3.1 of RFC1738.

It is always clear from context whether a WhoDP URL is being used as an identity or a location; for example, the meaning of a WhoDP URL in any particular header is constant. Further information on the interpretation of identity URLs is available in section 10.1.

If not explicitly specified, the default port value in a WhoDP URL is 2222.

6. Terminology

WhoDP Object: A discrete, addressable, identifiable package of

data which may be published or subscribed to through this protocol. WhoDP Objects have constant identities but may have transient locations -- and may in fact have multiple locations at once. A WhoDP Object need not appear to have the same state to all concurrent subscribers at all locations.

WhoDP Program: Any program which generates WhoDP message traffic conformant to this specification. The term "user-agent" is also used to describe a WhoDP program commonly used by a single individual to access multiple remote WhoDP services.

request: A message sent by a WhoDP program which requires a reply message. Requests may be standalone (GET and PUT), session-initiating (certain SUBs and PUBs), or in-session (UPD and other SUBs and PUBs).
response: The reply to a request.

WhoDP location: A WhoDP URL describing a host, port, and URI to which a message may be addressed.

WhoDP identity: A WhoDP URL which uniquely names a WhoDP object, as well as providing the default WhoDP location for that object.

session: A related series of WhoDP messages about a specific WhoDP object -- the "Subject" of the session. Only specific WhoDP messages, when successfully acknowledged, begin a session, and a session only ends when cancelled by either side, or communication fails for other reasons.

server: A WhoDP program which meaningfully responds to SUB, PUB, GET, and PUT requests, and generates in-session UPD requests; or, in any session, the side which granted the session.

client: A WhoDP program which generates SUB, PUB, GET, and PUT requests, and responds to in-session UPD requests; or, in any session, the side which initially requested the session. It is quite common for WhoDP programs to serve as both "clients" and "servers" at the same time.

subject: The WhoDP identity to which a request or session is directed.

target: The WhoDP location to which a request or session is directed.

sender: the WhoDP identity on whose behalf a request is sent or a session is initiated.

source: the WhoDP location from which a request is sent or a session is initiated.

subscription: A session expressing interest in the current state

of a WhoDP object at a particular location -- both at the moment the subscription is entered and whenever that state changes.

publishing-control-session: A session expressing interest in remotely controlling what is published, and how, from a particular WhoDP location.

authoritative location: A WhoDP identity treated as the default location for subscribing-to or publish-controlling a WhoDP object. The WhoDP program which hosts an object's authoritative location can be considered that object's "home server".

Content-Domain: A specific binding of an interesting problem domain or data model into the WhoDP protocol. Within the broad outlines of WhoDP publishing and subscribing, various additional domain-specific semantics can be inserted. A named Content-Domain specifies additional interpretation for WhoDP traffic, especially for the continuing traffic in an ongoing session.

7. Method Definitions

7.1 SUB

The "SUB" method is used in WhoDP requests to initiate, refresh, modify, or cancel subscriptions. Its effects depend on the headers present. The Request-URI in a SUB request always specifies the location expected to provide a subscription, which may or may not match the identity for which a subscription is sought.

For the purposes of this document, an "initiating SUB request" seeks to begin a new subscription. A "continuing SUB request" seeks to operate on an already-granted subscription. The detailed semantics of a subscription are described in Section 11.

7.2 PUB

The "PUB" method is used in WhoDP requests to initiate, refresh, modify, or cancel a publication-control-session for a WhoDP object. Its effects depend on the headers present. The Request-URI in a PUB request always specifies the remote location through which publishing will be affected, which may or may not match the identity that is being published.

For the purposes of this document, an "initiating PUB request" seeks to begin control of publishing through a remote location. A "continuing PUB request" seeks to operate on an already-established publishing-control-session. The detailed semantics of a publishing-control-session are described in Section 12.

7.3 UPD

The "UPD" method is used in WhoDP requests to update, modify the information carried by, or cancel an ongoing subscription or

publishing-control-session. Its effects depend on the headers present. UPD requests are always sent by the program that granted the subscription or publishing-control-session (the

server). The Request-URI used in an UPD request is dictate by the value of the "Reply-To" header used by the client when the session was initiated.

7.4 GET

The "GET" method allows one-time retrieval of information from a WhoDP location, without indicating a desire for a continuing session.

7.5 PUT

The "PUT" method allows one-time deposit of information to a WhoDP location, without indicating a desire for a continuing session.

8. Status Code Definitions

The following status codes are defined for use in WhoDP response messages:

8.1 Informational - 1xx

Informational codes are not used in this version of WhoDP.

8.2 Successful - 2xx

8.2.1 200 OK

Used in a response to indicate that the matching request has succeeded. Additional details may be found in the response headers. Note that successful initiating SUB requests and initiating PUB requests actually generate a 201 code, indicating that a session has been created.

8.2.2 201 Created

Used in a response to indicate that the matching request has succeeded, creating what was intended. For example, a continuing subscription or publishing-control-session was created. Details about what was created are included in the response headers.

8.3 Redirection - 3xx

8.3.1 301 Moved Permanently

The requested object has been assigned a new permanent URL and any future references to this resource SHOULD be done using the returned URL, which MUST be given by the "Location" header in the response. Clients determine their own policies for automatically following this redirection, updating internal references, and/or querying the user for guidance.

8.3.2 302 Moved Temporarily

The requested resource resides temporarily under a different URI. Since the redirection may often be different, the client SHOULD continue to use the Request-URI for future requests. The "Location" header of the response MUST give the current location for this resource. Clients SHOULD automatically follow this redirection, and MAY do so transparently to the user.

8.4 Client Error - 4xx

8.4.1 400 Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

8.4.2 401 Unauthorized

The request requires user authentication. The response MUST include headers indicating a preferred security-scheme and parameters the client can use in a follow-up request to attempt authorization via that scheme. If the request already included authorization credentials, then the 401 response indicates that authorization has been refused for those credentials.

8.4.3 403 Forbidden

The server understood the request, but is refusing to fulfill it. Authorization will not help and the request SHOULD NOT be repeated. This status code is commonly used when the server does not wish to reveal exactly why the request has been refused, or when no other response is applicable.

8.4.4 404 Not Found

The server has not found anything matching the Request-URI and/or the "Subject" header.

8.4.5 410 Gone

The requested object is no longer available at the server and no forwarding address is known. This condition SHOULD be considered permanent. Clients SHOULD delete references to the URI given in the "Subject" header after user approval.

8.4.6 424 Unsupported Publishing Option

The request specified a "Publish-Via" header which cannot be supported by the server.

8.4.7 425 Unsupported Content-Domain

The request specified a "Content-Domain" header which cannot be

supported by the server, addressed location, or addressed identity.

8.4.8 426 Sessionless

The request to grant a session cannot be fulfilled, but the addressed location or identity may be available for a sessionless response. A SUB request MAY then be retried as a GET, or a PUB request retried as a PUT, if appropriate in the given Domain.

8.4.9 427 Elsewhere

The request to assume publishing-control cannot be fulfilled, because a still-valid publishing-control session already exists to an alternate source location, and the request did not include a "Repossess" header of "true".

8.5 Server Error

8.5.1 500 Internal Server Error

The server encountered an unexpected condition which prevented it from fulfilling the request.

8.5.2 501 Not Implemented

The server does not support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource.

8.5.3 503 Service Unavailable

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. If known, the length of the delay may be indicated in a "Retry-After" header. If no Retry-After is given, the client SHOULD handle the response as it would for a 500 response.

8.5.4 505 Bad Version

The server does not support, or refuses to support, the WhoDP protocol version that was used in the request message. The server is indicating that it is unable or unwilling to complete the request using the same major version as the client, other than with this error message.

9. Header Field Definitions

The following are the headers which must be understood to minimally implement the WhoDP protocol and version described here. Overlaid security-schemes and Domains may add additional

meaningful headers.

9.1 Subject:

The "Subject" header specifies the identity of the WhoDP object that a message is regarding. In SUB, PUB, GET, and PUT requests, the "Subject" is the target of the operation; in responses and UPD requests, the "Subject" confirms what identity the response or update describes.

Acceptable values for the "Subject" header are WhoDP identity URLs.
"Subject" is abbreviated "S" in messages.

9.2 Sender:

The "Sender" header declares the identity of a requestor. In SUB, PUB, GET, and PUT requests, the "Sender" describes "on whose behalf" the operation is being requested.

Acceptable values for the "Sender" header are WhoDP identity URLs.
"Sender" is abbreviated "SE" in messages.

9.3 Reply-To:

The "Reply-To" header specifies addressing information for future responses and UPD requests related to the current message. The "Reply-To" URI will be reflected in the "To" header of future responses on the same session, and in the Request-URI of future UPD requests on the same session.

Acceptable values for the "Reply-To" header are URIs, as defined in section 3.2.1 of the HTTP/1.1 specification [RFC2068].
"Reply-To" is abbreviated "RT" in messages.

9.4 To:

The "To" header supplies additional addressing information in responses, analogous to the Request-URI in the Request-Line of requests.

Acceptable values for the "To" header are URIs, as defined in section 3.2.1 of the HTTP/1.1 specification [RFC2068].

"To" is abbreviated "T" in messages.

9.5 Request-ID:

The "Request-ID" header provides additional information to help match responses to requests. A response must include a "Request-ID" value identical to that in the matching request.

Acceptable values for the "Request-ID" header are tokens, as defined in section 2.2 of the HTTP/1.1 specification [RFC2068].

"Request-ID" is abbrieviated "RI" in messages.

9.6 Session-ID:

The "Session-ID" header provides additional information to help match requests and responses to an ongoing session. Once specified in a session-granting-response, all subsequent messages regarding that session must include an identical "Session-ID" value.

Acceptable values for the "Session-ID" header are tokens, as defined in section 2.2 of the HTTP/1.1 specification [RFC2068].

"Session-ID" is abbreviated "SI" in messages.

9.7 Location:

The "Location" header field is used to specify a target for redirections.

Acceptable values for the "Location" header are absolute URLs. URLs in "Location" headers are only WhoDP identity URLs in 301 ("Moved Permanently") responses, and are WhoDP location URLs in all other cases.

"Location" is abbreviated "L" in messages.

9.8 Refresh:

The "Refresh" headers suggests or dictates a number of seconds after which a subscription or publishing-control-session will decay without activity. "Refresh" values in SUB or PUB requests are suggestions; "Refresh" values in responses to SUBs and PUBs, or in UPD requests, set the prevailing refresh-interval for the current session. A "Refresh" value of zero (0) cancels a session.

Acceptable values for the "Refresh" header are integer second counts.
"Refresh" is abbreviated "R" in messages.

9.9 Sequence-Number:

The "Sequence-Number" header serializes communication in an ongoing session. All SUB and PUB requests which do not initiate a session must include a "Sequence-Number" header, whose value is the count of like requests sent on this session. All UPD requests similarly must include a "Sequence-Number" header whose value is the number of UPDs sent on this session.

Acceptable values for the "Sequence-Number" header are integers.

"Sequence-Number" is abbreviated "SN" in messages.

9.10 Content-Domain:

The "Content-Domain" header specifies a context for

understanding WhoDP traffic. The WhoDP protocol merely establishes a minimal framework which can be used to carry various sorts of dynamic information and notifications. Defining and implementing a Domain enables refined interpretation of WhoDP messages, and especially message content-entities, appropriate to a specific application.

Acceptable values for the "Content-Domain" header are absolute URLs which serve as the unique name of a defined and documented WhoDP Domain.

If absent, the default Domain described in Section 13.1 is assumed.

"Content-Domain" is abbreviated "CD" in messages.

9.11 Content-Type:

The "Content-Type" header describes the MIME-type of any included content-entity.

Acceptable values for the "Content-Type" header are valid MIME-types.
"Content-Type" is abbreviated "CT" in messages.

9.12 Subscriber:

The "Subscriber" header describes to which particular subscriber a bulletin or publishing command applies, in the context of a publishing-control-session.

Acceptable values for the "Subscriber" header are WhoDP identity URLs or the string "*" signifying all subscribers.

"Subscriber" is abbreviated "SU" in messages.

9.13 Expires:

The "Expires" header is used to express a time after which a response or successful request should cease having an effect.

Acceptable values for the "Expires" header are either the token "Never" or dates in the format specified by RFC 1123, and referred to as HTTP-dates in the HTTP/1.1 specification [RFC2068]. For example:

        Sun, 06 Nov 1994 08:49:37 GMT

"Expires" is abbreviated "EX" in messages.

9.14 Publish-Via:

The "Publish-Via" header is used to exercise fine-grained control in a publishing-control-session over how subscription requests are treated.

Acceptable values for the "Publish-Via" header are one or more of the tokens "Fulfill", "Redirect", "Consult", "Forbid", or "Proxy".

When a "Publish-Via" value is needed, but this header is absent, a default value of "Fulfill" is assumed.

"Publish-Via" is abbreviated "PV" in messages.

9.15 Repossess:

The "Repossess" header is used to indicate whether a request for publishing-control should displace an existing publishing-control-session from elsewhere.

Acceptable values for the "Repossess" header are "T", "t", "true", "F", "f", or "false".

When a "Repossess" value is needed, but this header is absent, a default value of "true" is assumed.

"Repossess" is abbreviated "REP" in messages.

9.16 Retry-After:

The "Retry-After" header can be used to indicate how long a service is expected to be unavailable to the requesting client. The time given, as either an absolute date or second-count from now, indicates the time after which a rejected request or cancelled session may be reattempted, with a possibility to success/re-establishment.

Acceptable values for the "Retry-After" header are either integer second counts or dates in the format specified by RFC1123, refered to as HTTP-dates in the HTTP/1.1 specification [RFC2068].

"Retry-After" is abbreviated "RA" in messages.

9.17 Software:

The "Software" header is used to identify the software which generated the WhoDP message. It is roughly analogous to the "User-Agent" or "Server" headers in HTTP.

Acceptable values for the "Software" header are arbitrary strings describing the name and version of the relevant WhoDP program.

"Software" is abbreviated "SW" in messages.

9.18 Date:

The "Date" header is used to specify the time at which the

current message was generated.

Acceptable values for the "Date" header are dates in the format specified by RFC1123, referred to as HTTP-dates in the HTTP/1.1 specification.

"Date" is abbreviated "D" in messages.

10. General Behavior

10.1 Handling of Identity URLs

Certain URLs in WhoDP are distinguished as "identity URLs". In particular, URLs which appear in the "Subject", "Sender", and "Subscriber" headers always refer to a WhoDP identity -- not a transitory WhoDP location from which that identity may currently be publishing or subscribing.

WhoDP URLs follow the "Common Internet Scheme Syntax" described in section 3.1 of RFC1738. For the purposes of comparing two WhoDP identity URLs, to determine if they refer to the same WhoDP object, WhoDP applications...

- MUST NOT treat case as significant in the 'scheme' or 'host' portions of a URL.
- MUST treat case as significant in the 'url-path' portion of a URL.
- MUST NOT consider hostnames which resolve to the same IP address as identical.
- MUST NOT consider the presence or absence of a trailing "/" to be significant, if the 'url-path' portion is empty.
- MUST NOT consider the explicit specification of the default port to be different than implied specification of the default port.

For example, the following WhoDP identity URLs are prima facie identical:

```
whodp://ding.activerse.com
WHODP://ding.activerse.com
whodp://DING.activerse.COM
whodp://ding.activerse.com/
whodp://ding.activerse.com:2222
WHODP://ding.ACTIverse.com:2222/
```

The following two URLs MUST NOT be assumed to represent the same identity:

```
whodp://ding.activerse.com/bill
whodp://ding.activerse.com/Bill
```

Similarly, the following two URLs MUST NOT be assumed to

represent the same identity, even if the current DNS resolution of "ding.activerse.com" is "207.8.29.7":

whodp://ding.activerse.com/bill
whodp://207.8.29.7/bill

If a particular WhoDP server wishes to adopt a policy of case-insensitivity or hostname-equivalence, it should choose a preferred identity URL representation for each WhoDP object hosted. Requests for that entity under acceptably-close "Subject" names MAY then generate "301-Moved Permanently" replies which include the preferred name. For example, a request for "whodp://207.8.29.7/BILL" might result in a 301 reply, with the new "Location" as "whodp://ding.activerse.com/Bill".

10.2 Requests

All requests are either "standalone" (GET, PUT), "session-initiating" (SUB, PUB, without "Session-ID" header), or "in-session" (SUB, PUB, UPD, with "Session-ID" header).

Standalone and session-initiating requests MUST include a "Subject" header, and MUST NOT include either a "Session-ID" header or "Sequence-Number" header.

Any request MAY include either or both of the "Request-ID" header and the "Reply-To" header.

In-session requests MUST include a "Session-ID" header and a "Sequence-Number" header. In-session requests MAY include more than one "Session-ID" header, and if so, only the first to appear actually describes the session to which the request applies. The additional headers only have significance as described in Section 10.5.1.

10.3 Responses

So that responses can be matched with requests, given a request, the corresponding response...

- MUST include a "Subject" header, IF the request included such a header, with an identical value.
- MUST include a "Request-ID" header, IF the request had such a header, with an identical value.
- MUST include a "To" header, IF the request had a "Reply-To" header, with a corresponding value.
- MUST include a "Session-ID" header, IF the request had such a header, with an identical value to the first "Session-ID" header on the request.
- MAY include additional "Session-ID" headers, IF the request had multiple "Session-ID" headers, as described in Section 10.5.1.
- MUST include a "Sequence-Number" header, IF the request had such a header, with an identical value.

Any error response (non-2xx) to a standalone or session-

initiating request MAY include a content-entity which describes the problem in a human readable format (most likely 'text/plain' or 'text/html').

A WhoDP program SHOULD send a valid response, even if just an error, to every request it receives and understands well enough to compose a legal error response.

10.4 UDP Retry Policies

WhoDP applications may adopt their own policies for resending unacknowledged requests, within the following parameters:

- A single request should be resent no more than 10 times.
- A single request should be resent no more than once per second.
- A single request should not be resent more than 90 seconds after the initial send.
- If multiple retries fail to generate a received response within the desired amount of time, an application should wait at least 30 seconds before performing any operation which effectively restarts the retry-cycle on a similar transaction with the same remote host.

For example, an acceptable retry policy within these parameters would be: resend a request every 3 seconds, up to 5 times, until a response is received; give up if no response is received within 30 seconds after the initial send, and refrain from sending a similar request to the same destination until 60 seconds after the initial send.

Beyond these guidelines, applications are free to choose their own retry repetition, frequency, and timeout policies. By separate, explicit agreement (through mechanisms unspecified), communicating WhoDP programs may adopt retry policies outside these parameters.

10.5 Session Decay

10.5.1 Refreshing Sessions

WhoDP clients MUST maintain for each ongoing session the time of last activity on that session, 'activity' defined as either the sending of a SUB or PUB request which referred to that session and was acknowledged, or the receipt of an UPD request. If the number of seconds since last activity is equal to or greater than the currently-established session "Refresh" value, and the client wishes the session to remain valid, it MUST send a request for the purpose of "refreshing" the session. Clients SHOULD NOT send requests which are solely for the purpose of refreshing a session significantly sooner than than specified by the "Refresh" value.

A request for the purpose of refreshing a session MUST include appropriate "Session-ID" and "Sequence-Number" headers.

Any client-generated request, on any session, MAY include additional "Session-ID" headers, beyond the first, to refresh

the named sessions in a "piggy-back" fashion. The named sessions must be served from the same WhoDP server (same host & port), and must have all been created with the same "Reply-To" header value.

The server MUST only consider refreshing these additional sessions if the primary request generates a successful response (2xx), and MUST list all sessions for which activity has been noted as additional "Session-ID" headers in the response. The client MUST only consider as successfully refreshed those sessions listed in a successful response.

10.5.2 Inactivity Expiration

If a server receives no activity which refers to a session -- neither requests nor responses to server-sent UPDs -- for a period of time equal to twice the "Refresh" interval, it MAY discard the session due to inactivity. Once discarded, requests referencing the session should be answered with a status-code 404 ("Not Found").

10.5.3 Communication Failure

If a WhoDP program makes repeated attempts to deliver a request, in accordance with the guidelines in Section 10.4, and receives no response, it SHOULD discard any session(s) associated with the request on account of communication-failure. Requests referencing discarded sessions should be answered with a status-code 404 ("Not found"). The program MAY attempt to reestablish those sessions for which it was serving as the client.

11. Subscriptions

11.1 Initiating a Subscription

11.1.1 Initial Request

A SUB request for the purpose of initiating a subscription...

- MUST include a "Subject" header, specifying the WhoDP identity to which a subscription is desired.
- SHOULD include a "Sender" header, specifying the WhoDP identity representing the subscriber.
- MAY include either or both of the headers "Request-ID" and "Reply-To", as may be necessary or convenient for the sending application to recognize a response to this SUB.
- MAY include a "Refresh" header, suggesting a refresh interval for a granted subscription.
- MAY include a "Domain" header, suggesting a context for interpreting this and subsequent messages in this subscription.
- MAY include additional headers, suggesting a security scheme, or headers defined by the Domain or security scheme in use.

11.1.2 Granting a Subscription

If a subscription is granted, the response...

- MUST use the response-code 201 ("Created").
- MUST include a "Subject" header, confirming the identity this subscription regards.
- MUST include a "Request-ID" header, IF the SUB request had such a header, with a matching value.
- MUST include a "To" header, IF the SUB request had a "Reply-To" header, with a matching value.
- MUST include a "Session-ID" header.
- MUST include a "Refresh" header, IF the SUB request did not suggest a refresh-value, and MAY include a "Refresh" header which overrides a suggested value.
- MUST include a "Domain" header, unless the subscription is to be interpreted according to the default Domain described in section 13.1.
- MAY include other headers related to a security scheme or Domain in use.
- SHOULD include a content-entity describing the subject WhoDP object.

11.1.3 Redirecting a Subscription

If a subscription cannot be granted, but should be tried elsewhere, the response...

- MUST use the response-code 301 ("Moved Permanently") or response-code 302 ("Moved-Temporarily")
- MUST include a "Subject" header, confirming the identity this response regards.
- MUST include a "Request-ID" header, IF the SUB request had such a header, with a matching value.
- MUST include a "To" header, IF the SUB request had a "Reply-To" header, with a matching value.
- MUST include a "Location" header, containing the WhoDP location URL where a subscription should currently be attempted.

11.1.4 Rejecting a Subscription

If a subscription cannot be granted, nor can a suggestion for obtaining it elsewhere be made, the response...

- MUST use a response-code describing the nature of the problem.
- MUST include a "Subject" header, confirming the identity this response regards.
- MUST include a "Request-ID" header, IF the SUB request had such a header, with a matching value.
- MUST include a "To" header, IF the SUB request had a "Reply-To" header, with a matching value.
- MAY include other headers related to a security scheme or

Domain in use, that might further detail why a subscription is unavailable or help a future request succeed.

11.2 Continuing Communication in a Subscription

11.2.1 Client-to-server SUBs

A SUB request relating to an existing subscription...

- MUST continue to use the same Request-URI used in the successful subscription-initiating SUB request.
- MUST include a "Session-ID" header, matching the value returned when the subscription was granted.
- MUST include a "Sequence-Number" header, indicating the number of SUB requests sent on this subscription.
- MAY include a "Refresh" header, to suggest a new refresh value, or with the value "0", to cancel the subscription.
- MAY include other headers related to a security scheme or Domain in use.
- MAY include a content-entity interpreted according to the Domain in use.

11.2.2 Server-to-client UPDs

An UPD request relating to an existing subscription...

- MUST use a Request-URI derived from the "Reply-To" header which was in the successful initiating-SUB.
- MUST include a "Session-ID" header, matching the value returned when the subscription was granted.
- MUST include a "Sequence-Number" header, indicating the number of UPD requests sent on this subscription.
- MAY include a "Refresh" header, dictating a new refresh value, or with the value "0", to cancel the subscription.
- MAY include a "Location" header, redirecting the subscription elsewhere, IF a "Refresh" header is canceling the subscription.
- MAY include other headers related to a security scheme or Domain in use.
- MAY include a content-entity interpreted according to the Domain in use.

12. Publishing-Control-Sessions

12.1 Control Options

WhoDP's PUB method allows flexible control over how a remote location handles subscriptions, as specified through the "Publish-Via" header.

12.1.1 Fulfill

"Fulfill", the simplest and default option, indicates that a subscription or subscriptions at the target location should be granted, maintained, or denied according to prevailing policies local to that location, with no effect on the publishing-control-session.

Asserting publishing-control with the option "Fulfill" avoids receiving any information about individual subscribers; instead,

a client only controls publishing by changing the WhoDP object state published by the server.

12.1.2 Redirect

"Redirect" indicates that a subscription or subscriptions at the target location should be redirected to the location given in an accompanying "Location" header. If no "Location" is specified, the client source location is assumed.

Asserting publishing-control with the option "Redirect" causes all current and future subscriptions to be redirected to the location provided.

12.1.3 Consult

"Consult" indicates that for the applicable subscription(s), the client should be asked what to do.

Asserting publishing-control with the option "Consult" causes all current and future subscriptions to generate an UPD message to the client, allowing the client to individually specify how to handle that subscription.

12.1.4 Forbid

"Forbid" indicates that for the applicable subscription(s), existing service should be cancelled or a subscription-request should be rejected with the 403 ("Forbidden") response.

12.1.5 Proxy

"Proxy" indicates that the target location will serve as a proxy for the source location. For as long as the publishing-control-session is active, all messages arriving at the target location from the source (which do not concern the active session) will be forwarded along to other final destinations, and all messages arriving at the target from elsewhere will be forwarded back to the source location.

"Proxy" is the most complicated and resource-consumptive mode of operation, and thus may not be supported by all servers. Its behavior is described separately from the more simple options, in Section 12.4.

12.2 Initiating a Publishing-Control-Session

12.2.1 Initial Request

A PUB request for the purpose of initiating a publishing-control-session...

- MUST include a "Subject" header, specifying the WhoDP identity for which control is desired.
- MAY include either or both of the headers "Request-ID" and

"Reply-To", as may be necessary or convenient for the sending application to recognize a response to this PUB.

- MAY include a "Refresh" header, suggesting a refresh interval for a granted publishing-control-session.
- MAY include a "Publish-Via" header, specifying how all current and future subscriptions are to be handled.
- MAY include a "Location" header, if a "Publish-Via" option of "Redirect" was specified, to give the location to which subscriptions should be redirected.
- MAY include a "Repossess" header, specifying whether this request should supercede a valid ongoing publishing-control-session from elsewhere.
- MAY include a "Domain" header, suggesting a context for interpreting this and subsequent messages in this session.
- MAY include additional headers, suggesting a security scheme, or headers defined by the Domain or security scheme in use.

12.2.2 Granting a publishing-control-session

If publishing-control is granted, the response...

- MUST use the response-code 201 ("Created").
- MUST include a "Subject" header, confirming the identity this subscription regards.
- MUST include a "Session-ID" header.
- MUST include a "Refresh" header, IF the PUB request did not suggest a refresh-value, and MAY include a "Refresh" header which overrides a suggested value.
- MUST include a "Domain" header, unless the publication-control is to be interpreted according to the default Content-Domain described in section 13.1.
- MAY include "Location" headers, describing alternate names under which the target location is known.
- MAY include other headers related to a security scheme or Content-Domain in use.
- SHOULD include a content-entity describing the current state of the subject WhoDP object, at the session target location.

Note that certain "Publish-Via" options, such as "Consult", may also trigger a flurry of UPD messages on the publishing-control-session as the client is asked how to handle each existing subscription at the server.

12.2.3 Rejecting publishing-control

If publishing-control cannot be granted for any reason, the response...

- MUST use a response-code describing the nature of the problem.
- MUST include a "Subject" header, confirming the identity that

this response regards.

- MAY include other headers related to a security scheme or Content-Domain in use, that might further detail why a subscription is unavailable or help a future request succeed.

Redirection of an initial PUB request is possible, and generally means that the client can achieve the desired effect by going to the redirected location. User-agents may adopt their own policies as to whether such redirections are revealed to the user.

If another publishing-control session exists and "Repossess" was not specified, the rejection MAY include a "Location" header describing the location of the active session.

12.3 Continuing Communication in Publishing-Control

12.3.1 Client-to-server PUBs

A PUB request relating to an existing publishing-control-session...

- MUST continue to use the same Request-URI used in the successful session-initiating PUB request.
- MUST include a "Session-ID" header, matching the value returned when the session was granted.
- MUST include a "Sequence-Number" header, indicating the number of PUB requests sent on this subscription.
- MAY include a "Publish-Via" header, specifying how the target location handles subscriptions.
- MAY include a "Subscriber" header, modifying which identities that an accompanying "Publish-Via" header affects.
- MAY include a "Refresh" header, to suggest a new refresh value, or with the value "0", to cancel the session.
- MAY include other headers related to a security scheme or Domain in use.
- MAY include a content-entity interpreted according to the Domain in use.

12.3.2 Server-to-client UPDs

An UPD request relating to an existing publishing-control-session...

- MUST use a Request-URI derived from the "Reply-To" header value that was in the successful initiating-PUB.
- MUST include a "Session-ID" header, matching the value returned when the session was granted.
- MUST include a "Sequence-Number" header, indicating the number of UPD requests sent on this session.
- MAY include a "Subscriber" header, indicating the identity of a subscriber which this UPD regards.
- MAY include a "Publish-Via" header, which indicates the options available for handling the subscriber named in an accompanying "Subscriber" header.
- MAY include a "Refresh" header, dictating a new refresh

value, or with the value "0", to cancel the session.

- MAY include a "Location" header, redirecting the session elsewhere, IF a "Refresh" header is canceling the session.

- MAY include other headers related to a security scheme or Domain in use.
- MAY include a content-entity interpreted according to the Domain in use.

In a publishing-control-session established with a "Publish-Via" option of "Consult", UPD messages which include a "Subscriber" header are information about that subscriber. If such a UPD also includes a "Publish-Via" value, it indicates that the named WhoDP identity either is currently subscribing or has requested a subscription at the target location, and the tokens in the "Publish-Via" header list possible options for handling this subscriber. The client's confirming response to this UPD should include a "Publish-Via" header which chooses one of the options. If the choice is again "Consult", the client wishes to receive notification when the subscription ends, and reserves the right to "Redirect" that subscriber in a future PUB message on the current session.

A UPD which includes a "Subscriber" but an empty "Publish-Via" header indicates that the subscriber's subscription has ended, and no further publishing-control options are available with regard to them.

12.4 "Proxy" mode

The "Proxy" publishing-control option must be set for an entire publishing-control-session, at initiation.

If the request for a "Proxy" publishing-control-session is granted, then for the duration of the session, the target server MUST forward messages according to the following rules:

- Any requests or responses which come from the client source location, and specify a Request-URI or "To" header value representing a location not on the target server, MUST be forwarded to that specified location. The proxying server MAY alter the Request-URI, "To" header, or "Reply-To" header to assist the ultimate destination in understanding the message, or to indicate how responses must be addressed to find their way back to the client.
- Any requests or responses that are directed at the target location, and received from other sources, MUST be forwarded to the publishing-control-session's source location. The proxying server MAY alter the Request-URI, "To" header, or "Reply-To" header as necessary.

Modifications to the Request-URI, "To" header, or "Reply-To" header should be limited to removing routing information only needed by the proxying server, and altering "Reply-To" values so that properly-formed responses to forwarded requests can

find their way back to the actual request-originator.

13. Content-Domains

"Content-Domains" provide a context for interpreting WhoDP activity in the same manner that "Content-Type" allows the interpretation of individual content entities. Specifying a Content-Domain dictates the meaning of headers and content entities, for either a single transaction or the duration of a session.

Thus, when a new application or notification/state-sharing scheme would like to make use of WhoDP's general session and control mechanisms, but requires a different interpretation of headers, content-entities, and continuing session messages than existing WhoDP applications, a new Content-Domain may be defined. Applications operating in the new problem domain can identify each other, and those not understanding the new domain can avoid misinterpreting new traffic.

Identifiers for Content-Domains are absolute URIs, following the example of the XML namespace mechanism [REF] or the extension-identifiers of PEP [REF].

When no "Content-Domain" header is present, programs MUST consider the default domain described below as being in effect.

13.1 Default Content-Domain

13.1.1 Meaning of Content Entities

13.1.1.1 In GET and Initiating SUB Requests

Content entities in GET and initiating SUB requests, if present, should be considered the request originator's reported state for the WhoDP object named in an accompanying "Sender" header.

In successful responses (2xx) to GET and initiating SUB requests, content MUST indicate the current state of the requested "Subject", at the target location, for the requestor.

13.1.1.2 In PUT and all PUB Requests

Content entities in PUT and all PUB requests must be considered an attempt to set the state of the target location, to be equal to the included content. Any state set as a result of a PUB operation only persists as long as the enclosing publishing-control-session is valid; when it is cancelled or decays, the object MUST revert to its previous state. (Only PUTs may be used to effect indefinitely persistent changes.)

13.1.1.3 In Subscription UPDs

In Subscription UPDs, a content entity, if present, must be considered a report of the current state of the subscription's subject, at the target location, for the purposes of the

sender/source location.

13.1.1.4 In Error Responses

Any unsuccessful response (status code other than 2xx) MAY include a human-readable explanation of the problem, typically of Content-Type 'text/plain' or 'text/html'.

13.1.1.5 Elsewhere

The meaning of content entities in all other messages is undefined and programs SHOULD NOT use content entities where their meaning is undefined. Specifically, there is no default meaning for content entities which appear in:

- successful responses to PUT, PUB, or in-session SUB requests
- UPD requests inside a publishing-control-session
- successful responses to UPD requests

13.1.2 Meaning of Headers

No special headers or header meanings are defined for the default Content-Domain.

13.2 Rules for Additional Content-Domains

Defining new Content-Domains requires explaining what significance, if any, content entities and headers have in various messages, and how those entities and headers must or should be interpreted.

For example, in the default Content-Domain, PUBs and subscription UPDs must include a complete representation of the Subject's current or desired state. Other Content-Domains could offer an interpretation of PUBs or UPDs which allows them to specify only that portion of the Subject that should or has changed.

Definition of new Content-Domains must not contradict with this base specification in ways that would confuse WhoDP programs unfamiliar with the new Content-Domain.

14. Security Issues

Security issues, such as authenticating subscribers or publishers, or encrypting traffic against eavesdropping, are explicitly not considered in this specification. It is expected that adequate schemes can be overlaid into WhoDP traffic, adapted or derived from other prevailing mechanisms. Companion documents to this specification will address these issues.

15. Examples

Assume that a number of individuals are using WhoDP, under just the default Content-Domain, to share short descriptions of their current moods. Consider a user James, working at the machine

'jim.activerse.com', who has advertised to others that his
current mood will always be available from the URL

"whodp://mood.activerse.com/james". For the purpose of clarity, full header names will be shown in example traffic, rather than the abbreviations which must actually be used.

James launches his mood-sharing WhoDP agent, which seeks to assert publishing-control over his the authoritative location for his mood:

```
[udp to mood.activerse.com:2222 from jim.activerse.com:2222]
PUB /james W/0.9
Subject: whodp://mood.activerse.com/james
Request-ID: Foo604
Publish-Via: Redirect
Refresh: 20
[end]
```

Understanding this message: This request indicates that James wishes to assert publishing control over the WhoDP object named "whodp://mood.activerse.com/james", at its authoritative location, also "whodp://mood.activerse.com/james". Since no "Reply-To" is specified, the source location of this request is the root location at the originating machine: "whodp://jim.activerse.com/". The type of control requested is for all subscriptions and subscription to be redirected elsewhere. Since no preferred "Location" is specified, the source location of the request, "whodp://jim.activerse.com/", is where redirections are desired. James' user-agent wishes to refresh a granted session once every 20 seconds, in the absence of other activity. Since no content entity is included, James does not want to change any state kept at his home server -- not even temporarily.

(In a real application, this initial PUB request would almost certainly include additional headers authenticating James to his home server, using mechanisms specified elsewhere.)

If this request for publishing-control is granted, a response like the following will ensue:

```
[udp to jim.activerse.com:2222 from mood.activerse.com:2222]
W/0.9 201 Session Created
Subject: whodp://mood.activerse.com/james
Request-ID: Foo604
Session-ID: 90210
Refresh: 60
Content-Type: text/plain

Healthy, wealthy, and wise!
[end]
```

Understanding this message: this response indicates that James'

request for publishing-control has been granted. The Session-ID
to use in future messages relating to this session is "90210".
Barring other activity, James' user-agent should only concern
itself with refreshing the session every 60 seconds -- not the

every 20 it originally suggested. Finally, the current server-side state of the Subject WhoDP object is reported -- perhaps to let James know how he has been seen while absent, or to seed his interface with an initial value for him to edit.

If there is no activity on session "90210" for 60 seconds, James' user-agent should generate keep-alive traffic reasserting his interest in keeping the session open. This would look like:

```
[udp to mood.activerse.com:2222 from jim.activerse.com:2222]
PUB /james W/0.9
Session-ID: 90210
Sequence-Number: 2
[end]
```

Receiving this message would cause the server to mark the publishing-control-session as still active, and generate the following minimal response:

```
[udp to jim.activerse.com:2222 from mood.activerse.com:2222]
W/0.9 200 OK
Session-ID: 90210
Sequence-Number: 2
[end]
```

Now, James' user-agent may turn its attention to retrieving live information on the moods of everyone else James is interested in. If James has registered an interest in the mood of his colleague Susan, with a WhoDP identity of "whodp://whodp.w3.org/susan", his program would generate the following:

```
[udp to whodp.w3.org:2222 from jim.activerse.com:2222]
SUB /susan W/0.9
Subject: whodp://whodp.w3.org/susan
Sender: whodp://mood.activerse.com/james
Request-ID: Bar321
[end]
```

Understanding this message: this request indicates an interest in a continuing subscription to the WhoDP object named "whodp://whodp.w3.org/susan", from the same location. Additionally, the WhoDP identity of the requester is identified. (Again, in a real application, this message could include additional information which would allow the server at whodp.w3.org to verify that the sender is authorized to use the identity "whodp://mood.activerse.com/james".)
If Susan's home server decides to grant this subscription, there will be a response like the following:

```
[udp to jim.activerse.com:2222 from whodp.w3.org:2222]
```

```
W/0.9 201 Subscription Created
Subject: whodp://whodp.w3.org/susan
Request-ID: Bar321
Session-ID: 78705
```

```
        Refresh: 45
        Content-Type: text/plain

        Acceptably jolly.
        [end]
```

Now, let's assume Susan launches her "mood client" from a machine named "dialup99.isp.net". She decides to assert publishing-control with the "Consult" option, and change her home-server state at the same time:

```
        [udp to whodp.w3.org:2222 from dialup99.isp.net:2222]
        PUB /susan W/0.9
        Subject: whodp://whodp.w3.org/susan
        Request-ID: Baz901
        Publish-Via: Consult
        Content-Type: text/plain

        Quasi-jolly.
        [end]
```

If successful, this request will cause three things to happen: a response to Susan confirming her publishing-control-session (not shown); an update on James' subscription reflecting the new state for "whodp://whodp.w3.org/susan"; and an update on Susan's publishing-control-session consulting her what to do with James' subscription. Let's look at the update sent to James first:

```
        [udp to jim.activerse.com:2222 from whodp.w3.org:2222]
        UPD / W/0.9
        Session-ID: 90210
        Sequence-Number: 1
        Content-Type: text/plain

        Quasi-jolly.
        [end]
```

If an alternate "Reply-To" had been specified in James' initial SUB, the Request-URI in this message would be something other than the default "/". James must send a simple confirming response to this message (not shown). Now, because Susan has requested the "Consult" publishing-control option, she will receive an update asking her what to do with any current or new subscriptions. For example:

```
        [udp to dialup99.isp.net:2222 from whodp.w3.org:2222]
        UPD / W/0.9
        Session-ID: 78705
        Sequence-Number: 1
        Subscriber: whodp://mood.activerse.com/james
        Publish-Via: Fulfill Redirect Consult
```

[end]

Susan's response to this message will determine how her home-

server treats James' subscription going forward: "Fulfill" indicates continue to serve the subscription from the server; "Redirect" indicates that a redirection to a given location should occur; "Consult" means fulfill but continue to give notifications about this subscriber (such as when he stops subscribing). Let's assume Susan's current client configuration demands a redirect, so she sends the response:

```
[udp to whodp.w3.org:2222 from dialup99.isp.net:2222]
200 OK
Session-ID: 78705
Sequence-Number: 1
Publish-Via: Redirect
Location: whodp://dialup99.isp.net/mood1
[end]
```

Above we see that Susan's user-agent has specified a non-default redirection target: "whodp://dialup99.isp.net/mood1".

Now, Susan's home-server must cancel James' existing subscription, redirecting him instead to the location Susan's program requested. It sends the following message on James' subscription:

```
[udp to jim.activerse.com:2222 from whodp.w3.org:2222]
UPD / W/0.9
Session-ID: 90210
Sequence-Number: 2
Refresh: 0
Location: whodp://dialup99.isp.net/mood1
[end]
```

James' user-agent must confirm this message with a simple response (not shown), then reattempt subscribing at the suggested location:

```
[udp to dialup99.isp.net:2222 from jim.activerse.com:2222]
SUB /mood1 W/0.9
Subject: whodp://whodp.w3.org/susan
Sender: whodp://mood.activerse.com/james
Request-ID: Joy407
[end]
```

Susan's user-agent would recognize this request for a direct subscription and grant the subscription with an appropriate response(not shown), including a content-entity reflecting Susan's current mood state -- which may be different than that reflected by her home server.

If and when Susan's user-agent attempts to subscribe to James, its initial SUB to "whodp://mood.activerse.com/james" would be

immediately redirected, with a 302 response, to the redirection target of James' publishing-control session: "whodp://jim.activerse.com/".

16. Issues

16.1 This Document

Related protocol efforts like RVP and SGAP should be referenced.

A future revision should describe specific headers and practices for negotiating a mutually acceptable authentication scheme.

For clarity, future revisions should explicitly describe features and formats inherited from HTTP, rather than including them by reference.

Additional HTTP headers and response status codes might be appropriate to bring over, in case they are needed -- especially for the GET and PUT methods.

A number of headers are overloaded, used for disparate purposes, which may be confusing. For example, "Location" is used to tell servers where to send people, to tell people where to go, and to give a permanent new authoritative address (in 301 responses). "Publish-Via" is used both to list publishing options and to choose one option. For clarity, perhaps different uses of these headers should occur under different header names.

The parameters for acceptable UDP retry policies may warrant further tuning for maximum network-friendliness.

Further explanations and examples of the "Proxy" mode are needed, as is further clarification of the role and use of "Content-Domains.

"References" section needs expansion.

16.2 For Future Versions

Should TCP transport be available as an option, and if so, how should UDP & TCP transports be mixed? Should TCP be used only under certain conditions, or as a fallback? Can a single session use both transports? Can a single TCP connection be used for multiple messages -- both requests and responses -- in both directions?

17. Version Notes

This is the first public version of the WhoDP specification. Currently shipping Activerse products use an earlier, undocumented version of WhoDP. Although closely related, that version uses different, sometimes binary request-line, response-line, and header formats, and several different message and header names to achieve similar effects. Also, the only publishing-control option in previous versions is "Relinquish",

which is analogous to "Redirect" here.

18. References

RFC 1123 Dates

RFC 1738 URLs

RFC 2068 HTTP/1.1

PEP

XML

19. Author Contact and Discussion Info

The author may be contacted at:
Gordon Mohr
Activerse, Inc.
1301 W. 25th St
Suite 500
Austin, TX 78705
gojomo@activerse.com

The author recommends discussing WhoDP via the discussion list established for discussing the "Rendezvous Protocol (RVP)" and related issues. For information, send a message with the body "info" to rvp-request@iastate.edu. To subscribe, send a message with the body "subscribe" to rvp-request@iastate.edu.

20. Expiration

This document expires in October 1998.