

WHOIS++ URL Specification

Filename: [draft-ietf-asid-whois-url-02.txt](#)

Status of This Memo

This document is an Internet-Draft. 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.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet-Drafts Shadow Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

Distribution of this memo is unlimited. Editorial comments should be sent directly to the author. Technical discussion will take place on the IETF ASID mailing list - ietf-asid@umich.edu.

This Internet Draft expires September 1998.

Abstract

This document defines a new Uniform Resource Locator (URL) scheme 'whois++', which provides a convention within the URL framework for referring to WHOIS++ servers and the data held within them.

1. Overview of the WHOIS++ protocol

[RFC 1835](#) [1] defines a simple Internet directory protocol known as WHOIS++. In order that WHOIS++ may be used within the Uniform Resource Locator (URL) framework defined by [RFC 1738](#) [2], a URL scheme definition for WHOIS++ is necessary. This document specifies a URL scheme "whois++", for use with the WHOIS++ protocol.

WHOIS++ is text based protocol after the fashion of many popular Internet application protocols, such as SMTP [3] and FTP [4]. Although the protocol is TCP based, WHOIS++ is effectively stateless - no state information is preserved across requests, there is no concept of a session per se since each request/response pair is self-contained, and there is no "login" phase.

WHOIS++ transactions normally consist of a single request from the client and response from the server, followed by the TCP connection between the two being torn down. Use of the "hold" constraint in the WHOIS++ request makes it possible for the client to indicate that it would like to keep the TCP connection open for more than one request/response pair, but whether this is actually done is at the discretion of the server.

2. WHOIS++ URL specification

The following information is necessary for a WHOIS++ client to formulate and deliver a request:

- o the domain name or IP address of the server to contact
- o the port number of the server (63 by default)
- o the request itself - normally a single line of text

This is a good match with the generic URL scheme specified in [RFC 1738](#), and so a URL following the generic syntax is appropriate.

The WHOIS++ URL scheme is defined as:

```
whoisppurl = "whois++://" hostport [ "/" whoisppsrch ]
```

where

```
whoisppsrch = *uchar
```

The definitions for "hostport" and "uchar" are imported from the BNF style grammar for URLs defined in [Section 5 of RFC 1738](#). BNF for the WHOIS++ request format ("whoisppsrch") is defined in [Appendix F of RFC 1835](#).

3. Examples

The whois++ URL scheme defined above makes it possible to write URLs for any of the following:

- (a) a reference particular WHOIS++ server, without implying that a search should be done
- (b) a "canned" search of a particular server

(c) individual objects within a server

Case (a) simply requires that the host and optionally the port number be specified, e.g.

```
whois++://acm.org/
```

or

```
whois++://acm.org:63/
```

When given a WHOIS++ URL of this format, implementations may choose to present the user with a search form or dialogue, contact the server for information about which WHOIS++ options it supports, and so on. The WHOIS++ default port 63 should be used if the port number is not specified.

Case (b) requires a search specification to be present, e.g.

```
whois++://acm.org/name=phil%20and%20name=zimmerman
```

This may be sent verbatim to the server, once hex escaped chars in the URL have been converted back to normal, e.g.

```
name=phil and name=zimmerman
```

Case, (c) is effectively an instance of (b). This may be implemented as a search where the request consists of the WHOIS++ "handle" of the requested object, e.g.

```
whois++://acm.org/handle=number6
```

Although there are no global constraints specified in these last two URLs, the WHOIS++ client may choose to add global constraints of its own, e.g. use of the "hold" constraint to request that the connection be held open for a further request.

If in addition, global constraints are part of the URL, this can easily be recognised by the presence of a colon ":" immediately after the slash "/" which separates the host and port information from the search specifier, e.g.

```
whois++://acm.org/:authenticate=password;name=foo;password=bar
```

At the implementor's discretion, the client may choose to pass these global constraints on in any queries which are passed to this server, e.g. if this URL was used in a search for "zimmerman", the request passed to the server might be either of

zimmerman

or

zimmerman:authenticate=password;name=foo;password=bar

or "zimmerman", followed by some combination of the global constraints specified in the URL and other global constraints introduced by the WHOIS++ client.

4. Issues

4.1 Relationship with WHOIS and RWhois

The three protocols in the WHOIS family, NICNAME/WHOIS [5], WHOIS++, and RWhois [6], are not particularly similar. WHOIS++ and RWhois use different request and response formats, and have different well-known port numbers. WHOIS responses are assumed to be plain text and human readable. Consequently, this document has not attempted to define a single URL scheme for use with all three protocols.

4.2 Localisation

WHOIS++ requests may contain "difficult characters" such as space, and characters drawn from non-ASCII character sets such as the UTF-8 variant of Unicode [7,8]. Hence, the usual rules about hex-escaping illegal and reserved characters should apply - and the definition of the WHOIS++ request as "uchar". Note that the default WHOIS++ port of 63 should be used if the port number component of the "hostport" construction is left out.

4.3 Use of global constraints

Since global constraints such as authentication information, language and character set preferences may be expressed as part of the WHOIS++ request, it is not thought necessary to specify them separately in a mechanism such as the "user@host" construction defined for the FTP URL.

4.4 Encoding multi-line WHOIS++ requests

Most WHOIS++ requests can be expected to consist of a single line of text, followed by carriage return and line feed characters. It should, however, be noted that it may be necessary to encode multi-line requests within WHOIS++ URLs. Software which implements WHOIS++ URLs should either be capable of handling this, or fail gracefully.

4.5 Integration with HTML/HTTP

WHOIS++ URLs may be used as hyperlinks in HTML [9] documents, though it should be noted that the relative URL syntax defined in RFC 1808 [10] is not appropriate for use in these links. This is because WHOIS++ requests do not map conveniently onto the generic resource locator syntax used for relative URLs - the syntactic conventions used in writing a WHOIS++ request are very different from those of the generic resource locator.

The WHOIS++ protocol and the WHOIS++ URL lend themselves to implementation via a proxy HTTP [11] gateway, since the information necessary to contact the server and deliver the request is embedded within the URL itself. A simple proof-of-concept proxy gateway has been implemented which takes an HTTP "GET" request containing a WHOIS++ URL, carries out a WHOIS++ transaction and returns the results formatted as HTML. This may be found at:

```
<URL:http://www.roads.lut.ac.uk/pickup/>
```

It is not appropriate to use any HTTP methods other than "GET" with WHOIS++ URLs.

The appearance of the "+" character in the protocol scheme component of a URL is legal, according to RFC 1738. The author has lingering doubts about the ability of all software which processes URLs, for example in parsing HTML documents, to cope with this character. No evidence has been found to back these doubts up, however.

5. Security Considerations

Client software should check both the contents of the WHOIS++ URL and the results returned from WHOIS++ search requests for any unsafe characters and character strings.

It is possible to embed requests for other protocols within this URL format. This is an approach which may be used to defeat security schemes, spoof protocols, and so on. Implementors should consider requiring user confirmation when requests are directed to reserved ports (i.e. those less than 1024) other than 63 and 43, or well-known ports in the unreserved range.

Implementations should take care not to cache authentication information. In some cases, as with the simple "password" authentication scheme defined in RFC 1835, authentication information may take the form of clear text user names and passwords. This is a WHOIS++ protocol issue and beyond the scope of this URL specification.

6. Acknowledgements

Thanks to Jeff Allen, Lorcan Dempsey, Patrik Faltstrom, Jon Knight, William F. Maton, Larry Masinter, and Scott Williamson for their comments on draft versions of this document.

This work was supported by grant 12/39/01 from the UK Electronic Libraries Programme (eLib) and grant RE 1004 from the European Commission's Telematics for Research Programme.

7. References

Request For Comments (RFC) and Internet Draft documents are available from <URL:ftp://ftp.internic.net> and numerous mirror sites.

- [1] P. Deutsch, R. Schoultz, P. Faltstrom and C. Weider. "Architecture of the WHOIS++ service", [RFC 1835](#). August 1995.
- [2] T. Berners-Lee, L. Masinter and M. McCahill (eds). "Uniform Resource Locators (URL)", [RFC 1738](#). December 1994.
- [3] J. Postel. "Simple Mail Transfer Protocol", [RFC 821](#). August 1982.
- [4] J. Postel, J. K. Reynolds. "File Transfer Protocol", [RFC 959](#). October 1985.
- [5] The Unicode Standard, Worldwide Character Encoding, Version 1.0, Volume 1, Addison-Wesley, 1990. ISBN 0-201-56788-1.
- [6] The Unicode Standard, Worldwide Character Encoding, Version 1.0, Volume 2, Addison-Wesley, 1992. ISBN 0-201-60845-6.
- [7] K. Harrenstien, M.K. Stahl, E.J. Feinler. "NICNAME/WHOIS", [RFC 954](#). October 1985.
- [8] S. Williamson & M. Kusters. "Referral Whois

Protocol (Rwhois)", [RFC 1714](#). November 1994.

- [9] T. Berners-Lee, D. Connolly. "Hypertext Markup Language - 2.0", [RFC 1866](#). November 1995.
- [10] R. Fielding. "Relative Uniform Resource Locators", [RFC 1808](#). June 1995.
- [11] T. Berners-Lee, R. Fielding, H. Frystyk. "Hyper-text Transfer Protocol -- HTTP/1.0", [RFC 1945](#). May 1996.

8. Author's address

Martin Hamilton
Department of Computer Studies
Loughborough University of Technology
Leics. LE11 3TU, UK

Email: m.t.hamilton@lut.ac.uk

This Internet Draft expires September 1998

