

APPSAWG
Internet-Draft
Intended status: BCP
Expires: August 16, 2012

P. Saint-Andre
Cisco Systems, Inc.
D. Crocker
Brandenburg InternetWorking
M. Nottingham
Rackspace
February 13, 2012

Deprecating Use of the "X-" Prefix in Application Protocols draft-ietf-appsawg-xdash-03

Abstract

Historically, designers and implementers of application protocols have often distinguished between "standard" and "non-standard" parameters by prefixing the latter with the string "X-" or similar constructions. In practice, this convention causes more problems than it solves. Therefore, this document deprecates the "X-" convention for textual parameters in application protocols.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on August 16, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | | |
|-----------------------------|---|--------------------|
| 1. | Introduction | 3 |
| 2. | Recommendations for Implementers of Application Protocols . . | 3 |
| 3. | Recommendations for Creators of New Parameters | 3 |
| 4. | Recommendations for Protocol Designers | 4 |
| 5. | Security Considerations | 4 |
| 6. | IANA Considerations | 5 |
| 7. | Acknowledgements | 5 |
| 8. | References | 5 |
| 8.1. | Normative References | 5 |
| 8.2. | Informative References | 5 |
| Appendix A. | Background | 7 |
| Appendix B. | Analysis | 9 |
| | Authors' Addresses | 11 |

1. Introduction

Many application protocols use parameters with textual names to identify data (media types, header fields in Internet mail messages and HTTP requests, vCard parameters and properties, etc.). Historically, designers and implementers of application protocols have often distinguished between "standard" and "non-standard" parameters by prefixing the latter with the string "X-" or similar constructions (e.g., "x."), where the "X" is commonly understood to stand for "eXperimental" or "eXtension".

Although in theory the "X-" convention was a good way to avoid collisions (and attendant interoperability problems) between standard parameters and non-standard parameters, in practice the benefits have been outweighed by the costs associated with the leakage of non-standard parameters into the standards space. Therefore this document deprecates the "X-" convention for named parameters in application protocols and makes specific recommendations about how to proceed in a world without the distinction between standard and non-standard parameters. Note that this document covers only parameters with textual names, not parameters that are expressed as numbers. In addition, this document makes no recommendation as to whether existing "X-" parameters ought to remain in use or be migrated to a format without the "X-".

See [Appendix A](#) for background information about the history of the "X-" convention, and [Appendix B](#) for the reasoning that led to the recommendations in this document.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

2. Recommendations for Implementers of Application Protocols

Implementers of application protocols MUST NOT treat the general categories of "standard" and "non-standard" parameters in programatically different ways within their applications.

3. Recommendations for Creators of New Parameters

Creators of new parameters to be used in the context of application protocols:

1. SHOULD assume that all parameters they create might become standardized, public, commonly deployed, or used across multiple implementations.
2. SHOULD employ meaningful names that they have reason to believe are currently unused (without the "X-" prefix).

Note: If the relevant parameter name space has conventions about associating parameter names with those who create them, a parameter name could incorporate the organization's name or primary domain name (see [Appendix B](#) for examples).

4. Recommendations for Protocol Designers

Designers of new application protocols that allow extensions using parameters:

1. SHOULD establish registries with potentially unlimited value-spaces, if appropriate including both permanent and provisional registries.
2. SHOULD define simple, clear registration procedures.
3. SHOULD mandate registration of all non-private parameters, independent of the form of the parameter names.
4. SHOULD identify a convention to allow local or implementation-specific extensions, and reserve delimiters for such uses as needed.
5. SHOULD NOT prohibit parameters with the "X-" prefix from being registered with the IANA.
6. MUST NOT assume that a parameter with an "X-" prefix is non-standard.
7. MUST NOT assume that a parameter without an "X-" prefix is standard.

5. Security Considerations

Interoperability and migration issues with security-critical parameters can result in unnecessary vulnerabilities (see [Appendix B](#) for further discussion).

6. IANA Considerations

This document does not modify registration procedures currently in force for various application protocols. However, such procedures might be updated in the future to incorporate the best practices defined in this document.

7. Acknowledgements

Thanks to Claudio Allocchio, Adam Barth, Nathaniel Borenstein, Eric Burger, Stuart Cheshire, Al Constanzo, Dave Cridland, Martin Duerst, Frank Ellermann, J.D. Falk, Ned Freed, Tony Finch, Randall Gellens, Tony Hansen, Ted Hardie, Joe Hildebrand, Alfred Hoenes, Paul Hoffman, Eric Johnson, John Klensin, Graham Klyne, Murray Kucherawy, Eliot Lear, John Levine, Bill McQuillan, Alexey Melnikov, Subramanian Moonesamy, Keith Moore, Ben Niven-Jenkins, Zoltan Ordogh, Tim Petch, Dirk Pranke, Randy Presuhn, Julian Reschke, Doug Royer, Andrew Sullivan, Martin Thomson, Matthew Wild, Nicolas Williams, Tim Williams, Mykyta Yevstifeyev, and Kurt Zeilenga for their feedback.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

8.2. Informative References

- [BCP9] Bradner, S., "The Internet Standards Process -- Revision 3", [BCP 9](#), [RFC 2026](#), October 1996.
- [BCP26] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [BCP82] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", [BCP 82](#), [RFC 3692](#), January 2004.
- [RFC691] Harvey, B., "One more try on the FTP", [RFC 691](#), June 1975.
- [RFC737] Harrenstien, K., "FTP extension: XSEN", [RFC 737](#), October 1977.
- [RFC743] Harrenstien, K., "FTP extension: XRSQ/XRCP", [RFC 743](#), December 1977.

- [RFC775] Mankins, D., Franklin, D., and A. Owen, "Directory oriented FTP commands", [RFC 775](#), December 1980.
- [RFC822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, [RFC 822](#), August 1982.
- [RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, [RFC 1123](#), October 1989.
- [RFC1154] Robinson, D. and R. Ullmann, "Encoding header field for internet messages", [RFC 1154](#), April 1990.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", [RFC 2045](#), November 1996.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), November 1996.
- [RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", [RFC 2047](#), November 1996.
- [RFC2068] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2068](#), January 1997.
- [RFC2426] Dawson, F. and T. Howes, "vCard MIME Directory Profile", [RFC 2426](#), September 1998.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [RFC2822] Resnick, P., "Internet Message Format", [RFC 2822](#), April 2001.
- [RFC2939] Droms, R., "Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types", [BCP 43](#), [RFC 2939](#), September 2000.
- [RFC3406] Daigle, L., van Gulik, D., Iannella, R., and P. Faltstrom, "Uniform Resource Names (URN) Namespace Definition Mechanisms", [BCP 66](#), [RFC 3406](#), October 2002.
- [RFC3427] Mankin, A., Bradner, S., Mahy, R., Willis, D., Ott, J., and B. Rosen, "Change Process for the Session Initiation

Protocol (SIP)", [RFC 3427](#), December 2002.

- [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", [BCP 90](#), [RFC 3864](#), September 2004.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", [RFC 4122](#), July 2005.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 4288](#), December 2005.
- [RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", [BCP 35](#), [RFC 4395](#), February 2006.
- [RFC4512] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Directory Information Models", [RFC 4512](#), June 2006.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC5064] Duerst, M., "The Archived-At Message Header Field", [RFC 5064](#), December 2007.
- [RFC5451] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", [RFC 5451](#), April 2009.
- [RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", [BCP 47](#), [RFC 5646](#), September 2009.
- [RFC5727] Peterson, J., Jennings, C., and R. Sparks, "Change Process for the Session Initiation Protocol (SIP) and the Real-time Applications and Infrastructure Area", [BCP 67](#), [RFC 5727](#), March 2010.

[Appendix A](#). Background

The beginnings of the "X-" convention can be found in a suggestion made by Brian Harvey in 1975 with regard to FTP parameters [[RFC691](#)]:

Thus, FTP servers which care about the distinction between Telnet print and non-print could implement SRVR N and SRVR T. Ideally the SRVR parameters should be registered with Jon Postel to avoid conflicts, although it is not a disaster if two sites use the same parameter for different things. I suggest that parameters be allowed to be more than one letter, and that an initial letter X be used for really local idiosyncracies.

This "X" prefix was subsequently used in [\[RFC737\]](#), [\[RFC743\]](#), and [\[RFC775\]](#). This usage was noted in [\[RFC1123\]](#):

FTP allows "experimental" commands, whose names begin with "X". If these commands are subsequently adopted as standards, there may still be existing implementations using the "X" form.... All FTP implementations SHOULD recognize both forms of these commands, by simply equating them with extra entries in the command lookup table.

The "X-" convention has been used for email header fields since at least the publication of [\[RFC822\]](#) in 1982, which distinguished between "Extension-fields" and "user-defined-fields" as follows:

The prefatory string "X-" will never be used in the names of Extension-fields. This provides user-defined fields with a protected set of names.

That rule was restated by [\[RFC1154\]](#) as follows:

Keywords beginning with "X-" are permanently reserved to implementation-specific use. No standard registered encoding keyword will ever begin with "X-".

This convention continued with various specifications for media types ([\[RFC2045\]](#), [\[RFC2046\]](#), [\[RFC2047\]](#)), HTTP headers ([\[RFC2068\]](#), [\[RFC2616\]](#)), vCard parameters and properties ([\[RFC2426\]](#)), Uniform Resource Names ([\[RFC3406\]](#)), LDAP field names ([\[RFC4512\]](#)), and other application technologies.

However, use of the "X-" prefix in email headers was effectively deprecated between the publication of [\[RFC822\]](#) in 1982 and the publication of [\[RFC2822\]](#) in 2001 by removing the distinction between the "extension-field" construct and the "user-defined-field" construct (a similar change happened with regard to Session Initiation Protocol "P-" headers when [\[RFC3427\]](#) was obsoleted by [\[RFC5727\]](#)).

Despite the fact that parameters containing the "X-" string have been effectively deprecated in email headers, they continue to be used in

a wide variety of application protocols. The two primary situations motivating such use are:

1. Experiments that are intended to possibly be standardized in the future, if they are successful.
2. Extensions that are intended to never be standardized because they are intended only for implementation-specific use or for local use on private networks.

Use of this naming convention is not mandated by the Internet Standards Process [[BCP9](#)] or IANA registration rules [[BCP26](#)]. Rather it is an individual choice by each specification that references the convention or each administrative process that chooses to use it. In particular, some standards-track RFCs have interpreted the convention in a normative way (e.g., [[RFC822](#)] and [[RFC5451](#)]).

[Appendix B](#). Analysis

The primary problem with the "X-" convention is that non-standard parameters have a tendency to leak into the protected space of standard parameters (whether de jure or de facto), thus introducing the need for migration from the "X-" name to the standard name. Migration, in turn, introduces interoperability issues (and sometimes security issues) because older implementations will support only the "X-" name and newer implementations might support only the standard name. To preserve interoperability, newer implementations simply support the "X-" name forever, which means that the non-standard name has become a de facto standard (thus obviating the need for segregation of the name space into "standard" and "non-standard" areas in the first place).

We have already seen this phenomenon at work with regard to FTP in the quote from [[RFC1123](#)] in the previous section. The HTTP community had the same experience with the "x-gzip" and "x-compressed" media types, as noted in [[RFC2068](#)]:

For compatibility with previous implementations of HTTP, applications should consider "x-gzip" and "x-compress" to be equivalent to "gzip" and "compress" respectively.

A similar example can be found in [[RFC5064](#)], which defined the "Archived-At" message header field but also found it necessary to define and register the "X-Archived-At" field:

For backwards compatibility, this document also describes the X-Archived-At header field, a precursor of the Archived-At header field. The X-Archived-At header field MAY also be parsed, but SHOULD NOT be generated.

One of the original reasons for segregation of name spaces into standard and non-standard areas was the perceived difficulty of registering names. However, the solution to that problem has been simpler registration rules, such as those provided by [[RFC3864](#)] and [[RFC4288](#)]. As explained in [[RFC4288](#)]:

[W]ith the simplified registration procedures described above for vendor and personal trees, it should rarely, if ever, be necessary to use unregistered experimental types. Therefore, use of both "x-" and "x." forms is discouraged.

For some name spaces, another helpful practice has been the establishment of separate registries for permanent names and provisional names, as in [[RFC4395](#)].

Furthermore, often standardization of a non-standard parameter or protocol element leads to subtly different behavior (e.g., the standard version might have different security properties as a result of security review provided during the standardization process). If implementers treat the old, non-standard parameter and the new, standard parameter as equivalent, interoperability and security problems can ensue.

For similar considerations with regard to the "P-" convention in the Session Initiation Protocol, see [[RFC5727](#)].

In some situations, segregating the parameter name space used in a given application protocol can be justified:

1. When it is extremely unlikely that some parameters will ever be standardized. However, in this case implementation-specific and private-use parameters could at least incorporate the organization's name (e.g., "ExampleInc-foo" or, consistent with [[RFC4288](#)], "VND.ExampleInc.foo") or primary domain name (e.g., "com.example.foo" or a Uniform Resource Identifier [[RFC3986](#)] such as "http://example.com/foo"). In rare cases, truly experimental parameters could be given meaningless names such as nonsense words, the output of a hash function, or UUIDs [[RFC4122](#)].
2. When parameter names might have significant meaning. However, this case too is rare, since implementers can almost always find a synonym for an existing term (e.g., "urgency" instead of "priority") or simply invent a more creative name (e.g., "get-it-

there-fast").

3. When parameter names need to be very short (e.g., as in [\[RFC5646\]](#) for language tags). However, in this case it can be more efficient to assign numbers instead of human-readable names (e.g., as in [\[RFC2939\]](#) for DHCP options) and to leave a certain numeric range for implementation-specific extensions or private use (e.g., as with the codec numbers used with the Session Description Protocol [\[RFC4566\]](#)).

There are three primary objections to deprecating the "X-" convention as a best practice for application protocols:

1. Implementers are easily confused and can't be expected to know that a parameter is non-standard unless it contains the "X-" prefix. However, implementers already are quite flexible about using both prefixed and non-prefixed names based on what works in the field, so the distinction between de facto names (e.g., "X-foo") and de jure names (e.g., "foo") is without force.
2. Collisions are undesirable and it would be bad for both a standard parameter "foo" and a non-standard parameter "foo" to exist simultaneously. However, names are almost always cheap, so an experimental, implementation-specific, or private-use name of "foo" does not prevent a standards development organization from issuing a similarly creative name such as "bar".
3. [\[BCP82\]](#) is entitled "Assigning Experimental and Testing Numbers Considered Useful" and therefore implies that the "X-" prefix is also useful for experimental parameters. However, [BCP 82](#) addresses the need for protocol numbers when the pool of such numbers is strictly limited (e.g., DHCP options) or when a number is absolutely required even for purely experimental purposes (e.g., the Protocol field of the IP header). In almost all application protocols that make use of protocol parameters (including email headers, media types, HTTP headers, vCard parameters and properties, URNs, and LDAP field names), the name space is not limited or constrained in any way, so there is no need to assign a block of names for private use or experimental purposes (see also [\[BCP26\]](#)).

Therefore it appears that segregating the parameter space into a standard area and a non-standard area has few if any benefits, and has at least one significant cost in terms of interoperability.

Authors' Addresses

Peter Saint-Andre
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Phone: +1-303-308-3282
Email: psaintan@cisco.com

D. Crocker
Brandenburg InternetWorking
675 Spruce Dr.
Sunnyvale
USA

Phone: +1.408.246.8253
Email: dcrocker@bbiw.net
URI: <http://bbiw.net>

Mark Nottingham
Rackspace

Email: mnot@mnot.net
URI: <http://www.mnot.net>

