Deprecating the X- Prefix and Similar Constructs in Application Protocols
draft-ietf-appsawg-xdash-05

Abstract

   Historically, designers and implementers of application protocols
   have often distinguished between standardized and unstandardized
   parameters by prefixing the names of unstandardized parameters with
   the string "X-" or similar constructs. In practice, that convention
   causes more problems than it solves. Therefore, this document
   deprecates the convention for newly-defined parameters with textual
   (as opposed to numerical) names 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 October 11, 2012.

Copyright Notice

Table of Contents

1. Introduction

   Many application protocols use parameters with textual (as opposed to
   numerical) 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 standardized
   and unstandardized parameters by prefixing the names of
   unstandardized parameters with the string "X-" or similar constructs
   (e.g., "x."), where the "X" is commonly understood to stand for
   "eXperimental" or "eXtension".

   Under this convention, the name of a parameter not only identified
   the data, but also embedded the status of the parameter into the name
   itself: a parameter defined in a specification produced by a
   recognized standards development organization (or registered
   according to processes defined in such a specification) did not start
   with "X-" or similar constructs, whereas a parameter defined outside
   such a specification or process started with "X-" or similar
   constructs.

   As explained more fully under Appendix A, this convention was
   encouraged for many years in application protocols such as file
   transfer, email, and the World Wide Web. In particular, it was
   codified for email by [RFC822] (via the distinction between
   "Extension-fields" and "user-defined-fields"), but then removed by
   [RFC2822] based on implementation and deployment experience. A
   similar progression occurred for SIP technologies with regard to the
   "P-" header, as explained in [RFC5727]. The reasoning behind those
   changes is explored under Appendix B.

   In short, although in theory the "X-" convention was a good way to
   avoid collisions (and attendant interoperability problems) between
   standardized parameters and unstandardized parameters, in practice
   the benefits have been outweighed by the costs associated with the
   leakage of unstandardized parameters into the standards space.

   This document generalizes from the experience of the email and SIP
   communities by doing the following:

   1. Deprecates the "X-" convention for newly-defined parameters in
       application protocols, even where that convention was only
       implicit instead of being codified in a protocol specification
       (as was done for email in [RFC822]).

   2. Makes specific recommendations about how to proceed in a world
       without the distinction between standardized and unstandardized
       parameters (although only for parameters with textual names, not

parameters that are expressed as numbers, which are out of
scope).

3. Does not recommend against the practice of private, local,
    preliminary, experimental, or implementation-specific parameters,
    only against the use of "X-" and similar constructs in the names
    of such parameters.

4. Makes no recommendation as to whether existing "X-" parameters
    ought to remain in use or be migrated to a format without the
    "X-"; this is a matter for the creators or maintainers of those
    parameters.

5. Does not override existing specifications that legislate the use
    of "X-" for particular application protocols (e.g., the "x-name"
    token in [RFC5545]); this is a matter for the designers of those
    protocols.

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**

Implementations of application protocols MUST NOT make any
assumptions about the status of a parameter, nor take automatic
action regarding a parameter, based solely on the presence or absence
of "X-" or a similar construct in the parameter's name.

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 parameter names that they have reason to
    believe are currently unused.

3. SHOULD NOT prefix their parameter names with "X-" or similar
    constructs.

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 defining 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 NOT prohibit parameters with the "X-" prefix or similar
    constructs from being registered.

5. MUST NOT assume that a parameter with an "X-" prefix or similar
    constructs is unstandardized.

6. MUST NOT assume that a parameter without an "X-" prefix or
    similar constructs is standard.

5. Security Considerations

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

As a corollary to the recommendation provided under Section 2,
implementations MUST NOT assume that standardized parameters are
"secure" whereas unstandardized parameters are "insecure", based
solely on the names of such parameters.

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, Ralph Droms,
   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, Scott Kelly, Scott Kitterman,
   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, Dan Romascanu, Doug Royer, Andrew
   Sullivan, Henry Thompson, 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.

   [RFC5545] Desruisseaux, B., "Internet Calendaring and Scheduling
              Core Object Specification (iCalendar)", RFC 5545,
              September 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 unstandardized
parameters have a tendency to leak into the protected space of
standardized parameters, thus introducing the need for migration from
the "X-" name to a standardized 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 standardized name. To
preserve interoperability, newer implementations simply support the
"X-" name forever, which means that the unstandardized name has
become a de facto standard (thus obviating the need for segregation
of the name space into standardized and unstandardized 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-compress" 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
   standardized and unstandardized 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 unstandardized parameter
   leads to subtly different behavior (e.g., the standardized version
   might have different security properties as a result of security
   review provided during the standardization process). If implementers
   treat the old, unstandardized parameter and the new, standardized
   parameter as equivalent, interoperability and security problems can
   ensue. Analysis of unstandardized parameters to detect and correct
   flaws is in general a good thing and is not intended to be
   discouraged by the lack of distinction in element names. Whenever an
   originally unstandardized parameter or protocol element is
   standardized and the new form has differences which affect
   interoperability or security properties, implementations MUST NOT
   treat the old form as identical to the new form.

   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. 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. 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").
    The existence of multiple similarly-named paramaters can be
    confusing, but this is true regardless if there is an attempt to
    segregate standardized and unstandardized (e.g., "X-Priority" can
    be confused with "Urgency").

3. When parameter names need to be very short (e.g., as in [RFC5646]
    for language tags). In this case it can be more efficient to
    assign numbers instead of human-readable names (e.g., as in
    [RFC2939] for DCHP 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 might mistake one parameter for another parameter
    that has a similar name; a rigid distinction such as an "X-"
    prefix can make this clear. However, in practice implementers
    are forced to blur the distinction (e.g., by treating "X-foo" as
    a de facto standard) and so it inevitably becomes meaningless.

2. Collisions are undesirable and it would be bad for both a
    standardized parameter "foo" and a unstandardized 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
   standardized area and a unstandardized 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