Use of the "X-" Prefix in Application Protocols
draft-saintandre-xdash-00

Abstract

Many application protocols use named parameters to identity data. Historically, protocol designers and implementers distinguished between "standard" and "non-standard" parameters by prefixing the latter with the string "X-". On balance, this "X-" convention has more costs than benefits, although it can be appropriate in certain circumstances.

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 December 29, 2011.

Copyright Notice

Copyright (c) 2011 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. Background ...................................................... 3
2. Analysis ......................................................... 3
3. Security Considerations ......................................... 6
4. IANA Considerations ............................................. 6
5. Acknowledgements ............................................... 6
6. Informative References ......................................... 6
Author’s Address .................................................... 8
1. Background

Many application protocols use named parameters to identity data (media types, header fields in Internet mail messages and HTTP requests, etc.). Historically, protocol designers and implementers have often distinguished between "standard" and "non-standard" parameters by prefixing the latter with the string "X-", where the "X" stands for "eXperimental".

This "X-" convention has been uses for email header fields at least since 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]), email headers ([RFC2821], [RFC5321]), HTTP headers ([RFC2068], [RFC2616]), Uniform Resource Names ([RFC3406]), Session Initiation Protocol "P-" headers ([RFC3427], obsoleted by [RFC5727]), and other technologies.

Parameters prefaced with the "X-" string (and similar constructions, such as "x.") are currently used in application protocols for two different purposes:

- Experiments that might lead to standardization in the future.
- Implementation-specific applications or private networks that are never intended to be standardized.

The remainder of this document analyzes the benefits and costs of the "X-" convention and specifies when it is appropriate to apply the convention in application protocols produced by the IETF.

2. Analysis

The primary problem with the "X-" convention is that non-standard parameters have a tendency to leak into the protected space of standardized parameters (whether de jure or de facto), thus introducing the need for migration from the "X-" name to the
standardized name. Migration, in turn, introduces interoperability 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 non-standard name becomes a de facto standard (thus obviating the need for segregation of the name spaces in the first place). As one example, we can see this phenomenon at work in [RFC2068] (similar examples can be found in [RFC5064]):

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

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 well as separate registries for permanent and provisional names.

[RFC4288] calls out one implication of non-standard names:

> With 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.

Furthermore, often standardization of a non-standard parameter or protocol element 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, 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 name space of parameters 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, private-use parameters can be URIs (e.g., "http://example.com/foo") or can be prepended with a string that is derived from the name or primary domain name of the organization that has defined the parameter (e.g., "Example-Foo" or "com.example.foo"). Similarly, truly experimental
parameters can be given meaningless names such as UUIDs [RFC4122].

2. When parameter names might have significant meaning. This case is rare, since implementers can almost always find a synonym (e.g., "urgency" instead of "priority") or simply invent a new name.

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 DHCP options) and to leave a certain numeric range for private use (e.g., as with the codec numbers used with the Session Description Protocol [RFC4566]).

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

- Implementers are easily confused. 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 meaningless to them.

- Collisions are undesirable. However, names are almost always cheap, so an experimental or implementation-specific name of "foo" does not prevent a standards development organization from issuing a similarly creative name such as "bar".

In addition, the existence of [BCP82] ("Assigning Experimental and Testing Numbers Considered Useful") might appear to provide an argument against deprecating the "X-" convention. However, BCP 82 addresses the need for protocols 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 (e.g., media types, email headers, HTTP headers, URIs), 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]).

The foregoing considerations lead to the conclusion that segregating non-standard parameters into an "X-" ghetto has few if any benefits, and has at least one significant cost in terms of interoperability. Therefore, this document recommends against the creation of new names with the special "X-" prefix in application protocols produced within the IETF.
3. Security Considerations

Interoperability and migration issues with security-critical parameters can result in unnecessary vulnerabilities.

4. IANA Considerations

This document requests no action by the IANA.

5. Acknowledgements

Thanks to Claudio Allocchio, Adam Barth, Nathaniel Borenstein, Eric Burger, Al Constanzo, Dave Cridland, Dave Crocker, Martin Duerst, J.D. Falk, Tony Finch, Tony Hansen, Ted Hardie, Joe Hildebrand, Alfred Hoenes, Paul Hoffman, Eric Johnson, John Klensin, Graham Klyne, Murray Kucherawy, Eliot Lear, Bill McQuillan, Alexey Melnikov, Subramanian Moonesamy, Keith Moore, Mark Nottingham, Randy Presuhn, Julian Reschke, Doug Royer, Andrew Sullivan, Martin Thomson, Nicolas Williams, and Kurt Zeilenga for feedback.

6. Informative References


RFC 2047, November 1996.


Author’s Address

Peter Saint-Andre
Cisco
1899 Wyknoop Street, Suite 600
Denver, CO 80202
USA

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