

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 3 October 2026

M. Ertl, Ed.
Ertl Software Productions
1 April 2026

UTF-8 Internet Protocol Addresses
draft-mer1l-afipv8-00

Abstract

This memo describes an alternate entry and display format for IPv6 addresses using UTF-8 instead of hextets. A mixed representation for traditional IPv6 address prefixes also is explored, along with means of transitioning between the two within an IPv6 address.

Additionally, an address extension is proposed to accommodate the inevitable need for longer addresses.

Finally, security implications of special UTF-8 glyphs are considered.

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 <https://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 3 October 2026.

Copyright Notice

Copyright (c) 2026 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	2
2. New display and entry format for IPv6 addresses	3
3. Proposal for address extension	5
4. IANA Considerations	5
5. Security Considerations	6
6. References	6
6.1. Normative References	6
6.2. Informative References	7
Acknowledgements	7
Author's Address	7

1. Introduction

The adoption and deployment of IPv6 has been slow due to several factors, including the perceived unwieldiness of the addresses themselves and the lack of memorable constants. Hextet notation doesn't lend itself well to memorization (especially if the entire length needs to be memorized) and, while better in this respect, there still are only a handful of recognizable, catchy constants like "dead", "beer", "face", etc. . Even resorting to "Leetspeak" (1337) will yield only a comparatively small pool of choices. Therefore, the most powerful drive for adoption, vanity, isn't readily applicable. This document aims to change this by presenting a way for organizations and individuals to express themselves within IPv6 addresses, which is impossible to do with IPv4 and therefore is expected to become a deciding factor in IPv6 adoption.

1.1. Requirements Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. New display and entry format for IPv6 addresses

Instead of the current hextet format, separated by colons, IPv6 addresses as defined by [RFC8200] MAY be displayed and entered as a sequence of UTF-8 characters as defined in [RFC3629]. These lend themselves well to the underlying octet format while placing no limitations on the characters.

Mixed representation MAY be used and is especially relevant to conventional prefixes that have been defined without this new representation in mind. Representations MUST NOT change within octets. Transitions MUST be delimited by a colon. Mixed notation MUST NOT transition to UTF-8 using a valid hexadecimal character (a-f, A-F, 0-9).

The new UTF-8 representation uses the tabulator (U+0009) as native replacement for the double colon used in IPv6. As with the double colon, a UTF-8 IP address MUST NOT contain more than one tabulator. It also MUST NOT contain both a tabulator and a double colon.

To avoid confusion, the displaying software SHOULD visually distinguish the tabulator from common whitespace. Likewise, the displaying software SHOULD visually distinguish traditional and nontraditional parts of an IPv6 address, since it will be common that characters from the hexadecimal range appear in the new notation. Also see Section 5 for security implications of possible usage of special formatting glyphs.

If the slash prefix notation "[/bits]" is used, there MUST be a transition to hextet notation immediately preceding the slash.

Even though it is NOT RECOMMENDED, the mixed representation allows transitions transition to / from UTF-8 after odd-numbered octets. If a double colon or tabulator is used with such a separation, all remaining octets MUST be spelt out fully (even those that would be compacted to a single "0" in traditional IPv6). Basic IPv6 colon separation and explosion rules MUST NOT be applied in this case.

The three characters: colon (:), slash (/) and tabulator (U+0009) are reserved and MUST NOT appear as part of the address itself.

Examples

- * IETF server:/128 (tabulator and mandatory transition before slash)
4945:5446::7365:7276:6572/128 (traditional notation of the above)
- * RFC Editor:/128 (tabulator and mandatory transition before slash)
5246:4300::4564:6974:6F72/128 (traditional notation of the above)

- * Campus dorm :/64 (tabulator and mandatory transition before slash)
4361:6D70:7573::646F:726D:2020/64 (traditional notation of the above)
- * This is an IPv6!:/128 (mandatory transition before slash)
5468:6973:2069:7320:616E:2049:5076:3621/128 (traditional notation of the above)

Examples for mixed addresses

- * 2001:db80:Peace please:/64 (one transition and mandatory transition before slash)
2001:db80:5065:6163:6520:706C:6561:7365/64 (traditional notation of the above)
- * 2001:db8::IETF-room _ :/64 (one transition and mandatory transition before slash)
2001:db80:4945:5446:2D72:6F6F:6D20:5F20/64 (traditional notation of the above)
- * 2001:db8::corp-servers:/64 (one transition and mandatory transition before slash)
2001:db80:636F:7270:2D73:6572:7665:7273/64 (traditional notation of the above)
- * 2001:db80:land:0123::3/64 (two transitions, double colon and no mandatory transition)
2001:db80:6C61:6E64:0123::3/64 (traditional notation of the above)
- * 2001:db80:land 0123::/64 (two transitions, including double colon and no mandatory transition)
2001:db80:6C61:6E64:2030:3132:3300::/64 (traditional notation of the above)
- * 2001:db80:trees:0123: :/64 (three transitions, tabulator and mandatory transition before slash)
2001:db80:7472:6565:7301:2300::/64 (traditional notation of the above)
- * 2001:db80:trees:0123::/64 (equivalent to above, two transitions and no mandatory transition)
2001:db80:7472:6565:7310:2300::/64 (traditional notation of the above)
- * 2001:db80:trees0:1230::/64 (not equivalent to above, two transitions and no mandatory transition)
2001:db80:7472:6565:7330:1230::/64 (traditional notation of the above)

- * 2001:db80::database:/64 (one transition and mandatory transition before slash)
2001:db80::6461:7461:6261:7365/64 (traditional notation of the above)
- * 2001:db80::Cameras:0/80 (not acceptable due to ambiguity)
2001:db80:0000:00:Cameras:0/80 (ambiguity removed, acceptable)
2001:db80::43:616D:6572:6173:0/80 (traditional notation of the above)

As seen above, it is often feasible to encode the entire hostname in the host part of the IPv6. However, the reverse translation into mixed notation cannot easily be done without additional user input, therefore it is RECOMMENDED to have the position of the transition to UTF-8 be defined by the prefix length, and also it is NOT RECOMMENDED to transition back from UTF-8 at all except for the the mandatory transition at the slash.

If the prefix length is used in this manner, automated reversal would be facilitated, which would make it significantly more easy to administer static IP addresses, oversee dynamic address leases as well as analyze log files, where the absence of meaningful UTF-8 representation may by itself be a clue to an issue. Essentially, the translation of a more or less arbitrary hextet sequence to any specific machine name would become largely unnecessary once the hostname *is* the IP address. This in turn has the potential of streamlining the workflow and thereby reduce both manual labour and opportunity for error.

3. Proposal for address extension

By transitioning to UTF-8 addressing it becomes immediately obvious that the current size of an IPv6 address is insufficient. There will undoubtedly be desire to use organization names, catchy phrases, or mottos for public-facing hosts, for the prefix, or both. It is therefore proposed to increase the address size by a factor of four.

When doing so, notation should be restricted to UTF-8 only, as the resulting longer addresses cannot be rendered in a concise way in hextet notation.

With this extension, the current DNS system would become obsolete, simplifying the internet architecture and removing vulnerable services.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

If the proposed address extension is implemented, the DNS system, both local and global, would become obsolete and all associated vulnerabilities and risks with it. Additionally, the direct representation of hostnames as their IP addresses would likely benefit security due to reduced risk of misidentification or misrepresentation. Care must be taken, however, with international UTF-8 glyphs that look similar to those used in a particular instance, just like in internationalized domain names (IDN), [RFC5890]. Additionally, glyphs that facilitate hiding of characters, like in the GlassWorm exploit, need to be taken into account. Finally, the tabulator may also look similar to a common whitespace, so fake addresses could be constructed to mimic official addresses exploiting this.

While hiding glyphs SHOULD simply be disallowed, the other glyphs may have legitimate purposes and cannot be universally restricted. Instead, the displaying software needs to visually distinguish them in an obvious but non-intrusive way so that the risk for misidentification is at least reduced.

The displaying software MUST ensure that all characters and glyphs that make up an address are clearly and immediately identifiable, including formatting.

The handling of special glyphs except for the tabulator by security software and devices can likely be heavy-handed because it is expected that these will be used either only internal to a specific network, or not at all, because entering them will be difficult and thus undesirable in any foreign script. External or global use will thus be restricted to fraud and forgery.

As a general rule, glyphs that are not native to the respective script of a network or organization should be visually distinguished. If a glyph or character cannot be rendered, it MUST be replaced by a clear indicator, ideally identifying the offending character or glyph by other means.

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/info/rfc3629>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

6.2. Informative References

- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <<https://www.rfc-editor.org/info/rfc5890>>.

Acknowledgements

This template uses extracts from templates written by Pekka Savola, Elwyn Davies and Henrik Levkowetz.

Author's Address

Marcus Ertl (editor)
Ertl Software Productions
An Peschbenden 13
47906 Kempen
Germany
Phone: +49 2152 53286
Email: marcus.ertl@gmx.net