

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: September 27, 2009

S. Kawamura  
NEC BIGLOBE, Ltd.  
M. Kawashima  
NEC AccessTechnica, Ltd.  
March 26, 2009

A Recommendation for IPv6 Address Text Representation  
draft-kawamura-ipv6-text-representation-00

## Status of this Memo

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

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on September 27, 2009.

## Copyright Notice

Copyright (c) 2009 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 in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Abstract

As IPv6 network grows, there will be more engineers and also non-engineers who will have the need to use an IPv6 address in text.

Internet-Draft

IPv6 Text Representation

March 2009

While the IPv6 address architecture [\[RFC4291\] section 2.2](#) depicts a flexible model for text representation of an IPv6 address, this flexibility has been causing problems for operators, system engineers, and customers. The following draft will describe the problems that a flexible text representation has been causing. This document also recommends a text representation method that best avoids confusion.

Internet-Draft

IPv6 Text Representation

March 2009

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">4</a>
<a href="#">1.1.</a>	<a href="#">Requirements Language . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Text representation flexibility of <a href="#">RFC4291</a> . . . . .</a>	<a href="#">4</a>
<a href="#">2.1.</a>	<a href="#">leading zeros . . . . .</a>	<a href="#">4</a>
<a href="#">2.2.</a>	<a href="#">zero compression . . . . .</a>	<a href="#">5</a>
<a href="#">2.3.</a>	<a href="#">Uppercase or Lowercase . . . . .</a>	<a href="#">6</a>
<a href="#">3.</a>	<a href="#">Problems Encountered with the Flexible Model . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.</a>	<a href="#">Searching . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.1.</a>	<a href="#">General Summary . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.2.</a>	<a href="#">Searching Spreadsheets and Text Files . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.3.</a>	<a href="#">Searching with Whois . . . . .</a>	<a href="#">7</a>
<a href="#">3.1.4.</a>	<a href="#">Searching for an Address in a Network Diagram . . . . .</a>	<a href="#">7</a>
<a href="#">3.2.</a>	<a href="#">Parsing and Modifying . . . . .</a>	<a href="#">7</a>
<a href="#">3.2.1.</a>	<a href="#">General Summary . . . . .</a>	<a href="#">7</a>
<a href="#">3.2.2.</a>	<a href="#">Logging . . . . .</a>	<a href="#">7</a>
<a href="#">3.2.3.</a>	<a href="#">Auditing. Case 1 . . . . .</a>	<a href="#">8</a>
<a href="#">3.2.4.</a>	<a href="#">Auditing. Case 2 . . . . .</a>	<a href="#">8</a>
<a href="#">3.2.5.</a>	<a href="#">Unexpected Modifying . . . . .</a>	<a href="#">8</a>
<a href="#">3.3.</a>	<a href="#">Operating . . . . .</a>	<a href="#">8</a>
<a href="#">3.3.1.</a>	<a href="#">General Summary . . . . .</a>	<a href="#">8</a>
<a href="#">3.3.2.</a>	<a href="#">Customer Calls . . . . .</a>	<a href="#">9</a>
<a href="#">3.3.3.</a>	<a href="#">Abuse . . . . .</a>	<a href="#">9</a>
<a href="#">3.4.</a>	<a href="#">Other Minor Problems . . . . .</a>	<a href="#">9</a>
<a href="#">3.4.1.</a>	<a href="#">Changing Platforms . . . . .</a>	<a href="#">9</a>
<a href="#">3.4.2.</a>	<a href="#">Preference in Documentation . . . . .</a>	<a href="#">9</a>
<a href="#">3.4.3.</a>	<a href="#">Legibility . . . . .</a>	<a href="#">9</a>
<a href="#">4.</a>	<a href="#">A Recommendation for IPv6 Text Representation . . . . .</a>	<a href="#">10</a>
<a href="#">4.1.</a>	<a href="#">Handling Leading Zeros . . . . .</a>	<a href="#">10</a>
<a href="#">4.2.</a>	<a href="#">Lower Case . . . . .</a>	<a href="#">10</a>
<a href="#">4.3.</a>	<a href="#">"::" usage . . . . .</a>	<a href="#">10</a>
<a href="#">4.3.1.</a>	<a href="#">shorten as much as possible . . . . .</a>	<a href="#">10</a>
<a href="#">4.3.2.</a>	<a href="#">one 16bit 0 field . . . . .</a>	<a href="#">10</a>
<a href="#">4.3.3.</a>	<a href="#">when "::" can be used twice . . . . .</a>	<a href="#">10</a>
<a href="#">5.</a>	<a href="#">Conclusion . . . . .</a>	<a href="#">10</a>

<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">11</a>
<a href="#">7.</a>	IANA Considerations . . . . .	<a href="#">11</a>
<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">11</a>
<a href="#">9.</a>	References . . . . .	<a href="#">11</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">11</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">11</a>
	Authors' Addresses . . . . .	<a href="#">11</a>

## [1.](#) Introduction

A single IPv6 address can be text represented in many ways. Examples are shown below.

2001:db8:0:0:1:0:0:1/128

2001:0db8:0:0:1:0:0:1/128

2001:db8::1:0:0:1/128

2001:db8::0:1:0:0:1/128

2001:0db8::1:0:0:1/128

2001:db8:0:0:1::1/128

2001:db8:0000:0:1::1/128

2001:DB8:0:0:1::1/128

All the above point to the same IPv6 address. This flexibility has caused many problems for operators, systems engineers, and customers. The problems will be noted in [section 3](#).

### [1.1.](#) Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

## [2.](#) Text representation flexibility of [RFC4291](#)

Examples of flexibility in [Section 2.2 of RFC4291](#) are described below.

### [2.1.](#) leading zeros

'It is not necessary to write the leading zeros in an individual field.'

In other words, it is also not necessary to omit leading zeros. This means that, it is possible to select such as the following example. Last 16bit is different, but all these addresses are the same.

2001:db8:aaaa:bbbb:cccc:dddd:eeee:0001/128

2001:db8:aaaa:bbbb:cccc:dddd:eeee:001/128

2001:db8:aaaa:bbbb:cccc:dddd:eeee:01/128

2001:db8:aaaa:bbbb:cccc:dddd:eeee:1/128

### [2.2.](#) zero compression

'A special syntax is available to compress the zeros. The use of "::" indicates one or more groups of 16 bits of zeros.'

It is possible to select whether or not to omit just one 16bits of zeros.

2001:db8:aaaa:bbbb:cccc:dddd::1/128

2001:db8:aaaa:bbbb:cccc:dddd:0:1/128

In case where there are more than one zero fields, there is a choice of how much fields to shorten, such as described in the following

example. It is not necessary to omit the fields completely.

2001:db8:0:0:0::1/128

2001:db8:0:0::1/128

2001:db8:0::1/128

2001:db8::1/128

... and more

In addition, [RFC4291](#) in [section 2.2](#) notes,

'The "::" can also be used to compress leading or trailing zeros in an address.'

Therefore, it is possible to select such as the following example.

2001:db8::aaaa:0:0:1/128

2001:db8:0:0:aaaa::1/128

### [2.3](#). Uppercase or Lowercase

[RFC4291](#) does not mention about Uppercase and Lowercase. Because it's probably not special. However, it is possible to select such as the following example. Last 16bit is different, but these are the same.

2001:db8:aaaa:bbbb:cccc:dddd:eeee:aaaa/128

2001:db8:aaaa:bbbb:cccc:dddd:eeee:AAAA/128

2001:db8:aaaa:bbbb:cccc:dddd:eeee:AaAa/128

... more combinations

### [3. Problems Encountered with the Flexible Model](#)

#### [3.1. Searching](#)

##### [3.1.1. General Summary](#)

A search of an IPv6 address if conducted through a UNIX system is usually case sensitive and extended options to allow for regular expression use will come in handy. However, there are many applications in the internet today that do not provide this capability. When searching for an IPv6 address in such systems, the system engineer will have to try each and every possibility to search for an address. This has critical impacts especially when trying to deploy IPv6 over an enterprise network.

##### [3.1.2. Searching Spreadsheets and Text Files](#)

Spreadsheet applications and text editors on GUI systems, rarely have the ability to search for a text using regular expression. Moreover, there are many non-engineers (who are not aware of case sensitivity and regular expression use) that use these application to manage IP addresses. This has worked quite well with IPv4 since text representation in IPv4 has very little flexibility. There is no incentive to encourage these non-engineers to change their tool or learn regular expression when they decide to go dual-stack. If the entry in the spreadsheet reads, 2001:db8::1:0:0:1, but the search was conducted as 2001:db8:0:0:1::1, this will show a result of no match. One example where this will cause problem is, when the search is being conducted to assign a new address from a pool, and a check was being done to see if it was not in use. This may cause problems to the end-hosts or end-users. This type of address management is very often seen in enterprise networks and also in ISPs.

##### [3.1.3. Searching with Whois](#)

The whois utility is used by a wide range of people today. When a record is set to the whois database, one will likely check the output to see if the entry is correct. If a entity was recorded as 2001:db8::/48, but the whois output showed 2001:0db8:0000::/48, most non-engineers would think that their input was wrong, and will likely retry several times or make a frustrated call to the database

hostmaster. If there was a need to register the same address on different systems, and each system showed a different text representation, this would confuse people even more.

#### [3.1.4.](#) Searching for an Address in a Network Diagram

Network diagrams and blue-prints contain IP addresses of systems. In times of trouble shooting, there may be a need to search through a diagram to find the point of failure (for example, if a traceroute stopped at 2001:db8::1, one would search the diagram for that address). This is a technique quite often in use in enterprise networks and managed services. Again, the different flavors of text representation will result in a time-consuming search, leading to longer MTTR in times of trouble.

### [3.2.](#) Parsing and Modifying

#### [3.2.1.](#) General Summary

With all the possible text representation ways, each application must include a module, object, link, etc. to a function that will parse IPv6 addresses in a manner that no matter how it is represented, they will mean the same address. This is not too much a problem if the output is to be just 'read' or 'managed' by a network engineer. However, many system engineers who integrate complex computer systems to corporate customers will have difficulties finding that their favorite tool will not have this function, or will encounter difficulties such as having to rewrite their macro's or scripts for their customers. It must be noted that each additional line of a program will result in increased development fees that will be charged to the customers.

#### [3.2.2.](#) Logging

If an application were to output a log summary that represented the address in full (such as 2001:0db8:0000:0000:1111:2222:3333:4444), the output would be highly unreadable compared to the IPv4 output. The address would have to be parsed and reformed to make it useful for human reading. This will result in additional code on the applications which will result in extra fees charged to the

customers. Sometimes, logging for critical systems is done by



mirroring the same traffic to two different systems. Care must be taken that no matter what the log output is, the logs should be parsed so they will mean the same.

### [3.2.3.](#) Auditing. Case 1

When a router or any other network appliance machine configuration is audited, there are many methods to compare the configuration information of a node. Sometimes, auditing will be done by just comparing the changes made each day. In this case, if configuration was done such that 2001:db8::1 was changed to 2001:0db8:0000:0000:0000:0000:0000:0001 just because the new engineer on the block felt it was better, a simple diff will tell you that a different address was configured. If this was done on a wide scale network, people will be focusing on 'why the extra zeros were put in' instead of doing any real auditing. Lots of tools are just plain diffs that do not take into account address representation rules.

### [3.2.4.](#) Auditing. Case 2

Node configurations will be matched against a information system that manages IP addresses. If output notation is different, there will need to be a script that is implemented to cover for this. An SNMP GET of an interface address and text representation in a humanly written text file is highly unlikely to match on first try.

### [3.2.5.](#) Unexpected Modifying

Sometimes, a system will take an address and modify it as a convenience. For example, a router may take an input of 2001:0db8:0::1 and make the output 2001:db8::1 (which is seen in some RIR databases). If the zeros were inputted for a reason, the outcome may be somewhat unexpected.

## [3.3.](#) Operating

### [3.3.1.](#) General Summary

When an operator sets an IPv6 address of a system as 2001:db8:0:0:1:0:0:1/128, the system may take the address and show the configuration result as 2001:DB8::1:0:0:1/128. A distinguished engineer will know that the right address is set, but an operator, or a customer that is communicating with the operator to solve a problem, is usually not as distinguished as we would like. Again, the extra load in checking that the IP address is the same as was intended, will result in fees that will be charged to the customers.

### [3.3.2.](#) Customer Calls

When a customer calls to inquire about a suspected outage, IPv6 address representation should be handled with care. Not all customers are engineers nor have the same skill in IPv6 technology. The NOC will have to take extra steps to humanly parse the address to avoid having to explain to the customers that 2001:db8:0:1::1 is the same as 2001:db8::1:0:0:0:1. This is one thing that will never happen in IPv4 because IPv4 address cannot be abbreviated.

### [3.3.3.](#) Abuse

Network abuse is reported along with the abusing IP address. This 'reporting' could take any shape or form of the flexible model. A team that handles network abuse must be able to tell the difference between a 2001:db8::1:0:1 and 2001:db8:1::0:1. Mistakes in the placement of the "::" will result in a critical situation. A system that handles these incidents should be able to handle any type of input and parse it in a correct manner. Also, incidents are reported over the phone. It is unnecessary to report if the letter is an uppercase or lowercase. However, when a letter is spelled uppercase, people tend to clarify that it is uppercase, which is unnecessary information.

## [3.4.](#) Other Minor Problems

### [3.4.1.](#) Changing Platforms

When an engineer decides to change the platform of a running service, the same code may not work as expected due to the difference in IPv6 address text representation. Usually, a change in a platform (e.g. Unix to Windows, Cisco to Juniper) will result in a major change of code, but flexibility in address representation will increase the work load which will again, result in fees that will be charged to the customers, and also longer down time of systems.

### [3.4.2.](#) Preference in Documentation

A document that is edited by more than one author, may become harder to read.

### [3.4.3.](#) Legibility

Capital case D and 0 can be quite often misread.

Internet-Draft

IPv6 Text Representation

March 2009

## [4.](#) A Recommendation for IPv6 Text Representation

### [4.1.](#) Handling Leading Zeros

Leading zeros should be chopped for human legibility and easier searching. Also, a single 16 bit 0000 field should be represented as just 0. Place holder zeros are often cause of mis-reading.

### [4.2.](#) Lower Case

Recent implementations tend to represent IPv6 address as lower case. It is better to use lower case to avoid problems such as described in [section 3.3.3](#) and 3.4.3.

### [4.3.](#) "::" usage

#### [4.3.1.](#) shorten as much as possible

The use of "::" should be used to its maximum capability (i.e. 2001:db8::0:1/128 is not very clean).

#### [4.3.2.](#) one 16bit 0 field

"::" should not be used to shorten just one 16bit 0 field for it would tend to mislead that there are more than one 16 bit field that is shortened.

#### [4.3.3.](#) when "::" can be used twice

When cases where it is possible to use "::" in two or more different sections of an address, implementation to shorten the side with more 16bit 0 fields are more common (i.e. latter is shortened in 2001:0:0:1:0:0:0:1/128). When the length of 16bit 0 fields are equal (i.e. 2001:db8:0:0:1:0:0:1/128), the former is usually shortened. One idea to avoid any confusion, is for the operator to not use 16bit field 0 in the first 64 bits. By nature IPv6 addresses are usually assigned or allocated to end-users as longer than 32 bits (typically 48bit or longer).

## [5.](#) Conclusion

For developers, it is recommended that they use `inet_ntop()` or `WSAAddressToString()`. These have a consistent rule with this draft. If you have difficulties using these functions, it is desirable to adapt the recommended rule. For all those who have the need of text representing an IPv6 address, the following is a summary of the recommended rules.

- (1) omit leading zeros
- (2) use lower case
- (3) "::" used to their maximum extent whenever possible
- (4) "::" used where shortens address the most
- (5) "::" used in the former part in case of a tie breaker
- (6) do not shorten one 16bit 0 field

## [6.](#) Security Considerations

None.

## [7.](#) IANA Considerations

None.

## [8.](#) Acknowledgements

The authors would like to thank Jan Zorz, Randy Bush, Yuichi Minami, Toshimitsu Matsuura for their generous and helpful comments.

## [9.](#) References

### [9.1.](#) Normative References

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), February 2006.

## [9.2.](#) Informative References

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

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", [RFC 3493](#), February 2003.

Kawamura & Kawashima Expires September 27, 2009

[Page 11]

---

Internet-Draft

IPv6 Text Representation

March 2009

### Authors' Addresses

Seiichi Kawamura  
NEC BIGLOBE, Ltd.  
14-22, Shibaura 4-chome  
Minatoku, Tokyo 108-8558  
JAPAN

Phone: +81 3 3798 6085  
Email: kawamucho@mesh.ad.jp

Masanobu Kawashima  
NEC AccessTechnica, Ltd.  
800, Shimomata  
Kakegawa-shi, Shizuoka 436-8501  
JAPAN

Phone: +81 537 23 9655  
Email: kawashimam@necat.nec.co.jp

