

GEOPRIV
Internet-Draft
Expires: April 5, 2007

J. Schnizlein
M. Linsner
Cisco Systems
October 2, 2006

**Binary to Decimal Conversion for Location Configuration Information
draft-schnizlein-geopriv-binary-lci-00**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of 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/1id-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 April 5, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006)

Abstract

This document describes the nature of the data expressed in the geographic LCI defined in [RFC 3825](#), and includes examples of conversion from its binary format to decimal character strings.

Schnizlein, Linsner

Expires April 5, 2007

[Page 1]

Table of Contents

1.	Terminology	2
2.	Definitions	2
3.	Introduction	2
4.	Overview	2
5.	Programming Hints	3
6.	Calculation of LCI values	5
7.	IANA Considerations	6
8.	Security	6
9.	References	6
9.1	Normative References	6
9.2	Informative References	6
10.	Author's Address	6
A.	Intellectual Property and Copyright Statements	7

[1.](#) Terminology

In this document, the key words "MUST", "MUSTNOT", "REQUIRED", "SHALL", "SHALLNOT", "SHOULD", "SHOULDNOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119](#) [[1](#)] and indicate requirement levels for compliant implementations.

[2.](#) Definitions

This document uses the following terms to describe geo LCI binary to decimal conversion:

Location Configuration Information: (LCI) An object that carries location information. LCI has no ability to express privacy rules as outlined in [[3](#)] and [[4](#)], therefore is considered part of the 'sighting' function. For purposes of this discussion, all references to LCI refer to its use in [[1](#)].

GNU Compiler Collection: (GCC) The GNU Compiler Collection is a set of programming language compilers produced by the GNU Project.

[3.](#)

Introduction

The LCI encodes a point's latitude, longitude and altitude, along with the resolution of that point. LCI does not encode boundaries of an arbitrary region. The resolution is nothing more than the representation of significant digits for the fixed-length, binary values in the LCI.

Format conversion is required between the binary LCI that a host can receive through DHCP [[1](#)] or LLDP-MED [[5](#)] and the decimal representation used by applications, e.g. PIDF-LO [[2](#)]. This conversion could be used by a host that provides its location to another party with the privacy rules of the [[2](#)], including to a server authorized to redistribute the

information. It is unclear why anyone would need to convert from the geographic-coordinate location format of [2] to the LCI.

[4.](#) Overview

Schnizlein, Linsner

Expires April 5, 2007

[Page 2]

This section provides an overview of the programming hints in the next section for the translation from the efficient binary representation of the LCI [1][5] to the decimal string representation of geographic location used in PIDF-LO [2], for example. GCC syntax is used because it is well known. The binary values are converted to decimal, with the invalid bits removed and with the number of significant digits determined by the resolution of the binary values.

After unpacking the network-order bytes of the LCI into C variables sufficiently large to accommodate the fields, the sign bit of the two's-complement integers are extended to the size of the variable. The sign bit at 34 bits to the left is tested with an octal constant containing 33 bits in 11 octal zeros. If negative, the lower 34 bits of a constant minus-one are inverted with XOR, and then (inclusive) ORed with the LCI value. This operation is safe to perform more than once.

Because [1] says "Contents beyond the claimed resolution MAY be randomized ...", these contents are erased, i.e. set to zero. The number of bits to erase is the field length minus the resolution of the value in that field. A mask is constructed by left-shifting a one into the right of the mask for as many bits as to be erased. ANDing the inverse of the mask with the value erases the invalid bits.

The fixed-point fraction values are scaled into a floating-point (double for enough precision) by dividing by the constant reflecting the number of fractional bits. Note that latitude and longitude have 25 bits of fraction, while altitude has only 22 bits. The number of significant digits to the right of the decimal point is the resolution minus its integer portion, scaled by 3 decimal digits for 10 binary digits because 10 to the 3rd = 1000 approximates 2 to the 10th = 1024.

5. Programming hints

The LCI format is as follows:

0																1																2																3															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																								
Code 123																16																LaRes																Latitude															
Latitude (cont'd)																LoRes																																															
Longitude																																																															
AT																AltRes																Altitude																															
Alt (cont'd)																Datum																																															

Schnizlein, Linsner

Expires April 5, 2007

[Page 3]

Assume the following element values have been unpacked from the 16 bytes of the wire protocol above.

```
struct {
    int8 code;          /* DHCP LCI option code = 123 */
    int8 length;        /* length 16 bytes - not incl code + length */
    int8 LaRes;         /* Latitude Resolution 6 bits */
    int64 Latitude      /* Latitude 34 bits, 25 fractional */
    int8 LoRes;         /* Longitude Resolution 6 bits */
    int64 Longitude     /* Latitude 34 bits, 25 fractional */
    int8 AltType        /* Altitude Type 4 bits */
    int8 AltRes         /* Altitude Resolution 6 bits */
    int64 Altitude      /* Altitude 30 bits 22 bits Fraction */
    int8 Datum          /* Datum code 8 bits */
} LCIOption;
```

Because the latitude, longitude, and altitude values are twos complement of non-standard length, they require sign-extension that is not built into typical variable types. For the Latitude example:

```
struct LCIOption OptIn;
/* unpack into elements */

if (OptIn.Latitude & 010000000000LL)
    OptIn.Latitude = (-1 ^ 017777777777LL) | OptIn.Latitude
    /* ^ is XOR to flip one bits to zero before ORing in the field */
    /* if negative 34-bit field, set all one-bits above 34-bit field */
```

Translation from the binary resolution of the LCI to the correct number of significant decimal digits in the character string representation used for numbers in PIDF-LO is as in the following example for Longitude:

```
eraseBits = 34 - OptIn.LoRes;
int64 mask = 0LL;
if (eraseBits > 0) while (eraseBits--) mask = (mask << 1) | 1;
/* mask bit == 1 for invalid bits */
OptIn.Longitude &= ~mask;
/* invert mask and AND to zero invalid bits */

double longitude = OptIn.Longitude / exp2 (25);
/* scale integer for 25 bits of fraction */
int8 LongFractDigits = (OptIn.LoRes - 9) * 3 / 10;
/* deduct integer part, 2 to 10 ~ 10 to 3 */
if (LongFractDigits < 0) LongFractDigits = 0;
/* 34 bits is about 10 digits, plus 1 for sign */
printf ("%11.*F", LongFractDigits, longitude);
```

Schnizlein, Linsner

Expires April 5, 2007

[Page 4]

6. Calculation of LCI values

Since the Global Positioning System (GPS) or survey methods do not provide location in the LCI format, this section illustrates how a network administrator might calculate the values in preparation for delivering them to hosts connected to her network.

Where geographic location is expressed with the correct number of significant digits, it is easy to compute resolution because 3 decimal digits approximate 10 bits. The number of digits to the right of the decimal point, times 10, divided by 3 is the number of fractional bits. Adding 9 for the integer part yields the resolution.

Where a geographic location comes with an explicit error specification, this error can be translated into the resolution of the LCI. If the error measure is in distance (e.g. meters) rather than degrees, the conversion of longitude to degrees depends on the distance from the equator. Dividing the error distance by the distance for one degree (computed with the method described at [6]) yields the error in (presumably fractional) degrees.

```
double ErrorDegrees;
int64 FixedPntErrDeg = ErrorDegrees / exp2(25);
    /* convert error to fixed point 25-bit fraction */

int 64 TopBit = 010000000000LL;
if (FixedPntErrDeg & TopBit) FixedPntErrDeg = - FixedPntErrDeg;
    /* if negative make positive */

int8 resolution = 0;

while (~(FixedPntErrDeg & (TopBit >= 1))) resolution++;
    /* shift test bit to find first non-zero error */
    /* this is the number of valid bits */
```

If all that is available is the bounding points of a region, the difference between the extremes and the center in both latitude and longitude estimates the error in degrees, which can be converted to resolution as above. Find the maximum and minimum of both, calculate the value of the latitude/longitude as the average, and half the difference as the error.

For the example bounds ranging about 0.5 meters in distances across about [32 degrees](#), the binary and decimal values are as follows:

binary	decimal	
000011111.11111111111111111001110	31.99999850	
000100000.000000000000000001011100	32.00000274	
001000000.0000000000000000000101010	64.00000124	sum
000100000.000000000000000000010101	32.00000062	average

000000000.00000000000000000010001110 00.00000423 difference

Schnizlein, Linsner

Expires April 5, 2007

[Page 5]

With 26 bits above the difference, which is twice the error, this example yields 27 bits of resolution (remembering to add 9 bits for left of the binary point).

7. IANA Considerations

No IANA Considerations

8. Security

This document discusses binary to decimal conversion within an end host, which raises no particular security considerations.

9. References

9.1 Normative References

- [1] [RFC3825](#) Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information. J. Polk, J. Schnizlein, M. Linsner. July 2004.
- [2] [RFC4119](#) A Presence-based GEOPRIV Location Object Format. J. Peterson. December 2005.
- [3] [RFC3693](#) Geopriv Requirements. J. Cuellar, J. Morris, D. Mulligan, J. Peterson, J. Polk. February, 2004.
- [4] [RFC3694](#) Threat Analysis of the Geopriv Protocol. M. Danley, D. Mulligan, J. Morris, J. Peterson. February 2004

9.2 Informative References

- [5] Telecommunications -- IP Telephony Infrastructure -- Link Layer Discovery Protocol (LLDP) for Media Endpoint Devices," TIA-1057
- [6] "Problem 2A.: Calculate path length along a meridian given starting and ending coordinates". Andy McGovern. April 2004
<http://www.codeguru.com/Cpp/Cpp/algorithms/general/article.php/c5115>

10. Author's Address

John Schnizlein
Cisco Systems, Inc.
Fort Washington, MD, USA
Email: john.schnizlein@cisco.com

Schnizlein, Linsner

Expires April 5, 2007

[Page 6]

Marc Linsner
Cisco Systems, Inc.
Marco Island, Florida, USA
Email: marc.linsner@cisco.com

Comments are solicited and should be addressed to the working group's mailing list at geopriv@ietf.org and/or the authors.

Appendix A.

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Schnizlein, Linsner

Expires April 5, 2007

[Page 7]

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.