

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 10, 2020

E. Kline
Loon LLC
K. Duleba
Z. Szamonek
S. Moser
Google Switzerland GmbH
W. Kumari
Google
February 7, 2020

A Format for Self-published IP Geolocation Feeds
draft-google-self-published-geofeeds-09

Abstract

This document records a format whereby a network operator can publish a mapping of IP address prefixes to simplified geolocation information, colloquially termed a geolocation "feed". Interested parties can poll and parse these feeds to update or merge with other geolocation data sources and procedures. This format intentionally only allows specifying coarse level location.

Some technical organizations operating networks that move from one conference location to the next have already experimentally published small geolocation feeds.

This document describes a currently deployed format. At least one consumer (Google) has incorporated these feeds into a geolocation data pipeline, and a significant number of ISPs are using it to inform them where their prefixes should be geolocated.

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 August 10, 2020.

Copyright Notice

Copyright (c) 2020 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 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. Introduction	3
1.1. Motivation	3
1.2. Requirements Notation	3
1.3. Assumptions About Publication	4
2. Self-Published IP Geolocation Feeds	4
2.1. Specification	4
2.1.1. Geolocation Feed Individual Entry Fields	5
2.1.1.1. IP Prefix	5
2.1.1.2. Alpha2code (previously: 'country')	5
2.1.1.3. Region	5
2.1.1.4. City	6
2.1.1.5. Postal Code	6
2.1.2. Prefixes With No Geolocation Information	6
2.1.3. Additional Parsing Requirements	7
2.2. Examples	7
3. Consuming Self-Published IP Geolocation Feeds	8
3.1. Feed Integrity	8
3.2. Verification of Authority	8
3.3. Verification of Accuracy	9
3.4. Refreshing Feed Information	9
4. Privacy Considerations	9
5. Relation to Other Work	10
6. Security Considerations	11
7. Planned Future Work	11
8. Finding Self-Published IP Geolocation Feeds	11
8.1. Ad Hoc 'Well Known' URIs	12
8.2. Other Mechanisms	12
9. IANA Considerations	12
10. Acknowledgements	12
11. References	13
11.1. Normative References	13

11.2. Informative References	14
11.3. URIs	15
Appendix A. Sample Python Validation Code	15
Authors' Addresses	22

1. Introduction

1.1. Motivation

Providers of services over the Internet have grown to depend on best-effort geolocation information to improve the user experience. Locality information can aid in directing traffic to the nearest serving location, inferring likely native language, and providing additional context for services involving search queries.

When an ISP, for example, changes the location where an IP prefix is deployed, services which make use of geolocation information may begin to suffer degraded performance. This can lead to customer complaints, possibly to the ISP directly. Dissemination of correct geolocation data is complicated by the lack of any centralized means to coordinate and communicate geolocation information to all interested consumers of the data.

This document records a format whereby a network operator (an ISP, an enterprise, or any organization which deems the geolocation of its IP prefixes to be of concern) can publish a mapping of IP address prefixes to simplified geolocation information, colloquially termed a "geolocation feed". Interested parties can poll and parse these feeds to update or merge with other geolocation data sources and procedures.

This document describes a currently deployed format. At least one consumer (Google) has incorporated these feeds into a geolocation data pipeline, and a significant number of ISPs are using it to inform them where their prefixes should be geolocated.

1.2. Requirements Notation

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

As this is an informational document about a data format and set of operational practices presently in use, requirements notation captures the design goals of the authors and implementors.

1.3. Assumptions About Publication

This document describes both a format and a mechanism for publishing data, with the assumption that the network operator to whom operational responsibility has been delegated for any published data wishes it to be public. Any privacy risk is bounded by the format, and feed publishers MAY omit prefixes or any location field associated with a given prefix to further protect privacy (see Section 2.1 for details about which fields exactly may be omitted). Feed publishers assume the responsibility of determining which data should be made public.

This document does not incorporate a mechanism to communicate acceptable use policies for self-published data. Publication itself is inferred as a desire by the publisher for the data to be usefully consumed, similar to the publication of information like host names, cryptographic keys, and SPF records [RFC4408] in the DNS.

2. Self-Published IP Geolocation Feeds

The format described here was developed to address the need of network operators to rapidly and usefully share geolocation information changes. Originally, there arose a specific case where regional operators found it desirable to publish location changes rather than wait for geolocation algorithms to "learn" about them. Later, technical conferences which frequently use the same network prefixes advertised from different conference locations experimented by publishing geolocation feeds, updated in advance of network location changes, in order to better serve conference attendees.

At its simplest, the mechanism consists of a network operator publishing a file (the "geolocation feed"), which contains several text entries, one per line. Each entry is keyed by a unique (within the feed) IP prefix (or single IP address) followed by a sequence of network locality attributes to be ascribed to the given prefix.

2.1. Specification

For operational simplicity, every feed should contain data about all IP addresses the provider wants to publish. Alternatives, like publishing only entries for IP addresses whose geolocation data has changed or differ from current observed geolocation behavior "at large", are likely to be too operationally complex.

Feeds MUST use UTF-8 [RFC3629] character encoding. Lines are delimited by a line break (CRLF) (as specified in [RFC4180]), and blank lines are ignored. Text from a '#' character to the end of the current line is treated as a comment only and is similarly ignored

(note that this does not strictly follow [RFC4180], which has no support for comments).

Feed lines that are not comments MUST be in comma separated value (CSV) format as described in [RFC4180]. Each feed entry is a text line of the form:

```
ip_prefix,alpha2code,region,city,postal_code
```

The IP prefix field is REQUIRED, all others are OPTIONAL (can be empty), though the requisite minimum number of commas SHOULD be present.

2.1.1. Geolocation Feed Individual Entry Fields

2.1.1.1. IP Prefix

REQUIRED. Each IP prefix field MUST be either a single IP address or an IP prefix in CIDR notation in conformance with section 3.1 [1] of [RFC4632] for IPv4 or section 2.3 [2] of [RFC4291] for IPv6.

Examples include "192.0.2.1" and "192.0.2.0/24" for IPv4 and "2001:db8::1" and "2001:db8::/32" for IPv6.

2.1.1.2. Alpha2code (previously: 'country')

OPTIONAL. The alpha2code field, if non-empty, MUST be a 2 letter ISO country code conforming to ISO 3166-1 alpha 2 [ISO.3166.1.alpha2]. Parsers SHOULD treat this field case-insensitively.

Earlier versions of this document called this field "country", and it may still be referred to as such in existing tools / interfaces.

Parsers MAY additionally support other 2 letter codes outside the ISO 3166-1 alpha 2 codes. For example, 2 letter codes from the "Exceptionally reserved codes" [3] set may appear in this field, e.g. "UK" [4] or "EU" [5].

Examples include "US" for the United States, "JP" for Japan, and "PL" for Poland.

2.1.1.3. Region

OPTIONAL. The region field, if non-empty, MUST be a ISO region code conforming to ISO 3166-2 [ISO.3166.2]. Parsers SHOULD treat this field case-insensitively.

Examples include "ID-RI" for the Riau province of Indonesia and "NG-RI" for the Rivers province in Nigeria.

2.1.1.4. City

OPTIONAL. The city field, if non-empty, SHOULD be free UTF-8 text, excluding the comma (',') character.

Examples include "Dublin", "New York", and "Sao Paulo" (specifically "S" followed by 0xc3, 0xa3, and "o Paulo").

2.1.1.5. Postal Code

OPTIONAL, DEPRECATED. The postal code field, if non-empty, SHOULD be free UTF-8 text, excluding the comma (',') character. The use of this field is deprecated; consumers of feeds should be able to parse feeds containing these fields, but new feeds SHOULD NOT include this field, due to the granularity of this information. See Section 4 for additional discussion.

Examples include "106-6126" (in Minato ward, Tokyo, Japan).

2.1.2. Prefixes With No Geolocation Information

Feed publishers may indicate that some IP prefixes should not have any associated geolocation information. It may be that some prefixes under their administrative control are reserved, not yet allocated or deployed, or are in the process of being redeployed elsewhere and existing geolocation information can, from the perspective of the publisher, safely be discarded.

This special case can be indicated by explicitly leaving blank all fields which specify any degree of geolocation information. For example:

```
192.0.2.0/24,,,
2001:db8:1::/48,,,
2001:db8:2::/48,,,
```

Historically, the user-assigned alpha2code identifier of "ZZ" has been used for this same purpose. This is not necessarily preferred, and no specific interpretation of any of the other user-assigned alpha2code codes is currently defined.

2.1.3. Additional Parsing Requirements

Feed entries missing an IP address or prefix field or having an IP address or prefix field which fails to parse correctly MUST be discarded.

While publishers SHOULD follow [RFC5952] style for IPv6 prefix fields, consumers MUST nevertheless accept all valid string representations.

Duplicate IP address or prefix entries MUST be considered an error, and consumer implementations SHOULD log the repeated entries for further administrative review. Publishers SHOULD take measures to ensure there is one and only one entry per IP address and prefix.

Multiple entries which constitute nested prefixes are permitted. Consumers SHOULD consider the entry with the longest matching prefix (i.e. the "most specific") to be the best matching entry for a given IP address.

Feed entries with non-empty optional fields which fail to parse, either in part or in full, SHOULD be discarded. It is RECOMMENDED that they also be logged for further administrative review.

For compatibility with future additional fields, a parser MUST ignore any fields beyond those it expects. The data from fields which are expected and which parse successfully MUST still be considered valid. Per Section 7 no extensions to this format are in use nor are any anticipated.

2.2. Examples

Example entries using different IP address formats and describing locations at alpha2code ("country code"), region, and city granularity level, respectively:

```
192.0.2.0/25,US,US-AL,,
192.0.2.5,US,US-AL,Alabaster,
192.0.2.128/25,PL,PL-MZ,,
2001:db8::/32,PL,,,
2001:db8:cafe::/48,PL,PL-MZ,,
```

The IETF network publishes geolocation information for the meeting prefixes, and generally just comment out the last meeting information and append the new meeting information. The [GEO_IETF] at the time of this writing contains:

```
# IETF106 (Singapore) - November 2019 - Singapore, SG
130.129.0.0/16,SG,SG-01,Singapore,
2001:df8::/32,SG,SG-01,Singapore,
31.133.128.0/18,SG,SG-01,Singapore,
31.130.224.0/20,SG,SG-01,Singapore,
2001:67c:1230::/46,SG,SG-01,Singapore,
2001:67c:370::/48,SG,SG-01,Singapore,
```

Experimentally, RIPE has published geolocation information for their conference network prefixes, which change location in accordance with each new event. [GEO_RIPE_NCC] at the time of writing contains:

```
193.0.24.0/21,NL,NL-ZH,Rotterdam,
2001:67c:64::/48,NL,NL-ZH,Rotterdam,
```

Similarly, ICANN has published geolocation information for their portable conference network prefixes. [GEO_ICANN] at the time of writing contains:

```
199.91.192.0/21,MA,MA-07,Marrakech
2620:f:8000::/48,MA,MA-07,Marrakech
```

A longer example is the [GEO_Google] Google Corp Geofeed, which lists the geo-location information for Google corporate offices.

At the time of writing, Google processes approximately 400 feeds comprising more than 750,000 IPv4 and IPv6 prefixes.

3. Consuming Self-Published IP Geolocation Feeds

Consumers MAY treat published feed data as a hint only and MAY choose to prefer other sources of geolocation information for any given IP prefix. Regardless of a consumer's stance with respect to a given published feed, there are some points of note for sensibly and effectively consuming published feeds.

3.1. Feed Integrity

The integrity of published information SHOULD be protected by securing the means of publication, for example by using HTTP over TLS [RFC2818]. Whenever possible, consumers SHOULD prefer retrieving geolocation feeds in a manner that guarantees integrity of the feed.

3.2. Verification of Authority

Consumers of self-published IP geolocation feeds SHOULD perform some form of verification that the publisher is in fact authoritative for the addresses in the feed. The actual means of verification is

likely dependent upon the way in which the feed is discovered. Ad hoc shared URIs, for example, will likely require an ad hoc verification process. Future automated means of feed discovery SHOULD have an accompanying automated means of verification.

A consumer should only trust geolocation information for IP addresses or prefixes for which the publisher has been verified as administratively authoritative. All other geolocation feed entries should be ignored and logged for further administrative review.

3.3. Verification of Accuracy

Errors and inaccuracies may occur at many levels, and publication and consumption of geolocation data are no exceptions. To the extent practical, consumers SHOULD take steps to verify the accuracy of published locality. Verification methodology, resolution of discrepancies, and preference for alternative sources of data are left to the discretion of the feed consumer.

Consumers SHOULD decide on discrepancy thresholds and SHOULD flag for administrative review feed entries which exceed set thresholds.

3.4. Refreshing Feed Information

As a publisher can change geolocation data at any time and without notification, consumers SHOULD implement mechanisms to periodically refresh local copies of feed data. In the absence of any other refresh timing information, it is recommended that consumers SHOULD refresh feeds no less often than weekly, and no more often than is likely to cause issues to the publisher.

For feeds available via HTTPS (or HTTP), the publisher MAY communicate refresh timing information by means of the standard HTTP expiration model (section 13.2 [6] of [RFC2616]). Specifically, publishers can include either an Expires header [7] or a Cache-Control header [8] specifying the max-age. Where practical, consumers SHOULD refresh feed information before the expiry time is reached.

4. Privacy Considerations

Publishers of geolocation feeds are advised to have fully considered any and all privacy implications of the disclosure of such information for the users of the described networks prior to publication. A thorough comprehension of the security considerations [9] of a chosen geolocation policy is highly recommended, including an understanding of some of the limitations of information obscurity [10] (see also [RFC6772]).

As noted in Section 2.1, each location field in an entry is optional, in order to support expressing only the level of specificity which the publisher has deemed acceptable. There is no requirement that the level of specificity be consistent across all entries within a feed. In particular, the Postal Code field (Section 2.1.1.5) can provide very specific geolocation, sometimes within a building. Such specific Postal Code values **MUST NOT** be published in geofeeds without the express consent of the parties being located.

Operators who publish geolocation information are strongly encouraged to inform affected users/customers of this fact and of the potential privacy-related consequences and trade-offs.

5. Relation to Other Work

While not originally done in conjunction with the [GEOPRIV] working group, Richard Barnes observed that this work is nevertheless consistent with that which the group has defined, both for address format and for privacy. The data elements in geolocation feeds are equivalent to the following XML structure (vis. [RFC5139]):

```
<civicAddress>
  <country>country</country>
  <A1>region</A1>
  <A2>city</A2>
  <PC>postal_code</PC>
</civicAddress>
```

Providing geolocation information to this granularity is equivalent to the following privacy policy (vis. the definition of the 'building' [11] level of disclosure):

```
<ruleset>
  <rule>
    <conditions/>
    <actions/>
    <transformations>
      <provide-location profile="civic-transformation">
        <provide-civic>building</provide-civic>
      </provide-location>
    </transformations>
  </rule>
</ruleset>
```

6. Security Considerations

As there is no true security in the obscurity of the location of any given IP address, self-publication of this data fundamentally opens no new attack vectors. For publishers, self-published data may increase the ease with which such location data might be exploited (it can, for example, make easy the discovery of prefixes populated with customers as distinct from prefixes not generally in use).

For consumers, feed retrieval processes may receive input from potentially hostile sources (e.g. in the event of hijacked traffic). As such, proper input validation and defense measures **MUST** be taken (see the discussion in Section 3.1).

Similarly, consumers who do not perform sufficient verification of published data bear the same risks as from other forms of geolocation configuration errors (see the discussion in Section 3.2 and Section 3.3).

Validation of a feed's contents includes verifying that the publisher is authoritative for the IP prefixes included in the feed. Failure to verify IP prefix authority would, for example, allow ISP Bob to make geolocation statements about IP space held by ISP Alice. At this time only out-of-band verification methods are implemented (i.e. an ISP's feed may be verified against publicly available IP allocation data).

7. Planned Future Work

In order to more flexibly support future extensions, use of a more expressive feed format has been suggested. Use of JavaScript Object Notation (JSON, [RFC4627]), specifically, has been discussed. However, at the time of writing no such specification nor implementation exists. Nevertheless, work on extensions is deferred until a more suitable format has been selected.

The authors are planning on writing a document describing such a new format. This document describes a currently deployed and used format. Given the extremely limited extensibility of the present format no extensions to it are anticipated. Extensibility requirements are instead expected to be integral to the development of a new format.

8. Finding Self-Published IP Geolocation Feeds

The issue of finding, and later verifying, geolocation feeds is not formally specified in this document. At this time, only ad hoc feed

discovery and verification has a modicum of established practice (see below); discussion of other mechanisms has been removed for clarity.

8.1. Ad Hoc 'Well Known' URIs

To date, geolocation feeds have been shared informally in the form of HTTPS URIs exchanged in email threads. Three of example URIs documented below ([GEO_IETF], [GEO_RIPE_NCC], [GEO_ICANN]) describe networks that change locations periodically, the operators and operational practices of which are well known within their respective technical communities.

The contents of the feeds are verified by a similarly ad hoc process including:

- o personal knowledge of the parties involved in the exchange, and
- o comparison of feed-advertised prefixes with the BGP-advertised prefixes of Autonomous System Numbers known to be operated by the publishers.

Ad hoc mechanisms, while useful for early experimentation by producers and consumers, are unlikely to be adequate for long-term, widespread use by multiple parties. Future versions of any such self-published geolocation feed mechanism SHOULD address scalability concerns by defining a means for automated discovery and verification of operational authority of advertised prefixes.

8.2. Other Mechanisms

Previous versions of this document referenced use of the WHOIS service ([RFC3912]) operated by RIRs as well as possible DNS-based schemes to discover and validate geofeeds. To the authors' knowledge support for such mechanisms has never been implemented, and this speculative text has been removed to avoid ambiguity.

9. IANA Considerations

This document makes no requests of the IANA.

10. Acknowledgements

The authors would like to express their gratitude to reviewers and early implementers, including but not limited to Mikael Abrahamsson, Andrew Alston, Ray Bellis, John Bond, Alissa Cooper, Andras Erdei, Stephen Farrell, Marco Hogewoning, Mike Joseph, Maciej Kuzniar, George Michaelson, Menno Schepers, Justyna Sidorska, Pim van Pelt, and Bjoern A. Zeeb.

Richard L. Barnes and Andy Newton in particular contributed substantial review, text, and advice.

11. References

11.1. Normative References

- [ISO.3166.1alpha2] International Organization for Standardization, "ISO 3166-1 decoding table", <http://www.iso.org/iso/home/standards/country_codes/iso-3166-1_decoding_table.htm>.
- [ISO.3166.2] International Organization for Standardization, "ISO 3166-2:2007", <http://www.iso.org/iso/home/standards/country_codes.htm#2012_iso3166-2>.
- [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, DOI 10.17487/RFC2616, June 1999, <<https://www.rfc-editor.org/info/rfc2616>>.
- [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>>.
- [RFC4180] Shafranovich, Y., "Common Format and MIME Type for Comma-Separated Values (CSV) Files", RFC 4180, DOI 10.17487/RFC4180, October 2005, <<https://www.rfc-editor.org/info/rfc4180>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <<https://www.rfc-editor.org/info/rfc4632>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<https://www.rfc-editor.org/info/rfc5952>>.

11.2. Informative References

- [GEO_Google]
Google, LLC, "Google Corp Geofeed",
<https://www.gstatic.com/geofeed/corp_external>.
- [GEO_ICANN]
Internet Corporation For Assigned Names and Numbers,
"ICANN Meeting Geolocation Data",
<<https://meeting-services.icann.org/geo/google.csv>>.
- [GEO_IETF]
Kumari, A., "IETF Meeting Network Geolocation Data",
<<https://noc.ietf.org/geo/google.csv>>.
- [GEO_RIPE_NCC]
Schepers, M., "RIPE NCC Meeting Geolocation Data",
<<https://meetings.ripe.net/geo/google.csv>>.
- [GEOPRIV] Internet Engineering Task Force, "IETF geopriv Working Group", <<http://datatracker.ietf.org/wg/geopriv/>>.
- [IPADDR_PY]
Shields, M. and P. Moody, "Python IP address manipulation library", <<http://code.google.com/p/ipaddr-py/>>.
- [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>>.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <<https://www.rfc-editor.org/info/rfc2818>>.
- [RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, DOI 10.17487/RFC3912, September 2004, <<https://www.rfc-editor.org/info/rfc3912>>.
- [RFC4408] Wong, M. and W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1", RFC 4408, DOI 10.17487/RFC4408, April 2006, <<https://www.rfc-editor.org/info/rfc4408>>.
- [RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, DOI 10.17487/RFC4627, July 2006, <<https://www.rfc-editor.org/info/rfc4627>>.

- [RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", RFC 5139, DOI 10.17487/RFC5139, February 2008, <<https://www.rfc-editor.org/info/rfc5139>>.
- [RFC6772] Schulzrinne, H., Ed., Tschofenig, H., Ed., Cuellar, J., Polk, J., Morris, J., and M. Thomson, "Geolocation Policy: A Document Format for Expressing Privacy Preferences for Location Information", RFC 6772, DOI 10.17487/RFC6772, January 2013, <<https://www.rfc-editor.org/info/rfc6772>>.
- [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>>.

11.3. URIs

- [1] <http://tools.ietf.org/html/rfc4632#section-3.1>
- [2] <http://tools.ietf.org/html/rfc4291#section-2.3>
- [3] <https://www.iso.org/glossary-for-iso-3166.html>
- [4] <https://www.iso.org/obp/ui/#iso:code:3166:UK>
- [5] <https://www.iso.org/obp/ui/#iso:code:3166:EU>
- [6] <http://tools.ietf.org/html/rfc2616#section-13.2>
- [7] <http://tools.ietf.org/html/rfc2616#section-14.21>
- [8] <http://tools.ietf.org/html/rfc2616#section-14.9>
- [9] <http://tools.ietf.org/html/rfc6772#section-13>
- [10] <http://tools.ietf.org/html/rfc6772#section-13.5>
- [11] <http://tools.ietf.org/html/rfc6772#section-6.5.1>

Appendix A. Sample Python Validation Code

Included here is a simple format validator in Python for self-published ipgeo feeds. This tool reads CSV data in the self-published ipgeo feed format from the standard input and performs basic validation. It is intended for use by feed publishers before launching a feed. Note that this validator does not verify the uniqueness of every IP prefix entry within the feed as a whole, but

only verifies the syntax of each single line from within the feed. A complete validator MUST also ensure IP prefix uniqueness.

The main source file "ipgeo_feed_validator.py" follows. It requires use of the open source ipaddr Python library for IP address and CIDR parsing and validation [IPADDR_PY].

```
<CODE BEGINS>
#!/usr/bin/python
#
# Copyright (c) 2012 IETF Trust and the persons identified as authors of
# the code. All rights reserved. Redistribution and use in source and
# binary forms, with or without modification, is permitted pursuant to,
# and subject to the license terms contained in, the Simplified BSD
# License set forth in Section 4.c of the IETF Trust's Legal Provisions
# Relating to IETF Documents (http://trustee.ietf.org/license-info).

"""Simple format validator for self-published ipgeo feeds.

This tool reads CSV data in the self-published ipgeo feed format from
the standard input and performs basic validation. It is intended for
use by feed publishers before launching a feed.
"""

import csv
import ipaddr
import re
import sys

class IPGeoFeedValidator(object):
    def __init__(self):
        self.prefixes = {}
        self.line_number = 0
        self.output_log = {}
        self.SetOutputStream(sys.stderr)

    def Validate(self, feed):
        """Check validity of an IPGeo feed.

        Args:
            feed: iterable with feed lines
        """

        for line in feed:
            self._ValidateLine(line)
```



```
def SetOutputStream(self, logfile):
    """Controls where the output messages go do (STDERR by default).

    Use None to disable logging.

    Args:
        logfile: a file object (e.g., sys.stdout or sys.stderr) or None.
    """
    self.output_stream = logfile

def CountErrors(self, severity):
    """How many ERRORS or WARNINGS were generated."""
    return len(self.output_log.get(severity, []))

#####
def _ValidateLine(self, line):
    line = line.rstrip('\r\n')
    self.line_number += 1
    self.line = line.split('#')[0]
    self.is_correct_line = True

    if self._ShouldIgnoreLine(line):
        return

    fields = [field for field in csv.reader([line])][0]

    self._ValidateFields(fields)
    self._FlushOutputStream()

def _ShouldIgnoreLine(self, line):
    line = line.strip()
    if line.startswith('#'):
        return True
    return len(line) == 0

#####
def _ValidateFields(self, fields):
    assert(len(fields) > 0)

    is_correct = self._IsIPAddressOrPrefixCorrect(fields[0])

    if len(fields) > 1:
        if not self._IsAlpha2CodeCorrect(fields[1]):
            is_correct = False

    if len(fields) > 2 and not self._IsRegionCodeCorrect(fields[2]):
        is_correct = False
```

```
if len(fields) != 5:
    self._ReportWarning('5 fields were expected (got %d).'
                        % len(fields))

#####
def _IsIPAddressOrPrefixCorrect(self, field):
    if '/' in field:
        return self._IsCIDRCorrect(field)
    return self._IsIPAddressCorrect(field)

def _IsCIDRCorrect(self, cidr):
    try:
        ipprefix = ipaddr.IPNetwork(cidr)
        if ipprefix.network._ip != ipprefix._ip:
            self._ReportError('Incorrect IP Network.')
            return False
        if ipprefix.is_private:
            self._ReportError('IP Address must not be private.')
            return False
    except:
        self._ReportError('Incorrect IP Network.')
        return False
    return True

def _IsIPAddressCorrect(self, ipaddress):
    try:
        ip = ipaddr.IPAddress(ipaddress)
    except:
        self._ReportError('Incorrect IP Address.')
        return False
    if ip.is_private:
        self._ReportError('IP Address must not be private.')
        return False
    return True

#####
def _IsAlpha2CodeCorrect(self, alpha2code):
    if len(alpha2code) == 0:
        return True
    if len(alpha2code) != 2 or not alpha2code.isalpha():
        self._ReportError(
            'Alpha 2 code must be in the ISO 3166-1 alpha 2 format.')
        return False
    return True

def _IsRegionCodeCorrect(self, region_code):
    if len(region_code) == 0:
        return True
```

```
if '-' not in region_code:
    self._ReportError('Region code must be in the ISO 3166-2 format.')
    return False

parts = region_code.split('-')
if not self._IsAlpha2CodeCorrect(parts[0]):
    return False
return True

#####
def _ReportError(self, message):
    self._ReportWithSeverity('ERROR', message)

def _ReportWarning(self, message):
    self._ReportWithSeverity('WARNING', message)

def _ReportWithSeverity(self, severity, message):
    self.is_correct_line = False
    output_line = '%s: %s\n' % (severity, message)

    if severity not in self.output_log:
        self.output_log[severity] = []
    self.output_log[severity].append(output_line)

    if self.output_stream is not None:
        self.output_stream.write(output_line)

def _FlushOutputStream(self):
    if self.is_correct_line: return
    if self.output_stream is None: return

    self.output_stream.write('line %d: %s\n\n'
                             % (self.line_number, self.line))

#####
def main():
    feed_validator = IPGeoFeedValidator()
    feed_validator.Validate(sys.stdin)

    if feed_validator.CountErrors('ERROR'):
        sys.exit(1)

if __name__ == '__main__':
    main()
<CODE ENDS>
```

A unit test file, "ipgeo_feed_validator_test.py" is provided as well. It provides basic test coverage of the code above, though does not test correct handling of non-ASCII UTF-8 strings.

```
<CODE BEGINS>
#!/usr/bin/python
#
# Copyright (c) 2012 IETF Trust and the persons identified as authors of
# the code. All rights reserved. Redistribution and use in source and
# binary forms, with or without modification, is permitted pursuant to,
# and subject to the license terms contained in, the Simplified BSD
# License set forth in Section 4.c of the IETF Trust's Legal Provisions
# Relating to IETF Documents (http://trustee.ietf.org/license-info).

import sys
from ipgeo_feed_validator import IPGeoFeedValidator

class IPGeoFeedValidatorTest(object):
    def __init__(self):
        self.validator = IPGeoFeedValidator()
        self.validator.SetOutputStream(None)
        self.successes = 0
        self.failures = 0

    def Run(self):
        self.TestFeedLine('# asdf', 0, 0)
        self.TestFeedLine(' ', 0, 0)
        self.TestFeedLine('', 0, 0)

        self.TestFeedLine('asdf', 1, 1)
        self.TestFeedLine('asdf,US,,, ', 1, 0)
        self.TestFeedLine('aaaa::,US,,, ', 0, 0)
        self.TestFeedLine('zzzz::,US', 1, 1)
        self.TestFeedLine(',US,,, ', 1, 0)
        self.TestFeedLine('55.66.77', 1, 1)
        self.TestFeedLine('55.66.77.888', 1, 1)
        self.TestFeedLine('55.66.77.asdf', 1, 1)

        self.TestFeedLine('2001:db8:cafe::/48,PL,PL-MZ,,02-784', 0, 0)
        self.TestFeedLine('2001:db8:cafe::/48', 0, 1)

        self.TestFeedLine('55.66.77.88,PL', 0, 1)
        self.TestFeedLine('55.66.77.88,PL,,, ', 0, 0)
        self.TestFeedLine('55.66.77.88,,,, ', 0, 0)
        self.TestFeedLine('55.66.77.88,ZZ,,, ', 0, 0)
        self.TestFeedLine('55.66.77.88,US,,, ', 0, 0)
        self.TestFeedLine('55.66.77.88,USA,,, ', 1, 0)
        self.TestFeedLine('55.66.77.88,99,,, ', 1, 0)
```

```

self.TestFeedLine('55.66.77.88,US,US-CA,,', 0, 0)
self.TestFeedLine('55.66.77.88,US,USA-CA,,', 1, 0)
self.TestFeedLine('55.66.77.88,USA,USA-CA,,', 2, 0)

self.TestFeedLine('55.66.77.88,US,US-CA,Mountain View,', 0, 0)
self.TestFeedLine('55.66.77.88,US,US-CA,Mountain View,94043', 0, 0)
self.TestFeedLine('55.66.77.88,US,US-CA,Mountain View,94043,'
                  '1600 Amphitheatre Parkway', 0, 1)

self.TestFeedLine('55.66.77.0/24,US,,, ', 0, 0)
self.TestFeedLine('55.66.77.88/24,US,,, ', 1, 0)
self.TestFeedLine('55.66.77.88/32,US,,, ', 0, 0)
self.TestFeedLine('55.66.77/24,US,,, ', 1, 0)
self.TestFeedLine('55.66.77.0/35,US,,, ', 1, 0)

self.TestFeedLine('172.15.30.1,US,,, ', 0, 0)
self.TestFeedLine('172.28.30.1,US,,, ', 1, 0)
self.TestFeedLine('192.167.100.1,US,,, ', 0, 0)
self.TestFeedLine('192.168.100.1,US,,, ', 1, 0)
self.TestFeedLine('10.0.5.9,US,,, ', 1, 0)
self.TestFeedLine('10.0.5.0/24,US,,, ', 1, 0)
self.TestFeedLine('fc00::/48,PL,,, ', 1, 0)
self.TestFeedLine('fe00::/48,PL,,, ', 0, 0)

print ('%d tests passed, %d failed'
      % (self.successes, self.failures))

def IsOutputLogCorrectAtSeverity(self, severity, expected_msg_count):
    msg_count = self.validator.CountErrors(severity)

    if msg_count != expected_msg_count:
        print ('TEST FAILED: %s\nexpected %d %s[s], observed %d\n%s\n' % (
            self.validator.line, expected_msg_count, severity, msg_count,
            str(self.validator.output_log[severity])))
        return False
    return True

def IsOutputLogCorrect(self, new_errors, new_warnings):
    retval = True

    if not self.IsOutputLogCorrectAtSeverity('ERROR', new_errors):
        retval = False
    if not self.IsOutputLogCorrectAtSeverity('WARNING', new_warnings):
        retval = False

    return retval

def TestFeedLine(self, line, warning_count, error_count):

```

```
self.validator.output_log['WARNING'] = []
self.validator.output_log['ERROR'] = []
self.validator._ValidateLine(line)

if not self.IsOutputLogCorrect(warning_count, error_count):
    self.failures += 1
    return False

self.successes += 1
return True

if __name__ == '__main__':
    IPGeoFeedValidatorTest().Run()
<CODE ENDS>
```

Authors' Addresses

Erik Kline
Loon LLC
1600 Amphitheatre Parkway
Mountain View, California 94043
United States of America

Email: ek@loon.com

Krzysztof Duleba
Google Switzerland GmbH
Brandschenkestrasse 110
Zuerich 8002
Switzerland

Email: kduleba@google.com

Zoltan Szamonek
Google Switzerland GmbH
Brandschenkestrasse 110
Zuerich 8002
Switzerland

Email: zszami@google.com

Stefan Moser
Google Switzerland GmbH
Brandschenkestrasse 110
Zuerich 8002
Switzerland

Email: smoser@google.com

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: warren@kumari.net

GEOPRIV
Internet-Draft
Intended status: Standards Track
Expires: February 28, 2014

M. Thomson
Microsoft
August 27, 2013

Expressing Confidence in a Location Object
draft-thomson-geopriv-confidence-04

Abstract

A confidence element is described that expresses the estimated probability that the associated location information is correct. This element conveys information that might otherwise be lost about the probability distribution represented by a region of uncertainty.

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 February 28, 2014.

Copyright Notice

Copyright (c) 2013 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. Introduction	2
1.1. Conventions used in this document	3
2. Representation of Confidence in PIDF-LO	3
2.1. Generating Locations with Confidence	4
2.2. Consuming and Presenting Confidence	4
3. Example	5
4. Confidence Schema	5
5. IANA Considerations	6
5.1. URN Sub-Namespace Registration for urn:ietf:params:xml:ns:geopriv:conf	6
5.2. XML Schema Registration	7
6. Security Considerations	7
7. References	7
7.1. Normative References	7
7.2. Informative References	8

1. Introduction

Location information is often less than perfect. Two measures are used to quantify how imperfect the location information is: uncertainty and confidence. These terms, and their relationship with location information are explored in detail in [I-D.thomson-geopriv-uncertainty]. Standard forms for the expression of uncertainty are included in [RFC5491], but confidence is fixed to a value of 95%.

On the whole, a fixed definition for confidence ensures consistency between implementations. Location generators that are aware of this constraint can generate location information at the required confidence. Location recipients are able to make sensible assumptions about the quality of the information that they receive.

In some circumstances - particularly with pre-existing systems - location generators might provide location information with some other confidence. Common values include 38%, 67% and 90%; all of which are prevalent in current systems. Existing forms of expressing location information, such as that defined in [TS-3GPP-23_032], contain elements that express the confidence in the result.

The addition of a confidence element provides information that was previously unavailable to recipients of location information. Without this information, a location server or generator that has access to location information with a confidence lower than 95% has two options:

- o The location server can scale regions of uncertainty in an attempt to achieve 95% confidence. This scaling process significantly degrades the quality of the information, because the location server might not have the necessary information to scale appropriately; the location server is forced to make assumptions that are likely result in either an overly conservative estimate with high uncertainty or a overestimate of confidence.
- o The location server can ignore the confidence entirely, which results in giving the recipient a false impression of its quality.

Both of these choices degrade the quality of the information provided.

The addition of a confidence element avoids this problem entirely if a location recipient supports and understands the element. A recipient that does not understand, and hence ignores, the confidence element is in no worse a position than if the location server ignored confidence.

1.1. Conventions used in this document

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 [RFC2119].

This document relies on the definitions in [I-D.thomson-geopriv-uncertainty] and [RFC3693].

2. Representation of Confidence in PIDF-LO

The confidence element MAY be added to the "location-info" element of the Presence Information Data Format - Location Object (PIDF-LO) [RFC4119] document. This element expresses the confidence in the associated location information as a percentage.

The confidence element optionally includes an attribute that indicates the shape of the probability density function (PDF) of the associated region of uncertainty. Three values are possible: unknown, normal and rectangular.

Indicating a particular PDF only indicates that the distribution approximately fits the given shape based on the methods used to generate the location information. The PDF is normal if there are a large number of small, independent sources of error; rectangular if all points within the area have roughly equal probability of being the actual location of the Target; otherwise, the PDF MUST either be set to unknown or omitted.

If a PIDF-LO does not include the confidence element, confidence is 95% [RFC5491]. A Point shape does not have uncertainty (or it has infinite uncertainty), so confidence is meaningless for a point; therefore, this element MUST be omitted if only a point is provided.

2.1. Generating Locations with Confidence

Location generators SHOULD attempt to ensure that confidence is equal in each dimension when generating location information. This restriction, while not always practical, allows for more accurate scaling, if scaling is necessary.

Confidence MUST NOT be included unless location information cannot be acquired with 95% confidence.

2.2. Consuming and Presenting Confidence

The inclusion of confidence that is anything other than 95% presents a potentially difficult usability for applications that use location information. Effectively communicating the probability that a location is incorrect to a user can be difficult.

It is inadvisable to simply display locations of any confidence, or to display confidence in a separate or non-obvious fashion. If locations with different confidence levels are displayed such that the distinction is subtle or easy to overlook - such as using fine graduations of color or transparency for graphical uncertainty regions, or displaying uncertainty graphically, but providing confidence as supplementary text - a user could fail to notice a difference in the quality of the location information that might be significant.

Depending on the circumstances, different ways of handling confidence might be appropriate. [I-D.thomson-geopriv-uncertainty] describes techniques that could be appropriate for consumers that use automated processing as well as background on the issue.

Providing that the full implications of any choice for the application are understood, some amount of automated processing could be appropriate. In a simple example, applications could choose to discard or suppress the display of location information if confidence does not meet a pre-determined threshold.

In settings where there is an opportunity for user training, some of these problems might be mitigated by defining different operational procedures for handling location information at different confidence levels.

3. Example

The PIDF-LO document in Figure 1 includes a representation of uncertainty as a circular area. The confidence element (on the line marked with a comment) indicates that the confidence is 67% and that it follows a normal distribution.

```
<pidf:presence
  xmlns:pidf="urn:ietf:params:xml:ns:pidf"
  xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
  xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
  xmlns:gs="http://www.opengis.net/pidflo/1.0"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:con="urn:ietf:params:xml:ns:pidf:geopriv:conf"
  entity="pres:alice@example.com">
  <dm:device id="sg89ab">
    <pidf:status>
      <gp:geopriv>
        <gp:location-info>
          <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:pos>42.5463 -73.2512</gml:pos>
            <gs:radius uom="urn:ogc:def:uom:EPSG::9001">
              850.24
            </gs:radius>
          </gs:Circle>
        <!-- c --> <con:confidence pdf="normal">67</con:confidence>
      </gp:location-info>
      <gp:usage-rules/>
    </gp:geopriv>
  </pidf:status>
  <dm:deviceID>mac:010203040506</dm:deviceID>
</dm:device>
</pidf:presence>
```

Figure 1: Example PIDF-LO with Confidence

4. Confidence Schema

```
<?xml version="1.0"?>
<xs:schema
  xmlns:conf="urn:ietf:params:xml:ns:geopriv:conf"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:ietf:params:xml:ns:geopriv:conf"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:appinfo
```

```
        source="urn:ietf:params:xml:schema:geopriv:conf">
    PIDF-LO Confidence
</xs:appinfo>
<xs:documentation source="http://www.ietf.org/rfc/rfcXXXX.txt">
    <!-- [[NOTE TO RFC-EDITOR: Please replace above URL with URL of
        published RFC and remove this note.]] -->
    This schema defines an element that is used for indicating
    confidence in PIDF-LO documents.
</xs:documentation>
</xs:annotation>

<xs:element name="confidence" type="conf:confidenceType"/>
<xs:complexType name="confidenceType">
    <xs:simpleContent>
        <xs:extension base="conf:confidenceBase">
            <xs:attribute name="pdf" type="conf:pdfType"
                default="unknown"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="confidenceBase">
    <xs:restriction base="xs:decimal">
        <xs:minExclusive value="0.0"/>
        <xs:maxExclusive value="100.0"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="pdfType">
    <xs:restriction base="xs:token">
        <xs:enumeration value="unknown"/>
        <xs:enumeration value="normal"/>
        <xs:enumeration value="rectangular"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>
```

5. IANA Considerations

5.1. URN Sub-Namespace Registration for urn:ietf:params:xml:ns:geopriv:conf

This section registers a new XML namespace,
"urn:ietf:params:xml:ns:geopriv:conf", as per the guidelines in
[RFC3688].

URI: urn:ietf:params:xml:ns:geopriv:conf

Registrant Contact: IETF, GEOPRIV working group,
(geopriv@ietf.org), Martin Thomson (martin.thomson@andrew.com).

XML:

```
BEGIN
  <?xml version="1.0"?>
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
    <head>
      <title>PIDF-LO Confidence Attribute</title>
    </head>
    <body>
      <h1>Namespace for PIDF-LO Confidence Attribute</h1>
      <h2>urn:ietf:params:xml:ns:geopriv:conf</h2>
      [[NOTE TO IANA/RFC-EDITOR: Please update RFC URL and replace XXXX
        with the RFC number for this specification.]]
      <p>See <a href="[[RFC URL]]">RFCXXXX</a>.</p>
    </body>
  </html>
END
```

5.2. XML Schema Registration

This section registers an XML schema as per the guidelines in [RFC3688].

URI: urn:ietf:params:xml:schema:geopriv:conf

Registrant Contact: IETF, GEOPRIV working group, (geopriv@ietf.org),
Martin Thomson (martin.thomson@andrew.com).

Schema: The XML for this schema can be found as the entirety of
Section 4 of this document.

6. Security Considerations

The security (and privacy) implications related to adding this
information are not significant.

7. References

7.1. Normative References

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

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.
- [RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.

7.2. Informative References

- [RFC3693] Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and J. Polk, "Geopriv Requirements", RFC 3693, February 2004.
- [I-D.thomson-geopriv-uncertainty]
Thomson, M. and J. Winterbottom, "Representation of Uncertainty and Confidence in PIDF-LO", draft-thomson-geopriv-uncertainty-07 (work in progress), March 2012.
- [RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.
- [TS-3GPP-23_032]
3GPP, "Universal Geographic Area Description (GAD)", 3GPP TS 23.032 11.0.0, September 2012.

Author's Address

Martin Thomson
Microsoft
Mountain View, VA 94043
US

Email: martin.thomson@gmail.com

GEOPRIV
Internet-Draft
Intended status: Informational
Expires: February 28, 2014

M. Thomson
Microsoft
J. Winterbottom
Andrew Corporation
August 27, 2013

Representation of Uncertainty and Confidence in PIDF-LO
draft-thomson-geopriv-uncertainty-08

Abstract

The key concepts of uncertainty and confidence as they pertain to location information are defined. Methods for the manipulation of location estimates that include uncertainty information are outlined.

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 February 28, 2014.

Copyright Notice

Copyright (c) 2013 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. Introduction	2
1.1. Conventions and Terminology	3
2. A General Definition of Uncertainty	3
2.1. Uncertainty as a Probability Distribution	4
2.2. Deprecation of the Terms Precision and Resolution	6
2.3. Accuracy as a Qualitative Concept	7
3. Uncertainty in Location	7
3.1. Representation of Uncertainty and Confidence in PIDF-LO	7
3.2. Uncertainty and Confidence for Civic Addresses	8
3.3. DHCP Location Configuration Information and Uncertainty	9
4. Manipulation of Uncertainty	9
4.1. Reduction of a Location Estimate to a Point	9
4.1.1. Centroid Calculation	10
4.1.1.1. Arc-Band Centroid	11
4.1.1.2. Polygon Centroid	11
4.2. Conversion to Circle or Sphere	14
4.3. Three-Dimensional to Two-Dimensional Conversion	14
4.4. Increasing and Decreasing Uncertainty and Confidence	15
4.4.1. Rectangular Distributions	16
4.4.2. Normal Distributions	16
4.5. Determining Whether a Location is Within a Given Region	17
4.5.1. Determining the Area of Overlap for Two Circles	18
4.5.2. Determining the Area of Overlap for Two Polygons	19
5. Examples	19
5.1. Reduction to a Point or Circle	19
5.2. Increasing and Decreasing Confidence	22
5.3. Matching Location Estimates to Regions of Interest	22
6. Security Considerations	22
7. Acknowledgements	23
8. Informative References	23
Appendix A. Conversion Between Cartesian and Geodetic Coordinates in WGS84	24
Appendix B. Calculating the Upward Normal of a Polygon	25
B.1. Checking that a Polygon Upward Normal Points Up	26
Authors' Addresses	27

1. Introduction

Location information represents an estimation of the position of a Target. Under ideal circumstances, a location estimate precisely reflects the actual location of the Target. In reality, there are many factors that introduce errors into the measurements that are used to determine location estimates.

The process by which measurements are combined to generate a location estimate is outside of the scope of work within the IETF. However,

the results of such a process are carried in IETF data formats and protocols. This document outlines how uncertainty, and its associated datum, confidence, are expressed and interpreted.

This document provides a common nomenclature for discussing uncertainty and confidence as they relate to location information.

This document also provides guidance on how to manage location information that includes uncertainty. Methods for expanding or reducing uncertainty to obtain a required level of confidence are described. Methods for determining the probability that a Target is within a specified region based on their location estimate are described. These methods are simplified by making certain assumptions about the location estimate and are designed to be applicable to location estimates in a relatively small area.

1.1. Conventions and Terminology

This document assumes a basic understanding of the principles of mathematics, particularly statistics and geometry.

Some terminology is borrowed from [RFC3693] and [RFC6280].

Mathematical formulae are presented using the following notation: add "+", subtract "-", multiply "*", divide "/", power "^" and absolute value "|x|". Precedence is indicated using parentheses. Mathematical functions are represented by common abbreviations: square root "sqrt(x)", sine "sin(x)", cosine "cos(x)", inverse cosine "acos(x)", tangent "tan(x)", inverse tangent "atan(x)", error function "erf(x)", and inverse error function "erfinv(x)".

2. A General Definition of Uncertainty

Uncertainty results from the limitations of measurement. In measuring any observable quantity, errors from a range of sources affect the result. Uncertainty is a quantification of what is known about the observed quantity, either through the limitations of measurement or through inherent variability of the quantity.

Uncertainty is most completely described by a probability distribution. A probability distribution assigns a probability to possible values for the quantity.

A probability distribution describing a measured quantity can be arbitrarily complex and so it is desirable to find a simplified model. One approach commonly taken is to reduce the probability distribution to a confidence interval. Many alternative models are used in other areas, but study of those is not the focus of this document.

In addition to the central estimate of the observed quantity, a confidence interval is succinctly described by two values: an error range and a confidence. The error range describes an interval and the confidence describes an estimated upper bound on the probability that a "true" value is found within the extents defined by the error.

In the following example, a measurement result for a length is shown as a nominal value with additional information on error range (0.0043 meters) and confidence (95%).

e.g. $x = 1.00742 \pm 0.0043$ meters at 95% confidence

This result indicates that the measurement indicates that the value of "x" between 1.00312 and 1.01172 meters with 95% probability. No other assertion is made: in particular, this does not assert that x is 1.00742.

This document uses the term `_uncertainty_` to refer in general to the concept as well as more specifically to refer to the error increment.

Uncertainty and confidence for location estimates can be derived in a number of ways. This document does not attempt to enumerate the many methods for determining uncertainty. [ISO.GUM] and [NIST.TN1297] provide a set of general guidelines for determining and manipulating measurement uncertainty. This document applies that general guidance for consumers of location information.

2.1. Uncertainty as a Probability Distribution

The Probability Density Function (PDF) that is described by uncertainty indicates the probability that the "true" value lies at any one point. The shape of the probability distribution can vary depending on the method that is used to determine the result. The two probability density functions most generally applicable most applicable to location information are considered in this document:

- o The normal PDF (also referred to as a Gaussian PDF) is used where a large number of small random factors contribute to errors. The value used for the error range in a normal PDF is related to the standard deviation of the distribution.

- o A rectangular PDF is used where the errors are known to be consistent across a limited range. A rectangular PDF can occur where a single error source, such as a rounding error, is significantly larger than other errors. A rectangular PDF is often described by the half-width of the distribution; that is, half the width of the distribution.

Each of these probability density functions can be characterized by its center point, or mean, and its width. For a normal distribution, uncertainty and confidence together are related to the standard deviation (see Section 4.4). For a rectangular distribution, half of the width of the distribution is used.

Figure 1 shows a normal and rectangular probability density function with the mean (m) and standard deviation (s) labelled. The half-width (h) of the rectangular distribution is also indicated.

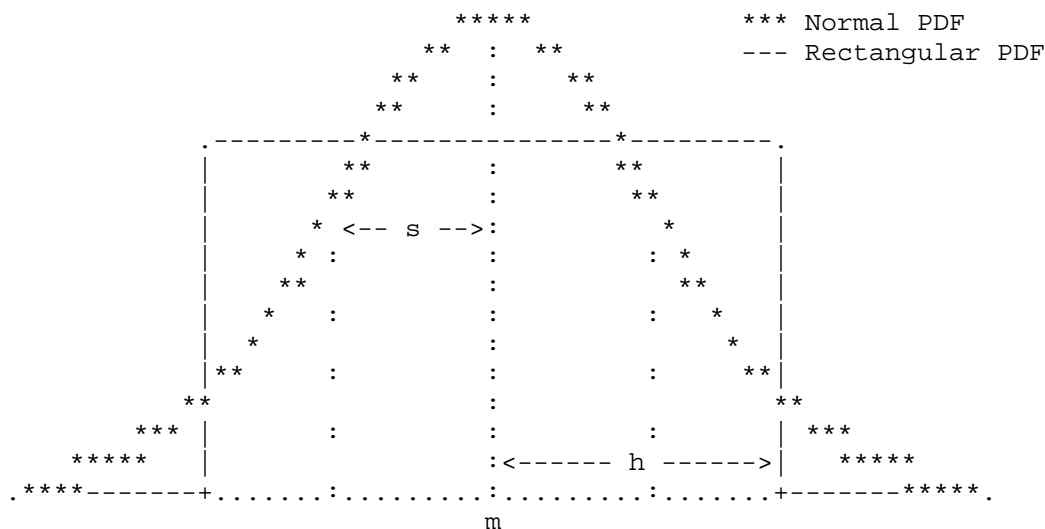


Figure 1: Normal and Rectangular Probability Density Functions

For a given PDF, the value of the PDF describes the probability that the "true" value is found at that point. Confidence for any given interval is the total probability of the "true" value being in that range, defined as the integral of the PDF over the interval.

The probability of the "true" value falling between two points is found by finding the area under the curve between the points (that is, the integral of the curve between the points). For any given PDF, the area under the curve for the entire range from negative infinity to positive infinity is 1 or (100%). Therefore, the

confidence over any interval of uncertainty is always less than 100%.

Figure 2 shows how confidence is determined for a normal distribution. The area of the shaded region gives the confidence (c) for the interval between "m-u" and "m+u".

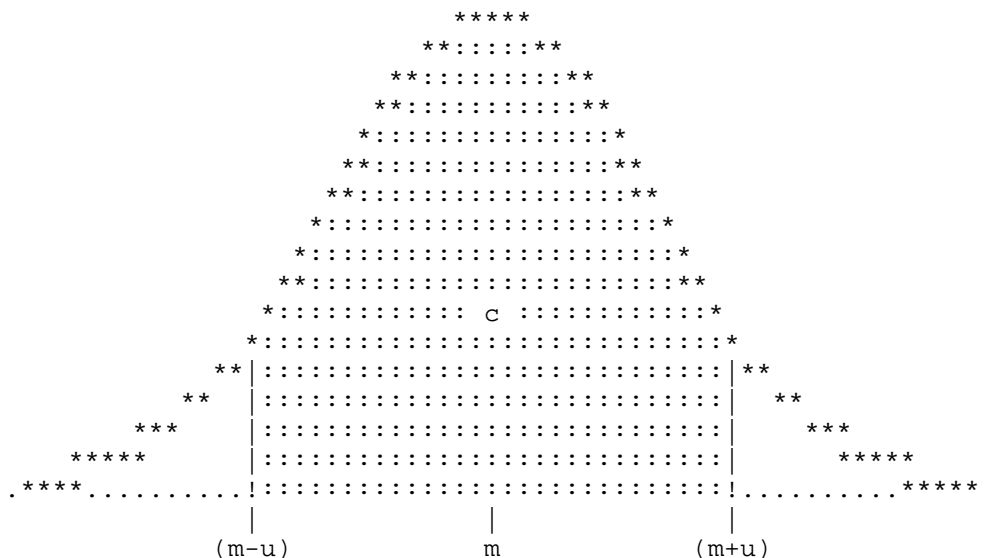


Figure 2: Confidence as the Integral of a PDF

In Section 4.4, methods are described for manipulating uncertainty if the shape of the PDF is known.

2.2. Deprecation of the Terms Precision and Resolution

The terms `_Precision_` and `_Resolution_` are defined in RFC 3693 [RFC3693]. These definitions were intended to provide a common nomenclature for discussing uncertainty; however, these particular terms have many different uses in other fields and their definitions are not sufficient to avoid confusion about their meaning. These terms are unsuitable for use in relation to quantitative concepts when discussing uncertainty and confidence in relation to location information.

2.3. Accuracy as a Qualitative Concept

Uncertainty is a quantitative concept. The term `_accuracy_` is useful in describing, qualitatively, the general concepts of location information. Accuracy is generally useful when describing qualitative aspects of location estimates. Accuracy is not a suitable term for use in a quantitative context.

For instance, it could be appropriate to say that a location estimate with uncertainty "X" is more accurate than a location estimate with uncertainty "2X" at the same confidence. It is not appropriate to assign a number to "accuracy", nor is it appropriate to refer to any component of uncertainty or confidence as "accuracy". That is, to say that the "accuracy" for the first location estimate is "X" would be an erroneous use of this term.

3. Uncertainty in Location

A `_location estimate_` is the result of location determination. A location estimate is subject to uncertainty like any other observation. However, unlike a simple measure of a one dimensional property like length, a location estimate is specified in two or three dimensions.

Uncertainty in 2- or 3-dimensional locations can be described using confidence intervals. The confidence interval for a location estimate in two or three dimensional space is expressed as a subset of that space. This document uses the term `_region of uncertainty_` to refer to the area or volume that describes the confidence interval.

Areas or volumes that describe regions of uncertainty can be formed by the combination of two or three one-dimensional ranges, or more complex shapes could be described.

3.1. Representation of Uncertainty and Confidence in PIDF-LO

A set of shapes suitable for the expression of uncertainty in location estimates in the Presence Information Data Format - Location Object (PIDF-LO) are described in [GeoShape]. These shapes are the recommended form for the representation of uncertainty in PIDF-LO [RFC4119] documents.

The PIDF-LO does not include an indication of confidence, but that confidence is 95%, by definition in [RFC5491]. Similarly, the PIDF-LO format does not provide an indication of the shape of the PDF.

Absence of uncertainty information in a PIDF-LO document does not indicate that there is no uncertainty in the location estimate. Uncertainty might not have been calculated for the estimate, or it may be withheld for privacy purposes.

If the Point shape is used, confidence and uncertainty are unknown; a receiver can either assume a confidence of 0% or infinite uncertainty. The same principle applies on the altitude axis for two-dimension shapes like the Circle.

3.2. Uncertainty and Confidence for Civic Addresses

Civic addresses [RFC5139] inherently include uncertainty, based on the area of the most precise element that is specified. Uncertainty is effectively defined by the presence or absence of elements -- elements that are not present are deemed to be uncertain.

To apply the concept of uncertainty to civic addresses, it is helpful to unify the conceptual models of civic address with geodetic location information.

Note: This view is one perspective on the process of geo-coding - the translation of a civic address to a geodetic location.

In the unified view, a civic address defines a series of (sometimes non-orthogonal) spatial partitions. The first is the implicit partition that identifies the surface of the earth and the space near the surface. The second is the country. Each label that is included in a civic address provides information about a different set of spatial partitions. Some partitions require slight adjustments from a standard interpretation: for instance, a road includes all properties that adjoin the street. Each label might need to be interpreted with other values to provide context.

As a value at each level is interpreted, one or more spatial partitions at that level are selected, and all other partitions of that type are excluded. For non-orthogonal partitions, only the portion of the partition that fits within the existing space is selected. This is what distinguishes King Street in Sydney from King Street in Melbourne. Each defined element selects a partition of space. The resulting location is the intersection of all selected spaces.

The resulting spatial partition can be considered to represent a region of uncertainty. At no stage does this process select a point; although, as spaces get smaller this distinction might have no practical significance and an approximation if a point could be used.

Uncertainty in civic addresses can be increased by removing elements. This doesn't necessarily improve confidence in the same way that arbitrarily increasing uncertainty in a geodetic location doesn't increase confidence.

3.3. DHCP Location Configuration Information and Uncertainty

Location information is often measured in two or three dimensions; expressions of uncertainty in one dimension only are rare. The "resolution" parameters in [RFC3825] provide an indication of uncertainty in one dimension.

[RFC3825] defines a means for representing uncertainty, but a value for confidence is not specified. A default value of 95% confidence can be assumed for the combination of the uncertainty on each axis. That is, the confidence of the resultant rectangular polygon or prism is 95%.

4. Manipulation of Uncertainty

This section deals with manipulation of location information that contains uncertainty.

The following rules generally apply when manipulating location information:

- o Where calculations are performed on coordinate information, these should be performed in Cartesian space and the results converted back to latitude, longitude and altitude. A method for converting to and from Cartesian coordinates is included in Appendix A.

While some approximation methods are useful in simplifying calculations, treating latitude and longitude as Cartesian axes is never advisable. The two axes are not orthogonal. Errors can arise from the curvature of the earth and from the convergence of longitude lines.

- o Normal rounding rules do not apply when rounding uncertainty. When rounding, the region of uncertainty always increases (that is, errors are rounded up) and confidence is always rounded down (see [NIST.TN1297]). This means that any manipulation of uncertainty is a non-reversible operation; each manipulation can result in the loss of some information.

4.1. Reduction of a Location Estimate to a Point

Manipulating location estimates that include uncertainty information requires additional complexity in systems. In some cases, systems only operate on definitive values, that is, a single point.

This section describes algorithms for reducing location estimates to a simple form without uncertainty information. Having a consistent means for reducing location estimates allows for interaction between applications that are able to use uncertainty information and those that cannot.

Note: Reduction of a location estimate to a point constitutes a reduction in information. Removing uncertainty information can degrade results in some applications. Also, there is a natural tendency to misinterpret a point location as representing a location without uncertainty. This could lead to more serious errors. Therefore, these algorithms should only be applied where necessary.

Several different approaches can be taken when reducing a location estimate to a point. Different methods each make a set of assumptions about the properties of the PDF and the selected point; no one method is more "correct" than any other. For any given region of uncertainty, selecting an arbitrary point within the area could be considered valid; however, given the aforementioned problems with point locations, a more rigorous approach is appropriate.

Given a result with a known distribution, selecting the point within the area that has the highest probability is a more rigorous method. Alternatively, a point could be selected that minimizes the overall error; that is, it minimises the expected value of the difference between the selected point and the "true" value.

If a rectangular distribution is assumed, the centroid of the area or volume minimizes the overall error. Minimizing the error for a normal distribution is mathematically complex. Therefore, this document opts to select the centroid of the region of uncertainty when selecting a point.

4.1.1. Centroid Calculation

For regular shapes, such as Circle, Sphere, Ellipse and Ellipsoid, this approach equates to the center point of the region. For regions of uncertainty that are expressed as regular Polygons and Prisms the center point is also the most appropriate selection.

For the Arc-Band shape and non-regular Polygons and Prisms, selecting the centroid of the area or volume minimizes the overall error. This assumes that the PDF is rectangular.

Note: The centroid of a concave Polygon or Arc-Band shape is not necessarily within the region of uncertainty.

4.1.1.1. Arc-Band Centroid

The centroid of the Arc-Band shape is found along a line that bisects the arc. The centroid can be found at the following distance from the starting point of the arc-band (assuming an arc-band with an inner radius of "r", outer radius "R", start angle "a", and opening angle "o"):

$$d = 4 * \sin(o/2) * (R*R + R*r + r*r) / (3*o*(R + r))$$

This point can be found along the line that bisects the arc; that is, the line at an angle of "a + (o/2)". Negative values are possible if the angle of opening is greater than 180 degrees; negative values indicate that the centroid is found along the angle "a + (o/2) + 180".

4.1.1.2. Polygon Centroid

Calculating a centroid for the Polygon and Prism shapes is more complex. Polygons that are specified using geodetic coordinates are not necessarily coplanar. For Polygons that are specified without an altitude, choose a value for altitude before attempting this process; an altitude of 0 is acceptable.

The method described in this section is simplified by assuming that the surface of the earth is locally flat. This method degrades as polygons become larger; see [GeoShape] for recommendations on polygon size.

The polygon is translated to a new coordinate system that has an x-y plane roughly parallel to the polygon. This enables the elimination of z-axis values and calculating a centroid can be done using only x and y coordinates. This requires that the upward normal for the polygon is known.

To translate the polygon coordinates, apply the process described in Appendix B to find the normal vector "N = [Nx,Ny,Nz]". This value should be made a unit vector to ensure that the transformation matrix is a special orthogonal matrix. From this vector, select two vectors that are perpendicular to this vector and combine these into a transformation matrix.

If "Nx" and "Ny" are non-zero, the matrices in Figure 3 can be used, given "p = sqrt(Nx^2 + Ny^2)". More transformations are provided later in this section for cases where "Nx" or "Ny" are zero.

$$\begin{array}{ccc}
 \begin{bmatrix} -Ny/p & Nx/p & 0 \\ -Nx*Nz/p & -Ny*Nz/p & p \\ Nx & Ny & Nz \end{bmatrix} & & \begin{bmatrix} -Ny/p & -Nx*Nz/p & Nx \\ Nx/p & -Ny*Nz/p & Ny \\ 0 & p & Nz \end{bmatrix} \\
 \text{(Transform)} & & \text{(Reverse Transform)}
 \end{array}$$

Figure 3: Recommended Transformation Matrices

To apply a transform to each point in the polygon, form a matrix from the ECEF coordinates and use matrix multiplication to determine the translated coordinates.

$$\begin{bmatrix} -Ny/p & Nx/p & 0 \\ -Nx*Nz/p & -Ny*Nz/p & p \\ Nx & Ny & Nz \end{bmatrix} * \begin{bmatrix} x[1] & x[2] & x[3] & \dots & x[n] \\ y[1] & y[2] & y[3] & \dots & y[n] \\ z[1] & z[2] & z[3] & \dots & z[n] \end{bmatrix} \\
 = \begin{bmatrix} x'[1] & x'[2] & x'[3] & \dots & x'[n] \\ y'[1] & y'[2] & y'[3] & \dots & y'[n] \\ z'[1] & z'[2] & z'[3] & \dots & z'[n] \end{bmatrix}$$

Figure 4: Transformation

Alternatively, direct multiplication can be used to achieve the same result:

$$\begin{aligned}
 x'[i] &= -Ny * x[i] / p + Nx * y[i] / p \\
 y'[i] &= -Nx * Nz * x[i] / p - Ny * Nz * y[i] / p + p * z[i] \\
 z'[i] &= Nx * x[i] + Ny * y[i] + Nz * z[i]
 \end{aligned}$$

The first and second rows of this matrix ("x'" and "y'") contain the values that are used to calculate the centroid of the polygon. To find the centroid of this polygon, first find the area using:

$$A = \text{sum from } i=1..n \text{ of } (x'[i]*y'[i+1]-x'[i+1]*y'[i]) / 2$$

For these formulae, treat each set of coordinates as circular, that is "x'[0] == x'[n]" and "x'[n+1] == x'[1]". Based on the area, the centroid along each axis can be determined by:

$$\begin{aligned}
 Cx' &= \text{sum } (x'[i]+x'[i+1]) * (x'[i]*y'[i+1]-x'[i+1]*y'[i]) / (6*A) \\
 Cy' &= \text{sum } (y'[i]+y'[i+1]) * (x'[i]*y'[i+1]-x'[i+1]*y'[i]) / (6*A)
 \end{aligned}$$

Note: The formula for the area of a polygon will return a negative value if the polygon is specified in clockwise direction. This can be used to determine the orientation of the polygon.

The third row contains a distance from a plane parallel to the polygon. If the polygon is coplanar, then the values for "z'" are identical; however, the constraints recommended in [RFC5491] mean that this is rarely the case. To determine "Cz'", average these values:

$$Cz' = \text{sum } z'[i] / n$$

Once the centroid is known in the transformed coordinates, these can be transformed back to the original coordinate system. The reverse transformation is shown in Figure 5.

$$\begin{bmatrix} -Ny/p & -Nx*Nz/p & Nx \\ Nx/p & -Ny*Nz/p & Ny \\ 0 & p & Nz \end{bmatrix} * \begin{bmatrix} Cx' \\ Cy' \\ \text{sum of } z'[i] / n \end{bmatrix} = \begin{bmatrix} Cx \\ Cy \\ Cz \end{bmatrix}$$

Figure 5: Reverse Transformation

The reverse transformation can be applied directly as follows:

$$Cx = -Ny * Cx' / p - Nx * Nz * Cy' / p + Nx * Cz'$$

$$Cy = Nx * Cx' / p - Ny * Nz * Cy' / p + Ny * Cz'$$

$$Cz = p * Cy' + Nz * Cz'$$

The ECEF value "[Cx,Cy,Cz]" can then be converted back to geodetic coordinates. Given a polygon that is defined with no altitude or equal altitudes for each point, the altitude of the result can either be ignored or reset after converting back to a geodetic value.

The centroid of the Prism shape is found by finding the centroid of the base polygon and raising the point by half the height of the prism. This can be added to altitude of the final result; alternatively, this can be added to "Cz'", which ensures that negative height is correctly applied to polygons that are defined in a "clockwise" direction.

The recommended transforms only apply if "Nx" and "Ny" are non-zero. If the normal vector is "[0,0,1]" (that is, along the z-axis), then no transform is necessary. Similarly, if the normal vector is "[0,1,0]" or "[1,0,0]", avoid the transformation and use the x and z coordinates or y and z coordinates (respectively) in the centroid calculation phase. If either "Nx" or "Ny" are zero, the alternative transform matrices in Figure 6 can be used. The reverse transform is the transpose of this matrix.

if Nx == 0:		if Ny == 0:
$T = \begin{bmatrix} 0 & -Nz & Ny \\ 1 & 0 & 0 \\ 0 & Ny & Nz \end{bmatrix}$	$T' = \begin{bmatrix} 0 & 1 & 0 \\ -Nz & 0 & Ny \\ Ny & 0 & Nz \end{bmatrix}$	$T = T' = \begin{bmatrix} -Nz & 0 & Nx \\ 0 & 1 & 0 \\ Nx & 0 & Nz \end{bmatrix}$

Figure 6: Alternative Transformation Matrices

4.2. Conversion to Circle or Sphere

The Circle or Sphere are simple shapes that suit a range of applications. A circle or sphere contains fewer units of data to manipulate, which simplifies operations on location estimates.

The simplest method for converting a location estimate to a Circle or Sphere shape is to determine the centroid and then find the longest distance to any point in the region of uncertainty to that point. This distance can be determined based on the shape type:

Circle/Sphere: No conversion necessary.

Ellipse/Ellipsoid: The greater of either semi-major axis or altitude uncertainty.

Polygon/Prism: The distance to the furthest vertex of the polygon (for a Prism, it is only necessary to check points on the base).

Arc-Band: The furthest length from the centroid to the points where the inner and outer arc end. This distance can be calculated by finding the larger of the two following formulae:

$$X = \sqrt{d*d + R*R - 2*d*R*\cos(o/2)}$$

$$x = \sqrt{d*d + r*r - 2*d*r*\cos(o/2)}$$

Once the Circle or Sphere shape is found, the associated confidence can be increased if the result is known to follow a normal distribution. However, this is a complicated process and provides limited benefit. In many cases it also violates the constraint that confidence in each dimension be the same. Confidence should be unchanged when performing this conversion.

Two dimensional shapes are converted to a Circle; three dimensional shapes are converted to a Sphere.

4.3. Three-Dimensional to Two-Dimensional Conversion

A three-dimensional shape can be easily converted to a two-dimensional shape by removing the altitude component. A sphere

becomes a circle; a prism becomes a polygon; an ellipsoid becomes an ellipse. Each conversion is simple, requiring only the removal of those elements relating to altitude.

The altitude is unspecified for a two-dimensional shape and therefore has unlimited uncertainty along the vertical axis. The confidence for the two-dimensional shape is thus higher than the three-dimensional shape. Assuming equal confidence on each axis, the confidence of the circle can be increased using the following approximate formula:

$$C[2d] \geq C[3d]^{(2/3)}$$

"C[2d]" is the confidence of the two-dimensional shape and "C[3d]" is the confidence of the three-dimensional shape. For example, a Sphere with a confidence of 95% can be simplified to a Circle of equal radius with confidence of 96.6%.

4.4. Increasing and Decreasing Uncertainty and Confidence

The combination of uncertainty and confidence provide a great deal of information about the nature of the data that is being measured. If both uncertainty, confidence and PDF are known, certain information can be extrapolated. In particular, the uncertainty can be scaled to meet a desired confidence or the confidence for a particular region of uncertainty can be found.

In general, confidence decreases as the region of uncertainty decreases in size and confidence increases as the region of uncertainty increases in size. However, this depends on the PDF; expanding the region of uncertainty for a rectangular distribution has no effect on confidence without additional information. If the region of uncertainty is increased during the process of obfuscation (see [I-D.thomson-geopriv-location-obscuring]), then the confidence cannot be increased.

A region of uncertainty that is reduced in size always has a lower confidence.

A region of uncertainty that has an unknown PDF shape cannot be reduced in size reliably. The region of uncertainty can be expanded, but only if confidence is not increased.

This section makes the simplifying assumption that location information is symmetrically and evenly distributed in each dimension. This is not necessarily true in practice. If better information is available, alternative methods might produce better results.

4.4.1. Rectangular Distributions

Uncertainty that follows a rectangular distribution can only be decreased in size. Since the PDF is constant over the region of uncertainty, the resulting confidence is determined by the following formula:

$$Cr = Co * Ur / Uo$$

Where "Uo" and "Ur" are the sizes of the original and reduced regions of uncertainty (either the area or the volume of the region); "Co" and "Cb" are the confidence values associated with each region.

Information is lost by decreasing the region of uncertainty for a rectangular distribution. Once reduced in size, the uncertainty region cannot subsequently be increased in size.

4.4.2. Normal Distributions

Uncertainty and confidence can be both increased and decreased for a normal distribution. However, the process is more complicated.

For a normal distribution, uncertainty and confidence are related to the standard deviation of the function. The following function defines the relationship between standard deviation, uncertainty and confidence along a single axis:

$$S[x] = U[x] / (\text{sqrt}(2) * \text{erfinv}(C[x]))$$

Where "S[x]" is the standard deviation, "U[x]" is the uncertainty and "C[x]" is the confidence along a single axis. "erfinv" is the inverse error function.

Scaling a normal distribution in two dimensions requires several assumptions. Firstly, it is assumed that the distribution along each axis is independent. Secondly, the confidence for each axis is the same. Therefore, the confidence along each axis can be assumed to be:

$$C[x] = Co ^ (1/n)$$

Where "C[x]" is the confidence along a single axis and "Co" is the overall confidence and "n" is the number of dimensions in the uncertainty.

Therefore, to find the uncertainty for each axis at a desired confidence, "Cd", apply the following formula:

$$U_d[x] \leq U[x] * (\text{erfinv}(C_d \wedge (1/n)) / \text{erfinv}(C_o \wedge (1/n)))$$

For regular shapes, this formula can be applied as a scaling factor in each dimension to reach a required confidence.

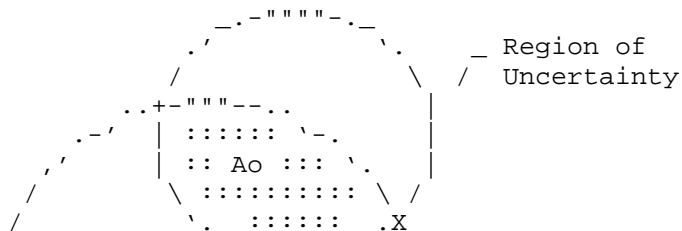
4.5. Determining Whether a Location is Within a Given Region

A number of applications require that a judgement be made about whether a Target is within a given region of interest. Given a location estimate with uncertainty, this judgement can be difficult. A location estimate represents a probability distribution, and the true location of the Target cannot be definitively known. Therefore, the judgement relies on determining the probability that the Target is within the region.

The probability that the Target is within a particular region is found by integrating the PDF over the region. For a normal distribution, there are no analytical methods that can be used to determine the integral of the two or three dimensional PDF over an arbitrary region. The complexity of numerical methods is also too great to be useful in many applications; for example, finding the integral of the PDF in two or three dimensions across the overlap between the uncertainty region and the target region. If the PDF is unknown, no determination can be made. When judging whether a location is within a given region, uncertainties using these PDFs can be assumed to be rectangular. If this assumption is made, the confidence should be scaled to 95%, if possible.

Note: The selection of confidence has a significant impact on the final result. Only use a different confidence if an uncertainty value for 95% confidence cannot be found.

Given the assumption of a rectangular distribution, the probability that a Target is found within a given region is found by first finding the area (or volume) of overlap between the uncertainty region and the region of interest. This is multiplied by the confidence of the location estimate to determine the probability. Figure 7 shows an example of finding the area of overlap between the region of uncertainty and the region of interest.



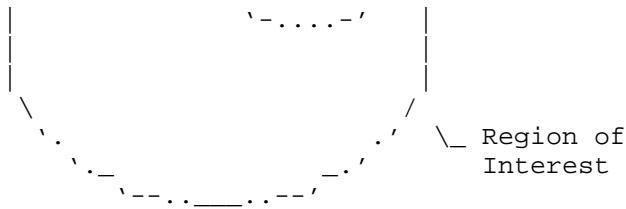


Figure 7: Area of Overlap Between Two Circular Regions

Once the area of overlap, "Ao", is known, the probability that the Target is within the region of interest, "Pi", is:

$$Pi = Co * Ao / Au$$

Given that the area of the region of uncertainty is "Au" and the confidence is "Co".

This probability is often input to a decision process that has a limited set of outcomes; therefore, a threshold value needs to be selected. Depending on the application, different threshold probabilities might be selected. In the absence of specific recommendations, this document suggests that the probability be greater than 50% before a decision is made. If the decision process selects between two or more regions, as is required by [RFC5222], then the region with the highest probability can be selected.

4.5.1. Determining the Area of Overlap for Two Circles

Determining the area of overlap between two arbitrary shapes is a non-trivial process. Reducing areas to circles (see Section 4.2) enables the application of the following process.

Given the radius of the first circle "r", the radius of the second circle "R" and the distance between their center points "d", the following set of formulas provide the area of overlap "Ao".

- o If the circles don't overlap, that is " $d \geq r+R$ ", "Ao" is zero.
- o If one of the two circles is entirely within the other, that is " $d \leq |r-R|$ ", the area of overlap is the area of the smaller circle.
- o Otherwise, if the circles partially overlap, that is " $d < r+R$ " and " $d > |r-R|$ ", find "Ao" using:

$$a = (r^2 - R^2 + d^2)/(2*d)$$

$$A_o = r^2 \cdot \text{acos}(a/r) + R^2 \cdot \text{acos}((d - a)/R) - d \cdot \text{sqrt}(r^2 - a^2)$$

A value for "d" can be determined by converting the center points to Cartesian coordinates and calculating the distance between the two center points:

$$d = \text{sqrt}((x1-x2)^2 + (y1-y2)^2 + (z1-z2)^2)$$

4.5.2. Determining the Area of Overlap for Two Polygons

A calculation of overlap based on polygons can give better results than the circle-based method. However, efficient calculation of overlapping area is non-trivial. Algorithms such as Vatti's clipping algorithm [Vatti92] can be used.

For large polygonal areas, it might be that geodesic interpolation is used. In these cases, altitude is also frequently omitted in describing the polygon. For such shapes, a planar projection can still give a good approximation of the area of overlap if the larger area polygon is projected onto the local tangent plane of the smaller. This is only possible if the only area of interest is that contained within the smaller polygon. Where the entire area of the larger polygon is of interest, geodesic interpolation is necessary.

5. Examples

This section presents some examples of how to apply the methods described in Section 4.

5.1. Reduction to a Point or Circle

Alice receives a location estimate from her LIS that contains a ellipsoidal region of uncertainty. This information is provided at 19% confidence with a normal PDF. A PIDF-LO extract for this information is shown in Figure 8.

```
<gp:geopriv>
  <gp:location-info>
    <gs:Ellipsoid srsName="urn:ogc:def:crs:EPSG::4979">
      <gml:pos>-34.407242 150.882518 34</gml:pos>
      <gs:semiMajorAxis uom="urn:ogc:def:uom:EPSG::9001">
        7.7156
      </gs:semiMajorAxis>
      <gs:semiMinorAxis uom="urn:ogc:def:uom:EPSG::9001">
        3.31
      </gs:semiMinorAxis>
      <gs:verticalAxis uom="urn:ogc:def:uom:EPSG::9001">
        28.7
```

```

    </gs:verticalAxis>
    <gs:orientation uom="urn:ogc:def:uom:EPSG::9102">
      43
    </gs:orientation>
  </gs:Ellipsoid>
</gp:location-info>
<gp:usage-rules/>
</gp:geopriv>

```

Figure 8

This information can be reduced to a point simply by extracting the center point, that is [-34.407242, 150.882518, 34].

If some limited uncertainty were required, the estimate could be converted into a circle or sphere. To convert to a sphere, the radius is the largest of the semi-major, semi-minor and vertical axes; in this case, 28.7 meters.

However, if only a circle is required, the altitude can be dropped as can the altitude uncertainty (the vertical axis of the ellipsoid), resulting in a circle at [-34.407242, 150.882518] of radius 7.7156 meters.

Bob receives a location estimate with a Polygon shape. This information is shown in Figure 9.

```

<gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">
  <gml:exterior>
    <gml:LinearRing>
      <gml:posList>
        -33.856625 151.215906 -33.856299 151.215343
        -33.856326 151.214731 -33.857533 151.214495
        -33.857720 151.214613 -33.857369 151.215375
        -33.856625 151.215906
      </gml:posList>
    </gml:LinearRing>
  </gml:exterior>
</gml:Polygon>

```

Figure 9

To convert this to a polygon, each point is firstly assigned an altitude of zero and converted to ECEF coordinates (see Appendix A). Then a normal vector for this polygon is found (see Appendix B). The results of each of these stages is shown in Figure 10. Note that the numbers shown are all rounded; no rounding is possible during this process since rounding would contribute significant errors.

```

Polygon in ECEF coordinate space
(repeated point omitted and transposed to fit):
      [ -4.6470e+06  2.5530e+06 -3.5333e+06 ]
      [ -4.6470e+06  2.5531e+06 -3.5332e+06 ]
pecef = [ -4.6470e+06  2.5531e+06 -3.5332e+06 ]
      [ -4.6469e+06  2.5531e+06 -3.5333e+06 ]
      [ -4.6469e+06  2.5531e+06 -3.5334e+06 ]
      [ -4.6469e+06  2.5531e+06 -3.5333e+06 ]

Normal Vector: n = [ -0.72782  0.39987 -0.55712 ]

Transformation Matrix:
      [ -0.48152 -0.87643  0.00000 ]
t = [ -0.48828  0.26827  0.83043 ]
      [ -0.72782  0.39987 -0.55712 ]

Transformed Coordinates:
      [ 8.3206e+01  1.9809e+04  6.3715e+06 ]
      [ 3.1107e+01  1.9845e+04  6.3715e+06 ]
pecef' = [ -2.5528e+01  1.9842e+04  6.3715e+06 ]
      [ -4.7367e+01  1.9708e+04  6.3715e+06 ]
      [ -3.6447e+01  1.9687e+04  6.3715e+06 ]
      [ 3.4068e+01  1.9726e+04  6.3715e+06 ]

Two dimensional polygon area: A = 12600 m^2
Two-dimensional polygon centroid: C' = [ 8.8184e+00  1.9775e+04 ]

Average of pecef' z coordinates: 6.3715e+06

Reverse Transformation Matrix:
      [ -0.48152 -0.48828 -0.72782 ]
t' = [ -0.87643  0.26827  0.39987 ]
      [ 0.00000  0.83043 -0.55712 ]

Polygon centroid (ECEF): C = [ -4.6470e+06  2.5531e+06 -3.5333e+06 ]
Polygon centroid (Geo): Cg = [ -33.856926  151.215102 -4.9537e-04 ]

```

Figure 10

The point conversion for the polygon uses the final result, "Cg", ignoring the altitude since the original shape did not include altitude.

To convert this to a circle, take the maximum distance in ECEF coordinates from the center point to each of the points. This results in a radius of 99.1 meters. Confidence is unchanged.

5.2. Increasing and Decreasing Confidence

Assuming that confidence is known to be 19% for Alice's location information. This is typical value for a three-dimensional ellipsoid uncertainty of normal distribution where the standard deviation is supplied in each dimension. The confidence associated with Alice's location estimate is quite low for many applications. Since the estimate is known to follow a normal distribution, the method in Section 4.4.2 can be used. Each axis can be scaled by:

$$\text{scale} = \text{erfinv}(0.95^{(1/3)}) / \text{erfinv}(0.19^{(1/3)}) = 2.9937$$

Ensuring that rounding always increases uncertainty, the location estimate at 95% includes a semi-major axis of 23.1, a semi-minor axis of 10 and a vertical axis of 86.

Bob's location estimate covers an area of approximately 12600 square meters. If the estimate follows a rectangular distribution, the region of uncertainty can be reduced in size. To find the confidence that he is within the smaller area of the concert hall, given by the polygon [-33.856473, 151.215257; -33.856322, 151.214973; -33.856424, 151.21471; -33.857248, 151.214753; -33.857413, 151.214941; -33.857311, 151.215128]. To use this new region of uncertainty, find its area using the same translation method described in Section 4.1.1.2, which is 4566.2 square meters. The confidence associated with the smaller area is therefore $95\% * 4566.2 / 12600 = 34\%$.

5.3. Matching Location Estimates to Regions of Interest

Suppose than a circular area is defined centered at [-33.872754, 151.20683] with a radius of 1950 meters. To determine whether Bob is found within this area, we apply the method in Section 4.5. Using the converted Circle shape for Bob's location, the distance between these points is found to be 1915.26 meters. The area of overlap between Bob's location estimate and the region of interest is therefore 2209 square meters and the area of Bob's location estimate is 30853 square meters. This gives the probability that Bob is less than 1950 meters from the selected point as 67.8%.

Note that if 1920 meters were chosen for the distance from the selected point, the area of overlap is only 16196 square meters and the confidence is 49.8%. Therefore, it is marginally more likely that Bob is outside the region of interest, despite the center point of his location estimate being within the region.

6. Security Considerations

This document describes methods for managing and manipulating uncertainty in location. No specific security concerns arise from most of the information provided.

7. Acknowledgements

Peter Rhodes provided assistance with some of the mathematical groundwork on this document. Dan Cornford provided a detailed review and many terminology corrections.

8. Informative References

[Convert] Burtch, R., "A Comparison of Methods Used in Rectangular to Geodetic Coordinate Transformations", April 2006.

[GeoShape] Thomson, M. and C. Reed, "GML 3.1.1 PIDF-LO Shape Application Schema for use by the Internet Engineering Task Force (IETF)", Candidate OpenGIS Implementation Specification 06-142r1, Version: 1.0, April 2007.

[I-D.thomson-geopriv-location-obscuring] Thomson, M., "Obscuring Location", draft-thomson-geopriv-location-obscuring-03 (work in progress), June 2011.

[ISO.GUM] ISO/IEC, "Guide to the expression of uncertainty in measurement (GUM) ", Guide 98:1995, 1995.

[NIST.TN1297] Taylor, B. and C. Kuyatt, "Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results ", Technical Note 1297, Sep 1994.

[RFC3693] Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and J. Polk, "Geopriv Requirements", RFC 3693, February 2004.

[RFC3694] Danley, M., Mulligan, D., Morris, J., and J. Peterson, "Threat Analysis of the Geopriv Protocol", RFC 3694, February 2004.

[RFC3825] Polk, J., Schnizlein, J., and M. Linsner, "Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information", RFC 3825, July 2004.

[RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.

- [RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", RFC 5139, February 2008.
- [RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, August 2008.
- [RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.
- [RFC6280] Barnes, R., Lepinski, M., Cooper, A., Morris, J., Tschofenig, H., and H. Schulzrinne, "An Architecture for Location and Location Privacy in Internet Applications", BCP 160, RFC 6280, July 2011.
- [Sunday02] Sunday, D., "Fast polygon area and Newell normal computation ", Journal of Graphics Tools JGT, 7(2):9-13,2002, 2002,
<<http://www.acm.org/jgt/papers/Sunday02/>>.
- [Vatti92] Vatti, B., "A generic solution to polygon clipping ", Communications of the ACM Vol35, Issue7, pp56-63, 1992,
<<http://portal.acm.org/citation.cfm?id=129906> >.
- [WGS84] US National Imagery and Mapping Agency, "Department of Defense (DoD) World Geodetic System 1984 (WGS 84), Third Edition ", NIMA TR8350.2, January 2000.

Appendix A. Conversion Between Cartesian and Geodetic Coordinates in WGS84

The process of conversion from geodetic (latitude, longitude and altitude) to earth-centered, earth-fixed (ECEF) Cartesian coordinates is relatively simple.

In this section, the following constants and derived values are used from the definition of WGS84 [WGS84]:

{radius of ellipsoid} $R = 6378137$ meters

{inverse flattening} $1/f = 298.257223563$

{first eccentricity squared} $e^2 = f * (2 - f)$

{second eccentricity squared} $e'^2 = e^2 * (1 - e^2)$

To convert geodetic coordinates (latitude, longitude, altitude) to ECEF coordinates (X, Y, Z), use the following relationships:

```
N = R / sqrt(1 - e^2 * sin(latitude)^2)

X = (N + altitude) * cos(latitude) * cos(longitude)

Y = (N + altitude) * cos(latitude) * sin(longitude)

Z = (N*(1 - e^2) + altitude) * sin(latitude)
```

The reverse conversion requires more complex computation and most methods introduce some error in latitude and altitude. A range of techniques are described in [Convert]. A variant on the method originally proposed by Bowring, which results in an acceptably small error, is described by the following:

```
p = sqrt(X^2 + Y^2)

r = sqrt(X^2 + Y^2 + Z^2)

u = atan((1-f) * Z * (1 + e'^2 * (1-f) * R / r) / p)

latitude = atan((Z + e'^2 * (1-f) * R * sin(u)^3) /
  (p - e^2 * R * cos(u)^3))

longitude = atan(Y / X)

altitude = sqrt((p - R * cos(u))^2 + (Z - (1-f) * R * sin(u))^2)
```

If the point is near the poles, that is " $p < 1$ ", the value for altitude that this method produces is unstable. A simpler method for determining the altitude of a point near the poles is:

```
altitude = |Z| - R * (1 - f)
```

Appendix B. Calculating the Upward Normal of a Polygon

For a polygon that is guaranteed to be convex and coplanar, the upward normal can be found by finding the vector cross product of adjacent edges.

For more general cases the Newell method of approximation described in [Sunday02] may be applied. In particular, this method can be used if the points are only approximately coplanar, and for non-convex polygons.

This process requires a Cartesian coordinate system. Therefore, convert the geodetic coordinates of the polygon to Cartesian, ECEF coordinates (Appendix A). If no altitude is specified, assume an altitude of zero.

This method can be condensed to the following set of equations:

$$N_x = \text{sum from } i=1..n \text{ of } (y[i] * (z[i+1] - z[i-1]))$$

$$N_y = \text{sum from } i=1..n \text{ of } (z[i] * (x[i+1] - x[i-1]))$$

$$N_z = \text{sum from } i=1..n \text{ of } (x[i] * (y[i+1] - y[i-1]))$$

For these formulae, the polygon is made of points "(x[1], y[1], z[1])" through "(x[n], y[n], z[n])". Each array is treated as circular, that is, "x[0] == x[n]" and "x[n+1] == x[1]".

To translate this into a unit-vector; divide each component by the length of the vector:

$$N_x' = N_x / \text{sqrt}(N_x^2 + N_y^2 + N_z^2)$$

$$N_y' = N_y / \text{sqrt}(N_x^2 + N_y^2 + N_z^2)$$

$$N_z' = N_z / \text{sqrt}(N_x^2 + N_y^2 + N_z^2)$$

B.1. Checking that a Polygon Upward Normal Points Up

RFC 5491 [RFC5491] stipulates that polygons be presented in anti-clockwise direction so that the upward normal is in an upward direction. Accidental reversal of points can invert this vector. This error can be hard to detect just by looking at the series of coordinates that form the polygon.

Calculate the dot product of the upward normal of the polygon (Appendix B) and any vector that points away from the center of the Earth from the location of polygon. If this product is positive, then the polygon upward normal also points away from the center of the Earth.

The inverse cosine of this value indicates the angle between the horizontal plane and the approximate plane of the polygon.

A unit vector for the upward direction at any point can be found based on the latitude (lat) and longitude (lng) of the point, as follows:

$$Up = [\cos(lat) * \cos(lng) ; \cos(lat) * \sin(lng) ; \sin(lat)]$$

For polygons that span less than half the globe, any point in the polygon - including the centroid - can be selected to generate an approximate up vector for comparison with the upward normal.

Authors' Addresses

Martin Thomson
Microsoft
3210 Porter Drive
Palo Alto, CA 94304
US

Email: martin.thomson@gmail.com

James Winterbottom
Andrew Corporation
Andrew Building (39)
Wollongong University Campus
Northfields Avenue
Wollongong, NSW 2522
AU

Email: james.winterbottom@andrew.com