ECRIT	M. Thomson	
Internet-Draft	Andrew Corporation	
Intended status: Informational	K. Wolf	
Expires: August 5, 2011	nic.at GmbH	
	February 1, 2011	

TOC

Describing Boundaries for Civic Addresses
draft-thomson-ecrit-civic-boundary-02

Abstract

Algorithms for decision-making based on civic address inputs are described. This includes an algorithm for determining whether one civic address is entirely contained within another. Other algorithms and supplementary discussions relating to the use of civic addresses in describing boundaries are included.

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 August 5, 2011.

Copyright Notice

Table of Contents

1. Introduction

TOC

Civic address information ([RFC4776] (Schulzrinne, H., “Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information,” November 2006.), [RFC5139] (Thomson, M. and J. Winterbottom, “Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO),” February 2008.)) can be used to describe the location of an entity in terms of the human-constructed environment. This description can be be made more or less precise through the addition or removal of labels (respectively). A less precise civic address can be used to describe a zone or region by only including sufficient labels to identify the region. This method is used in the Location-to-Service Translation (LoST) protocol (Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, “LoST: A Location-to-Service Translation Protocol,” August 2008.) [RFC5222], to convey information to a client about the extents of a particular region where service is guaranteed to be consistent.
This information, called a service boundary, allows a client to make decisions about civic addresses other than the one used to query. If another civic address is determined to be "within" the service boundary, the client does not need to request service information from the LoST server.

LoST does not provide a definition of "within" for civic addresses. This document describes an algorithm that provides a definition for whether a civic address is contained within another address. Other operations on civic addresses are described, allowing client software to make decisions about the intersection and union of two civic addresses.

---

2. Civic Address Model

TOC

The simplest model of a civic address is that which considers it as an unordered set of labels. Each label is assigned zero or more values; each value has an associated language (and script).
The format in RFC 4776 (Schulzrinne, H., “Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information,” November 2006.) [RFC4776] allows for values to be given to labels with different languages or scripts. No special considerations apply in applying this model.
The format in RFC 5139 (Thomson, M. and J. Winterbottom, “Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO),” February 2008.) [RFC5139] uses multiple `civicAddress` elements to form a single address if labels are provided in multiple languages. Thus, when extracting address information from a PIDF-LO (Peterson, J., “A Presence-based GEOPRIV Location Object Format,” December 2005.) [RFC4119] document, a civic address in this model is formed from all `civicAddress` elements in the same tuple.
A civic address describes a series of spatial partitions or regions. Every address includes an implied partition that identifies the habited portion of the Earth. Every label with a value describes a partition of space at a specific scale. The intersection of the spaces described by all the included labels is the resulting location.
The algorithm described in this document relies on the following rule:

> The location described by a set of civic address labels is entirely contained within the location described by any subset of those labels.

The following characteristics of civic addresses have no bearing on the algorithms described:

1. The civic address formats of [RFC4776] (Schulzrinne, H., “Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information,” November 2006.) and [RFC5139] (Thomson, M. and J. Winterbottom, “Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO),” February 2008.) include a limited set of hierarchical elements. The `country` and A1 through A6 labels

follow a strict hierarchy. The algorithms described do not rely on this hierarchy.

2. The physical region described by a civic address is not necessarily contiguous. For instance, an address might omit a thoroughfare name, but include a house number of 23. Such an address identifies every house at number 23 within the area described by other labels.

More sophisticated models and algorithms are possible in the presence of additional information about the address data. If this sort of information is present, many more options are available for processing addresses. The simple algorithms in this document operate on the address information only, but do not preclude use of outside information.

---

3. Civic Address Boundaries

TOC

A civic address boundary has the same format as a civic address.
A civic address boundary describes a region by containing fewer labels than the addresses of locations contained within the boundary. In that respect, the boundary might be considered an incomplete address, allthough a boundary is actually a valid civic address that simply describes a larger location.
A larger region is described by including fewer labels; a smaller region might be described by including more labels, or labels that are more specific.
For example, if the described region is the province of Zeeland in the Netherlands, only two labels are required: a country of NL and an A1 field of ZE.
A label that is omitted from the civic address boundary indicates that civic addresses within the boundary may have any value for the label.
This process does not provide any assurance that a civic address exists, only that if it does exist, it is contained entirely within the described boundary. Determining whether such an address actually exists usually requires additional information, and is therefore not considered by this document.

3.1. Determining if an Address is Within a Boundary

TOC

A civic address is entirely enclosed within a boundary if every label of the boundary that has a value has an equivalent value in the address. A civic address boundary can entirely enclose another civic address boundary.

Case folding is performed on values before comparison.
A label is considered equivalent if at least one value from the boundary has the same value in the address for the same language (and script). If values are provided in multiple languages, any language that is present in both boundary and address can be used.
[[TBD: If the label contains different values for the same language, does this override the above - in light of the Austrian example below, it's probably better that the more lenient equivalence test is used.]]
Labels that have the same value, but a different language, are not equivalent. Without information on different translations of the label, the label must be considered to be different.

3.2. Algorithm Summary

TOC

The algorithm for determining whether an address is contained entirely within a given boundary can be summarized by the following pseudocode:

```
SET iswithin = true
FOR EACH label IN boundary DO:
   IF boundary[label] exists THEN:
      SET equivalent = false
      FOR lang IN boundary[label] DO:
         IF boundary[label][lang] == address[label][lang] THEN:
            SET equivalent = true
         END
      END
      IF NOT equivalent THEN SET iswithin = false
   END
END
RETURN iswithin
```

3.3. False Negatives

TOC

This test can produce false negatives for a number of reasons:

1. A particular label might be specified with different languages in the boundary and the address. This label might be considered equivalent if the two values have the same meaning.

   For instance, the German city Muenchen is known as Munich in English - knowledge of this translation is required to determine that these two values are equivalent.

2. A label might have equivalent values, but subtly different language tags (Phillips, A. and M. Davis, “Tags for Identifying Languages,” September 2009.) [RFC5646] that result in a failed comparison.

   For instance, in many cases, a language tag of en is not significantly different from variants that use the same primary language subtag. Identical values with en and en-US or en-Latn would compare as different, even though the latter two tags are simply more specific than the first. Even en-GB is rarely different to these for text that is used in addressing.

   For creators of civic addresses and boundaries, the guidance in [RFC5646] (Phillips, A. and M. Davis, “Tags for Identifying Languages,” September 2009.), Section 4.1 recommends that subtags are only added if they include useful distinguishing information. This is intended to avoid processing errors such as this. This guidance is particularly relevant in relation to use of CAtype 128 (script) in the binary encoding of [RFC4776] (Schulzrinne, H., “Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information,” November 2006.), which is often unnecessarily specified.

3. The address might use different labels than the boundary to produce the same result.

   For instance, a boundary might use labels A1 through A6 to describe a location, whereas the same location is described using a postal code in place of these elements. The address is within the described boundary, but this cannot be determined without knowing that the postal code and A-labels are equivalent.

   In some countries, specific address codes can be used to replace some or all of the other address labels. In some instances, an address consisting of the country and the ADDCODE label can be sufficiently descriptive for an application; however, this would not be identified as being within a boundary that was specified using other address labels.

4. There are many cases where a value can be expressed in different ways. This includes abbreviations, commonly accepted misspellings, and generally recognized variations in addresses.

   For instance, abbreviations are common for thoroughfare suffixes, like Street (St. or St) or Road (Rd. or Rd).

   In another example, the Austrian addressing recommendations

(Wolf, K. and A. Mayrhofer, “Considerations for Civic Addresses in the Presence Information Data Format Location Object (PIDF-LO): Guidelines and IANA Registry Definition,” March 2010.) [RFC5774] let certain labels contain either a code or a descriptive name. Without knowing that Oberbaumgarten and Oberbaumgarten;1208 refer to the same Katastralgemeindenamen, these values must be considered to be different.

By using additional information, a system might be able to identify more equivalent labels than the basic algorithm. This can remove some, if not all such false negatives. However, a system should not rely on another system having and employing such knowledge.

3.4. False Positives

TOC

This algorithm guarantees, that a civic address that exists is entirely contained within a boundary.
No allowance is made for addresses that do no exist. It is trivially possible to construct a non-sensical or non-existent civic address that is considered "within" a boundary using this algorithm. This can be done by starting with the civic address boundary and adding arbitrary values to labels that do not already have values.

3.5. Address Boundary Limitations

TOC

The address format allows a limited expression for address boundaries. This representation can only be used in limited applications. This simple boundary expression is not suitable for any application that is sensitive to false negatives.
In the case of boundary interchange between LoST servers (Schulzrinne, H. and H. Tschofenig, “Synchronizing Location-to-Service Translation (LoST) Protocol based Service Boundaries and Mapping Elements,” March 2010.) [I-D.ietf-ecrit-lost-sync] would likely require multiple specific boundaries to describe a single boundary. A number of well-known cases would generate a very large number of such boundaries. For instance, if a boundary runs up the middle of street that places odd and even house numbers on opposite sides of the street, each house on that street would require an individual address.
Concatenation of address data can introduce other limitations. The limited set of address labels can mean that each field can hold several discrete units of data. An address mapping (Wolf, K. and A. Mayrhofer, “Considerations for Civic Addresses in the Presence Information Data Format Location Object (PIDF-LO): Guidelines and IANA Registry

Definition," March 2010.) [RFC5774] might specify that underlying data be concatenated and mapped to a single label.
The algorithm described here operates on entire labels only. The algorithm and boundary expression does not allow only part of a label to change. If concatenated data is included in the one label, a generic processer cannot know of this and distinguish between parts that must match and parts that do not matter. To do so would require knowledge of the modes of concatenation, what delimiters were used and it would require syntax that distinguishes important parts from unimportant parts.

---

4. Boundary Combining Algorithms

TOC

In some cases, it might be necessary to combine boundaries. This section describes three algorithms, including simple union and intersection.

4.1. Boundary Unions

TOC

The union of two civic address boundaries (or addresses) is a single boundary that contains all civic addresses that are contained within either original boundary.
The boundary that forms the union of two other boundaries is formed of all labels that are equivalent in both boundaries.
If labels differ in the two boundaries, then the resulting union might also include addresses that are in neither boundary. Only use this algorithm if false positives are acceptable.

```
SET union = empty address
FOR EACH label IN boundary1,boundary2 DO:
   IF boundary1[label] EQUIV boundary2[label] THEN
      SET union[label] = boundary1[label]
   END
END
RETURN union
```

The value of any label in a union of boundaries should include a value for all languages that are present in both boundaries.

TOC

4.2. Boundary Intersections

The intersection of two boundaries (or addresses) is a single boundary that contains all addresses that are found in both original boundaries. The boundary that forms the intersection of two other boundaries is formed of the combined value of the labels from both boundaries. If the value of any label is present in both boundaries, but not equivalent, the two boundaries do not intersect for the purposes of this algorithm. In practice, while it is possible that two such boundaries could overlap, this algorithm cannot detect this. Furthermore, the civic address representation does not provide a way to express such an overlap.

```
SET intersection = empty address
FOR EACH label IN boundary1,boundary2 DO:
   IF boundary1[label] exists AND boundary2[label] exists
       AND boundary1[label] NOT EQUIV boundary2[label] THEN
       ERROR No overlap
   END
   IF boundary1[label] exists THEN:
      SET intersection[label] = boundary1[label]
   ELSE
      SET intersection[label] = boundary2[label]
   END
END
RETURN intersection
```

The value of any label in an intersection of boundaries may include a value for all languages that are present in both boundaries.

4.3. Avoiding False Positives

TOC

The service boundaries in LoST (Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol," August 2008.) [RFC5222] rely on the absence of false positives when determining if an address is within a boundary. False negatives are tolerated, because this only results in the LoST client making another request to discover an unchanged service. A false positive is not desirable because a device could retain a service mapping that is likely to be invalid.
[I-D.ietf-ecrit-rough-loc] (Barnes, R. and M. Lepinski, "Using Imprecise Location for Emergency Context Resolution," August 2010.) describes an algorithm where an address is formed of the intersection of multiple boundaries. If no intersection the result of this algorithm was failure, then usable location information cannot be provided to the LoST client.

For a LoST service boundary, the goal is to provide a boundary that contains as few labels as possible. For a rough location, the goal is to provide an address with as few labels as possible. False positives are not desirable in either case.
Thus, both cases have a similar goal, and both have a more precise address to operate on. In LoST, this is the address used to query the server; for rough locations, this is the address that is to be hidden. To avoid false positives in determining whether an address falls within a boundary, labels from the more precise address are added. Any label where the inputs to the union or intersection disagree is given a value from the precise address. This ensures that the resulting address or boundary entirely contains the precise address.
Alternatively, given a precise address and a number of boundaries that are to be combined by either union or intersection, an address or boundary can be formed by removing all labels from the precise address that do not have a value in any boundary. This algorithm produces an identical result.

```
SET result = precise address
FOR EACH label IN result DO:
   SET found = false
   FOR EACH boundary IN boundaries DO:
      IF boundary[label] exists THEN SET found = true
   END
   IF not found THEN CLEAR result[label]
END
RETURN result
```

---

5. Example

TOC

The following civic address boundary (shown in XML form (Thomson, M. and J. Winterbottom, “Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO),” February 2008.) [RFC5139]), describes a region of Los Angeles, USA.

```
<civicAddress
    xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
  <country>US</country>
  <A1>CA</A1>
  <A2>Orange</A2>
  <A3>Los Angeles</A3>
  <A4>Anaheim</A4>
  <PC>92802</PC>
</civicAddress>
```

The following address includes additional labels, it does not change the value of any label.

```
<civicAddress
    xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
  <country>US</country>
  <A1>CA</A1>
  <A2> Orange </A2>
  <A3>Los Angeles</A3>
  <A4>Anaheim</A4>
  <PRD>South</PRD><RD>Harbor</RD><STS>Boulevard</STS>
  <HNO>1313</HNO><NAM>Disneyland</NAM>
  <PC>92802</PC>
</civicAddress>
```

An equivalent address that would not be considered "within" the boundary by the generic algorithm would be one that specified Orange County for the A2 label. It's also possible to unambiguously describe the location without fields A1 through A4, since a ZIP (postal) code of 92802 provides sufficient information.
Thus, the following variant is not considered "within" the boundary, even if it is the same address. That is, without context-specific knowledge, the algorithm produces a false negative on this address.

```
<civicAddress
    xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
  <country>US</country>
  <PRD>South</PRD><RD>Harbor</RD><STS>Boulevard</STS>
  <HNO>1313</HNO><NAM>Disneyland</NAM>
  <PC>92802</PC>
</civicAddress>
```

---

6. IANA Considerations

TOC

This document has no IANA actions.
[[RFC Editor: please remove this section prior to publication.]]

---

7. Security Considerations

TOC

This document describes a civic address model and algorithms for manipulating civic addresses and boundaries in the same format. There are no known security considerations arising from the described application of these algorithms.

8. Informative References

TOC

[RFC4119]	Peterson, J., "A Presence-based GEOPRIV Location Object Format," RFC 4119, December 2005 (TXT).
[RFC4776]	Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information," RFC 4776, November 2006 (TXT).
[RFC5139]	Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)," RFC 5139, February 2008 (TXT).
[RFC5222]	Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol," RFC 5222, August 2008 (TXT).
[RFC5646]	Phillips, A. and M. Davis, "Tags for Identifying Languages," BCP 47, RFC 5646, September 2009 (TXT).
[RFC5774]	Wolf, K. and A. Mayrhofer, "Considerations for Civic Addresses in the Presence Information Data Format Location Object (PIDF-LO): Guidelines and IANA Registry Definition," BCP 154, RFC 5774, March 2010 (TXT).
[I-D.ietf-ecrit-rough-loc]	Barnes, R. and M. Lepinski, "Using Imprecise Location for Emergency Context Resolution," draft-ietf-ecrit-rough-loc-03 (work in progress), August 2010 (TXT).
[I-D.ietf-ecrit-lost-sync]	Schulzrinne, H. and H. Tschofenig, "Synchronizing Location-to-Service Translation (LoST) Protocol based Service Boundaries and Mapping Elements," draft-ietf-ecrit-lost-sync-09 (work in progress), March 2010 (TXT).

Authors' Addresses

TOC

	Martin Thomson
	Andrew Corporation
	Andrew Building (39)
	Wollongong University Campus
	Northfields Avenue
	Wollongong, NSW 2522
	Australia
Phone:	+61 2 4221 2915
Email:	martin.thomson@andrew.com

	Karl Heinz Wolf
	nic.at GmbH
	Karlsplatz 1/2/9
	Wien A-1010
	Austria
Phone:	+43 1 5056416 37
Email:	karlheinz.wolf@nic.at
URI:	http://www.nic.at/