

ECRIT
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

R. Marshall
J. Martin
Comtech TCS
B. Rosen
Neustar
March 5, 2018

**A LoST extension to return complete and similar location info
draft-ietf-ecrit-similar-location-05**

Abstract

This document introduces a new way to provide returned location information in LoST responses that is either of a completed or similar form to the original input civic location, based on whether valid or invalid civic address elements are returned within the findServiceResponse message. This document defines a new extension to the findServiceResponse message within the LoST protocol [[RFC5222](#)] that enables the LoST protocol to return a completed civic address element set for a valid location response, and one or more suggested sets of similar location information for invalid LoST responses. These two types of civic addresses are referred to as either "complete location" or "similar location", and are included as a compilation of CAType xml elements within the existing LoST findServiceResponse message structure.

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 September 6, 2018.

Copyright Notice

Copyright (c) 2018 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	2
2.	Terminology	3
3.	Overview of Returned Location Information	4
4.	Returned Location Information	7
5.	Examples	8
5.1.	Complete Location returned for Valid Location response .	8
5.2.	Similar Location returned for Invalid Location response .	10
6.	XML Schema	12
7.	Security Considerations	13
8.	IANA Considerations	14
8.1.	XML Schema Registration	14
8.2.	LoST-RLI Namespace Registration	14
9.	References	15
9.1.	Normative References	15
9.2.	Informative References	15
	Authors' Addresses	16

[1.](#) Introduction

The LoST protocol [[RFC5222](#)] supports the validation of civic location information as input, by providing a set of validation result status indicators. The current usefulness of the supported xml elements, "valid", "invalid", and "unchecked", is limited, because while they each provide an indication of validity for any one location element as a part of the whole civic address, the mechanism is insufficient in providing either the complete set of civic address elements that the LoST server contains, or of providing alternate suggestions (hints) as to which civic address is intended for use.

Whether the input civic location is valid and missing information, or invalid due to missing or wrong information during input, this

document provides a mechanism to return a complete set of civic address elements for those valid or invalid cases.

This enhancement to the validation feature within LoST is required by systems that rely on accurate location for processing in order to increase the likelihood that the correct and/or complete form of a civic location becomes known in those cases where it is incomplete or incorrect. One such use case is that of location based emergency calling. The use of this protocol extension will facilitate the correction of errors, and allow location servers to be more easily provisioned with complete address information.

The structure of this document includes terminology, [Section 2](#), followed by a discussion of the basic elements involved in location validation. The use of these elements, by way of example, is discussed in an overview section, [Section 3](#), with accompanying rationale, and a brief discussion of the impacts to LoST, and its current schema.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

The following terms are defined in this document:

Location: The term Location can be used to refer to either a civic location or a geodetic location.

Geodetic Location: a geographic coordinate set of values that describes a point within a defined geographic datum. For example, a WGS84 referenced latitude, longitude coordinate pair (2D), or latitude, longitude, and altitude (3D). Note: geodetic location is defined here for context, but is not used elsewhere within this document.

Civic Location: The term civic location applies to a set of one or more civic address elements that are used in conjunction with each other, and in accordance with a known ruleset to designate a specific place within a region of geography, or a region of geography by itself as defined in [[RFC5139](#)].

Civic Address: The term Civic Address is used interchangeably with the term Civic Location within this document.

Civic Address Element: The term Civic Address Element is used within this document to apply to an individual CAtype data descriptor,

for example, as is described in [\[RFC4776\]](#), [\[RFC5774\]](#), and [\[RFC6848\]](#).

Invalid Location: A Civic Location that was included in a LoST request and subsequently returned with one or more civic address elements marked as invalid. Note that location information may be submitted in the findRequest that causes the LoST server to return locationInvalid. It is also possible that the location information submitted is so inaccurate that this extension can not be used, and the LoST server may return a notFound. In this document, we use the term Invalid Location only to refer to a case where the LoST server returns one or more elements in the invalid list.

Valid Location: A Civic Location that was included in a LoST request and subsequently returned with all civic address elements in the valid or unchecked lists.

Complete Location: An expanded civic location that includes other civic address elements in addition to the existing validated civic address elements provided as input to a LoST server.

Similar Location: A suggested civic location that is comparatively close to the civic location which was input, but which had one or more invalid civic address elements returned by the LoST server.

Returned Location Information: A set of civic locations returned in a LoST response.

3. Overview of Returned Location Information

This document describes an extension to LoST [\[RFC5222\]](#) to allow additional location information to be returned in a findServiceResponse where the location information in the request is in a civic profile as described in [RFC5222](#) or location in another profile derived from that civic profile, for two different use cases.

When a LoST server is asked to validate a civic location, its goal is to take the set of civic address elements provided as the location information in the LoST request, and find a unique location in its database that matches the information in the request. Uniqueness might not require values for all possible elements in the civic address that the database might hold. Further, the input location information might not represent the form of location the users of the LoST service prefer to have. As an example, there are LoST civic address elements that could be used to define a postal location, suitable for delivery of mail as well as a municipal location suitable for responding to an emergency call. While the LoST server

might be able to determine the location from the postal elements provided, the emergency services would prefer that the municipal location be used for any subsequent emergency call. Since validation is often performed well in advance of an end-user placing an emergency call, if the LoST server could return the preferred form of location (or more properly in this example, the municipal elements in addition to the postal elements), those elements could be stored in a LIS and used in a later emergency call.

In addition, this document describes the reuse of the same mechanism, but for a different purpose: to supply similar location information in the case where a LoST server response includes one or more civic address elements marked as invalid, constituting an invalid location response. In this case, the response contains one or more suggested alternative, but valid locations.

In a LoST `<findServiceResponse>` indicating a valid location - i.e. containing a `<locationValidation>` element with no elements listed as invalid - the `<locationValidation>` element may use this extension to include additional location information. As an example, the query might contain a HNO (house number), RD (road name) A3 (city), At (state/province) and a few more CAtype elements, but might not contain A2 (county) or PC (Postal Code) CAtypes. The HNO, RD, STS, POD, A3 and A1 civic address elements might be sufficient enough to the LoST server to uniquely locate the address specified in the request and thus be considered valid. Yet, downstream entities might find it helpful to have the additional A2 (county), and PC, (Postal Code), civic address elements that are present within the LoST server, be included as part of a complete location response. Since [\[RFC5222\]](#) currently does not have a way for this additional location information to be returned in the `findServiceResponse`, this document extends the LoST protocol so that it can include a `completeLocation` element within the `findServiceResponse` message, allowing for the representation of complete location information.

An example showing complete location information supplied:

input address: 6000 15th Ave NW Seattle

complete location: 6000 15th Ave NW Seattle, WA 98105 US

The information provided in the request may be enough to identify a unique location in the LoST server, but that may not be the location intended by the user. The `completeLocation` information may alert the user to a mismatch between the provided location information and the unique location the server interpreted that information to identify.

By contrast, when invalid location is received from the LoST server, with this extension, the same mechanism works as follows: if a LoST server returns a response to a findService request that contains a set of civic address elements with one or more labeled as invalid, the location information in the findServiceResponse is extended to include additional location information that it suspects might be the location desired.

In the example cited above, policy at the LoST server might deem a missing A3 element as invalid, even if the location information in the request was sufficient to identify a unique address. In that case, the missing element would be listed in the invalid list, and similarLocation could be returned in the response showing the missing elements including A3, the same as the above example.

As another example of the use of similarLocation, consider the results based on a similar data set as used above, where the HNO, RD, STS, A1, and A3 civic address elements are not sufficient to locate a unique address, which leads to an invalid location result. This is the case, despite the fact that the LoST server typically contains additional civic address elements which could have resulted in a uniquely identifiable location if additional data had been supplied with the query. Since [\[RFC5222\]](#) currently does not have a way for this additional location information to be returned in the findServiceResponse, this document extends [\[RFC5222\]](#) so that the LoST findServiceResponse message can include one or more similarLocation elements within the findServiceResponse message representing similar civic locations.

To show this, suppose that a slightly modified address as above is inserted within a Lost findService request:

input address: 6000 15th Ave N Seattle, WA.

This time we make the assumption that the address is deemed "invalid" by the LoST server because there is no such thing as "15th Ave N" within the LoST server's data for the city of Seattle. However, we also happen to know for this example that there are two addresses within the address dataset that are "similar", when all parts of the address are taken as a whole. These similar addresses that could be suggested to the user are as follows:

similar address #1: 6000 15th Ave NW Seattle, WA 98107

similar address #2: 6000 15th Ave NE Seattle, WA 98105

This extension would allow the LoST server to include the above similar addresses as civicAddress elements in the response to

locationValidation. The next section shows examples of the LoST request and response xml message fragments for the above valid and invalid scenarios, returning the complete or similar addresses, respectively.

4. Returned Location Information

The LoST server implementing this extension MAY include completeLocation or similarLocation in the findService response. completeLocation and similarLocation contain a list of civic address elements identical to the elements used in the location element with the "civic profile" in [[RFC5222](#)] or another profile derived from the civic profile.

The LoST server MAY include more than one similarLocation elements in the response, but SHOULD NOT return more than a few possible similar locations. If there are too many possible locations, the server MAY return none, or it MAY return the few it considers most likely. The definition of "few" is left to the implementation of the LoST server. The server is unable to know what the intended location information was suppose to be; it is guessing. Therefore the correct location may or may not be one of the similarLocation elements the server provides, and the client cannot assume that any of them are the correct one.

Where a LoST server contains additional location information relating to that civic address, the findServiceResponse message MAY return additional location information along with the original validated civic address elements in order to form a complete location based on local implementation policy in the completeLocation element. completeLocation MUST NOT be returned in response messages where any civic address elements occur in the invalid list of the response, or where the set of civic address elements in the request do not identify a unique location. The complete location MUST NOT contain any elements that would be marked as invalid, or cause an error, if a recipient of that location performs a subsequent findService request using the complete location. However, if a subsequent request includes the complete location, the corresponding request MAY include elements in the unchecked list.

Clients can control the return of additional location information by including an optional <returnAdditionalLocation> element with possible values "none", "similar", "complete" or "any". Where "none" means to not return additional location information, "similar" and "complete" mean to only return the respective type of additional location information (if the server could send any) and "any" means to include similar and/or complete location (if the server could send any). If the request includes this option, the server MUST NOT send

location information contravening the client's request. Not including this option in the request is equivalent to "any".

The server may determine that there are many possible similar locations and decide not to send them all. The number of similar locations sent is entirely up to the server, but MUST be less than 10. The server MAY include a <similarLocationsLimited> element which contains a non-zero integer of the number of similar locations not included in the response. The server is NOT obligated to make this number accurate, in that there may be more than the indicated similar locations available in the data held by the server.

Clients MAY ignore the location information this extension defines in the response. The information is optional to send, and optional to use. In the case where the location information in the request was valid, this extension does not change the validity. In the case where the location information in the request is invalid, but alternate location information is returned, the original location remains invalid, and the LoST server does not change the mapping response other than optionally including the information defined by this extension.

completeLocation and similarLocation use the locationInformation element from [\[RFC5222\]](#) including the profile attribute which is useful if the request contains location information in a profile derived from the civic profile.

5. Examples

5.1. Complete Location returned for Valid Location response

Based on the example input request, returned location information is provided in a findServiceResponse message when the original input address is considered valid, but is missing some additional data that the LoST server has.

```
<!-- =====Request===== -->

<findService xmlns="urn:ietf:params:xml:ns:lost1"
  validateLocation="true">

  <location id="587cd3880" profile="civic">
    <civicAddress
      xmlns="urn:ietf:params:mxl:ns:pidf:geopriv10:civicAddr">

      <ca:country>US</ca:country>
```



```
<A1>WA</A1>
<A3>Seattle</A3>
<RD>15th</RD>
<STS>Ave</STS>
<POD>NW</POD>
<HNO>60000</HNO>

</civicAddress>
</location>

<service>urn:service:sos</service>

</findService>

<!-- =====Response===== -->

<findServiceResponse >
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:rli="urn:ietf:params:xml:ns:lost-rli1">
  xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

  <mapping
    expires="NO-CACHE"
    lastUpdated="2006-11-01T01:00:00Z"
    source="authoritative.example"
    sourceId="8799e346000098aa3e">

    <displayName xml:lang="en">Seattle 911</displayName>
    <service>urn:service:sos</service>
    <uri>sip:seattle-911@example.com</uri>
    <serviceNumber>911</serviceNumber>

  </mapping>

  <locationValidation

    <valid>ca:country ca:A1 ca:A3 ca:RD ca:STS ca:POD ca:HNO<
      /valid>

    <invalid></invalid>
    <unchecked></unchecked>

  <rli:completeLocation> <!-- completed address -->
    <ca:civicAddress>
      <ca:country>US</ca:country>
      <ca:A1>WA</ca:A1>
      <ca:A2>KING COUNTY</ca:A2>
```



```

        <ca:A3>SEATTLE</ca:A3>
        <ca:RD>15TH</ca:RD>
        <ca:STS>AVENUE</ca:STS>
        <ca:POD>NORTHWEST</ca:POD>
        <ca:HNO>60000</ca:HNO>
        <ca:PC>98106</ca:PC>
        <ca:PCN>SEATTLE</ca:PCN>
      </ca:civicAddress>

    </rli:completeLocation>

  </locationValidation>

  <path>
    <via source="authoritative.example"/>
  </path>

  <locationUsed id="587cd3880"/>

</findServiceResponse>

<!-- ===== -->

```

5.2. Similar Location returned for Invalid Location response

The following example shows returned location information provided in a findServiceResponse message when the original input address is considered invalid, because of the unmatchable POD data (in this example) that the LoST server needs to provide a unique mapping.

```

<!-- =====Request===== -->

<findService xmlns="urn:ietf:params:xml:ns:lost1"
  validateLocation="true">

  <location id="587cd3880" profile="civic">
    <civicAddress
      xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

      <country>US</country>
      <A1>WA</A1>
      <A3>Seattle</A3>
    </civicAddress>
  </location>
</findService>

```



```
<RD>15th</RD>
<STS>Ave</STS>
<POD>N</POD>
<HNO>6000</HNO>

</civicAddress>
</location>

<service>urn:service:sos</service>

</findService>

<!-- =====Response===== -->

<findServiceResponse>
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:rli="urn:ietf:params:xml:ns:lost-rli1">
    xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

      <mapping
        expires="NO-CACHE"
        lastUpdated="2006-11-01T01:00:00Z"
        source="authoritative.example"
        sourceId="8799e346000098aa3e">

        <displayName xml:lang="en">Seattle 911</displayName>
        <service>urn:service:sos</service>
        <uri>sip:seattle-911@example.com</uri>
        <serviceNumber>911</serviceNumber>

      </mapping>

      <locationValidation

        <valid>ca:country ca:A1 ca:A3 ca:STS ca:RD</valid>
        <invalid>ca:POD</invalid>
        <unchecked>ca:HNO</unchecked>

        <rli:similarLocation> <!-- similar location info -->
          <ca:civicAddress> <!-- similar address #1 -->
            <ca:country>US</ca:country>
            <ca:A1>WA</ca:A1>
            <ca:A2>KING COUNTY</ca:A2>
            <ca:A3>SEATTLE</ca:A3>
            <ca:RD>15TH</ca:RD>
            <ca:STS>AVENUE</ca:STS>
            <ca:POD>NORTHWEST</ca:POD>
```



```

        <ca:HNO>6000</ca:HNO>
        <ca:PC>98106</ca:PC>
        <ca:PCN>SEATTLE</ca:PCN>
      </ca:civicAddress>
    </rli:similarLocation>
    <rli:similarLocation>
      <ca:civicAddress>  <!-- similar address #2 -->
        <ca:country>US</ca:country>
        <ca:A1>WA</ca:A1>
        <ca:A2>KING COUNTY</ca:A2>
        <ca:A3>SEATTLE</ca:A3>
        <ca:RD>15TH</ca:RD>
        <ca:STS>AVENUE</ca:STS>
        <ca:POD>NORTHEAST</ca:POD>
        <ca:HNO>6000</ca:HNO>
        <ca:PC>98105</ca:PC>
        <ca:PCN>SEATTLE</ca:PCN>
      </ca:civicAddress>
    </rli:similarLocation>

  </locationValidation>

  <path>
    <via source="authoritative.example"/>
  </path>

  <locationUsed id="587cd3880"/>

</findServiceResponse>

<!-- ===== -->

```

6. XML Schema

This section provides the schema of LoST extensions based on the schema in [draft-ietf-lost-planned-changes](#) />


```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:lost-rli1"
  targetNamespace="urn:ietf:params:xml:ns:lost-rli1"
  elementFormDefault="qualified">
  <!-- Import base Lost -->
  <xs:import namespace="urn:ietf:params:xml:ns:lost1"/>
  <!-- extend the extensionPoint of commonRequestPattern of findService to
  include: -->
  <xs:attribute name="asOf" type="xs:dateTime" minOccurs="0"/>
  <xs:attribute name="returnAdditionalLocation" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:token">
        <xs:enumeration value="none"/>
        <xs:enumeration value="similar"/>
        <xs:enumeration value="complete"/>
        <xs:enumeration value="any"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>

  <!-- extend the extensionPoint of commonResponsePattern of findServiceReponse
  to include -->
  <xs:group ref="similarLocation"/>
  <xs:group ref="completeLocation"/>
  <!-- where -->
  <xs:group name="similarLocation">
    <xs:sequence>
      <xs:element ref="location" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:group name="completeLocation">
    <xs:sequence>
      <xs:element ref="location" maxOccurs="1"/>
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

7. Security Considerations

Whether the input to the LoST server is valid or invalid, the LoST

server ultimately determines what it considers to be valid. Even in the case where the input location is valid, the requester still might

not actually understand where that location is. For this kind of valid location use case, this described extension would typically return more location information than the requester started with, which might reveal more about the location. While this might be very desirable in some scenarios including, for example, supporting an emergency call, it might not be as desirable for other services. Individual LoST server implementations SHOULD consider the risk of releasing more detail versus the value in doing so. Generally, it is not expected that this would be a significant problem as the requester must have enough location information to be considered valid, which in most cases is enough to uniquely locate the address. Providing more CATypes generally doesn't actually reveal anything more. For invalid locations that are submitted, this extension would allow the LoST response to include location information which is similar to what was input, again resulting in more information provided in the response than was known during input. LoST server implementations SHOULD evaluate the particular use cases where this extension is supported, and weigh the risks around its use. Many similar database services available today via the Internet offer similar features, such as "did you mean", and address completion, so this capability is not introducing any fundamentally new threat.

8. IANA Considerations

8.1. XML Schema Registration

URI: urn:ietf:params:xml:schema:lost-rli1

Registrant Contact: IETF ECRIT Working Group, Brian Rosen
(br@brianrosen.net).

XML Schema: The XML schema to be registered is contained in [Section 7](#).

8.2. LoST-RLI Namespace Registration

URI: urn:ietf:params:xml:ns:lost-rli1

Registrant Contact: IETF ECRIT Working Group, Brian Rosen
(br@brianrosen.net).

XML:

BEGIN

```
<?xml version="2.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml1-basic/xhtml1-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1"/>
  <title>LoST Returned Location Information Namespace</title>
</head>
<body>
  <h1>Namespace for LoST Returned Location Information extension</h1>
  <h2>urn:ietf:params:xml:ns:lost-rli1</h2>
  <p>See <a href="http://www.rfc-editor.org/rfc/rfc?????.txt">
    RFC????</a>.</p>
</body>
</html>
END
```

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", [RFC 5222](#), DOI 10.17487/RFC5222, August 2008, <<https://www.rfc-editor.org/info/rfc5222>>.

9.2. Informative References

- [RFC4776] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information", [RFC 4776](#), DOI 10.17487/RFC4776, November 2006, <<https://www.rfc-editor.org/info/rfc4776>>.

- [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>>.
- [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](#), DOI 10.17487/RFC5774, March 2010, <<https://www.rfc-editor.org/info/rfc5774>>.
- [RFC6848] Winterbottom, J., Thomson, M., Barnes, R., Rosen, B., and R. George, "Specifying Civic Address Extensions in the Presence Information Data Format Location Object (PIDF-LO)", [RFC 6848](#), DOI 10.17487/RFC6848, January 2013, <<https://www.rfc-editor.org/info/rfc6848>>.

Authors' Addresses

Roger Marshall
Comtech TCS
2401 Elliott Avenue
2nd Floor
Seattle, WA 98121
US

Email: rmarshall@comtechtel.com

Jeff Martin
Comtech TCS
2401 Elliott Avenue
2nd Floor
Seattle, WA 98121
US

Email: jmartin@comtechtel.com

Brian Rosen
Neustar
470 Conrad Dr
Mars, PA 16046
US

Email: br@brianrosen.net

