

ECRIT
Internet-Draft
Updates: [5222](#) (if approved)
Intended status: Standards Track
Expires: 5 September 2022

B. Rosen

R. Marshall
J. Martin
Comtech TCS
4 March 2022

A LoST extension to return complete and similar location info
draft-ietf-ecrit-similar-location-19

Abstract

This document describes an extension to the LoST protocol of [RFC 5222](#) that allows additional civic location information to be returned in a `<findServiceResponse>`. This extension supports two use cases: First, when the input location is valid but lacks some Civic Address elements, the LoST server can provide a completed form. Second, when the input location is invalid, the LoST server can identify one or more feasible ("similar") locations. This extension is applicable when the location information in the `<findService>` request uses the Basic Civic profile as described in [RFC 5222](#) or another profile whose definition provides instructions concerning its use with this extension.

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 5 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

Returned Location Extensions to LoST

March 2022

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 Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	Overview of Returned Location Information	4
4.	Returned Location Information	8
5.	Examples	10
5.1.	Complete Location returned for Valid Location	10
5.2.	Similar Location returned for Invalid Location	12
6.	XML Schema	14
7.	Security Considerations	16
8.	IANA Considerations	16
8.1.	XML Schema Registration	16
8.2.	LoST-RLI Namespace Registration	17
9.	References	17
9.1.	Normative References	17
9.2.	Informative References	18
	Authors' Addresses	18

[1.](#) Introduction

The LoST protocol [[RFC5222](#)] supports the validation of civic location information sent in a <findService> request, by providing a set of validation result status indicators in the response. The current usefulness of the supported XML elements <valid>, <invalid>, and <unchecked> is limited. They each provide an indication of validity for any one location element as a part of the whole civic address, but this 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 queried civic location is valid but missing information, or invalid due to missing or wrong information, this document

provides a mechanism to return a complete set of civic address elements.

This enhancement to the validation feature within LoST is required by systems that rely on accurate location for processing. Use of this enhancement increases the likelihood that incorrect or incomplete civic location will be updated. One such use case is that of location-based emergency calling. The use of this protocol extension facilitates the timely correction of errors, and allows LoST clients to be more easily provisioned with complete address information.

[2.](#) Terminology

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

The following terms are defined in this document:

Location: The term Location is in general used to refer to either a civic location or a geodetic location. In the context of this document, location is restricted to civic locations.

Geodetic Location: a geographic coordinate set of values that describes a point within a defined geographic datum. For example, a 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: A set of Civic Address Elements as defined in [[RFC5139](#)] that are used in conjunction with each other, and in accordance with a known ruleset to designate a specific place within a country.

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 indicate any individual XML element used within the <civicAddress> type defined in [[RFC5139](#)], including elements used at the extension point therein such as those defined in [[RFC6848](#)] and elsewhere. This term also includes the reference to such elements by qualified name as defined within the <locationValidation> element in [[RFC5222](#)].

Invalid Location: A Civic Location that, when included in a LoST query with 'validateLocation' set, will receive a response having one or more Civic Address Elements in the <invalid> list. Note that it is also possible that the location information submitted

is so inaccurate that location validation cannot be performed, and the LoST server may return a <notFound> or <locationInvalid> error. In this document, the term Invalid Location only refers to a case where the LoST server returns one or more elements in the <invalid> list; the error conditions are not considered.

Valid Location: A Civic Location that, when included in a LoST query with 'validateLocation' set, will receive a response having all Civic Address Elements in the <valid> or <unchecked> lists.

Incomplete Location: "A Civic Location that the LoST server considers valid but which omits one or more Civic Address Elements that the LoST server considers necessary or helpful.

Complete Location: A Civic Location that was considered by the LoST server to be an Incomplete Location but that with the addition of one or more Civic Address Elements is considered complete by the LoST server.

Similar Location: A suggested civic location that is similar to an Invalid Location which was used in a LoST query, but which has one or more elements added, modified, or removed such that the suggested location is a Valid Location. Similar Location may be returned when the input location is invalid.

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](#)] that allows additional location information to be returned in the <locationValidation> element of a <findServiceResponse>. This extension has two different use cases: First, when the input location is incomplete but the LoST server can identify the intended unique address, and second, when the input location is invalid and the LoST server can identify one or more likely intended locations. This extension is applicable when the location information in the <findService> request is in the Basic Civic profile as described in [[RFC5222](#)] or in another profile whose definition provides instructions concerning its use with this extension. As of this document's publication, no such additional location profiles have been defined, so this document describes the returned location extension using the Basic Civic profile. In addition, the following restriction is imposed: A server MUST NOT include Returned Location Information using a location profile that differs from the profile of the location used to answer the query and, by extension, MUST NOT include Returned Location Information using a location profile that

was not used by the client in the request.

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 client application such as a Location Information Server (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, indicating an Invalid Location used in the query. In this case, the response contains one or more suggested alternative Valid Locations.

In a LoST `<findServiceResponse>` indicating a Valid Location ([Section 2](#)) i.e., containing the `<locationValidation>` element with no elements listed as invalid, the LoST server can use this extension to include additional location information in a `<locationValidation>` element. As an example, a query contains `<HNO>` (house number), `<RD>` (road name) `<A3>` (city), `<A1>` (state/province) but not `<POD>` (Post-Directional). In this example, there is no street with just the street name, all streets have a post-directional. However, the LoST server is able to uniquely locate the intended address and thus consider the queried location Valid, as there is only one street with the street name and house number in the city that it could be. As the queried location is missing the post-directional element, a more strict entity might consider it Invalid (e.g., during a subsequent query) or worse, the lack of a Post-Directional element might cause confusion and delay an emergency response. The server can use this extension to supply the missing post-directional element `<POD>` in a `<completeLocation>` element within the `<locationValidation>` element.

In another example, the civic location in a request might lack `<A2>` (county) and `<PC>` (Postal Code) Civic Address Elements. In this example, too, the LoST server is able to uniquely locate the intended address and consider the location `<Valid>`, but other entities involved in a subsequent emergency call might find it helpful to have the additional Civic Address Elements. The LoST server can use this extension to supply the missing `<A2>` and `<PC>` Civic Address Elements.

Since [\[RFC5222\]](#) 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 `<locationValidation>` element of the `<findServiceResponse>` message, allowing for the representation of complete location information.

An example showing complete location information supplied:

Input address:

6000 15th Avenue
Leets, WA US

Complete Location:

6000 15th Avenue Northwest
Leets, 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 end 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.

The other use case for this extension is when the <findService> request contains an Invalid Location. When a LoST server returns a response to a <findService> request that contains a set of Civic Address Elements with one or more listed as invalid, this extension allows the server to include in the <locationValidation> element one or more locations that might be the intended location.

In the example cited above, policy at the LoST server might deem a missing <A3> element as being invalid, even if the location information in the request is sufficient to identify a unique address. In this case, the missing element is listed in the <invalid> list, and a <similarLocation> element could be returned in the response showing a complete civic location that includes the missing <A3> element, just as in the above example the missing <A3> element is included in a <completeLocation> element.

As another example of the use of <similarLocation>, consider the results based on a similar data set as used above, but where the <HNO>, <RD>, <STS>, <A1>, and <A3> Civic Address Elements are not sufficient to locate a unique address, resulting in an invalid location response. Because the LoST server contains additional civic address elements which, had they been included in the query, would have resulted in a uniquely identifiable location, the server can include one or more <similarLocation> elements containing the supplied Civic Address Elements plus the omitted ones. Since [\[RFC5222\]](#) does not have a way for this additional location information to be returned in the <findServiceResponse>, this document extends [\[RFC5222\]](#) so that the LoST <locationValidation> element of the <findServiceResponse> can include one or more <similarLocation> elements representing similar civic locations.

To show this, suppose that a slightly modified version of the above address is sent within a Lost <findService> request:

Input address:

6000 15th Avenue North
Leets, WA US

This time we make the assumption that the address is deemed invalid by the LoST server because there is no such thing as "15th Avenue North" within the LoST server's data for the city of Leets. 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 returned to the client are as follows:

Similar address #1:

6000 15th Avenue Northwest
Leets, WA 98107 US

Similar address #2:

6000 15th Avenue Northeast

This extension allows the LoST server to include the above similar addresses in the response to a <findService> request with 'validateLocaton' set to true. [Section 5](#) 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

A LoST server implementing this extension MAY include <completeLocation> or <similarLocation> elements within the <locationValidation> portion of the <findServiceResponse>. The <completeLocation> and <similarLocation> elements are of type "locationInformation" as defined by the LoST schema in [\[RFC5222\]](#). These elements MAY contain location information either in the Basic Civic profile defined in [\[RFC5222\]](#) or in another profile whose definition provides instructions concerning its use with this extension; this MUST be the same profile as the location in the query. When used with the Basic Civic profile, these elements contain a <civicAddress> element as defined in [\[RFC5139\]](#).

If there is at least one Civic Address Element in the <invalid> list, the LoST server MAY include one or more <similarLocation> element in the response. If there are too many possible locations, the server MAY return none, or it MAY return a subset considered most likely. How many to return is left to the implementation of the LoST server. The server is unable to know what the intended location information was supposed 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 one of them is the correct location.

Where a LoST server contains additional location information relating to the Civic Address used in a <findService> request, the <findServiceResponse> message MAY include a <completeLocation> element containing additional location information along with the original validated Civic Address Elements; the additional Civic Address Elements may be deemed by local policy as necessary to form a Complete Location. The <completeLocation> element 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 response MAY include elements in the unchecked list.

Clients can control the return of additional location information by including the optional 'returnAdditionalLocation' attribute with possible values 'none', 'similar', 'complete' or 'any'. The value '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 attribute, the server MUST NOT send location information contravening the client's request. Omitting this attribute in the request is equivalent to including it with the value 'none'.

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. The server MAY include a 'similarLocationsOmitted' attribute which contains a non-zero integer indicating the minimum number of Similar Locations not included in the response. There may be more than the indicated similar locations available in the data held by the server, but no mechanism to request more Similar Locations is provided.

Clients MAY ignore the location information this extension defines. 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.

The <completeLocation> and <similarLocation> elements use the <locationInformation> element from [[RFC5222](#)] updated by [[I-D.ietf-ecrit-lost-planned-changes](#)], including the 'profile' attribute, which is useful if the request contains location information in a profile other than the 'civic' profile. The 'profile' attribute MUST be included in both the request and the response and MUST be the same profile in both.

[5.](#) Examples

[5.1.](#) Complete Location returned for Valid Location

This example is based on the example request above; the LoST server considered the location in the query to be valid but missing some Civic Address elements, so in the Returned Location Information in the <findServiceResponse>, the server includes a <completeLocation> element supplying the omitted Civic Address elements <A2>, <PC>, and <PCN>.

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

<findService xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:rli="urn:ietf:params:xml:ns:lost-rli1"
  validateLocation="true" rli:returnAdditionalLocation="any">

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

      <country>US</country>
      <A1>WA</A1>
      <A3>Leets</A3>
      <RD>15th</RD>
      <STS>Avenue</STS>
      <POD>Northwest</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">Leets 911</displayName>
    <service>urn:service:sos</service>
    <uri>sip:leets-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 profile="civic"><!--completed address-->
      <ca:civicAddress>
        <ca:country>US</ca:country>
        <ca:A1>WA</ca:A1>
        <ca:A2>SHOWAK COUNTY</ca:A2>
        <ca:A3>LEETS</ca:A3>
        <ca:RD>15TH</ca:RD>
      </ca:civicAddress>
    </rli:completeLocation>
  </locationValidation>
</findServiceResponse>

```

```

        <ca:STS>AVENUE</ca:STS>
        <ca:POD>NORTHWEST</ca:POD>
        <ca:HNO>6000</ca:HNO>
        <ca:PC>98106</ca:PC>
        <ca:PCN>LEETS</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

The following example shows two Similar Locations returned in a <findServiceResponse> message when the original input address is considered invalid, in this example because the LoST server needed the omitted POD data to match a unique address.

```

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

```

```

<findService xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:rli="urn:ietf:params:xml:ns:lost-rli1"
  validateLocation="true"
  rli:returnAdditionalLocation="any">

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

      <country>AU</country>

```

```

        <A1>QLD</A1>
        <A3>PIMPANA</A3>
        <RD>SHIRLEY</RD>
        <STS>STREET</STS>
        <HNO>98</HNO>
        <PC>4209</PC>
        <PCN>PIMPANA</PCN>

    </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">Australian 000</displayName>
  <service>urn:service:sos</service>
  <uri>sip:000@example.com</uri>
  <serviceNumber>000</serviceNumber>

</mapping>

<locationValidation similarLocationsOmitted="5">

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

```

```

    <rli:similarLocation profile="civic"><!--similar location-->
      <ca:civicAddress> <!-- similar address #1 -->
        <ca:country>AU</ca:country>
        <ca:A1>QLD</ca:A1>
        <ca:A3>PIMPANA</ca:A3>
        <ca:RD>SHIRLEY</ca:RD>
        <ca:STS>STREET</ca:STS>
        <ca:POD>NORTH</ca:POD>
        <ca:HNO>98</ca:HNO>
        <ca:PC>4209</ca:PC>
        <ca:PCN>PIMPANAS</ca:PCN>
      </ca:civicAddress>
    </rli:similarLocation>
    <rli:similarLocation profile="civic" >
      <ca:civicAddress> <!-- similar address #2 -->
        <ca:country>AU</ca:country>
        <ca:A1>QLD</ca:A1>
        <ca:A3>PIMPANA</ca:A3>
        <ca:RD>SHIRLEY</ca:RD>
        <ca:STS>STREET</ca:STS>
        <ca:POD>SOUTH</ca:POD>
        <ca:HNO>98</ca:HNO>
        <ca:PC>4209</ca:PC>
        <ca:PCN>PIMPANAS</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 the LoST extensions, based on the schema in [[I-D.ietf-ecrit-lost-planned-changes](#)]

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:lost1="urn:ietf:params:xml:ns:lost1"
            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 findService by placing the following
at the extensionPoint
in the included commonRequestPattern: -->
    <xs:attribute name="returnAdditionalLocation" use="optional">
        <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 locationValidation by placing the following
        at the extensionPoint -->
    <xs:group>
        <xs:choice minOccurs="0">
            <xs:element name="similarLocation"
                type="lost1:locationInformation"
                minOccurs="1" maxOccurs="unbounded" />
            <xs:element name="completeLocation"
                type="lost1:locationInformation"/>
        </xs:choice>
    </xs:group>

    <!-- and also at the locationValidation extensionPoint -->
    <xs:attribute name="similarLocationsOmitted" use="optional">
        <xs:simpleType>
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="1"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>

</xs:schema>

```

7. Security Considerations

As this document defines extensions to [\[RFC5222\]](#), the Security Considerations of that document apply here.

Whether the input to the LoST server is a Valid or Invalid Location, the LoST server ultimately determines what it considers to be a Valid Location. 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 extension will typically return more location information than what the requester started with, which might reveal to the requester additional information (additional CATypes) about the location. While this is very desirable in some scenarios such as 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, supplying more information (CATypes) is not considered to be a significant problem because the requester has to already have enough information for the location 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. When Invalid Locations are submitted, this extension allows the LoST response to include locations that are similar to what was input, again resulting in more information provided in the response than was sent in the request. LoST server implementations should evaluate the particular use cases where this extension is supported, and weigh the risks around its use. Many services available today via the Internet offer similar features, such as "did you mean" or address completion, so this capability is not introducing any fundamentally new security concern.

The similar location service could be misused to attempt to enumerate the entire database by running a high volume of invalid or partial queries. The LoST server can limit the volume of similar locations it returns. It can also authenticate queries and limit the service to known queriers

8. IANA Considerations

8.1. XML Schema Registration

IANA is requested to register the following in the "schema" sub-registry of the IETF XML Registry per [\[RFC3688\]](#).

Internet-Draft

Returned Location Extensions to LoST

March 2022

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 6](#).

[8.2](#). LoST-RLI Namespace Registration

IANA is requested to register the following in the "ns" sub-registry of the IETF XML registry per [[RFC3553](#)].

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

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

XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-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

Rosen, et al.

Expires 5 September 2022

[Page 17]

Internet-Draft

Returned Location Extensions to LoST

March 2022

[I-D.ietf-ecrit-lost-planned-changes]

Rosen, B., "Validation of Locations Around a Planned Change", Work in Progress, Internet-Draft, [draft-ietf-ecrit-lost-planned-changes-05](#), 11 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-ecrit-lost-planned-changes-05.txt>>.

[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>>.

[RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", [BCP 73](#), [RFC 3553](#), DOI 10.17487/RFC3553, June 2003, <<https://www.rfc-editor.org/info/rfc3553>>.

[RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

[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>>.

[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>>.

[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>>.

9.2. Informative References

- [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

Rosen, et al. Expires 5 September 2022 [Page 18]

Internet-Draft Returned Location Extensions to LoST March 2022

Brian Rosen
470 Conrad Dr
Mars, PA 16046
United States of America
Email: br@brianrosen.net

Roger Marshall
Comtech TCS
2401 Elliott Avenue
2nd Floor
Seattle, WA 98121
United States of America
Email: roger.marshall@comtechtel.com

Jeff Martin
Comtech TCS
2401 Elliott Avenue
2nd Floor
Seattle, WA 98121
United States of America
Email: jeff.martin@comtechtel.com

