Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 19, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

The Presence Information Data Format Location Object (PIDF-LO) specification provides a flexible and versatile means to represent location information. There are, however, circumstances that arise when information needs to be constrained in how it is represented so that the number of options that need to be implemented in order to make use of it are reduced. There is growing interest in being able
to use location information contained in a PIDF-LO for routing applications. To allow successfully interoperability between applications, location information needs to be normative and more tightly constrained than is currently specified in the PIDF-LO. This document makes recommendations on how to constrain, represent and interpret locations in a PIDF-LO. It further looks at existing communications standards that make use of geodetic information for routing purposes and recommends a subset of GML that MUST be implemented by applications to allow location dependent routing to occur.
Table of Contents

1. CHANGES SINCE LAST TIME .................................... 4
  1.1 01 changes ........................................... 4
2. Introduction .............................................. 5
3. Terminology .............................................. 6
4. Using Location Information ................................. 7
   4.1 Single Civic Location Information .................... 8
   4.2 Civic and Geospatial Location Information ............ 9
   4.3 Manual/Automatic Configuration of Location Information 11
5. Geodetic Coordinate Representation ...................... 12
6. Uncertainty in Location Representation .................. 14
   6.1 Arc band ........................................... 14
   6.2 Ellipsoid Point With Uncertainty Circle ............... 18
   6.3 Polygon ............................................ 21
7. Baseline Geometry ......................................... 23
   7.1 Zero Dimensions .................................... 23
   7.2 One Dimensions .................................... 23
   7.3 Two Dimensions .................................... 24
   7.4 Three Dimensions .................................. 24
   7.5 Envelopes ......................................... 24
   7.6 Temporal Dimensions ................................ 25
   7.7 Units of Measure .................................... 25
   7.8 Coordinate Reference System (CRS) ..................... 25
8. Recommendations ............................................ 27
9. Security Considerations .................................. 28
10. IANA Considerations ...................................... 29
11. Acknowledgments ......................................... 30
12. Open Issues ............................................. 31
13. References ............................................... 32
   13.1 Normative references ............................... 32
   13.2 Informative References ............................. 32
   Authors’ Addresses ..................................... 33
A. Creating a PIDF-LO from DHCP Geo Encoded Data ............ 34
   A.1 Latitude and Longitude ............................... 34
   A.2 Altitude ........................................... 36
   A.3 Generating the PIDF-LO ............................... 36
   Intellectual Property and Copyright Statements ............ 41
1. CHANGES SINCE LAST TIME

1.1 01 changes

   minor changes to the abstract.

   Minor changes to the introduction.

   Added and appendix to take implementers through how to create a
   PIDF-LO from data received using DHCP option 123 as defined in [3].

   Rectified examples to use position and pos rather than location and
   point.

   Corrected example 3 so that it does not violate SIP rules.

   Added addition geopriv elements to the status component of the figure
   in "Using Location Information" to more accurately reflect the
   cardinality issues.

   Revised text in section "Geodetion Coordinate Representation."
   Removed last example as this was addressed with the change to
   position and pos in previous examples.
2. Introduction

The Presence Information Data Format Location Object (PIDF-LO) [1] is the IETF recommended way of encoding location information and associated privacy policies. Location information in PIDF-LO may be described in a geospatial manner based on a subset of GMLv3, or as civic location information. Uses for PIDF-LO are envisioned in the context of numerous location based applications. This document makes recommendations for formats and conventions to make interoperability less problematic. To enhance clarify formats comparisons between GML and the 3GPP Mobile Location Protocol (MLP) standard [4], are examined.
3. 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 [2].
4. Using Location Information

The PIDF format provides for an unbounded number of tuples. The geopriv element resides inside the status component of a tuple, hence a single PIDF document may contain an arbitrary number of location objects some or all of which may be contradictory or complementary. The actual location information is contained inside a <location-info> element, and there may be one or more actual locations described inside the <location-info> element.

Graphically, the structure of the PIDF/PIDF-LO can be depicted as follows:

```
PIDF document
    tuple 1
        status
        geopriv
            location-info
                civicAddress
                location
                usage-rules
        geopriv 2
        geopriv 3
    ...
    tuple 2
    tuple 3
```

All of these potential sources and storage places for location lead to confusion for the generators, conveyors and users of location information. Practical experience within the United States National Emergency Number Association (NENA) in trying to solve these ambiguities led the following conventions being adopted:

Rule #1: A GeoPriv tuple MUST completely define a specific location.

Rule #2: Where a location can be uniquely described in more than one way, each location description SHOULD reside in a separate tuple.
Rule #3: Providing more than one location in a single presence document (PIDF) MUST only be done if all objects describe the same location.

Rule #4: Providing more than one location in a single <location-info> element SHOULD be avoided where possible.

Rule #5: When providing more than one location in a single <location-info> element they MUST be provided by a common source. If you have more than one location in the <location-info> element, then the combination (complex of) these elements defines the complete location.

Rule #6: Providing more than one location in a single <location-info> element SHOULD only be done if they form a complex to describe the same location. For example, a geodetic location describing a point, and a civic location indicating the floor in a building.

Rule #7: Where a location complex is provided in a single <location-info> element, the higher precision locations MUST be provided first. For example, a geodetic location describing a point, and a civic location indicating the floor MUST be represented with the point first followed by the civic location.

Rule #8: Where a PIDF document contains more than one tuple containing a status element with a geopriv location element, the priority of tuples SHOULD be based on tuple position within the PIDF document. That is to say, the tuple with the highest priority location occurs earliest in the PIDF document. Initial priority SHOULD be determined by the originating UA, the final priority MAY be determined by a proxy along the way.

Rule #9: Where multiple PIDF documents are contained within a single request, document selection SHOULD be based on document order.

The following examples illustrate the useful of these rules.

4.1 Single Civic Location Information

Jane is at a coffee shop on the ground floor of a large shopping mall. Jane turns on her laptop and connects to the coffee-shop’s WiFi hotspot, Jane obtains a complete civic address
for her current location, for example using [5]. She constructs a Location Object which consists of a single PIDF document, with a single geopriv tuple, with a single location residing in the <location-info> element. This is largely unambiguous, and if this location is sent over the network, providing it understands civic addresses, correct handling of any request should be possible.

4.2 Civic and Geospatial Location Information

Mike is visiting his Seattle office and connects his laptop into the Ethernet port in a spare cube. Mike’s computer receives a location over DHCP as defined in [3]. In this case the location is a geodetic location, with the altitude represented as a building floor number. This is constructed by Mike’s computer into the following PIDF document:
The constructed PIDF document contains two `geopriv` elements each in a separate PIDF tuple, the first being a civic address made up of only floor, the second containing the provided geodetic information. If the location is required for routing purposes, which information is used? Applying rule #8, we will likely fail, or at a minimum need to fall back to the second tuple describing the geodetic location, a route described by floor only is precise enough in the normal case to permit route selection. If rule #6 and #7 are applied, then the revised PIDF-LO document would look creates a complex as shown below.
  <tuple id="sg89ab">
    <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:position>
            <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">
              <gml:pos>37.775 -122.4194</gml:pos>
            </gml:Point>
          </gml:position>
          <cl:civilAddress>
            <cl:FLR>2</cl:FLR>
          </cl:civilAddress>
        </gp:location-info>
      </gp:geopriv>
      <timestamp>2003-06-22T20:57:29Z</timestamp>
    </status>
  </tuple>
</presence>

It is now clear that the main location of user is a geodetic location at latitude 37.775 and longitude -122.4194. Further that the user is on the second floor of the building located at those coordinates.

4.3 Manual/Automatic Configuration of Location Information

Erin has a predefined civic location stored in her laptop, since she normally lives in Sydney, the address in her address is for her Sydney-based apartment. Erin decides to visit sunny San Francisco, and when she gets there she plugs in her laptop and makes a call. Erin’s laptop receives a new location from the visited network in San Francisco and adds this to her existing PIDF location document. Applying rule #9, the resulting order of location information in the PIDF document should be San Francisco first, followed by Sydney. Since the information is provided by different sources, rule #8 should also be applied and the information places in different tuples with San Francisco first.
5. Geodetic Coordinate Representation

The geodetic examples provided in [1] are illustrated using the gml:location element which uses the gml:coordinates elements (inside the gml:Point element) and this representation has several drawbacks. Firstly, it has been deprecated in later versions of GML (3.1 and beyond) making it inadvisable to use for new applications. Secondly, the format of the coordinates type is opaque and so can be difficult to parse and interpret to ensure consistent results, as the same geodetic location can be expressed in a variety of ways. An alternative is to use the gml:position and gml:pos elements. These elements have a structured format, in that each field is represented as a double, and a single space exists between each field. Such a format does not introduce the same degree of misinterpretation. The recommended representation therefore for expressing geodetic coordinates for location based routing applications would be:

<gml:position>
  <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">
    <gml:pos>37.775 -122.422</gml:pos>
  </gml:Point>
</gml:position>

The coordinate reference system (CRS) indicates which numbers in the sequence equate to latitude, longitude etc, and in addition to this the CRS also provides an indication of direction represented by the sign of the number. For example, in WGS-84 (represented as CRS: 4326), as shown in the code snippet above, the format is latitude followed by longitude. A positive value for latitude represents a location north of the equator while a negative value represents a location south of the equator. Similarly for longitude, a positive value represents a location east of Greenwich, while a negative value represents a location west of Greenwich.

EPSG 4326 is the two dimensional WGS-84 representation, if we wanted to represented this in three dimensions, that is with an altitude as well, then we would use EPSG 4979 and the format would be as follows:

<gml:position>
  <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4979">
    <gml:pos>37.775 -122.422 22</gml:pos>
  </gml:Point>
</gml:position>

The format using CRS:4979 is similar to CRS:4326, though we now have an altitude value. Specifically the altitude is provided in metres.
above the geoid, which will not be useful for general routing applications since the geoid is generally neither ground-level nor sea-level. However, for more specialized geographic applications it may be useful.
6. Uncertainty in Location Representation

The cellular mobile world today makes extensive use of geodetic based location information for emergency and other location-based applications. Generally these locations are expressed as a point (either in two or three dimensions) and an area or volume of uncertainty around the point. In theory, the area or volume represents a coverage in which the user has a relatively high probability of being found, and the point is a convenient means of defining the centroid for the area or volume. In practice, most systems today use the point as an absolute value and ignore the uncertainty. It is difficult to determine if systems have been implemented in this manner for simplicity, and even more difficult to predict if uncertainty will play a more important role in the future. An important decision is whether an uncertainty area should be specified.

There are six common ways to represent location and uncertainty, but are listed below for completeness:

- Arc band
- Ellipsoid point with uncertainty circle
- Polygon
- Ellipsoid point with altitude
- Ellipsoid point with uncertainty ellipse
- Ellipsoid point with altitude and uncertainty ellipsoid

GML was designed to provide a very flexible abstraction on which specific representations of geometric and geographic schemes could be extended. Representing some of the above shapes is difficult if not impossible using base GML. However, only a subset of GML, namely feature.xsd, is mandatory for a PIDF-LO implementation. Extending GML to easily represent these shapes may lead to interoperability issues and so is not recommended. The authors of this document were unable to find a means to express either an ellipse or an ellipsoid using only the elements defined in feature.xsd.

The following sections describe four shapes that can be defined in GML, and show the equivalent representation in 3GPP MLP [4].

6.1 Arc band

Arc band is used primarily where timing advance (TA) information is
known. Timing advance is a mechanism used in wireless communications
to help ensure that handsets and base-stations remained synchronized.
Timing advance is stepped based on signal propagation and is fairly
deterministic, for GSM each increase in TA value represents 553.85
metres.

The arc band type was developed to represent the area between two
successive TA values and an antenna opening. This is presented in
3GPP as a point, two radii, and two angles representing the start and
the stop of the angles for the opening.

```xml
<pd>
<time utc_off="+1000">20041201092843</time>
<shape>
<CircularArcArea>
<coord>
<X>42.5463</X>
<Y>-73.2512</Y>
</coord>
<inRadius>1938.5</inRadius>
<outRadius>2492.3</outRadius>
<startAngle>63.7</startAngle>
<stopAngle>118.4</stopAngle>
</CircularArcArea>
</shape>
</pd>
```
The GML representation of this is below:

```xml
<?xml version="1.0"?>
<pres xmlns="urn:ietf:params:xml:ns:pidf"
     xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
     xmlns:gml="http://opengis.net/gml"
     entity="pres:user@example.com">
  <tuple id="a6fe09">
    <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:extentOf>
            <gml:Polygon>
              <gml:exterior>
                <gml:Ring>
                  <gml:curveMember>
                    <gml:Curve>
                      <gml:segments>
                        <gml:ArcByCenterPoint>
                          <gml:pos
                            srsName="urn:EPSG:geographicCRS:4326">
                            42.5463 -73.2512
                          </gml:pos>
                          <gml:radius uom="urn:EPSG:uom:9001">
                            2492.3
                          </gml:radius>
                          <!-- It is difficult to determine the correct
                              interpretation of GML and EPSG #4326 to
                              determine how these angles are to be interpreted.
                              Neither specification specifies how the values
                              are to be interpolated. That is, the direction
                              of rotation from start angle to end angle. It
                              is therefore assumed that a "clockwise"
                              (Northing to Easting) direction is chosen to
                              link the two points on the arc.
                              It is also assumed that a value of 0 degrees
                              indicates Northing and 90 degrees indicates
                              Easting. -->
                          <gml:startAngle uom="urn:EPSG:uom:9102">
                            63.7
                          </gml:startAngle>
                          <gml:endAngle uom="urn:EPSG:uom:9102">
                            118.4
                          </gml:endAngle>
                        </gml:ArcByCenterPoint>
                      </gml:segments>
                    </gml:Curve>
                  </gml:Ring>
                </gml:exterior>
              </gml:Polygon>
            </gml:extentOf>
          </gp:location-info>
        </gp:geopriv>
      </status>
    </tuple>
  </pres>
```
Note that the decision to go with a "clockwise" pass means that the start position of this second arc is not contiguous with the end of the last line.

```xml
<!--
Note that the decision to go with a "clockwise" pass means that the start position of this second arc is not contiguous with the end of the last line.
-->  
<gml:LineStringSegment>
  <gml:posList>
    srsName="urn:EPSG:geographicCRS:4326">
      42.535651 -73.224473 42.538018 -73.230411
    </gml:posList>
</gml:LineStringSegment>
<gml:ArcByCenterPoint>
  <gml:pos>
    srsName="urn:EPSG:geographicCRS:4326">
      42.5463 -73.2512
    </gml:pos>
  </gml:ArcByCenterPoint>
  <gml:radius uom="urn:EPSG:uom:9001">
    1938.5
  </gml:radius>
  <gml:startAngle uom="urn:EPSG:uom:9102">
    63.7
  </gml:startAngle>
  <gml:endAngle uom="urn:EPSG:uom:9102">
    118.4
  </gml:endAngle>
</gml:ArcByCenterPoint>
<gml:LineStringSegment>
  <gml:posList>
    srsName="urn:EPSG:geographicCRS:4326">
      42.554016 -73.230007 42.556220 -73.223952
    </gml:posList>
</gml:LineStringSegment>
</gml:LineStringMember>
</gml:Curve>
<gml:Ring>
  <gml:exterior>
    <gml:Polygon>
      <gml:exterior>
        <gp:location-info>
          <gp:usage-rules>
          </gp:usage-rules>
        </gp:location-info>
      </gml:exterior>
    </gml:Polygon>
  </gml:exterior>
</gml:Polygon>
</gml:extentOf>
</gp:location-info>
</gp:usage-rules>
</gp:geopriv>
This representation poses a few potential problems over the 3GPP representation. In the 3GPP representation the point is absolute, and everything else is defined relative to this point, ensuring that the band is indeed bounded. The representation of arc band above does not share all of these properties. In the GML arc band representation above, the point and radii are relative, but the bounding lines of the starting and finishing angles are not, these are necessarily defined as independent line segments. By having to define the arc enclosures as individual line segments it is possible to define an unbounded arc band which would consist of two arcs some arbitrary distance apart with two lines that may or may not intersect them.

A second concern with this representing uncertainty using this method, is that there is no explicit statement or way of indicating to the receiving application what type of uncertainty is being represented. Today several different representations of uncertainty are valid within the same application, so knowing which type is being used, and how to interpret it is important, and this is particularly true if the shape must also be validated as is the case above.

Ensuring the legality of this shape type when represented in GML is more complex than in MLP as the type must first be determined before its validity can be assessed. Users of this shape type may be better served by a formal shape definition being introduced into GeoPriv so that these problems can be more readily overcome.

6.2 Ellipsoidal Point With Uncertainty Circle

This shape type is used extensively over the North American NENA defined E2 interface for transporting mobile geodetic location from the MPC/GMLC to the ALI and subsequently the PSAPs. In 3GPP this is defined as a WGS-84 point (ellipsoid point), and a radius or uncertainty around that point, specified in metres. The 3GPP MLP representation for an ellipsoidal point with uncertainty is defined as follows:
This shape is similarly defined in GML below:
This type does not have all of the problems associated with the arc band representation, in that the radius of the circle is relative to the centre, and so the validation is unnecessary. However it does suffer from the potential problem that the application still needs to determine the type of uncertainty being represented, though this maybe made more clear through the explicit use of the gml:
6.3 Polygon

A polygon is defined as a set of points to form an enclosed bounded shape. It is here that GML and the 3GPP shapes are most similar. The representation for a polygon in GML is given first:

```xml
<?xml version="1.0"?>
     xmlns:gml="http://opengis.net/gml" entity="pres:user@example.com">
  <tuple id="a6fe09">
    <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:extentOf>
            <gml:Polygon>
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList srsName="urn:EPSG:geographicCRS:4326">
                    42.556844 -73.248157
                    42.549631 -73.237283
                    42.539087 -73.240328
                    42.535756 -73.254242
                    42.542969 -73.265115
                    42.553513 -73.262075
                    42.556844 -73.248157
                  </gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:extentOf>
        </gp:location-info>
        <gp:usage-rules>
          </gp:usage-rules>
      </gp:geopriv>
    </status>
    <timestamp>2004-12-13T14:49:53+10:00</timestamp>
  </tuple>
</pres>
```

The GML object here is clear in its definition. A gml:LinearRing MUST have a minimum of four points, with the first and last points
being the same. The 3GPP MLP representation for a polygon is provided below.

```xml
<pd>
  <time utc_off="+1000">20041201092843</time>
  <shape>
    <Polygon>
      <outerBoundaryIs>
        <LinearRing>
          <coord>
            <X>42.556844</X>
            <Y>-73.248157</Y>
          </coord>
          <coord>
            <X>42.549631</X>
            <Y>-73.237283</Y>
          </coord>
          <coord>
            <X>42.539087</X>
            <Y>-73.240328</Y>
          </coord>
          <coord>
            <X>42.535756</X>
            <Y>-73.254242</Y>
          </coord>
          <coord>
            <X>42.535756</X>
            <Y>-73.254242</Y>
          </coord>
          <coord>
            <X>42.542969</X>
            <Y>-73.265115</Y>
          </coord>
          <coord>
            <X>42.553513</X>
            <Y>-73.262075</Y>
          </coord>
          <coord>
            <X>42.556844</X>
            <Y>-73.248157</Y>
          </coord>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </shape>
</pd>
```

While these two representations are very similar and precise, they are not widely used at present. If only a coverage area is required without a nominal central point requiring specification, then this form is ideal for representation using GML.
7. Baseline Geometry

PIDF-LO suggests to use GMLv3 feature.xsd, which provides a subset of the available GML functionality. As a consequence a number of further XML files are implicitly included, namely geometryBasic0d1d.xsd, geometryBasic2d.xsd, temporal.xsd, measure.xsd, units.xsd, gmlBase.xsd, dictionary.xsd, xlink.xsd and basicTypes.xsd, as being necessary to support. This provides for a vast range of possibilities which would pose significant complications to implementors wish to develop location dependent routing applications. By agreeing to a minimal set of data appropriate for routing, a minimum set of GML that MUST be implemented by a given application type can also be set. This does not preclude the additional functionality from being implemented, merely that it may not be understood by some nodes.

7.1 Zero Dimensions

The minimum supported set of elements is position/Point/pos provided by geometryBasic0d1d.xsd.

Thus a point location has only one representation as follows:

```xml
<gml:position xmlns:gml="http://www.opengis.net/gml">
  <gml:Point srsName="urn:ogc:def:crs:EPSG:4326">
    <gml:pos>4.5 -36.2</gml:pos>
  </gml:Point>
</gml:position>
```

The <location> and <coord> objects MUST NOT be used since they are deprecated in GML 3.1 and their functionality can be substituted with the above-described elements.

Note that pos allows altitude to be expressed based on the selected Coordinate Reference Systems (e.g., EPSG:4979 or EPSG:4326). Most Coordinate Reference Systems use altitude above the geoid and not altitude above the ground.

7.2 One Dimensions

Support for one dimensional shapes (such as the LineString or the posList object) is not required except as a part of two dimensional shapes.

geometryBasic0d1d.xsd provides these geometric properties and objects.
7.3 Two Dimensions

The examples previously used were all constructed using elements from this schema which reuse functionality from geometryBasic2d.xsd. As was described earlier the arcband definition in GML is problematic for producing a closed solid and SHOULD consequently be avoided. As a result of this, elements required exclusively for representing the arcband shape have not been included in the minimum supported element set. The minimum element set is therefore restricted to circle and polygon.

Circle:

```
estentOf/
Polygon/
  exterior/
    Ring/
      curveMember/
        Curve/
          segments/
            CircleByCentrePoint/  -> Circle
              pos
              radius
```

Alternatively it would be possible to use the following structure to express a circle using the `<gml:Circle>` element with three pos elements as well. However, the usage of pos and radius, as shown above, is inline with the model used by the 3GPP.

Polygon:

```
estentOf/
Polygon/
  exterior/
    LinearRing/
      pos or posList    -> Polygon
```

7.4 Three Dimensions

Support for three dimensions is not required

7.5 Envelopes

The Envelope element is a representation of a bounding box and can be expressed in two or three dimensions. Defining a space using the Envelope element should be done with extreme caution due to continuity problems at the extremities of the CRS. In WGS-84, two
envelopes are required at the 180th meridian. The minimum set of elements required to support an Envelope are:

    boundBy/
        Envelope/
            upperCorner/
                Point/
                    Pos

    lowerCorner/
        Point/
            Pos/

7.6 Temporal Dimensions

Support for temporal elements is not required.

7.7 Units of Measure

The base SI units as a minimum MUST be supported. For measures of distance this is metres. The EPSG URN for metres is:

    metres = urn:ogc:def:uom:EPSG:9001:6.6

Angles are frequently expressed in terms of both degrees and radians, consequently both MUST be implemented.

    degrees = urn:ogc:def:uom:EPSG:9102:6.6
    radians = urn:ogc:def:uom:EPSG:9101:6.6

Further units of measurement are not required.

7.8 Coordinate Reference System (CRS)

There are a very large number of coordinate reference systems in existence today, but many are, however, not in widespread use. Existing communications protocols such as those used in both the ANSI, 3GPP and NENA standards (see [6], [7], [8]) have standardized on WGS-84. It is recommended for routing purpose that only WGS-84 coordinate types MUST be implemented and further that this set be restricted to the following:

    WGS84(2D) = urn:ogc:def:crs:EPSG:6.6:4326
WGS84(3D) = urn:ogc:def:crs:EPSG:6.6:4979
8. Recommendations

As a summary this document gives a few recommendations on the usage
of location information in PIDF-LO. Nine rules specified in
Section 4 give guidelines on the ambiguity of PIDF-LO with regard
to the occurrence of multiple location information. It is recommend
that gml:position, gml:pos types be used to specify locations when
locations are needed for routing and specifically emergency routing.
Enhancements to GMLv3 feature.xsd may need to be defined to allow
complex shapes types to be specified in a way that makes them easy to
distinguish and validate. This is particularly important if the data
is to be used during the decision making process of routing signaling
messages.

Only a limited subset of GML functionality from the feature.xsd
schema is necessary to describe a geodetic location with sufficient
precision to allow a routing decision to be made. Restricting both
the amount of GML that MUST be implemented, and the number of
variations in which this data can be expressed significantly reduces
the likelihood of interoperability issues in the future. Precedents
exist in the other communications protocols for restricting CRS types
and representations for the sake of simplicity and interoperability,
and the recommendation is made to adopt similar restrictions for
mandatory implementable components of GeoPriv.

If Geodetic information is to be provided via DHCP, then a minimum
resolution of 20 bits SHOULD be specified for both the Latitude and
Longitude fields so that sub 100 metre precision is achieved. Where
only two dimensional objects are required polygons SHOULD be used to
express the enclosed area. Where 3 dimensions are required a 3
dimensional bounding box representing a rectangular prism SHOULD be
used with care taken around the 180th meridian.
9. Security Considerations

The primary security considerations relate to how location information is conveyed and used, which are outside the scope of this document. This document is intended to serve only as a set of guidelines as to which elements MUST or SHOULD be implemented by systems wishing to perform location dependent routing. The ramification of such recommendations is that they extend to devices and clients that wish to make use of such services.
10. IANA Considerations

This document does not introduce any IANA considerations.
11. Acknowledgments

The authors would like to thank the GEOPRIV working group for their discussions in the context of PIDF-LO, in particular James Polk and Henning Schulzrinne. Furthermore, we would like to thanks Jon Peterson as the author of PIDF-LO and Nadine Abbott for her constructive comments in clarifying some aspects of the document.
12. Open Issues

Need to define minimal subset of Civic information that is useful for routing purposes. May be hard to get normative, but hopefully we can get something that is generally representative.

Need agreement on minimal set of shape support.

Need to go through the rules to enhance clarity. These rules are highly likely to be important in quite a number of Location Dependent Routing (LDR) based applications, including ECRIT. General feedback is that they are not clear or precise enough yet. Henning has provided some good feedback here that I have not had time to incorporate yet, some of these comments will hopefully be easier to resolve if open issue 1 above is also resolved.
13. References

13.1 Normative references


13.2 Informative References


[7] "3GPP TS 23.032 V6.0.0 3rd Generation Partnership Project; Technical Specification Group Code Network; Universal Geographic Area Description (GAD)".


Authors’ Addresses

James Winterbottom
Nortel
Wollongong
NSW Australia

Email: winterb@nortel.com

Martin Thomson
Nortel
Wollongong
NSW Australia

Email: martin.thomson@nortel.com

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria  81739
Germany

Email: Hannes.Tschofenig@siemens.com
Appendix A. Creating a PIDF-LO from DHCP Geo Encoded Data

RFC-3825 [3] describes a means by which an end-point may learn its location from information encoded into DHCP option 123. The following section describes how an end-point can take this information and represent it in a well-formed PIDF-LO describing this geodetic location.

The location information described in RFC-3825 consists of a latitude, longitude, altitude and datum.

A.1 Latitude and Longitude

The latitude and longitude values are represented in degrees and decimal degrees. Latitude values are positive if north of the equator, and negative if south of the equator. Similarly longitudinal values are positive if east of the Greenwich meridian, and negative if west of the Greenwich meridian.

The latitude and longitude values are each 34 bit long fields consisting of a 9 bit integer component and a 25 bit fraction component, with negative numbers being represented in 2s complement notation. The latitude and longitude fields are each proceeded by a 6 bit resolution field, the LaRes for latitude, and the LoRes for longitude. The value in the LaRes field indicates the number of significant bits to interpret in the Latitude field, while the value in the LoRes field indicates the number of significant bits to interpret in the Longitude field.

For example, if you are in Wollongong Australia which is located at 34 Degrees 25 minutes South and 150 degrees 32 minutes East this would translate to -34.41667, 150.53333 in decimal degrees. If these numbers are translated to their full 34 bit representations, then we arrive the following:

Latitude = 111011101.10010101010101000111010
Longitude = 0100101101000100010001000010100001

RFC-3825, uses the LaRes and LoRes values to specify a lower and upper boundary for location thereby specifying an area. The size of the area specified is directly related to the value specified in the LaRes and LoRes fields.

Using the previous example, if LaRes is set 7, then lower latitude boundary can be calculated as -256+128+64+16+8+4, which is -36 degrees, the upper boundary then becomes -256+128+64+16+8+4+2+1 which is -35 degrees. LoRes may be used similarly for Longitude.
So what level of precision is useful? Well, certain types of applications and regulations call for different levels of precision, and the required precision may vary depending on how the location was determined. For Cellular 911 calls in the United States, for example, if the network measures the location then the caller should be within 100 metres, while if the handset does the measurement then the location should be within 50 metres. Since DHCP is a network based mechanism we will benchmark off 100 metres (approximately 330 ft) which is still a large area.

For simplicity we shall assume that we are defining a square, in which we are equally to appear anywhere. The greatest distance through this square is across the diagonal, so we make this 100 metres.

```
+----------------------+
|                    _/|
|                  _/  |
|                _/    |
|              _/      |
|            _/        |
|          _/          |
|       100_/ metres   |
|        _/            |
|      _/              |
|    _/                |
|  _/                  |
|_/                    |
+----------------------+
```

The distance between the top and the bottom and the left and the right is the same, the area being a square, and this works out to be 70.7 metres. When expressed in decimal degrees, the third point after the decimal place represents about 100 metre precision, this equates to 10 binary places of fractional part. A 70 metre distance is required, so 11 fractional binary digits are necessary resulting in a total of 20 bits of precision.

With -34.4167, 150.5333 encoded with 20 bits of precision for the LaRes and LoRes, the corners of the enclosing square are:

Point 1 (-34.4170, 150.5332)
Point 2 (-34.4170, 150.5337)
Point 3 (-34.4165, 150.5332)
Point 4 (-34.4165, 150.5337)
A.2 Altitude

The altitude elements define how the altitude is encoded and to what level of precision. The units for altitude are either metres, or floors, with the actual measurement being encoded in a similar manner to those for latitude and longitude, but with 22 bit integer, and 8 bit fractional components.

A.3 Generating the PIDF-LO

If altitude is not required, or is expressed in floors then a geodetic location expressed by a polygon SHOULD be used. If the altitude is expressed in floors and is required, the altitude SHOULD be expressed as a civic floor number as part of the same location-info element. In the example above the GML for the location would be expressed as follows:

```xml
<gml:extentOf>
  <gml:Polygon>
    <gml:exterior>
      <gml:LinearRing>
        <gml:posList srsName="urn:ogc:def:crs:EPSG:6.6:4326">
          -34.4165 150.5332
          -34.4165 150.5337
          -34.4170 150.5537
          -34.4170 150.5532
          -34.4165 150.5332
        </gml:posList>
      </gml:LinearRing>
    </gml:exterior>
  </gml:Polygon>
</gml:extentOf>
```

If a floor number of say 3 were included, then the location-info element would contain the above information and the following:

```xml
<cl:civilAddress>
  <cl:FLR>2</cl:FLR>
</cl:civilAddress>
```

When altitude is expressed as an integer and fractional component, as with the latitude and longitude, it expresses a range, and this cannot be easily expressed in polygon form. Envelopes, as described earlier, define upper and lower bounds for rectangular enclosures, both in two and 3 dimensions, and SHOULD be used where an altitude
range is specified. Care must be taken around the 180th meridian to ensure a misrepresentation does not occur should the 180th meridian be crossed.

Extending the previous example to include an altitude expressed in metres rather than floors. AltRes is set to a value of 19, and the Altitude value is set to 34. Using similar techniques as shown in the latitude and longitude section, a range of altitudes between 32 metres and 40 metres is described. The Envelope would therefore be defined as follows:

```xml
<gml:boundBy>
  <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.6:4976">
    <gml:upperCorner>
      <gml:Point>
        <gml:Pos>-34.4165 150.5337 40</gml:Pos>
      </gml:Point>
    </gml:upperCorner>
    <gml:lowerCorner>
      <gml:Point>
        <gml:Pos>-34.4170 150.5332 32</gml:Pos>
      </gml:Point>
    </gml:lowerCorner>
  </gml:Envelope>
</gml:boundBy>
```

The Method value SHOULD be set to DHCP. Note that this case, the DHCP is referring to the way in which location information was delivered to the IP-device, and not necessarily how the location was determined.

The timestamp value SHOULD be set to the time that location was retrieved from the DHCP server.

The client application MAY insert any usage rules that are pertinent to the user of the device and that comply with [9]. A guideline is that the any retention-expiry value SHOULD NOT exceed the current lease time.

The Provided-By element SHOULD NOT be populated as this is not provided by the source of the location information.

The 3 completed PIDF-LO representations are provided below, and represent a location without altitude, a location with a civic altitude, and a location represented as a 3 dimensional rectangular prism.
<presence xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:pidf="urn:ietf:params:xml:ns:pidf"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    xmlns:gml="http://opengis.net/gml"
    entity="pres:user@example.com">
    <tuple id="a6fea09">
        <status>
            <gp:geopriv>
                <gp:location-info>
                    <gml:extentOf>
                        <gml:Polygon>
                            <gml:exterior>
                                <gml:LinearRing>
                                    <gml:posList
                                        srsName="urn:ogc:def:crs:EPSG:6.6:4326">
                                        -34.4165 150.5332
                                        -34.4165 150.5337
                                        -34.4170 150.5537
                                        -34.4170 150.5532
                                        -34.4165 150.5332
                                    </gml:posList>
                                </gml:LinearRing>
                            </gml:exterior>
                        </gml:Polygon>
                    </gml:extentOf>
                    <gp:usage-rules>
                        <method>DHCP</method>
                    </gp:usage-rules>
                </gp:location-info>
            </gp:geopriv>
        </status>
        <timestamp>2005-07-05T14:49:53+10:00</timestamp>
    </tuple>
</presence>
<?xml version="1.0"?>
<pres xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:pidf="urn:ietf:params:xml:ns:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
xmlns:gml="http://opengis.net/gml"
entity="pres:user@example.com">
<tuple id="a6fe0a09">
<status>
<gp:geopriv>
<gp:location-info>
<gml:extentOf>
<gml:Polygon>
<gml:exterior>
<gml:LinearRing>
<gml:posList
srsName="urn:ogc:def:crs:EPSG:6.6:4326">
-34.4165 150.5332
-34.4165 150.5337
-34.4170 150.5537
-34.4170 150.5532
-34.4165 150.5332
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:extentOf>
<cl:civilAddress>
<cl:FLR>2</cl:FLR>
</cl:civilAddress>
</gp:location-info>
<gp:usage-rules>
</gp:usage-rules>
</gp:geopriv>
<method>DHCP</method>
<timestamp>2005-07-05T14:49:53+10:00</timestamp>
</tuple>
</presence>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:pidf="urn:ietf:params:xml:ns:pidf"
  xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
  xmlns:gml="http://opengis.net/gml"
  entity="pres:user@example.com">
  <tuple id="a6fea09">
    <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:boundBy>
            <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.6:4976">
              <gml:upperCorner>
                <gml:Point>
                  <gml:Pos>-34.4165 150.5337 40</gml:Pos>
                </gml:Point>
              </gml:upperCorner>
              <gml:lowerCorner>
                <gml:Point>
                  <gml:Pos>-34.4170 150.5332 32</gml:Pos>
                </gml:Point>
              </gml:lowerCorner>
            </gml:Envelope>
          </gml:boundBy>
          <gp:usage-rules>
            <method>DHCP</method>
          </gp:usage-rules>
        </gp:location-info>
      </gp:geopriv>
    </status>
    <timestamp>2005-07-05T14:49:53+10:00</timestamp>
  </tuple>
</presence>
Intellectual Property Statement

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

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

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

Disclaimer of Validity

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

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

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