

Internet Engineering Task Force (IETF)
Request for Comments: 5870
Category: Standards Track
ISSN: 2070-1721

A. Mayrhofer
IPCom
C. Spanring
June 2010

A Uniform Resource Identifier for Geographic Locations ('geo' URI)

Abstract

This document specifies a Uniform Resource Identifier (URI) for geographic locations using the 'geo' scheme name. A 'geo' URI identifies a physical location in a two- or three-dimensional coordinate reference system in a compact, simple, human-readable, and protocol-independent way. The default coordinate reference system used is the World Geodetic System 1984 (WGS-84).

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc5870>.

[RFC 5870](#)

'geo' URI Scheme

June 2010

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	4
2.	Terminology	5
3.	IANA Registration of the 'geo' URI Scheme	6
3.1.	URI Scheme Name	6
3.2.	Status	6
3.3.	URI Scheme Syntax	6
3.4.	URI Scheme Semantics	7
3.4.1.	Coordinate Reference System Identification	7
3.4.2.	Component Description for WGS-84	8
3.4.3.	Location Uncertainty	8
3.4.4.	URI Comparison	9
3.4.5.	Interpretation of Undefined Altitude	10
3.5.	Encoding Considerations	10
3.6.	Applications/Protocols That Use This URI Scheme	11
3.7.	Interoperability Considerations	11
3.8.	Security Considerations	11
3.9.	Contact	11
3.10.	Author/Change Controller	12
3.11.	References	12
4.	'geo' URI Parameters Registry	12
5.	URI Operations	13
6.	Use Cases and Examples	13
6.1.	Plain 'geo' URI Example	13
6.2.	Hyperlink	14
6.3.	'geo' URI in 2-Dimensional Barcode	15
6.4.	Comparison Examples	15
7.	GML Mappings	16
7.1.	2D GML 'Point'	17
7.2.	3D GML 'Point'	17
7.3.	GML 'Circle'	17
7.4.	GML 'Sphere'	18
8.	IANA Considerations	18
8.1.	'geo' URI Scheme	18
8.2.	URI Parameter Registry	19

8.2.1.	Registry Contents	19
8.2.2.	Registration Policy	19
8.3.	Sub-Registry for 'crs' Parameter	20
8.3.1.	Registry Contents	20
8.3.2.	Registration Policy	20
9.	Security Considerations	20
9.1.	Invalid Locations	21
9.2.	Location Privacy	21
10.	Acknowledgements	21
11.	References	22
11.1.	Normative References	22
11.2.	Informative References	22

1. Introduction

An increasing number of Internet protocols and data formats are extended by specifications for adding spatial (geographic) location. In most cases, latitude as well as longitude of simple points are added as new attributes to existing data structures. However, all those methods are very specific to a certain data format or protocol, and don't provide a protocol-independent, compact, and generic way to refer to a physical geographic location.

Location-aware applications and location-based services are fast emerging on the Internet. Most web search engines use geographic information, and a vivid open source mapping community has brought an enormous momentum into location aware technology. A wide range of tools and data sets that formerly were accessible to professionals only recently have become available to a wider audience.

The 'geo' URI scheme is another step in that direction and aims to facilitate, support, and standardize the problem of location identification in geospatial services and applications. Accessing information about a particular location or triggering further services shouldn't be any harder than clicking on a 'mailto:' link and writing an email straight away.

According to [[RFC3986](#)], a Uniform Resource Identifier (URI) is "a compact sequence of characters that identifies an abstract or physical resource". The 'geo' URI scheme defined in this document identifies geographic locations (physical resources) in a coordinate reference system (CRS), which is, by default, the World Geodetic

System 1984 (WGS-84) [[WGS84](#)]. The scheme provides the textual representation of the location's spatial coordinates in either two or three dimensions (latitude, longitude, and optionally altitude for the default CRS of WGS-84). An example of such a 'geo' URI follows:

```
geo:13.4125,103.8667
```

Such URIs are independent from a specific protocol, application, or data format, and can be used in any other protocol or data format that supports inclusion of arbitrary URIs.

For the sake of usability, the definition of the URI scheme is strictly focused on the simplest, but also most common representation of a spatial location -- a single point in a well known CRS. The provision of more complex geometries or locations described by civic addresses is out of scope of this document.

The optional 'crs' URI parameter described below may be used by future specifications to define the use of CRSes other than WGS-84. This is primarily intended to cope with the case of another CRS replacing WGS-84 as the predominantly used one, rather than allowing the arbitrary use of thousands of CRSes for the URI (which would clearly affect interoperability). The definition of 'crs' values beyond the default of "wgs84" is therefore out of scope of this document.

This specification discourages use of alternate CRSes in use cases where comparison is an important function.

Note: The choice of WGS-84 as the default CRS is based on the widespread availability of Global Positioning System (GPS) devices, which use the WGS-84 reference system. It is anticipated that such devices will serve as one of the primary data sources for authoring 'geo' URIs, hence the adoption of the native GPS reference system for the URI scheme. Also, many other data formats for representing geographic locations use the WGS-84 reference system, which makes transposing from and to such data formats less error prone (no re-projection involved). It is also believed that the burden of potentially required spatial transformations should be put on the

author rather than the consumer of 'geo' URI instances.

Because of their similar structure, 'geo' URI instances can also be mapped from and to certain ISO 6709 [[ISO.6709.2008](#)] string representations of geographic point locations.

[2.](#) Terminology

Geographic locations in this document are defined using WGS-84 (World Geodetic System 1984), which is equivalent to the International Association of Oil & Gas Producers (OGP) Surveying and Positioning Committee EPSG (European Petroleum Survey Group) codes 4326 (2 dimensions) and 4979 (3 dimensions). This document does not assign responsibilities for coordinate transformations from and to other Spatial Reference Systems.

A 2-dimensional WGS-84 coordinate value is represented here as a comma-delimited latitude/longitude pair, measured in decimal degrees (un-projected). A 3-dimensional WGS-84 coordinate value is represented here by appending a comma-delimited altitude value in meters to such pairs.

Latitudes range from -90 to 90 and longitudes range from -180 to 180. Coordinates in the Southern and Western hemispheres as well as altitudes below the WGS-84 reference geoid (depths) are signed negative with a leading dash.

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](#)].

[3.](#) IANA Registration of the 'geo' URI Scheme

This section contains the fields required for the URI scheme registration, following the guidelines in [Section 5.4 of \[RFC4395\]](#).

[3.1.](#) URI Scheme Name

geo

[3.2.](#) Status

permanent

3.3. URI Scheme Syntax

The syntax of the 'geo' URI scheme is specified below in Augmented Backus-Naur Form (ABNF) [[RFC5234](#)]:

```
geo-URI      = geo-scheme ":" geo-path
geo-scheme   = "geo"
geo-path     = coordinates p
coordinates  = coord-a "," coord-b [ "," coord-c ]

coord-a     = num
coord-b     = num
coord-c     = num

p           = [ crsp ] [ uncp ] *parameter
crsp        = ";crs=" crslabel
crslabel    = "wgs84" / labeltext
uncp        = ";u=" uval
uval        = pnum
parameter   = ";" pname [ "=" pvalue ]
pname       = labeltext
pvalue      = 1*paramchar
paramchar   = p-unreserved / unreserved / pct-encoded

labeltext   = 1*( alphanum / "-" )
pnum        = 1*DIGIT [ "." 1*DIGIT ]
num         = [ "-" ] pnum
unreserved  = alphanum / mark
mark        = "-" / "_" / "." / "!" / "~" / "*" /
             "' " / "(" / ")"
pct-encoded = "%" HEXDIG HEXDIG
```

```
p-unreserved = "[" / "]" / ":" / "&" / "+" / "$"
alphanum     = ALPHA / DIGIT
```

Parameter names are case insensitive, but use of the lowercase representation is preferred. Case sensitivity of non-numeric parameter values MUST be described in the specification of the respective parameter. For the 'crs' parameter, values are case insensitive, and lowercase is preferred.

Both 'crs' and 'u' parameters MUST NOT appear more than once each. The 'crs' and 'u' parameters MUST be given before any other parameters that may be defined in future extensions. The 'crs' parameter MUST be given first if both 'crs' and 'u' are used. The definition of other parameters, and <crslabel> values beyond the default value of "wgs84" is out of the scope of this document. [Section 8.2](#) discusses the IANA registration of such additional parameters and values.

The value of "-0" for <num> is allowed and is identical to "0".

In case the URI identifies a location in the default CRS of WGS-84, the <coordinates> sub-components are further restricted as follows:

coord-a	=	latitude
coord-b	=	longitude
coord-c	=	altitude
latitude	=	["-"] 1*2DIGIT ["." 1*DIGIT]
longitude	=	["-"] 1*3DIGIT ["." 1*DIGIT]
altitude	=	["-"] 1*DIGIT ["." 1*DIGIT]

[3.4.](#) URI Scheme Semantics

Data contained in a 'geo' URI identifies a physical resource: a spatial location identified by the geographic coordinates and the CRS encoded in the URI.

[3.4.1.](#) Coordinate Reference System Identification

The semantics of <coordinates> depends on the CRS of the URI. The CRS itself is identified by the optional 'crs' parameter. A URI instance uses the default WGS-84 CRS if the 'crs' parameter is either missing or contains the value of 'wgs84'. Other <crslabel> values are currently not defined, but may be specified by future documents.

Interpretation of coordinates in the wrong CRS produces invalid location information. Consumers of 'geo' URIs therefore MUST NOT ignore the 'crs' parameter if given, and MUST NOT interpret the

the 'crs' parameter value.

The following component description refers to the use of the default CRS (WGS-84) only. Future documents specifying other 'crs' parameter values MUST provide similar descriptions for the <coordinates> sub-components in the described CRS.

[3.4.2.](#) Component Description for WGS-84

The <latitude>, <longitude>, and <altitude> components as specified in the URI scheme syntax ([Section 3.3](#)) are to be used as follows:

- o <latitude> MUST contain the latitude of the identified location in decimal degrees in the reference system WGS-84.
- o <longitude> MUST contain the longitude of the identified location in decimal degrees in the reference system WGS-84.
- o If present, the OPTIONAL <altitude> MUST contain the altitude of the identified location in meters in the reference system WGS-84.

If the altitude of the location is unknown, <altitude> (and the comma before) MUST NOT be present in the URI. Specifically, unknown altitude MUST NOT be represented by setting <altitude> to "0" (or any other arbitrary value).

The <longitude> of coordinate values reflecting the poles (<latitude> set to -90 or 90 degrees) SHOULD be set to "0", although consumers of 'geo' URIs MUST accept such URIs with any longitude value from -180 to 180.

'geo' URIs with longitude values outside the range of -180 to 180 decimal degrees or with latitude values outside the range of -90 to 90 degrees MUST be considered invalid.

[3.4.3.](#) Location Uncertainty

The 'u' ("uncertainty") parameter indicates the amount of uncertainty in the location as a value in meters. Where a 'geo' URI is used to identify the location of a particular object, <uval> indicates the uncertainty with which the identified location of the subject is known.

The 'u' parameter is optional and it can appear only once. If it is not specified, this indicates that uncertainty is unknown or unspecified. If the intent is to indicate a specific point in space,

<uval> MAY be set to zero. Zero uncertainty and absent uncertainty are never the same thing.

The single uncertainty value is applied to all dimensions given in the URI.

Note: The number of digits of the values in <coordinates> MUST NOT be interpreted as an indication to the level of uncertainty.

3.4.4. URI Comparison

Comparison of URIs intends to determine whether two URI strings are equivalent and identify the same resource (rather than comparing the resources themselves). Therefore, a comparison of two 'geo' URIs does not compare spatial objects, but only the strings (URIs) identifying those objects.

The term "mathematically identical" used below specifies that some components of the URI MUST be compared as normalized numbers rather than strings to account for the variety in string representations of identical numbers (for example, the strings "43.10" and "43.1" are different, but represent the same number).

Two 'geo' URIs are equal only if they fulfill all of the following general comparison rules:

- o Both URIs use the same CRS, which means that either both have the 'crs' parameter omitted, or both have the same <crslabel> value, or one has the 'crs' parameter omitted while the other URI specifies the default CRS explicitly with a <crslabel> value of "wgs84".
- o Their <coord-a>, <coord-b>, <coord-c> and 'u' values are mathematically identical (including absent <uval> meaning undefined 'u' value).
- o Their sets of other parameters are equal, with comparison operations applied on each parameter as described in its respective specification.

Parameter order is not significant for URI comparison.

Since new parameters may be registered over time, legacy implementations of the 'geo' URI might encounter unknown parameters. In such cases, the following rules apply:

- o Two 'geo' URIs with unknown parameters are equivalent only if the same set of unknown parameter names appears in each URI, and their values are bitwise identical after percent-decoding.
- o Otherwise, the comparison operation for the respective URIs is undefined (since the legacy implementation cannot be aware of the comparison rules for those parameters).

Designers of future extension parameters should take this into account when choosing the comparison rules for new parameters.

A URI with an undefined (missing) <coord-c> (altitude) value MUST NOT be considered equal to a URI containing a <coord-c>, even if the remaining <coord-a>, <coord-b>, and 'u' values are equivalent.

For the default CRS of WGS-84, the following comparison rules apply additionally:

- o Where <latitude> of a 'geo' URI is set to either 90 or -90 degrees, <longitude> MUST be ignored in comparison operations ("poles case").
- o A <longitude> of 180 degrees MUST be considered equal to <longitude> of -180 degrees for the purpose of URI comparison ("date line" case).

[3.4.5.](#) Interpretation of Undefined Altitude

A consumer of a 'geo' URI in the WGS-84 CRS with undefined <altitude> MAY assume that the URI refers to the respective location on Earth's physical surface at the given latitude and longitude.

However, as defined above, altitudes are relative to the WGS-84 reference geoid rather than Earth's surface. Hence, an <altitude> value of 0 MUST NOT be mistaken to refer to "ground elevation".

[3.5.](#) Encoding Considerations

The <coordinates> path component of the 'geo' URI (see [Section 3.3](#))

uses a comma (",") as the delimiter for subcomponents. This delimiter MUST NOT be percent-encoded.

It is RECOMMENDED that for readability the contents of <coord-a>, <coord-b>, and <coord-c> as well as <crslabel> and <uval> are never percent-encoded.

Regarding internationalization, the currently specified components do allow for ASCII characters exclusively, and therefore don't require

internationalization. Future specifications of additional parameters might allow the introduction of non-ASCII values. Such specifications MUST describe internationalization considerations for those parameters and their values, and MUST require percent-encoding of non-ASCII values.

[3.6.](#) Applications/Protocols That Use This URI Scheme

As many other URI scheme definitions, the 'geo' URI provides resource identification independent of a specific application or protocol. Examples of potential protocol mappings and use cases can be found in [Section 6](#).

[3.7.](#) Interoperability Considerations

Like other new URI schemes, the 'geo' URI requires support in client applications. Users of applications that are not aware of the 'geo' scheme are likely not able to make direct use of the information in the URI. However, a client can make indirect use by passing around 'geo' URIs, even without understanding the format and semantics of the scheme. Additionally, the simple structure of 'geo' URIs would allow even manual dereference by humans.

Clients MUST NOT attempt to dereference 'geo' URIs given in a CRS that is unknown to the client, because doing so would produce entirely bogus results.

Authors of 'geo' URIs should carefully check that coordinate components are set in the right CRS and in the specified order, since the wrong order of those components (or use of coordinates in a different CRS without transformation) are commonly observed mistakes producing completely bogus locations.

The number of digits in the <coordinates> values MUST NOT be interpreted as an indication of a certain level of accuracy or uncertainty.

[3.8.](#) Security Considerations

See [Section 9 of RFC 5870](#).

[3.9.](#) Contact

Alexander Mayrhofer <axelm@ipcom.at>, <<http://geouri.org/>>

Christian Spanring <christian@spanring.eu>

[3.10.](#) Author/Change Controller

The 'geo' URI scheme is registered under the IETF part of the URI tree. As such, change control is up to the IETF.

[3.11.](#) References

[RFC 5870](#)

[4.](#) 'geo' URI Parameters Registry

This specification creates a new IANA Registry named "'geo' URI Parameters" registry for the <parameter> component of the URI. Parameters for the 'geo' URI and values for these parameters MUST be registered with IANA to prevent namespace collisions and provide interoperability.

Some parameters accept values that are constrained by a syntax definition only, while others accept values from a predefined set only. Some parameters might not accept any values at all ("flag" type parameters).

The registration of values is REQUIRED for parameters that accept values from a predefined set.

The specification of a parameter MUST fully explain the syntax, intended usage, and semantics of the parameter. This ensures interoperability between independent implementations.

For parameters that are neither restricted to a set of predefined values nor the "flag" type described above, the syntax of allowed values MUST be described in the specification, for example by using ABNF.

Documents defining new parameters (or new values for existing parameters) MUST register them with IANA, as explained in [Section 8.2](#).

The 'geo' URI Parameter Registry contains a column named "Value Restriction" that describes whether or not a parameter accepts a value, and whether values are restricted to a predefined set. That column accepts the following values:

- o "No value": The parameter does not accept any values and is to be used as a "flag" only.

- o "Predefined": The parameter does accept values from a predefined set only, as specified in an RFC or other permanent and readily available public specification.
- o "Constrained": The parameter accepts arbitrary values that are only constrained by a syntax as specified in an RFC or other permanent and readily available public specification.

[Section 8.2.1](#) contains the initial contents of the Registry.

5. URI Operations

Currently, just one operation on a 'geo' URI is defined - location dereference: in that operation, a client dereferences the URI by extracting the geographical coordinates from the URI path component <geo-path>. Further use of those coordinates (and the uncertainty value from <uval>) is then up to the application processing the URI, and might depend on the context of the URI.

An application may then use this location information for various purposes, for example:

- o A web browser could use that information to open a mapping service of the user's choice, and display a map of the location.
- o A navigational device such as a Global Positioning System (GPS) receiver could offer the user the ability to start navigation to the location.

Note that the examples and use cases above as well as in the next section are non-normative, and are provided for information only.

[6.](#) Use Cases and Examples

[6.1.](#) Plain 'geo' URI Example

The following 3-dimensional 'geo' URI example references to the office location of one of the authors in Vienna, Austria:

```
geo:48.2010,16.3695,183
```

Resolution of the URI returns the following information:

- o The 'crs' parameter is not given in the URI, which means that the URI uses the default CRS of WGS-84.
- o The URI includes <coord-c>, is hence 3-dimensional, and therefore uses 'urn:ogc:def:crs:EPSG::4979' as the WGS-84 CRS identifier.

- o The <coord-a> value (latitude in WGS-84) is set to '48.2010' decimal degrees.
- o The <coord-b> value (longitude in WGS-84) is set to '16.3695' decimal degrees.
- o The <coord-c> value (altitude in WGS-84) is set to 183 meters.
- o Uncertainty is undefined.

A user could type the data extracted from this URI into an electronic

navigation device, or even use it to locate the identified location on a paper map.

[6.2.](#) Hyperlink

'geo' URIs (like any other URI scheme) could also be embedded as hyperlinks in web pages. A Hyper Text Markup Language (HTML) snippet with such a hyperlink could look like:

```
<p>one of Vienna's popular sights is the  
<a href='geo:48.198634,16.371648;crs=wgs84;u=40'>Karlskirche</a>.
```

Resolution of the URI returns the following information:

- o The 'crs' is given in the URI and sets the CRS used in the URI to WGS-84 explicitly.
- o The URI does omit `<coord-c>`, is hence 2-dimensional, and therefore uses 'urn:ogc:def:crs:EPSG::4326' as the WGS-84 CRS identifier.
- o The `<coord-a>` value (latitude in WGS-84) is set to '48.198634' decimal degrees.
- o The `<coord-b>` value (longitude in WGS-84) is set to '16.371648' decimal degrees.
- o The `<coord-c>` (altitude) value is undefined; therefore, the client MAY assume the identified location to be on Earth's physical surface.
- o The 'u' parameter is included in the URI, setting uncertainty to 40 meters.

A web browser could use this information from the HTML snippet, and offer the user various options (based on configuration, context), for example:

- o Display a small map thumbnail when the mouse pointer hovers over the link.
- o Switch to a mapping service of the user's choice once the link is

selected.

- o Locate nearby resources, for example by comparing the 'geo' URI with locations extracted from GeoRSS feeds to which the user has subscribed.
- o Convert the coordinates to a format suitable for uploading to a navigation device.

Note that the URI in this example also makes use of the explicit specification of the CRS by using the 'crs' parameter.

[6.3.](#) 'geo' URI in 2-Dimensional Barcode

Due to its short length, a 'geo' URI could easily be encoded in 2-dimensional barcodes. Such barcodes could be printed on business cards, flyers, and paper maps, and subsequently used by mobile devices, for example as follows:

1. User identifies such a barcode on a flyer and uses the camera on his mobile phone to photograph and decode the barcode.
2. The mobile phone dereferences the 'geo' URI, and offers the user the ability to calculate a navigation route to the identified location.
3. Using the builtin GPS receiver, the user follows the navigation instructions to reach the location.

[6.4.](#) Comparison Examples

This section provides examples of URI comparison. Note that the unknown parameters 'foo' and 'bar' and unregistered 'crs' values in this section are used for illustrative purposes only, and their inclusion in the examples below does not constitute any formal parameter definition or registration request.

- o The two URIs `<geo:90,-22.43;crs=WGS84>` and `<geo:90,46>` are equal, because both use the same CRS, and even though the longitude values are different, both reflect a location on the north pole (special "poles" rule for WGS-84 applies - longitude is to be ignored). Note that the 'crs' parameter values are case insensitive.

- o The URIs `<geo:22.300;-118.44>` and `<geo:22.3;-118.4400>` are equal, because their coordinate components are mathematically identical.
- o The set of `<geo:66,30;u=6.500;FOo=this%2dthat>` and `<geo:66.0,30;u=6.5;foo=this-that>` are identical, because the value of the unknown parameter 'foo' is bitwise identical after percent-decoding; parameter names are case insensitive, and coordinates and uncertainty are mathematically identical.
- o The comparison operation on `<geo:70,20;foo=1.00;bar=white>` and `<geo:70,20;foo=1;bar=white>` in a legacy implementation is undefined, because the normalization rules for 'foo' are not known, and hence the implementation cannot identify whether or not '1.00' is identical to '1' for the 'foo' parameter.
- o Comparing `<geo:47,11;foo=blue;bar=white>` and `<geo:47,11;bar=white;foo=blue>` returns true, because parameter order is insignificant in comparison operations.
- o The comparison operation on `<geo:22,0;bar=Blue>` and `<geo:22,0;BAR=blue>` is undefined, because even though parameter names are case insensitive, this is not necessarily the case for the values of the unknown 'bar' parameter.

7. GML Mappings

The Geographic Markup Language (GML) by the Open Geospatial Consortium (OGC) is a set of XML schemas that represent geographical features. Since GML is widely accepted, this document includes instructions on how to transform 'geo' URIs from and to GML fragments. The instructions in this section are not normative.

For the following sections, "%lat%", "%lon%", "%alt%", and "%unc%" are placeholders for latitude, longitude, altitude, and uncertainty values, respectively. The mappings use WGS-84 and are defined in the following sections.

Note: GML fragments in other reference systems could be used as well if a transformation into "urn:ogc:def:crs:EPSG::4979" or "urn:ogc:def:crs:EPSG::4326" is defined and applied before the mapping step. Such transformations are typically not lossless.

GML uses the 'double' type from XML schema, and the mapping examples assume that numbers in the form of "3.32435e2" in GML are properly converted to fixed point when placed into the 'geo' URI.

[7.1.](#) 2D GML 'Point'

A 2D GML 'Point' [[RFC5491](#)] is constructed from a 'geo' URI that has two coordinates and an uncertainty ('u') parameter that is absent or zero. A GML point is always converted to a 'geo' URI that has no uncertainty parameter.

'geo' URI:

```
geo:%lat%,%lon%
```

GML fragment:

```
<Point srsName="urn:ogc:def:crs:EPSG::4326"  
  xmlns="http://www.opengis.net/gml">  
  <pos>%lat% %lon%</pos>  
</Point>
```

Note that a 'geo' URI with an uncertainty value of zero is converted to a GML 'Point', but a GML 'Point' cannot be translated to a 'geo' URI with zero uncertainty.

[7.2.](#) 3D GML 'Point'

A 3D GML 'Point' [[RFC5491](#)] is constructed from a 'geo' URI that has three coordinates and an uncertainty parameter that is absent or zero. A GML point is always converted to a 'geo' URI that has no uncertainty parameter.

'geo' URI:

```
geo:%lat%,%lon%,%alt%
```

GML fragment:

```
<Point srsName="urn:ogc:def:crs:EPSG::4979"  
  xmlns="http://www.opengis.net/gml">  
  <pos>%lat% %lon% %alt%</pos>  
</Point>
```

[7.3.](#) GML 'Circle'

A GML 'Circle' [[RFC5491](#)] is constructed from a 'geo' URI that has two coordinates and an uncertainty parameter that is present and non-zero.

'geo' URI:

```
geo:%lat%,%lon%;u=%unc%
```

GML fragment:

```
<gs:Circle srsName="urn:ogc:def:crs:EPSG::4326"  
  xmlns:gml="http://www.opengis.net/gml"  
  xmlns:gs="http://www.opengis.net/pidflo/1.0">  
  <gml:pos>%lat% %lon%</gml:pos>  
  <gs:radius uom="urn:ogc:def:uom:EPSG::9001"  
    %unc%  
  </gs:radius>  
</gs:Circle>
```

[7.4.](#) GML 'Sphere'

A GML 'sphere' [[RFC5491](#)] is constructed from a 'geo' URI that has three coordinates and an uncertainty parameter that is present and non-zero.

'geo' URI:

```
geo:%lat%,%lon%,%alt%;u=%unc%
```

GML fragment:

```
<gs:Sphere srsName="urn:ogc:def:crs:EPSG::4979"  
  xmlns:gml="http://www.opengis.net/gml"  
  xmlns:gs="http://www.opengis.net/pidflo/1.0">  
  <gml:pos>%lat% %lon% %alt%</gml:pos>  
  <gs:radius uom="urn:ogc:def:uom:EPSG::9001"  
    %unc%  
</gs:Sphere>
```

```
</gs:radius>
</gs:Sphere>
```

[8.](#) IANA Considerations

[8.1.](#) 'geo' URI Scheme

This document creates the 'geo' URI scheme in the IETF part of the URI scheme tree, according to the guidelines in [BCP 115 \(RFC 4395\)](#) [[RFC4395](#)]. The definitions required for the assignment are contained in [Section 3](#).

[8.2.](#) URI Parameter Registry

This document creates a new IANA Registry named "'geo' URI Parameters", according to the information in [Section 4](#) and the definition in this section.

[8.2.1.](#) Registry Contents

When registering a new 'geo' URI Parameter, the following information MUST be provided:

- o Name of the Parameter.
- o Whether the Parameter accepts no value ("No value"), values from a predefined set ("Predefined"), or values constrained by a syntax only ("Constrained").
- o Reference to the RFC or other permanent and readily available public specification defining the parameters and the new values.

Unless specific instructions exist for a Parameter (like the definition of a Sub-registry), the following information MUST be provided when registering new values for existing "Predefined" 'geo' URI Parameters:

- o Name of the Parameter.

- o Reference to the RFC or other permanent and readily available public specification defining the new values.

The following table provides the initial values for this registry:

Parameter Name	Value Restriction	Reference(s)
crs	Predefined	[RFC5870]
u	Constrained	[RFC5870]

[8.2.2.](#) Registration Policy

The Registration Policy for 'geo' URI Parameters and their value definitions is "Specification Required" (which implies "Designated Expert"), as defined in [[RFC5226](#)].

[8.3.](#) Sub-Registry for 'crs' Parameter

This document creates a new IANA Sub-registry named "'geo' URI 'crs' Parameter Values", based on the Registry specified in [Section 8.2](#) and the information in this section and [Section 4](#). The syntax of the 'crs' parameter is constrained by the ABNF given in [Section 3.3](#).

[8.3.1.](#) Registry Contents

When registering a new value for the 'crs' parameter, the following information MUST be provided:

- o Value of the parameter.
- o Reference to the RFC or other permanent and readily available public specification defining the use of the CRS in the scope of the 'geo' URI. The specification should contain information that is similar to the WGS-84-specific text given in this document.

- o Reference to the definition document of the CRS. If a URN is assigned to the CRS, the use of such URN as reference is preferred. Note that different URNs may exist for the 2-dimensional and 3-dimensional case.

The following table provides the initial values for this registry:

crs Value	CRS definition(s)	Reference(s)
wgs84	urn:ogc:def:crs:EPSG::4326 urn:ogc:def:crs:EPSG::4979	[RFC5870] [RFC5870]

[8.3.2.](#) Registration Policy

The registration policy for the "'geo' URI 'crs' Parameter Values" Registry shall require both "Specification Required" and "IESG Approval", as defined in [[RFC5226](#)].

[Section 1](#) contains some text about the motivation for when to introduce new 'crs' values.

[9.](#) Security Considerations

Because the 'geo' URI is not tied to any specific protocol and identifies a physical location rather than a network resource, most of the general security considerations on URIs (Section 7 of [RFC 3986](#)) do not apply. However, the following (additional) issues apply:

[9.1.](#) Invalid Locations

The URI syntax ([Section 3.3](#)) makes it possible to construct 'geo' URIs that don't identify a valid location. Applications MUST NOT use URIs with such values and SHOULD warn the user when such URIs are encountered.

An example of such a URI referring to an invalid location would be <geo:94,0> (latitude "beyond" north pole).

[9.2.](#) Location Privacy

A 'geo' URI by itself is just an opaque reference to a physical location, expressed by a set of spatial coordinates. This does not fit the "Location Information" definition according to [Section 5.2](#) of GEOPRIV Requirements [[RFC3693](#)], because there is not necessarily a "Device" involved.

Because there is also no way to specify the identity of a "Target" within the confines of a 'geo' URI, it also does not fit the specification of a "Location Object" ([Section 5.2 of RFC 3693](#)).

However, if a 'geo' URI is used in a context where it identifies the location of a Target, it becomes part of a Location Object and is therefore subject to GEOPRIV rules.

Therefore, when 'geo' URIs are put into such contexts, the privacy requirements of [RFC 3693](#) MUST be met.

[10.](#) Acknowledgements

Martin Thomson has provided significant text around the definition of the "uncertainty" parameter and the GML mappings.

The authors further wish to acknowledge the helpful contributions from Carl Reed, Bill McQuillan, Martin Kofal, Andrew Turner, Kim Sanders, Ted Hardie, Cullen Jennings, Klaus Darilion, Bjoern Hoehrmann, Alissa Cooper, and Ivan Shmakov.

Alfred Hoenes has provided an extremely helpful in-depth review of the document.

[11.](#) References

[11.1.](#) Normative References

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform

Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.
- [RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", [RFC 5491](#), March 2009.

[11.2](#). Informative References

- [RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", [BCP 35](#), [RFC 4395](#), February 2006.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [RFC3693] Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and J. Polk, "Geopriv Requirements", [RFC 3693](#), February 2004.
- [WGS84] National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984, Third Edition", NIMA TR8350.2, January 2000.
- [ISO.6709.2008] International Organization for Standardization, "Standard representation of geographic point location by coordinates", ISO Standard 6709, 2008.

Authors' Addresses

Alexander Mayrhofer
IPCom GmbH
Karlsplatz 1/2/9
Wien A-1010
Austria

Phone: +43 1 5056416 34
Email: alexander.mayrhofer@ipcom.at
URI: <http://www.ipcom.at/>

Christian Spanring
73 Josephine Ave
Somerville 02144

Email: christian@spanring.eu
URI: <http://www.spanring.eu/>

