Network Working Group
Internet-Draft
Expires: July 6, 2007

A. Mayrhofer
enum.at
C. Spanring
OIR-ID
January 02, 2007

A Uniform Resource Identifier for Geographic Locations ('geo' URI)
draft-mayrhofer-geo-uri-00

Abstract

This document specifies an Uniform Resource Identifier (URI) for geographic locations using the 'geo' scheme name. A 'geo' URI provides latitude, longitude and optionally altitude of a location in a simple, human-readable form. The 'geo' URI is not tied to a specific application or protocol.

Table of Contents

1. Changes & Supplemental Information

[Note to editors: This section is to be removed before publication - XML source available on request]

- draft-mayrhofer-geo-uri-00
  - initial draft

A supplemental web site for the development of this draft and the 'geo' URI in general has been set up at <http://geouri.org/>

2. Introduction

An increasing number of Internet protocols and data formats are being enriched by specifications on how to add information about geographic location to them. In most cases, latitude as well as longitude are added as attributes to existing data structures. However, all those methods are specific to a certain application, data format or protocol, and don't provide a generic way to protocol independent location identification.

Over the past few years, emerging location aware applications and location based services were observable on the Internet. Most Internet search engines and a vivid open source mapping community brought an enormous momentum into location aware technology. A wide range of tools and data formerly available to professionals only were provided free of charge for everyday use on the mass market.

The 'geo' Uniform Resorce Identifier (URI) [1] scheme is another step into that direction and aims to facilitate, support and standardize part of the interaction with geospatial services and applications. Accessing information about or trigger further services based on a particular place on earth shouldn't be any harder than writing an email by clicking on a 'mailto:' link.

A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource. This document specifies the 'geo' URI scheme for identifying geographic locations in the WGS84 [5] reference system, independent of any specific application, data format or protocol.

'Geo' URIs identify a geographic location by the textual representation of the location's spatial coordinates in either two or 3 dimensions (latitude, longitude, and optionally altitude). An optional query string contains additional parameters.

The provision of civic addresses (street, city, country, etc.) to identify locations is out of scope for the 'geo' URI scheme.

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 RFC 2119 [2].

4. IANA Registration of the 'geo' URI Scheme

This section contains information required for the URI scheme registration, following the guidelines in section 5.4 of RFC 4395 [4].

4.1. URI Scheme Name

geo

4.2. Status

permanent

4.3. URI Scheme Syntax

The syntax of the 'geo' URI scheme is specified below in Augmented Backus-Naur Form (ABNF) [3]:

```
geo-URI       =  geo-scheme ":" geo-path [ "?" geo-query ]
geo-scheme    =  "geo"
geo-path      =  geo-location
geo-query     =  query

geo-location  =  latitude "," longitude [ "," altitude ]

latitude      =  [ "-" ] 1*2DIGIT [ "." *DIGIT ]
longitude     =  [ "-" ] 1*3DIGIT [ "." *DIGIT ]
altitude      =  [ "-" ] *DIGIT [ "." *DIGIT ]
```

The 'query' component is specified in secion 3.4 of RFC 3986.

4.4. URI Scheme Semantics

Generally, data contained in a 'geo' URI describes the geographic coordinates of the identified location, and contains an optional query string.

Note: In order to achieve high user acceptance it seems inevitable to adopt commonly known GPS parameters (latitude, longitude, altitude) where possible.

4.4.1. Component Description

The "latitude", "longitude", "altitude" and "query" components as specified in the URI scheme syntax (Section 4.3) are to be used as follows:

- The "latitude" component MUST contain the decimal latitude of the identified location in the reference system WGS84.
- The "longitude" componont MUST contain the decimal longitude of the identified location in the reference system WGS84.
- If present, the OPTIONAL "altitude" component MUST contain the WGS84 decimal altitude of the identified location in meters (elevation above mean seal level).

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

The number of decimal places indicates the precision of the value. As one degree equals 111319.45m at the equator (40075.004km / 360 degrees), five decimal places (0.00001 degree) correspond to roughly one meter, which seem to provide sufficient accuracy for civil use.

4.4.2. URI Comparison

Two 'geo' URIs MUST be considered equal when their 'longitude', 'latitude' and 'altitude' values are mathematically identical, and their decoded 'query' strings match.

An URI with undefined (missing) 'altitude' value MUST NOT be considered identical to an URI with an 'altitude' value, even if the remaining components 'latitude', 'longitude' and 'query' match.

4.4.3. Interpretation of Undefined Altitude

A consumer of an 'geo' URI with undefined 'altitude' MAY assume that the URI refers to the location on earth's surface at the given 'latitude' and 'longitude' coordinate.

4.5. Encoding Considerations

The 'geo-location' path component of the 'geo' URI (see Section 4.3) uses a comma (",") as a delimiter for subcomponents. This delimiter MUST NOT be percent encoded.

It is RECOMMENDED that for readability the contents of 'latitude', 'longitude' and 'altitude' subcomponents are never percent encoded.

Contents of the 'query' component is to be encoded according to RFC 3986.

4.6. Applications/protocols That Use This URI Scheme

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.

4.7. Interopability Considerations

While the interpretation of 'latitude', 'longitude' and 'altitude' is quite clear, the interpretation of the 'query' component may vary by application or protocol to which a 'geo' URI is being mapped. Consumers MUST ignore unknown query parameters they encounter while authors of 'geo' URIs SHOULD only use well known parameters in the 'query' component.

To reduce interopability issues, it might be neccessary to create a registry of query parameters.

4.8. Security Considerations

See Section 8 of [insert reference to this document]

4.9. Contact

Christian Spanring (mailto:spanring@oir.at, http://spanring.eu/), Alexander Mayrhofer (mailto:alexander.mayrhofer@enum.at, http://nona.net/)

4.10. 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.

4.11. References

The 'geo' URI is specified in [insert reference to this document].

5. Use of 'geo' URIs

5.1. URI Construction

The production of a 'geo' URI involves the following steps:

1. Aquire coordinates of the location to be identified: latitude, longitude, and optionally altitude.
2. If coordinates are given in degrees/minutes/seconds, transform them to their respective decimal representation.
3. Ensure that coordinates are represented in the correct reference system / units as described in Section 4.4.1. Transform coordinates if neccessary.
4. Round coordinate values to a sensible number of decimal places, according to the presumed accuracy of the data source used.
5. Remove abundant leading zero's from values, keeping at least one digit in the integer part. Make sure that negative values have a minus sign, while positive values don't have a sign.
6. concatenate prepared latitude, longitude, optionally altitude strings with the subcomponent delimiter ",", not adding any whitespace or other characters.
7. prepend the result with the string 'geo:' (the URI scheme and delimiter)
8. if the final URI is to include a 'query' component, add the component delimiter "?" to the end of the result, followed by the encoded query string.

The following example constructs a 'geo' URI from the location message of a Global Positioning System (GPS) receiver:

1. Aquire coordinates (data split in two lines):

   $GPGGA,124951.000,4812.0556,N,01622.1729,E,1,
   05,3.3,192.4,M,43.4,M,,0000*5D
   latitude = 48 degrees, 12.0556 minutes north.
   longitude = 16 degrees, 22.1729 minutes east.
   altitude = 192.4 meters (using a geoid height of 43.4 meters)
   Horizontal Dilution of Precision (HDOP) = 3.3 (corresponds to about 10 meters)

2. Transform into decimal values:

   latitude = +48.2009266666 degrees
   longitude = +16.3695483333 degrees
   altitude = 192.4 meters

3. Reference system: longitude and latitude values are already given in WGS84, altitude already refers to mean sea level (according to the geoid correction value of 43.4 meters)
4. Round values:

   latitude = +48.20093 degrees
   longitude = +016.36955 degrees
   altitude = 192 meters

(five decimal rougly correlate to about one meter on the equator. This is more precise than the indicated HDOP of about 10 meters).

5. Remove abundant stuff:

   latitude = 48.200927
   longitude = 16.369548
   altitude = 192

6. Concatenate:

   48.200927,16.369548,192

7. Prepend with URI scheme name and delimiter:

   geo:48.200927,16.369548,192

8. In this example, no query component is to be added.

5.2. URI Dereference

A consumer of a 'geo' URI has to perform the following operation for dereference:

1. Validate the 'geo' URI against the syntax specification (Section 4.3). URIs which do not match the syntax SHOULD NOT be used (see note below).
2. Remove the URI scheme and the URI delimiter ("geo:") from the beginning of the string.
3. If the string contains a query delimiter ("?"), split off the 'query' component and the query delimiter
4. Split the remaining string into subcomponents, using the comma (",") as the delimiter
5. Use the first subcomponent as latitude, the second one as longitude. If a thrid subcomponent is present, use it as altitude.

Note: An application MAY use URIs which contain whitespace in the 'geo-location' component by stripping it off before the dereference process.

5.3. URI Operations

Currently, just one operation on a 'geo' URI is defined - location retrieval: In that operation, a client uses the data from the URI to retrieve the geographical location to which the URI refers to.

A client may then, though, use this location information for various

purposes:

A web browser may rewrite that information into the URI of a web mapping service of the user's choice, and display a map of the location
A navigational device such as a Global Positioning System (GPS) receiver may offer the user to start navigation to the location.
Examples of such uses can be found in Section 6.

6. Examples and Use Cases

6.1. Plain 'geo' URI

The following 3-dimensional 'geo' URI example references to the bottom of the stairs leading to the Karlskirche in Vienna, Austria:

<geo:48.19858,16.37164,171>

A user could type the data extracted from this URI into a electronic navigation device, or even use it to locate the identified location on a paper map.

6.2. Hyperlink

'geo' URIs could (like any other URI scheme) 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 most popular sights is the <a href='geo:
48.19858,16.37164'>Karlskirche</a>
```

6.3. Header Field

Many protocols support the use of arbitrary URI schemes, for example in their header Fields. A Session Initiation Protocol (SIP) [7] REGISTER request could contain a 'Contact' header with a 'geo' URI, to reflect the geographic location to be used to contact the registering entity physically:

```
REGISTER sip:geoaware.example.com SIP/2.0
Via: SIP/2.0/UDP mypc.example.org:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
To: Joe Geo <sip:joe@example.com>
From: Joe Geo <sip:joe@example.com>;tag=456248
Call-ID: aaafafff-84230@7afagggdd
CSeq: 42 REGISTER
Contact: <sip:joe@192.168.1.1>
```

```
Contact: <geo:48.19858,16.37164>
```

6.4. Web Mapping Services

A rather common method for accessing geographic information on the Internet are web mapping services (WMS) [6]. An image containing a map is delivered by a mapserver as response to a WMS request. WMS requests usually contain a bounding box (enclosing rectangle), the spatial reference system (e.g. 'EPSG:4326' for WGS84), displayed layers (e.g. roads, borders), image dimensions and image format (e.g. 'image/png'):

```
http://map.example.org/maps/wms.cgi?BBOX=16,48,18,50&SRS=EPSG:4326 \
&LAYERS=roads,borders&WIDTH=400&HEIGHT=400&FORMAT=image/png
```

A location identified by a 'geo' URI could be used to support input parameters in terms of a center point of a WMS request. Query parameters could be used to reflect the requested type of service, eg. a 'geo' URI could be mapped to a WMS request as follows:

```
geo:48.20833,16.37278,171?service=wms&scale=5000&layers=roads,borders
\ &height=400&width=400&format=image/png
```

A bounding box can be calculated by given scale, height and width. In our case, based on an output scale of 1:5000 and 400px image height and width the bounding box width and height is 0.00634 degrees. A 'geo' URI is limited to one spatial reference system, thus 'SRS=EPSG:4326' is a constant parameter in WMS transformations. The resulting WMS request could look like:

```
http://map.example.org/maps/wms.cgi?BBOX=16.05690,47.89167, \
16.69142,48.52619&SRS= EPSG:4326&LAYERS=roads,borders&WIDTH=400 \
&HEIGHT=400&FORMAT=image/png
```

7. IANA Considerations

This document requests assignment of the 'geo' URI scheme in the IETF part of the URI scheme tree, according to the guidelines in BCP 115 (RFC 4395) [4]. The definitions required for the assignment are contained in Section 4.

8. 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:

8.1. Invalid Locations

The URI syntax (Section 4.3) makes it possible to construct valid 'geo' URIs which don't identify a valid location on earth. Applications MUST NOT use URIs which such invalid values, and SHOULD warn the user when such URIs are encountered.

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

8.2. Location Privacy

Location information about individuals is an extremely sensitive topic, especially when location is combined with Personally Identifyable Information (PII).

In cases where location information about individuals is used in a publicly available 'geo' URI, the explicit consent of the individual is REQUIRED.

8.3. Malicious Locations

As with other URI schemes, the information provisioned in 'geo' URIs cannot be trusted unless some kind of trust relation with the author of a URI is in place. Applications of the 'geo' URI SHOULD consider methods of validating and protecting URIs.

9. References

9.1. Normative References

[1] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[3] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

9.2. Informative References

[4] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", BCP 115, RFC 4395, February 2006.

[5] National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984, Third Edition", NIMA TR8350.2, January 2000.

[6] Open GIS Consortium Inc., "Web Map Service Implementations Specification, Version 1.1.1", OGC 01-068r3, January 2002.

[7] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

Authors' Addresses

Alexander Mayrhofer
enum.at GmbH
Karlsplatz 1/9
Wien A-1010
Austria

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

Christian Spanring
OIR-Informationsdienste GmbH
Franz-Josefs-Kai 27
Wien A-1010

Phone: +43 1 5338747 36
Email: spanring@oir.at
URI: http://www.oir.at/

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).