How to Store Geo Coordinates on an NFC Tag

This article shows how to store geo coordinates of a place (longitude and latitude), as you'd get from GPS, on an NFC Tag.

Introduction

Have you ever been in the situation where you read about a sight, city or a country, and wanted to know where it is? If the brochure includes an NFC tag that stores geo information, you'd just need to tap it with your phone and instantly see the location in the maps app. Other use cases include geocaching or a treasure hunt game, where you could store the final coordinates on a tag, which are only revealed by tapping the tag with the phone.

To enable these use cases, the tag needs to contain the coordinates of the place: longitude and latitude, as you'd get from GPS. This article takes a look at alternatives how to store those, to enable cross platform use. Large parts of these findings are also relevant for simple URL links on the web or 2D Barcodes, and don't need to be on an NFC tag.

Representing Geo Coordinates

Unfortunately, no format to store coordinates has been standardized by the NFC Forum so far. The NDEF URI RTD contains several common URI types (like obviously *http:*, as well as *tel:* or *sip:*), but no special recommendation for geo coordinates. Therefore, some platforms might not handle geo coordinate links at all, or the default apps might listen to different URI schemes. It is currently not possible to create a single Geo Tag that directly works for all platforms. As a consequence, you have the choice between using:

- Special URI schemes
- Links to a web-based maps client
- Redirection script that determines the client's operating system

URI Schemes

Different URI schemes (also commonly called URL schemes) are in use for geo coordinates. Which one is best depends on the operating system of your customers. For each URI scheme, this section contains a short overview of the standard, an example that encodes the decimal coordinates with latitude of 60.17 and longitude 24.829 in WSG-84, as well as which platforms have built-in support for understanding and handling a URL that is formatted by that scheme (= without downloading a 3rd party app from the store).

Geo: URI Scheme

The geo: URI scheme (RFC 5870) proposes an easy way to encode latitude and longitude into an URI. The scheme itself can also encode more information into the URI, but for showing a place on a map and especially when considering the usually very limited writable space on a typical NFC tag, you will be happy to have longitude, latitude and the title of the place written (by using a Smart Poster record).

Format:

geo:{0},{1}
{0} … latitude, {1} … longitude

Example:
geo:60.17,24.829

Platforms with native support:

- MeeGo
- Android

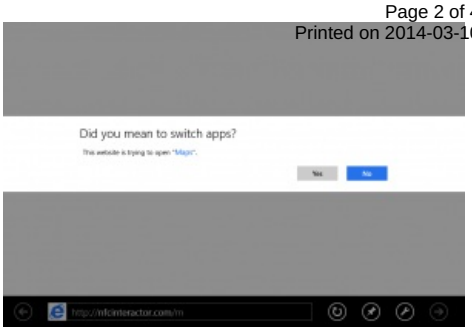
The Maps client on Windows (Phone) 8 doesn't register for this URI scheme. You can of course write an own maps client that does; but this would require the user to download an extra app from the Windows store.

Bing Maps URI Scheme

Used by the default Maps client on Windows 8. When the user encounters such a

(C) Copyright Nokia 2014. All rights reserved.

URL, it will open the maps client and show the map at the specified position. The screenshot on the right shows hwo the prompt looks like, if the Windows 8 Internet Explorer encounters such a link. More information about this URI scheme and how to use it from a C# app was blogged by dream team.



Format:

bingmaps://?cp={0}~{1}
{0} … latitude, {1} … longitude

Example:
bingmaps://?cp=60.17~24.829

Platforms with native support:

- Windows 8

Maps URI Scheme

Used by the Bing Maps client on Windows Phone 7. Obviously, WP7 devices generally don't have NFC support, but you could use this URI scheme for links on the web, or the redirection script.

That this scheme is also connected to Bing Maps on WP8 - however, the app starts to search for the coordinates, but then gives up and does NOT center the map on the specified coordinates! You can use the scheme to search for a place name, however.

Note that this scheme also launches Bing Maps on Nokia Lumia WP8 phones, where Bing Maps is not visible, as it has been replaced by Nokia Maps - however, Bing Maps is still present on the phone and can be launched through this scheme.

Format:

maps:{0} {1}
{0} … latitude, {1} … longitude

Example:
maps:60.17 24.829

Platforms with native support:

- Windows Phone 7.x / (8.x: doesn't work with coordinates, always launches Bing Maps)

Navigation URI Schemes

Microsoft defined additional URI schemes that work only on Windows Phone 8 and trigger starting navigation using the main maps client right away.

These schemes only define the target of the navigation, but not the source to specify a complete route; the starting point for navigation is always the user's current GPS position.

A severe limitation is that the schemes only show the target if the client is indeed capable of calculating a route. E.g., if you're in Austria and want open the link to navigate to Australia, the route calculation in the maps app will fail, and the user won't actually see the target position.

The scheme can also be used in app-to-app scenarios, e.g., to launch a navigation task from within your own WP8 app.

Format:

ms-drive-to:?destination.latitude={0}&destination.longitude={1}
ms-walk-to:?destination.latitude={0}&destination.longitude={1}
{0} … latitude, {1} … longitude

Examples:
ms-drive-to:?destination.latitude=60.17&destination.longitude=24.829
ms-walk-to:?destination.latitude=60.17&destination.longitude=24.829

Platforms with native support:

- Windows Phone 8.x

Links to a Web-Based Maps Client

(C) Copyright Nokia 2014. All rights reserved.

If a client doesn't support a URI scheme or doesn't have a pre-installed maps app that you could directly open, you can link to a web-based maps site, like Nokia Maps, Bing Maps, Open Street Maps or Google Maps. The maps web site should then automatically detect the client version and optimize its user experience. For example, Nokia Maps provides the full desktop maps client if the link is opened from a PC, or the mobile HTML5 version if opened from Android or an iPhone.

Here, we're taking a look at Nokia Maps. You can of course create similar links for other map providers.

Format:

http://m.nokia.me/?c={0},{1}
{0} … latitude, {1} … longitude

Example:
http://m.nokia.me/?c=60.17,24.829

Note: As a special note: if you use the "old" URL m.ovi.me instead of m.nokia.me, this has the additional advantage that Symbian phones recognize the URL and still open the full, native maps client. On some platforms, this might only provide a static map image, however, and therefore not be the best user experience.

Web Redirection Script

To create a truly cross-platform Geo Tag (or a link on a web page), which provides the best possible user experience on any platform, you need to create a special web script that determines the client's operating system, and then redirects to the best URI scheme or web based client for that particular platform. The disadvantage is of course that the browser will always (at least briefly) open on the user's device; but given the advantage of giving the best possible user experience, this trade-off is worth considering.

To create a working redirection, you need to determine two factors, depending on the platform:

- **Redirection URI:** Either a URI scheme, or a link to a web-based maps client
- **Redirection Method:** Either by sending the browser a header that directly points to the location, or through a JavaScript redirect

For example, if you find out that the user agent of the browser indicates Windows 8 as the client's OS, you'd format the location to contain the ~ character as the separator between latitude and longitude, and use JavaScript redirection. Find the full code in the download section below, the principle works as follows:

```
<?php
// Latitude and longitude - you can also get this from the query parameters
$location = "60.17,24.829";
$latlong = str_split($location, ",");

// Determine the user agent string
$user_agent = $_SERVER['HTTP_USER_AGENT'];

// Check client's platform
if (preg_match('/WPDesktop/i',$user_agent)) {
    $ua_enc = 'wp8';    // IE 10 on WP8 in desktop mode
    $locationUri = "ms-drive-to:?destination.latitude=" + $latlong[0] +
"&destination.longitude=" + $latlong[1];
} elseif (preg_match('/windows phone 8/i',$user_agent)) {
    $ua_enc = 'wp8';    // IE 10 on WP8 in mobile mode
    $locationUri = "ms-drive-to:?destination.latitude=" + $latlong[0] +
"&destination.longitude=" + $latlong[1];
} elseif (preg_match('/Windows NT 6.2/i',$user_agent)) {
    $ua_enc = 'w8';        // Windows 8
    $locationTilde = str_replace(',', '~', $location);
    $locationUri = "bingmaps://?cp=".$locationTilde;
} elseif (preg_match('/windows phone/i',$user_agent)) {
    $ua_enc = 'wp7';    // Windows Phone 7
    $locationSpace = str_replace(',', ' ', $location);
    $locationUri = "maps:".$locationSpace;
```


(C) Copyright Nokia 2014. All rights reserved.

```
}
?>
```

```
<html>
<head>
<script type="text/javascript">
<!--
window.location = "<?php echo $locationUri; ?>";
//-->
</script>
</head>
</html>
```

Download

To easily format and write Geo URI to a NFC tag, you can use the NdefGeoRecord class of the NDEF Library for Proximity APIs (LGPL license). The full code for a PHP-based web redirection script is also part of the library and can be hosted on your server, or tested at http://nfcinteractor.com/m?c=60.17,24.829.

References

You can read more information about the geo: URI scheme, which supports encoding more than just latitude and longitude (which are the two most important for the limited writable size of typical NFC tags). More information about the geo tag redirection script is also available on the Nfc Interactor project site.

You can also read the blog post about Geo Tags from March 2012, with a more Nokia-focused perspective.

For more information about using NFC on Windows 8, take a look at the Proximity API specification. The article "how to aquire and publish content from NFC tags and proximity peers" is recommended to get an understanding of how to use the APIs to read and write tags.

(C) Copyright Nokia 2014. All rights reserved.