



[Location-Aware Applications](#)

By Richard Ferraro and Murat Aktihanoglu

In this article, based on chapter 4 of [Location-Aware Applications](#), the authors discuss the strengths and weaknesses of three content distribution formats for location-based service (LBS) applications—GeoRSS, GeoJSON, and KML.

To save 35% on your next purchase use Promotional Code **ferraro0435** when you check out at www.manning.com.

[You may also be interested in...](#)

Content Distribution Formats

Most of the content for location-based service (LBS) applications comes as *data feed* in different distribution formats. The top distribution formats for LBS applications are GeoRSS, GeoJSON, and Keyhole Markup Language (KML). Each of these formats has its own strengths and weaknesses, so let's look at each of them closely.

GeoRSS

GeoRSS is an emerging standard for encoding location as part of a data feed. It is derived from Really Simple Syndication (RSS), the most popular data feed and syndication format used by blogs and websites.

GeoRSS so far has two primary encodings that are called GeoRSS Geography Markup Language (GML) and GeoRSS Simple. GeoRSS GML is a formal Open Geospatial Consortium (OGC) GML Application Profile and supports a greater range of features than GeoRSS Simple, notably coordinate reference systems other than WGS84 latitude/longitude. There is also a W3C GeoRSS serialization, which is older and partly deprecated but still the most widely used.

Listings 1 and 2 display some GeoRSS-format data samples in two different flavors.

Listing 1 Sample GeoRSS Simple feed

```
<?xml version="1.0" encoding="utf-8"?>

<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:georss="http://www.georss.org/georss"> #1

  <title>Recommendations</title> #2
  <subtitle>Today's top recommendations</subtitle>
  <link href="http://centrl.com"/>
  <updated>2008-12-13T18:30:02Z</updated>
  <author>
    <name>Centrl</name>
    <email>support@centrl.com</email>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <entry> #3
    <title>Sushi Yasuda</title> #4
```

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/ferraro/>

```

<link href="http://centrl.com/location/ba44373ba0521e0e3cb7210"/>
<id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
<updated>2008-08-17T07:02:32Z</updated>
<summary>Great sushi</summary>
<georss:point>40.714172,-74.006393</georss:point>
</entry>
<entry>
<title>Blue Ribbon Sushi</title>
<link href="http://centrl.com/location/ff43473ba0521ade3cb1828"/>
<id>urn:uuid:324356321-46fb8-4ebb-aaaa-80da344ef321</id>
<updated>2009-08-17T07:02:32Z</updated>
<summary>Festive atmosphere, excellent sake selection and great sushi</summary>
<georss:point>40.714321,-74.12393</georss:point>
</entry>
</feed>

```

- #1 Specify the attributes of the feed tag**
- #2 The feed tag contains various tags related to the specific feed**
- #3 The feed tag also contains a number of entry tags**
- #4 Each entry contains a number of property tags related to the entry**

This sample displays how to specify the attributes of the feed tag (#1) and that each feed tag may contain many other property tags (#2) as well as other entry tags (#3). Each entry tag in turn may contain many property tags (#4).

Listing 2 Sample W3C GeoRSS feed

```

<?xml version="1.0"?>
<?xml-stylesheet href="/centrlstyle.xml" type="text/xsl" media="screen"?>

<rss version="2.0"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  #1

  <channel>
  #2
    <title>Recommendations from Centrl.com</title>
    <description>Today's top recommendations</description>
    <link>http://centrl.com/</link>
    <dc:publisher>Centrl</dc:publisher>
    <pubDate>Thu, 27 Aug 2008 23:56:15 PST</pubDate>

    <item>
    #3
      <pubDate> Thu, 27 Aug 2008 22:52:15 PST </pubDate>
      #4
      <title>Sushi Yasuda</title>
      <description>Great sushi</description>
      <link>http://centrl.com/location/ba44373ba05a/</link>
      <geo:lat>40.714172</geo:lat>
      <geo:long>-74.006393</geo:long>
    </item>
    <item>
      <pubDate> Thu, 28 Aug 2008 22:52:15 PST </pubDate>
      <title>Blue Ribbon Sushi</title>
      <description>Festive atmosphere, excellent sake selection and great
sushi</description>
      <link>http://centrl.com/location/ ff43473ba0521ade3cb1828</link>
      <geo:lat>40.714321</geo:lat>
      <geo:long>-74.12393</geo:long>
    </item>
  </channel>
</rss>

```

- #1 Specify the attributes of the rss tag**
- #2 Specify the attributes of the channel tag inside the rss tag**
- #3 Each channel contains a number of item tags**
- #4 Each item contains a number of property tags related to the item**

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/ferraro/>

This sample displays how to specify the attributes of the rss tag (#1) and that each feed tag may contain many other property tags (#2) as well as other item tags (#3). Each item tag, in turn, may contain many property tags (#4).

GeoJSON

Based on JavaScript Object Notation (JSON), GeoJSON is a new data format for encoding a wide variety of geographic features, including points, line-strings, polygons, multi-polygons, and geometry collections. Figure 1 is the OpenLayers GeoJSON tool that makes it easy to create polygons and shapes in GeoJSON¹.

Vector Formats Example

Shows the wide variety of vector formats that open layers supports.



Use the drop-down below to select the input/output format for vector features. New features can be added by using the drawing tools above or by pasting their text representation below.

Format Pretty print
 Input Projection:
 Output Projection:

Use the tools to the left to draw new polygons, lines, and points. After drawing some new features, hover over a feature to see the serialized version below.

```
{
  "type": "Feature",
  "id":
  "OpenLayers.Feature.Ve
ctor_128",
  "properties": {
  },
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          7.03125,
          27.7734375
        ],
        [
          -2.109375,
```

Figure 4.2 OpenLayers GeoJSON tool showing the GeoJSON encoding of shapes drawn on the map

GeoJSON is a more compact data format than GeoRSS, which is based on XML. So when compactness is important, GeoJSON presents a good solution.

Listing 3 displays some sample GeoJSON-format data to give you an idea about what this format looks like.

Listing 3 Sample GeoJSON feed

```
{
  "type": "Feature",
  "geometry": {
    "type": "GeometryCollection",
    "geometries": [
      {
        "type": "Point",
        "coordinates": [-74.006393,40.714172]
      },
      {
        "type": "LineString",
```

¹ <http://openlayers.org/dev/examples/vector-formats.html>

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/ferraro/>

```

        "coordinates": [
            [101.0, 0.0], [102.0, 1.0]
        ]
    }
]
},
"properties": {
    "name": "Sushi Yasuda",
    "category": "Restaurant",
    "link": "http://centrl.com/location/ba44373ba05a",
    "source": "http://centrl.com",
    "description": "Great Sushi"
}
}
}

```

- #A JSON is pure javascript syntax, hence can be parsed with 'eval' directly**
- #1 Here is a key-value pair**
- #2 Here is a key-object pair**
- #3 The geometry object contains bunch of key-value pairs and also some key-object pairs**
- #4 A mixed array of geometry objects**
- #5 A collection of key-value pairs related to this object**

This JSON example displays a simple key-value pair (#1) and a key-object pair (#2) as well as a geometry object that contains bunch of key-value pairs and also some key-object pairs (#3). Each geometry collection is a mixed array of geometry objects (#4). And each object can contain a collection of property key-value pairs.

KML

KML is an XML-based language schema for expressing geographic annotation and visualization on existing or future Web-based, two-dimensional maps and three-dimensional Earth browsers. KML was developed for use with Google Earth and originally named Keyhole Earth Viewer. It was created by Keyhole, Inc., which was acquired by Google in 2004.

KML is now an international standard of the Open Geospatial Consortium. KML support is increasing rapidly. Many applications display KML, including Google Earth, Google Maps, Google Maps for mobile, NASA WorldWind, ESRI ArcGIS Explorer, Adobe PhotoShop, AutoCAD, and Yahoo! Pipes.

Listing 4 displays some sample KML data showing some restaurants as PlaceMarks.

Listing 4 Sample KML feed

```

<?xml version="1.0" encoding="UTF-8"?>

<kml xmlns="http://www.opengis.net/kml/2.2">

  <!-- A Placemark tag that represents a geographical entity -->
  <Placemark>

    <!-- Each Placemark tag contains attributes and other tags -->
    <name>Sushi Yasuda</name>
    <description>Great Sushi</description>
    <Point>
      <!-- Note that the order for coordinates is longitude, latitude and elevation -->
      <coordinates>-74.006393,40.714172,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>Blue Ribbon Sushi</name>
    <description>Festive atmosphere, excellent sake selection and great sushi</description>
    <Point>
      <coordinates>-74.12393,40.714321,0</coordinates>
    </Point>
  </Placemark>
</kml>

```

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/ferraro/>

Summary

After you decide on a content source, choose the best distribution format according to your deployment platform. If you have a lot of data that has to be downloaded to a mobile phone over unreliable data connections, choose a lighter format, such as GeoJSON.

Here are some other Manning titles you might be interested in:



[Objective-C for the iPhone](#)

Christopher K. Fairbairn and Collin Ruffenach



[iPhone in Practice](#)

Bear P. Cahill



[Unlocking Android, Second Edition](#)

W. Frank Ableson and Robi Sen

Last updated: April 13, 2011